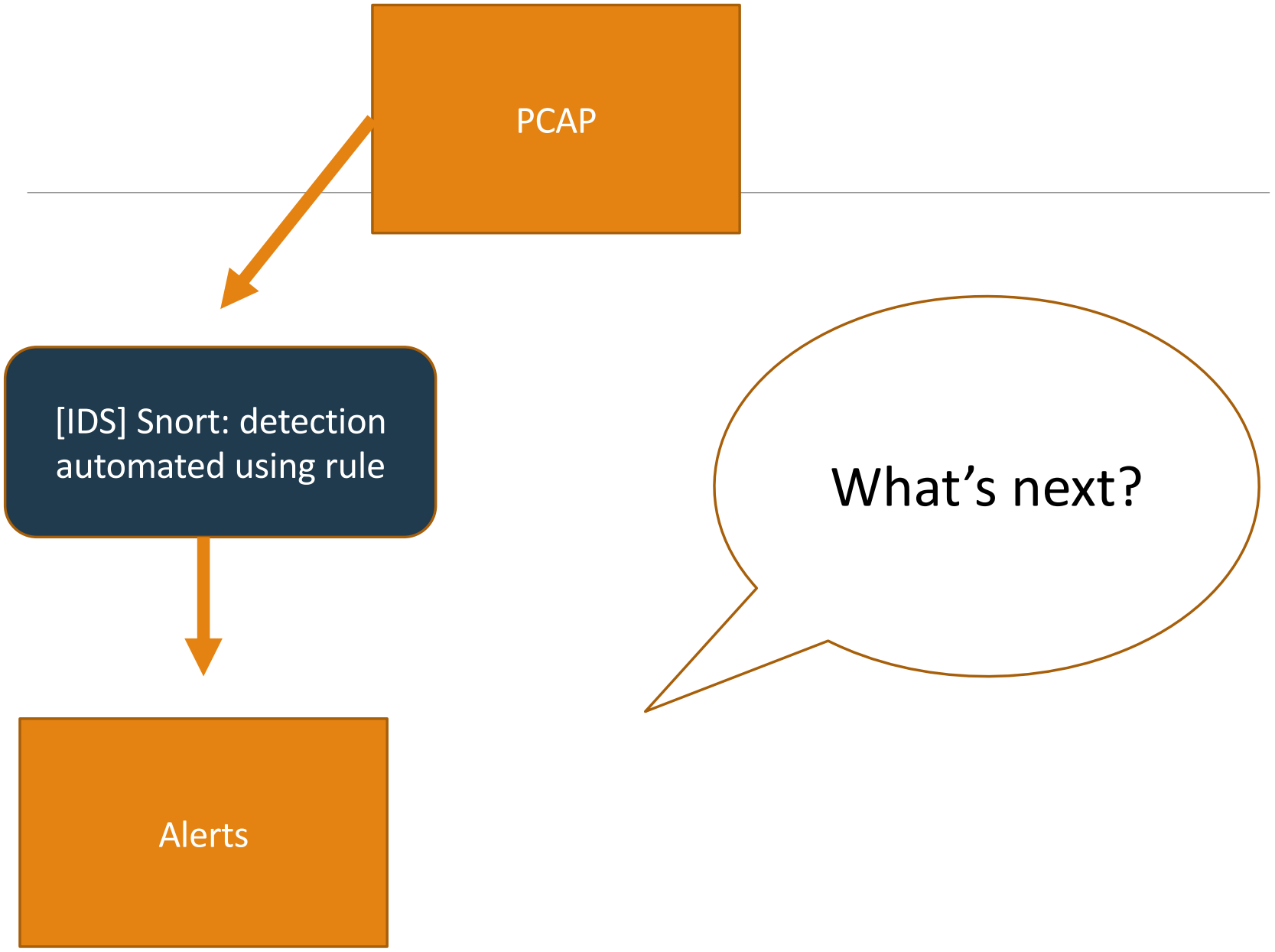
Basic Intrusion Analysis

TRAFFIC – EXAMPLES - ATTACKS

Learning outcome

1. Analyse port scanning and false positive
2. Analyse SYN scan and false positive
3. Telnet scan
4. IP spoofing
5. Denial of service
6. Client-side attack
7. Local port scanning

Stephen Northcutt, Judy Novak. *Network Intrusion Detection, Third Edition.* Sams Publishing 2002 (also available through Metalib, Safari Books Online: page 46, 65, 100, 274, 281 395, 396, 397, 433


PCAP
[IDS] Snort: detection automated using rule
What's next?
Alerts

Categories of traffic

Reconnaissance, exploits

False positives

Someone is “spoofing” your IP’s

Client-side attacks

Denial of Service

Reconnaissance, Exploits

Stages of attacks

- Network mapping: Mapping is to discover the live hosts in a target network.
- Identifying listening ports
- Breaking into a host

To detect these attacks:

- Look at TCP initial connections
- Look at malicious UDP port activity
- Look for scans (network scanning is a procedure for identifying active devices on a network)

Port scanning

Port scanning is a method of determining which ports on a network are open and could be receiving or sending data

- TCP SYN scanning
- TCP Connect scanning
- TCP ACK scanning
- TCP FIN scanning

Ref: De Vivo, Marco, et al. "A review of port scanning techniques." *ACM SIGCOMM Computer Communication Review* 29.2 (1999): 41-48.

TCP SYN scan (half-open scan)

TCP SYN scanning: sending a SYN packet to a selected port on the target machine.

- If the answer is RST (reset), the port is closed
- If the answer is SYN/ACK, the port is open and listening
- If no response received (even after retransmissions), probably a firewall has blocked/silenced the port
- How to detect: under SYN scanning, the connection will be not fully established (i.e. no ACK packet is sent). This technique is called as half-open scanning.
- Using nmap: https://nmap.org/book/synscan.html

Example: *nmap -p22,113,139 scanme.nmap.org*

nmap –d –packet-trace -p22,113,139 scanme.nmap.org

Port scanning example 1

```
scanner.net.4033 > mydns.com.220: S 93266:93266(0) win 8192
scanner.net.4003 > mydns.com.ftp: S 92918:92918 (0) win 8192
scanner.net.4005 > mydns.com.telnet: S 92946:92946 (0) win 8192
scanner.net.4004 > mydns.com.22: S 93094:93094(0) win 8192
scanner.net.4019 > mydns.com.110: S 93106:93106(0) win 8192
scanner.net.4020 > mydns.com.nntp: S 93076:93076(0) win 8192
scanner.net.4016 > mydns.com.88: S 93154:93154(0) win 8192
```

scanner.net is attempting to find listening ports on mydns.com (menti)

Exercise: write Snort rule to detect the above SYN scan

Port scanning example 2

```
218.233.173.60.2622 > 192.168.189.103.1433: S 3737503365:3737503365
  (0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

218.233.173.60.2654 > 192.168.189.135.1433: S 3739053261:3739053261
  (0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

218.233.173.60.2673 > 192.168.189.154.1433: S 3740070850:3740070850
  (0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

218.233.173.60.2689 > 192.168.189.170.1433: S 3740966222:3740966222
  (0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

218.233.173.60.2690 > 192.168.189.171.1433: S 3741051982:3741051982
  (0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
```

Activity generated by infected host 218.233.173.60, trying to identify vulnerable MSSQL servers. Worm SQLSnake or Spida (2002) looked for listening MSSQL servers. Once servers with no administrator password were found, a guess administrator account was created.

Port scanning example 3

```
Scanner.se.45820 > 192.168.209.5.23: S 4195942931:4195942935(4) win 4096
Scanner.se.45820 > 192.168.216.5.23: S 4195944723:4195944727(4) win 4096
Scanner.se.52526 > 172.16.68.5.23: S 357331986:357331990(4) win 4096
Scanner.se.45820 > 192.168.183.5.23: S 4196001810:4196001814(4) win 4096
Scanner.se.52526 > 172.16.248.5.23: S 357312531:357312535(4) win 4096
Scanner.se.45820 > 192.168.205.5.23: S 4196007442:4196007446(4) win 4096
Scanner.se.52526 > 172.16.250.5.23: S 357313043:357313047(4) win 4096
```

- Source ports are the same for different destination subnets
- Data on SYN

Ref: Stephen Northcutt, *Network Intrusion Detection, subsection A Telnet scan,* page 45

Exercise

1. Create 1 copy of the virtual machine
2. Use Vmware to open 2 similar virtual machine. We will use 1 machine as the target, and the other as the attacker
3. At the target, run “ifconfig” to identify the IP address
4. *At the target, run tcpdump to filter packets with SYN flag:*

sudo tcpdump 'tcp[tcpflags]==tcp-syn’

5. At the attacker, run “*nmap -p22,113,139 [ip address of the target machine]” to perform SYN scan*

TCP Connect scan (full scan)

TCP Connect scanning: establishing regular connections (i.e. using the full three-way handshake) with chosen ports on the target machine.

- How to detect: under Connect scanning, logs will report a rapid sequence of connections and error
- Using nmap: https://nmap.org/book/scan-methods-connect-scan.html

Example: *nmap -T4 -sT scanme.nmap.org*

TCP ACK scan

TCP ACK scanning: sending packets with only ACK flag set.

- If the answer is RST (reset), the port is reachable (i.e. unfiltered), but whether they are open or closed is undetermined
- If there is no answer (even after retransmissions) or ICMP unreachable error, the port is filtered (i.e. firewall has filtered the port)
- Using nmap: https://nmap.org/book/scan-methods-ack-scan.html
- Example: nmap -sA -T4 scanme.nmap.org

Port scanning example 4

ack.com.23 > 192.168.2.112.23: . ack 778483003 win 1028

ack.com.23 > 192.168.31.4.23: . ack 778483003 win 1028

ack.com.143 > 192.168.2.112.143: . ack 778483003 win 1028

ack.com.143 > 192.168.31.4.143: . ack 778483003 win 1028

ack.com.110 > 192.168.2.112.110: . ack 778483003 win 1028

ack.com.110 > 192.168.31.4.110: . ack 778483003 win 1028

ack.com.23 > 192.168.14.19.23: . ack 778483003 win 1028

ack.com.143 > 192.168.14.19.143: . ack 778483003 win 1028

ack.com.110 > 192.168.14.19.110: . ack 778483003 win 1028

ack.com.23 > 192.168.33.53.23: . ack 778483003 win 1028

ack.com.23 > 192.168.37.3.23: . ack 914633252 win 1028

ack.com.23 > 192.168.14.49.23: . ack 3631132968 win 1028

- No traffic is found from ack.com to indicate these ACK packets are reaction to some natural catalyst
- Source ports are the same as the destination ports

Ref: Stephen Northcutt, *Network Intrusion Detection, subsection An ACK scan,* page 43

FIN scan

```
stealthy.com.50141 > victim.org.5: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.3: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.26: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.45: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.17: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.7: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.51: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.52: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.30: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.53: F 0:0(0) win 4096 (DF)
stealthy.com.50141 > victim.org.20: F 0:0(0) win 4096 (DF)
```

nmap -sF -p21 target_ip address

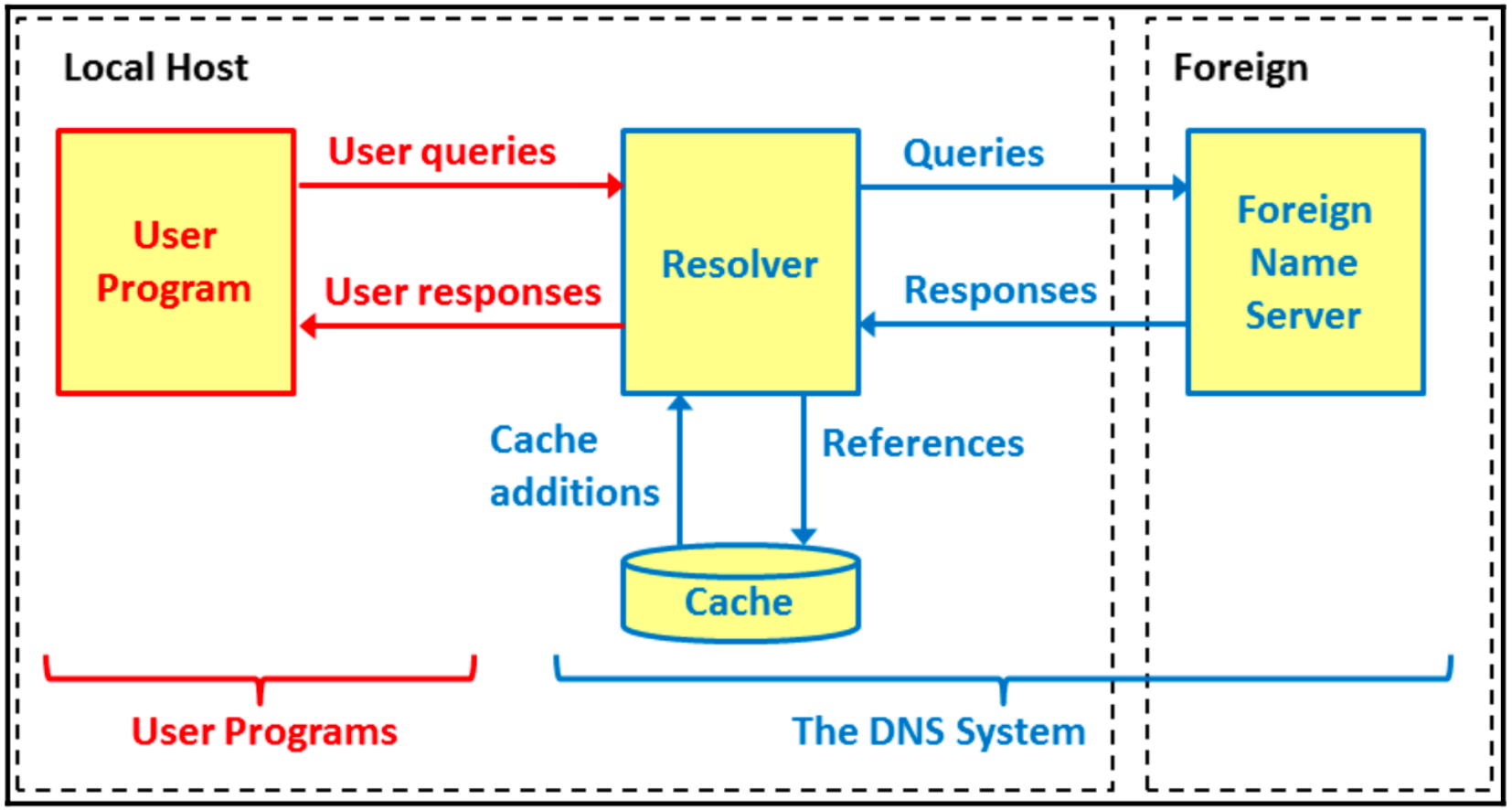
Exercise – UDP scan

1. Create 1 copy of the virtual machine
2. Use Vmware to open 2 similar virtual machine. We will use 1 machine as the target, and the other as the attacker
3. At the target, run “ifconfig” to identify the IP address
4. *At the target, run tcpdump to filter DNS query packets:*

sudo tcpdump port 53

5. At the attacker, run “sudo nmap -T4 –sU *[ip address of the target machine]” to perform UDP scan (by sending DNS packets)*

DNS operation



DNS query

Read Stephen Northcutt. *Network Intrusion Detection, page 113-116, subsection TCPdump output of resolution*

```
1. host.my.com.1716 > dns.my.com.53: 1+ (35)
2. dns.my.com.53 > h.root-servers.net.53: 12420 (30) (DF)
3. h.root-servers.net.53 > dns.my.com.53: 12420- 0/3/3 (153) (DF)
4. dns.my.com.53 > server1.sans.org.53: 12421+ (30) (DF)
5. server1.sans.org.53 > dns.my.com.53: 12421* 1/3/3 (172)
6. dns.my.com.53 > host.my.com.1716: 1* 1/3/3 (197) (DF)
```

- 1, 12420, 12421: DNS identification number which is used to match a query and a response
- (35), (30) (153) (172): size of UDP payload
- DF: don't fragment
- 0/3/3: 0 IP address was found, 3 authority records (i.e. referenced DNS servers), 3 additional records (i.e. IP addresses of the referenced DNS servers)
- 1/3/3: 1 IP address was found, 3 authority records (i.e. referenced DNS servers), 3 additional records (i.e. IP addresses of the referenced DNS servers)
- * : this is an authoritative response (i.e. IP address is found)

Ref: Stephen Northcutt. *Network Intrusion Detection, page 113-116, subsection TCPdump output of resolution*

DNS query (continued)

```
1. host.my.com.1716 > dns.my.com.53: 1+ (35)
2. dns.my.com.53 > h.root-servers.net.53: 12420 (30) (DF)
3. h.root-servers.net.53 > dns.my.com.53: 12420- 0/3/3 (153) (DF)
4. dns.my.com.53 > server1.sans.org.53: 12421+ (30) (DF)
5. server1.sans.org.53 > dns.my.com.53: 12421* 1/3/3 (172)
6. dns.my.com.53 > host.my.com.1716: 1* 1/3/3 (197) (DF)
```

1. host.my.com issues the request to resolve www.sans.org to dns.my.com
2. dns.my.com attempts a connection to h.root-servers.net on port 53.
3. h.root-servers.net does not have the answer to the query. It has a reference of another DNS server that either has the answer or has a reference of who might have the answer. h.root-servers referred dns.my.com to another DNS server
4. dns.my.com query the referenced server server1.sans.org
5. server1.sans.org happens to "own" the DNS record for www.sans.org and can return the IP address associated with www.sans.org to dns.my.com.

Ref: Stephen Northcutt. *Network Intrusion Detection, page 113-116, subsection TCPdump output of resolution*

False positive example 1

Following traffic appeared in logs

```
18:20:12:080000 dns.com.53 > myhost.com.1080: 5 NXDomain* 0/1/0 (128)

18:20:12:300000 dns.com.53 > myhost.com.1080: 6 NXDomain* 0/1/0 (119)

18:20:12:410000 dns.com.53 > myhost.com.1080: 7* 1/0/0 (48)
```

The NXDOMAIN (Non-existent Domain Name) is a DNS message type received by the DNS resolver (i.e. client) when a request to resolve a domain is sent to the DNS and cannot be resolved to an IP address

Further investigation shows that myhost.com initiated the activity

```
18:20:11:870000 myhost.com.1080 > dns.com.domain: 5+ (50)

18:20:12:090000 myhost.com.1080 > dns.com.domain: 6+ (41)

18:20:12:310000 myhost.com.1080 > dns.com.domain: 7+ (32)
```

Ref: Stephen Northcutt. *Network Intrusion Detection, page 395, subsection Scan or response*

False positive example 2?

IDS reported the following scan

shieldsup.grc.com.3663 > myhost.com.ftp: S 672038779:2672038779(0) win 8192 <mss 1460> (DF)
shieldsup.grc.com.3901 > myhost.com.telnet: S 2673332025:2673332025(0) win 8192 <mss 1460> (DF)
shieldsup.grc.com.4463 > myhost.com.finger: S 2675789141:2675789141(0) win 8192 <mss 1460> (DF)

It appears that the source host is performing a port scan by attempting to connect to the ftp, telnet, and finger ports in a short span of time

myhost.com initiated activity

```
15:09:07.938739 myhost.com.2471 > grc.com.https: S 168486550:1168486550(0) win 8192 <mss 1380> (DF)

15:09:28.189300 shieldsup.grc.com.netbios-ns > myhost.com.netbios-ns: udp 50

15:10:54.855900 shieldsup.grc.com.3663 > myhost.com.ftp: S 2672038779:2672038779(0) win 8192 <mss
   1460> (DF)

15:11:44.072390 shieldsup.grc.com.3901 > myhost.com.telnet: S 2673332025:2673332025(0) win 8192
   <mss 1460> (DF)

15:12:59.537989 shieldsup.grc.com.4463 > myhost.com.finger: S 2675789141:2675789141(0) win
   15:14:41.036441

15:14:41.036441 shieldsup.grc.com.1096 > myhost.com.pop3: S 2677222128:2677222128(0) win 8192 <mss
   1460> (DF)
```

myhost.com made a connection to grc.com destination to a secure web server port. This activity occurred before the alleged scan.

Shortly thereafter, it appears that a different host from the grc.com network initiates a scan on myhost.com.

Checking ports locally

How to check the listening ports and applications on Linux:

1.Open a terminal application i.e. shell prompt.

2.Run any one of the following command:

```
sudo lsof -i -P -n | grep LISTEN
sudo netstat -tulpn | grep LISTEN
sudo nmap -sTU -O IP-address-Here
```

Checking ports locally

How to check the listening ports and applications on Windows:

powershell

```
Get-Process -Id (Get-NetTCPConnection -LocalPort
portNumber).OwningProcess
```

cmd

```
 C:\> netstat -a -b
```

Someone is “spoofing” your IP’s

- Traffic entering your network
- Traffic appears to be a response to something
- No internal hosts initiated this activity

```
Router1 > 192.168.234.0: icmp: time exceeded in-transit [tos 0xc0]
Router1 > 192.168.126.0: icmp: time exceeded in-transit [tos 0xc0]
Router1 > 192.168.17.0: icmp: time exceeded in-transit [tos 0xc0]
Router1 > 192.168.113.0: icmp: time exceeded in-transit [tos 0xc0]
Router1 > 192.168.100.0: icmp: time exceeded in-transit [tos 0xc0]
Router2 > 172.16.118.0: icmp: time exceeded in-transit
Router2 > 172.16.125.0: icmp: time exceeded in-transit
```

Ref: Stephen Northcutt. *Network Intrusion Detection, page 100, subsection No stimulus, all response*

Denial of Service example

```
14: 18: 22.5660 flooder. 601 > server. login: S 1382726961:1382726961(0) win 4096
14: 18: 22.7447 flooder. 602 > server. login: S 1382726962:1382726962(0) win 4096
14: 18: 22.8311 flooder. 603 > server. login: S 1382726963:1382726963(0) win 4096
14: 18: 22.8868 flooder. 604 > server. login: S 1382726964:1382726964(0) win 4096
14: 18: 22.9434 flooder. 605 > server. login: S 1382726965:1382726965(0) win 4096
14: 18: 23.0025 flooder. 606 > server. login: S 1382726966:1382726966(0) win 4096
14: 18: 23.1035 flooder. 607 > server. login: S 1382726967:1382726967(0) win 4096
14: 18: 23.1621 flooder. 608 > server. login: S 1382726968:1382726968(0) win 4096
14: 18: 23.2284 flooder. 609 > server. login: S 1382726969:1382726969(0) win 4096
14: 18: 23.2825 flooder. 610 > server. login: S 1382726970:1382726970(0) win 4096
14: 18: 23.3457 flooder. 611 > server. login: S 1382726971:1382726971(0) win 4096
14: 18: 23.4083 flooder. 612 > server. login: S 1382726972:1382726972(0) win 4096
14: 18: 23.9030 flooder. 613 > server. login: S 1382726973:1382726973(0) win 4096
14: 18: 24.0052 flooder. 614 > server. login: S 1382726974:1382726974(0) win 4096
```

This is an actual SYN scan

Multiple new connection attempts against the same target

Ref: Stephen Northcutt. *Network Intrusion Detection, page 396, 281, 274*

SYN Scan False Positive

```
14: 18:22. 5166 host. 600 > server. 25: S 1382726960: 1382726960(0) win 4096
14: 18:22. 5660 host. 601 > server. 25: S 1382726961: 1382726961(0) win 4096
14: 18:22. 7447 host. 602 > server. 25: S 1382726962: 1382726962(0) win 4096
14: 18:22. 8311 host. 603 > server. 25: S 1382726963: 1382726963(0) win 4096
14: 18:22. 8868 host. 604 > server. 25: S 1382726964: 1382726964(0) win 4096
14: 18:22. 9434 host. 605 > server. 25: S 1382726965: 1382726965(0) win 4096
14: 18:23. 0025 host. 606 > server. 25: S 1382726966: 1382726966(0) win 4096
14: 18:23. 1035 host. 607 > server. 25: S 1382726967: 1382726967(0) win 4096
14: 18:23. 1621 host. 608 > server. 25: S 1382726968: 1382726968(0) win 4096
14: 18:23. 2284 host. 609 > server. 25: S 1382726969: 1382726969(0) win 4096
14: 18:23. 2825 host. 610 > server. 25: S 1382726970: 1382726970(0) win 4096
```

Very similar trace, yet this is false alert

Service 25/tcp is associated with email (SMTP)

When a mail server can't deliver emails the first time, it will keep trying every hour until it eventually gives up

Ref: Stephen Northcutt. *Network Intrusion Detection, page 397*

Client-side attacks

As security becomes tighter for servers, the next target is clients:

- Browsers (Internet Explorer, Mozilla)
- Image viewers
- Media players
- Audio players
- Microsoft Office

Often involve social engineering

- Entice user into clicking a link to a malicious site
- As a result, malicious file is downloaded to vulnerable client

Example 1 – Microsoft XP Office URL Link

February 2005 – "Microsoft Office XP HTML Link Processing Remote Buffer Overflow Vulnerability"[1]

Involves enticing a user to click on a link and download a Microsoft Office document with overly long URL

When Microsoft Office tries to open the file, the long URL causes a buffer overflow

Could result in DoS or code execution

1 http://www.securityfocus.com/bid/12480/info

Microsoft XP Office URL Link - Step 1

Subject: Lose weight fast!!!!

From: Help-u-Slimdown

Date: Feb. 10, 2005

To: Mr. Gullible

Click here to discover how to lose 5 pounds in 5 days!!!!

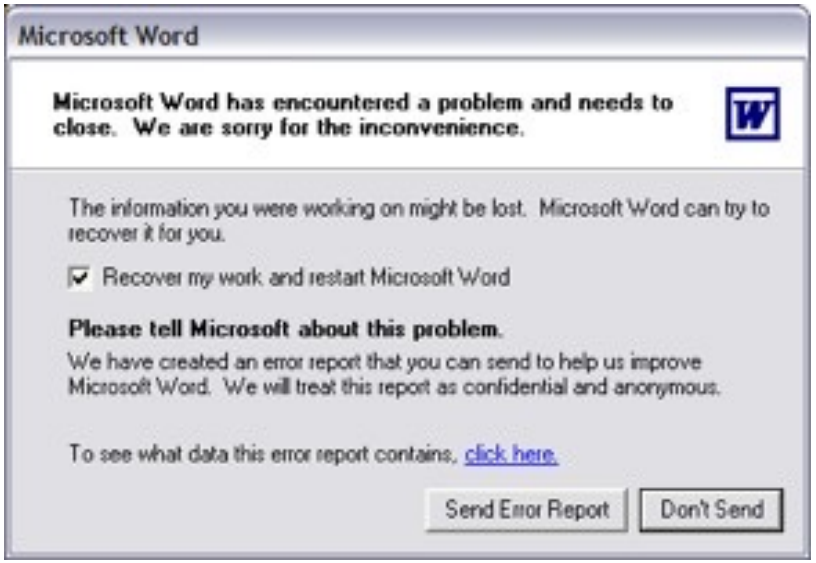
HTML source of the above link:

<html> <a href="http://10.4.12.23/AAAA

Microsoft XP Office URL Link –Step 2

10.4.12.23 listens on port 80 and returns the MS document

When Microsoft Office attempts to open the document, the long URL accompanies it, causing a buffer overflow.



Conclusions

knowledge about

- Protocols
- Applications
- IDS Technology
- Attacks

Intrusion Analysis is a labour intensive process

Requires in-depth extensive

Also requires knowledge of which traffic is normal for your own network or systems

Thank You

LECTURE 7 WILL BE ON INTRUSION ANALYSIS - WIRESHARK