Global Information Assurance Certification Paper

Practical Assignment

for

GIAC Intrusion Analyst Certification

Table of Contents

Assignment #1 - Network Traces

- **Detect #1**
- **Detect #2**
- **Detect #3**
- **Detect #4**
- **Detect #5**

Assignment #2 - Evaluate an Attack

Assignment #3 - "Analyze this" Scenario

Assignment #1 - Network Traces

Detect #1

```
Jun 30 15:09:27 195.186.212.130:2131 -> a.b.e.177:139 SYN **S*****
Jun 30 15:09:25 195.186.212.130:2136 -> a.b.e.182:139 SYN **S*****
Jun 30 15:09:25 195.186.212.130:2140 -> a.b.e.186:139 SYN **S*****
Jun 30 15:09:25 195.186.212.130:2141 -> a.b.e.187:139 SYN **S*****
Jun 30 15:09:25 195.186.212.130:2145 -> a.b.e.191:139 SYN **S*****
Jun 30 15:09:25 195.186.212.130:2146 -> a.b.e.192:139 SYN **S*****
Jun 30 15:10:56 195.186.212.130:2274 -> a.b.e.217:139 SYN **S*****
Jun 30 15:10:56 195.186.212.130:2277 -> a.b.e.220:139 SYN **S*****
Jun 30 15:10:56 195.186.212.130:2280 -> a.b.e.223:139 SYN **S*****
Jun 30 15:10:56 195.186.212.130:2281 -> a.b.e.224:139 SYN **S*****
```

1. Source of trace

http://www.sans.org/y2k/070600.htm

```
Jun 30 15:09:27 195.186.212.130:2131 -> a.b.e.177:139 SYN **S*****
       (1)             (2)           (3)        (4)     (5)      (6)
```

(1) - Time and Date stamp
(2) - Source IP address
(3) - Source port
(4) - Destination IP address
(5) - Destination port
(6) - Flags set on this TCP packet

2. Detect was generated by

TCPDump utility

3. Description of attack

Note: I have flip-flopped number 3 and number 4 since the probability of the source address being spoofed depends partially on the nature of the attack. For example, if the attack is simply a DoS, the source address is likely to be spoofed. However, if the attack requires a Three-way handshake, the source address is NOT likely to be spoofed.

Attacker was doing a Recon scan to see which servers have NetBIOS shares available. Its interesting to note that as the destination addresses are incrementing, so too are the source ports (ie. whenever there is a gap in the destination addresses, an equal gap exists in the source ports). Obviously, the attacker was doing a scan on the entire network, however, some packets seemed to have not been logged by the IDS.

This scan is well known and has even been listed as #7 in *SANS' Top 10 vulnerabilities*:

> **7. Global file sharing and inappropriate information sharing via NetBIOS and Windows NT ports 135->139 (445 in Windows2000), or UNIX NFS exports on port 2049, or Macintosh Web sharing or AppleShare/IP on ports 80, 427, and 548.**

4. Probability the source address was spoofed

The probability of the address being spoofed is very low. The attacker needs to receive a Syn-Ack packet back from the destination server in order to detect if the port is open. If its not open, a Reset-Ack packet would be sent to the source. Either way, the source address needs to receive something to determine if the port is open or closed.

5. Attack mechanism

By sending a Syn packet to all the addresses in the network, the attacker can determine who is 'advertising' their NetBIOS share. If the recipient sends back a Syn-Ack packet, the attacker has

a victim. If the recipient sends back a Reset-Ack packet, the attacker knows that the port is closed on that host.

6. Correlations
Scans to port 139 are very common especially nowadays with imbecile Cable Modem users running Windows leaving their entire hard drive shared for the world. However, this trace was taken from the GIAC website and I saw no similar traces elsewhere.

7. Evidence of active targeting
The attacker has no idea of his target. The intent behind the scan is to find a target that has the NetBIOS share open for exploit.

8. Severity
It's a little difficult to calculate the Severity since the network architecture is not known. However, I will make some assumptions and make the calculations based on those assumptions. I will assume that the some of the hosts have a NetBIOS share, however I will assume that the firewall and perimeter routers will not allow ports 137-139 in and out of the network.

(Criticality + Lethality) - (System countermeasures + Network countermeasures)
Criticality = nature of the systems is unknown. I will assume that most of the hosts are workstations and some are core NT servers. Since servers are being targeted, the criticality will be higher than normal - 4
Lethality = Getting access to a NetBIOS share is very dangerous. The attacker could potentially get access to critical files - 4
System Countermeasures = NT Servers that are sharing files need to open this port, however the best System countermeasure is to have strong usernames and passwords. If this is done properly the attacker will have a difficult time getting through - 4
Network Countermeasures = Due to the basic nature of this attack, many firewalls and routers deny any traffic on these ports - 5
Severity = -1 (as long as the base assumptions are true. Note that the scan itself is not dangerous. What is dangerous is the results of the scan which could lead to an all-out attack)

9. Defensive recommendation
In general, network scans are harmless, in and of themselves. However, since they are a good precursor to an attack, it is good to deny network scans. Programs like Psion Software's PORTSENTRY can look for scans and take effective actions against scanners. As mentioned above, the other way to combat this type of scan/attack is to simply not have the port open, if feasible. If the port must remain open, then TCP WRAPPERS can add an extra layer of security.

10. Multiple choice test question
On which port does NetBIOS Session Service operate?

A. 137
B. 138
C. 139
D. 140

Answer: C
Detect #2

Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.19:109 SYNFIN **SF****
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.32:109 SYNFIN **SF****

```
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.33:109 SYNFIN **SF****
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.62:109 SYNFIN **SF****
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.71:109 SYNFIN **SF****
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.80:109 SYNFIN **SF****
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.101:109 SYNFIN **SF****
Jul 5 10:43:50 212.210.111.68:109 -> a.b.c.114:109 SYNFIN **SF****
```

1. Source of trace

http://www.sans.org/y2k/070700.htm

2. Detect was generated by

TCPDump utility

3. Description of attack

Note: I have flip-flopped number 3 and number 4 since the probability of the source address being spoofed depends partially on the nature of the attack. For example, if the attack is simply a DoS, the source address is likely to be spoofed. However, if the attack requires a Three-way handshake, the source address is NOT likely to be spoofed.

Attacker was doing a Recon scan to see which servers have POP2 port open. The high speed at which the scan is taking place clearly points to a script. However, the destination addresses are not indicative of an entire network scan, unless the IDS dropped that many packets. Its also interesting to note that the attacker is using the Syn-Fin combo to try and bypass some older IDSes.

This scan is well known and has even been listed as #9 in *SANS' Top 10 vulnerabilities*:

9. IMAP and POP buffer overflow vulnerabilities or incorrect configuration.

Also, it has been listed as the following CVE entries:

CVE-1999-0005, CVE-1999-0006, CVE-1999-0042, CVE-1999-0920, CVE-2000-0091

4. Probability the source address was spoofed

The probability of the address being spoofed is very low. The attacker needs to receive a Syn-Fin-Ack packet back from the destination server in order to detect if the port is open. If its not open, a Reset-Ack packet would be sent to the source. Either way, the source address needs to receive something to determine if the port is open or closed.

5. Attack mechanism

By sending a Syn-Fin packet to the chosen addresses, the attacker can determine who has the POP2 port open. If the recipient sends back a Syn-Fin-Ack packet, the attacker knows the port is open. If the recipient sends back a Reset-Ack packet, the attacker knows that the port is closed on that host.

6. Correlations

Port scans on port 109 are very rare due to POP2's lack of usage. I would be very surprised to find this type of scan again.

7. Evidence of active targeting

The attacker has no idea of his target. The intent behind the scan is to find a target that has the POP2 open for exploit.

8. Severity

It's a little difficult to calculate the Severity since the network architecture is not known. However, I will make some assumptions and make the calculations based on those assumptions. I will assume that the some of the hosts have the POP2 port open. I will also assume that the firewall and perimeter routers allow traffic on ports 109 in and out of the network for their email users.

(Criticality + Lethality) - (System countermeasures + Network countermeasures)
Criticality = If a systems responds with a positive Syn-Fin-Ack, the assumption is that the servers is a POP2 server (ie. I have assumed that all non-mail servers have port 109 turned off). Since mail servers are being targeted, the criticality will be above average - 4
Lethality = Getting access to a POP2 port is dangerous. The attacker could potentially get root access if the system is improperly configured - 5
System Countermeasures = Mail servers that are providing mail to internal/external users need to open this port. The basic way to strengthen the system is to run the POP2 service as someone other than root. Another way around this is to enable encryption. Finally, there are patches to counter this potential attack - 4
Network Countermeasures = Since many users who access their mail are outside the network, the Firewalls and routers must allow this traffic through - 1
Severity = 4 (as long as the base assumptions are true. Note that the scan itself is not dangerous. What is dangerous is the results of the scan, which could lead to an all-out attack)

9. Defensive recommendation

Firstly, all hosts not needing to have port 109 open must make sure they are not advertising is as open. Second, those POP2 servers that require port 109 to be open must make sure that all the latest patches have been applied. Lastly, to ensure that no one sniffs the passwords, setup encryption of the passwords using SSL or Secure Shell.
The countermeasures listed under Detect #1 also apply - use PORTSENTRY and TCP WRAPPERS to monitor and combat port scans.

10. Multiple choice test question

What are Syn-Fin scans **primarily** used for?

A. Port scanning/Network mapping
B. DoS attack
C. OS Fingerprinting
D. Three-way handshakes

Answer: A

Detect #3

210.208.138.4 - - [09/May/2000:00:59:20 +0200] "GET /cgi-bin/php.cgi HTTP/1.0" 404 -

210.208.138.4 - - [09/May/2000:01:00:48 +0200] "GET /cgi-bin/ews/ews/architext_query.pl HTTP/1.0" 404 -
210.208.138.4 - - [09/May/2000:01:00:48 +0200] "GET /cgi-bin/jj HTTP/1.0" 404 -
210.208.138.4 - - [09/May/2000:01:00:48 +0200] "GET /cgi-bin/wwwboard.pl HTTP/1.0" 404 -
210.208.138.4 - - [09/May/2000:01:00:48 +0200] "GET /cgi-bin/ews/ews/architext_query.pl HTTP/1.0" 404 -
210.208.138.4 - - [09/May/2000:01:00:48 +0200] "GET /cgi-bin/jj HTTP/1.0" 404 -

1. Source of trace
http://www.sans.org/y2k/053000.htm

2. Detect was generated by
Utility unkown

3. Description of attack
Note: I have flip-flopped number 3 and number 4 since the probability of the source address being spoofed depends partially on the nature of the attack. For example, if the attack is simply a DoS, the source address is likely to be spoofed. However, if the attack requires a Three-way handshake, the source address is NOT likely to be spoofed.

Attacker is looking for weaknesses in the destination web server's CGI-scripts. If the web server replies to any of these attacks with a positive reply, the attacker can wreak havoc by remotely executing commands via root shell access or by capturing password files.

This scan is well known and has even been listed as #2 in *SANS' Top 10 vulnerabilities*:

> **2. Vulnerable CGI programs and application extensions (e.g., ColdFusion) installed on web servers.**

Also, it has been listed as the following CVE entries:

> **CVE-1999-0279 - Excite for Web Servers (EWS) allows remote command execution via shell metacharacters.**
> **CVE-1999-0058 - Buffer overflow in PHP cgi program, php.cgi allows shell access.**
> **CVE-1999-0260 - The jj CGI program allows command execution via shell metacharacters.**
> **CVE-1999-0953 - WWWBoard stores encrypted passwords in a password file that is under the web root and thus accessible by remote attackers.**

4. Probability the source address was spoofed
The probability of the address being spoofed is very low. The attacker needs to get the response. The intent is not DoS, but complete exploitation of the target system.

5. Attack mechanism
Depending on the web server application, certain CGI-scripts can act as backdoors whereby attackers can victimize a web server. The CGI scripts can be executed with administrator/root privileges to do just about anything the attacker wishes. The attack is carried on either the default scripts that are installed with the Web server program or using poorly programmed code.

6. Correlations
This trace was obtained from GIAC web site and I saw no other related attacks. However, I'm sure this type of attack is very popular with script kiddies since it easily checks web server for CGI script (and other web apps) weaknesses.

7. Evidence of active targeting
The attacker has decided to attack this specific server (210.208.138.4) after having determined that it is listening on port 80 (ie. it's a web server).

8. Severity
It's a little difficult to calculate the Severity since the actual status of the web server is not known. However, I will make some assumptions and make the calculations based on those assumptions. I will assume that the web server has all the patches to defend against these specific attacks (ie. simply installing all applicable patches).

(Criticality + Lethality) - (System countermeasures + Network countermeasures)
Criticality = Web servers are the 'front end' to many businesses and so an attack on a web server is critical to a business's image. However, normal companies do not place business critical applications and data on web servers, so it doesn't have the same level of criticality as a File server or DNS server. - 3
Lethality = The attacker is checking for potentially crippling weaknesses in the web server. If any of the scripts get executed, it could mean total access to the web server. - 5
System Countermeasures = Web Admins must be very familiar with what is running on their servers. Poorly written scripts as well as the default scripts installed with the Web server program can open up backdoors for hackers. If all scripts are accounted for, the system is very secure. - 4
Network Countermeasures = Network security plays a minor role here since the attack is taking place at the application layer - 1
Severity = 3 (as long as the base assumptions are true. Note that the scan itself is not dangerous. What is dangerous is the results of the scan, which could lead to an all-out attack)

9. Defensive recommendation
Since I can not tell if the target system was victimized, I can not say if the Web admin did a good job of securing the system. Thus, my suggestions will be generic and applicable to any web server. First, make sure that all sample, default CGI scripts are removed or disabled. Second, apply all patches to the Web server program. Also, make sure that the web server is not executed as root. Finally, check and recheck all CGI scripts to make sure they are not vulnerable to attacks.

Multiple choice test question

When looking for web server attacks, what layer must be analyzed?

A. Network Layer
B. Data Link Layer
C. Transport Layer
D. Application Layer

Correct Answer: D

Detect #4

Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 12345

Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 12345
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 12346
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 12346
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 20034
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 20034
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 31337
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 31337
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 40421
Jun 20 01:46:11 stealth portsentry[190]: attackalert: Connect from host:
195.clearwater-03-04rs.fl.dial-access.att.net/12.77.207.195 to TCP port: 40421

1. Source of trace
http://www.sans.org/y2k/063000-1400.htm

2. Detect was generated by
Portsentry program

3. Description of attack
Note: I have flip-flopped number 3 and number 4 since the probability of the source address being spoofed depends partially on the nature of the attack. For example, if the attack is simply a DoS, the source address is likely to be spoofed. However, if the attack requires a Three-way handshake, the source address is NOT likely to be spoofed.

This attack is very interesting. The attacker is trying to connect to well-known Trojan ports. PORTSENTRY has detected this attempt and if setup properly, it will deny future access to the source address. The attack is very dangerous in that it may totally open a system to root/administrator access.

The ports being attacked are well known Trojan ports. Port 12345 and 12346 are the default ports for NetBus 1.x, Port 20034 is used for NetBus Pro, port 31337 is obviously Back Orifice. Finally, port 40421 is used for a trojan called Masters Paradise.

4. Probability the source address was spoofed
The probability of the address being spoofed is very low. The attacker is attempting to setup a full-fledged communication link with the target system. If the attacker gets through, he/she will control the system. Since the attack is to a Trojan port, detection is not a threat to the attacker. Once connection is established, the attacker can do as he/she wishes without being detected.

5. Attack mechanism
Trojans operate in the background of target systems and allows remote users to access and control the system. Once installed on a target system, the systems owner will never know of its existence unless special Trojan scanning programs are used (or sometime a simple NETSTAT.EXE

can show open ports). Finally, one common usage of these Trojans is to install other potentially damaging progams (other lesser-known Trojans, FTP or HTTP servers, or create shares).

6. Correlations
This trace was obtained from GIAC web site and I saw no other related attacks. However, I'm sure this type of attack is very popular with script kiddies since it quickly checks for any open Trojan ports.

7. Evidence of active targeting
The attacker has decided to attack this specific host hoping to strike 'Trojan' gold with one of the targeted ports.

8. Severity
I will make some assumptions and make the calculations based on those assumptions. I will assume that the target system does not have a Trojan running on it. I will also assume that the Firewall allowed this traffic through.

(Criticality + Lethality) - (System countermeasures + Network countermeasures)
Criticality = The nature of this target system is unknown. It could be a DNS or HTTP server or it could be a secretary's workstation - 3
Lethality = A Trojan is extremely dangerous and if found by an attacker, the security becomes totally compromised. Full control of a system by an attacker is NOT a good thing - 5
System Countermeasures = Based on the assumption above that the system is NOT running any Trojans software, the system will be immune to this attack. Plus, the fact that PORTSENTRY was running on the system adds another level of security - 5
Network Countermeasures = If the assumption is made that the system lies behind the firewall, it can be concluded that the firewall did not deny traffic to these well-known Trojan ports. It is a good idea to deny access to these ports - 2
Severity = 1 (as long as the base assumptions are true)

9. Defensive recommendation
Firstly, all systems should be scanned for Trojans. There are many programs out there that can be used to check for well-known Trojans. In addition, NETSTAT can show if any unknown ports are open. Running PORTSENTRY is a good way to detect attacks, but it would be better to deny all traffic to these ports at the firewall.

10. Multiple choice test question
What utility can be used to detect if backdoor ports are open on your system?

A. tcpdump
B. nbtstat
C. netstat
D. rexec

Answer: C

Detect #5

12:51:23.900575 xxx.xxx.53.110.1457 > xxx.xxx.53.72.137: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:23.900691 xxx.xxx.53.72.137 > xxx.xxx.53.110.1457: R 0:0(0) ack 49151 win 0
12:51:24.332267 xxx.xxx.53.110.1457 > xxx.xxx.53.72.137: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:24.332336 xxx.xxx.53.72.137 > xxx.xxx.53.110.1457: R 0:0(0) ack 1 win 0
12:51:24.832973 xxx.xxx.53.110.1457 > xxx.xxx.53.72.137: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:24.833043 xxx.xxx.53.72.137 > xxx.xxx.53.110.1457: R 0:0(0) ack 1 win 0
12:51:25.333680 xxx.xxx.53.110.1457 > xxx.xxx.53.72.137: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:25.333762 xxx.xxx.53.72.137 > xxx.xxx.53.110.1457: R 0:0(0) ack 1 win 0
12:51:29.294018 xxx.xxx.53.110.1464 > xxx.xxx.53.72.138: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:29.294116 xxx.xxx.53.72.138 > xxx.xxx.53.110.1464: R 0:0(0) ack 49151 win 0
12:51:29.739891 xxx.xxx.53.110.1464 > xxx.xxx.53.72.138: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:29.739969 xxx.xxx.53.72.138 > xxx.xxx.53.110.1464: R 0:0(0) ack 1 win 0
12:51:30.240586 xxx.xxx.53.110.1464 > xxx.xxx.53.72.138: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:30.240654 xxx.xxx.53.72.138 > xxx.xxx.53.110.1464: R 0:0(0) ack 1 win 0
12:51:30.741298 xxx.xxx.53.110.1464 > xxx.xxx.53.72.138: S 49150:49150(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:30.741388 xxx.xxx.53.72.138 > xxx.xxx.53.110.1464: R 0:0(0) ack 1 win 0
12:51:33.630037 xxx.xxx.53.110.1471 > xxx.xxx.53.72.139: S 49154:49154(0) win 8192 <mss 1460> (DF) [tos 0x10]
12:51:33.630162 xxx.xxx.53.72.139 > xxx.xxx.53.110.1471: S 42414:42414(0) ack 49155 win 8760 <mss 1460> (DF)
12:51:33.630284 xxx.xxx.53.110.1471 > xxx.xxx.53.72.139: . ack 1 win 8760 (DF) [tos 0x10]
12:51:33.639266 xxx.xxx.53.110.1471 > xxx.xxx.53.72.139: P 1:4(3) ack 1 win 8760 urg 3 (DF) [tos 0x10]
12:51:33.639367 xxx.xxx.53.72.139 > xxx.xxx.53.110.1471: FP 1:6(5) ack 4 win 8758 (DF)
12:51:33.639514 xxx.xxx.53.110.1471 > xxx.xxx.53.72.139: . ack 7 win 8755 (DF) [tos 0x10]
12:51:33.643577 xxx.xxx.53.110.1471 > xxx.xxx.53.72.139: R 49158:49158(0) win 0 (DF) [tos 0x10]

1. Source of trace
My personal LAN

2. Detect was generated by
WinDump program

3. Description of attack
Note: I have flip-flopped number 3 and number 4 since the probability of the source address being spoofed depends partially on the nature of the attack. For example, if the attack is simply a DoS, the source address is likely to be spoofed. However, if the attack requires a Three-way handshake, the source address is NOT likely to be spoofed.

This is the sign of the well-known WinNuke attack. The attacker is attempting to connect on one of the three NetBIOS ports 137-139 and send on OOB nuke. As can be seen from the trace, the

attacker failed to connect to ports 137 and 138, but successfully connected to port 139. Once connected, the attacker sends a single Push-Urg packet.

This attack has been listed as the following CVE entry:

CVE-1999-0153 - Windows 95/NT out of band (OOB) data denial of service through NETBIOS port, aka WinNuke.

4. Probability the source address was spoofed

The probability of the address being spoofed is very low. The attacker is attempting to test port availability by setting up a three-way handshake. To fully accomplish his/her objective of sending a single Push-Urg packet, the attacker needs to successfully receive packets from the victim.

5. Attack mechanism

When a Windows system receives a packet with the "URGENT" flag set, it expects data will follow that flag. The exploit consists of setting the URGENT flag, but not following it with data. The port most susceptible is TCP Port 139, the Netbios Session Service port. Although port 139 is the most commonly attacked port, there is potential for successful attacks on other ports as well. This attack is effective remotely or locally (it also works on the machine it's executing from). When Windows NT is successfully attacked, it crashes. The system displays the "blue screen of death", and is not respondent. Except for losing the contents of unsaved documents and files, there are no long-lasting effects from this attack.

6. Correlations

Since this trace was generated, there is no correlation information

7. Evidence of active targeting

The attacker has picked this specific host to target hoping it isn't patched above SP3.

8. Severity

I will make some assumptions and make the calculations based on those assumptions. I will assume that the target NT system has SP3 or above loaded. I will also assume that the Firewall denies all traffic to port 139, therefore the traffic must have originated from within the network (actually, this is not really an assumption since I recreated this attack from within my LAN).

(Criticality + Lethality) - (System countermeasures + Network countermeasures)
Criticality = The nature of this target system is unknown. It could be a DNS or HTTP server or it could be a secretary's workstation - 3
Lethality = If the WinNuke exploit accomplishes what it is intended to do, the end result is more of an annoyance than a true hack. The NT box will crash and simply require a reboot. No threat to applications or data running on the server - 2
System Countermeasures = If the NT box is patched at SP3 or above, as most servers are nowadays, the attack will be easily rejected - 5
Network Countermeasures = Most firewall do not allow traffic on ports 137-139 anyway, so for the attempt to be successful, it must originate from within the firewall - 5
Severity = -5 (as long as the base assumptions are true)

9. Defensive recommendation

Not very much to recommend except to make sure that all NT servers have the latest SP. Also, it would be good to confirm that the Firewall is denying traffic to ports 137-139.

10. Multiple choice test question

Which of the following descriptions best characterizes the WinNuke attack?

A. Flood of spoofed Syn packets
B. Packet sent with Urg bit set followed by a packet with the Fin bit set
C. Fragments with overlapping offsets
D. The Destination address and port is the same as the Source address and port

Answer: B

Assignment #2 - Evaluate an Attack

1. Exploit name and source
'IGMP Nuker' from http://www.xcoder.com/xproducts.asp

2. Exploit description
The IGMP Nuker is a Denial of Service attack. The attempt is to crash the target system by sending oversized IGMP packets. The attacker is hoping to catch an OS that does not understand how to process a large group of oversized IGMP packets.

3. Exploit trace
```
12:06:03.714973 XXX.XXX.219.148 > XXX.XXX.219.139: igmp-0 [v15][|igmp] (frag
9738:1480@0+)
12:06:03.715094 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@1480+)
12:06:03.715217 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@2960+)
12:06:03.715340 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@4440+)
12:06:03.715462 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@5920+)
12:06:03.715584 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@7400+)
…[snipped]…
12:06:03.719405 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@53280+)
12:06:03.719523 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@54760+)
12:06:03.719647 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@56240+)
12:06:03.719775 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:1480@57720+)
12:06:03.719827 XXX.XXX.219.148 > XXX.XXX.219.139: (frag 9738:800@59200)
```

Using Windump (http://netgroup-serv.polito.it/windump) on the target NT host, I recorded this IGMP Nuker attack. Due to the oversized IGMP Packet (60,000 bytes), the packet gets fragmented into smaller 1480 byte chunks (typical size for an Ethernet segment). The fragmentation itself is completely normal (ie. no overlapping ala Teardrop or open gaps or oversized packets (>65535)). Also, the time is noteworthy since the attack is attempting to flood the target system. The IGMP Nuker allows the attacker to send upto 1000 packets in a matter of seconds.

Another key feature of this type of attack is that most IDSes do not track IGMP packets, much less fragmented packets. So, an attacker trying to bring down an target system may be able to 'hide' via IGMP fragmented packets.

The countermeasure to this attack is to prevent any outside multicast traffic destined to any internal unicast addresses. A filter like that will prevent any such attacks since normal IGMP traffic is destined for multicast addresses. Also, if possible, this attack can be simply detected by any IDS by looking for IGMP packets with MF (More Fragments) bit set. Finally, at the system level, personal firewall systems such as WinRoute (by Tiny Software) can be installed to prevent any IGMP packets from hitting the protected host. This can be used to defend against internal attacks, which is a serious possibility since the Firewall and IDS are setup to detect outside attacks.

Assignment 3 - "Analyze This" Scenario

1. At first glance at the snort traces, it is very obvious that outside users are proxying through an internal Wingate server. This is dangerous since it allows outside users to anonymously surf the web while displaying your proxy server's IP address.

[snipped]
05/24-14:26:33.564233 [**] WinGate 8080 Attempt [**] 136.160.4.159:1982 - MY.NET.253.105:8080
05/24-14:26:35.066971 [**] WinGate 8080 Attempt [**] 136.160.4.159:1989 - MY.NET.253.105:8080
05/24-14:26:35.627300 [**] WinGate 8080 Attempt [**] 136.160.4.159:1992 - MY.NET.253.105:8080
05/24-14:26:36.490496 [**] WinGate 8080 Attempt [**] 136.160.4.159:1999 - MY.NET.253.105:8080
05/24-14:26:36.807356 [**] WinGate 8080 Attempt [**] 136.160.4.159:2002 - MY.NET.253.105:8080
05/24-14:26:40.783081 [**] WinGate 8080 Attempt [**] 136.160.4.159:2032 - MY.NET.253.105:8080
05/24-14:26:41.065141 [**] WinGate 8080 Attempt [**] 136.160.4.159:2034 - MY.NET.253.105:8080
[snipped]

The simplest way to combat this type of abuse is to properly setup the Wingate proxy to only accept traffic from the internal interface of the proxy server.

2. Next I noticed that outside users are coming in to inside hosts on port 137, NetBIOS name service:

05/24-20:53:41.680590 [**] SMB Name Wildcard [**] 166.90.30.149:137 - MY.NET.100.130:137
05/24-20:53:43.074002 [**] SMB Name Wildcard [**] 166.90.30.149:137 - MY.NET.100.130:137
05/24-20:53:43.074063 [**] SMB Name Wildcard [**] 166.90.30.149:137 - MY.NET.100.130:137
05/24-20:53:46.531240 [**] SMB Name Wildcard [**] 166.90.30.149:137 - MY.NET.100.130:137

This is a sure sign of poor Firewall configuration. Although the 'attack' in this case seems to be simply a query for name, domain, and other NT information (since that is what port 137 is used for), the attacker is most probably collecting information for a future attack. Therefore, it would be most wise to deny any traffic on ports 137-139 on the Firewall.

3. Noticed that snort was running the Port Preprocessor. It seems that it picked up the portscans and recorded them properly. Although, I might question the threshold setting of 7 connections in 2 seconds depending on how busy are the server and network. It may be leading to some false positives:

05/24-18:35:21.363973 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.1.3 (THRESHOLD 7 connections in 2 seconds) [**]
05/24-18:35:23.555804 [**] spp_portscan: portscan status from MY.NET.1.3: 10 connections across 1 hosts: TCP(0), UDP(10) [**]
05/24-18:35:25.484551 [**] spp_portscan: End of portscan from MY.NET.1.3 (TOTAL HOSTS:1 TCP:0 UDP:10) [**]

[snip]

05/24-19:16:33.152446 [**] spp_portscan: PORTSCAN DETECTED from 132.250.1.131 (THRESHOLD 7 connections in 2 seconds) [**]
05/24-19:16:34.282076 [**] spp_portscan: portscan status from 132.250.1.131: 12 connections across 1 hosts: TCP(0), UDP(12) [**]
05/24-19:16:35.050360 [**] spp_portscan: End of portscan from 132.250.1.131 (TOTAL HOSTS:1 TCP:0 UDP:12) [**]
05/24-19:16:35.634488 [**] spp_portscan: PORTSCAN DETECTED from 132.250.1.131 (THRESHOLD 7 connections in 2 seconds) [**]
05/24-19:16:36.293390 [**] spp_portscan: portscan status from 132.250.1.131: 14 connections across 1 hosts: TCP(0), UDP(14) [**]
05/24-19:16:36.962342 [**] spp_portscan: End of portscan from 132.250.1.131 (TOTAL HOSTS:1 TCP:0 UDP:14) [**]

[snip]

05/24-19:18:41.589848 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.1.3 (THRESHOLD 7 connections in 2 seconds) [**]
05/24-19:18:42.532695 [**] spp_portscan: portscan status from MY.NET.1.3: 8 connections across 1 hosts: TCP(0), UDP(8) [**]
05/24-19:18:44.236037 [**] spp_portscan: End of portscan from MY.NET.1.3 (TOTAL HOSTS:1 TCP:0 UDP:8) [**]

Another point, the portscans detected from the internal host, MY.NET.1.3, signal that the snort detector should be setup to only detect scans from the outside network. The scans from the internal network are false positive maybe pointing to an NFS server taking multiple UDP connections.

4. After tracking down some of the sources that are popping up heavily on the traces, I found that the 159.226.0.0 Class B domain is registered to "The Computer Network Center Chinese Academy of Sciences (NET-NCFC)" and the 212.179.XX.0 Class C Domain is registered to what seesms as mainly ISP's in Israel. It would be a good idea to filter against these domains since these sources are probably not friendly traffic. Also, the fact that they are targeting port 25 on the target host, the intent is definitely not good.

05/23-15:44:27.025108 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289
05/23-15:44:27.810159 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289
05/23-15:44:28.213391 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289
05/23-15:44:29.691547 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289
05/23-15:44:29.691594 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289
05/23-15:44:29.832505 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289
05/23-15:44:31.781434 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.26.233:6700 - MY.NET.203.194:1289

05/23-02:57:58.933543 [**] Watchlist 000222 NET-NCFC [**] 159.226.21.134:3352 - MY.NET.6.35:25
05/23-03:27:49.534828 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:2018 - MY.NET.253.43:25
05/23-03:27:52.421919 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:2018 - MY.NET.253.43:25
05/23-03:27:55.696954 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:2018 - MY.NET.253.43:25
05/23-03:43:50.800325 [**] Watchlist 000222 NET-NCFC [**] 159.226.5.222:1693 - MY.NET.100.230:113
05/23-03:44:01.663962 [**] Watchlist 000222 NET-NCFC [**] 159.226.5.222:25 - MY.NET.100.230:48800
05/23-03:44:03.236705 [**] Watchlist 000222 NET-NCFC [**] 159.226.5.222:25 - MY.NET.100.230:48800
05/23-03:44:08.205004 [**] Watchlist 000222 NET-NCFC [**] 159.226.5.222:25 - MY.NET.100.230:48800

5. The traffic capture below is alerting of an Attempted Sun RPC high port access. Normal Sun RPC operates at port 111, however, recently some UNIX flavors run RPCs at ports around 32771. Firewalls may not always deny traffic to these ports, therefore its important to check that the proper rules are set on the Firewall. However, on the contrary, this could just as well be a false positive. The 4000 source port is also the default port of ICQ. Snort may have recorded the inbound portion of the ICQ communication with a host inside our network.

05/28-01:53:14.374269 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771
05/28-01:54:14.314767 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771
05/28-01:55:14.252511 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771
05/28-01:55:45.039233 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771
05/28-01:56:14.193962 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771
05/28-01:57:14.164986 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771
05/28-01:58:14.121833 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771

6. The pattern below leads me to believe that an FTP server is being attacked. The FTP server's permission list should be checked to make sure proper rights are assigned and that security has not been compromised. If an FTP server is required to be running, another suggestion is to place it in a DMZ, so if it ever does get attacked, the attacker can't get to the corporate network.

06/19-03:43:26.790157 [**] Watchlist 000222 NET-NCFC [**] 159.226.91.59:46162 - MY.NET.100.165:21
06/19-03:43:37.616276 [**] Watchlist 000222 NET-NCFC [**] 159.226.91.59:46162 - MY.NET.100.165:21
06/19-03:43:37.616713 [**] Watchlist 000222 NET-NCFC [**] 159.226.91.59:46162 - MY.NET.100.165:21
06/19-03:43:38.324196 [**] Watchlist 000222 NET-NCFC [**] 159.226.91.59:46162 - MY.NET.100.165:21

7. Finally, the SNMP public access alert is worth looking into. The most common SNMP attack is caused by poor passwords such as 'public' and 'private'. Therefore, it would be wise to confirm that the public and private community strings have good passwords on any network resources running SNMP.

05/23-13:38:40.170513 [**] SNMP public access [**] MY.NET.97.133:1131 - MY.NET.101.192:161
05/23-13:38:41.770327 [**] SNMP public access [**] MY.NET.97.133:1132 - MY.NET.101.192:161