Global Information Assurance Certification Paper

GIAC Intrusion Detection Curriculum Practical Assignment

James Haywood

Assignment 1 Network Detects

Detect 1

[**] MISC-DNS-version-query [**]
09/12-12:15:58.648953 badguy.com:61340 -> dns1.net:53
UDP TTL:48 TOS:0x0 ID:41062
Len: 38
E8 BC 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.....

[**] MISC-DNS-version-query [**]
09/12-12:18:20.413098 badguy.com:62592 -> dns2.net:53
UDP TTL:48 TOS:0x0 ID:61527
Len: 38
63 2E 01 00 00 01 00 00 00 00 00 00 07 76 65 72 c.............ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03   sion.bind.....

1. Source of trace:
    My Network

2. Detect was generated by:
    Snort intrusion detection system.
    Fields:
[**] **MISC-DNS-version-query** [**] [Rule Matched]
09/12-12:18:20.413098 [Date and time] **badguy.com:62592** [Source IP and port] -> **dns2.net:53** [Destination IP and port] **UDP** [Protocal] **TTL:48** [Time-to-live] **TOS:0x0** [Type of service] **ID:61527** [Packet ID] **Len: 38** [Packet length]
63 2E 01 00 00 01 00 00 00 00 00 00 07 76 65 72 c.............ver
73 69 6F 6E 04 62 69 6E 64 00 00 10 00 03   sion.bind..... [Data]

3. Probability the source address was spoofed:
Low - This is an informational probe so the attacker would need a response back.

4. Description of attack:
The older versions of BIND have buffer overflow vulnerabilities that may allow Denial of Service, Cache poisoning and even execution of commands. The first step in this type of attack is to find out which version of BIND the victims DNS server is running.

CVE's these address this type of attack
CVE-1999-0009
CVE-1999-0010
CVE-1999-0835
CVE-1999-0848
CVE-1999-0849
CVE-1999-0851
CVE-1999-0824

5. Attack mechanism:

The attacker queries the DNS server with domain name ="version.bind", a querytype ="TXT" and a class = "chaos". This type of query causes a vulnerable DNS to respond with the version of BIND that is running on it. With this information, the attacker can determine if the DNS server is running a vulnerable version of BIND.

6. Correlations:

This attack is listed as one of the top ten

7. Evidence of active targeting:

The only two systems that got the probes were DNS systems so there are some signs of active targeting.

8. Severity:

(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Criticality – DNS Server = 2
Lethality - Just a probe = 1
Systems Countermeasures – System has all patches = 5
Network Countermeasures – The firewall allows DNS requests and responses = 3
(2 + 1) – (5 + 3) = -1

9. Defensive recommendation:

Run the latest version of BIND 8.02 patch level 5. Configure BIND not to return the version level for a DNS version request.

10. On UNIX what program is used to run DNS?

A. Sunrpc
B. nslookup
C. BIND
D. Apache

Answer: C

Detect #2

04:44:47.524709 194.90.210.93.1541 > a.b.131.241.http: S
148350153:148350153(0) win 8192 (DF)
04:45:05.487390 194.90.210.93.1541 > a.b.131.241.http: S
148350153:148350153(0) win 8192 (DF)
04:45:11.486419 194.90.210.93.1541 > a.b.131.241.http: S
148350153:148350153(0) win 8192 (DF)
04:45:23.483576 194.90.210.93.1541 > a.b.131.241.http: S
148350153:148350153(0) win 8192 (DF)
04:45:47.487391 194.90.210.93.1586 > a.b.131.241.8080: S
148395559:148395559(0) win 8192 (DF)
04:45:50.484222 194.90.210.93.1586 > a.b.131.241.8080: S
148395559:148395559(0) win 8192 (DF)
04:45:56.482801 194.90.210.93.1586 > a.b.131.241.8080: S
148395559:148395559(0) win 8192 (DF)
04:46:08.480777 194.90.210.93.1586 > a.b.131.241.8080: S
148395559:148395559(0) win 8192 (DF)
04:46:32.481807 194.90.210.93.1626 > a.b.131.241.3128: S
148440556:148440556(0) win 8192 (DF)
04:46:35.473232 194.90.210.93.1626 > a.b.131.241.3128: S
148440556:148440556(0) win 8192 (DF)
04:46:41.471978 194.90.210.93.1626 > a.b.131.241.3128: S
148440556:148440556(0) win 8192 (DF)
04:46:53.473925 194.90.210.93.1626 > a.b.131.241.3128: S
148440556:148440556(0) win 8192 (DF)

1. Source of the trace:
My Network

2. Detect was generated by:
Shadow
Fields:
04:46:41.471978 **[Time]** 194.90.210.93.1626 **[Source IP and Source port] >** a.b.131.241.3128 **[Destination IP and Destination Port]**: S **[TCP Flags]** 148440556:148440556(0) **[Sequence Number]** win 8192 **[Window Size]** (DF)

3. Probability the source was spoofed:
High – There are signs that there are crafted packets. Examining the packets revealed that all for each port the attacker used the same sequence number for each of three tries per port and we know that sequence numbers are supposed to

4. Description of the attack:
The attacker is searching for well know proxy ports. This type of activity is associated with the Ring Zero Trojan.

5. Attack Mechanism:

The attacker could be searching for an open proxy server so that he can use that to mask his identity. But because of the all the ports that the attacker is probing it is very likely that he id searching for the Ring Zero Trojan. This works by an attacker installing a trojan on a system that reports back open proxy ports and IP addresses to a centralized server. There is sign that there are crafted packets because at certain times the sequence numbers do not increment.

6. Correlations:

This was a wide spread attack. Information regarding this attack can be found at: www.sans.org/newlook/resources/ringzero.htm

7. Evidence of targeting:

The attacker is targeting one specific IP address looking for the trojan.

8. Severity:

(Criticality + Lethality) – (System + Net Countermeasures) = Severity

Criticality – Windows NT desktop system = 2

Lethality - Attack searches for open proxies or a Trojan = 3

Systems Countermeasures – System may be missing some patches = 3

Network Countermeasures – System is behind a firewall but I don't know if attack was allowed through. = 3

(2 + 3) – (3 + 3) = -1

9. Defensive recommendations:

Make sure that that IP was not running any proxy services and investigate all outbound traffic from ports 8080 or 3128 to see if all systems have the Ring Zero Trojan.

10. What type of trojan are ports 8080, 3128, and 80 associated with?

a. Back Orifice
b. PC Anywhere
c. Ring Zero
d. Netbus

Answer: C

Detect #3

```
Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
  130.111.50.93:685 -> z.y.w.66:111
Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
  130.111.50.93:687 -> z.y.w.66:111
Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
  130.111.50.93:686 -> z.y.w.66:111
Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
  130.111.50.93:690 -> z.y.w.66:111
```

Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
130.111.50.93:689 -> z.y.w.66:111
Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
130.111.50.93:688 -> z.y.w.66:111
Sep 24 10:23:10 hostj snort[341]: RPC Info Query:
130.111.50.93:692 -> z.y.w.66:111
Sep 24 10:23:11 hostmi snort[15718]: RPC Info Query:
130.111.50.93:697 -> z.y.w.98:111
Sep 24 10:23:11 hostmi snort[15718]: RPC Info Query:
130.111.50.93:698 -> z.y.w.98:111
Sep 24 10:23:11 hostmi snort[15718]: RPC Info Query:
130.111.50.93:702 -> z.y.w.98:111
Sep 24 10:23:11 hostmi snort[15718]: RPC Info Query:
130.111.50.93:701 -> z.y.w.98:111
Sep 24 10:23:11 hostmi snort[15718]: RPC Info Query:
130.111.50.93:704 -> z.y.w.98:111

1. Source of Detect:

http://www.sans.org/y2k/092800.htm

2. Detect was generated by:

UNIX syslog

Fields:

Sep 24 10:23:11 **[Date and Time]** hostmi **[Hostname]** snort[15718]**[Process name and ID]**: RPC Info Query**[Error]**:130.111.50.93:704 **[Source IP:Port]**-> z.y.w.98:111**[Destination IP:Port]**

3. Probability the source was spoofed:

Low- this is an information gathering attack that requires a response sent back to the source.

4. Description of the attack:

This attack is an attempt to get information on what types of rpc services this system is running. Associated CVEs:

CVE-1999-0003
CVE-1999-0008
CVE-1999-0212
CVE-1999-0228
CVE-1999-0320
CVE-1999-0353
CVE-1999-0687
CVE-1999-0696
CVE-1999-0900
CVE-1999-0969
CVE-1999-0974

5. Attack Mechanism:

This is just a simple query to see what RPC services are running. This query could then lead to an attack on a specific RPC service.

6. Correlations

None

7. Evidence of targeting:

No

8. Severity:

(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Criticality – UNIX system = 3
Lethality - Information on RPC services = 3
Systems Countermeasures – System logged probe = 3
Network Countermeasures – The informational probe reached the source = 2
(3 + 3) – (3 + 2) = 1

9. Defensive recommendations:

Make sure the system is not running any unnecessary RPC services. If the system has to run any RPC services make sure that they all have the latest versions and patches.

10. RPC services can run on which operating system?

A. Windows
B. UNIX
C. None of the Above
D. All of the above

Answer: D

Detect #4

Feb 22 11:38:56.877 firewall kernel: 226 IP packet dropped
(www.flushing.org[194.217.220.117]->firewall.mydomain.edu[10.0.0.1]:
Protocol=TCP[SYN PUSH] Port 3338->6565): Restricted Port:
Protocol=TCP[SYN PUSH] Port 3338->6565 (received on interface 10.0.0.1)

Feb 22 11:39:02.542 firewall kernel: 226 IP packet dropped
(www.flushing.org[194.217.220.117]->firewall.mydomain.edu[10.0.0.1]:
Protocol=TCP[SYN] Port 30722->49456): Restricted Port:
Protocol=TCP[SYN] Port 30722->49456 (received on interface 10.0.0.1)

Feb 22 11:39:06.774 firewall kernel: 226 IP packet dropped
(www.flushing.org[194.217.220.117]->firewall.mydomain.edu[10.0.0.1]:
Protocol=TCP[] Port 30720->32988): Restricted Port:
Protocol=TCP[] Port 30720->32988 (received on interface 10.0.0.1)

Feb 22 11:40:13.175 firewall kernel: 347 Possible Port Scan
detected on Interface 10.0.0.1 (www.flushing.org[194.217.220.117]->
firewall.mydomain.edu[10.0.0.1]: Protocol=TCP[RST] Port 30724->64)

Feb 22 11:40:25.381 firewall kernel: 347 Possible Port Scan
detected on Interface 10.0.0.1 (www.flushing.org[194.217.220.117]->
firewall.mydomain.edu[10.0.0.1]: Protocol=TCP[SYN PUSH URG FIN RST ACK]
Port 30975->24)

Feb 22 11:40:48.659 firewall kernel: 347 Possible Port Scan
detected on Interface 10.0.0.1 (www.flushing.org[194.217.220.117]->
firewall.mydomain.edu[10.0.0.1]: Protocol=TCP[SYN PUSH URG RST ACK]
Port 30974->208)

Feb 22 11:41:22.917 firewall kernel: 347 Possible Port Scan
detected on Interface 10.0.0.1 (www.flushing.org[194.217.220.117]->
firewall.mydomain.edu[10.0.0.1]: Protocol=TCP[SYN URG FIN RST ACK]
Port 30967->28)

Feb 22 11:41:23.833 firewall kernel: 226 IP packet dropped
(www.flushing.org[194.217.220.117]->firewall.mydomain.edu[10.0.0.1]:
Protocol=TCP[SYN] Port 30722->104): Restricted Port: Protocol=TCP[SYN]
Port 30722->104 (received on interface 10.0.0.1)

1. Source of trace:
http://www.sans.org/y2k/022700.htm

2. Detect was generated by:
IPChains
Fields:
Feb 22 11:41:22.917 [Date and time] **firewall** [Hostname] kernel: 347 Possible Port Scan detected on Interface 10.0.0.1 (**www.flushing.org[194.217.220.117]**[Source IP]-> **firewall.mydomain.edu[10.0.0.1]**[Destination IP]: **Protocol=TCP[SYN URG FIN RST ACK]** [Protocal and Flags] **Port 30967->28**[Source port -> Destination port])

3. Probability the source was spoofed:
Medium to High – The packets are defiantly crafted because of the flags that are set in some of the packets for example:
Feb 22 11:40:48.659 firewall kernel: 347 Possible Port Scan
detected on Interface 10.0.0.1 (www.flushing.org[194.217.220.117]->
firewall.mydomain.edu[10.0.0.1]: **Protocol=TCP[SYN PUSH URG RST ACK]**
Port 30974->208)

In this packet we can see that the TCP flags SYN, PUSH, URG, RST, ACK. This is appropriately called a Christmas Tree packet because all of the bits are set. This cannot occur naturally we know that the packets are crafted.

4. Description of the attack:

This attack is a probe that is designed either penetrate the firewall or to see if the system has any open ports. This is a reconnaissance gathering technique.

5. Attack Mechanism:

The attacker crafts a packet using some type of packet generator. The purpose of this packet is to either get it past the firewall, because most firewalls will allow through packets with SYN ACK set but I am not sure if it will allow through packets with all of the flags set. The

6. Correlations:

None

7. Evidence of targeting:

The attacker is doing a port scan on the firewall.

8. Severity:

(Criticality + Lethality) – (System + Net Countermeasures) = Severity
Criticality – Firewall = 5
Lethality - Attack is searching for a hole in the firewall = 3
Systems Countermeasures – System appears to be current with all patches = 4
Network Countermeasures – None to my knowledge = 2
(5 + 3) – (4 + 2) = 2

9. Defensive Recommendations:

None, it appears as if the firewall did a good job of blocking and logging the packets.

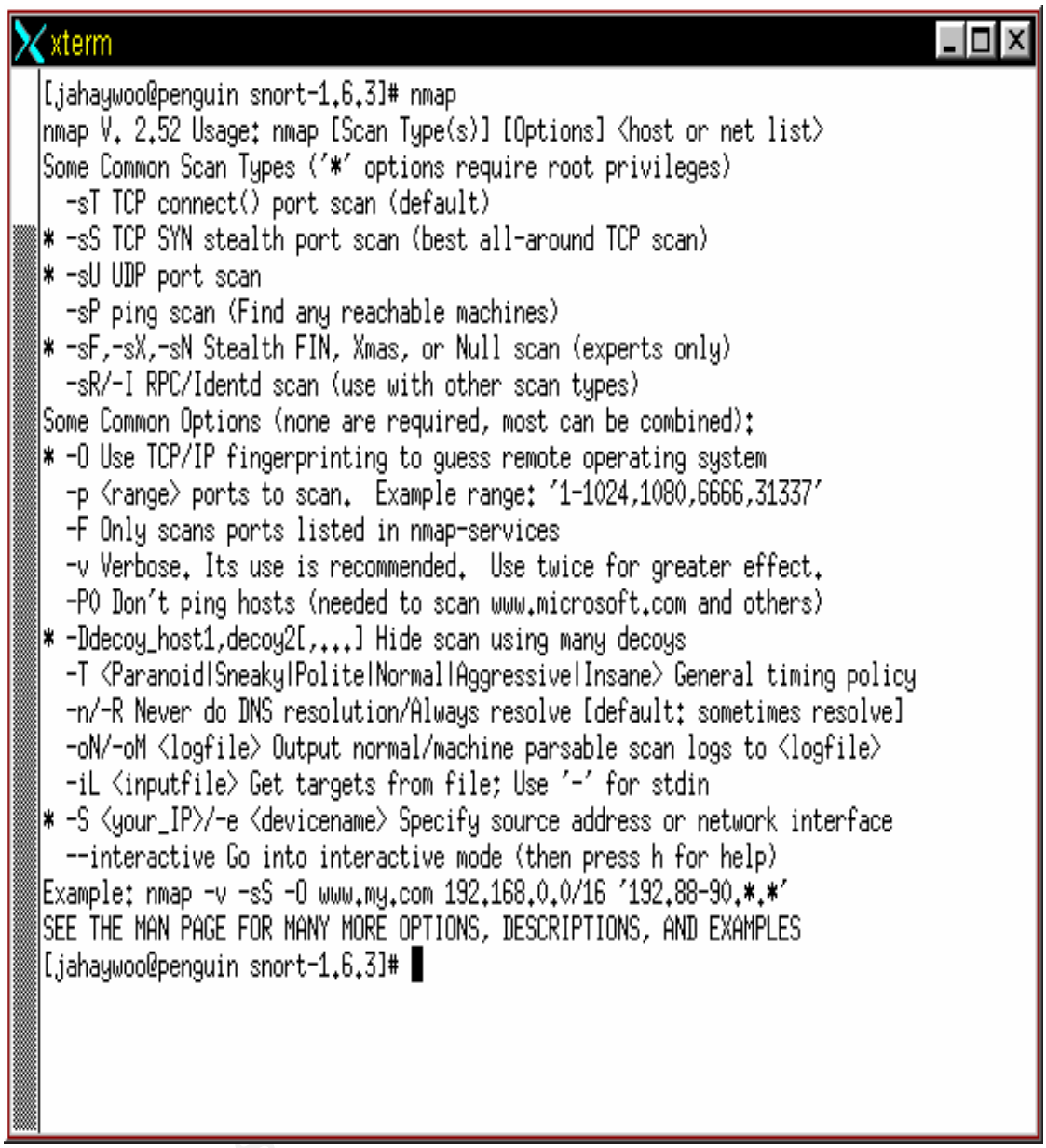
Assignment 2 Evaluate an Attack

1. Give the URL, location or command that you acquired the attack from.

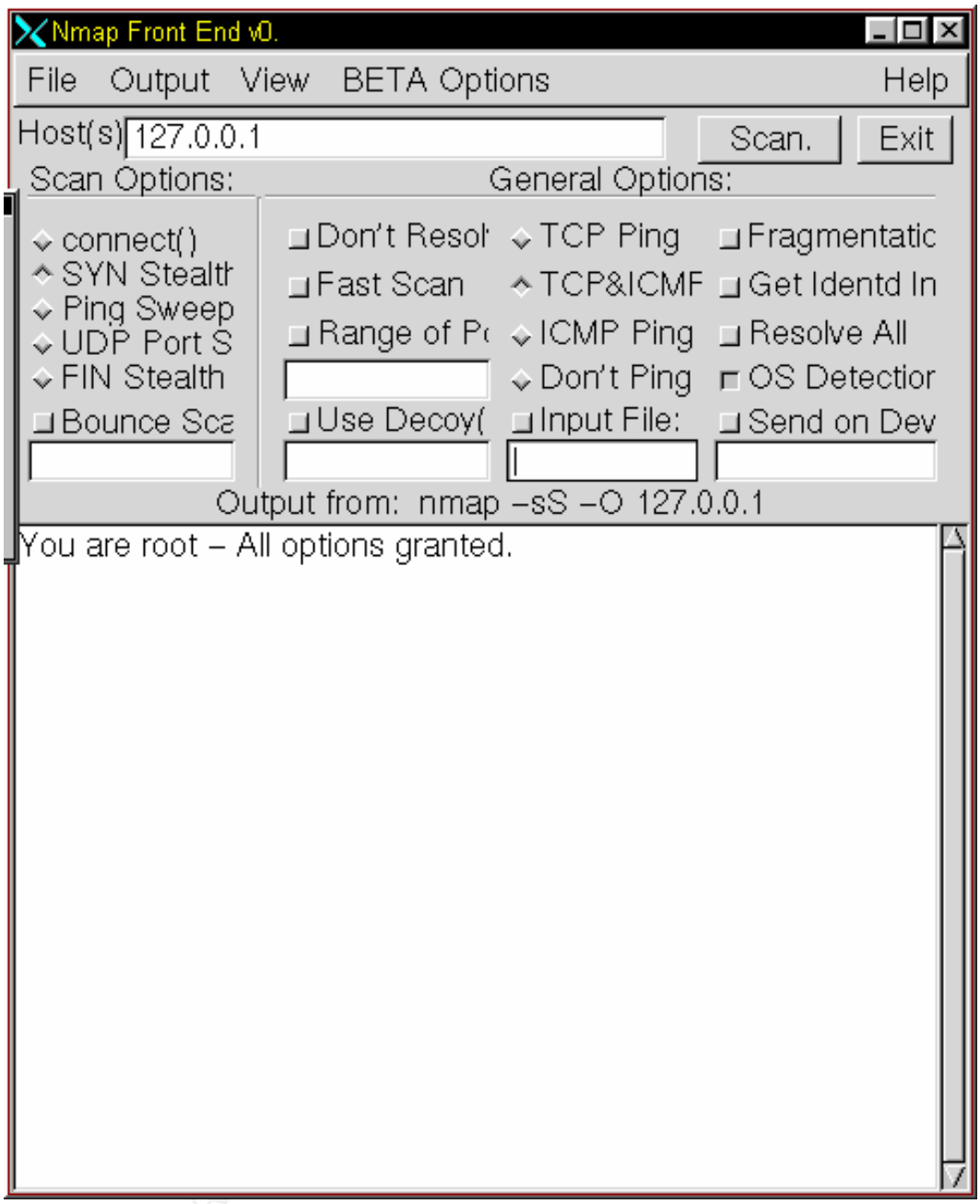
Nmap is available at http://www.insecure.org/nmap/dist/nmap-2.54BETA7.tgz

2. Describe the attack including how it works.

Nmap is a utility for network exploration or security auditing. It supports ping scanning (determine which hosts are up), many port scanning techniques (determine what services the hosts are offering), and TCP/IP fingerprinting (remote host operating system identification). Nmap also gives you flexible target and port specification, decoy scanning, determination of TCP sequence predictability characteristics, sunRPC scanning, reverse-identd scanning, and much more.

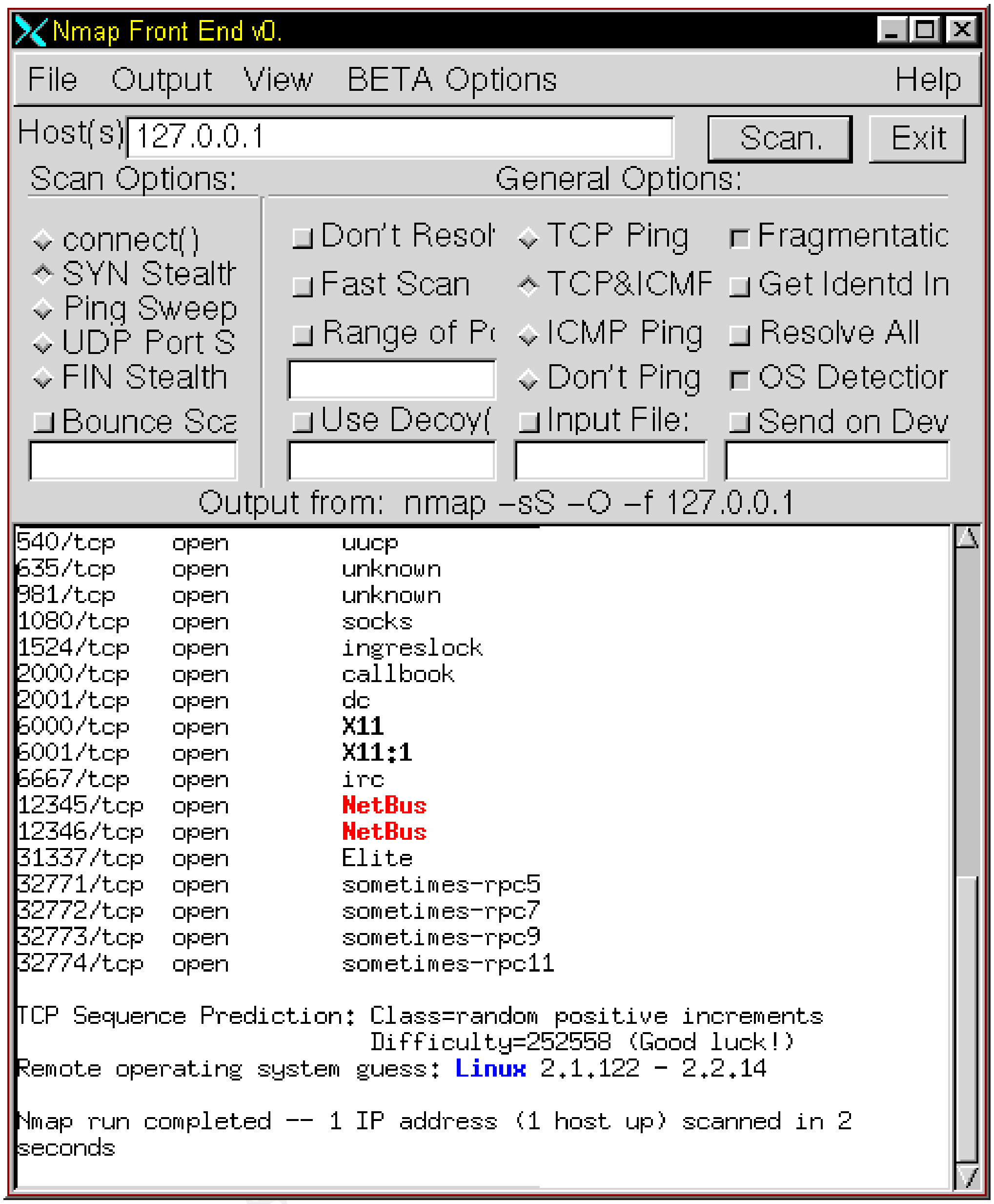


There is also a front-end GUI interface to nmap that you can use if you are unfamiliar with the command line interface.



In this analysis I will use the GUI because it provides better feedback to the user.

I ran a nmap scan using SYN and IP fragments and on the GUI it looked like:



The output from the scan are open services, OS Type and TCP sequence number prediction

3. Provide an annotated trace of the attack in action.

When the tcp fragment nmap scan was run Snort picked up the following:

=+
+=+

[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.197932 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:13599 MF
Frag Offset: 0x0 Frag Size: 0x10

=+=
+=+
[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.198061 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:61895 MF
Frag Offset: 0x0 Frag Size: 0x10
=+=
+=+
[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.198238 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:60238 MF
Frag Offset: 0x0 Frag Size: 0x10
=+=
+=+
[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.198284 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:42032 MF
Frag Offset: 0x0 Frag Size: 0x10
=+=
+=+
[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.198462 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:8684 MF
Frag Offset: 0x0 Frag Size: 0x10
=+=
+=+
[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.198507 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:21718 MF
Frag Offset: 0x0 Frag Size: 0x10
=+=
+=+
[**] Tiny Fragments - Possible Hostile Activity [**]
10/10-12:16:11.203426 10.10.14.77 -> 10.10.14.77
TCP TTL:60 TOS:0x0 ID:53938 MF
Frag Offset: 0x0 Frag Size: 0x10
=+=
+=+

Assignment 3 Analyze This Scenario

Report

ANALYSIS #1

```
Jun 27 01:55:37 193.251.35.190:1785 -> MY.NET.181.88:3152 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1787 -> MY.NET.181.88:3154 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1790 -> MY.NET.181.88:3157 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1791 -> MY.NET.181.88:3158 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1792 -> MY.NET.181.88:3159 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1793 -> MY.NET.181.88:3160 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1794 -> MY.NET.181.88:3161 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1795 -> MY.NET.181.88:3162 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1796 -> MY.NET.181.88:3163 SYN **S*****
Jun 27 01:55:37 193.251.35.190:1798 -> MY.NET.181.88:3165 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1803 -> MY.NET.181.88:3169 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1806 -> MY.NET.181.88:3172 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1811 -> MY.NET.181.88:3177 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1813 -> MY.NET.181.88:3179 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1814 -> MY.NET.181.88:3180 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1815 -> MY.NET.181.88:3181 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1816 -> MY.NET.181.88:3182 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1817 -> MY.NET.181.88:3183 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1818 -> MY.NET.181.88:3184 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1819 -> MY.NET.181.88:3185 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1820 -> MY.NET.181.88:3186 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1823 -> MY.NET.181.88:3189 SYN **S*****
Jun 27 01:55:38 193.251.35.190:1824 -> MY.NET.181.88:3190 SYN **S*****
Jun 27 01:55:42 193.251.35.190:1859 -> MY.NET.181.88:3191 SYN **S*****
Jun 27 01:55:42 193.251.35.190:1860 -> MY.NET.181.88:21 SYN **S*****
Jun 27 01:55:43 193.251.35.190:1868 -> MY.NET.181.88:3192 SYN **S*****
Jun 27 03:19:56 193.251.35.190:3664 -> MY.NET.181.88:2519 SYN **S*****
Jun 27 03:19:56 193.251.35.190:3670 -> MY.NET.181.88:2522 SYN **S*****
Jun 27 03:19:57 193.251.35.190:3671 -> MY.NET.181.88:2523 SYN **S*****
Jun 27 03:19:57 193.251.35.190:3673 -> MY.NET.181.88:2524 SYN **S*****
Jun 27 03:20:00 193.251.35.190:3691 -> MY.NET.181.88:21 SYN **S*****
Jun 27 03:19:58 193.251.35.190:3678 -> MY.NET.181.88:2526 SYN **S*****
Jun 27 03:19:58 193.251.35.190:3682 -> MY.NET.181.88:2528 SYN **S*****
Jun 27 03:19:58 193.251.35.190:3685 -> MY.NET.181.88:2529 SYN **S*****
```

Looking at this trace from June 27 I see that source IP 193.251.35.190 is sending a lot of SYN's to MY.NET.181.88 to several ports. This activity is not that alarming except within the data you can see that the attacker is sporadically trying to establish a telnet connection. It is my opinion that the attacker is trying to hide the attempted telnet connections by sending SYN's to various other ports.

ANALYSIS #2

06/27-07:39:28.385752 [**] NMAP TCP ping! [**] 209.218.228.46:80 -> MY.NET.1.8:53
06/27-07:39:28.388448 [**] NMAP TCP ping! [**] 209.218.228.46:53 -> MY.NET.1.8:53
06/27-07:39:33.390475 [**] NMAP TCP ping! [**] 209.218.228.46:80 -> MY.NET.1.8:53
06/27-07:39:33.390629 [**] NMAP TCP ping! [**] 209.218.228.46:53 -> MY.NET.1.8:53
06/27-07:51:18.494768 [**] NMAP TCP ping! [**] 195.54.105.6:80 -> MY.NET.1.9:53
06/27-07:51:18.494815 [**] NMAP TCP ping! [**] 195.54.105.6:53 -> MY.NET.1.9:53
06/27-07:51:23.472464 [**] NMAP TCP ping! [**] 195.54.105.6:80 -> MY.NET.1.9:53
06/27-07:51:23.472859 [**] NMAP TCP ping! [**] 195.54.105.6:53 -> MY.NET.1.9:53

Here Snort shows us a NMAP TCP ping alarm. NMAP is a very popular hacker tool that is used to map networks. In this track we have two different source addresses doing the same type of scan to MY.NET.1.9 from ports 80 and 53 to port 53. These ports were probably used because most firewalls allow access from them. The attacker could be trying to determine the operating system type of this machine.

ANALYSIS #3

06/28-06:53:09.088168 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.18:53
06/28-06:53:09.146838 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.23:53
06/28-06:53:09.186880 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.25:53
06/28-06:53:09.187045 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.24:53
06/28-06:53:09.248085 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.27:53
06/28-06:53:09.248382 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.28:53
06/28-06:53:09.348108 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.32:53
06/28-06:53:09.386343 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.33:53
06/28-06:53:09.404516 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.34:53
06/28-06:53:09.416009 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.37:53
06/28-06:53:09.421588 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.38:53
06/28-06:53:09.442192 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.39:53
06/28-06:53:09.461970 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.40:53
06/28-06:53:09.700533 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.50:53
06/28-06:53:09.740478 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.5.52:53
.
.
.
.
.
06/28-07:14:22.637857 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.254.199:53
06/28-07:14:22.660875 [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.254.200:53

06/28-07:14:22.705313 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.202:53
06/28-07:14:22.802030 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.206:53
06/28-07:14:22.836063 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.209:53
06/28-07:14:22.843120 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.210:53
06/28-07:14:22.944211 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.213:53
06/28-07:14:22.988584 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.216:53
06/28-07:14:23.069607 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.220:53
06/28-07:14:23.152080 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.225:53
06/28-07:14:23.260603 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.231:53
06/28-07:14:23.268197 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.230:53
06/28-07:14:23.270818 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.232:53
06/28-07:14:23.320774 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.233:53
06/28-07:14:23.344401 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.234:53
06/28-07:14:23.394192 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.236:53
06/28-07:14:23.482999 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.240:53
06/28-07:14:23.646996 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.249:53
06/28-07:14:23.649590 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.248:53
06/28-07:14:23.702364 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.251:53
06/28-07:14:23.732747 [**] SYN-FIN scan! [**] 202.0.178.98:53 ->
MY.NET.254.255:53

This trace concerns me because the source IP 202.0.178.98 did a SYN-FIN scan across the entire class B. This guy was obviously trying to map the network. He even used source and destination ports 53 so that it could go through most firewalls. You should defiantly contact the source IP and see why is scanning all of these addresses

ANALYSIS #4

08/05-18:30:49.580603 [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.051209 [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.053254 [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.056476 [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.059569 [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463

A host on your network appears to be running Napster as shown in the trace above. Napster is a peer-to-peer client that that is used for music and file sharing. The Napster client itself is not a security risk but in my opinion there are two reasons that you should not allow Napster on your network. Uploading and downloading of files uses a significant amount valuable network resources and the files that are downloaded could contain Trojans, viruses and other types of malicious code hidden within them. There are also two candidates for CVE's that address Napster issues, CAN-2000-0281 and CAN-2000-0412.

ANALYSIS #5

08/01-01:12:45.392236 [**] WinGate 1080 Attempt [**] 202.212.5.30:46984 -> MY.NET.100.203:1080
08/01-01:12:48.309347 [**] WinGate 1080 Attempt [**] 202.212.5.30:47131 -> MY.NET.100.203:1080
08/01-01:12:57.981789 [**] WinGate 1080 Attempt [**] 216.67.50.180:1476 -> MY.NET.60.11:1080
08/01-01:12:58.747444 [**] WinGate 1080 Attempt [**] 216.67.50.180:1476 -> MY.NET.60.11:1080
08/01-01:12:59.440469 [**] WinGate 1080 Attempt [**] 216.67.50.180:1476 -> MY.NET.60.11:1080
08/01-01:13:00.154420 [**] WinGate 1080 Attempt [**] 216.67.50.180:1476 -> MY.NET.60.11:1080
08/01-04:39:53.686247 [**] WinGate 1080 Attempt [**] 63.168.242.5:19564 -> MY.NET.98.147:1080
08/01-04:39:53.760486 [**] WinGate 1080 Attempt [**] 63.168.242.5:19617 -> MY.NET.98.147:1080
08/01-04:39:53.875562 [**] WinGate 1080 Attempt [**] 216.198.1.4:3898 -> MY.NET.98.147:1080
08/01-04:39:53.983566 [**] WinGate 1080 Attempt [**] 216.217.216.5:4771 -> MY.NET.98.147:1080

08/01-04:39:54.030175 [**] WinGate 1080 Attempt [**] 216.217.216.5:4772 -> MY.NET.98.147:1080
08/01-04:39:54.701659 [**] WinGate 1080 Attempt [**] 24.165.34.160:3531 -> MY.NET.98.147:1080
08/05-01:42:10.350076 [**] WinGate 1080 Attempt [**] 207.126.106.118:1391 -> MY.NET.98.111:1080
08/05-01:42:11.995196 [**] WinGate 1080 Attempt [**] 216.67.50.140:2166 -> MY.NET.98.111:1080
08/05-01:42:12.420590 [**] WinGate 1080 Attempt [**] 166.62.205.88:3283 -> MY.NET.98.111:1080
08/05-01:42:12.944614 [**] WinGate 1080 Attempt [**] 216.67.50.140:2166 -> MY.NET.98.111:1080
08/05-01:42:13.294578 [**] WinGate 1080 Attempt [**] 166.62.205.88:3283 -> MY.NET.98.111:1080
08/05-01:42:14.189420 [**] WinGate 1080 Attempt [**] 166.62.205.88:3283 -> MY.NET.98.111:1080
08/05-01:42:15.052669 [**] WinGate 1080 Attempt [**] 166.62.205.88:3283 -> MY.NET.98.111:1080
08/05-01:42:15.059522 [**] WinGate 1080 Attempt [**] 216.67.50.140:2166 -> MY.NET.98.111:1080

Across several days there are these WinGate alarms. WinGate is a windows proxy server. There are thousands of these events in the log files coming from various sources going to the same few IP addresses. You should check to see if these source IP's are running any proxy services and if so make sure that are all authorized. Some CVE references are:

CVE-1999-0290
The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.

CVE-1999-0291
The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.

CVE-1999-0441
Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.

CVE-1999-0494
Denial of service in WinGate proxy through a buffer overflow in POP3.

ANALYSIS #6

08/05-02:14:30.907158 [**] Attempted Sun RPC high port access [**] 205.188.153.111:4000 -> MY.NET.217.126:32771

08/05-02:15:30.875817 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:20:30.555538 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:21:30.511518 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:22:30.450138 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:22:38.632914 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:23:30.397350 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:24:30.368618 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:25:30.282563 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:26:30.223594 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:28:30.119074 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:29:30.005417 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:30:30.005269 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771
08/05-02:31:29.921235 [**] Attempted Sun RPC high port access [**]
205.188.153.111:4000 -> MY.NET.217.126:32771

07/29-16:24:35.903923 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:36.856924 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:37.660049 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:37.681154 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:37.886257 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:42.003947 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:43.950767 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:48.732229 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771
07/29-16:24:50.394313 [**] SUNRPC highport access! [**] 205.188.3.205:5190->
MY.NET.98.145:32771

07/29-16:24:50.395577 [**] SUNRPC highport access! [**] 205.188.3.205:5190-> MY.NET.98.145:32771
07/29-16:24:51.127465 [**] SUNRPC highport access! [**] 205.188.3.205:5190-> MY.NET.98.145:32771
07/29-16:24:51.872439 [**] SUNRPC highport access! [**] 205.188.3.205:5190-> MY.NET.98.145:32771
07/29-16:24:51.879197 [**] SUNRPC highport access! [**] 205.188.3.205:5190-> MY.NET.98.145:32771
07/29-16:24:53.247324 [**] SUNRPC highport access! [**] 205.188.3.205:5190-> MY.NET.98.145:32771

There are various vulnerabilities in the RPC services that allow an attacker to gain root level access. In these two traces we can see that two of your computers allowed sunrpc highport access. I would suggest that these computers should be taken off line until you can determine if there was a compromise. There are several advisories and CVE's to reference for this type of attack.

CVE-1999-0003
Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd)
CVE-1999-0008
Buffer overflow in NIS+, in Sun's rpc.nisd program
CVE-1999-0212
Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.
CVE-1999-0228
Denial of service in RPCSS.EXE program (RPC Locator) in Windows NT.
CVE-1999-0320
SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.
CVE-1999-0353
rpc.pcnfsd in HP gives remote root access by changing the permissions on the main printer spool directory.
CVE-1999-0687
The ToolTalk ttsession daemon uses weak RPC authentication, which allows a remote attacker to execute commands.
CVE-1999-0696
Buffer overflow in CDE Calendar Manager Service Daemon (rpc.cmsd)
CVE-1999-0900
Buffer overflow in rpc.yppasswdd allows a local user to gain privileges via MD5 hash generation.
CVE-1999-0969
The Windows NT RPC service allows remote attackers to conduct a denial of service using spoofed malformed RPC packets which generate an error message that is sent to the spoofed host, potentially setting up a loop, aka Snork.
CVE-1999-0974

Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad service.

Also CERT/CC's incident note IN-2000-10 describes a increase in activity in rpc probes.

ANALYSIS #7

07/11-19:28:57.652242 [**] Happy 99 Virus [**] 200.223.11.7:4836-> MY.NET.110.150:25
08/05-11:22:48.017066 [**] Happy 99 Virus [**] 206.67.51.242:4889 -> MY.NET.6.47:25
07/19-04:28:40.867369 [**] Happy 99 Virus [**] 203.251.136.2:4985-> MY.NET.253.42:25
07/26-07:50:56.700210 [**] Happy 99 Virus [**] 208.130.42.17:40221-> MY.NET.6.34:25

It appears as if some of your computers are infected with the Happy 99 computer virus of one of its variants. Isolate these boxes from the network and run an antivirus tool with the latest signature on these computers to get rid of the viruses. My suggestion is that you should make sure that all of the computers on your network have up to date virus signature files and that they run a weekly check for viruses.

ANALYSIS #8

06/30-16:33:57.773279 [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 151.164.223.206:4499 -> MY.NET.99.16:21
06/30-16:34:00.037398 [**] Possible wu-ftpd exploit - GIAC000623 [**] 151.164.223.206:4499 -> MY.NET.99.16:21
06/30-16:35:11.406398 [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 151.164.223.206:4500 -> MY.NET.144.59:21
06/30-16:35:13.560305 [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 151.164.223.206:4500 -> MY.NET.144.59:21
06/30-16:35:13.626498 [**] Possible wu-ftpd exploit - GIAC000623 [**] 151.164.223.206:4500 -> MY.NET.144.59:21

The wu-ftp daemon comes standard with several flavors of UNIX and there are several vulnerabilities to this program that allow unauthorized access. To fix this you should apply the latest patch from the vendor. Reference these CVE's:

CVE-1999-0075
PASV core dump in wu-ftpd daemon when attacker uses a QUOTE PASV command after specifying a username and password.
CVE-1999-0080

wu-ftp FTP server allows root access via "site exec" command.

CVE-1999-0081

wu-ftp allows files to be overwritten via the rnfr command.

CVE-1999-0368

Buffer overflows in wuarchive ftpd (wu-ftpd) and ProFTPD lead to remote root access, a.k.a. palmetto.

CVE-1999-0720

The pt_chown command in Linux allows local users to modify TTY terminal de vices that belong to other users.

CVE-1999-0878

Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via MAPPING_CHDIR.

CVE-1999-0879

Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via macro variables in a message file.

CVE-1999-0880

Denial of service in WU-FTPD via the SITE NEWER command, which does not free memory properly.

CVE-1999-0955

Race condition in wu-ftpd and BSDI ftpd allows remote attackers gain root access via the SITE EXEC command.

CVE-1999-0997

wu-ftp with FTP conversion enabled allows an attacker to execute commands via a malformed file name that is interpreted as an argument to the program that does the conversion, e.g. tar or uncompress.

Also you can look at CERT's advisory CA-1999-13.

ANALYSIS #9

```
6/28-06:35:13.540772  [**] Tiny Fragments - Possible Hostile Activity [**]
63.236.34.174 -> MY.NET.1.8
06/28-06:35:13.540827  [**] Tiny Fragments - Possible Hostile Activity [**]
63.236.34.174 -> MY.NET.1.8
06/28-06:35:13.540878  [**] Tiny Fragments - Possible Hostile Activity [**]
63.236.34.174 -> MY.NET.1.8
06/28-06:37:13.538078  [**] Tiny Fragments - Possible Hostile Activity [**]
63.236.34.174 -> MY.NET.1.8
06/28-06:37:13.538175  [**] Tiny Fragments - Possible Hostile Activity [**]
63.236.34.174 -> MY.NET.1.8
06/28-06:37:13.538272  [**] Tiny Fragments - Possible Hostile Activity [**]
63.236.34.174 -> MY.NET.1.8
```

A lot of modern network scanners use small fragments to penetrate firewalls and scan hosts.

ANALYSIS #10

07/17-11:56:33.121716 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.4.238:1072 -> MY.NET.53.28:4110
07/19-09:57:05.082211 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.4.238:1182 -> MY.NET.53.28:4110
07/26-00:21:42.187486 [**] Watchlist 000222 NET-NCFC [**] 159.226.5.77:1114 -> MY.NET.100.230:25

07/26-00:39:38.061305 [**] Watchlist 000222 NET-NCFC [**] 159.226.42.3:1182 -> MY.NET.145.18:21

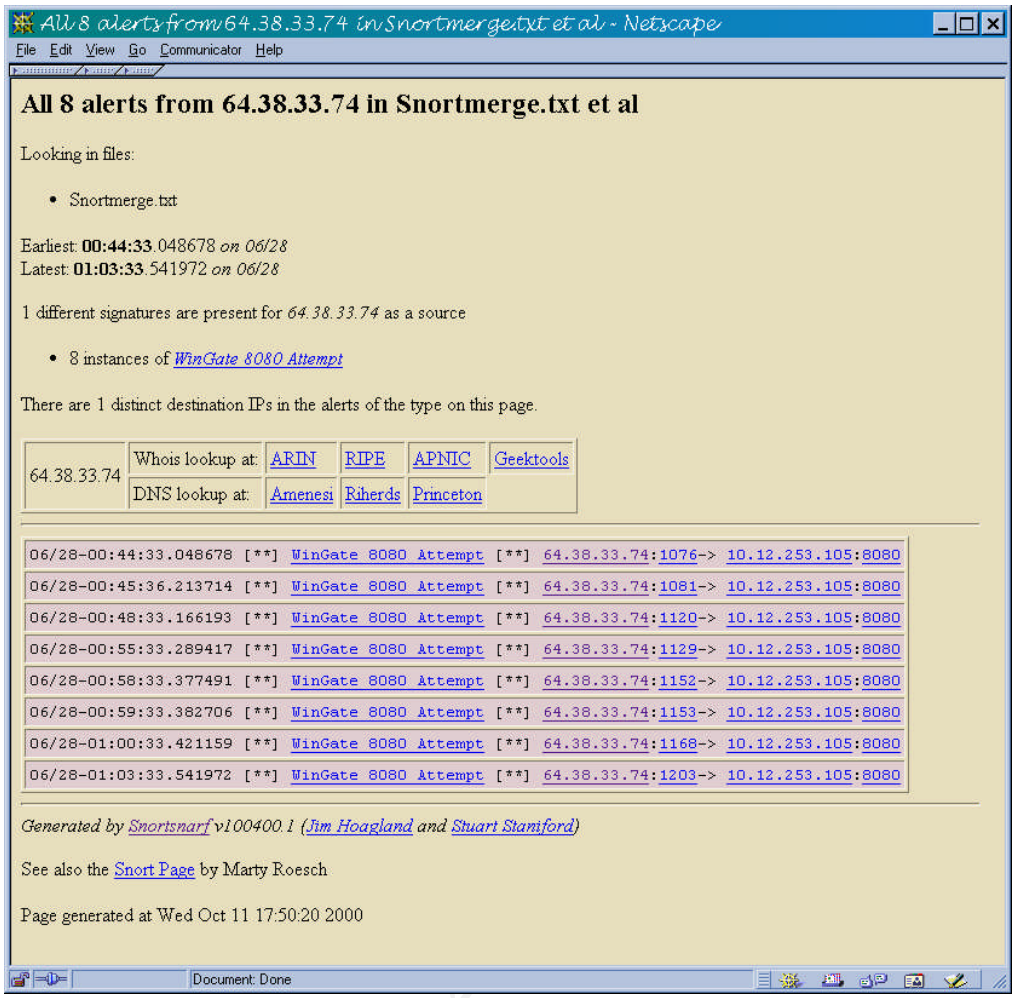
There were some snort rules that were set up to watch certain IP's activities. These alarms are a result of that rule set. IP 159.226.42.3 is from China and IP 212.179.4.238 is from Israel. The Chinese IP was trying do a lot of email port 25 connections and also tried to telnet. The Israeli IP was trying a lot of high port connections. You should defiantly monitor these two IPs closely.

Overview

The traces from your network traffic showed that there is a great deal of potentially hostile activity to your network. I would suggest that some sort of intrusion detection strategy should be put in place. This would require re-evaluating your perimeter defense and possibly installing some sort of intrusion detection sensor on a permanent basis. My company would be more than willing to give you some suggestions and help with implementation.

Assignment 4 Analysis Process

The process that I used to analyze the data was to first merge all of the Snort alert files together and then merge all of the Snort scan files together. Once I concatenated the files I was left with two huge Snort files. In order to parse out all the data so that I would not have to look at every event I took the data and use a perl script called SnortSnarf written by Jim Hoagland and Stuart Staniford to break the data out into different attack signatures and show me the different types of activities fron the logs.



Then using the techniques that I learned it the SANS class I gave what I think is an accurate account of what was going on in the traces.

Last Updated: July 27th, 2017

Upcoming Training

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 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
Baltimore Fall 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
SANS Baltimore Fall 2017	Baltimore, MD	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 Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced