Global Information Assurance Certification Paper

*** Northcutt, research is evident here, mostly has his own detects and from a variety of sources. This submittal illustrates the value of home field advantage, good accuracy, process is a bit free form but he pulls it off, I'd be happy to have this guy watching my network! 95 ***

GIAC LevelTwo Practical

10 Detects with analysis

Jerry Shenk

The primary purpose of these analyses is to fulfill the practical requirements for the SNAP Level 1 certification. The logs have been cleaned except as noted. My reason for cleaning the attacking machine's addresses is that in most cases, the machine is not owned by the actual attacker. Most of the people reading this will be seeing it well past the time it was created so the actual machines have hopefully been fixed by this time.

One thing that all these traces have in common is the need for just a little more data. I don't have a Shadow system running at any of these locations yet but this exercise has proven that it would be nice to be able to look at a more complete historical picture of the data sessions.

Detect 1

In this trace, 100.100.100.1 is a linux box running snort that also does some remote system monitoring. 200.200.200.2 is an NT box on the internet that is owned by a client of ours that we monitor for connectivity and service availability (http, smtp and pop3).

This initial sequence of 6 packets was being recorded by the IDS every 5 minutes. Since it was occurring at a regular basis I thought that it probably was not hostile but wanted to figure out what it was before we programmed the IDS to ignore this traffic. I ran an analysis of all traffic going to and from 200.200.200.2 and have a portion of that listed as the detail packet sequence. This shows that when 100.100.100.1 initiates a connection to 200.200.200.2 to test the SMTP services the SMTP host (NT server) initiates a connection to the client (100.100.100.1) on port 137.

This is determined not to be hostile traffic. It is just the NT server attempting some type of netbios verification which is refused by the linux box because netbios services are not installed. The IDS was adjusted to allow this traffic without alerting.

Initial packet sequence

03/23-21:52:30.806173 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
200.200.200.2:62076 -> 100.100.100.1:137 UDP TTL:117 TOS:0x0 ID:42501 Len: 58

03/23-21:52:30.807099 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
200.200.200.2:137 -> 100.100.100.1:137 UDP TTL:117 TOS:0x0 ID:42757 Len: 58

03/23-21:52:32.318619 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
200.200.200.2:62076 -> 100.100.100.1:137 UDP TTL:117 TOS:0x0 ID:44293 Len: 58

03/23-21:52:32.320260 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
200.200.200.2:137 -> 100.100.100.1:137 UDP TTL:117 TOS:0x0 ID:44037 Len: 58

03/23-21:52:33.889927 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
200.200.200.2:62076 -> 100.100.100.1:137 UDP TTL:117 TOS:0x0 ID:44549 Len: 58

03/23-21:52:33.891690 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
200.200.200.2:137 -> 100.100.100.1:137 UDP TTL:117 TOS:0x0 ID:44805 Len: 58

Detail packet sequence

SMTP service testing

22:47:27.295047 eth0 > 100.100.100.1.4255 > 200.200.200.2.smtp: S 139982045:139982045(0) win 32120 <mss 1460,sackOK,timestamp 12326328 0,nop,wscale 0> (DF)
22:47:27.428076 eth0 < 200.200.200.2.smtp > 100.100.100.1.4255: S 1655581664:1655581664(0) ack 139982046 win 8760 <mss 1460> (DF)
22:47:27.428160 eth0 > 100.100.100.1.4255 > 200.200.200.2.smtp: . 1:1(0) ack 1 win 32120 (DF)
22:47:27.428577 eth0 > 100.100.100.1.4255 > 200.200.200.2.smtp: P 1:7(6) ack 1 win 32120 (DF)
22:47:27.552038 eth0 < 200.200.200.2.smtp > 100.100.100.1.4255: . 1:1(0) ack 7 win 8754 (DF)

Netbios query from 200.200.200.2

22:47:27.826998 eth0 < 200.200.200.2.61911 > 100.100.100.1.netbios-ns:NBT UDP PACKET(137): QUERY; REQUEST; UNICAST

22:47:27.827070 eth0 > 100.100.100.1 > 200.200.200.2: icmp: 100.100.100.1 udp port netbios-ns unreachable [tos 0xc0]

Detect 2

The following sequence was detected on an internal IDS.

This is only a part of the total amount of captured data. There were a lot of snmp requests (port 161) in this same pattern. Both addresses are internal addresses. I was not aware of what either one was. Further inspection of the body using tcpdump to view the raw packet data showed that the traffic looked like it was being requested from a switch our router because of the progression through the interfaces branch of the snmp tree. Reviewing the network documentation showed that there was in fact a switch at 10.1.6.41. There is a monitoring computer on this network at 10.1.1.8 that collects SNMP data from various servers, switches and routers. Further inspection of 10.1.1.8 showed that it also had a secondary address of 10.1.6.1 bound to it's ethernet interface.

This traffic is found to be friendly traffic and the IDS was modified to allow this traffic without logging it. I will pursue the ramifications of removing the 10.1.6.1 address from eth0.

Initial packet sequence

03/11-15:09:02.797862 0:60:8:A3:BF:25 -> 8:0:4E:35:C8:3B type:0x800 len:0x96
10.1.6.1:1595 -> 10.1.6.41:161 UDP TTL:64 TOS:0x0 ID:9907
Len: 116

03/11-15:09:02.866064 0:60:8:A3:BF:25 -> 8:0:4E:35:C8:3B type:0x800 len:0x93
10.1.6.1:1595 -> 10.1.6.41:161 UDP TTL:64 TOS:0x0 ID:9908
Len: 113

03/11-15:09:02.932064 0:60:8:A3:BF:25 -> 8:0:4E:35:C8:3B type:0x800 len:0x93
10.1.6.1:1595 -> 10.1.6.41:161 UDP TTL:64 TOS:0x0 ID:9909
Len: 113

03/11-15:09:02.998981 0:60:8:A3:BF:25 -> 8:0:4E:35:C8:3B type:0x800 len:0x93
10.1.6.1:1595 -> 10.1.6.41:161 UDP TTL:64 TOS:0x0 ID:9910
Len: 113

03/11-15:09:03.067278 0:60:8:A3:BF:25 -> 8:0:4E:35:C8:3B type:0x800 len:0x93
10.1.6.1:1595 -> 10.1.6.41:161 UDP TTL:64 TOS:0x0 ID:9911
Len: 113

Tcpdump output of log data

15:09:02.797862 10.1.6.1.1595 > 10.1.6.41.snmp: GetRequest(93)
interfaces.ifTable.ifEntry.ifInOctets.1001 interfaces.ifTable.ifEntry
.ifOutOctets.1001 system.sysUpTime.0 system.sysName.0 interfaces.ifTable.ifEntry.ifDescr.1001
15:09:02.866064 10.1.6.1.1595 > 10.1.6.41.snmp: GetRequest(90) interfaces.ifTable.ifEntry.ifInOctets.101
interfaces.ifTable.ifEntry.
ifOutOctets.101 system.sysUpTime.0 system.sysName.0 interfaces.ifTable.ifEntry.ifDescr.101
15:09:02.932064 10.1.6.1.1595 > 10.1.6.41.snmp: GetRequest(90) interfaces.ifTable.ifEntry.ifInOctets.102
interfaces.ifTable.ifEntry.
ifOutOctets.102 system.sysUpTime.0 system.sysName.0 interfaces.ifTable.ifEntry.ifDescr.102
15:09:02.998981 10.1.6.1.1595 > 10.1.6.41.snmp: GetRequest(90) interfaces.ifTable.ifEntry.ifInOctets.103
interfaces.ifTable.ifEntry.
ifOutOctets.103 system.sysUpTime.0 system.sysName.0 interfaces.ifTable.ifEntry.ifDescr.103

Detect 3

In this sequence of packets, both addresses are Netware servers. 10.94.3.1 is running Netware 5.0 and 10.1.1.8 is running Netware 4.

This traffic bears further review because of the destination port 1234 on udp. This looks like it could be an Ultors Trojan back door 10.94.3.1 and 10.1.1.8 are both Netware servers. I then collected a large sequence of all traffic between these two hosts. This sample shows no traffic other than the udp traffic between port 524 on the Netware 5 server and port 1234 on the Netware 4. This revealed that there was also tcp traffic between port 4326 on 10.1.1.8 and port 524 on 10.94.3.1. I am suspecting that this is not in fact a trojan because of the computers involved so I went to the Novell web site and found a document on the IP ports used in Netware 5 (TID#**10013531)**. It turns out that Novell servers do NCP (Netware Core Protocol) communication over tcp 524 and time syncronization over udp 524. The use of port 1234 seems to simply be the ephemeral port that was next in line for use.

This network has 25-30 Netware servers so this will be a difficult traffic to eliminate from the IDS without allowing a hole for the Ultors Trojan. Given the fact that this network has all private addresses on the inside I think it's safe to eliminate the port 1234 alarms for the internal sensors but we can still allow it on the external sensor because this traffic still should not be being sent out to the internet.

Initial Packet Sequence (collected using snort)

```
04/05-21:43:00.471844 0:90:27:54:B5:D -> 0:10:4B:97:F:40 type:0x800 len:0x3C
10.94.3.1:524 -> 10.1.1.8:1234 UDP TTL:128 TOS:0x0 ID:62921
Len: 22
3E 3E 00 62 00 00 00 00 62 3F 00 00 00 00 00 00  >>.b....b?......
51 00                                            Q.

04/05-21:43:00.471844 0:10:4B:97:F:40 -> 0:A0:C9:FB:53:BB type:0x800 len:0x3C
10.94.3.1:524 -> 10.1.1.8:1234 UDP TTL:127 TOS:0x0 ID:62921
Len: 22
3E 3E 00 62 00 00 00 00 62 3F 00 00 00 00 4B 4B  >>.b....b?....KK
4B 4B                                            KK

04/05-21:49:03.811844 0:90:27:54:B5:D -> 0:10:4B:97:F:40 type:0x800 len:0x3C
10.94.3.1:524 -> 10.1.1.8:1234 UDP TTL:128 TOS:0x0 ID:8138
Len: 22
3E 3E 00 62 00 00 00 00 62 3F 00 00 00 00 63 50  >>.b....b?....cP
00 00                                            ..

04/05-21:49:03.811844 0:10:4B:97:F:40 -> 0:A0:C9:FB:53:BB type:0x800 len:0x3C
10.94.3.1:524 -> 10.1.1.8:1234 UDP TTL:127 TOS:0x0 ID:8138
Len: 22
3E 3E 00 62 00 00 00 00 62 3F 00 00 00 00 17 17  >>.b....b?......
17 17                                            ..
```

2nd Packet Sequence (collected using tcpdump)

```
Wed Apr  5 21:54:52 EDT 2000
tcpdump: listening on eth0
21:55:07.141844 10.94.3.1.524 > 10.1.1.8.1234: udp 14
21:55:07.141844 10.94.3.1.524 > 10.1.1.8.1234: udp 14
21:55:07.181844 10.1.1.8.1234 > 10.94.3.1.524: udp 14
21:55:07.181844 10.1.1.8.1234 > 10.94.3.1.524: udp 14
21:55:39.911844 10.1.1.8.4326 > 10.94.3.1.524: P 915600012:915600268(256) ack 1972944771 win
65424 (DF)
```

21:55:39.921844 10.1.1.8.4326 > 10.94.3.1.524: P 0:256(256) ack 1 win 65424 (DF)
21:55:39.921844 10.94.3.1.524 > 10.1.1.8.4326: P 1:207(206) ack 256 win 20496 (DF)
21:55:39.921844 10.94.3.1.524 > 10.1.1.8.4326: P 1:207(206) ack 256 win 20496 (DF)
21:55:40.001844 10.1.1.8.4326 > 10.94.3.1.524: P 256:372(116) ack 207 win 65218 (DF)
21:55:40.001844 10.1.1.8.4326 > 10.94.3.1.524: P 256:372(116) ack 207 win 65218 (DF)
21:55:40.001844 10.94.3.1.524 > 10.1.1.8.4326: P 207:397(190) ack 372 win 20380 (DF)
21:55:40.001844 10.94.3.1.524 > 10.1.1.8.4326: P 207:397(190) ack 372 win 20380 (DF)
21:55:40.091844 10.1.1.8.4326 > 10.94.3.1.524: . ack 397 win 65028 (DF)
21:55:40.091844 10.1.1.8.4326 > 10.94.3.1.524: . ack 397 win 65028 (DF)
22:00:23.041844 10.1.1.8.524 > 10.94.3.1.4105: . 918296195:918296196(1) ack 3820007976 win 22404 (DF)
22:00:23.041844 10.1.1.8.524 > 10.94.3.1.4105: . 0:1(1) ack 1 win 22404 (DF)
22:00:23.041844 10.94.3.1.4105 > 10.1.1.8.524: . ack 1 win 65535 (DF)
22:00:23.041844 10.94.3.1.4105 > 10.1.1.8.524: . ack 1 win 65535 (DF)
22:01:10.481844 10.94.3.1.524 > 10.1.1.8.1234: udp 14

3rd Packet Sequence (collected using tcpdump w/-w option and then using snort to view)
04/05-22:05:06.371844 0:A0:C9:FB:53:BB -> 0:10:4B:97:F:40 type:0x800 len:0x6E
10.1.1.8:427 -> 10.94.3.1:427 UDP TTL:32 TOS:0x0 ID:1914
Len: 75
01 02 00 43 00 00 65 6E 00 6A 00 25 00 00 00 01 ...C..en.j.%....
00 00 00 2F 73 65 72 76 69 63   .../servic

…..much similar traffic deleted to shorten the report

04/05-22:05:58.471844 0:10:4B:97:F:40 -> 0:90:27:54:B5:D type:0x800 len:0x21A
10.1.1.8:427 -> 10.94.3.1:427 UDP TTL:31 TOS:0x0 ID:1922
Len: 504
01 07 01 F0 00 00 65 6E 00 6A 00 8D 00 00 01 E0 ......en.j......
28 73 76 63 6E 61 6D 65 2D 77   (svcname-w

04/05-22:06:07.241844 0:90:27:54:B5:D -> 0:10:4B:97:F:40 type:0x800 len:0x136
10.94.3.1:4105 -> 10.1.1.8:524 TCP TTL:128 TOS:0x0 ID:38275 DF
*****PA* Seq: 0xE3B0B228   Ack: 0x36BC1684   Win: 0xFFFF
44 6D 64 54 00 00 01 00 00 00 00 01 00 00   DmdT..........

04/05-22:06:07.241844 0:10:4B:97:F:40 -> 0:A0:C9:FB:53:BB type:0x800 len:0x136
10.94.3.1:4105 -> 10.1.1.8:524 TCP TTL:127 TOS:0x0 ID:38275 DF
*****PA* Seq: 0xE3B0B228   Ack: 0x36BC1684   Win: 0xFFFF
44 6D 64 54 00 00 01 00 00 00 00 01 00 00   DmdT..........

…..much similar traffic deleted to shorten the report

04/05-22:06:47.931844 0:10:4B:97:F:40 -> 0:90:27:54:B5:D type:0x800 len:0x3C
10.1.1.8:4326 -> 10.94.3.1:524 TCP TTL:127 TOS:0x0 ID:1933 DF
******A* Seq: 0x3692FD04   Ack: 0x7598C60B   Win: 0xFE02
00 00 00 00 00 00   ......

04/05-22:07:13.801844 0:90:27:54:B5:D -> 0:10:4B:97:F:40 type:0x800 len:0x3C
10.94.3.1:524 -> 10.1.1.8:1234 UDP TTL:128 TOS:0x0 ID:40248
Len: 22
3E 3E 00 62 00 00 00 00 62 3F 00 00 00 00 63 50 >>.b....b?....cP
00 00   ..

…..much similar traffic deleted to shorten the report

Detect 4

In this trace, 200.200.200.0 is an internal NATted network. 200.200.200.34 is an IDS on this network and 200.200.200.35 is a mailserver. Both are behind a firewall with a default deny rule.

This is a hostile attempt that I would classify as a reconnaissance because there is no case when a Syn flag and a Fin flag should be set in the same packet. These packets also have the same sequence number which is further evidence of a 'crafted' packet. We can assume that this is some type of an attempt to slip by security systems while looking for active hosts.

This is not a particularly severe detection since both machines are behind a well-configured firewall with currently patched operating systems. The 'victim' machines are both configured with only the needed services running.

The addresses were tracked back to an ISP in Illinois and they were contacted. The ISP was very thankful and glad for the traces with specific information.

Initial Trace

```
04/06-19:48:45.962911 0:D0:BA:22:1D:37 -> 0:10:4B:97:F:44 type:0x800 len:0x3C
100.100.100.11:53 -> 200.200.200.34:53 TCP TTL:26 TOS:0x0 ID:39426
**SF**** Seq: 0x22D4226A   Ack: 0x6F721642   Win: 0x404
30 00 49 54 0D 0A                        0.IT..

04/06-19:48:45.982911 0:D0:BA:22:1D:37 -> 0:10:4B:97:F:44 type:0x800 len:0x3C
100.100.100.11:53 -> 200.200.200.35:53 TCP TTL:26 TOS:0x0 ID:39426
**SF**** Seq: 0x22D4226A   Ack: 0x6F721642   Win: 0x404
30 00 49 54 0D 0A                        0.IT..

19:48:45.962911 100.100.100.11.domain > 200.200.200.34.domain: SF 584327786:584327786(0) win
1028
19:48:45.982911 100.100.100.11.domain > 200.200.200.35.domain: SF 584327786:584327786(0) win
1028
```

Detect 5

These alerts have all come from Linux boxes running portsentry. They are connected to the internet using 2 different ISPs.

Early Friday AM I received alerts from 2 monitored systems. At this point, I looked up the source address and found it to be owned by an overseas university. I e-mailed an administrator on that site. Later in the AM, we picked up a scan on 2 more machines that are on a completely different network and later in the afternoon a 5th hit on the same network as 2 of the initial 3 networks were located. I am not aware of a service or trojan on port 32773 so it's probably a relatively new variant of an exiting trojan.

By looking at the times of the attacks on the various networks, it would seem that the scanner is mixing the addresses up into a number of bands to avoid being quite so obvious but it's still going quite fast over a large area.

From my perspective, this detection would have a low severity because none of the machines are running anything on these ports. I verified this fact by running netstat. From a wider perspective, this could have more far-reaching affects because of what might be done with the machines that will invariably be found with a scan of this magnitude. I'm disappointed but not surprised that I did not hear back from the university. Perhaps the lack of response is a reason to NOT have cleaned the logs of the attacker in this case.

Initial alerts

```
---------------------------------------------------------------------
Date Sent: Friday, April 07, 2000 2:29 PM
Subject: ***_xxxxxxx_Attack_From_193.49.xx.xxx_on_32773
host ip address 208.170.xxx.xxx
---------------------------------------------------------------------
Date Sent: Friday, April 07, 2000 8:25 AM
Subject: ***_Attack_From_193.49.xx.xxx_on_32773_xxxxxxx_xxxxxxxx
host ip address 208.36.xx.xxx
---------------------------------------------------------------------
Date Sent: Friday, April 07, 2000 8:14 AM
Subject: ***_Attack_From_193.49.xx.xxx_on_32773_xxxxxxx_xxxxxx
Host ip address 208.36.xx.xxx
---------------------------------------------------------------------
Date Sent: Friday, April 07, 2000 6:54 AM
Subject: ***_xxxxxx_Attack_From_193.49.xx.xxx_on_32773
Host ip address 208.7.xxx.xxx
---------------------------------------------------------------------
Date Sent: Friday, April 07, 2000 5:41 AM
Subject: ***_Attack_From_193.49.xx.xxx_on_32773_xxxxxxxxxx
Host ip address not known – in the 208.170.*.* range
---------------------------------------------------------------------
```

Log from 2nd site

Apr 7 06:54:52 SentryBox portsentry[2979]: attackalert: SYN/Normal scan from host: xxxxxxxx.xxxx.x-xx.xx.xx
ud.fr/193.49.xx.xxx to TCP port: 32773
Apr 7 06:54:52 SentryBox portsentry[2979]: attackalert: External command run for host: 193.49.xx.xxx using command: "/usr/bin/tail /var/log/messages | /bin/mail xxx@xxxx.xxx,xxxxx@xxxxx.xxx,xxx@xxxx.xxx
-s ***_xxxxxx_Attack_From_193.49.xx.xxx_on_32773"
Apr 7 06:54:52 SentryBox portsentry[2979]: attackalert: Host 193.49.xx.xxx has been blocked via wrapp ers with string: "ALL: 193.49.xx.xxx"
Apr 7 06:54:53 SentryBox portsentry[2979]: attackalert: Host 193.49.xx.xxx has been blocked via dropp ed route using command: "/sbin/ipfwadm -I -i deny -S 193.49.xx.xxx -o"

Apr 7 06:54:53 SentryBox portsentry[2979]: attackalert: SYN/Normal scan from host: xxxxxxxx.xxxx.x-xx.xx.xx
ud.fr/193.49.xx.xxx to TCP port: 32773
Apr 7 06:54:53 SentryBox portsentry[2979]: attackalert: Host: xxxxxxxx.xxxx.x-xx.xx.xxud.fr/193.49.xx.xxx i
s already blocked Ignoring

Detect 6

The server 100.10.1.12 is an IIS web server that hosts customer web sites. The host 100.10.1.4 is the firewall with a web proxy running. It sits between the clients network (including the web servers) and the internet.

This looks like it could possibly be an information gathering probe. This page (_vti_inf.html) is a page that is generated by Microsoft FrontPage and may be used in recon attempts. If there is a possibility that the web proxy is responding to public requests and hiding those addresses behind it's private address, that would be a problem. I tested the proxy from the outside and found that it was in fact NOT answering requests from the public interface. This would indicate that the logged traffic came from a user inside the network who was probably using a WebTrends (see reference to WebTrends in the GET) product to check the website for problems, monitoring the log or something along those lines.

Follow-up: Saturday AM, I logged this same type of traffic. At this point, I modified the IDS to log this particular signature on port 8080 (web proxy) as well as port 80. This will point to the actual machine that's generating this activity and I am assuming it will probably be somebody in the Web development team.

Initial log entry

Apr 5 11:57:29 xxxxxxxxx snort: IIS vti_inf access attempt: 100.10.1.4:3626 -> 100.10.1.12:80

Packet detail

```
04/05-11:57:29.667439 0:50:4:6B:1C:11 -> 0:60:8:A5:AF:AE type:0x800 len:0x148
100.10.1.4:3626 -> 100.10.1.12:80 TCP TTL:128 TOS:0x0 ID:5652  DF
*****PA* Seq: 0x360C58FF   Ack: 0xAF13FEAD   Win: 0xFFFF
GET /_vti_inf.html HTTP/1.0..Connection: keep-alive..Host: www.x
xx.xxxxxxxxx.xx.xx..User-Agent: WebTrends/3.0 (WinNT)..Referrer:
 http://www.WebTrends.com/webtrend.htm..From: info@WebTrends.com
..Accept: */*..Accept: image/gif..Accept: image/x-xbitmap..Accep
t: image/jpeg.....
```

Saturday AM log

```
04/08-11:57:16.753245 0:50:4:6B:1C:11 -> 0:60:8:A5:AF:AE type:0x800 len:0x148
100.10.1.4:1241 -> 100.10.1.12:80 TCP TTL:128 TOS:0x0 ID:2576  DF
*****PA* Seq: 0x23AC8F2C   Ack: 0xA3BBD23   Win: 0xFFFF
```

Detect 7

In this detect, 'monitor' is a machine running snort that is sitting behind the firewall.

I have been running snort on this network for about a month and detected a number of portscans yesterday and today from 3 IP addresses, 111.1.11.11, 222.222.222.22 & 333.33.333.333. In this case, a portscan is defined as 7 ports in 2 seconds. I did some research on the sites and found that 1st site is the web site for the San Francisco Examiner (an on-line newspaper). The 2nd site is the mail server for a local high school and the 3rd site (the one that got the most rule hits) seems to be part of a business-related ISP of some sort. They advertise a worldwide presence on their web site so this host could belong to anybody.

I searched the logs and found identical entries on March 20, 21, 23 and 30 between 8:30 and 9:30 in the morning coming from the 2nd host. I found similar activity from the 3rd host on March 31st between 8:15 and 8:20 and between 11:55 and 12:10. I found no similar activity from the 1st host. I maintain a log of machines that have been detected running portscans against machines that I work with and I was not able to find either of these last two IP addresses in any of my logs.

I feel that this activity warrants a little more attention. I have modified the IDS to log all packets to and from these hosts so that we can determine if this is in fact 'normal' behavior. This is one case where Shadow would have provided much more information and it would not be necessary to wait for the additional information.

It may seem a little odd to be grouping 3 totally unrelated source addresses into one 'incident' but my reason for that is timing. This site is not picking up any portscanning activity other than this group perhaps worked together on April 6 & 7.

The severity of this is rather high because the monitor is behind the firewall. This would indicate that the firewall needs to be tightened up quite a bit. It is also possible that the scans are coming from the inside.

Initial log

```
Apr  6 07:46:57 monitor snort: spp_portscan: PORTSCAN DETECTED from 111.1.11.11
Apr  6 08:15:05 monitor snort: spp_portscan: PORTSCAN DETECTED from 222.222.222.22
Apr  6 08:38:16 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 08:41:47 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 08:46:17 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 08:47:02 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 08:53:27 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 08:55:58 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 08:57:31 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  6 12:51:21 monitor snort: spp_portscan: PORTSCAN DETECTED from 111.1.11.11
Apr  7 07:08:29 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 07:13:25 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 07:15:06 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 07:15:20 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 07:16:20 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 08:26:41 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 08:36:20 monitor snort: spp_portscan: PORTSCAN DETECTED from 333.33.333.333
Apr  7 12:42:27 monitor snort: spp_portscan: PORTSCAN DETECTED from 111.1.11.11
Apr  7 12:46:53 monitor snort: spp_portscan: PORTSCAN DETECTED from 111.1.11.11
```

Detect 8

In this detect, linux (100.100.100.88) is a linux box running the snort IDS. 63.22.222.2 is the public address of a firewall I was setting up earlier today. 10.6.251.1 was the address of my notebook behind the firewall (NATted).

I noticed that one of our boxes saw the initial log entries below in the syslog file. The thing that makes this noteworthy is that the host that initiated the trace (this address was not sanitized) is using an address that is reserved for internal use. I then looked in the snort logs to get more detail and noticed this additional entry. I recognized the source address as a client where I was setting up NAT on a new internet connection this afternoon. Shortly before I left the site, internet routing stopped working. I left planning to resume work on this on Monday. From this trace, it seems obvious that I messed up something on the NAT configuration 'cuz the internal addresses aren't being NATted any more.

One other interesting this about this trace is the amount of port 137 traffic that my notebook (NT) was sending. My first trace in this assignment was of a 'port 137 scan' that was in fact only a friendly NT web host doing an SMB query on the box that was connecting to it. Here in this example, my notebook is doing an SMB attachment of some type to one of our boxes. It seems like there is so much port 137 'stuff' in my logs (I didn't analyze more of it here) that it's gonna be difficult to watch for SMB scans because that's almost part of the normal operation of the OS it seems.

Initial log entry

```
Apr  7 17:11:26 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:11:28 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:11:29 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:12:47 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:12:48 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:12:50 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:15:10 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:15:12 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:15:13 linux snort: SMB Name Wildcard: 10.6.251.1:137 -> 100.100.100.88:137
Apr  7 17:16:16 linux snort: Windows Traceroute: 10.6.251.1 -> 100.100.100.88
```

Snort logs

```
04/07-16:26:48.992967 0:0:C:90:2B:19 -> 0:50:DA:B6:6F:D4 type:0x800 len:0x5C
63.22.222.2:59920 -> 100.100.100.88:137 UDP TTL:121 TOS:0x0 ID:1292
Len: 58
83 EE 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..
```

Detect 9

This trace has not been sanitized by me – I have included it exactly as posted on the GIAC site in http://www.sans.org/y2k/040500-1230.htm.

Snort seems to be the IDS that is running in all three cases.

It seems that there are traces from three separate machines (dns1 – a.b.c.34, dns2 – a.b.c.66 & dns3 – a.b.c.98) that were attacked by three other machines (194.27.40.19, 208.185.54.22, 216.160.38.5). I assume that dns1, dns2 & dns3 are all on the same network because they all start with a.b.c and because the TTLs from dns2 & dns3 are the same for 216.160.38.58. I would also say that this is a coordinated attack – the recon started at 8:49 on the 3rd. In fact, that may have been late recon or pre-planning at that point since it was a very heavy scan (assuming that the log was trimmed). I am guessing that the log was trimmed a little since we see the portscan being started at 49:22 and ending at 49:34 but we only see log entries for 49:22. This attack seems to be specifically geared toward gaining access to this particular computer at that time. I would be interested in scanning all the logs for the machines in this network for traffic from any of the three hosts.

In the 1st section, it seems like we have a portscan coming from 208.185.54.22, 216.160.38.58 against dns1. I would think that this is probably not an instance of spoofed addresses because the destination port is incrementing.

In the 2nd section, we have 216.160.38.58 attempting to gain info from dns2 & dns3 on port 111. Port 111 is typically used by portmapper (sunrpc). This is a very commonly exploited port. This info is picked up by snort running on each of the boxes.

We then see a FIN-scan being run by 194.27.40.19 against all three dns boxes. We know that this is in fact a hostile attempt and not a case of hitting the wrong ip address or normal traffic because the source ports and the sequence numbers are the same in all three cases.

I hesitate to offer too much in the way of suggestions because I know so little about the network but I need to post something as port of my analysis. I would say that this is a fairly serious threat but it seems like the countermeasures are fairly well in place also. I am assuming (based on the post to GIAC and the information included) that the host is fairly secure – not running portmapper, probably using tcpwrappers and running a reasonably up-to-date OS. I think that port 111 should be blocked at the firewall. The 'problem hosts' like the three noted here could be thrown into some type of a filter but there are some problems associated with that: 1)the dialin 'problem' for the 3rd isn't the same guy at that address today. This seems (based on the monthly report) like the kind of a system that's almost under constant attack. Having additional traffic information would be helpful in this case because then we could see if there were any responses from the DNS servers. One simple option would be to use tcpdump to snort to capture all traffic to and from 208.185.54.22 & 194.27.40.19 to a file to allow subsequent analysis. There is no sense capturing the traffic from 216.160.38.58 because it's a dialup connection although it might be interesting to watch traffic on the 216.160 subnet to see if more traffic is detected there. If there is, it could be possible that the attacker has a dialup account with that ISP.

(DNS server (5) + possible root access (5)) – (System countermeasures (5) + Network countermeasures (2))

GIAC Trace from 4/5/00

Abovenet Communications, Inc., San Jose CA, USA

```
Apr 3 08:49:22 dns1 snort[4415]: spp_portscan:
PORTSCAN DETECTED from 208.185.54.22
Apr 3 08:49:28 dns1 snort[4415]: spp_portscan: portscan status
from 208.185.54.22: 14 connections across 1 hosts: TCP(0), UDP(14)
Apr 3 08:49:34 dns1 snort[4415]: spp_portscan: End of portscan
from 208.185.54.22
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33512 UDP
```

```
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33513 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33514 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33515 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33516 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33517 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33518 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33519 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33520 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33521 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33522 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33523 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33524 UDP
Apr 3 08:49:22 208.185.54.22:33161 -> a.b.c.34:33525 UDP
**********
```

Eutech, MPLS MN, USA (dialupM58.mpls.uswest.net)

```
Apr 3 12:56:39 dns1 snort[4415]: IDS013 - RPC -
portmap-request-mountd: 216.160.38.58:761 -> a.b.c.34:111
--------
[**] IDS013 - RPC - portmap-request-mountd [**]
04/03-12:56:39.550530 216.160.38.58:761 -> a.b.c.34:111
UDP TTL:49 TOS:0x0 ID:47954
Len: 64
7A 62 57 13 00 00 00 00 00 00 00 02 00 01 86 A0 zbW.............
00 00 00 02 00 00 00 03 00 00 00 00 00 00 00 00 ................
00 00 00 00 00 00 00 00 00 01 86 A5 00 00 00 01 ................
00 00 00 11 00 00 00 00 ........
--------
Apr 3 12:56:39 dns3 snort[9658]: IDS013 - RPC -
portmap-request-mountd: 216.160.38.58:750 -> a.b.c.98:111
--------
[**] IDS013 - RPC - portmap-request-mountd [**]
04/03-12:56:39.480862 216.160.38.58:750 -> a.b.c.98:111
UDP TTL:49 TOS:0x0 ID:47947
Len: 64
0B 3A 2F 6B 00 00 00 00 00 00 00 02 00 01 86 A0 .:/k............
00 00 00 02 00 00 00 03 00 00 00 00 00 00 00 00 ................
00 00 00 00 00 00 00 00 00 01 86 A5 00 00 00 01 ................
00 00 00 11 00 00 00 00 ........
**********
```

Zonguldak Karaelmas Universitesi, Turkey

```
Apr 3 14:33:46 dns1 snort[4415]: spp_portscan:
PORTSCAN DETECTED from 194.27.40.19
Apr 3 14:33:46 dns1 snort[4415]: IDS027 - SCAN-FIN:
194.27.40.19:47850 -> a.b.c.34:23
Apr 3 14:33:52 dns1 snort[4415]: spp_portscan: portscan status
from 194.27.40.19: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH
Apr 3 14:33:58 dns1 snort[4415]: spp_portscan: End of portscan
from 194.27.40.19
Apr 3 14:34:04 dns1 snort[4415]: spp_portscan: PORTSCAN DETECTED
from 194.27.40.19
Apr 3 14:34:04 dns1 snort[4415]: IDS027 - SCAN-FIN:
194.27.40.19:47850 -> a.b.c.34:23
Apr 3 14:34:10 dns1 snort[4415]: spp_portscan: portscan status
from 194.27.40.19: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH
Apr 3 14:34:16 dns1 snort[4415]: spp_portscan: End of portscan
from 194.27.40.19
--------
```

```
Apr 3 14:33:46 194.27.40.19:47850 -> a.b.c.34:23 FIN ***F****
Apr 3 14:34:04 194.27.40.19:47850 -> a.b.c.34:23 FIN ***F****
--------
[**] IDS027 - SCAN-FIN [**]
04/03-14:33:46.924487 194.27.40.19:47850 -> a.b.c.34:23
TCP TTL:229 TOS:0x0 ID:37380
***F**** Seq: 0x64780000 Ack: 0x0 Win: 0x200
00 00 00 00 00 00 ......
[**] IDS027 - SCAN-FIN [**]
04/03-14:34:04.467750 194.27.40.19:47850 -> a.b.c.34:23
TCP TTL:229 TOS:0x0 ID:37380
***F**** Seq: 0x96780000 Ack: 0x0 Win: 0x200
00 00 00 00 00 00 ......


Apr 3 14:33:48 dns3 snort[9658]: IDS027 - SCAN-FIN:
194.27.40.19:47850 -> a.b.c.98:23
--------
[**] IDS027 - SCAN-FIN [**]
04/03-14:33:48.206721 194.27.40.19:47850 -> a.b.c.98:23
TCP TTL:229 TOS:0x0 ID:39424
***F**** Seq: 0x64780000 Ack: 0x0 Win: 0x200
00 00 00 00 00 00 ......

Apr 3 14:33:47 dns2 snort[5950]: spp_portscan: PORTSCAN DETECTED
from 194.27.40.19
Apr 3 14:33:47 dns2 snort[5950]: IDS027 - SCAN-FIN:
194.27.40.19:47850 -> a.b.c.66:23
Apr 3 14:33:53 dns2 snort[5950]: spp_portscan: portscan status
from 194.27.40.19: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH
Apr 3 14:33:59 dns2 snort[5950]: spp_portscan: End of portscan
from 194.27.40.19
Apr 3 14:34:05 dns2 snort[5950]: spp_portscan: PORTSCAN DETECTED
from 194.27.40.19
Apr 3 14:34:05 dns2 snort[5950]: IDS027 - SCAN-FIN:
194.27.40.19:47850 -> a.b.c.66:23
Apr 3 14:34:11 dns2 snort[5950]: spp_portscan: portscan status
from 194.27.40.19: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH
Apr 3 14:34:17 dns2 snort[5950]: spp_portscan: End of portscan
from 194.27.40.19
--------
Apr 3 14:33:47 194.27.40.19:47850 -> a.b.c.66:23 FIN ***F****
Apr 3 14:34:05 194.27.40.19:47850 -> a.b.c.66:23 FIN ***F****
--------
[**] IDS027 - SCAN-FIN [**]
04/03-14:33:47.562600 194.27.40.19:47850 -> a.b.c.66:23
TCP TTL:229 TOS:0x0 ID:4354
***F**** Seq: 0x64780000 Ack: 0x0 Win: 0x200
50 10 16 CE D2 80 P.....
[**] IDS027 - SCAN-FIN [**]
04/03-14:34:05.102481 194.27.40.19:47850 -> a.b.c.66:23
TCP TTL:229 TOS:0x0 ID:4354
***F**** Seq: 0x96780000 Ack: 0x0 Win: 0x200
50 10 21 80 9C 3E P.!..>
--
```

Detect 10

In this detect, localhost is a rather old linux box (Red Hat 5.2) that is running portsentry. This is my home firewall and it doesn't have any unneeded ports open. The other computer aaa.bbb.142.xx is a sentry on the outside of a firewall.

This seems like a rather standard scan. This would be qualified as recon. The severity is low on this system because NFS is not running and tcp-wrappers was implemented to specifically deny this ip address using portsentry as soon as the scan happened.

I've checked other sites for this particular host and have logged one hit from another site that is using the same ISP. The other site also has NFS turned off and portsentry running. The other site got hit 2 minutes before the site that is logged below. These two sites are fairly far apart numerically for having only 2 minutes between scans. The addresses are aaa.bbb.218.62 and aaa.bbb.142.xx so this seems like a very fast scan or possibly a loss of clock synchronization because of the recent time change. I don't have access to the other box so I can't verify the clocks or the exact IP address.

The severity of this traffic is fairly low (unless they take out MY box!!). Both boxes have uneeded ports closed and neither one has any particularly valuable information on it.

Initial log entry

Apr 4 22:58:46 localhost tcplog[368]: port 635 connection attempt from p152_165.kyungpook.ac.kr:13464

Last Updated: February 24th, 2018

Upcoming Training

SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, United Arab Emirates	Apr 07, 2018 - Apr 12, 2018	Live Event
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
Baltimore Spring 2018 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Apr 23, 2018 - Apr 28, 2018	vLive
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201805,	May 02, 2018 - Jun 14, 2018	vLive
Community SANS Virginia Beach SEC503	Virginia Beach, VA	May 07, 2018 - May 12, 2018	Community SANS
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
Mentor Session - SEC503	Dulles, VA	May 24, 2018 - Jun 28, 2018	Mentor
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event
Minneapolis 2018 - SEC503: Intrusion Detection In-Depth	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	vLive
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LA	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, Japan	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Network Security 2018	Las Vegas, NV	Sep 23, 2018 - Sep 28, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced