Global Information Assurance Certification Paper

Intrusion Detection Practical

Level Two Intrusion Detection In Depth
GCIA Practical Assignment
Version 2.8
Conference Attended: New Orleans 2001

Charles L. Hutson

April 4, 2001

Assignment 1 – Network Detects

1. Detect #1: Persistent Port 80 Scan

```
URL: http://www.sans.org/y2k/032001.htm
Submitted by: Sven Olensky
Date: 3/20/01
Submission Comments: "they mainly checked for IIS Unicode vulnerability."

1)  20:47:21.034593 aph-aug-101-1-3-151.abo.wanadoo.fr.1811 >
208.193.36.3.www: S 4201115446:4201115446(0) win 16384  (DF)

2)  20:47:21.175221 aph-aug-101-1-3-151.abo.wanadoo.fr.radacct >
208.193.36.5.www: S 4201414990:4201414990(0) win 16384  (DF)

3)  20:47:23.444599 aph-aug-101-1-3-151.abo.wanadoo.fr.1834 >
208.193.36.16.www: S 4204893175:4204893175(0) win 16384  (DF)

4)  20:47:23.748518 aph-aug-101-1-3-151.abo.wanadoo.fr.1840 >
208.193.36.19.www: S 4205746136:4205746136(0) win 16384  (DF)

5)  20:47:23.754542 208.193.36.19.www > aph-aug-101-1-3-
151.abo.wanadoo.fr.1840: S 950796544:950796544(0) ack 4205746137 win 16384
(DF)

6)  20:47:23.916062 aph-aug-101-1-3-151.abo.wanadoo.fr.1840 >
208.193.36.19.www: . ack 1 win 17424 (DF)

7)  20:47:23.924765 aph-aug-101-1-3-151.abo.wanadoo.fr.1840 >
208.193.36.19.www: P :19(18) ack 1 win 17424 (DF)

8)  20:47:24.169434 aph-aug-101-1-3-151.abo.wanadoo.fr.radacct >
208.193.36.5.www: S 4201414990:4201414990(0) win 16384  (DF)
```

1.1 Source

This trace was obtained from the SANS GIAC website. The specific URL, submitter, submission date and relevant comments by the submitter are provided at the beginning of the trace. For reference purposes, I've manually numbered each line in the trace and will refer to them throughout the analysis.

1.2 Detect generated by

The output of the trace was generated by the tcpdump program. It's unclear from the submission whether the actual data was captured by a running tcpdump program or by another program, such as snort, capable of storing the data in a raw tcpdump format. The operating system and location of the detecting server are also unknown. The only thing known for certain about the detecting system is that it was located in a position between the Internet and the 208.193.36.X network which allowed for this capture.

1.3 Probability the source was spoofed

Because the 3-way TCP handshake was successfully completed between aph-aug-101-1-3-151.abo.wanadoo.fr and 208.193.36.19 in line #6, it is extremely unlikely that the source address was spoofed.

1.4 Description of attack

Upon review of the tcpdump data above, it seems apparent that the attacking system (aph-aug-101-1-3-151.abo.wanadoo.fr or 193.252.111.151) is searching for systems on the 208.193.36.x network running web services. This is typical reconnaissance activity for a hacker who is looking to exploit a specific vulnerability, in this case, most likely a web server vulnerability. Although the trace doesn't give us enough in-depth information to determine the exact nature of the intended web attack, the submitter comments indicate that the attacker did try to exploit a Unicode bug on IIS servers.

1.5 Attack mechanism

Upon initial analysis, it seemed as though this particular trace may have been more than just a typical scan for running web servers. While there are curious anomalies in the trace, it ultimately was simply a scan for running web servers. Items and anomalies of specific interest are analyzed below:

Targeted Systems

There seems to be no discernable pattern regarding the choice of target systems. As shown in trace lines 1-4, certain systems in the 208.193.36.0 range are scanned while others aren't. The target IP's do ascend but in no discernable pattern. This may indicate that prior reconnaissance was done which led the attacker to explore these specific targets further but this isn't evident in the provided trace. Another possibility is that all systems in the 208.193.36.0 address ranger *are* being probed but perhaps in a pseudo-random method determined by the attacker's program.

Attacker's Source Ports

The attacker's source ports are interesting in a couple of ways. First, they do not increase sequentially as seen in the difference between lines 1 and 2 as well as between 3 and 4. While not entirely uncommon, it is interesting. The attacking system may be scanning other targets on another network or perhaps his system is busy opening legitimate connections. Either way the system is relatively busy since 6 new TCP connections were opened in 3/10 of a second as illustrated in the timestamps of lines 3 and 4. It is possible, but not likely, that the packets are arriving out of order and that the other, missing source ports are simply in another area of our trace log.

The second interesting fact regarding source ports is related to the source port reported as "radacct" (lines 2 and 8). The detecting system apparently mapped this source port number to an entry in its /etc/services file. Unfortunately, we do not have the detecting system's /etc/services file to know the source port for sure. Internet searches show that UDP port 1646 is commonly registered as "radacct" but I was unable to find any reference to "radacct" and TCP numbers. It's common, however, to use the same port number for the UDP and the TCP versions of a

service. Not having any other information to go on, I'm assuming this is source port 1646. If this is, in fact, true then the source port ordering becomes even more interesting. Line 1 shows a source port of 1811, line 2 a source port of 1646 (radacct), and line 3 a source port of 1834. It's very unlikely on most systems that an ordering such as this would be used. Even with TCP/IP patches and modifications that are used to better randomize source port selection, its unlikely this particular ordering would be selected because there is a pseudo-pattern (most packets are in the ascending 1800 range.) This suggests that packet crafting is likely to be taking place.

Successful 3-way Handshake and Data Exchange

As shown in lines 4-6, the attacker completes a 3-way handshake with a service running on port 80 on the 208.193.36.19. In line 7, data is pushed and more communication takes place between the two systems. Unfortunately, we don't have the hex output from tcpdump to further analyze the communication. Again, the submitter claims an IIS Unicode exploit was attempted.

Attempts by the analysis team to connect to port 80 on 208.193.36.19 with a web browser failed as of 3/21/01. Due to that, we are unable to determine what service is actually running or what its intended use was.

Try, Try Again

One of the most interesting, and possibly misleading, parts of this analysis has to do with the fact that in line 8, the attacking system attempts to connect to port 80 on 208.193.36.5 again. The packet looks exactly the same as the first connect attempt in line 2. The same source port of "radacct" (1646?) is used, the same TCP sequence number (4201414990) but it takes place roughly 1 second later. My "first reaction analysis" was that my suspicion of packet crafting had been validated since the second packet looked exactly the same as the first. After further consideration and research, this second attempt is most likely a simply TCP Retry after the first attempt in line 2 had failed. TCP is a PAR (Packet Acknowledgement with Retransmission) protocol which means that every X number of packets in a TCP segment should be acknowledged by the receiver or the packet should be retransmitted by the sender. In this particular case, the attacking system probably did not receive the proper acknowledgement (ACK flag set with correct Sequence Number), if any, and is simply retransmitting. Hence the reliability features of TCP. If that is true, then an obvious question would be "Then why don't we see a retransmission of the packet in line 1 with a source port of 1811?". Possible answers could be that its further in the detect logs which are unknown to us or that that particular TCP connection has not yet timed out. TCP uses an adaptive retransmission algorithm based on the performance of the connection. It's possible that the connection attempt in line 1 has not timed out yet.

1.6 Attack correlation

As described above, the attacking system (aph-aug-101-1-3-151.abo.wanadoo.fr) is searching various systems with services running on port 80. Once a response is received it is likely that the next step would involve one or more Web server exploitation attempts. While the submitter suggests that the attacker then attempted to exploit the IIS Unicode vulnerability, we will assume that this isn't the case for now. With that assumption we only have a few pieces of relevant information from which to correlate this attack:

1) The source IP

2) Attacks against port 80 in general.
3) Attacks with similar source ports or suspicious source ports such as "radacct".

Searching the Common Incident Detection database at www.incidents.org/cid reveals the following information regarding these points:

- Attacks or probes on port 80 were relatively active with a total of 664.
- The attacking IP was not actively involved in any other attacks.
- The destination IP was not attacked by other systems.

Searching other incident databases located at www.securityfocus.com revealed no record of activity on either the source address, source port, or destination address.

The conclusion is that while scans for running web servers are common, this specific attacking system doesn't appear to be involved in anything widespread or covert.

1.7 Evidence of active targeting

There is no real evidence of active targeting here. The attacker appears to be searching a range of IP addresses for an active web server on which to attempt an exploit. Once a response on port 80 is received the attacker does target that system for further connection attempts.

1.8 Severity

The severity of this attack is a 4.

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

$$(5 + 2) - (1 + 2) = 4$$

- Criticality = 5. Without knowing the details of the systems being attacked on the 208.193.36.0 network we are unsure about the importance of the role they play for the organization. That uncertainty, in itself, increases the criticality in my opinion. I would rather be too concerned than not concerned enough. The fact that 208.193.36.19 responded to a request on port 80 indicates that it may have been an Internet or Intranet web server. As stated above, recent attempts to connect with this system on port 80 failed. We'll assume the highest level of criticality.
- Lethality = 2. The only verifiable activity provided by this trace involves reconnaissance. A simple scan for servers running a web server is being performed but nothing more. If the submitter's comments regarding ISS Unicode attacks are being attempted then the severity would increase depending on the web server being used, its patch level, etc.
- System Countermeasures = 1. Again, without knowing the details of the systems being attacked on the 208.193.36.0 network we are unsure about their level of countermeasures. We'll have to assume that no system countermeasures are being enforced to ensure the appropriate level of urgency.
- Network Countermeasures = 2. The submitter, unfortunately, does not give us details regarding the network infrastructure. We do, however, know that there is a server recording network traffic and an analyst reviewing the logs for interesting detail. For that

reason, we know that some countermeasures are in place. Aside from that, again, we'll assume the worst.

1.9 Defense recommendation

It's unclear whether or not the trace data was captured with an IDS or if it was just a thorough administrator using tcpdump and diligently perusing logs. An IDS, such as SNORT, would have identified this as a specific attack (possibly IIS Unicode as mentioned by the submitter) and would have stored decoded more of the data for more in-depth analysis. A properly configured firewall would block unauthorized attempts to connect to servers such as 208.193.36.19. Because the 208.193.36.19 server no longer responds to connect attempts on port 80, this potentially has already been accomplished. If the 208.193.36.19 web server is a server that is supposed to be accessible by the Internet community then the following countermeasures should also be implemented:

- Perform operating system "hardening" by removing unneeded services, reconfiguring insecure defaults, etc.
- Ensure OS and Web server patches are consistently up-to-date.
- Run the HTTPD service in a chrooted environment and as a non-privileged user.

1.10 Exam Question

20:47:21.175221 aph-aug-101-1-3-151.abo.wanadoo.fr.radacct > 208.193.36.5.www: S 4201414990:4201414990(0) win 16384 (DF)
20:47:24.169434 aph-aug-101-1-3-151.abo.wanadoo.fr.radacct > 208.193.36.5.www: S 4201414990:4201414990(0) win 16384 (DF)

The following is evident about the above trace:

a) The packet was crafted because the time stamp is different but the Initial Sequence Number is the same.
b) The packet was crafted because the time stamp is different but source port is the same.
c) The probing system is attempting to determine if RADIUS accounting is enabled on the destination system.
d) None of the above

Answer: d
(a) and (b) are incorrect because having a different timestamp and duplicate Initial Sequence Number or source port alone don't necessarily indicate packet crafting. When data is sent to a destination system using TCP, an ACK is expected in return to acknowledge the successful receipt of data by the receiving system. If the sending system doesn't receive the ACK response within a given time period, the packet is resent. This is most likely what is happening in the above example. Answer (c) is also incorrect because port 80 is the port being probed, not the RADIUS accounting services. The radacct service runs on an ephemeral port (1646) and was most likely chosen at random by the TCP program on the attacking system. Tcpdump mistakenly associated a name in /etc/services (radacct) with the source port 1646.

2. Detect #2: Possible coordinated attack from 2 sources

URL: http://www.sans.org/y2k/032201-1700.htm
Submitted by: Richard Leach
Date: 3/22/01
Submission Comments: "Greetings all, Had 2 surprising scans in my daily log file checks today. Both of these occurred in the past 24 hours, and both originated from addresses controlled by Sprint Corporate Security, as you'll see below. Makes you wonder... "

Scan Source #1: 207.52.185.152

Unsuccessful NetBIOS name resolution attempt #1

```
1)  03/20/01 19:50:02.959651 207.52.185.152.netbios-ns > bond03.netbios-ns:
udp 50
2)  03/20/01 19:50:02.959752 bond03 > 207.52.185.152: icmp: bond03 udp port
netbios-ns unreachable [tos 0xc0]
3)  03/20/01 19:50:02.959903 bond03 > 207.52.185.152: icmp: bond03 udp port
netbios-ns unreachable [tos 0xc0]
```

Strange incomplete of 3-way handshake

```
4)  03/20/01 19:50:03.102128 207.52.185.152.9786 > forrites.my.net.netbios-
ssn: S 3609477791:3609477791(0) win 16384  (DF)
5)  03/20/01 19:50:03.102397 forrites.my.net.netbios-ssn >
207.52.185.152.9786: S 3390403946:3390403946(0) ack 3609477792 win 8760  (DF)
```

Unsuccessful NetBIOS name resolution attempt #2

```
6)  03/20/01 19:50:04.142458 207.52.185.152.netbios-ns > my.net.165.netbios-
ns: udp 50
7)  03/20/01 19:50:04.142774 my.net.165 > 207.52.185.152: icmp: my.net.165
udp port netbios-ns unreachable [tos 0xc0]
```

Potential successful NetBIOS name resolution attempt

```
8)  03/20/01 19:50:04.146180 207.52.185.152.netbios-ns > my.net.164.netbios-
ns: udp 50
9)  03/20/01 19:50:04.146484 my.net.164.netbios-ns > 207.52.185.152.netbios-
ns: udp 175

10)  03/20/01 19:50:04.171978 207.52.185.152.netbios-ns > my.net.169.netbios-
ns: udp 50
```

Scan Source #2: 206.230.84.152

Network mapping via ping

```
11)  03/20/01 19:49:57.788171 206.230.84.152 > 255.255.255.255: icmp: echo
request
12)  03/20/01 19:49:57.788387 bond03 > 206.230.84.152: icmp: echo reply
13)  03/20/01 19:49:57.788588 my.net.165 > 206.230.84.152: icmp: echo reply
14)  03/20/01 19:49:57.788658 my.net.178 > 206.230.84.152: icmp: echo reply
15)  03/20/01 19:49:57.802824 206.230.84.152 > my.net.132: icmp: echo request
16)  03/20/01 19:49:57.802891 206.230.84.152 > my.net.133: icmp: echo request
17)  03/20/01 19:49:57.802958 206.230.84.152 > my.net.131: icmp: echo request
18)  03/20/01 19:49:57.806516 206.230.84.152 > my.net.134: icmp: echo request
```

```
19)  03/20/01 19:49:59.771931 206.230.84.152 > my.net.135: icmp: echo request
20)  03/20/01 19:50:00.756475 206.230.84.152 > my.net.136: icmp: echo request
21)  03/20/01 19:50:01.019015 206.230.84.152 > my.net.132: icmp: echo request
22)  03/20/01 19:50:01.037263 206.230.84.152 > my.net.131: icmp: echo request
23)  03/20/01 19:50:01.037397 206.230.84.152 > my.net.134: icmp: echo request
24)  03/20/01 19:50:01.040953 206.230.84.152 > my.net.133: icmp: echo request
25)  03/20/01 19:50:01.059202 206.230.84.152 > 255.255.255.255: icmp: echo
request
26)  03/20/01 19:50:01.059285 bond03 > 206.230.84.152: icmp: echo reply
27)  03/20/01 19:50:01.059520 my.net.178 > 206.230.84.152: icmp: echo reply
28)  03/20/01 19:50:02.864437 206.230.84.152 > my.net.146: icmp: echo request
29)  03/20/01 19:50:02.868101 206.230.84.152 > my.net.147: icmp: echo request
30)  03/20/01 19:50:02.868168 206.230.84.152 > my.net.148: icmp: echo request

Successful 3-way handshake
31)  03/20/01 19:50:02.981573 206.230.84.152.9782 > forrites.my.net.netbios-
ssn: S 3609233012:3609233012(0) win 16384  (DF)

32)  03/20/01 19:50:02.981743 forrites.my.net.netbios-ssn > 33)
206.230.84.152.9782: S 3390403730:3390403730(0) ack 3609233013 win 8760  (DF)

33)  03/20/01 19:50:03.040018 206.230.84.152.9782 > forrites.my.net.netbios-
ssn: F 3609233013:3609233013(0) ack 3390403731 win 17520 (DF)

34)  03/20/01 19:50:03.040077 206.230.84.152.9783 > forrites.my.net.netbios-
ssn: S 3609304933:3609304933(0) win 16384  (DF)

35)  03/20/01 19:50:03.040162 forrites.my.net.netbios-ssn >
206.230.84.152.9782: F 3390403731:3390403731(0) ack 3609233014 win 8760 (DF)

36)  03/20/01 19:50:03.040231 forrites.my.net.netbios-ssn >
206.230.84.152.9783: S 3390403823:3390403823(0) ack 3609304934 win 8760  (DF)

37)  /20/01 19:50:03.098486 206.230.84.152.9783 > forrites.my.net.netbios-
ssn: F 3609304934:3609304934(0) ack 3390403824 win 17520 (DF)

38)  03/20/01 19:50:03.098634 forrites.my.net.netbios-ssn >
206.230.84.152.9783: F 3390403824:3390403824(0) ack 3609304935 win 8760 (DF)

More Ping Scans
39)  03/20/01 19:50:04.041890 206.230.84.152 > my.net.174: icmp: echo request
40)  03/20/01 19:50:04.041959 my.net.171 > 206.230.84.152: icmp: echo reply
41)  03/20/01 19:50:04.043665 206.230.84.152 > my.net.175: icmp: echo request
42)  03/20/01 19:50:04.043735 206.230.84.152 > my.net.176: icmp: echo request
43)  03/20/01 19:50:04.047343 206.230.84.152 > my.net.172: icmp: echo request

Web Connect Attempts
44)  03/20/01 19:50:04.080208 206.230.84.152.9790 > my.net.132.www: S
  3609847590:3609847590(0) win 16384  (DF)
45)  03/20/01 19:50:04.087517 206.230.84.152.9793 > my.net.131.www: S
3609946979:3609946979(0) win 16384  (DF)
46)  03/20/01 19:50:04.087654 206.230.84.152.9794 > my.net.134.www: S
  3609984510:3609984510(0) win 16384  (DF)
```

2.1 Source

This trace was obtained from the SANS GIAC website. The specific URL, submitter, submission date and relevant comments by the submitter are provided at the beginning of the trace. For reference purposes, I've manually numbered each line in the trace and will refer to them throughout the analysis.

Furthermore, because of the amount of data submitted in this trace, the separate sources and the varying types of attacks, I've added white space and titles to make the log more readable. The ordering and grouping of the entries has not been modified in any way.

2.2 Detect generated by

The output of the trace was generated by the tcpdump program. It's unclear from the submission whether the actual data was captured by a running tcpdump program or by another program, such as snort, capable of storing the data in a raw tcpdump format. The operating system and location of the detecting server are also unknown. The only thing known for certain about the detecting system is that it was located in a position between the Internet and the submitter's "my.net.X" network which allowed for this capture.

2.3 Probability the source was spoofed

Since there were 2 IP addresses involved in this attack, this question must be answered individually for each.

Using IP researching capabilities of www.dshield.org the following information was discovered about the two, attacking IP addresses:

IP Address: 207.52.185.152
HostName: 207.52.185.152
Whois:
```
Sprint (NETBLK-SPRINTIPDIAL-BLKC) SPRINTIPDIAL-BLKC207.52.0.0 -
207.52.255.255
Sprint Corporate Security (NETBLK-SPRINT-CF34B9) SPRINT-CF34B9
                                                    207.52.185.0 -
207.52.185.255
```

IP Address: 206.230.84.152
HostName: 206.230.84.152
Whois:
```
Sprint (NETBLK-SPRINTLINK-BLKQ)       SPRINTLINK-BLKQ 206.228.0.0 -
206.231.255.255
Corporate Security (NETBLK-SPRINT-CEE655) SPRINT-CEE655
                                                    206.230.84.0 -
206.230.85.255
```

Confirming the research of the submitter, both source IP's are owned by Sprint Corporate Security.

Source 1: 207.52.185.152
It is possible that the source IP address 207.52.185.152 used in the first scan was spoofed due to the fact that there is never a successful 3-way handshake. Possible evidence to support this is shown on lines 4-5 titled "Incomplete 3-way handshake" where a SYN is sent by the attacker, a SYN-ACK returned by the target but no ACK returned by the attacker to complete the 3-way handshake. This would be expected if the source IP was spoofed.

Source 2: 206.230.84.152
It is unlikely that the IP address of the second attacking system was spoofed due to the completion of a TCP 3-way handshake as illustrated in lines 31-38 titled "Completion of 3-way handshake.

Taking into consideration that both IP address ranges are owned by the same company, it seems unlikely that either IP address was spoofed. Most hackers spoof source IP addresses when they want to confuse the analyst with a decoy, perform Denial of Service attacks (such as smurf) on the spoofed source, etc. A spoofed IP of 207.52.185.152 does nothing to protect the hacker since it's within the same company.

2.4 Description of attack

Two systems originating from systems registered to Sprint Corporate Security appear to be probing multiple systems on the submitter's network. Based on the time stamps recorded in the detect log, the attacks happened at the same time which suggests cooperation by the two attacking systems despite their differing source IP addresses. The following attempts to gain information about the submitter's network were attempted:

- NetBIOS name resolution attempts
- Network mapping via ping
- Attempts to connect with or locate web servers

The attack is most likely an attempt to gain reconnaissance about the system to plan a more thorough attack in the future. It is also possible that this data is the result of a misconfigured vulnerability assessment for a legitimate client by Sprint's security team. The probe should be treated as malicious until that theory is (in)validated.

2.5 Attack mechanism

This particular detect is interesting for a several reasons. A thorough analysis of the attack is provided below.

Source IP Addresses
The first point of interest about this detect is the fact that two IP addresses are attacking systems within the submitter's network simultaneously. At first glance, the network portion of the IP addresses are dissimilar enough (207.52.185 and 206.230.84) to suggest that two attackers at two separate locations are coordinating an attack. As shown previously, information from www.dshield.org reveals that the two IP ranges are owned by the same company. Further inspection shows that the host portion of the IP address is the same on both systems (x.x.x.152).

Most likely this isn't a coincidence. Referencing the timestamps for both attacks shows us that they took place within 1-2 seconds of each other. Its almost a sure bet that the attacking systems are being used by a single attacker or a single team.

Scan #1 Detail: Unsuccessful NetBIOS name resolution attempt #1
The first attacking system (207.52.185.152) attempts to gather information about 5 systems on the submitter's network referred to as "my.net.x". The first attempt is to perform a NetBIOS name resolution on bond03. NetBIOS traffic is extremely common on the Internet and is often referred to as "background noise". This is shown in lines 1-3 where the attacking system sends a packet with source port of UDP 137 (netbios-ns) to the same source port on bond03. On Microsoft based systems when an IP address can't be resolved using DNS NetBIOS is used. Most operating systems use the generic call gethostbyaddr() when attempting to map an IP address to a hostname. Gethostbyaddr() can use different resources to retrieve this information such as DNS, NIS, WINS, local host tables, etc. The order is dictated by the flavor of operating system. When this lookup is determined, the questioning system (the attacker in our case) performs a "Node Status" which has a source port of 137 and a destination port of 137. This is what we see in lines 1-3. In our particular example, bond03 replies with an ICMP port unreachable. While the attacker was unable to discern the hostname, he may have been able to determine that bond03 exists due to the ICMP response. Because there was apparently no firewall blocking this inbound attempt, it's unlikely that any firewall blocked the ICMP response.

A recon attack like this could have been attempted with basic Microsoft OS commands such as nbstat or through more malicious programs like legion. An excellent explanation of typical NetBIOS traffic can be found at http://www.robertgraham.com/pubs/firewall-seen.html - 10

Scan #1 Detail: Strange incomplete of 3-way handshake
In lines 4-5 it seems as though the attacking system has found that netbios-ssn (TCP 139) responds on a system known as forrites.my.net. Netbios-ssn is another service associated with NetBIOS specifically registered as the NetBIOS Session Service. This port is used primarily for communication between two systems in the Microsoft world and is often exploited on DSL and cable modem users that have insecure Windows desktops. Specifically programs like SMBScanner can be used to search for poorly considered services such as file sharing. In our case, the attacker found a system willing to complete the 3-way handshake but decided not to send the final ACK. As stated above, this could imply that the attacking IP address was spoofed. Another possible explanation could be that the attacker wished to avoid completing the 3-way handshake for fear of drawing attention to himself.

Scan #1 Detail: Unsuccessful NetBIOS name resolution attempt #2
As shown in lines 6-7, the attacker tried another system on the submitter's network known as my.net.165. See the previous explanation for more detail.

Scan #1 Detail: Potential successful NetBIOS name resolution attempt
In lines 8-10 we see our first response to a NetBIOS "Node Status" command. A system known as my.net.164 replies to the attacking system with a UDP packet of length 175. The detail of the trace makes in impossible to tell what information may have been passed back but, assuming a

firewall doesn't block the response, the attacker now knows that NetBIOS is alive and well on the remote system.

Scan #2 Detail: Network mapping via ping
In the second scan, our second attacker (206.230.84.152) joins in to attempt to gain information about even more systems. First, the attacker attempts to make the network via ping. The goal being to find all possible systems on a network very quickly. Line 11 shows a ping of the entire broadcast address which will cause listening systems to respond. Several systems do respond to the ICMP echo requests and alert the attacker (lines 13, 14, 26, 2740) of their existence including my.net.165,178,bond03, 171). For good measure, the attacker even pings several specific IP address as shown in lines 15-25 and 39-43.

The most interesting aspect of this section is line 11. Typically, most properly configured Internet routers will now forward requests to a destination of 255.255.255.255 since it is a broadcast address. For that reason, this type of destination should only be seen on a local subnet. Because of that, I suspect that the submitter's network is very close, if not a part of, Sprint's network.

Scan #2 Detail: Successful 3-way handshake
In our first successful 3-way TCP communication in this detect, we see on lines 31-38 that the attacker and the system forrites.my.net open and close 3 TCP connections without passing any data whatsoever. Had it not been for this single piece of data, I may have strongly suspected that the source IP addresses were spoofed. The attacking system completed connects to the netbios-ssn (UDP 139) which purpose was explained in the analysis of lines 4-5. An extremely interesting point is a correlation between the source port of these lines (9782) and the source port of lines 4-5 (9786) supposedly originating from a completely separate system! The fact that both attacking systems have a .152 as the host portion of the IP and such a close source port lead me to believe that they may, in fact, be two interfaces on the same system or that the packets are crafted, again, by the same attacker.

Scan #2 Detail: Web Connect Attempts
Finally, the attacking system performs a quick probe for running web services on systems my.net.131, 132, 134 to complete their information gathering run on the submitter's network.

2.6 Attack correlation

As detailed in the description and mechanism analysis above, there is definitely a correlation between the two scans submitted. For summary purposes, these include:

- Both IP addresses are owned by Sprint Corporate Security (207.52.185.x and 206.230.84.x).
- The host portion of the IP address is .152 in both cases.
- As shown on lines 4 and 31, the source port on two supposedly separate systems is very close (9786 and 9782 respectively).
- Target systems were duplicated by both attacking systems. Bond03 for example.

Searching other incident databases located at www.securityfocus.com and www.dshield.org revealed no record of activity on either the source address, source port, or destination address.

2.7 Evidence of active targeting

There is no real evidence of active targeting here. The attacker appears to be performing reconnaissance on several systems in the submitter's range. Furthermore, there is no evidence of intelligent follow up attacks on systems that respond to initial probes but that may come later.

2.8 Severity

The severity of this attack is a 3.

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

$$(4 + 1) - (1 + 1) = 3$$

- Criticality = 4. Without knowing the details of the systems being attacked on the my.net network we are unsure about the importance of the role they play for the organization. That uncertainty, in itself, increases the criticality in my opinion. I would rather be too concerned than not concerned enough. We'll assume a reasonably high level of criticality but not the highest.
- Lethality = 1. The only verifiable activity provided by this trace involves reconnaissance. Nothing more of significance was attempted….yet.
- System Countermeasures = 1. Again, without knowing the details of the systems being attacked on the my.net network we are unsure about their level of countermeasures. We'll have to assume that no system countermeasures are being enforced to ensure the appropriate level of urgency.
- Network Countermeasures = 1. The submitter, unfortunately, does not give us details regarding the network infrastructure. It is doubtful, in my opinion, that a firewall is being used to protect his infrastructure as shown by the fact that simple NetBIOS 137-139 traffic gets to internal systems. Even if a firewall is in place, it's not configured in a very secure manner. Finally, it is likely that a network intrusion detection system is not in place since no NIDS logs were supplied.

2.9 Defense recommendation

A properly configured firewall is a core component to the perimeter defense of any organization. It seems that this site either has a poorly configured firewall or no firewall at all as evidenced by the fact that simple, Internet based NetBIOS traffic is allowed through. Furthermore, the only detect information that we have is tcpdump output. The following countermeasures should also be implemented:

- Implement a firewall to deny inbound traffic unless it is explicitly allowed. Especially unneeded traffic such as NetBIOS 137, 139 and ping.
- Deploy a network based intrusion detection system for better logging. SNORT would have provided more in-depth information into the payload of packets. This would have been useful in the analysis of 8-9 and 31-38 to discern what type of NetBIOS

communication may have taken place. Furthermore, it would have provided better alerting to the submitter.

2.10 Exam Question

03/20/01 19:50:02.959651 207.52.185.152.netbios-ns > bond03.netbios-ns: udp 50

The above traffic could be which of the following:

a) 207.52.185.152 attempting to perform an IP to hostname mapping.
b) A DNS zone transfer from 207.52.185.152
c) A DNS query response by 207.52.185.152
d) None of the above

Answer: a
Both (b) and (c) are DNS related which would show activity on TCP or UDP ports 53. (a) is a likely explanation to a trace such as this.

3. Detect #3: Suspicious pings from multiple sources

```
URL: http://www.sans.org/y2k/032401-1230.htm
Submitted by: Gerald Swann
Date: 3/24/01
Submission Comments: "A tcpdump seen this morning on a machine with cable
modem connection."

Ping Request 1
1) 06:58:18.859170 pc-62-31-21-136-sh.blueyonder.co.uk > mycom.mynet: icmp:
echo request
2) 07:00:23.971973 m53-mp1-cvx1a.lis.ntl.com > mycom.mynet: icmp: echo
request
3) 07:00:23.972459 mycom.mynet > m53-mp1-cvx1a.lis.ntl.com: icmp: echo reply

Ping Request 2
4) 07:01:03.616221 mcns245.docsis166.singa.pore.net > mycom.mynet: icmp: echo
request
5) 07:01:03.616699 mycom.mynet > mcns245.docsis166.singa.pore.net: icmp: echo
reply

Ping Request 3
6) 07:01:38.157608 63-83-209-133.nas08.siouxfalls.sdnet.net > mycom.mynet:
icmp: echo request
7) 07:01:38.158086 mycom.mynet > 63-83-209-133.nas08.siouxfalls.sdnet.net:
icmp: echo reply

Ping Request 4
8) 07:01:42.027344 cc972755-a.mdltwn1.nj.home.com > mycom.mynet: icmp: echo
request
9) 07:01:42.027776 mycom.mynet > cc972755-a.mdltwn1.nj.home.com: icmp: echo
reply

Ping Request 5
10) 07:02:25.892080 cc972755-a.mdltwn1.nj.home.com > mycom.mynet: icmp: echo
request
11) 07:02:25.892486 mycom.mynet > cc972755-a.mdltwn1.nj.home.com: icmp: echo
reply

Ping Request 6
12) 07:03:00.487342 waba-cas5-cs-13.dial.bright.net > mycom.mynet: icmp: echo
request
13) 07:03:00.487796 mycom.mynet > waba-cas5-cs-13.dial.bright.net: icmp: echo
reply
```


3.1 Source

This trace was obtained from the SANS GIAC website. The specific URL, submitter, submission date and relevant comments by the submitter are provided at the beginning of the trace. For reference purposes, I've manually numbered each line in the trace and will refer to them throughout the analysis.

Furthermore, because of the amount of data submitted in this trace, the separate sources and the varying types of attacks, I've added white space and titles to make the log more readable. The ordering and grouping of the entries have not been modified in any way.

3.2 Detect generated by

Judging by the format of the logs, the trace was generated by the tcpdump program as reported by the submitter. The operating system and location of the detecting server are also unknown. The only thing known for certain about the detecting system is that it was located in a position between the cable modem enabled Internet and the submitter's "my.net.X" network which allowed for this capture.

3.3 Probability the source was spoofed

It is extremely likely that the majority of these IP addresses were spoofed due to the fact that so many ping requests were generated by so many diverse IP address ranges in such a short time.

3.4 Description of attack

Several possibilities exist that explain what may be happening. A most likely possibility is that a single attacker is attempting to gather information about possible targets to attack using the ping command. In an effort to "put up a smokescreen" the attacker is likely spoofing the source IP addresses of several system in purposefully diverse locations. By filling the target's logs with faulty information and suggesting a "globally coordinated attack", the attacker hopes to "pepper his trail" so to speak.

Another unlikely possibility is that the submitter's IP address (mycom.mynet) is susceptible to unwittingly taking part in a smurf attack as a "smurf amplifier". If this were the case, the attacker would be actively targeting six, different, unrelated, seemingly unimportant systems across the globe. The smurf attack, being one of the most well known attacks, has been documented thoroughly in the Internet community. Due to that fact, and the fact that this is most likely not a smurf attack, the topic will not be re-covered in this analysis.

3.5 Attack mechanism

A single system on the submitter's network is being sent ICMP echo request packets by six different systems located in unrelated locations across the Internet. The ICMP echo requests were sent from these seemingly unrelated source IP addresses in only a five-minute time span. Interestingly enough, the submitter's system happily responds to these pings.

The first task in the analysis was to determine if the systems were in any way related. After all, if the systems had something in common then the possibility of a coordinated attack from the seven systems was more likely.

Using nslookup to map the hostnames to IP addresses produced the following mapping:

Hostname	IP
pc-62-31-21-136-sh.blueyonder.co.uk	62.31.21.136
m53-mp1-cvx1a.lis.ntl.com	62.254.36.53

mcns245.docsis166.singa.pore.net	202.156.166.245
63-83-209-133.nas08.siouxfalls.sdnet.net	63.83.209.133
cc972755-a.mdltwn1.nj.home.com	65.11.218.138
waba-cas5-cs-13.dial.bright.net	216.201.28.15

Then using the capabilities of www.dshield.org to find out more information about the hosts, the following information was discovered about the six, attacking IP addresses

IP Address: 62.31.21.136

HostName: pc-62-31-21-136-sh.blueyonder.co.uk

Whois:

```
% Rights restricted by copyright. See
http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:      62.31.0.0 - 62.31.39.255
netname:      HSD-EDI
descr:        Edinburgh HSD platform
country:      GB
admin-c:      MG645-RIPE
tech-c:       SB264-RIPE
status:       ASSIGNED PA
changed:      mike@cableinet.net 20000328
source:       RIPE

route:        62.30.0.0/15
descr:        Cable Internet
descr:        UK ISP
origin:       AS5462
notify:       netmail@cableinet.net
mnt-by:       AS5462-MNT
changed:      mike@cableinet.net 20001012
source:       RIPE

person:       Mike Garrett
address:      Cable Internet Ltd.
address:      Unit 2, Genesis Busines Park
address:      Woking, Surrey
address:      GU21 5RW
phone:        +44 1483 251 842
fax-no:       +44 1483 251 810
e-mail:       Mike@cableinet.net
nic-hdl:      MG645-RIPE
changed:      mike@cableinet.net 19971112
source:       RIPE
```

IP Address: 62.254.36.53

HostName: m53-mp1-cvx1a.lis.ntl.com

Whois:

```
% Rights restricted by copyright. See
http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:      62.254.32.0 - 62.254.63.255
netname:      NTL
```

```
descr:       NTL Internet
descr:       Belfast site
country:     GB
admin-c:     NNMC1-RIPE
tech-c:      COH1-RIPE
status:      ASSIGNED PA
mnt-by:      AS5089-MNT
changed:     hostmaster@ntli.net 20010108
source:      RIPE

route:       62.252.0.0/14
descr:       NTL-UK-IP-BLOCK-3
origin:      AS5089
mnt-by:      AS5089-MNT
changed:     bob.procter@ntli.net 20010205
source:      RIPE

role:        NTLI Network Management Centre
address:     NTL Internet
address:     Crawley Court
address:     Winchester
address:     Hampshire
address:     SO21 2QA
phone:       +44 1633 670317
fax-no:      +44 1483 875150
e-mail:      nmc@ntli.net
trouble:     abuse@ntlworld.com (Internet abuse mailbox)
```

In the interest of conserving space here, I have decided not to include all the information discovered about the six systems. Unfortunately, there was no discernable pattern to the systems that supposedly initiated the ping requests to our submitter's system.

Turning back to the information documented in the tcpdump log there are two anomalies. First, the submitter's system chooses not to reply with an Echo Reply to the first system as shown in line 1. This is unusual since the system *did* reply to the requests of all other Echo Requests. There is no discernable reason why this may be the case. The only clue might be that the timestamp between the first and second Echo Request (shown in lines 1 and 2) is longer than the time between any of the other two Echo Request packets. It may mean that the submitter missed something in his cut-and-paste submission or it could be something more devious.

The second anomalous piece of information about this traffic is in lines 8 and 10. The cc972755-a.mdltwn1.nj.home.com sends two Echo Requests instead of one like all the others. This is further evidence of foul play in my opinion.

After careful analysis it appears that the best explanation is that these source IP addresses are being spoofed for some reason known only to the attacker. The likelihood of that many systems coordinating for no other purpose than to ping a single system is unlikely and not worth the effort. It is my opinion that the more likely explanation is that this is a decoy meant to take the analysts attention away from the real attacking source. A common approach by hacker's is to create "noise" in the target's environment to make it more difficult to discern the meaningful

data from the useless data. If the logs are cluttered enough and the analyst has been chasing false alarms for days, then it is possible that his guard would be down when the real attack comes in.

This technique can be deployed very easily with common tools such as nmap. This very attack could be duplicated with the following nmap command:

nmap –sP –P0 –D 62.31.21.136, 62.254.36.53, 202.156.166.245, 63.83.209.133, 65.11.218.138, 65.11.218.138 216.201.28.15 mycom.mynet

nmap is a very popular port scanning tool available at www.insecure.org. Briefly, the options listed above would accomplish the following:

- -sP = perform a ping scan to determine if hosts respond
- -P0 = don't ping the systems before doing the –sP command above
- -D = provides a list of decoy systems that look as if they're pinging too

Finally, the target system to ping is listed as the last argument. Interestingly, the nmap "man" page states that "in some port scan detectors, your system (the real system) may not even be logged!!!" Although we don't know that this is exactly what happened here, I believe it likely.

3.6 Attack correlation

As detailed in the description and mechanism analysis above, there appears to be no correlation between the systems listed as the sources of the ICMP Echo Requests.

Searching other incident databases located at www.securityfocus.com and www.dshield.org revealed no record of activity on either the source address or destination address.

There doesn't appear to be any widespread attacks involving these sources or destinations at this time.

3.7 Evidence of active targeting

If the theory I propose above is true then this would be a case of active targeting. An attacker has specifically targeted the mycom.mynet system to gain reconnaissance, clutter logs, etc.

3.8 Severity

The severity of this attack is a 3.

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

$$(4 + 1) – (1 + 1) = 3$$

- Criticality = 4. Without knowing the details of the system being pinged on the mycom.mynet network we are unsure about the importance of the role it plays for the submitter's organization. That uncertainty, in itself, increases the criticality in my opinion. I would rather be too concerned than not concerned enough. We'll assume a reasonably high level of criticality but not the highest.

- Lethality = 1. The only verifiable activity provided by this trace involves reconnaissance. Nothing more of significance was attempted….yet.
- System Countermeasures = 1. Again, without knowing the details of the mycom.mynet system we are unsure about its level of countermeasures. We'll have to assume that no system countermeasures are being enforced to ensure the appropriate level of urgency.
- Network Countermeasures = 1. The submitter, unfortunately, does not give us details regarding the network infrastructure. It is doubtful, in my opinion that a firewall is being used to protect his infrastructure as shown by the fact that that ICMP Echo Requests get through to internal systems. Even if a firewall is in place, it's not configured in a very secure manner. Finally, it is likely that a network intrusion detection system is not in place since no NIDS logs were supplied.

3.9 Defense recommendation

A properly configured firewall is a core component to the perimeter defense of any organization. It seems that this site either has a poorly configured firewall or no firewall at all as evidenced by the fact that simple, ICMP traffic is allowed through. Furthermore, the only detect information that we have is tcpdump output. The following countermeasures should also be implemented:

- Implement a firewall to deny inbound traffic unless it is explicitly allowed. Specifically, ICMP Echo Requests and Replies should be blocked to discourage the use of this system as a smurf amplifier.
- Deploy a network based intrusion detection system to provide real time (ish) alerting of events and further logging of payload data.

3.10 Exam Question

Your site can be used as a "smurf amplifier" if it allows the following:

a) Outbound ICMP Redirects
b) Outbound ICMP Echo Requests
c) Outbound ICMP Echo Replies
d) All of the above

Answer: c
The smurf attack uses a site as a smurf amplifier by sending it spoofed ICMP Echo Requests. It then sends outbound Echo Replies to the unfortunate owner of the spoofed address. If enough systems respond to the ping request that was never made by the victim, it is overrun and the attack is successful.

4. Detect #4: Valid DNS responses (False Alarm)

```
URL: http://www.sans.org/y2k/032101.htm
Submitted by: Hal
Date: 3/21/01
Submission Comments: Included in trace as presented on GIAC web site.

For the past few days, I've had this problem where once a day, at random
time, the .1 & .2 addresses shown below begin aggressively doing some odd
port scans. The .1 & .2 addresses are our ISP's DNS servers. After a few
minutes of this, all the memory resources on the firewall are consumed and it
has to reboot. Other times the firewall blocks the site, but then our DNS
becomes blocked also, so Internet traffic comes to a stop. As a financial
firm which requires our Bloomberg terminals to be up during the stock trading
period, not to mention portfolio security, this has become a real concern.
I've asked our ISP to examine this, but I have serious reservations about
their being able to decipher what their looking at. Here is a portion of the
log, sans last ocet of our IP.

1) 3/19 10:58:14   192.168.1.254 0   deny   in   eth0    udp  216.98.160.1
  216.98.161.xxx     53      63662    (blocked site)
2) 3/19 10:58:15   192.168.1.254 0   deny   in   eth0    udp  216.98.160.1
  216.98.161.xxx     53      63662    (blocked site)
3) 3/19 10:58:15   192.168.1.254 0   deny   in   eth0    udp  216.98.160.2
  216.98.161.xxx     53      63663    (blocked site)
4) 3/19 10:58:17   192.168.1.254 0   deny   in   eth0    udp  216.98.160.1
  216.98.161.xxx     53      63662    (blocked site)
5) 3/19 10:58:17   192.168.1.254 0   deny   in   eth0    udp  216.98.160.2
  216.98.161.xxx     53      63663    (blocked site)
6) 3/19 10:58:21   192.168.1.254 0   deny   in   eth0    udp  216.98.160.1
  216.98.161.xxx     53      63662    (blocked site)
7) 3/19 10:58:21   192.168.1.254 0   deny   in   eth0    udp  216.98.160.2
  216.98.161.xxx     53      63663    (blocked site)

After the firewall reboots or I reboot it, that specific type activity stops.
But, I notice these type entries, which appear a few times randomly
throughout the day. Nothing unusual, except that again, this is a DNS server,
which appears to have sent something to our Exchange server, as headers were
removed. But..I've only been doing minimal logging on the MTA, and I see
nothing unusual with the authenticating to the exchange server.

8) 3/19 15:10:28   192.168.1.254 0   allow  in   eth0    tcp  216.98.160.1
  216.98.161.xxx     15920   25       syn (SMTP)
9) 3/19 15:10:28   192.168.1.254 1   smtp-proxy [216.98.160.1:15920
  192.168.1.3:25] removing unknown or denied header "Importance"

If you could offer me any insight, or point me in a direction, would be
greatly appreciated. I would be happy to provide a list of all ports and
sites I have blocked at the firewall, and more of the log if necessary. I can
say that I do not have that site on the blocked list nor ports which would be
required for proper DNS. Since I don't have the ports that its trying to
access blocked either, I'm at a loss as to why the firewall is blocking it,
as it is set to auto-block sites which attempt to connect to a blocked port.
```

4.1 Source

This trace was obtained from the SANS GIAC website. The specific URL, submitter, submission date and relevant comments by the submitter are provided at the beginning of the trace. For reference purposes, I've manually numbered each line in the trace and will refer to them throughout the analysis.

4.2 Detect generated by

The output was generated by a firewall at the submitter's site. While I am unsure of the exact brand of firewall, it looks similar to that of Watchguard's firewalls. The log format is as follows: Date, Timestamp, Firewall-IP, Unsure, Action, Direction, Interface, Protocol, Src-IP, Dst-IP, Src-Port , Dst-Port

Unfortunately, no IDS logs were submitted.

4.3 Probability the source was spoofed

It is possible that the source UDP packets were spoofed but highly unlikely based on the analysis given below.

4.4 Description of attack

After several hours of research and careful analysis, it is the analyst's opinion that this is NOT an attack but rather normal response to DNS queries by a single system on the 216.98.161.0 network. The stimulus, a DNS lookup request by a system on the internal 216.98.161.0 network, was not submitted. This information was most likely not logged by the firewall since it was considered valid or the submitter didn't realize it was pertinent.

The reason for the firewall reboot is most likely due to the fact that the DNS servers located at the submitter's ISP (216.98.160.1 and 216.98.160.2) are desperately trying to respond to DNS queries that never seem to get filled. As time progresses, the DNS traffic becomes overwhelming and consumes all the memory resources available to the firewall. This is a problem inherent to stateful inspection based firewalls that must maintain an entry for all outbound UDP and TCP traffic.

It's possible that the submitter's DNS server at 216.98.161.xxx simply queries other DNS servers when these fail and users never see a DNS resolution problem.

4.5. Attack Mechanism

Note: Because this trace ended up being valid traffic and I had already invested several hours into the analysis, this section essentially describes the thought process and steps that I used to arrive at my conclusion. My hope is that other analysts can learn from my process and realize that not everything ends up being malicious.

As shown in lines 1 through 7, two DNS servers located at the submitter's ISP are sending several UDP packets at one or more systems inside the 216.98.161.0 network.. After hours of

analysis I realized that it would have been far beneficial had the submitter sanitized the first three octets of the IP address rather than the last one. Because of this, I was unable to ascertain whether the destination systems were one system or many. As described below, I now believe this to be a single system but, initially, was unsure of this. Knowing this information would have proven very helpful in the analysis.

At first glance (and for hours thereafter) I was suspicious of the source UDP port of 53 from the "attacking" systems. Attempts to research this at www.whitehats.com and www.securityfocus.com seemed to validate my suspicions by explaining that many hackers use a source port of 53 in an attempt to circumvent older firewalls that only validated that the source OR the destination ports were 53. Another reason to use source port 53 in an attack is that there are many firewalls on the Internet that have been mis-configured in an attempt to allow valid DNS traffic through. Sometimes these mis-configurations allow any connection with a source port of 53 to any server instead of allowing connections to any destination port of 53.

Armed with this information (or misinformation) I began to theorize that perhaps these two systems (216.98.160.1 and 216.98.160.2) had been compromised. This led me to do some basic reconnaissance on the two servers. Using basic nslookup commands and a web browser I discovered the following:

- 216.98.160.1 was not only a DNS server for the ISP but also a web server.
- 216.98.160.2 was not only a DNS server for the ISP but also an email server.
- The servers did, in fact, belong to an ISP (www.nwark.com)

With a quick visit to www.netcraft.com, which can be used to gather basic web server and operating system information, I determined the following information about the Web server (and consequently one of the DNS servers since it runs on the same server):

- They're running on Linux.
- They're running Apache 1.2.5 (an older version).

The theory that perhaps these systems had been compromised seemed to hold some water. After all, the servers were being used for more than just DNS so the secure concept of "separation of duties" was not being adhered to. Inherently, the more services that run on a server the more insecure it is. Furthermore, they were running an older version of the Apache Web server which surely has some vulnerabilities and implies that they ISP may not be keeping things as upgraded and patched as they should (sorry!). Suspecting a possible Trojan horse had been planted on the unsuspecting ISP, I began searching Incident databases in an attempt to find other records of attacks with a source port of UDP 53 and a destination port of 63662 or 63663. The fact that the two, separate DNS servers were targeting system(s) on the 216.98.161.0 network at very specific ports (63662 and 63663), I suspected that these two ports were being used to communicate with a new Trojan via UDP. Not knowing if the destination system was one or several servers due to the obfuscation by the submitter as well as not having record of a stimulus, this idea had to be explored. Unfortunately, no correlation was found in either www.incidents.org/cid or www.dshield.org.

Suspecting that I may have been on a wild goose chase, I began to question the assumption that UDP source port 53 was really suspicious. Referencing the trusty O'Reilly DNS and BIND

Handbook yielded no useful results. Finally, searches on www.google.com lead me to the following link:

```
Source: http://www.intac.com/~cdp/cptd-faq/section2.html#ports
Date: Wed Jun 16 21:57:36 EDT 1999
```

The following table shows what TCP/UDP ports bind before 8.x DNS uses to send and receive queries:

```
Prot Src   Dst   Use
udp  53    53    Queries between servers (eg, recursive queries)
                 Replies to above
tcp  53    53    Queries with long replies between servers, zone
                 transfers Replies to above
udp  >1023 53    Client queries (sendmail, nslookup, etc ...)
udp  53    >1023 Replies to above
tcp  >1023 53    Client queries with long replies
tcp  53    >1023 Replies to above

Note: >1023 is for non-priv ports on Un*x clients. On other client
      types, the limit may be more or less.
```

BIND 8.x no longer uses port 53 as the source port for recursive queries, nor uses it as the destination port for corresponding replies. By default it uses a random port >1023, although you can configure a specific port (and it be port 53 if you want).

Another point to keep in mind when designing filters for DNS is that a DNS server uses port 53 both as the source and destination for its queries. So, a client queries an initial server from an unreserved port number to UDP port 53. If the server needs to query another server to get the required info, it sends a UDP query to that server with both source and destination ports set to 53. The response is then sent with the same src=53 dest=53 to the first server which then responds to the original client from port 53 to the original source port number.

The point of all this is that putting in filters to only allow UDP between a high port and port 53 will not work correctly, you must also allow the port 53 to port 53 UDP to get through.

Also, ALL versions of BIND use TCP for queries in some cases. The original query is tried using UDP. If the response is longer than the allocated buffer, the resolver will retry the query using a TCP connection. If you block access to TCP port 53 as suggested above, you may find that some things don't work.

Newer version of BIND allow you to configure a list of IP addresses from which to allow zone transfers. This mechanism can be used to prevent people from outside downloading your entire namespace.

Knowing now that UDP source port 53 from a DNS server could be valid responses to a DNS query, I began to make headway. With an understanding that the traffic in trace output lines 1-7 was most likely a response rather than a stimulus, the pieces began to fall into place. The obfuscated destination 216.98.161.xxx is most likely a single system's BIND daemon making

DNS requests to both 216.98.160.1 and 216.98.160.2. When the 216.98.161.xxx system generated the stimulus packets, it chose a source port of 63662 and 63663 respectively. This would definitely explain the pattern to the two destination ports of 63662 and 63663.

Finally, lines 8 and 9 show a TCP port 25 connection into the submitter's Exchange server from the ISP system 216.98.160.1. Port 25 is, of course, SMTP traffic. This is most likely the ISP's server attempting to notify a contact at the submitter's site that something is amiss.

Although the submitter's firewall is brought down by the activity, as described above, this is likely not a malicious attack.

4.6 Attack correlation

There was no attack to correlate. In section 5.6 above, I described how I attempted to correlate my data with that of others. It was the lack of correlating data that led me to suspect that I may be running down the wrong path and eventually lead to a more valid analysis.

4.7 Evidence of active targeting

There is no malicious activity here therefore no active targeting.

4.8 Severity

Due to the fact that this isn't truly an attack, the use of the algorithm below provides for an interesting exercise.

Ironically, the severity of this attack is a 7. Because this environment creates an ongoing denial-of-service "attack" the severity is high.

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

$$(5 + 4) – (1 + 1) = 7$$

- Criticality = 5. While this isn't a true attack per se, the systems involved are most likely critical to the operation of the submitter's site. They include the firewall (192.168.1.254) and an internal DNS server (216.98.161.xxx).
- Lethality = 4. The result of this activity causes a successful denial-of-service attack on the submitter's firewall.
- System Countermeasures = 1. Any system countermeasures that may be in place will not provide any assistance with this particular situation.
- Network Countermeasures = 1. The firewall is configured in such a way that it actually is more of a hindrance than a help in this particular example. Even though the firewall may be configured "properly" to protect the internal network, it's not helping at all with this particular "attack". Hence the low rating.

4.9 Defense recommendation

A properly configured, stateful inspection firewall would, most likely have allowed the return UDP connection to successfully be delivered. It's my theory that because these responses were

not being allowed, the problem snowballed to a point where the firewall simply ran out of resources. A less expensive solution would be to create a rule in the firewall to allow UDP 53 packets originating on the two DNS servers (216.98.160.1 and 216.98.160.2) to pass through to the 216.98.161.xxx address. This would allow the DNS UDP 53 responses to get through properly.

4.10 Exam Question

```
3/19 10:58:15   192.168.1.254 0   deny   in   eth0    udp  216.98.160.1
  216.98.161.1     53      63662    (blocked site)
3/19 10:58:15   192.168.1.254 0   deny   in   eth0    udp  216.98.160.2
  216.98.161.1     53      63663    (blocked site)
```

The firewall log entries above are most likely:

a) Responses from 216.98.160.1 and 216.98.160.2 to DNS queries made by a DNS daemon running at IP address 216.98.161.1
b) Crafted packets originating on systems 216.98.160.1 and 216.98.160.2
c) Attempts to communicate with a Trojan horse program listening on ports 63662 and 63663 on 216.98.161.1.
d) DNS requests being made to a DNS server running on 216.98.161.1

Answer: a

As described throughout this detect, this is a valid response to a simple DNS query.

5. Detect #5: Hack-a-Tack Trojan signature

```
URL: http://www.sans.org/y2k/040301-1430.htm
Submitted by: Fredrik Ljunggren
Date: 4/3/01
Submission Comments: This is a log from Snort on a small subnet
195.163.112.80/27

Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.80:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.84:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.83:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.85:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.87:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.86:28431 UDP
```

5.1 Source

This trace was obtained from the SANS GIAC website. The specific URL, submitter, submission date and relevant comments by the submitter are provided at the beginning of the trace.

5.2 Detect generated by

As reported by the submitter, the output was generated by a Snort IDS located on a small subnet 195.163.112.80/27.

5.3 Probability the source was spoofed

It is possible that the source UDP packets were spoofed, however, highly unlikely due to that fact that this is a program looking for a specific Trojan program known as Hack-a-tack. This is the first step in searching for this Trojan. If the source IP address is spoofed, the attacker would not get the return information to act upon.

5.4 Description of attack

The probe submitted above is a search for a relatively well-known and recurring use of a Trojan program known as Hack-a-tack. As shown in the correlation section below, this attack shows up occasionally in the GIAC Current Detects section on the SANS web site. Because this pattern shows up relatively infrequently, it is apparently unknown to many would-be analysts based on the number of requests for explanation.

The attempt to contact a listening Hack-a-tack server on the submitter's internal network was unsuccessful since we did not see a response to the request.

5.5. Attack Mechanism

As described on the web site of the same name, The Hack-a-tack program (www.hack-a-tack.com) is a Trojan horse program much like Back Orifice which allows remote access and control of a compromised server. The program is broken into a client and server portion. Step 1 involves getting the server portion of the program onto a target system in some manner. This can involve obtaining physical access to a system and loading the software through a floppy or, more covertly, through a social engineering attempt that manipulates the end user into unintentionally

installing it. Whatever means is used, the server portion must be loaded first. Next, the client portion of Hack-a-tack is used to communicate with the server in order to establish a connection. Once this connection has been established, an entire host of things may be manipulated on the remote system including:

- Gather basic system information
- Pop up messages on the remote system
- Disable keys, change mouse key mappings, etc.
- Open/close CD-ROM
- Kill processes
- Log passwords
- Shutdown systems
- Etc.

Essentially, the attack gains complete control of the remote server.

There are two ways to get connected to a compromised system. The first method is to configure the client program to scan a range of IP addresses for a process listening on UDP port 28431. This is the default port used by the Hack-a-tack server daemon. This appears to be the method in use by the attacker in the submitter's detect information. Another indication of the Hack-a-tack Trojan is that the source port is typically UDP port 28432 also evident in this trace.

The second method, known as 'Transmit IP' is somewhat more interesting. With this option, the attacker can elect to upload his IP address to an FTP server that is queried by running Hack-a-tack servers. When one of these servers downloads the list of transmitted IP's, it proactively contacts the client software to let it know it has been successfully loaded on a system.

5.6. Attack Correlation

As shown below, this attack was emerged in the June 2000 timeframe when multiple sites reported the strange activity between UDP ports 28431 and 28432. My research indicates that the attack was first analyzed in that timeframe. Since then, the attack appears to surface occasionally to confuse intrusion analysts that hadn't seen it before.

A check of the most popular source ports on www.incidents.org/cid/ reveals that the 28431 port is used relatively frequently and ranked 197th for a total of 108 incidents. While not immensely popular, enough activity is present to warrant attention.

The information below was gathered from various detects on the GIAC section of the SANS website.

Collaborating Hack-a-tack attacks from www.sans.org

URL: http://www.sans.org/y2k/062100-1030.htm
Submitted by: Jens Hektor

```
Date: 6/21/01

> Hello,
> our intrusion detection facilities have detected a portscan from one
> of your machines. Portscans like this one usually precede concrete
> attacks towards our machine, therefor we consider this portscan as an
> "unfriendly act" against our computers.
> We think that someone is misusing your system (usually the machine
> we notice portscans from are cracked).
> Check your system and ensure that this does not happen again.
> Here follow the logs:
> ---------------------------------------

/var/log/advanced/local7.info:Jun 10 04:24:30 c6k-rz 143: Jun 10 04:24:29.838
  MEZS: %SEC-6-IPACCESSLOGP: list 100 denied udp 4.54.120.174(28432) ->
  134.130.74.57(28431), 1 packet
/var/log/advanced/local7.info:Jun 10 04:24:31 c6k-rz 144: Jun 10 04:24:30.966
  MEZS: %SEC-6-IPACCESSLOGP: list 100 denied udp 4.54.120.174(28432) ->
  134.130.74.74(28431), 1 packet
/var/log/advanced/local7.info:Jun 10 04:24:33 c6k-rz 145: Jun 10 04:24:32.222
  MEZS: %SEC-6-IPACCESSLOGP: list 100 denied udp 4.54.120.174(28432) ->
  134.130.74.192(28431), 1 packet


URL: http://www.sans.org/y2k/062700-0900.htm
Submitted by: Matt Scarborough
Date: 6/27/00
```

This trace has been shortened somewhat. Two other attackers connected to this victim machine, listed drive contents and grabbed (intentionally falsified) passwords and user info. Certainly the inbound TCP connections to the victim would have been blocked in a properly secured environment. But, would the outbound HTTP GET or the outbound UDP "pings" seeking handlers also have been blocked? UDP 28432 > 28431 to a single IP address might indicate a compromise. UDP 28432 > 28431 across a class C might indicate Trojan trolling. Matt 2000-06-23

Entering readback mode....

```
-*> Snort! <*-
Version 1.6-WIN32
By Martin Roesch)
WIN32 Port By Michael Davis
```

<snip> Hack'a'Tack 2000 in "scan above" mode looking for victims

```
06/22-10:13:59.065448 192.168.1.65:28432 -> 192.168.1.75:28431
UDP TTL:128 TOS:0x0 ID:1024 Len: 9
41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A...............00 00
```

```
URL: http://www.sans.org/y2k/010101.htm
Submitted by: Luis Mendoza
Date: 01/01/01

Hi: My firewall detected the following port scan through 28431 UDP port. Do you know what
trojan program use this port? Thanks Luis Mendoza

     1 M       [64.161.24.39]     [aaa.bbb.ccc.67]      60 0:00:00.000
0.000.000      12/27/2000 10:15:09 PM UDP: D=28431 S=28432  LEN=9
     2         [64.161.24.39]     [aaa.bbb.ccc.68]      60 0:00:00.014
0.014.389      12/27/2000 10:15:09 PM UDP: D=28431 S=28432  LEN=9
     3         [64.161.24.39]     [aaa.bbb.ccc.69]      60 0:00:00.014
0.000.179      12/27/2000 10:15:09 PM UDP: D=28431 S=28432  LEN=9
     4         [64.161.24.39]     [aaa.bbb.ccc.71]      60 0:00:00.027
0.013.057      12/27/2000 10:15:09 PM UDP: D=28431 S=28432  LEN=9
```

Attempts to correlate the source IP of the current attack with some of the attacks of 1 year ago using www.geektools.com showed no apparent similarities.

5.7 Evidence of active targeting

The pattern submitted indicates that this is essentially the client-side portion of the Hack-a-tack Trojan looking for systems that have been infected with the server-side portion of the Hack-a-tack Trojan. Because the client is just trolling for listening servers, this is assuredly not active targeting.

5.8 Severity

The severity of this attempt is a –2.

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

$$(1 + 1) – (1 + 3) = -2$$

- Criticality = 1. There is no indication that the servers targeted are in any way critical to the submitter's operations. Due to their random nature, it is unlikely these are of any significance.
- Lethality = 1. The probe here is relatively harmless and there is no indication that a successful connection was made.
- System Countermeasures = 1. There is no evidence here of any system countermeasures. We will assume that there are none of any significance to ensure the proper level of urgency is provided.
- Network Countermeasures = 3. While we have no indication of a firewall in place at this location, we do know that they have network based IDS (Snort in this case). Furthermore, an intrusion analyst is monitoring alerts and submitting them to GIAC for analysis. This implies a better than average level of network countermeasures.

5.9 Defense recommendation

The primary recommendation would be to update the Snort signature database at the submitter's site to include the signature for Hack-a-tack included below:

```
alert UDP $INTERNAL 28431 -> $EXTERNAL 28432 (msg: "IDS289/trojan-active-
hack-a-tack-2000"; content: "H"; depth: 1;)
```

Specific information for this attack can be found at http://www.whitehats.com/info/IDS289.

5.10 Exam Question

```
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.80:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.84:28431 UDP
Mar 30 05:29:02 62.100.195.96:28432 -> 195.163.112.83:28431 UDP
```

The Snort log entries are generated by which of the following:

a) Napster traffic.
b) Hack-a-tack Trojan.
c) Back Orifice Trojan
d) None of the above.

Answer: b

Assignment 2 – Intrusion Detection Systems: In-house vs. Outsourcing

6.1 Introduction

Intrusion detection systems have made significant in-roads in the marketplace over the past 2 years. This can be attributed primarily to the fact that, despite the widespread deployment of firewall technology, the number of compromised systems and networks has increased. Networks that were once thought to be "secured" because they were protected by a firewall have had their web sites defaced, exposed sensitive client information through DNS/BIND exploits, etc. Despite the wishes of most IT department decision makers, firewalls aren't the "silver bullets" that most firewall product vendors would have you think. Time has shown that deploying a firewall simply isn't enough. The IT community is discovering, painfully in some situations, that good information security requires multiple levels of protection more often known as Defense-In-Depth techniques. Intrusion Detection Systems (IDS) provide the next, logical technology layer of defense. Time will tell if IDS are being used as second layer in a Defense-In-Depth strategy or simply another "silver bullet".

With the popularity of IDS in the IT community growing, a critical decision must be made. Should this technology and it's expertise be outsourced or deployed and maintained in-house? In many cases the design, deployment and ongoing management of other security technologies such as firewalls and digital certificates are being outsourced. Many organizations are making the critical decision of whether or not they should outsource intrusion detection services as well. This document attempts to:

- Describe the challenges with IDS technology design, deployment and maintenance.
- Highlight the pros and cons of an in-house vs. an outsourced strategy.

(For the scope of this document, Intrusion Detection Systems (IDS) will refer solely to the use of network based intrusion detection systems.)

6.2 The Challenges with Security Technology

As previously stated, a good information security architecture is a process rather than a product. At some point, however, technologies (products) enter that process. When that time comes, it is important to remember that information security technologies are dramatically different from other technologies in several ways. A few of those key differences are:

1) **It's extremely easy to mis-configure a security technology and never know it**.

Firewalls, for example, are commonly mis-configured to allow far more traffic to pass through them than the administrator had intended. In the majority of cases, a $20k firewall is rendered useless by a misunderstood rule or default setting. The firewall blissfully allows the undesirable traffic without having any way of alerting the unsuspecting administrator.

2) Security technology requires constant "care and feeding".

Any security product vendor that tells you their solution "installs in minutes" and "requires no ongoing maintenance" has no idea what information security is all about. Unfortunately this addresses the vast majority of vendors. To quote well-known information security expert and author Bruce Schneier, "Security is a process, not a product." While other products like databases, disk storage arrays, email systems, etc. all require some routine maintenance, few of them can be rendered useless by missing a maintenance task like installing a patch. This is a stark reality with security technology.

3) The tools (products) aren't as important as the people who wield them.

It's common for a security technology, or set of technologies, to be considered the answer to the security problem. The common question of "What are you doing to secure your e.business initiatives?" is often met with "We've deployed the market leading firewall and IDS technologies". This response erringly places faith in a set of products alone to solve a "process" problem. Only knowledgeable people who understand the discipline of information security can effectively wield the tools and bring them to bear as part of a secure solution. Consider, for example, the construction of a home. The type of hammer being used to build the structure isn't as important as the carpenter using it or, even more so, the architect who designed the plans. A below average hammer can still be effective in the hands of a skilled carpenter. The reverse can not be said.

4) Being trained or "certified" on a security product isn't a good measure of security adeptness.

Unfortunately, security product certification tests are designed only to demonstrate knowledge of how to configure specific features of a given product. They do not (and can not) measure the candidate's overall information security adeptness. Knowing how to configure a given feature in a specific environment and whether you should or not are two different things. Being trained and certified on Microsoft Word doesn't make you a writer!

6.3 The Challenges with Intrusion Detection Technologies

Being one of the more complex security technologies, intrusion detection systems are subject to all the challenges of security technologies described above. Beyond these, there are several other challenges unique to IDS.

1) Still considered "early adopter" technology.

While they have matured greatly in the past couple of years, intrusion detection technology is essentially where firewall technology was 4-5 years ago. Configuration interfaces, alerting, ability to update alert signatures, etc. are still relatively cryptic compared to many of the more mature firewall products available today.

Furthermore, IDS is not seen as a necessity in many non-security circles. Most CIO's or IT Directors wouldn't dream of not having a firewall in 2001. The same can not be said of IDS

technology. Because firewalls are still considered the "security silver bullet", IDS hasn't been funded as well in many organizations. The result is that IDS technology remains somewhat of an unknown to many organizations and, therefore, there aren't as many peers available to bounce ideas off of. If you're an early adopter that has the wisdom to deploy a defense-in-depth strategy using IDS, you're most likely going to be pioneering it unless you're in an industry where information security has been a priority for some time (banking, finance, government, high profile e.commerce sites, etc.)

2) False positives comprise the majority of alerts received by security administrators.

Intrusion detection technology works by looking for known signatures in network traffic patterns. There are hundreds of attack signatures out there with new ones being created everyday. Unfortunately, legitimate traffic or legitimate mistakes by end users, can sometimes look like an attack. Some Managed Security Service Provider's (MSSP) have reported an average rate of 98% false positives!

3) Analysis of alerts or suspicious activity is extremely complex and time consuming

Most IDS provide easy-to-read, standard alerts and many vendors tout these in their product demonstrations, marketing material, etc. However, analysis of specific traffic patterns and alerts in order to discern whether the alert was a legitimate concern or a false positive requires extensive technical knowledge. This analysis requires a level of specialized knowledge that most companies, including those with network and operating system administrators, simply do not have. Training on the specific IDS technology, as stated above, simply isn't enough.

4) Special dedication is required to stay abreast of emerging attacks and to update IDS appropriately.

In the previous section, we mentioned that security technologies require a lot of "care and feeding". This is more true with IDS than with any other security technology. Even on a site with moderate Internet activity, for example, attacks will occur on a daily basis. The philosophy behind IDS is that this suspicious traffic must be recognized and a human being must be alerted in many cases. In order for IDS to be effective, attack signatures must be up-to-date, suspicious activity must be analyzed, well thought out responses must be taken, etc.

6.4. Build it or Outsource it?

Due to the factors listed above, many organizations are considering the outsourcing of all their security services in general. According to The Yankee Group:

- The large enterprise market is expected to grow from $268 million in 2000 to nearly $1.2 billion in 2005.
- The small-to-medium sized business market is expected to grow from $50 million in 2000 to $637 million in 2005.

Today, much of this market is comprised of managed firewall services but the addition of managed IDS is already available by dozens of firms ranging from security consulting firms to ISP's to dedicated MSSP companies. The pros and cons for each IDS approach must be weighed against what is most critical for the specific organization its being considered for. These are explored in more depth in the following sections.

6.5. Key Questions and Considerations

Before a decision can be made regarding an in-house or outsourced IDS solution, it is important to reflect on what the key considerations are. Once these are brought to light, it's relatively easy to map the goals and requirements of a specific business to a given approach. Inherently, the value placed on several of these key considerations moves you towards an in-house or outsourced decision relatively quickly. These considerations are identified and discussed in no particular order below.

1) How sensitive is the information being stored on the system or networks you're deploying intrusion detection systems on?
In many cases, allowing a 3rd party, such as a Managed Security Services Provider (MSSP), access to networks where sensitive client information flows us unacceptable. In these cases, the decision to use an MSSP may go against the corporate information security policy, best practices, etc. However, many MSSP's make the case that this is the perfect place to use them since the information is so sensitive. Their argument being "if it's that important, you'd better trust it to analysts who do this everyday!"

2) Does your organization have trained intrusion analysts?
Effectively analyzing and responding to intrusion alerts requires a very specialized knowledge unique even in the information security field. Being an effective intrusion analyst requires several things:

- Extensive TCP/IP experience at the packet decoding level.
- Extensive experience in varied operating systems.
- Experience with a wide range of "core" application technologies such as DNS, email, Web servers, etc.
- Intrusion analysis training such as that provided by GIAC.
- Training on specific IDS products.
- Incident response training

3) Is the culture at your organization conducive to outsourcing?

In many cases, the decision to outsource or not is made here alone. Many industry experts and authors, such as Geoffrey Moore in his book "Living on The Fault Line" make the recommendation that anything that isn't considered a "core competency" should be outsourced to organizations where it is. In these organizations, the vast majority of services are outsourced. Many organizations, however, do not take this stance and have a culture more conducive to building and maintaining it internally. Even in these environments, however, managed IDS have made inroads because they can be deployed as an extra level of protection on the Internet outside

the corporate firewall. In this scenario, the service can be treated almost like a "monitored home security system" where the model of allowing an external party to monitor technology is acceptable.

4) How in-depth should the analysis of suspicious events be?

The obvious answer is "As in-depth as it needs to be in order to determine what's going on and who is causing it?" In the world of IDS, this translates specifically to a few things:

- How much packet information should be captured?
- Should only events that trigger alerts be logged or normal traffic as well?
- How long should suspicious and normal information be stored?

In the eyes of many, hard-core intrusion analysis, the above questions are of the highest concern. Essentially, the more data that can be captured about a given event the better. When something suspicious occurs, the analyst needs to have as much data to analyze as possible. Furthermore, new attacks are often preceded by strange behavior that didn't trigger an alert. By logging "normal" traffic, the data is there to analyze at a later data should new attacks dictate.

Herein lies a major concern for organizations that want to have thorough logging. Most MSSP's don't log extensive information about the packets that flow through their IDS. Typically it's only the bare minimum required to store the alert in some meaningful form. This is often known as the packet header which stores the majority of the useful information. However, there are occasions where it is important to look, at a deeper level, into the application data payload. This would be unavailable to analysts in most MSSP. Furthermore, information (packets) about "normal" traffic isn't stored. Finally, information isn't kept for more than a couple of days so analysis of historical information is unlikely.

6.6. Keeping IDS In-House: The Pros and Cons

Several of the pros and cons for designing, deploying and maintaining the intrusion detection solution in house are shown in the matrix below:

Pros	Cons
• No external parties have access to sensitive internal information. • Analysis process is controlled internally. • Alerting response times are controlled internally. • Information logging is controlled internally.	• Initial infrastructure hardware and software must be procured. • Internal staff must be trained on IDS technology and security processes. • Internal staff must dedicate time to analysis of IDS alerts. • Internal staff must remain abreast of daily security alerts and vulnerabilities. • Internal staff must dedicate time to the maintenance of IDS. • IDS hardware and software must be periodically updated.

6.7. Outsourcing it: The Pros and Cons

Pros	Cons
• No staff to hire or train. • No infrastructure to design or deploy. • No infrastructure to maintain. • Allows you to focus on your core competencies.	• External parties may have access to sensitive data. • Analysis process is controlled by MSSP. • Alerting process is controlled by MSSP. • Reliance on an external organization that may not

Upon initial inspection, it appears that there are far more Cons than Pros to using the in-house approach. However, it may be that there is a specific Pro that your organization is unwilling or unable to do without. For example, organizations where traffic flowing to and from its networks is extremely sensitive, such as in the banking or healthcare industries, may want to consider deploying and maintaining an IDS in-house for the sole purpose of maintaining control of access to sensitive information. That reason alone may be enough to incur the negatives associated with the Cons in this example. Furthermore, thorough analysis of and response to events may be a very high priority in this same environment. Under a MSSP, you may be unsure of the level of thoroughness that you'll receive in this area since each MSSP is different.

Assignment 3 - Analyze This

7.1 Global Information

Numerous Snort IDS log output was submitted to ColdLabs by GIAC Enterprises for analysis. With this much data, it's important to quickly summarize the information in an effort to uncover patterns in attacks, sources, destinations, etc. Using the tools developed for this analysis (as described in "Assignment 3- The Process") the following summary information has been gathered. GIAC Enterprises corporate network uses the 192.168.x.x network range (manually modified from MY.NET.x.x in the logs).

First, there were 25 unique alerts that were logged in the submitted files. These are listed below:

Number of Alerts by Type

Number of Alerts	Alert Signature
105918	Watchlist 000220 IL-ISDNNET-990517
51192	SYN-FIN scan!
16146	DNS udp DoS attack described on unisog
5340	Tiny Fragments - Possible Hostile Activity
4238	connect to 515 from outside
2401	Watchlist 000222 NET-NCFC
2239	WinGate 1080 Attempt
2053	Attempted Sun RPC high port access
826	Null scan!
710	Queso fingerprint
591	SNMP public access
558	NMAP TCP ping!
546	Russia Dynamo - SANS Flash 28-jul-00
515	SMB Name Wildcard
204	SUNRPC highport access!
159	connect to 515 from inside
154	Broadcast Ping to subnet 70
100	TCP SMTP Source Port traffic
77	Back Orifice
59	External RPC call
8	Probable NMAP fingerprint attempt
2	site exec - Possible wu-ftpd exploit - GIAC000623
1	Happy 99 Virus
1	STATDX UDP attack
1	SITE EXEC - Possible wu-ftpd exploit - GIAC000623

It is apparent by the myriad attacks that the client's network has, indeed, been the target of some reconnaissance as well as some serious attempts at access on several systems.

Reviewing the "The Ten Most Critical Internet Security Threats" listed on www.sans.org/topten.htm reveals that several of the top 10 attacks were actually attempted against target systems within the client's network. These include:

- #3 Sun RPC Attacks
- #5 SMTP attacks (possibly but not certain)
- #7 Windows Files Sharing (SMB Attacks)
- #10 SNMP attacks

Unfortunately this doesn't give us much indication as to the level of experience of the potential attacker. These attacks are well documented, scripted and used by script kiddies as well as the experts.

The second important step in summarizing the information was to determine who the top attacking sites were. These are summarized below.

Top 20 Attack Source IP Addresses

Attack Occurrences	Attack Source	Hostname
48786	212.179.79.2	-
39015	212.179.27.111	clnt-27111.bezeqint.net
17604	211.34.40.1	-
16132	209.67.50.203	Futuresite.register.com
9878	195.56.182.206	-
8565	194.234.48.26	-
4563	212.179.95.5	cable-95005.bezeqint.net
4096	147.8.182.157	dhcp-2157.eee.hku.hk
3052	194.204.224.131	-
2353	212.179.77.20	-
1951	139.130.61.206	-
1790	200.194.102.99	server.7setembro.com.br
1580	194.197.170.7	www.turunhippos.fi
1517	212.179.44.105	bzq-44-105.bezeqint.net
1387	212.179.42.102	fr-c42102.bezeqint.net
1242	63.204.152.253	adsl-63-204-152-253.dsl.snfc21.pacbell.net
1221	212.179.38.135	clnt-38135.bezeqint.net
1054	212.179.58.12	-
1002	212.179.45.241	fr-c27241.bezeqint.net
926	212.179.56.5	dsl-213-023-056-005.arcor-ip.net
900	159.226.121.37	-

Interestingly a pattern emerged here. Systems in the **bezeqint.net** domain were responsible for 6 of the top 20 attacking hosts. This is explored in more detail further into the analysis.

With so much data to review, it's important to quickly determine where most of the activity is coming from in order to better target the investigation. Another important grouping of data is the top attacking networks. These consist of the networks with the most unique attacking IP addresses. These are listed below:

Top 10 Attacking Networks (First 2 Octets)

Unique Addresses	Network	Domain	Owner
158	63.253.	Splitrock.net	Splitrock Services Inc.
77	63.252	Splitrock.net	Splitrock Services Inc.
47	212.179	bezeqint.net	Bezeq International, Isreal
44	209.255	Splitrock.net	Splitrock Services Inc.
31	159.226	Ac.cn	The Computer Network Center Chinese Academy of Sciences
25	192.168	Internal Network	GIAC Enterprises
19	203.134	Primustel.com.au	Primus Telecommunications
18	63.254	Splitrock.net	Splitrock Services Inc.
17	205.188	Aol.com	America Online
16	209.156	Splitrock.net	Splitrock Services Inc.

Most likely, GIAC Enterprises will need to contact some of the administrators at these locations to assist in the data forensics. The information, gathered at www.geektools.com, will be included in the Appendix B for your convenience. Systems located at Splitrock.net, a nationwide ISP, were responsible for the top two attacking networks as well as the 3 other offending networks. It is further evidenced, as observed in the prior section, that the bezeqint.net domain has a high number of attackers.

Top 10 Attacking Networks (First 3 Octets)

Unique Addresses	Network	Domain	Owner
25	63.252.92	Splitrock.net	Splitrock Services Inc.
22	63.253.140	Splitrock.net	Splitrock Services Inc.
20	63.253.106	Splitrock.net	Splitrock Services Inc.
14	63.253.110	Splitrock.net	Splitrock Services Inc.
14	205.188.153	Aol.com	America Online
13	63.254.34	Splitrock.net	Splitrock Services Inc.
13	63.253.115	Splitrock.net	Splitrock Services Inc.
13	209.255.180	Splitrock.net	Splitrock Services Inc.

13	141.30.228	Dfn.de	Technische Universitaet Dresden
12	63.253.112	Splitrock.net	Splitrock Services Inc.

Activity by systems in the Splitrock domain is emphasized even more in this section. This requires more serious investigation.

Finally, understanding the systems that the attackers are may be targeting was the next step. This is summarized below.

Top 15 Attack Destinations

Occurrences	Target	External
37609	192.168.201.222	
25183	192.168.220.126	
9314	192.168.225.234	
5452	192.168.1.3	
5408	192.168.1.4	
5352	192.168.1.5	
5253	192.168.202.94	
5082	192.168.229.114	
4448	192.168.228.214	
3148	192.168.1.8	
2291	192.168.202.30	
2236	141.211.176.99	Y
2046	192.168.201.130	
1520	192.168.130.187	
1447	192.168.217.138	

Interestingly enough, one external target made the top 15. This is possible reason to suspect that either a user on the internal GIAC enterprises network is attempting to hack a remote site or that a source server was compromised.

7.2 Correlation with Prior GIAC Enterprise Reports

Two prior studies were reviewed in an effort to correlate the top attacking hosts in the past with the current submission. A review of Marc Bayerkohler's analysis shows that the top 4 targets hosts were not in the top 15 targets as determined in this submission. Furthermore there were no similarities in the top 4 attacking IP's of the previous scan with the top 20 attackers of this scan.

However, there was evidence that these hosts were still being attacked. Using the format created by Mr. BayerKohler correlation between his and Lenny Zeltser's we can correlate between 3 separate scans.

Top Alert Destination Hosts (per Lenny Zelter's Analysis)

Host	# Alerts in Mr. Zelter's Scan	# Alerts in Mr. Bayerkohler's Scan	Current # of Alerts	Updated Status
192.168.253.105	22118	47	8	Minor Target
192.168.217.2	4197	6	2	Minor Target
192.168.253.41	4176	4387	296	Target
192.168.100.230	3462	749	853	Target

192.168.253.105: In previous reports, this target host had received some ftp activity from Israel and also some NTP and DNS activity. The current activity has been reduced to basic reconnaissance probes in the form of a few port scans and TCP pings from several unique sources.

```
01/15-09:30:17.458403  | Null scan! | 216.51.102.134:0 | 192.168.253.105:0
11/28-20:24:14.587844  | SYN-FIN scan! | 139.130.61.206:109 | 192.168.253.105:109
01/01-23:36:46.274089  | Null scan! | 216.51.111.47:0 | 192.168.253.105:0
01/07-04:08:38.478434  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.253.105:53
01/07-15:35:54.590039  | Null scan! | 216.51.107.237:0 | 192.168.253.105:0
01/06-09:51:51.620346  | Null scan! | 216.51.104.65:0 | 192.168.253.105:0
01/04-06:44:46.140661  | NMAP TCP ping! | 212.114.5.2:1064 | 192.168.253.105:21
01/04-07:29:32.215787  | NMAP TCP ping! | 212.114.5.2:1112 | 192.168.253.105:21
```

192.168.217.2: Still relatively untargeted this particular target was only probed with 2 SYN-FIN scans by two servers.

```
12/13-03:13:42.225570  | SYN-FIN scan! | 200.194.102.99:21 | 192.168.217.2:21
01/07-04:05:32.824147  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.217.2:53
```

192.168.253.41: As reported by Mr. Bayerkohler in his analysis, the majority of the alerts are still related to the Watchlist 000220 alert due to traffic from China and Israel. For a full explanation of this exploit see his full report at http://www.sans.org/y2k/practical/Marc_Bayerkohler_GCIA.html

192.168.100.230: Continuous spp_portscan detects.

Host	# Alerts in Mr. Zelter's Scan	# Alerts in Mr. Bayerkohler's Scan	Current # of Alerts	Updated Status
202.38.128.188	22338	0	0	No Activity
192.168.253.12	18869	0	0	No Activity
204.60.176.2	13619	0	0	No Activity
159.226.45.3	5066	1558	0	No Activity
142.150.225.137	4594	0	0	No Activity

Unfortunately (or fortunately), the attacking systems listed in Mr. Zelter's analysis and somewhat still active in Mr. Bayerkohler's analysis are no longer active.

7.2 Splitrock.net attacks

Systems within this domain make up the majority of the top attacking systems and networks overall (see summary tables above). Further investigation of the logs reveals the following:

Network Range: 63.252.92

01/18-22:49:39.987380 | Null scan! | 63.252.92.108:65532 | 192.168.60.8:9216
01/12-08:56:39.257731 | Null scan! | 63.252.92.105:65531 | 192.168.60.11:6144
01/13-20:14:33.475218 | Null scan! | 63.252.92.66:0 | 192.168.60.8:0
12/28-13:53:51.506513 | Null scan! | 63.252.92.165:65533 | 192.168.60.16:256
12/28-13:53:51.869753 | Null scan! | 63.252.92.165:65533 | 192.168.60.16:1023
12/28-13:53:52.218536 | Null scan! | 63.252.92.165:65531 | 192.168.60.16:6144
12/28-13:53:52.235421 | Null scan! | 63.252.92.165:65532 | 192.168.60.16:8192
12/24-23:29:40.551781 | Null scan! | 63.252.92.73:65531 | 192.168.60.8:6144
12/24-23:29:40.991539 | Null scan! | 63.252.92.73:0 | 192.168.60.8:0
<….several other Null scans cut for brevity….>

Network Range: 63.253.140

12/30-11:21:36.692376 | Null scan! | 63.253.140.94:65532 | 192.168.60.16:8960
12/30-11:21:36.699079 | Null scan! | 63.253.140.94:65532 | 192.168.60.16:9984
01/09-16:02:31.266869 | Null scan! | 63.253.140.85:65531 | 192.168.60.11:6144
01/08-22:07:27.822459 | Null scan! | 63.253.140.104:65533 | 192.168.60.38:1023
<….several other Null scans cut for brevity….>

Network Range: 63.253.106

12/31-01:41:13.633561 | Null scan! | 63.253.106.1:0 | 192.168.60.38:0
12/31-09:21:35.013102 | Null scan! | 63.253.106.13:0 | 192.168.60.8:0
01/12-07:48:14.515185 | Null scan! | 63.253.106.12:65531 | 192.168.60.8:6144
01/01-00:48:04.129455 | Null scan! | 63.253.106.23:65531 | 192.168.60.16:6144
<….several other Null scans cut for brevity….>

Network Range: 209.255.180
01/03-18:42:01.463754 | Null scan! | 209.255.180.173:65531 | 192.168.60.38:6144
12/31-15:38:30.883746 | Null scan! | 209.255.180.247:0 | 192.168.60.11:0
<….several other Null scans cut for brevity….>

A quick analysis of the captured information for network ranges in the splitrock.net domain above indicate that there are definite similarities in the attacks. More specifically, all the attacks

are Null scans. Null scanning is a way to discover running services (and in some cases operating system type) by sending TCP packets with no flags set. Interestingly enough, several systems on several different networks at relatively the same times are all performing Null scans. Common sense suggests that this it is highly unlikely that a coordinated Null scan of this magnitude is being perpetrated. More likely, the nmap program is being used with the Decoy option (-D) set. This allows the attacker to specify a range of IP addresses to include as the source of the Null scans. This creates a kind of "smoke screen" by filling the victim's logs with packets from the real attacking system and the spoofed IP address of multiple innocent systems. This is most likely what is happening in this situation. The real attacker is probably a single system within the splitrock.net domain.

7.3 bezeqint.net attacks

The most interesting aspect of the top 20 attackers is the occurrence of systems from the **beqeqint.net** domain is represented a total of 8 times. Furthermore, the 10 attacking networks showed that the bezeqint.net domain initiates the third highest number of attacks. This, of course, warrants some further analysis to determine if the attack is targeted, what specific attackers are being attempted, etc. Searches on www.dshield.com reveal that the hosts in this address space have the following information:

```
inetnum:      212.179.27.96 - 212.179.27.127
netname:      INOBIZ
descr:        INOBIZ-LAN
country:      IL
admin-c:      NP469-RIPE
tech-c:       NP469-RIPE
status:       ASSIGNED PA
notify:       hostmaster@isdn.net.il
changed:      hostmaster@isdn.net.il 20000106
source:       RIPE

route:        212.179.0.0/17
descr:        ISDN Net Ltd.
origin:       AS8551
notify:       hostmaster@isdn.net.il
mnt-by:       AS8551-MNT
changed:      hostmaster@isdn.net.il 19990610
source:       RIPE

person:       Nati Pinko
address:      Bezeq International
address:      40 Hashacham St.
address:      Petach Tikvah  Israel
phone:        +972 3 9257761
e-mail:       hostmaster@isdn.net.il
nic-hdl:      NP469-RIPE
changed:      registrar@ns.il 19990902
source:       RIPE
```

Based on the SNORT alert and the information above, this belongs to a network in Israel. Furthermore, referencing the www.dshield.org database, this particular domain is responsible for quite a bit of activity as shown below.

Date	Source	Source Port	Target Port	Protocol
2001-03-13	212.179.161.96	4236	21	6
2001-03-13	212.179.161.96	4236	21	6
2001-03-03	212.179.142.134	0	31789	17
2001-03-03	212.179.142.134	31790	31789	17
2001-03-01	212.179.138.24	0	31789	0
2001-02-26	212.179.183.219	137	137	17
2001-02-26	212.179.183.219	137	137	17
2001-02-26	212.179.183.219	137	137	17
2001-02-26	212.179.183.219	137	137	17
2001-02-26	212.179.183.219	137	137	17
2001-02-26	212.179.183.219	137	137	17
2001-02-18	212.179.140.97	0	31789	17
2001-02-18	212.179.140.97	31790	31789	17
2001-02-14	212.179.178.201	0	0	1
2001-02-12	212.179.41.242	0	0	1
2001-02-12	212.179.224.92	0	0	1
2001-02-12	212.179.184.111	0	0	1
2001-02-12	212.179.41.242	0	0	1
2001-02-12	212.179.224.92	0	0	1
2001-02-12	212.179.184.111	0	0	1
2001-02-05	212.179.175.230	0	12345	6
2001-02-05	212.179.72.20	137	137	17
2000-10-02	212.179.175.70	3405	21	6
2000-10-02	212.179.175.70	3405	21	6
2000-03-27	212.179.225.241	8	0	1
2000-03-27	212.179.171.232	8	0	1
2000-03-27	212.179.172.215	8	0	1
2000-03-27	212.179.129.224	8	0	1
2000-03-25	212.179.224.28	8	0	1
2000-03-22	212.179.175.86	3438	21	6

2000-03-22	212.179.175.86	3438	21	6
2000-03-22	212.179.175.86	3438	21	6
2000-03-21	212.179.164.225	8	0	1

212.179.27.111 – Napster Activity: Analysis of the traffic initiated by the first server 212.179.27.111 reveals that the majority of the traffic appears to be on port 6688 which is Napster traffic. Its target, 192.168.201.222 appears to be a Napster server. To eliminate this traffic a firewall should be configured to disallow Napster traffic.

```
01/04-02:55:43.511596  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.27.111:1778 |
192.168.201.222:6688
01/04-02:55:43.529810  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.27.111:1778 |
192.168.201.222:6688
```

In another section of the log, this server attempts to connect to TCP port 138 which is associated with Microsoft file sharing.

```
11/26-10:41:26.926519  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.27.111:2019 |
192.168.217.138:41038
11/26-10:41:28.728592  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.27.111:2019 |
192.168.217.138:41038
```

212.179.95.5 – Oracle Connect Request?: The second server from this active domain makes several attempts to connect to port 1525 on 192.168.202.94 which is an Oracle port.

```
01/01-22:20:34.463598  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.95.5:4260 |
192.168.202.94:1525
01/01-22:20:34.476101  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.95.5:4260 |
192.168.202.94:1525
```

Further analysis shows that a lot of activity to 192.168.209.154 on port 4808. Searches of www.dshield.com, www.sans.org and www.google.com turned up no information on the identity of this port or any other sites reporting activity on it.

```
12/03-22:16:11.529082  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.95.5:4292 |
192.168.209.154:4808
12/03-22:16:13.703035  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.95.5:4292 |
192.168.209.154:4808
```

212.179.44.105 - Port 2209: Definitely a crafted packet since the source port is 1. The first openly malicious (but not suspicious) activity we've gotten from this address space.

```
12/28-18:54:16.674222  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.44.105:1 |
192.168.130.187:2209
12/28-18:54:16.827783  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.44.105:1 |
192.168.130.187:2209
```

212.179.42.102- SSL Connect Request: Activity to port 443 (SSL) on 192.168.5.29. No evidence of packet crafting here. Is SSL traffic valid on this server?

```
01/04-12:33:21.313166  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.42.102:2116 |
192.168.5.29:443
01/04-12:33:21.313212  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.42.102:2114 |
192.168.5.29:443
```

Sample traffic for the remaining servers on this network include:

```
Telnet Attempts
01/07-03:52:34.584598  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.58.12:49089 |
192.168.60.11:23
01/07-03:52:35.044774  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.58.12:49089 |
192.168.60.11:23

SubSeven 2.1 Activity
01/16-08:23:39.377628  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.45.241:2728 |
192.168.202.94:7000
01/16-08:27:55.868894  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.45.241:2728 |
192.168.202.94:7000

Unknown Port
12/06-17:51:18.315767  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.56.5:61309 |
192.168.219.62:4772
12/06-17:51:18.365288  | Watchlist 000220 IL-ISDNNET-990517 | 212.179.56.5:61309 |
192.168.219.62:4772
```

Final analysis of traffic from this domain is that there is, indeed, malicious activity. Most notably the crafted packets and the attempts to access a SubSeven Trojan. There doesn't appear to be much intelligence to the attempts since they come from multiple servers and are directed to multiple servers with no apparent methodology. Recommendations would include blocking all traffic from the 212.179.x.x network unless explicitly required by GIAC Enterprises. For thoroughness, the 192.168.202.94 server should be checked for the existence of the SubSeven Trojan.

7.4. AOL Attacks

14 unique systems within the AOL owned address space of 205.188.153 perpetrated attacks against systems in the GIAC network. The Snort logs reveal the following:

```
12/04-00:02:23.861286  | Attempted Sun RPC high port access | 205.188.153.106:4000 |
 192.168.98.226:32771
12/04-10:39:51.432938  | Attempted Sun RPC high port access | 205.188.153.100:4000 |
 192.168.213.158:32771
<…..several more Sun RPC high port attacks were perpetrated against multiple other systems in
the GIAC network from the .100 and .106 systems…..>

12/13-19:06:09.004267  | Attempted Sun RPC high port access | 205.188.153.107:4000 |
 192.168.225.234:32771
12/21-00:58:02.549503  | SUNRPC highport access! | 205.188.153.139:9898 | 192.168.21
3.158:32771
12/12-12:08:50.679224  | Attempted Sun RPC high port access | 205.188.153.99:4000 |
192.168.213.158:32771
01/15-13:42:26.403669  | Attempted Sun RPC high port access | 205.188.153.105:4000 |
 192.168.97.45:32771
01/15-22:00:57.728763  | Attempted Sun RPC high port access | 205.188.153.104:4000 |
 192.168.97.74:32771

< several more attempts at Sun RPC high port access was attempted by various systems in this
AOL address space >
```

What's most interesting about this traffic is that despite the source and destination IP addresses changing, the source and destination ports remain the same. A destination port of 32771 is oftentimes related to one of the rpc services available on Solaris. Several references to port 32771 are listed in the Snort signature database on www.whitehats.com. The fact that the several (but not all) of the source ports is 4000 highly suggests that one person or group is collaborating to launch these attacks or some standard program is using 4000 as a source port.

7.5 Other Attacking Servers in the Top 20

A few other systems outside those listed in the Top Attacking Networks, made attacks on systems inside the GIAC network. A few of these are briefly analyzed below in an attempt to locate more targeted attacks.

211.34.40.1 – SYN-FIN Scans: This attacker is scanning the entire 192.168.1.1 to 192.168.255.255 address range in search of an active DNS server. This may be a precursor to a DNS exploit of some kind. My recommendation would be to create a SNORT rule to alert us on any future activity from this source IP address.

```
01/07-04:04:47.785385  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.44:53
01/07-04:04:47.826141  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.46:53
01/07-04:04:47.864299  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.48:53
01/07-04:04:47.946500  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.53:53
01/07-04:04:47.964666  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.54:53
01/07-04:04:47.978231  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.55:53
```

```
01/07-04:04:48.015702  | SYN-FIN scan! | 211.34.40.1:53 | 192.168.208.56:53
```

209.67.50.203 – DNS Dos Exploit:

The attacker is randomly searching for systems on which to perform a DNS Denial-of-Service attack on. This does not appear to be targeted, and therefore, probably not much of a threat.

```
01/06-18:30:03.735366  | DNS udp DoS attack described on unisog | 209.67.50.203:7115 |
192.168.1.5:53
01/06-18:30:03.870078  | DNS udp DoS attack described on unisog | 209.67.50.203:16707 |
192.168.1.4:53
01/06-20:00:00.725528  | DNS udp DoS attack described on unisog | 209.67.50.203:1329
9 | 192.168.1.4:53
01/06-20:00:00.826654  | DNS udp DoS attack described on unisog | 209.67.50.203:1203
2 | 192.168.1.4:53
01/06-20:00:00.939018  | DNS udp DoS attack described on unisog | 209.67.50.203:2006
5 | 192.168.1.5:53
01/06-20:00:00.968516  | DNS udp DoS attack described on unisog | 209.67.50.203:2805
4 | 192.168.1.4:53
01/06-20:00:01.567114  | DNS udp DoS attack described on unisog | 209.67.50.203:2351
6 | 192.168.1.3:53
```

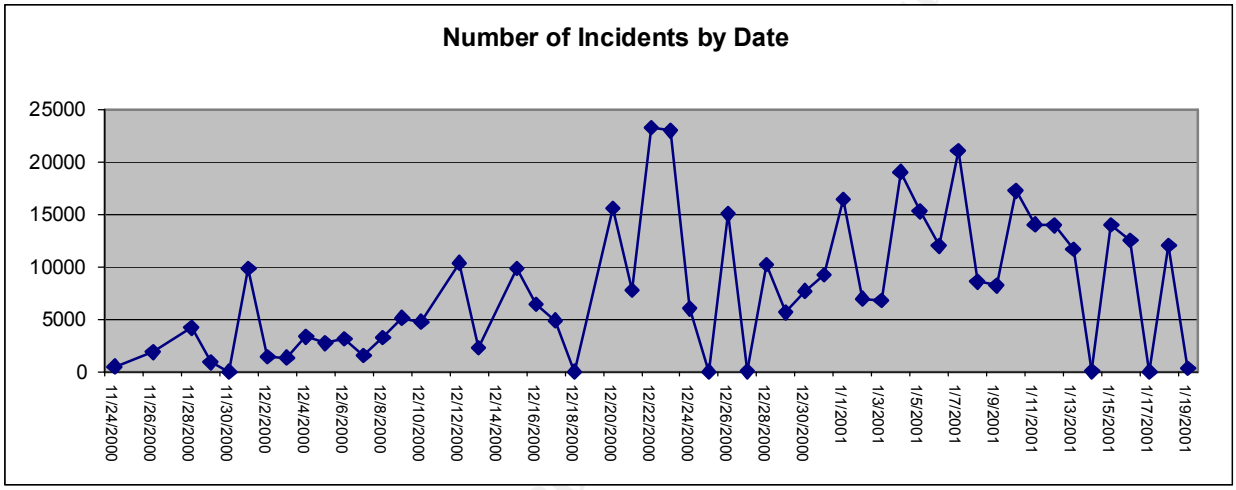
195.56.182.206, 194.234.48.26, 147.8.182.157, 194.204.224.131, 139.130.61.206, 200.194.102.99, – SYN-FIN Scans

All attackers are performing SYN-FIN Scans on a DNS port of multiple internal systems. These are also not targeted attacks of significant concern. These IP's should be noted their administrators contacted (if possible).

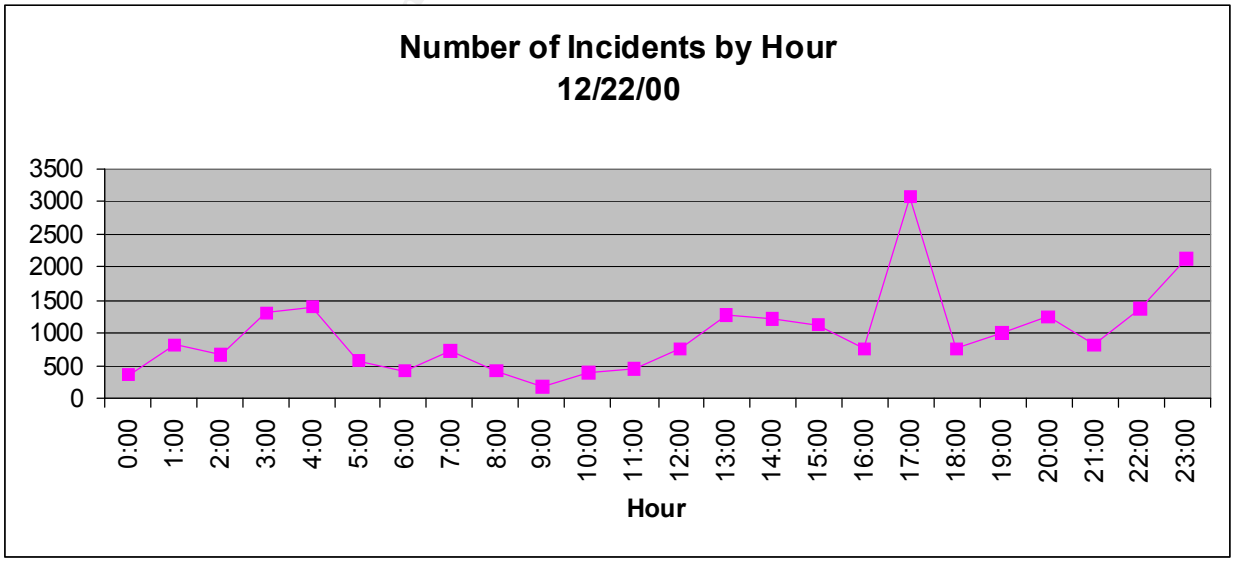
```
01/10-12:22:29.457805  | SYN-FIN scan! | 195.56.182.206:21 | 192.168.254.240:21
01/10-12:22:29.479675  | SYN-FIN scan! | 195.56.182.206:21 | 192.168.254.241:21
01/10-12:22:29.498409  | SYN-FIN scan! | 195.56.182.206:21 | 192.168.254.242:21
01/10-12:22:29.519740  | SYN-FIN scan! | 195.56.182.206:21 | 192.168.254.243:21
01/10-12:22:29.578100  | SYN-FIN scan! | 195.56.182.206:21 | 192.168.254.246:21
01/10-12:22:29.618261  | SYN-FIN scan! | 195.56.182.206:21 | 192.168.254.248:21
.
.<cut for brevity>
.
<several other systems, as listed above, initiated SYN-FIN attacks
```

7.6. Time Correlation

In the above approach we focused on increased activity or patters from specific IP address ranges in order to find meaningful data. A second approach is to look for increased activity based on time. To accomplish this, the data was formatted extensively (as described in Assignment 3 – The Process) and graphed to highlight areas of increased activity. First, a daily graph was created to quickly reveal areas of interest.

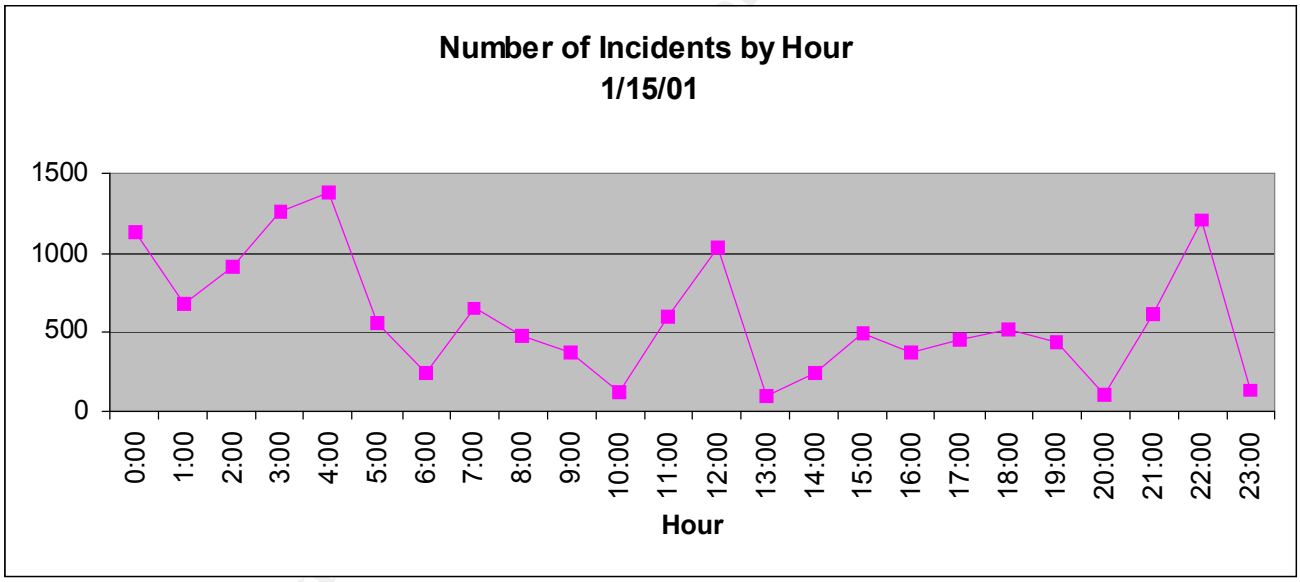


Based on significant increases in activity, December 22, 2000 and January 15, 2001 require further analysis. The next step is to create a similar graph based on each hour in the targeted days. These are shown below.



As shown above, on 12/22 a significant increase took place around 5:00p.m.. A closer look at this time window reveals that there were a significant number of portscans taking place along with Tiny Fragment alerts from 61.140.75.3, 61.140.75.4, 202.205.5.10 and others.

```
12/22-17:13:46.317899  | spp_portscan-portscan status from| 192.168.214.166|3 connec
tions across 3 hosts|TCP(0), UDP(3) |
12/22-17:13:50.139055  | spp_portscan-portscan status from| 192.168.214.166|2 connec
tions across 2 hosts|TCP(0), UDP(2) |
12/22-16:59:57.884435  | Tiny Fragments - Possible Hostile Activity | 61.140.75.3 |
192.168.1.8
12/22-17:14:34.813312  | Tiny Fragments - Possible Hostile Activity | 202.205.5.10 |
 192.168.1.8
12/22-17:14:34.815371  | Tiny Fragments - Possible Hostile Activity | 202.205.5.10 |
 192.168.1.8
```



On 1/15, increased activity took place around 4:00 a.m., 12:00 p.m. and 10:00 p.m.

This technique can be used by GIAC Enterprises to explore other suspicious segments of the log in a quick fashion.

7.7 Conclusion

According to the logs submitted by GIAC Enterprises, there are a significant number of attacks being performed on various systems on their internal network. Based on the analysis above, the majority of these attacks are:

- random scanning attempts to find running services in an attempt to exploit them.
- Originating from systems in 2 networks owned by splitrock.net and bezeqint.net

Several recommendations should be taken based on the analysis including:

- Contacting systems administrators for each of these networks and asking them to investigate.
- Configuration of firewalls to block activity from these common offenders if possible.
- Submission of logging information to intrusion correlation databases such as www.incidents.org/cid/ or www.dshield.com.

7.8. Assignment 3 – The Process

Note: While not specifically a part of this assignment, I think that it's extremely important for other GIAC students to understand the process that I used in the analysis. Furthermore, it demonstrates the analysis skills, attention to detail, and time that was committed to the assignment. And extra credit never hurt in an effort to attain an Honors Status. ☺

Snortsnarf to the rescue…...kind of

In an effort to analyze this massive amount of data quickly I began searching the Internet for tools that might have been helpful. Unfortunately, after searching popular Snort resource sites like www.snort.org and www.whitehats.com, I discovered that only SnortSnarf had the potential to provide assistance.

I downloaded snortsnarf 011601.1 from www.whitehats.com, to a Linux 6.2 server running on a Pentium 500 with 128 MB of RAM and 20GB hard drive. My first step was to concatenate all the SnortA*.txt files into one, large alert file to minimize the number of output files generated by snortsnarf. This was accomplished with the following command line:

 # cat SnortA*.txt >> all-alerts.txt

This successfully created one, large alert file of ~57 MB. As a test, I then ran snortsnarf on a couple of the small to medium sized SnortA*.txt files but it was taking an extremely long time to generate reports. After looking at the file (and perusing previous GIAC practical submissions) I determined that it was, most likely, getting hung up on the reference to MY.NET.X.X within the text file. In order to remove that as an issue, I executed the following command to create a new, single file to run snortsnarf on:

 # cat all-alerts.txt | sed 's/MY.NET/192.168/g' >> all-alerts-newaddr.txt

This sed command searches for all instances of MY.NET and replaces it with an IP address scheme from RFC 1597.

With a single, modified text file, I ran snortsnarf with no parameters. After about 2 hours of processing, I received an "Out of memory" error on my Linux server and the process died. With RAM being the issue I set out in search of a more capable server. The largest server I had available to me was a Sun 450 with a single 300 MHz processor, 512 MB RAM running Solaris 5.6. After downloading and running this process for approximately 6 hours, over 524 MB of HTML data was created. Much of this data us utilized above. In that time, however I began looking at other ways to glean the information just in case the Sun 450 didn't come through. After several hours invested in my custom scripts, I didn't look back at snortsnarf and used my scripts for 90% of the analysis. Thanks to the Teri Bidwell practical (Honors October 2000) for some good ideas on getting started!

A Custom Approach

Unfortunately, snortsnarf was not extremely helpful on this much information. It provided an excellent summary but was unable to give me the type of information I needed such the top attacking networks, the top attacking hosts or the top attacked systems. Furthermore, I was uneasy about the results I received from snortsnarf since it died on one system I was running. I realized that I would have to create my own scripts in order to pull out the data I needed to see.

The first step here was to determine which tool(s) to use. Not being adept at Perl the only tools that were available (and viable for that matter) were standard Unix tools such as sed, grep and Bourne shell programming. With this in mind, I began studying the snort alert files for a way to group the data. Unfortunately snort's alert format isn't extremely consistent. Some entries, such as standard alerts, maintained some type of consistent format such as:

11/29-00:40:34.565840 [**] Attempted Sun RPC high port access [**]
205.188.153.100:4000 -> 192.168.224.138:32771

This format can be described as:

Timestamp [**] Alert [**] SourceIP:Port -> DestIP:Port

Unfortunately, all the data in the SnortA*.txt files were not of this same standard format. For example, some lines looked like the following:

11/29-04:51:42.976561 [**] spp_portscan: PORTSCAN DETECTED from
24.212.9.78 (STEALTH) [**]

Different from the other format, this can be described as:

Timestamp [**] Alert Text from SourceIP [**]

The fact that the SourceIP was buried in the alert text is cause for concern when determining if there is any consistency to the log file formatting. A consistent format would be crucial to writing any customer scripts to help analyze the massive amount of data provided by the GIAC organization. With that I set out to place the shmega-alerts-newaddr.txt file into a common format.

File Format Processing

The most common format seemed to be that of the first example above. Knowing that I would be using tools such as awk to perform my searches, I knew that I needed a common format and a common delimiter. With that I created the following /bin/sh script shown below:

```
#!/bin/sh
#
# snortalf - snort alert formatter
```

```
#
# Charles L. Hutson
# 3/26/01
#
# Takes the file listed as the first command line argument and formats
# for better intrusion analysis.  A '|' is used to delimit fields for later
# manipulation using awk.
#
# The following modifications to the standard Snort Alert output files
# are made:
#
# sed #1: removes [**] and replaces with delimiter
# sed #2: removes  -> and replaces with delimiter
# sed #3: delimits the "from" field from the IP address field on
#         "spp_portscan status entries"
# sed #4: remove : followed by a space on "spp_portscan" entries and
#         add a delimiter
# sed #5: insert a delimiter befor the left parenthesis thats common on
#         "spp_portscan status entries"
# sed #6: take a mistakenly placed delimiter out on "spp_portscan" entries.

cat $1 | sed 's/\[\*\*\]/\|/g' | sed 's/->/\|/g' | sed 's/from/from\|/g' | sed 's/: /\|/g
' | sed 's/ (/\|(/g' | sed 's/spp_portscan\|/spp_portscan-/g' >> $1.alf
```

This one line sed program takes the file given as the first argument on the command line and creates a "|" delimited file with an .alf extension. With the file now in a more "common" format, we can apply awk commands to give us the information we need. The following command produces a file with the new format:

 # snortalf all-alerts-newaddr.txt

Once this step is complete, you have a file that you can now run the ever useful "awk" command on to pull out the fields that you find interesting.

Extracting Information

The first step in extracting useful information was to go to the main output of SnortSnarf. This showed me the varying types of attacks, their numbers, etc. Using this information, I called upon the **snorttb** script to give me relevant information about each attack. This script is included, with comments, below:

```
#!/bin/sh
#
# snorttb - this script was inspired by (and is based on) a script used
#         by Teri Bidwell (hence the name snort<TB) in honor of that work)
```

```
#          in a prior GCIA practical.
#
# Charles L. Hutson
# 3/22/01
#
# Argument 1 = unique text which specifies an alert (SYN-FIN for example)
# Argument 2 = .alf formatted source file (see snortalf script by Charles Hutson).
#
# The snorttb program was created to glean alert specific information from
# the formated SNORT files (.alf files).  Essentially it pulls information out and
# groups it by Source IP, Destination IP, etc.
#
# Example usage:
# snorttb SNY-FIN GIAC-info.alf
#
# The above command will search the GIAC-info.alf file and pull out information
# associated with the SYN-FIN attacks that were found.
#

mkdir $1
grep -i $1 $2 >> $1/$2.grep

cat $1/$2.grep | awk -F"|" '{print $3}' >> $1/$2.src.grep
cat $1/$2.grep | awk -F"|" '{print $4}' >> $1/$2.dst.grep
cat $1/$2.src.grep | awk -F":" '{print $1}' | sort|uniq -c| sort -r >> data/$1.srcip
.sorted
cat $1/$2.src.grep | awk -F":" '{print $2}' | sort|uniq -c| sort -r >> data/$1.srcpr
t.sorted
cat $1/$2.dst.grep | awk -F":" '{print $1}' | sort|uniq -c| sort -r >> data/$1.dstip
.sorted
cat $1/$2.dst.grep | awk -F":" '{print $2}' | sort|uniq -c| sort -r >> data/$1.dstpr
t.sorted
```

As stated in the comments, the **snorttb** script pulls out the useful information regarding each type of attack. The output is very much like the output of the **snortsnarf** command itself in that it shows the attackers using a specific attack, the number of attacks, their targets, etc.

While the information from the **snorttb** and the **snortsnarf** command was useful, it was almost too specific. With this much data, it's important to cut through the wave of data to find trends to begin exploring. For that, I devised a useful script called snortsome to extract even more useful information included in the analysis above. The most useful information that snortsome extracts is the number of unique attackers and their destinations while ignoring the specific attack types. This is extremely useful in determining where your real activity is.

```
# !/bin/sh
#snortsome - snort summarization tool.
# Charles L. Hutson
# 3/27/01
#
# Uses filename.alf files as a source (see 'snortalf' program developed by
# Charles Hutson) and generates various snort summaries.
#
# Input: 1st argument is the .alf formatted file to summarize
#

#
# Group by attack
#
cat $1 | grep -v spp | awk -F"|" '{print $2 ":" $3}' | awk -F":" '{print $1 ":" $2}'
 | sort | uniq -c | sort >> attack-src

#
# Group by Source IP
#
cat $1 | grep -v spp | awk -F"|" '{print $2 ":" $3}' | awk -F":" '{print $2 ":" $1}'
 | sort | uniq -c | sort >> src-attack

#
# All Sources Sorted
#
cat $1 | grep -v spp | awk -F"|" '{print $2 ":" $3}' | awk -F":" '{print $2}' | sort
 | uniq -c | sort -r >> attackers

#
# All Destinations Sorted
#
cat $1 | grep -v spp | awk -F"|" '{print $4 }' | awk -F":" '{print $1}' | sort | uni
q -c | sort -r >> targets
```

The last 2 commands created the following output which many of the most useful pieces of data were gathered. The data has been shortened for brevity.

```
# more attackers
9878  195.56.182.206
8565  194.234.48.26
48786  212.179.79.2
4563  212.179.95.5
4096  147.8.182.157
```

```
39015  212.179.27.111

# more targets
9314  192.168.225.234
5452  192.168.1.3
5408  192.168.1.4
5352  192.168.1.5
5253  192.168.202.94
5082  192.168.229.114
4448  192.168.228.214
```

A final, quick script was written to take the "attackers" file above an generate some fascinating reports output which really helped us hone in on the attacking networks. The following quick script groups all the attackers into their respective Class A and Class B networks to help us identify our real problem areas.

```
#!/bin/sh
#
# snorthacks - generates the top attacking networks from the output of "snortsome"
#
# Charles Hutson
# 3/22/01

cat attackers | awk -F" " '{print $2}' | sort -r | awk -F"." '{print $1 "." $2}' | u
niq -c | sort -r >> snorthacks.topnetworks2
cat attackers | awk -F" " '{print $2}' | sort -r | awk -F"." '{print $1 "." $2 "." $
3}' | uniq -c | sort -r >> snorthacks.topnetworks3
```

Finally, the information that was needed about each offending IP address was gleaned using basic grep, sed, and awk commands on the command line as needed.

The time correlation data was generated by manipulating the .alf files with other sed and awk commands in the script below.

```
#!/bin/sh
#
# Time Correlation
#
cat $1 | awk -F"|" '{print $1}' | sort >> tc.out
cat tc.out | sed 's/\./-/g' | awk -F'-' '{print $1 "/00:" $2}' >> tc.closer
cat tc.closer | awk -F':' '{print $1 " " $2}' | uniq -c >> tc.hourly.xls
cat tc.closer | awk -F':' '{print $1}' | uniq -c >> tc.monthly.xls
```

The hourly and monthly data was output into files and the pulled into Excel for graphing.

Appendix A: References

Web Sites

www.google.com
www.dshield.com
www.hack-a-tack.com
www.sans.org
www.incidents.org
www.geektools.com
www.whitehats.com
www.snort.org
www.securityfocus.com
http://www.robertgraham.com/pubs/firewall-seen.html - 10

Appendix B: Attacking Domain Information (Assignment 3)

Splitrock Services Inc.

```
Splitrock Services, Inc (NETBLK-SPLITROCK99)
   8665 New Trails Drive
   The, 77381
   US

   Netname: SPLITROCK99
   Netblock: 63.252.0.0 - 63.255.255.255
   Maintainer: SPLT

   Coordinator:
      Splitrock Services, Inc  (IS1-ARIN)  netadmin@SPLITROCK.NET
      281.465.1200
```

America Online

```
America Online, Inc (NETBLK-AOL-DTC)
   22080 Pacific Blvd
   Sterling, VA 20166
   US

   Netname: AOL-DTC
   Netblock: 205.188.0.0 - 205.188.255.255

   Coordinator:
      America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
   703-265-4670
```

Technische Universitaet Dresden

```
inetnum:      141.30.0.0 - 141.30.255.255
netname:      TUDR
descr:        Technische Universitaet Dresden
country:      DE
admin-c:      WW155
tech-c:       VR27-RIPE
status:       ASSIGNED PI
mnt-by:       DFN-NTFY
changed:      knocke@nic.de 19931220
changed:      poldi@dfn.de 20000719
source:  RIPE
```

The Computer Network Center Chinese Academy of Sciences

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

   Netname: NCFC
   Netblock: 159.226.0.0 - 159.226.255.255
```

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

Primus Telecommunications

```
inetnum:      203.134.0.0 - 203.134.31.255
netname:      INTERNETPRIMUS
descr:        Primus Telecommunications
descr:        Internet Services Network
country:      AU
admin-c:      IN1-AP
tech-c:       IN1-AP
mnt-by:       APNIC-HM
mnt-lower:    MAINT-PRIMUS-AU
changed:      dlambert@primustel.com.au 20000218
changed:      apnic-dbm@apnic.net 20000310
source:       APNIC

role:         IPRIMUS NET
address:      Level 2
address:      19 Pitt Street
address:      Circular Quay, Sydney
phone:        +61-2-9423-2400
fax-no:       +61-2-9423-2410
e-mail:       netops@iprimus.com.au
trouble:      PLEASE EMAIL abuse@iprimus.com.au FOR
trouble:      ALL NETWORK ABUSE MATTERS.
trouble:      Please include detailed information and times in UTC.
admin-c:      JD29-AP
tech-c:       JD29-AP
nic-hdl:      IN1-AP
remarks:      http://www.iprimus.com.au
notify:       netops@iprimus.com.au
mnt-by:       MAINT-PRIMUS-AU
changed:      netops@iprimus.com.au 20000321
source:       APNIC
```