IDIC Curriculum

Thomas Shepherd

Practical Assignment v2.9

Alert #1 – Synscan Alert

Section 1

My first network detect comes from an often overlooked, yet common, place where many attacks occur. It comes from my home PC connected to the Internet through a dial-up modem. At the time, I was using NetZero, a common free ISP, but this type of activity is by no means limited to that company. Dial-up accounts, and even more so DSL connections, have become a highly sought out target by attackers due to their relative lack of security.

Section 2

Since I am a fairly concerned IT professional I have a personal firewall on my home PC for the times that I connect to non-trusted networks, and most times trusted networks as well. The personal firewall that I have is Zone Labs' personal firewall ZoneAlarm v2.6.88. The software is set up with medium security on the local network and high security on the Internet network as defined by the software. According to the software this means that on the local network the software:

Enforces application privileges
Internet lock blocks all traffic
Allows local network access to Windows services and shares
Leaves your computer and servers visible to the local network

and on the Internet network:

Enforces application privileges
Internet lock blocks all traffic
Blocks Internet access to Windows services and file/printer shares
Stealth mode: firewall hides all ports not in use by a program

In this mode I set up the full IP address of my PC as the local network and everything else as the Internet network. While I get a lot of false positives of the DNS Portscan detect on my PC as a result of this configuration, having it set up this way allows me to have a more fine grained look at what is coming into and going out of my computer.

With this configuration I was able to obtain the following detect:

```
ZoneAlarm Logging Client v2.6.88
Windows 98-4.10.2222- A –SP
type,date,time,source,destination,transport
PE,2001/07/22,22:37:07 -6:00 GMT,ZCast,209.247.164.35:53,N/A
FWIN,2001/07/22,22:56:15 -6:00 GMT,209.247.164.28:0,MyPC:0,ICMP (type:3/subtype:3)
PE,2001/07/22,23:02:20 -6:00 GMT,Internet Explorer,127.0.0.1:1627,N/A
PE,2001/07/22,23:06:12 -6:00 GMT,Internet Explorer,127.0.0.1:1658,N/A
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

This shows the ZoneAlarm log file with the detect being the last line. While the log file includes definitions as to what each field represents, the third line, a more in depth look at the line of the alert is required to gain a full understanding of what is happening.

Section 3

It is a high probability that the source IP address is spoofed. The tool used to produce this alert is designed to spoof IP addresses. The tool is explained in more detail later in Section 6. This packet has both the SYN and FIN flags set. This usually indicates that the person wants a response to see whether or not a port, in this case 21, is operational so that a more coordinated attack can be launched. By sending the packet with both the SYN and FIN flags set, a response can be obtained without actually connecting to the computer. With this type of a scan, however, the attacker is probably listening close to the spoofed source address for the response.

It is also unlikely that the IP address was spoofed for a DOS attack since the log shows only the single attempt to gain entry. A DOS attack would send multiple attempts to connect in order to flood the spoofed IP address with responses. Since only a single response would have been sent it is unlikely that even if the address were spoofed the victim machine would have had any DOS degradation.

Section 4

The attack in this instance is a simple attack. The attacker sends the packet with the SYN and FIN flags turned on in order to elicit a response from the system as to the state of either the system or the service. The ports that are used for this type of attack are usually well-known ports such as 21-FTP, 23-Telnet, 25-SMTP, 53-DNS, 80-HTTP, 111-RPC, etc. If the destination port is not a well-known port, it could be a commonly used Trojan port such as 27374, used by Sub7, in order to see whether or not the system has been infected or hacked before. If the destination port is not a well-known port or a common hacked port, then the port is probably not significant, but the response from the system is. System responses are used to determine what type of system is being contacted in order to use exploits that are common to that system.

In this detect, the attack was directed at a well-known port, port 21-FTP. This in itself is not that significant since computer systems may make mistakes on which port is the destination port, or someone may have mistyped the destination address for a FTP server. What makes the destination port significant is that the source port is also 21-FTP. Normal FTP communication occurs between an ephemeral port on a client, which is usually a port above 1024, and the listening FTP port on the FTP server, port 21-FTP. FTP communication does not usually occur between port 21-FTP on both machines.

```
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

The first field in the detect shows what type of detect it was. ZoneAlarm has three different types of detects: Firewall In (FWIN), Firewall Out (FWOUT), and PE. A FWIN detect "indicates that the firewall blocked an incoming request to connect to your computer." A FWOUT detect "indicates that the firewall blocked an outbound request from your computer." And a PE detect "indicates that an application on your computer attempted to access the

Internet." This FWIN detect shows that a source external to my local network, which is my PC, is attempting to access a resource on my computer.

```
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

The second field in the detect shows the date that the detect was logged. The format for this field is: {Four digit year}/{two digit month}/{two digit day}. This means that this detect was taken on July 22, 2001. This was a Sunday evening when I was just browsing the Internet for interesting information.

```
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

The third field in the detect shows the local time that the detect was logged along with the offset from Greenwich Mean Time. I had to take a second look at this field when I was analyzing the log. Although the time of the detect is right, the offset is said to be –6 GMT which is CST and I am in the Mountain time zone which is –7 GMT. At first I though that this might be an error in the program or a mislead in the log file, but then I realized that while Mountain Standard Time (MST) is –7 GMT, the detect was taken during Mountain Daylight Time (MDT) which is technically –6 GMT, which means that the time in the log is correct.

```
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

The fourth field in the detect shows the source IP address and port. The IP address has been changed in order to not unduly incriminate anyone. The IP address is clearly not my local network - my PC, and by checking the IPCONFIG program, it was not listed as one of my allocated DNS servers. This means that it is not a commonly trusted source for this arrangement. Also, the source port is a well-known port, 21, used to establish FTP communications.

```
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

The fifth field in the detect shows the destination IP address and port. The IP address has been changed in order to avoid future targeting, and in this instance because the IP address is dynamically assigned. The destination port is a well-known port, 21, used for FTP communications.

```
FWIN,2001/07/22,23:41:07 -6:00 GMT,192.252.89.212:21,MyPC:21,TCP (flags:SF)
```

The last field in the detect shows the protocol that generated this particular alert and any other information that may be pertinent. The other information shows that the packet that was received, which was a TCP packet, has both the SYN and FIN flags set.

Section 5

By obtaining a response from the system about the 21-FTP port, the attacker could determine whether or not the system is set up to receive FTP communication. If the system is capable of FTP communication, then the attacker could try to utilize a weakness in the system to transfer files from my computer to the attacker's computer to gain information, and possibly use that information to gain the attacker greater access. The other possibility is that the attacker could exploit a weakness to transfer files from the attacker's computer to my computer to plant a Trojan program that would then give the attacker access to my system.

There is another possibility that must be explored, and that is that this packet was somehow corrupted as it went through the network and ended up falsely triggering the alert on my system. The correlation to the alert should make sure that the alert isn't just a fluke.

Section 6

As I am also a conscientious IT professional I do not rely solely on Zone Alarm to be my only means of tracking my network activity. For logging and intrusion detection I also run the win32 version of Snort v1.7. When this attack came in I also examined my Snort log files and found that Snort made the same determination as ZoneAlarm, that it was a port scan.

```
[**] spp_portscan: PORTSCAN DETECTED from 192.252.89.212 (STEALTH) [**]
07/22-23:41:06.220000

[**] SCAN synscan portscan [**]
07/23-00:41:06.244771 20:53:52:43:0:0 -> 44:45:53:54:0:0 type:0x800 len:0x3C
192.252.89.212:21 -> MyPC:21 TCP TTL:35 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x53C270BF  Ack: 0x34AB3610  Win: 0x404  TcpLen: 20

[**] spp_portscan: portscan status from 192.252.89.212: 1 connections across 1 hosts: TCP(1), UDP(0)STEALTH [**]
07/22-23:42:41.410000

[**] spp_portscan: End of portscan from 192.252.89.212: TOTAL time(0s) hosts(1) TCP(1) UDP(0) STEALTH [**]
07/22-23:42:49.210000
```

The snort log shows the correlation that the alert that ZoneAlarm made is valid. If this had been a corrupted packet then either of the programs would probably have reported it as a malformed packet. The fact that both programs report the same attack shows that the packet is not corrupt.

The Snort log file shows a little more information which helps to formulate how this alert was generated. The log shows that the scan is from a program called synscan. The program is a port scanner that scans for socks, proxy, cgi-bin, and rfc vulnerabilities. Most recently it was used as part of the Ramen and Lion worms. More information about synscans can be obtained through most any security site.

In looking at the attack signature of the synscan tool, it is noted that the packet ID produced by the synscan tool is always 39426, which the log shows that it is. It is also noted that the TTL value produced by the synscan tool is 42, meaning that the attacker in this case is 7 hops away. Since there were no other arriving packets it is assumed that the attacker did not receive a positive response from my computer in order to initiate a more coordinated attack.

Section 7

What can be determined from the examination of this event is that this was most likely a one-time event for my computer and was not a directed attack. Most likely the attacker was making a port scan of a subnet on which I just happened to be. Since the alert was only triggered the one time there is no evidence of a directed attack against my computer. And since only the one port was scanned there is no evidence that it was a means of gathering directed reconnaissance about my system. Rather the evidence shows that the attacker was probably fishing for potential open FTP targets that could be revisited later with a more concerted effort.

Section 8

The severity of this attack is a (-2). I arrive at that conclusion through the severity of attack formula in the IDIC course using the following values. Since this was my home computer, I am going to put the criticality level at a 5. If this were a single computer on the network at the office I might put it at a 2 or a 3, but at home this rates a 5. I rated the lethality of this attack as a 3. While it is possible that the attacker may persist and eventually end up with root access, the immediate danger is somewhat less than root access. The counter-measures included a personal firewall and an intrusion detection sensor. All of the security patches were up to date and there wasn't any software running that could provide a back-door into the system. Therefore the rating for the system and network countermeasures both rate 5s. So, the formula expands to (5+3)-(5+5) which results in a score of (-2).

Section 9

According to the ZoneAlarm program, a positive response was not sent to the attacker. The packet was logged and then dropped so no further information was sent into the system to be processed. Given that the attacker did not make any more intrusion attempts, I would say that the security of the system was adequate enough to deal with the situation. I do not think that any modifications to the defenses need to be made at this time.

Section 10

Multiple Choice Question:
What causes you to be the most suspicious about the following alert produced by the synscan tool?

```
[**] spp_portscan: PORTSCAN DETECTED from 192.252.89.212 (STEALTH) [**]
07/22-23:41:06.220000

[**] SCAN synscan portscan [**]
07/23-00:41:06.244771 20:53:52:43:0:0 -> 44:45:53:54:0:0 type:0x800 len:0x3C
192.252.89.212:21 -> MyPC:21 TCP TTL:35 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x53C270BF  Ack: 0x34AB3610  Win: 0x404  TcpLen: 20

[**] spp_portscan: portscan status from 192.252.89.212: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH [**]
07/22-23:42:41.410000

[**] spp_portscan: End of portscan from 192.252.89.212: TOTAL time(0s) hosts(1) TCP(1) UDP(0) STEALTH [**]
07/22-23:42:49.210000
```

a) This is a stealth portscan.
b) Port 21 is used for both source and destination ports.
c) Both the SYN and FIN flags are set.
d) The datagram length is equal to the IpLen plus the TcpLen.

Answer: b. Other than the fact that the rule that generated this alert says that it is a synscan alert, the first clue that this is a problem packet is that both the source and destination ports are 21, which should never happen during normal network traffic.

Alert #2 – SYN Scan of Network for an Active Sub7 Trojan

Section 1

This detect is from the network where I work. The network is a small network consisting of a Class C subnet. The network uses DHCP for all workstations and assigned addresses for all other network attached devices. The network runs Novell's Netware, Windows NT/2K, and Sun's Solaris for network operating systems and Windows 98/NT/2K on the workstations. The network is connected to a trusted WAN system that uses routing rules for internal security and is protected by firewalls from external access, and another trusted network that has their own security in place.

Section 2

The Intrusion Detection System used to obtain this detect is Snort version 1.8 running on a Unix server. The Intrusion Detection System is placed on a hub at the front of our network so that it can monitor the incoming and outgoing traffic to our network without slowing down the network or causing packet loss on the network. This alert is a SYN scan logged by Snort's portscan preprocessor. Since the traffic does not have a corresponding rule, SYN packets are normal traffic, an alert is logged due to a large number of connections that are made over a short time.

Section 3

In a SYN scan of this type, the attacker wants to know which machines respond and which do not. Therefore, it is unlikely that the source address of this attack is spoofed. In this particular scan the attacker is trying to connect to port 27374. Although this is not a well-known port for common services, it has been used many times by a program called Sub7. It is highly likely that this attacker is trying to determine whether or not there is an instance of Sub7 on the network that could be used to gain further access to the network.

Section 4

This detect began at 10:55pm on October 15th; however, I was unaware of it until October 16th when I reviewed the security logs from the previous day. Usually the portscan log in the Snort log directory is empty, but this day it had data in it. By looking at the log I could tell almost immediately that it is an attempt at an intrusion, but I need to find out what the intruder is trying to do and did they gain access to the network.

```
Oct 15 22:55:56 65.7.196.121:3972 -> 10.10.10.5:27374 SYN ******S*
Oct 15 22:55:56 65.7.196.121:4015 -> 10.10.10.48:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4012 -> 10.10.10.45:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4014 -> 10.10.10.47:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4017 -> 10.10.10.50:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4018 -> 10.10.10.51:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4027 -> 10.10.10.60:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4078 -> 10.10.10.111:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4082 -> 10.10.10.115:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4083 -> 10.10.10.116:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4084 -> 10.10.10.117:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4087 -> 10.10.10.120:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4028 -> 10.10.10.61:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4066 -> 10.10.10.99:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4067 -> 10.10.10.100:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4069 -> 10.10.10.102:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4070 -> 10.10.10.103:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4075 -> 10.10.10.108:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4080 -> 10.10.10.113:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4086 -> 10.10.10.119:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4088 -> 10.10.10.121:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4090 -> 10.10.10.123:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4091 -> 10.10.10.124:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4092 -> 10.10.10.125:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4093 -> 10.10.10.126:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4094 -> 10.10.10.127:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4095 -> 10.10.10.128:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4099 -> 10.10.10.132:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4101 -> 10.10.10.134:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4103 -> 10.10.10.136:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4105 -> 10.10.10.138:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4106 -> 10.10.10.139:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4108 -> 10.10.10.141:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4109 -> 10.10.10.142:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4110 -> 10.10.10.143:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4111 -> 10.10.10.144:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4112 -> 10.10.10.145:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4116 -> 10.10.10.149:27374 SYN ******S*
```

From the date field I am able to determine that the detect occurred at 10:55pm and lasted only a couple of seconds. While it is possible that a computer may make this many connections over a few seconds, following fields indicate that this is probably not a normal occurrence.

```
Oct 15 22:55:56 65.7.196.121:3972 -> 10.10.10.5:27374 SYN ******S*
Oct 15 22:55:56 65.7.196.121:4015 -> 10.10.10.48:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4012 -> 10.10.10.45:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4014 -> 10.10.10.47:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4017 -> 10.10.10.50:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4018 -> 10.10.10.51:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4027 -> 10.10.10.60:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4078 -> 10.10.10.111:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4082 -> 10.10.10.115:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4083 -> 10.10.10.116:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4084 -> 10.10.10.117:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4087 -> 10.10.10.120:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4028 -> 10.10.10.61:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4066 -> 10.10.10.99:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4067 -> 10.10.10.100:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4069 -> 10.10.10.102:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4070 -> 10.10.10.103:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4075 -> 10.10.10.108:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4080 -> 10.10.10.113:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4086 -> 10.10.10.119:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4088 -> 10.10.10.121:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4090 -> 10.10.10.123:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4091 -> 10.10.10.124:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4092 -> 10.10.10.125:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4093 -> 10.10.10.126:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4094 -> 10.10.10.127:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4095 -> 10.10.10.128:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4099 -> 10.10.10.132:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4101 -> 10.10.10.134:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4103 -> 10.10.10.136:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4105 -> 10.10.10.138:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4106 -> 10.10.10.139:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4108 -> 10.10.10.141:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4109 -> 10.10.10.142:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4110 -> 10.10.10.143:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4111 -> 10.10.10.144:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4112 -> 10.10.10.145:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4116 -> 10.10.10.149:27374 SYN ******S*
```

As the correlations will show, the tool used to perform this scan is most likely automated so it is likely that the source address and ports are spoofed. Looking at all of the source IP addresses shows that the connections came from a single computer rather than multiple sources.

```
Oct 15 22:55:56 65.7.196.121:3972 -> 10.10.10.5:27374 SYN ******S*
Oct 15 22:55:56 65.7.196.121:4015 -> 10.10.10.48:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4012 -> 10.10.10.45:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4014 -> 10.10.10.47:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4017 -> 10.10.10.50:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4018 -> 10.10.10.51:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4027 -> 10.10.10.60:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4078 -> 10.10.10.111:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4082 -> 10.10.10.115:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4083 -> 10.10.10.116:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4084 -> 10.10.10.117:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4087 -> 10.10.10.120:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4028 -> 10.10.10.61:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4066 -> 10.10.10.99:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4067 -> 10.10.10.100:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4069 -> 10.10.10.102:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4070 -> 10.10.10.103:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4075 -> 10.10.10.108:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4080 -> 10.10.10.113:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4086 -> 10.10.10.119:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4088 -> 10.10.10.121:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4090 -> 10.10.10.123:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4091 -> 10.10.10.124:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4092 -> 10.10.10.125:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4093 -> 10.10.10.126:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4094 -> 10.10.10.127:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4095 -> 10.10.10.128:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4099 -> 10.10.10.132:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4101 -> 10.10.10.134:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4103 -> 10.10.10.136:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4105 -> 10.10.10.138:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4106 -> 10.10.10.139:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4108 -> 10.10.10.141:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4109 -> 10.10.10.142:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4110 -> 10.10.10.143:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4111 -> 10.10.10.144:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4112 -> 10.10.10.145:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4116 -> 10.10.10.149:27374 SYN ******S*
```

The destination IP address and port are what makes this a clear detect of an attempted intrusion. There are multiple machines, but all of the destination ports are identical. The odds of this many machines on the same network all wanting to connect to the same computer across the same port is nearly impossible.

```
Oct 15 22:55:56 65.7.196.121:3972 -> 10.10.10.5:27374 SYN ******S*
Oct 15 22:55:56 65.7.196.121:4015 -> 10.10.10.48:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4012 -> 10.10.10.45:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4014 -> 10.10.10.47:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4017 -> 10.10.10.50:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4018 -> 10.10.10.51:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4027 -> 10.10.10.60:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4078 -> 10.10.10.111:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4082 -> 10.10.10.115:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4083 -> 10.10.10.116:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4084 -> 10.10.10.117:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4087 -> 10.10.10.120:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4028 -> 10.10.10.61:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4066 -> 10.10.10.99:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4067 -> 10.10.10.100:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4069 -> 10.10.10.102:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4070 -> 10.10.10.103:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4075 -> 10.10.10.108:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4080 -> 10.10.10.113:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4086 -> 10.10.10.119:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4088 -> 10.10.10.121:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4090 -> 10.10.10.123:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4091 -> 10.10.10.124:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4092 -> 10.10.10.125:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4093 -> 10.10.10.126:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4094 -> 10.10.10.127:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4095 -> 10.10.10.128:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4099 -> 10.10.10.132:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4101 -> 10.10.10.134:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4103 -> 10.10.10.136:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4105 -> 10.10.10.138:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4106 -> 10.10.10.139:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4108 -> 10.10.10.141:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4109 -> 10.10.10.142:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4110 -> 10.10.10.143:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4111 -> 10.10.10.144:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4112 -> 10.10.10.145:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4116 -> 10.10.10.149:27374 SYN ******S*
```

The last field of the detect shows that the only flag that is set is the SYN flag. A SYN/ACK response was not logged from any of the machines. Since no other traffic between this source and the network was logged, it would be safe to say that a connection was not made to any of the machines that were probed.

Section 5

In a SYN scan, the attacker sends a SYN packet to a particular port on a particular machine. This is the initiation of the three-way handshake that connects two machines together in a session. The attacker may or may not be interested in a connection to that particular service. If the attacker is interested in connecting to a service, such as a backdoor or Trojan program, the three-way handshake may be completed. An attacker may be simply looking for a service to exploit, in which case the three-way handshake is not completed and the computers do not connect. However, if the attacker receives a response from a machine then the attacker knows that the service is available and may try to exploit that service.

SYN scans come in a couple of varieties, a general port scan or a specific port scan and a general IP scan or a directed scan. SYN scans are generally used as reconnaissance for a more directed attack or intrusion. It is easy to alert on a clumsy attacker who pounds on a network with SYN packets. It is inversely as difficult to alert on a sophisticated and patient attacker since SYN traffic is a normal part of the network environment.

In a general port scan an attacker may try several different ports to find out if a service is available. Some of the more clumsy attackers will step through ports one at a time rather than target known service ports. The more sophisticated attacker will attempt a scan of well-known service ports before moving on to unknown ports. For the most part this type of scan is a reconnaissance scan designed to map out what services are available on a particular computer or network. A clumsy attacker will also scan many ports in a short amount of time, which usually triggers an IDS alert. The more sophisticated attacker will scan more slowly trying to avoid a multiple connection sensor.

If the scan is used as a directed scan, the attacker will pick a single IP address and try all of the ports on that one address. This is useful for determining what services a particular machine has available. This may also be used in fingerprinting which OS is installed on the machine. If the scan is used in a general IP scan, the attacker will pick a particular subnet or IP address range in order to determine what services are available on a particular network. The attacker may also be able to determine some information about the structure of the network through this process.

In a specific port scan an attacker tries to connect to a specific service rather than multiple services. This is usually done once the initial reconnaissance has been performed in order to determine whether or not the service is active. If the service is active, then the attacker can use other methods to exploit any vulnerabilities of the service.

A specific port scan is usually used in conjunction with a general IP scan by an attacker that has not collected previous information about the network. The attacker usually has a particular exploit in mind and is looking for a place to use it. A clumsy attacker will once again try to connect to many different machines in a short amount of time trying to find all of the places that may be attacked. The more sophisticated attacker, however, will take longer to scan, knowing that most networks do not change all that quickly.

An attacker that has gathered information about a network previously will usually use a specific port scan in conjunction with only those IP addresses that will probably give the attacker a response. As in a general IP scan, the attacker is looking for a particular service, but with a little more reconnaissance the attacker may only scan those machines that are likely to have the service available. A clumsy attacker may not be detected on this type of scan depending on how many machines are being scanned. If it is one or two, the attacker may slip through depending upon the threshold of the IDS sensor. A sophisticated attacker will usually try to determine what that IDS threshold is and then try to scan at a slower pace than it detects.

Section 6

This particular detect has not been seen before on this network, but SYN scans are a popular method of performing reconnaissance on a network. A great deal of information may be obtained by performing a SYN scan. Previously SYN scans were used in a particular attack in which the attacker would send a certain number of SYN packets to a machine in order to try to guess at the sequence number of the packets. If the attacker could guess the sequence number then the attacker could hijack a session between two computers and gather all the data of the traffic that flowed between them. These types of attacks are fairly old now, but they are still common since most machines are still vulnerable to them.

Beyond session stealing, SYN scans are also useful to an attacker by performing what is called OS fingerprinting. An attacker sends out certain packets, some ICMP, some UDP, and some TCP. The responses that the attacker receives back will usually allow the attacker to fingerprint which operating system is being used. This will give the attacker a little advantage in knowing which exploits may or may not work.

There have been many other reports of this type of activity on other networks, which has been discussed recently on the Incidents.org mailing list (www.incidents.org). According to one email this is most likely attributed to an automated type of attack. The attack is carried out by a program called a 'Leaves Bot.' The program is designed to scan networks looking for a particular service, in this case a system that has the Sub7 trojan program on it. The program can then either report back to the attacker on which targets are viable; or if it is sophisticated enough, it will use other means to attempt an intrusion into the system.

This program is most often found on home computers. The IP address in this alert resolves back to a popular home computer account network, @Home. This type of attack has been increasing over the last few months according to Incidents.org. Many of the monthly journals since August have included some information about this particular type of SYN scan. A more detailed description of the workings of a 'Leaves Bot' can be found on the Incidents.org website at: http://www.incidents.org/react/w32leavesworm.php.

Section 7

This particular detect is a specific port scan but a non-directed IP attack. This can be determined by the fact that not all of the IP addresses of this class C subnet are enumerated and the destination port for each machine does not change. However, a look at the source ports would seem to indicate that each destination IP has a corresponding incremental source port. It could be concluded then that the attacker did scan the entire subnet but only those computers that were active were logged.

```
Oct 15 22:55:56 65.7.196.121:3972 -> 10.10.10.5:27374 SYN ******S*
Oct 15 22:55:56 65.7.196.121:4015 -> 10.10.10.48:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4012 -> 10.10.10.45:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4014 -> 10.10.10.47:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4017 -> 10.10.10.50:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4018 -> 10.10.10.51:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4027 -> 10.10.10.60:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4078 -> 10.10.10.111:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4082 -> 10.10.10.115:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4083 -> 10.10.10.116:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4084 -> 10.10.10.117:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4087 -> 10.10.10.120:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4028 -> 10.10.10.61:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4066 -> 10.10.10.99:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4067 -> 10.10.10.100:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4069 -> 10.10.10.102:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4070 -> 10.10.10.103:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4075 -> 10.10.10.108:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4080 -> 10.10.10.113:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4086 -> 10.10.10.119:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4088 -> 10.10.10.121:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4090 -> 10.10.10.123:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4091 -> 10.10.10.124:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4092 -> 10.10.10.125:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4093 -> 10.10.10.126:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4094 -> 10.10.10.127:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4095 -> 10.10.10.128:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4099 -> 10.10.10.132:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4101 -> 10.10.10.134:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4103 -> 10.10.10.136:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4105 -> 10.10.10.138:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4106 -> 10.10.10.139:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4108 -> 10.10.10.141:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4109 -> 10.10.10.142:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4110 -> 10.10.10.143:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4111 -> 10.10.10.144:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4112 -> 10.10.10.145:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4116 -> 10.10.10.149:27374 SYN ******S*
```

Section 8

The severity of this attack is a 4. I arrive at that conclusion through the severity of attack formula in the IDIC course using the following values. This attack was conducted across the network on many machines including workstation, servers, and some intelligent network attached equipment. This makes the criticality portion rate a 5. Since this attacker was looking for the Sub7 program, which compromises root on the network, the lethality of this attack is a 5. Almost all of the machines attacked had current software and recent security patches in place; however, back door Trojans such as Sub7 may get in anyway, so the system countermeasures rates a 4. The weak point of this system is on the network countermeasures. The local network is only protected by the corporate firewall, which wasn't tuned enough to block this attack. There isn't a local firewall, which drops the rating even more. The only saving grace is that there is an IDS which will at least alert on the attack. For this reason I rated the network countermeasures as a 2. So, the formula expands to (5+5)-(4+2) which results in a score of 4.

Section 9

The network was lucky this time; it may not be so lucky next time. Some networks are required to rely on corporate security and hope that it is enough to prevent incidents on the local network. This is not the case on this network. While it is good that the software is kept up to date and that security is beginning to be put into place, more needs to be done. First and foremost a local network firewall must be put into place; if nothing else, perhaps a proxy server. Without the firewall the network will still be as vulnerable as the corporate firewall allows. It will also be open to attacks from other networks within the corporate firewall but across the WAN.

Having the Intrusion Detection System is a good thing. It doesn't tell whether or not there are unauthorized or Trojan programs on the network, though. Since this is my network, it makes it a little easier to say that before this happened there weren't any procedures in place to check for these programs on a regular basis. Therefore, procedures need to be put into place to check for unauthorized programs and Trojan programs on a regular basis.

Lastly, it is my recommendation that the incident handling procedures need to be more formalized and implemented better. While it may not come out in this paper, there weren't any formal procedures in place on how to handle an incident such as this. Once the incident was reported, it was up to the office director on what he wanted to do about the situation, but since he had not dealt with something such as this, he turned back to his network administrator for the resolution. Having the procedures in place would have made the whole process easier so that there would be little or no question on what should happen.

Section 10

What was this attacker trying to do on this network?

```
Oct 15 22:55:56 65.7.196.121:3972 -> 10.10.10.5:27374 SYN ******S*
Oct 15 22:55:56 65.7.196.121:4015 -> 10.10.10.48:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4012 -> 10.10.10.45:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4014 -> 10.10.10.47:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4017 -> 10.10.10.50:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4018 -> 10.10.10.51:27374 SYN ******S*
Oct 15 22:55:57 65.7.196.121:4027 -> 10.10.10.60:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4078 -> 10.10.10.111:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4082 -> 10.10.10.115:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4083 -> 10.10.10.116:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4084 -> 10.10.10.117:27374 SYN ******S*
Oct 15 22:55:58 65.7.196.121:4087 -> 10.10.10.120:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4028 -> 10.10.10.61:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4066 -> 10.10.10.99:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4067 -> 10.10.10.100:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4069 -> 10.10.10.102:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4070 -> 10.10.10.103:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4075 -> 10.10.10.108:27374 SYN ******S*
Oct 15 22:55:59 65.7.196.121:4080 -> 10.10.10.113:27374 SYN ******S*
```

a) determine which machines are active on the network
b) connect to machines running Microsoft's Messenger program
c) guess the session ID's of the computers on the network
d) connect to an active Sub7 trojan program on the network

Answer: d. While the obvious answer is that it is a SYN scan of the network, the fact that all of the destination ports are 27374 should indicate that the attacker was looking for an active Sub7 program.

Alert #3 – SnowWhite Mailvirus Alert

Section 1

This detect is from the network where I work. The network is a small network consisting of a Class C subnet. The network uses DHCP for all workstations and assigned addresses for all other network attached devices. The network runs Novell's Netware, Windows NT/2K, and Sun's Solaris for network operating systems and Windows 98/NT/2K on the workstations. The network is connected to a trusted WAN system that uses routing rules for internal security and is protected by firewalls from external access, and another trusted network that has their own security in place.

Section 2

The Intrusion Detection System used to obtain this detect is Snort version 1.7 running on a Unix server. The Intrusion Detection System is placed on a hub at the front of our network so that it can monitor the incoming and outgoing traffic to our network without slowing down the network or causing packet loss on the network.

At the time of this detect our office was receiving a high number of emails with the Snow White virus attached. The director asked the IT staff to try to stop the influx. As part of this the IT staff was monitoring for virus signatures through various means. One of these means was by adding a rule to Snort's rulset tuned for our network: alert tcp 192.70.15.10 25 -> 10.10.10.45 25 (msg:"MAILVIRUS - SnowWhite Trojan Outgoing"; content:"Suddlently";). The rule is basically used for tracking and not for prevention.

Section 3

The source address for this alert is not spoofed. This is known because the alert is a known event for which we were looking and the two computers involved in this transaction are within the scope of our network. If there had been multiple instances of this alert within a short time frame I would be more inclined to look at a possible DOS attack, such as a mailbomb. However, that is not the instance in this case. Only the single alert was logged and the source is a known source.

Section 4

This particular attack is not an attack directed against a system in the same manner that most intrusions are carried out. Rather this is evidence of an attack carried out through the use of virus code and is not an effective attack unless the code is executed. This particular virus is the Snow White virus that, if executed, will attempt to infect the WSOCK32.DLL file. Once this file has been infected, the virus will scan for email addresses passing through it so that the virus can spread to other systems through that email address. Variants will also try to download plug-ins that can cause further harm to the computer, including loading Trojans such as Sub7.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
```

```
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The first section of the alert shows the predetermined name for the type of alert that Snort is reporting. In this case the alert is a detect of a virus signature named SnowWhite. This virus/Trojan is particularly hard to deal with since the email does not contain return information and the trace-back information only gives limited information. In most cases the information shows up as a dial-up account to AOL, Juno, NetZero, etc. If a trace yields enough information we notify the person who sent the file of the infection of their system.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The second section of the alert shows the date and time that the alert was generated. In this incident the alert was generated on June 29 at 7:15am. The time-index shows down to the millisecond of when the alert was generated. This level of detail is helpful if you are working with multiple alerts, but in this case there is only the single alert and therefore the time is not crucial other than using it in correlation to other logs.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The third section of the alert shows the Ethernet information of the alert. This section of the alert has no real bearing on the detect at this point other than providing some information about the packet itself.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The fourth section of the alert shows the source and destination IP addresses and ports. The source IP address is our mail server that we have published to the outside world. This server provides a single point of presence for all of our email servers behind it. In this case the server is passing on an email to our office email server inside the network. The published email server has some filtering that it does, but it doesn't always catch everything.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The fifth section of the alert shows the packet type, in this case it is TCP, and the time-to-live (TTL) of the packet when it arrives, in this case it is 54. It also shows the type of service (TOS) which is 0x0. This also is not crucial to the analysis of this detect.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
```

```
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The sixth section of the alert shows the packet ID, the IP header length, the datagram length, and the don't fragmet (DF) flag. Looking at these fields, nothing seems out of the ordinary for them. Since there is only the one packet to analyze, there is nothing to compare it to in order to see if it is out of sequence. The IP header length is correct and the datagram length is the usual Ethernet datagram size, especially since the don't fragment bit is set. These fields are also not crucial to the detect.

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

The seventh section of the alert shows the TCP flags that are set, the TCP sequence number in hex, the acknowledgement number in hex, the Windows size in hex, and the TCP header length. It is important to note that this packet has the ACK bit set. Most likely this packet is the final ACK in the threeway handshake. Since the first two packets in this handshake do not appear, it is most likely that they did not contain information and were normally formed packets. Since only the one packet is logged, there is no way to compare the TCP sequence number to see if the sequence is out of order or not. We also do not have the previous ACK number to compare with to see whether or not the acknowledgement number is correct. The remaining two fields do not have much bearing on this detect and appear normal.

Section 5

This particular type of attack is a fairly common attack in the virus community. The attack relies on the victim opening an attachment to an email that contains code as part of the payload. Once the attachment is opened the code is executed and the virus attempts to spread. It attempts to protect itself by making modifications to files and the registry on the computer, which reruns the virus code when various conditions occur. It attempts to make alterations to programs that could be used to remove the code from the system. Some programs attempt to gain access to information that can then be sent to another location and used there.

Section 6

At the time of this detect, an entry was also entered into our email gateway log showing the arrival of the message containing the virus. This is shown in the following:

```
I,06/29/01,13:15:52,3B3C7F88.B82:11:64386,Unknown,,GWDOMAIN,GWPO,USER,,Snowhite and the Seven
Dwarfs - ,GWDOMAIN,INET,,MIME,hahaha@sexyfun.net,1,0,30816,0,
```

This log shows that the message was an incoming message, the date and time that the message was received, email system configuration information, that it was received from an unknown recipient, that it was received into the mail system domain and post office, that it was addressed to a user, the subject line, the gateway it was received through, the encoding scheme, the network in the receipt from, and more mail system information.

This virus had been in the wild for some time and we were getting an average of four to five attempts per week. Most of our other filtering attempts did not work well, since the traceback on the email did not always produce results.

Section 7

This attack is a form of active targeting. While the attack is directed at a single destination, the attack mechanism is non-discriminatory and does not have any sort of intelligence behind it. The attack is directed more at a user rather than an individual system and may attack any or all systems that are used by the user.

Section 8

The severity of this attack is a 1. I arrive at that conclusion through the severity of attack formula in the IDIC course using the following values. This attack was conducted to and from email servers with the corporate network; however, the actual transfer of information was a naturally occurring process. This makes the criticality portion rate a 4. While this attack is unable to succeed without user intervention, the potential loss from the attack succeeding is a DOS attack, the lethality of this attack is a 3. The server in this incident was running current software with all of the patches in place; however, the servers were only used as transport mediums in the attack, so the system countermeasures rates a 4. The weak point of this system is on the network countermeasures. The servers in question do not have any packet filtering capabilities and the local network is only protected by the corporate firewall, which wasn't tuned enough to block this attack. There isn't a local firewall, which drops the rating even more. The only saving grace is that there is an IDS which will at least alert on the attack. For this reason I rated the network countermeasures as a 2. So, the formula expands to (4+3)-(4+2) which results in a score of 1.

Section 9

Since it is not clear whether or not the attack can be stopped at the packet level, the recommendation is to move forward with more packet level detection and analysis. At the same time, efforts should continue to train users to be aware of this type of attack and the ways that it can be avoided. The anti-virus software should be updated to a sufficient level to detect the spread of this virus and to prevent it and administer and alert if it is found. This time the defenses were not compromised, but the type of attack indicates that constant monitoring is needed. No major changes in defenses are necessary.

Section 10

If this packet was part of a complete session, why doesn't the IDS contain alerts for the initial SYN, SYN/ACK and the final FIN, FIN/ACK, ACK?

```
[**] MAILVIRUS - SnowWhite Trojan Outgoing [**]
06/29-07:15:32.230178 0:0:C:9:F1:F9 -> 0:A0:C9:C8:5B:79 type:0x800 len:0x5EA
192.70.15.10:3055 -> 10.10.10.45:25 TCP TTL:54 TOS:0x0 ID:36049 IpLen:20 DgmLen:1500 DF
```

```
***A**** Seq: 0x42261C59  Ack: 0xDF4EA9A7  Win: 0x832C  TcpLen: 20
```

a) an alert such as this triggers on the payload of the packet which would be empty for the initial and final packets
b) it did, they just aren't shown here
c) because it is an outgoing packet and the IDS only tracks certain packets that go from the inside to the outside of the network
d) because there weren't any, this is a SMTP attack

Answer: a. Many alert signatures are based on the payload of traffic, which usually does not get transferred during the initial two thirds of the connection handshake or the disconnection traffic.

Alert #4 – IIS Unicode Alert

Section 1

This detect is from the network where I work. The network is a small network consisting of a Class C subnet. The network uses DHCP for all workstations and assigned addresses for all other network attached devices. The network runs Novell's Netware, Windows NT/2K, and Sun's Solaris for network operating systems and Windows 98/NT/2K on the workstations. The network is connected to a trusted WAN system that uses routing rules for internal security and is protected by firewalls from external access, and another trusted network that has their own security in place.

Section 2

The Intrusion Detection System used to obtain this detect is Snort version 1.7 running on a Unix server. The Intrusion Detection System is placed on a hub at the front of our network so that it can monitor the incoming and outgoing traffic to our network without slowing down the network or causing packet loss on the network. The alert is generated by Snort applying its http_decode preprocessor to the packet. The preprocessor converts payload characters %XX to the ASCII equivalents and then analyzing the payload for intrusion signatures.

Section 3

In this instance, a computer at work that belongs to our Operations Manager, who is technically proficient, generated the alert. She also spends a good deal of time working on the Internet. We were in the middle of performing a security audit when this alert triggered our Intrusion Detection System. Since security concerns were already high, this alert sent a couple of people scrambling for disconnection of the workstation from the network or for alerting supervisors that suspicious activity was taking place on the network.

We looked at the type of attack, which is an IIS Unicode attack, and determined that if someone wanted to use this type of attack the person would want information back. If an attacker spoofed the source address of the attack then the information would be sent back to the spoofed address and not to the attacker. Given that this is the case, it was determined that the source address was most likely not spoofed. Also, given that it was a single destination address and not very many alerts, it was not likely that it was a DOS attack, which would be the only other reason to spoof the address in this type of attack.

Section 4

The IIS Unicode attack, CVE-2000-0884, is a simple attack that can do a significant amount of damage to a system. The attack uses a flaw in the way that an IIS server interprets directory information in URLs. By using this exploit an attacker could bypass file and directory security and retrieve any file on the system, not just those in the web directories. In this manner the attacker could retrieve sensitive information, personal files, and system files. In its more advanced form, an attacker could go beyond retrieving information to the point of actually replacing files on the web server with a file of their own which could potentially give the attacker root access on the system.

While many alerts were logged, only a couple of the typical alerts are shown in this paper.

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

The first section of the alert shows the messages that snort generated for the alert. In this instance it is Snort's stream preprocessor looking for the IIS Unicode attack. Stream preprocessors are a little different than the simple Snort rules in that they look at more than a single packet and perform some advanced packet normalization. This particular stream preprocessor looks at http code in the payload of packets in order to match a particular attack signature. In this case the Unicode escape character combined with other characters were present in the payload of the packets that triggered the alert.

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

The second section shows the date and time of the alert. From examining the time signatures, it is clear that the attack isn't a large scale attack since the time between alerts is more than a couple of seconds.

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20
```

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

The third section of the alert shows the source address and port and the destination address and port. All of the alerts are being generated by the same address, so our focus could be centered on the activity of the source address. The attack is also focused on a single Internet address, which will make notifications easier if the attack is determined to be valid. The incrementing source port numbers show that the packets are probably not crafted and that the computer system does not have a great many programs running. The single destination port shows that the system being attacked is most likely a web server, the focus of this particular kind of attack.

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

The fourth section of the alert shows the packet type, time-to-live, type of service, and packet ID. The consistency shown across all of the packets as to the fact that it is TCP traffic with the same TTL values and TOS values shows that the packet is most likely not crafted or from various servers.

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

The fifth section of the alert shows the IP header length, the length of the datagram, and that the Don’t Fragment bit is set. This shows that the IP header is properly formatted since the length is the appropriate 20 bytes in length. While the datagram length varies, that just means that the various datagrams carried varying amounts of information. Finally, the DF bit being set is not unusual for this type of traffic.

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

The sixth section of the alert shows that the ACK and PUSH flags are set, which means that the packet is acknowledging a previous packet and sending or pushing data in this packet. It shows nothing significant about the sequence number, Ack number, and Window frame, which are all consistent with TCP traffic. It finally shows that the TCP header is the requisite length of 20 bytes.

Section 5

The attack works by using the [..], called dot-dot, to traverse directories. The dot-dot was used in DOS to move up one directory. This made it so that the user would not have to know the name of parent directories. When this reached web programming, it became an easy way to signify to the system the relative path to another file on the system. For example, a graphic used in a banner on the system may be stored in a central location so that it can be reused on many different pages, thus cutting down on the amount of storage needed for the page. The path to this file from the current directory may be ../../graphics/file. In DOS this isn't bad, but with an exploit, it could allow an attacker to pull up any file on the system.

A person visiting a web-site interacts with the server on an anonymous user level. This user has certain file and system privileges. This is of itself not enough for this attack to succeed. Most servers, however, contain small programs (CGI scripts, Javascript, or small binaries) that execute at a higher privilege level than the anonymous user. It is an exploit in these programs that allows the program to move throughout the system when it receives a request for information from a file using the dot-dot relative file location. The system allows the access since it is a program and not a user accessing the file. The program, not knowing a violation has occurred, passes the information back to the attacker the same it would any other information on the server.

Section 6

The attack originated on the computer of our operations manager, who is very active on the Internet. Since distrusting fellow employees is something that I try to avoid, my first thought was that someone was using her computer as a pass-through to attack a site on the Internet. She reported that on that morning she had visited many web sites, one of which was the site found on the destination address' server. I began thinking that perhaps the alert was generated by traffic that only looked like an IIS Unicode attack to the sensor.

After further investigation with the company who owned the web-site it was determined that a programmer had programmed a scriptlet that would load into the browser and retrieve information from the server on demand. The unfortunate part is that the script used relative file information in order to retrieve picture files. Since the scriptlet was sending back file addresses that contained the dot-dot notation, the ID sensor triggered because the payload of the packet matched what would be in the payload of an IIS Unicode attack.

The traffic was known traffic for the web site this time, but next time the exploit may not be due to a programming error. The IIS Unicode attack is not performed through a particular piece of software, but rather through a bug in the web server code. Information can be found on many different sites about the particulars of the attack: http://www.securityfocus.com/bid/1806 and http://www.eeye.com/html/Research/Advisories/AD20001003.html. The attack can be prevented, however, by downloading the patch from Microsoft at: http://download.microsoft.com/download/winntsp/Patch/q269862/NT4ALPHA/EN-US/prmcan4i.exe.

Section 7

In some respects you could say that this attack was evidence of active targeting. The target server was specific and the information that was being retrieved from the server was highly specific. Although in this instance the information retrieval was innocent, not all such attacks are like this one. In an actual IIS Unicode attack the attack would be directed at a specific web server even if the information that could be gathered may not be known. The attack also did not move from one server to another. All of the packets were directed to the same destination to the http port on the server. Therefore, it was a directed attack that turned out to be a false positive.

Section 8

The severity of this attack is a 5. I arrive at that conclusion through the severity of attack formula in the IDIC course using the following values. This attack was conducted from a computer on the local network and directed at a web server on the Internet. Since the target machine was a web server the criticality of the attack is a 5. The potential loss from the attack succeeding is the gathering of data that may be sensitive in nature, so the lethality of this attack is a 4. As the server in this incident was a web server that was on a foreign network, it is unknown whether it was running current software with all of the patches in place; however, since the code was used deliberately and a patch was out to fix this exploit, it is safe to say that not all of the security patches had been applied. For this reason the system countermeasures rates a 2. The network countermeasures are also on the foreign network so the full extent of the network security is unknown. However, the company who owned the website was unaware that this was going on so the network defenses must allow this type of traffic to occur without any detection. For this reason I rated the network countermeasures as a 2. So, the formula expands to (5+4)-(2+2) which results in a score of 5.

Section 9

Once the extent of the issue is known, it is easier to make a defensive recommendation. First and foremost it is recommended that all of the servers, especially the web servers, be brought up to date by applying the most recent security patches for the system. Second, some type of intrusion detection should be employed by the company to determine what traffic is going back and forth from their web server. And finally, some procedures need to be put into place to check the work of programmers to ensure that they are not violating any security protocols by programming around security procedures. This would include the reverification of all of the custom programs currently installed on the server. While these recommendations would not ensure that the server would not be attacked, it would make it easier to detect and recover from an attack if one succeeded.

Section 10

What information is this attacker most likely hoping to get from the attacked server?

```
[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:46.021669 10.10.10.113:1686 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:49445 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x8F269B  Ack: 0xF7BB80  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:20:52.688599 10.10.10.113:1688 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:57381 IpLen:20 DgmLen:534 DF
***AP*** Seq: 0x8F26BF  Ack: 0xF79D30  Win: 0x2238  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
06/19-11:22:00.637708 10.10.10.113:1705 -> 192.35.13.12:80
TCP TTL:128 TOS:0x0 ID:17191 IpLen:20 DgmLen:536 DF
***AP*** Seq: 0x911B88  Ack: 0xF7BFFC  Win: 0x1F4A  TcpLen: 20
```

a) Website information
b) A list of users on the website
c) files outside of the website's public directories
d) hardware configuration information

Answer: c. While an attacker can use this exploit to gain further access to the system, this access can only come from either pulling user/password files from the system or by planting a Trojan on the system. The main use of the IIS Unicode attack is to get around file system ACLs.

Alert #5 – Large Amount of ICMP Traffic

Section 1

This detect is from the network where I work. The network is a small network consisting of a Class C subnet. The network uses DHCP for all workstations and assigned addresses for all other network attached devices. The network runs Novell's Netware, Windows NT/2K, and Sun's Solaris for network operating systems and Windows 98/NT/2K on the workstations. The network is connected to a trusted WAN system that uses routing rules for internal security and is protected by firewalls from external access, and another trusted network that has their own security in place.

Section 2

The Intrusion Detection System used to obtain this detect is Snort version 1.8 running on a Unix server. The Intrusion Detection System is placed on a hub at the front of our network so that it can monitor the incoming and outgoing traffic to our network without slowing down the network or causing packet loss on the network. The rule used to generate the alert is a common information rule: alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)"; itype: 3; icode:4; sid:396; rev:1;). Normally this is not considered suspicious traffic.

Section 3

In looking at this information, it is unlikely that the source IP of the traffic is spoofed. However, it is a possibility that the destination server's IP is spoofed somewhere on the Internet and that traffic is being sent back to the destination server. This doesn't make much sense, though, given the response from the server. It could also be possible that someone is trying to use the server as an IP forwarding agent to attack another machine, which would be an interesting possibility given that the destination IPs correlate to Microsoft, Amazon.com, and UUNET (all popular targets).

Section 4

In looking at the logs and alerts generated by Snort, most of the time I gloss over the traffic that I know I need to look at, but that is usually benign. Most of the time these alerts are ICMP informational messages such as Destination Unreachable or TTL Expired. These do catch my eye, however, when there are a great number of them grouped closely together, as is the case in this detect. In this case, there were a large number of ICMP control messages relating to the Don't Fragment bit being set.

```
10/15/01-21:41:40.484019 [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
10/15/01-21:41:41.093900 [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
10/15/01-21:41:41.101778 [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
```

```
10/15/01-21:41:44.196857  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
10/15/01-21:41:55.537614  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
10/15/01-21:41:55.545573  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
10/15/01-21:41:56.030069  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 206.29.192.200
10/15/01-21:41:56.587768  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.144.81
10/15/01-21:41:56.595653  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.144.81
10/15/01-21:41:56.603639  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.144.81
10/15/01-21:41:58.764209  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.46.238.15
10/15/01-21:42:07.474219  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.252.68.19
10/15/01-21:43:04.500612  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.171.182.16
10/15/01-21:43:04.508413  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.171.182.16
10/15/01-21:43:08.849865  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.171.169.17
10/15/01-21:43:12.255977  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.171.169.17
10/15/01-21:43:13.498710  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.216.183.15
10/15/01-21:43:13.506843  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.216.183.15
10/15/01-21:43:19.035794  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.216.183.15
10/15/01-21:43:19.043744  [**] [1:396:1] ICMP Destination Unreachable (Fragmentation Needed and DF bit was
set) [**] {ICMP} 10.10.10.45 -> 192.216.183.15
```

What caught my attention about this traffic is that it is a large amount of ICMP messages that generally aren't encountered on this network occurring so close together. The second attention grabber is that this traffic is originating from a server on our network going to an address outside of the network. Usually this type of traffic originates outside of the network and is sent to a machine on the network. Finally, it is important to note that the messages are going to more than one external address, which means that this is occurring for more than a single session.

Section 5

The possible attack in this scenario is simply a group of anomalous alerts that may or may not be an attack. An attacker often tries to use normal traffic to do network mapping or computer OS fingerprinting. A group of alerts such as this could be such an attempt. The attacker sends a particular type of packet to a destination in order to observe the response from the system. Any response from the computer is considered a positive response. This response can then be analyzed to try to determine the operating system of the computer, and any possible vulnerability that it might have.

Section 6

A closer look at what is occurring on the network to prompt this activity is needed. Generating a full alert on these packets gives the originating packet information, and perhaps some insight into what is happening.

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

The first line of the full alert shows that there wasn't a mistake by the first scan through the traffic, that this is indeed an ICMP information packet on the Don't Fragment bit.

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

The second line confirms the time that the alert occurred, the source IP address (a server on the network), and the destination IP address.

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

The next two lines give the detailed information about the IP datagram including that the protocol is ICMP, the Time To Live (TTL) of the packet, the Type Of Service (TOS) of the packet, the packet ID, the IP header length, the datagram length, the ICMP message type and code, and the explanation of what that type and code mean in English.

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

What becomes important at this point is the Original Datagram Dump, which begins by showing that the original packet is from the http port of a server on the Internet and goes to a machine on the network at work. Analysis could probably stop right there, as I will explain later, but finishing the analysis of the packet will only confirm my suspicions that this is a false positive that something strange is occurring.

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

The next line begins to give the detailed information about the original packet. From this can be determined that the protocol used was TCP, a commonly used protocol against which the other details will be compared; that the Type Of Service is typical for TCP; that the packet ID is not uncommon; that the length of the IP header is typical for TCP; and that the datagram length is typical for TCP across an Ethernet connection.

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

The next line gives the rest of the detailed information about the original packet. It shows that the Urgent and FIN flags are set, not uncommon for TCP; that the sequence numbers, when compared with surrounding packets, were consistent with TCP traffic; that the ACK and WIN were consistent with TCP; that the TCP header length was appropriate; and that the UrgPtr information was consistent with TCP.

The suspicion that the destination server's IP address had been hijacked didn't pan out, especially once I was able to see the original datagram packet. The destination address of the original packet falls in the range of our dial-in machines. These addresses are virtual machines that use the responding server as their default gateway. This is why the server responded the way it did, because it couldn't forward the packets to the virtual machine address.

Section 7

The initial inspection of this incident would suggest that there is active targeting. In this incident we have multiple sources and a single destination. The multiple sources are also from various points on the Internet. If the attack were an outgoing attack, then there would also be evidence of active targeting since several of the IP addresses were to the same company, all of which are high target values for attackers. Upon further inspection, though, the evaluation of whether or not there was active targeting doesn't matter since the event was determined to be a false positive.

Section 8

The severity of this attack is a (-3). I arrive at that conclusion through the severity of attack formula in the IDIC course using the following values. This alert was generated based on the response from a server on the network. However, further evidence shows that the original traffic was based on dial-in workstation traffic. Based on this I would give the criticality portion rate a 3. While this was not an actual attack, it did make me take notice. However, since there was a low risk for a Denial Of Service attack, I would rate the lethality of this attack as a 2. The server in this incident was running current software with all of the patches in place; however, the server was only sending out the informational packets. The dial-in system was probably up to date since dial-in access is tightly controlled. So the system countermeasures rating should be a 4. Since these were alerts from an internal system, the network countermeasures don't play a large role. The alert were logged by the network IDS, however, which I believe allows for a network countermeasures rating of 4. So, the formula expands to (3+2)-(4+4) which results in a score of (-3).

Section 9

This situation certainly doesn't call for throwing up barriers around the network and disabling dial-in access. Since this alert was proven to not be an attack but rather an event of significance that turned out to be a false positive, the defensive recommendation is to review how events of significance are determined. If it turns out that procedures were followed and that the procedures were effective, then no change needs to be made. If, however, events of this nature are always being investigated, then it may be time to revamp the procedures or retune the ID sensor to be aware of situations such as this that can be high false positives. The dial-in access also needs to be examined to determine whether or not certain parameters need to be set as minimum requirements to connect in order to avoid possible future alerts.

Section 10

If the following packet was sent from a server with IP Forwarding enabled, what would be the most likely explanation for the traffic?

```
[**] ICMP Destination Unreachable (Fragmentation Needed and DF bit was set) [**]
10/15-21:14:04.684432 10.10.10.45 -> 192.252.68.18
ICMP TTL:128 TOS:0x0 ID:64897 IpLen:20 DgmLen:56
Type:3  Code:4  DESTINATION UNREACHABLE: FRAGMENTATION NEEDED
** ORIGINAL DATAGRAM DUMP:
192.252.68.18:80 -> 10.10.10.247:1036
TCP TTL:46 TOS:0x0 ID:37685 IpLen:20 DgmLen:1500
**U****F Seq: 0xB4FF8E1E  Ack: 0x1C1D1E1F  Win: 0x2223  TcpLen: 8  UrgPtr: 0x2627
** END OF DUMP
```

a) the packet was corrupted in transmission and sent to the server by mistake
b) the original datagram contained a spoofed address

c) the transmission window on the network is too small for the server to receive all of the packet that it needs to forward
d) the server needs to route the packet to the destination computer but the don't fragment bit is set and the transmission window of the destination computer is too small

Answer: d. This can be determined from the original datagram dump which shows that the original request came from a different IP address.

The Merging of Virus and Intrusion Incident Handling

GIAC Certification Whitepaper
Thomas Shepherd
November 2001

Introduction

The response measures to a virus alert are often developed separately from the response measures to an intrusion alert. Viruses are largely considered to be automated attacks on computers systems that an attacker releases and then sits back to watch what happens. Intrusions, on the other hand, are considered to be active directed attacks by an individual on computer systems where the attacker gets some personal gain from the attack. Over the past year and a half the distinction between viruses and intrusions have been blending. Attackers are turning to viruses to do the traditional grunt work of reconnaissance on networks, including the distribution of rogue code known as trojan programs. Security experts are also seeing more intrusion exploits used in virus code. As these viruses become more "hacker" like in nature, security experts are dealing with them in the same manner that they do more traditional intrusions.

Enter Melissa

When the Melissa virus hit the Internet community many security people were confident that they were going to be able to deal with this virus the same way that they had dealt with the hundreds if not thousands of viruses that had come before it. Melissa had other ideas. The traditional attack vectors for a virus were monitored and protected by anti-virus software and by the vigilance of IT professionals who had trained their users to not open unknown or executable email attachments. Melissa, however, spread through new attack vectors that hadn't been considered before.

The Melissa virus arrives in an attachment to an email. Since this is a Word/Macro virus, the attachment looks like a normal word document and not an executable program. The unknowing user opens the document thinking that it is okay. If macros are enabled then the virus macro runs automatically and infects the computer system. If macros are disabled, the virus macro does not run but remains dormant in the document waiting for the document to be opened on a system that does run macros. Upon execution of the viral code the virus sets the macro security settings to their lowest level so that the user will not be notified any time in the future that the macro is being executed.

The virus proceeds to propagate itself by sending an email with an infected attachment to the first 50 entries in every Microsoft MAPI address book that can be accessed on the system. Also, if any of the entries are mailing list addresses then the email will go to everyone on the mailing list. This is essentially the same as an attacker obtaining the address book files and directly sending the virus to the first 50 entries in the book. At the same time, though, this is more insidious in that the email looks like it is coming from a trusted source, which it is, and not a spoofed or unknown source.

After this has been done, the virus infects the Normal.dot template file with the viral code. The Normal.dot template file is the file that Microsoft Word uses as the template to build all new documents. This means that any new documents will be infected with the virus. This is potentially damaging in that every time a new document is created or a document that has been created with the infected Normal.dot template is opened, the virus will try to propagate itself. Since the file that it uses to propagate itself is the file running the macro, damage could occur if there is any sensitive information in the file that was created or opened and is now being sent out to the first 50 entries in the address book.

This was the first large-scale virus epidemic that began to use attack mechanisms that resembled the use of intrusion methods. After this epidemic there came other viruses; some that were similar, such as Anna Kournikova and Hybris, and others that employed other intrusion vectors, such as the Lion worm and the Adore worm.

Enter Code Red

Most of the security community learned from Melissa that the new generation of viruses would be more deadly, more complex, and would propagate quicker throughout the Internet community. When Code Red was discovered the Internet community was quick to react with analysis of the attack and security patches which would not just clean up the machine after the attack but that would prevent the attack in the first place. Like many traditional intrusions, the main targets of the Code Red attack were servers.

Code Red arrives via the .ida buffer overflow attack. The target server is attacked through the execution of code that is supplied at the end of the http request that overflows the buffer. Once the virus arrives it initially sets up some environmental resources that it will need. It then creates 100 threads of the virus. If the server has the Chinese language pack installed then it uses all 100 threads to try to infect other servers. Otherwise, the first 99 threads are used to try to infect other servers and the final thread is used to deface the system's website. The worm also checks the system time and if the date is past the 20th of the month it will use all 100 threads to attack www.whitehouse.gov.

Like most viruses the main goal of Code Red is to propagate itself around the Internet. However, unlike most viruses the attack vector for code red was an intrusion exploit in the idq.dll, announced just 26 days before the virus hit. The virus was also designed to launch a Denial-Of-Service attack against www.whitehouse.gov; whereas most viruses attack the system on which they are executed. Also unlike most viruses the infection remained in a volatile state, which meant that it was dropped out of memory when the computer was turned off or rebooted.

Since the attack vector for this virus was through an http get request, intrusion detection systems (IDS) were able to track and to some degree defend against the attack. The payload signature was posted on many IDS related websites. The following is the packet signature that was posted on www.eeye.com.

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%
u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%
u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%
u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

(Line breaks were added between the N's for ease-of-reading on this page. Please remove them prior to updating your signature database.)[1]

[1] EEye Digital Security alert AL20010717.

While the use of an IDS will help, the best way to defend against this attack is to apply the security patch that is being distributed by Microsoft at:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

Enter Nimda

The fight against Code Red showed that it was possible to track and defend against viruses using an IDS. So, when the first alerts of Nimda came in, procedures were followed to analyze the attack and get an alert out. No one ever dreamed that a virus could spread as quickly as Nimda did, but this was due to the fact that Nimda combined many attack vectors in order to propagate itself. According to Richard D. Pethia, the director of the CERT centers, Nimda is particularly bad because it "is the first significant worm or virus that attacks both computers that act as servers and those that are desktop computers."[2] Due to the multiple attack vectors and types of targets, many different methods of combating the spread of Nimda were employed.

In its spread across the Internet, the Nimda virus exploited up to 16 previously published vulnerabilities. According to CERT advisory CA200126_FA200126 the virus can spread through the following means:

- from client to client via email
- from client to client via open network shares
- from web server to client via browsing of compromised web sites
- from client to web server via active scanning for and exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities (VU#111677 and CA-2001-12)
- from client to web server via scanning for the back doors left behind by the "Code Red II" (IN-2001-09), and "sadmind/IIS" (CA-2001-11) worms[3]

Although Nimda retained the traditional vector of attack for viruses, through email propagation, the majority of its attack vectors are clearly through means that are used by attackers to gain access to vulnerable systems. The main reason that this is a considered a virus is that it is an automated attack that propagates itself.

The most obvious method of Nimda replication uses the methodology that was so effective in the spread of the Melissa virus, through mass emailings. Unlike Melissa, which would send emails to the first 50 entries in an address book, Nimda sends email to every entry in the address book along with any addresses that it can find in an inbox or outbox and to any addresses that it can find in local .htm(l) files. It sends itself as a MIME encoded attachment that can be automatically run by email systems that are not protected against the "Automatic Execution of Embedded MIME Types" exploit. Users and companies found that fairly quickly an email system could become overwhelmed with the number of incoming and outgoing emails. Many companies were forced to shut down their email servers just to have time to clean their

[2] Pethia, Richard D., "Information Technology…," pg 2.
[3] Danyliw, Roman, Chad Dougherty, Allen Householder, and Robin Ruefle.

systems. This in effect became a non-directed Denial-Of-Service attack against corporate email systems.

Code Red and its variants demonstrated the second propagation method as being effective. Nimda uses the following scheme to pick an IP address on the Internet to try to propagate itself:

50% of the time the address will be from the same Class B subnet
25% of the time the address will be from the same Class A subnet
25% of the time the address will be chosen at random

Nimda will then try three different methods to gain control of the machine. The vulnerabilities exploited are the back doors produced by either the Code Red II or sadmind/IIS worms, or through exploitation of the "IIS/PWS Extended Unicode Directory Traversal Vulnerability", or the "IIS/PWS Escaped Character Decoding Command Execution Vulnerability". Once Nimda has control of the machine it attempts to transfer a copy of the code through TFTP to port 69.

Also virus-like, Nimda infects Internet related files that have the .HTM, .HTML, or .ASP extensions. Each file with one of these extensions that Nimda finds will have Java code appended to the file. This Java code is designed to run in the background when a person browses to the page. This code then downloads a copy of the virus code, a file named README.EML, from the server to the client. Some browsers will also run this file, infecting the new machine, while others will simply download the file, leaving the virus dormant until the file is executed. Thus the program is placed on the system in a trojan-like manner, waiting to be executed and exploited.

In the same trojan-like manner Nimda will also infect other binaries, which can then reinfect the system. Nimda will search through any writable directories on any shared drives, including mapped network drives, about which the computer is aware and infect any executable files in those directories. Nimda will also make a MIME-encoded copy of itself in each directory with either a .EML or .NWS extension. These files will then be run if the file is viewed in the Windows Explorer with the preview option enabled.

In a more intrusion-like manner Nimda will change several system security permissions. It will share the C: drive as the hidden share C$ if it is not already there, thus allowing connection to the entire system drive of the computer. And it will also create a "Guest" account on the computer with a group membership in the Administrators group. This gives the worm, and a potential attacker, root access to the system in order to perform more involved operations and to ensure that the worm may fully infect the system.

The spread of Nimda saw widespread use of Intrusion Detection Systems to not only detect the spread of the worm but to combat its spread through the use of security modifications enacted by IDS alerts. Companies were seeing so much traffic caused by Nimda that some elected to shut down routers and firewalls in order to curtail the spread, in effect causing a Denial-Of-Service. Others, however, were using IDS technology that would alter or modify router ACLs or firewall permissions based on alerts that were generated. And, since the spread of the virus was through traditional intrusion means, companies were able to follow the spread of the virus through their own companies to make sure that the virus had been eradicated from their systems.

Security Suites

If Melissa was a wake-up call and Code Red was a call to arms, Nimda was a declaration of war. Most security companies realize that the security battle is being fought on many different fronts and until security can be consolidated and coordinated, attackers will be able to find and exploit the many holes in a network's defense systems. In order to make this coordination happen, security suites have begun to be developed. These suites combine many individual security packages into a comprehensive and integrated security solution.

Firewalls are typically used for blocking specific addresses or types of traffic, but they aren't usually used to block specific data from being transferred across a network. This is where a Network Intrusion Detection System (NIDS) becomes effective. The new security packages are blending these two technologies into a single solution that will be able to dynamically allow or disallow traffic across a network based on a set of rules. These rules can be either locally programmed or received as updates from the security suite manufacturer. As new vulnerabilities and attacks are discovered, rules will be updated and sent out; thus automatically changing the traffic patterns of a network. These updates will not only allow for the typical firewall tactics of blocking specific addresses or types of traffic, but they will also allow for the blocking of traffic that contains a known virus, trojan program, or attack code.

The advent of network firewalls and NIDS was a boon to the network security field. Firewalls could protect an internal network from outside intruders and NIDS could alert on any anomalous or bad traffic that was going across the network. The only problem with this is that about 82% of all attacks on a network are from inside the system. Data that appears to these security systems as normal network traffic goes by without an alert being generated. If that data is harmful and there isn't a corresponding traffic rule set up on the firewall or NIDS, then a computer can still be attacked without being detected.

The new Host Intrusion Detection Systems (HIDS) pick up where the NIDS and network firewalls leave off. These systems reside on the host computer and look for any anomalies on the system. Most of the HIDS will look at the data that is coming into the computer for any traffic that is anomalous or contains harmful data. The difference between a NIDS and a HIDS is that a HIDS will also look at traffic coming from a disk drive or external port as well as the network. Many HIDS will also include a personal firewall. In this way a computer can reach a higher level of security from the network just as an internal LAN can reach a higher level of security from the Internet by installing a network solution.

On the computer itself, whether it is a workstation or a server, a HIDS will do consistency checking to determine whether or not there has been any file corruption or tampering. A HIDS will take a "snapshot" of the computer system and then at intervals it will check to see whether or not the system is in the same state. This is helpful to determine whether or not system files have been changed. Sometimes another administrator changes them and sometimes an attacker changes them. Either way, any changes cause an alert, which in turn generates a response. Many times this is coupled with anti-virus software that can not only detect changes to system files but can also disinfect them if the changes are caused by a known attack. In this way, both known and unknown attacks are most likely to be detected and handled.

Developing a comprehensive security suite is not an easy task. Currently there are no industry standards that provide direction on how security packages should communicate between each piece of the suite or with other security suites. There are several initiatives that have been started to provide this communication by organizations such as Incidents.org (www.incidents.org) and Dshield.org (www.dshield.org). Some companies have a jump on this by providing several individual security packages that can communicate with the other security products from the same company. Companies such as Symantec (www.symantec.com), which has several products designed to secure the enterprise; Cisco (www.cisco.com), which is also leading the way in providing security products for the network infrastructure; and probably the most comprehensive security suite is from Computer Associates (www.cai.com) with their eTrust suite of security software, which includes a network intrusion detection program, a firewall, and anti-virus software.

Incident Handling

Viruses are often dealt with in the same manner for each new virus. System administrators hope that they have taught their users well enough to not open unknown email attachments or download unscrupulous programs from the Internet so that new viruses will pass them by. The unlucky ones, who one way or another end up being infected, hope that the new anti-virus signature updates and cleaning programs will be released before any real harm can happen to their systems. Once the cleaners are released, system administrators rush around cleaning off systems. Once the anti-virus software is done cleaning off the virus, the system administrators sit back and hope that the next virus passes them by.

These powerful new tools are changing the incident handling procedures for companies that have them installed. It used to be that the announcement of a new virus meant that virus signature updates were soon to follow. Now the announcement of a new virus is taking on the same ramifications as a precursor to a more directed intrusion. Attackers are using viruses as means to map vulnerable systems or to automatically deploy rogue code through the use of "Trojan Dropper" viruses.

Some system administrators who have had more technical training or experience move beyond the typical reaction to a virus by examining their systems for any part of the virus that the cleaners have missed. With the new tools being offered by the security suites, however, a system administrator will be alerted to any changes in the system and can more readily identify those changes that may be a problem. Open security vulnerabilities or rogue code that is left by a virus will be much easier to detect and newly opened ports will be reported as such. Much more information will be at the disposal of the system or security administrator.

Among this information will also be the originating attack address. While more often than not this will be another unsuspecting victim who has been infected and is passing the virus along, knowing where the virus is coming from will make it easier to notify that system administrator to clean off their systems. It is expected that within the next year viruses will begin to interoperate with one another making it ever harder to defend against and eradicate them. Having security software suites that will work together and communicate with each other like a net across the Internet will make it just that more difficult for not only the writers of virus and rogue code but for intruders and hackers as well.

Another issue surrounding the response to viruses and intruders is the legal issue of due diligence. While due diligence has often been discussed in legal terms surrounding intrusions and Denial-Of-Service attacks, it is not usually discussed when it comes to the spread of viruses. Many security experts are coming to the realization that there's little difference between an intruder gaining information that allows an attack on another computer and a virus that obtains information that allows it to attack another computer. The release of sensitive information, whether it is by direct means from an intruder or by automatic means by a virus, is also hotly discussed. Security experts are now asked to deal with network incidents, whether it is a virus or an intrusion, using the same set of response guidelines.

Conclusion

Computer viruses are not a new threat, but the response to them is merging with the response to more traditional forms of intrusion. The definitions of a virus, a worm, and a trojan program are blending into a single definition of an 'Automated Intrusion.' As this definition approaches the more traditional definition of an attack or intrusion on a network or system, the employment of more coordinated and connected security suites or improved incident handling procedures is taking place. Attacks, whether they are an 'Automated Intrusion' or a direct intrusion, are beginning to be treated the same and are being handled with the same procedures, which are aided by the development of comprehensive security suites.

References

Ulsch, Macdonnell and Joseph Judge. "Bitter-Suite Security." PriceWaterhouseCoopers. January 1999. URL: http://www.pwcglobal.com/extweb/manissue.nsf/2e7e9636c6b92859852565e00073d2fd/a1adbd9bcbff9fb085256738006dafbc/$FILE/Bitter+Suite+Security.PDF (30 October 2001).

Pethia, Richard. "The Melissa Virus: Inoculating Our Information Technology from Emerging Threats." Testimony before the Subcommittee on Technology, Committee on Science, U.S. House of Representatives. 15 April 1999. URL: http://www.cert.org/congressional_testimony/pethia9904.html (30 October 2001).

Northcutt, Stephen. "What was the Melissa virus and what can we learn from it?" SANS Institute Intrusion Detection FAQ. 22 April 1999. URL: http://www.sans.org/newlook/resources/IDFAQ/What_Melissa_teaches_us.htm (30 October 2001).

"Personal Firewalls: Firewall Protection for PCs and Home Networks." Vectors White Paper. July 2001. URL: http://www.dell.com/us/en/gen/topics/vectors_2001-personal_firewalls.htm (30 October 2001).

".ida 'Code Red' Worm." EEye Digital Security alert AL20010717. 17 July 2001. URL: http://www.eeye.com/html/Research/Advisories/AL20010717.html (30 October 2001).

Wong, Nicole C. "'Code Red' Worm Likely to Return." Washington Post Washtech.com. 30 July 2001. URL: http://www.washtech.com/cgi-bin/udt/WTW.PRINT.STORY?client=washtech-test&storyid=11547 (30 October 2001)

"CodeRedII Worm Analysis." EEye Digital Security alert AL20010804. 4 August 2001 URL: http://www.eeye.com/html/Research/advisories/al20010804.html (30 October 2001).

Donze, Kenneth. "Defending Against Code Red II Using Symantec NetProwler and Intruder Alert." SANS Institute Information Security Reading Room. 15 August 2001. URL: http://www.sans.org/infosecFAQ/threats/netprowler.htm (30 October 2001).

Danyliw, Roman, Chad Dougherty, Allen Householder, and Robin Ruefle. "CERT Advisory CA-2001-26." CERT. 25 September 2001. URL: http://www.cert.org/body/advisories/CA200126_FA200126.html (30 October 2001).

Pethia, Richard D. "Information Technology – Essential But Vulnerable: How Prepared Are We for Attacks?" Testimony before the House Committee on Government Reform, Subcommittee on Government Efficiency, Financial Management and Intergovernmental Relations. 26 September 2001. URL: http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html (30 October 2001).

"Nimda Worm." Counterpane Security Alert. 27 September 2001. URL: http://www.counterpane.com/alert-nimda.html (30 October 2001).

"NIMDA Worm/Virus Report – Final." SANS Institute. 3 October 2001. URL: http://www.incidents.org/react/nimda.pdf (30 October 2001).

"The Latest Kaspersky Reports (#24)." Kaspersky Lab News. 11 October 2001. URL: http://www.kaspersky.com/news.asp?tnews=0&nview=1&id=240&page= (30 October 2001).

SANS GIAC
Intrusion Detection Certification
Analyze This Scenario

Data Analysis Corp.

November 15, 2001

RE: Security analysis of data traffic

Ladies and Gentlemen,

We're glad we have the opportunity to help you with an analysis of the data traffic going across your network. We would like to thank you for providing the alert, scan, and out-of-spec files that we requested in order to do our analysis. In looking at the volume of information that was provided it was determined that we would not have the time or resources to look over all 3 ½ months of information. Looking at an average week of alerts, however, can produce a competent analysis. For this reason the first seven days in May were analyzed.

Our analysis of the data that was provided indicates that there were 57,427 events logged by the system across 23 different classifications during this time frame. For these events, the system logged 498 distinct sources and 4,264 distinct destinations. The sources and destinations were a mixture of University owned and external machines. The system also logged 2,798 different port scans during the same time frame. The detailed analysis of this traffic is attached. Along with this are defensive recommendations and listings of machines that are possibly compromised.

It is recommended that, based on the analysis of your systems, a more comprehensive analysis of your network be performed. Looking at network topology, security policies and procedures, and host-based intrusion detection systems, including anti-virus software, would provide this comprehensive analysis. This can be a lengthy process, but a process that would increase the security of your network to the level at which it needs to be. We would be happy to talk with you about this service.

Sincerely,

Thomas Shepherd

File Selection

Upon receiving the information from the university, it was determined that there would be insufficient time and resources to examine and analyze all of the files provided. For that reason, all of the files were scanned for an appropriate sampling of data that would yield a sufficient analysis. The following table shows the files that were selected for this analysis.

Date	Alert File	OOS File	Scan File
May 1, 2001	Alert.010501.log	Oos_May.1.2001.log	Scans.010501.log
May 2, 2001	Alert.010502.log	Oos_May.2.2001.log	Scans.010502.log
May 3, 2001	Alert.010503.log	Oos_May.3.2001.log	Scans.010503.log
May 4, 2001	Alert.010504.log	Oos_May.4.2001.log	Scans.010504.log
May 5, 2001	Alert.010505.log	Oos_May.5.2001.log	Scans.010505.log
May 6, 2001	Alert.010506.log	Oos_May.6.2001.log	Scans.010506.log
May 7, 2001	Alert.010507.log	Oos_May.7.2001.log	Scans.010507.log

Alert Overview

The seven alert log files were combined in order to gain an understanding of what types of alerts were logged. The following table shows a summary of the 23 different alerts that were categorized from the files.

Alert Type	# Alerts	# Sources	# Destinations
UDP SRC and DST outside network	38519	48	537
Watchlist 000220 IL-ISDNNET-990517	7592	49	52
Attempted Sun RPC high port access	5871	3	3
SYN-FIN scan!	2460	3	2415
External RPC call	1168	11	768
Possible trojan server activity	551	100	110
connect to 515 from outside	235	2	213
Watchlist 000222 NET-NCFC	221	7	14
Queso fingerprint	216	25	52
SMB Name Wildcard	167	109	118
WinGate 1080 Attempt	123	53	83
Back Orifice	62	3	58
SNMP public access	42	2	42
TCP SRC and DST outside network	40	15	30
Port 55850 tcp - Possible myserver activity - ref. 010313-1	37	10	12
High port 65535 udp - possible Red Worm - traffic	26	16	13
Null scan!	26	20	21
ICMP SRC and DST outside network	19	7	12
Tiny Fragments - Possible Hostile Activity	19	4	15
High port 65535 tcp - possible Red Worm - traffic	15	8	8
SUNRPC highport access!	9	3	3

NMAP TCP ping!	5	5	5
connect to 515 from inside	4	1	1
Totals	**57427**	**504**	**4585**

The above list is ordered by the frequency of occurrence of alerts. It should be noted that logging a great many alerts of one type does not necessarily show activity of an intrusion or criminal actions. An alert type that has a few occurrences may be just as problematic, if not more, than an alert type that has many thousand. Each alert type is explained in greater detail later in this analysis.

Top Ten Alert Sources

All Sources	Total Alerts	University Owned	Total Alerts
206.190.36.120	31969	10.10.97.200	197
24.21.203.64	5864	10.10.220.110	20
134.129.71.203	3186	10.10.253.24	12
212.179.43.225	3254	10.10.227.250	9
192.168.0.1	1581	10.10.228.82	8
212.179.8.194	1293	10.10.6.7	6
211.130.90.210	1154	10.10.98.153	4
134.129.125.158	1325	10.10.219.38	3
212.179.79.2	811	10.10.253.41	3
212.179.43.107	460	10.10.227.14	2

Top Ten Alert Destinations

All Destinations	Total Alerts	University Owned	Total Alerts
233.28.65.62	31969	10.10.229.166	5864
10.10.229.166	5864	10.10.210.86	3540
233.24.119.155	4577	10.10.227.6	1294
10.10.210.86	3540	10.10.217.130	510
10.10.227.6	1294	10.10.207.82	429
10.10.217.130	510	10.10.229.206	304
10.10.207.82	429	10.10.214.138	272
233.28.65.222	420	10.10.207.6	257
233.28.65.45	319	10.10.219.154	228
10.10.229.206	304	10.10.227.90	166

Detailed Alert Analysis

UDP SRC and DST outside network

This type of activity is caused by one of two events. Either the packets are being introduced from an outside source into the network erroneously or they are packets with spoofed source addresses being sent to off network destinations. The only other cause of this activity would be misplacement of the Snort sensor; however, it is assumed that the sensor was placed correctly for this evaluation.

A look at the sources of these alerts shows a diversity of source addresses. Quite a few sources are from the Yahoo subnet. Several sources are from educational institutions. And quite a few sources are from addresses that suggest other activity. On the whole, the destinations show how this traffic is entering the system. The destination addresses are usually in the range of the multicast address space. I would agree with the analysis of Chris Baker that this type of traffic suggests that the university network "belongs to a multicast group."[4]

The traffic that needs more in-depth analysis is traffic with a source address of either 169.254.x.x or 192.168.x.x. These address ranges are provided for private use on networks and should not be used in normal routed Internet traffic. The fact that they show up here suggests that there is a problem with this traffic. Looking at the destinations for these addresses suggests activity in which the sender of the traffic is either trying to hide their identity or they are probing other networks for vulnerabilities.

More information about private addresses may be found here: http://www.ja.net/CERT/JANET-CERT/prevention/cisco/private_addresses.html

Defensive Recommendation for UDP SRC and DST outside network

It is recommended that the University review the multicast groups to which it belongs to make sure that there is a need for the traffic. If there is no need to belong to a multicast group, then it would be wise to drop out of it. If there is a need to belong to a multicast group, then it would be prudent to modify the rules of your firewall and router to only allow certain multicast addresses into the network and to tune the intrusion sensor to filter out these events when the source or destination is a multicast address.

More in-depth analysis needs to be performed to find out the source of the 169.254.x.x and 192.168.x.x traffic. However, since these should not be routed across the Internet, it is advisable to place a block on them, both inbound and outbound, on the firewall and the router. As the University's network is MY.NET.x.x, it is advisable that an intrusion rule be set up to detect when these addresses are used in order to get a better understanding of what is happening.

Watchlist 000220 IL-ISDNNET-990517

This type of activity is caused by traffic originating from an ISP in Israel. An alert is generated by any traffic with a source IP on the Class B subnet of 212.179.x.x. Logging this traffic does not immediately mean that it is harmful traffic, it means that at one point there was enough

[4] Baker, Chris. GIAC Practical.

harmful or erroneous traffic coming from these addresses to put them on a list of traffic about which to be aware.

Since the rule that generates these alerts shows all traffic coming from a source of 212.179.x.x regardless of the traffic contained within the packets, it is sometimes difficult to examine the alerts for harmful traffic. In this instance, most of the traffic was directed at ports 6346, 6688, and 6699. These ports are most commonly associated with Gnutella and Napster traffic. There was a mixture of other ethereal ports with did not indicated any suspicious activity.

There were three alerts that bear further analysis by the University.

The first was traffic received on May 6th at 3:49am from 212.179.61.243:1058 destined to MY.NET.60.11:23. This is indicative of telnet traffic, which is usually not allowed across insecure Internet connections. The source port could show that it was traffic from an Internet based game called NIM, but this can't be confirmed.

The second was traffic received on May 7th at 4:45pm from 212.179.72.53:32880 destined to 10.10.253.43:25. This is indicative of SMTP traffic. If 10.10.253.43 is an email server this may not be a problem. If 10.10.253.43 is not an email server, then this traffic should be considered highly suspicious.

The third was traffic received on May 4th at 6:13am from 212.179.80.165:4992 destined to 10.10.178.42:317. If the destination port is correct, this is indicative of access to a program called Zannet, which is a program designed to do remote Windows 95/98 management. Unless there is a good reason for someone outside of the network to be doing remote management, this traffic is highly suspicious.

Defensive Recommendation for Watchlist 000220 IL-ISDNNET-990517
The University should review its concerns about traffic from 212.179.x.x. If there aren't sufficient reasons to allow the traffic into the network, then it would be prudent to add rules to the routers and firewalls to prevent access from 212.179.x.x into the network unless the traffic was initiated from within the network.

The University should also review its policies concerning Gnutella and Napster traffic. Gnutella and Napster are both programs that allow a user to share files on their local computer across the Internet. Many security professionals consider Gnutella or Napster on a network to be a security risk. If it is decided that Gnutella or Napster traffic is not allowed, then rules should be added to the firewall to prevent related traffic. More information about Gnutella may be found here:
http://www.incidents.org/detect/gnutella.php
More information about Napster may be found on Napster's home page at:
http://www.napster.com/

There are also 2 CVE candidates for Napster exploits: CAN-2000-0281 and CAN-2000-0412.

The University should also look at the three machines that had questionable traffic, 10.10.60.11, 10.10.253.43, and 10.10.178.42 for possible access violations or compromise.

Attempted Sun RPC high port access

This type of activity is caused by attempts to access a service on the machine at port 32771. While it is possible that this is an ethereal port on the machine, it is also the port that is sometimes used by UNIX machines as an alternate to the portmapper service. The portmapper service can be queried to find out what other ports on the machine have services available. More information about securing UNIX services, such as portmapper, may be found here: http://www.cert.org/tech_tips/usc20.html

All of the traffic from these alerts was focused at port 32771. Looking at the source addresses for these alerts shows that it is highly probable that this traffic was reconnaissance traffic looking for machines that have port 32771 enabled.

Looking at the registration information from the two different networks show that the connections were made from AT&T's @Home dial-up network and AOL's dial-up network.

Query for 24.21.203.64: http://www.arin.net/cgi-bin/whois.pl?queryinput=24.21.203.64&B1=Submit+Query

```
@Home Network (NETBLK-ATHOME)   ATHOME            24.0.0.0 - 24.23.255.255
@Home Network (NETBLK-HOOVER1-AL-2) HOOVER1-AL-2   24.21.200.0 - 24.21.207.255
```

Query for 205.188.153.98 or 205.188.153.101: http://www.arin.net/cgi-bin/whois.pl?queryinput=205.188.153.98&B1=Submit+Query

```
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

   Domain System inverse mapping provided by:

   DNS-01.NS.AOL.COM          152.163.159.232
   DNS-02.NS.AOL.COM          205.188.157.232

   Record last updated on 27-Apr-1998.
   Database last updated on  12-Nov-2001 19:54:32 EDT.
```

External access to the portmapper service is highly questionable, especially from dial-up accounts such as these. While the traffic may be innocent, it is more likely that it is a reconnaissance effort.

Defensive Recommendations for Attempted Sun RPC high port access

The University should review its systems to determine whether or not there are any systems running the portmapper service on port 32771. If there are machines that are running the service on that port, then the University needs to review its policies concerning access to the portmapper service from external networks. It is recommended, though, that the service should not be able to be accessed from outside of the local network. If the descision is to block the portmapper service, then a firewall rule can be put in place to block inbound traffic to port 32771 that does not have a corresponding outbound session.

There are 4 CVE entries or candidates for portmapper exploits: CVE-1999-0168, CAN-1999-0195, CAN-1999-0632, and CAN-2001-0617.

SYN-FIN scan!

A packet entering the network with both the SYN and FIN flags set causes this type of activity. This is often used to perform reconnaissance on the network to map those machines that are available and what services are available on them. A packet with both the SYN and FIN flags set are not part of normal network activity. As PJ Goodwin states it in his GIAC practical, "These types of scans are often used to fingerprint operating systems and are a precursor to a more directed attack. A diligent scanner can provide an entire map of the target network."[5] More information about SYN/FIN scans can be obtained here: http://www.whitehats.com/info/IDS198

The attack analysis of this traffic suggests that the traffic indicated here is from a program called Synscan. The significant indication of this is that the source and destination ports are both port 21, the FTP service. The Synscan program scans subnets looking for open FTP servers.

Since there were only three sources for this activity, it was easy to trace down from where they originated. The first source address is a private network address and probably came from within the University network. The second address originated from Japan. The final source address is not significant since only the single alert was generated.

Defensive Recommendation for SYN-FIN scan!

Given that the source of the most amount of traffic was a private network address, the University needs to review its policies about placing other networks, such as laboratory or trial networks, onto the same wire as the University network. In any respect, unless it is needed, traffic coming from or going to a private network address should be suspect.

The only other recommendation that I would give at this point is to obtain a firewall product that will work in conjunction with an intrusion detection system that would allow for dynamic updates to the firewall rules based on an attack posture from the intrusion detection system. If the intrusion detection system recognizes a scan of this type then it can update the firewall rules to block this traffic for a specified amount of time so as to minimize the information gathered from the scan.

[5] Goodwin, PJ. GIAC Practical.

External RPC call

This type of activity is caused by an attempt to access the portmapper service on port 111 of the destination machine. As stated earlier, the portmapper service provides a listing of other ports and their associated services on the machine. There are many existing exploits for the RPC service, some of which are listed in the following CVEs:

CVE-1999-0003, CVE-1999-0008, CVE-1999-0208, CVE-1999-0212, CVE-1999-0228, CVE-1999-0320, CVE-1999-0353, CVE-1999-0493, CVE-1999-0687, CVE-1999-0696, CVE-1999-0900, CVE-1999-0969, CVE-1999-0974, CVE-2000-0508, CVE-2000-0771, CVE-2001-0331, CAN-1999-0078, CAN-1999-0195, CAN-1999-0568, CAN-1999-0613, CAN-1999-0625, CAN-1999-0632, CAN-1999-0795, CAN-1999-1127, CAN-1999-1225, CAN-1999-1258, CAN-2000-0114, CAN-2000-0544, CAN-2000-0800, CAN-2001-0509, CAN-2001-0662, CAN-2001-0717, and CAN-2001-0779.

Looking at the destinations of these alerts shows that those sources that logged significant alerts were performing reconnaissance scanning on the University network. The destinations for each of these significant alert sources ranged from MY.NET.132.x to MY.NET.137.x. Each scan was used to either determine that a connection could be made to the portmapper service on port 111, or to make a connection to the service to request information.

Defensive Recommendation for External RPC call

Unless there is a very good reason to share the portmapper service with machines that are outside of the local network, it is recommended that rules be added to the University's routers and firewalls to block inbound and outbound communication to port 111.

Possible trojan server activity

Packets that have a source or destination port of 27374 cause this type of activity. A machine may use this port as a normal ethereal port, but it is also a well-known port for a program called Sub7. The Sub7 program can act as a Trojan server to give attackers root access to the system and possibly the network. More information about the Sub7 program can be found here: http://www.commodon.com/threat/threat-sub7.htm

Since port 27374 can also be used as an innocent ethereal port, analyzing traffic of this nature is sometimes difficult. Most of the traffic shown in these alerts is simply normal network traffic that happened to have port 27374 as either the source or destination ports.

Of all the sources that triggered an alert, the alert traffic indicates that five internal machines may be compromised.

The first machine is 10.10.227.250. The relevant traffic is shown here.

```
05/01-13:50:40.709377 [**] Possible trojan server activity [**] 10.10.227.250:27374 ->
213.112.59.55:2961
05/01-13:50:42.524902 [**] Possible trojan server activity [**] 10.10.227.250:27374 ->
213.112.59.55:2961
05/01-14:05:52.715105 [**] Possible trojan server activity [**] 10.10.227.250:27374 ->
213.112.59.55:3303
```

This traffic shows several connections to address 213.112.59.55 on May 1st. The first set of connection traces could be passed off as an ethereal port connecting to a remote system. However, the likelihood that an internal machine would be able to connect to the same remote system 15 minutes later over the same port is possible but not likely. It is possible that this machine has been compromised.

The second machine is 10.10.228.82. The relevant traffic is shown here.

```
05/07-07:49:15.882807 [**] Possible trojan server activity [**] 10.10.228.82:27374 -> 64.180.2.183:2528
05/07-07:49:15.886039 [**] Possible trojan server activity [**] 10.10.228.82:27374 -> 64.180.2.183:2529
```

This traffic shows two almost simultaneous connections to 64.180.2.183 on May 7th. The issue here is that either the internal system created two sessions from the same port within a second of each other, which is anomalous, or this is response traffic going from the machine to the two sessions opened by the external machine. In either case there is a strong possibility that this machine has been compromised.

The third machine is 10.10.227.82. The relevant traffic is shown here.

```
05/01-01:06:53.977757 [**] Possible trojan server activity [**] 10.10.227.82:27374 -> 213.112.59.55:2097
05/01-01:51:35.339998 [**] Possible trojan server activity [**] 10.10.227.82:27374 -> 213.112.59.55:2684
```

The traffic shows two connections to 213.112.59.55 on May 1st. While it is possible that the 10.10.227.82 machine went through all of its ports for making session connections in 45 minutes, it is highly improbable that the computer would make two different sessions to the same external machine using the same source port. It is likely that this machine has been compromised.

The fourth machine is 10.10.97.165. The relevant traffic is shown here.

```
05/05-03:30:53.642872 [**] Possible trojan server activity [**] 10.10.97.165:27374 -> 210.56.24.91:4387
05/05-03:30:55.965342 [**] Possible trojan server activity [**] 10.10.97.165:27374 ->
202.124.202.30:4256
```

The traffic shows almost simultaneous connections to two external machines across the same internal port, which is anomalous. Either this is response traffic from the service as port 27374 on the internal machine to the two open sessions or it is the internal machine establishing two open sessions on the same port, which is not allowed. It is highly likely that this machine is compromised.

The fifth machine is 10.10.153.203. The relevant traffic is shown here.

```
05/04-18:17:08.930321 [**] Possible trojan server activity [**] 10.10.153.203:27374 ->
217.52.60.174:1142
 05/04-18:17:21.065135 [**] Possible trojan server activity [**] 10.10.153.203:27374 ->
217.52.60.174:1152
```

The traffic shows two different connections to 217.52.60.174 on May 4th. While it is possible that the 10.10.153.203 machine went through all of its ports for making session connections in 13 minutes, it is highly improbable that the computer would make two different sessions to the same external machine using the same source port. It is likely that this machine has been compromised.

Defensive Recommendations for Possible trojan server activity

Traffic going to or coming from port 27374 does not guarantee that a Trojan server program exists on a machine; so simply blocking the port is not feasible. More comprehensive attack signatures are available for detecting possible Trojan server activity on this port. It is recommended that the University employ an intrusion detection sensor with the updated attack signatures to more refine the alerts on this port.

It is also recommended that the University make a close examination of the five machines that were identified as questionable to determine whether or not they have been compromised. Once the new alert signatures are in place, the University should also make it a regular procedure to examine any machine that triggers the new alerts. This should minimize, but not eliminate, the risk of the spread of Trojan programs.

connect to 515 from outside

This type of activity is caused by a machine outside of the home network attempting to access port 515 on the destination machine. The service that usually resides on port 515 is the printer spooler port. While it is possible that external access to this port is allowed, there are also known vulnerabilities in this service that would allow an attacker to gain root access to the system. According to Becky Bogle this type of traffic has been seen before and can "lead to root compromise from both local and remote systems."[6] More information about probes to this port may be found at: http://www.sans.org/newlook/alerts/port515.htm

Looking at the destinations of this alert shows that the first source was performing a scan of the network. Traffic analysis shows that the first source scanned the 10.10.132.x network twice, the 10.10.133.x network once, and the 10.10.137.x network once looking for machines where the printer spooler service is available. From the traffic presented here it is unclear whether or not such machines were discovered.

The second source made only the single session connection. This was either allowed traffic or a failed scan attempt. The traffic presented here did not indicate any further activity for the source and destination presented.

Defensive Recommendation for connect to 515 from outside

It is recommended that the University review its policies regarding access to the printer spooler port from machines outside of the University's network. It is recommended, though, that access be restricted to the local network. If the University wishes to restrict access to the local network, then rules should be put in place on the University's routers and firewalls to block inbound and outbound traffic on port 515.

[6] Bogle, Becky. GIAC Practical.

Watchlist 000222 NET-NCFC

This type of activity is caused by traffic originating from an ISP in China. An alert is generated by any traffic with a source IP on the Class B subnet of 159.226.x.x. Logging this traffic does not immediately mean that it is harmful traffic, it means that at one point there was enough harmful or erroneous traffic coming from these addresses to put them on a list of traffic about which to be aware.

Since the rule that generates these alerts shows all traffic coming from a source of 159.226.x.x regardless of the traffic contained within the packets, it is sometimes difficult to examine the alerts for harmful traffic. In this instance, however, most of the traffic was directed at port 25. This port is most commonly associated with SMTP email traffic. The only other suspicious port activity was port 113, associated with the IDENT service and email traffic. On some alerts there was also a few other ethereal ports with did not indicated any suspicious activity.

Defensive Recommendation for Watchlist 000222 NET-NCFC

The University should review its concerns about traffic from 159.226.x.x. If sufficient reasons do not exist to allow the traffic into the network, then it would be prudent to add rules to the routers and firewalls to prevent access from 159.226.x.x into the network unless the traffic was initiated from within the network.

The University should also look at all of the destination machines in order to determine whether or not they have been compromised. If the machines are running the SMTP service, then email traffic logs should be examined to determine whether or not the traffic sent to that machine was valid traffic.

Queso fingerprint

A program named Queso, which is designed to do Operating System (OS) fingerprinting, causes this type of activity. The program sends a series of packets to a machine in order to observe the response from the machine. The responses can then be analyzed and an attempt is made to match the responses to a known set of responses from a particular OS. Once an OS has been identified, the attacker can then attempt to find known exploits for that OS. This tool is mainly used in reconnaissance efforts by attackers. More information about this type of stealth scanning may be found here:
http://www.cert.org/incident_notes/IN-98.04.html

Since the Queso program attempts to guess which operating system is used on a computer, it is often seen as a precursor to a more directed attack. Looking at the traffic patterns in this alert shows that most of the scans were looking for machines that are running the Gnutella server (see Watchlist 000220 IL-ISDNNET-990517 for more information on Gnutella). It is unknown how many of the machines scanned were running the Gnutella server.

There were several scans which attempted to identify machines that are running the SMTP service, port 25, and/or the IDENT service, port 113. It is also unknown whether or not these machines were identified or compromised.

The top producing source, 24.180.133.11, is significant in that it only connected to the single destination host, but it scanned many different ports. It is unclear whether or not the person was looking for anything in particular or just having a hard time correlating which operating system is in use on that machine. A lookup of the address of that machine reveals that it is on the AT&T @Home network and is likely a dial-up account, notoriously used in network scanning.

Query request for 24.180.133.11: http://www.arin.net/cgi-bin/whois.pl?queryinput=24.180.133.11&B1=Submit+Query

```
   @Home Network (NETBLK-HOME-2BLK)HOME-2BLK         24.176.0.0 - 24.183.255.255
   @Home Network (NETBLK-BLTMMD1-MD-1) BLTMMD1-MD-1 24.180.128.0 - 24.180.143.255
```

The second most producing source, 152.66.214.122, scanned several machines looking at port 8080, a popular alternate port for HTTP services or a WinGate proxy server. Intranet servers often use Port 8080 for their HTTP service port. The traffic does not indicate that an attempt was made to compromise any of the destination machines. A lookup of the address of that machine reveals that the traffic originated from Budapest, Hungary, which suggests that it is suspicious traffic.

Query request for 152.66.214.122: http://www.arin.net/cgi-bin/whois.pl?queryinput=152.66.214.122&B1=Submit+Query

```
Technical University of Budapest Centre of Information Systems (NET-HUNGARNET-B01)
      Muegyetem rkp. 9. R. III. 310.
      BUDAPEST, H-1111
      HU

      Netname: HUNGARNET-B01
      Netblock: 152.66.0.0 - 152.66.255.255

      Coordinator:
        Technical University of Budapest(BME) Centre of Information Systems (EISzK)  (ZT9-ARIN)
remzso@eik.bme.hu
        +36 1 4631821

      Domain System inverse mapping provided by:

      NIC.BME.HU            152.66.115.1
      NS.BME.HU             152.66.116.1

      Record last updated on 09-Dec-1999.
      Database last updated on  13-Nov-2001 19:54:52 EDT.
```

Traffic analysis on other sources does show that there is a possible compromise of system 10.10.100.225. Alerts show that a Queso fingerprint was performed on the system on May 3rd by the external host 128.46.156.117. The following day an attempt was made by the same host to access the SUNRPC highport of 32771.

What makes this a difficult determination is the possibility that the University has an agreement with other universities regarding this type of traffic. Looking at the network block of the above source shows that the traffic originated at Purdue University.

Query for 128.46.156.117: http://www.arin.net/cgi-bin/whois.pl?queryinput=128.46.156.117&B1=Submit+Query

```
    Purdue University (NET-PURDUE-NET)
      Engineering Computer Network Electrical Engineering Building
      West Lafayette, IN 47907
      US

      Netname: PURDUE-ECN-NET
      Netblock: 128.46.0.0 - 128.46.255.255

      Coordinator:
         Moya, James Michael  (JMM118-ARIN)  moyman@ECN.PURDUE.EDU
         +1 765 494 2349

      Domain System inverse mapping provided by:

      HARBOR.ECN.PURDUE.EDU 128.46.154.76
      MOE.RICE.EDU                 128.42.5.4
      NS.PURDUE.EDU         128.210.11.5
      PENDRAGON.CS.PURDUE.EDU      128.10.2.5

      Record last updated on 24-May-1999.
      Database last updated on  14-Nov-2001 19:54:47 EDT.
```

Defensive Recommendation for Queso fingerprint
See Watchlist 000220 IL-ISDNNET-990517 for the recommendation on the Gnutella program.

Since the Queso program uses anomalous traffic to guess at the operating system on a machine, the only real defense to the program is to make sure that all of the security patches have been installed on the machine. This will minimize the potential vulnerabilities of the machine that can be connected with the operating system. Care should still be taken to examine the services that are available to external sources as these can also be identified by this program.

SMB Name Wildcard
This type of activity is caused by a connection to port 137 on the destination machine. Windows uses this port for the Netbios name lookup. When a machine needs to know the name of another machine but only has the IP address, it attempts to connect to port 137 on the IP address that it knows to query the name of the machine. This is often used in conjunction with port 139 to enumerate the open shares on the machine. Unless there is a policy allowing for the connection to this port from external sources, it is almost always an attempt to gain information for attacking open drive shares on the network. This type of activity on the internal network is almost always benign.

Most of this traffic is benign traffic that is common to Windows NT systems. Windows uses SMB traffic in normal network operations to perform name resolution. Most of the traffic logged on this alert can be considered normal network activity if the network is running Windows NT on some of the machines. More information may be obtained at:
http://www.sans.org/newlook/resources/IDFAQ/port_137.htm

or
http://www.dshield.org/ports/port137.html

Traffic logged from the top alert source, 10.10.220.110, indicates that the machine performed two scans on the 10.10.19.x subnet of the University network.

Traffic was also logged from the private network address of 192.168.0.1, which made attempts to connect to three different University machines.

Defensive Recommendation for SMB Name Wildcard
Since this traffic is normal network activity for Windows NT networks, the University should review the NT network connections that exist. While it is possible, it is not recommended that access to port 137 be allowed for machines that are external to the local network. It is thus recommended that rules be set up in the University's routers and firewalls to block inbound and outbound access to port 137.

The University should also examine the machine at 10.10.220.110 for possible compromise or illegal activity.

The University should also examine its policies surrounding the use of private network addressing on systems connected to the main University network.

WinGate 1080 Attempt
This type of activity is caused by an attempt to scan for and access a proxy server, known as WinGate, on port 1080. WinGate is a proxy server that can be used by external sources to act as a proxy for them. Many attackers use this server to either surf the web anonymously or to mask the origination of an attack on another destination. More information about the WinGate proxy server vulnerabilities can be found in the following CVEs: CVE-1999-0290, CVE-1999-0291, CVE-1999-0441, CVE-1999-0494, CAN-1999-0657, and CAN-2000-1048.

Since port 1080 is not used as an ethereal port it is safe to say that all of the sources shown in this type of alert are trying to connect to a possible WinGate proxy server on the destination machine. Since the bulk of the sources attempted to connect to only one or two machines it is likely that the destination addresses got published somewhere on the Internet as being possible proxy machines.

Analysis of the traffic from the second most alert producer, 24.234.89.152, suggests that machine 10.10.98.193 may be running the WinGate proxy service. On May 4th several alerts were generated by the address 24.234.89.152 attempting to connect to port 1080 on the destination machine.

Defensive Recommendation for WinGate 1080 Attempt
It is recommended that the University scan their network, especially machine 10.10.98.193, to determine whether or not there are any unknown operating WinGate proxy servers. If unknown proxy servers are detected, it is recommended that they be shut down and the machines checked

for possible compromises. All known WinGate servers should be patched and brought up to current security standards.

It is also recommended that the University review their policies regarding connection by external machines to local proxy services. It is recommended that proxy services on the local network should not be available to external machines. Stateful firewalls and routers should block any inbound traffic to a proxy server without having corresponding traffic originating from within the network.

Back Orifice

This type of activity is caused by attempts to connect to port 31337. This port is most commonly used as the service port for a program known as Back Orifice. The Back Orifice program is usually installed on a system through an email attachment. Once on the system it will commonly listen on port 31337 for connections. Any person connecting to this service will have more control over the system than a person who is sitting at the computer. More information about Back Orifice may be found in the following CVEs: CAN-1999-0660 and CAN-2000-0562.

Analysis of the traffic generated by these alerts shows that the first source address, 203.155.244.220, scanned the 10.10.98.x network looking for active instances of the Back Orifice program. Further analysis also shows that the second source, 62.136.14.116, scanned the 10.10.97.x network looking for active instances of the Back Orifice program. The third source, 203.45.203.107, only attempted the one connection, which makes it unclear what analysis to make of it.

A look at the network blocks of the first two addresses indicates that the scans were probably performed from dial-up accounts, the first being in Bankok and the second being in the UK.

Query for 203.155.244.220: http://www.apnic.net/apnic-bin/whois.pl?search=203.155.244.220

```
% Rights restricted by copyright. See
http://www.apnic.net/db/dbcopyright.html
% (whois6.apnic.net)

inetnum:     203.155.192.0 - 203.155.255.255
netname:     COMNET-TH
descr:       KSC Commercial Internet Co. Ltd.
descr:       2/4 Samaggi Insurance Tower 10th Fl.,
descr:       Viphavadee-Rangsit RD
descr:       Thungsonghong, Laksi
descr:       Bangkok 10210
country:     TH
admin-c:     CW246-AP
tech-c:      TO94-ORG
remarks:     service provider
mnt-by:      APNIC-HM
mnt-lower:   KSC-ADMIN
changed:     hostmaster@apnic.net 19990218
changed:     hostmaster@apnic.net 20011016
source:      APNIC
```

```
person:     Craig White
address:     KSC Commercial Internet Co.,Ltd.
address:     2/4 Samaggi Insurance Tower 10th Fl., Viphavadee-Rangsit
Rd.,
address:     Thungsonghong, Laksi
address:     Bangkok 10210
country:    TH
phone:      +66-2-9797777 ext. 7071
e-mail:    cwhite@ksc.net
nic-hdl:    CW246-AP
mnt-by:     KSC-ADMIN
changed:     netadmin@ns.ksc.co.th 20011012
source:     APNIC
```

And the query for 62.136.14.116: http://www.ripe.net/perl/whois?query=62.136.14.116&.=Submit+Query

```
inetnum:     62.136.0.0 - 62.136.127.255
netname:     POL-CAG1
descr:       Energis Squared Dynamic IP Allocation
descr:       In case of problems please call +44 113 234 6068
descr:       Please do not send abuse reports to tech or admin contacts
descr:       Abuse reports to abuse@planet.net.uk please!
country:      GB
admin-c:      PJ3130-RIPE
tech-c:      PJ3130-RIPE
status:      ASSIGNED PA
notify:     ripe-adm@planet.net.uk
mnt-by:      AS5388-MNT
changed:      pedro.jones@energis-squared.com 20011015
source:      RIPE

route:       62.136.0.0/15
descr:       Planet Online Limited
descr:       The White House
descr:       Melbourne St.
descr:       Leeds LS2 7PS United Kingdom
origin:     AS5388
mnt-by:      AS5388-MNT
changed:      matthew@planet.net.uk 19990521
source:      RIPE
```

Further traffic analysis does not indicate that any of the scanned systems have been compromised.

Defensive Recommendation for Back Orifice

Since port 31337 is a possible ethereal port it is not recommended to block all traffic to this port. It is recommended that the University update their intrusion detection attack signatures that contain new rules on Back Orifice that will cause fewer false positives. After this alerts on the Back Orifice program will be easier to investigate.

SNMP public access

This type of activity is caused by an attempt to access SNMP resources using the default community of "Public." The SNMP protocol is used to exchange management information between machines that are running the SNMP agent. While management commands usually cannot be sent to machines using the Public community, a great deal of information about the machine may be obtained in this manner.

Traffic analysis indicates that the first source, 209.236.199.29, scanned the networks ranging from 10.10.132.x to 10.10.137.x looking for machines that not only have the SNMP agent installed on them, but machines that also have the default communities set. A look at the network block for this address indicates that this is probably a dial-up account.

Query for 209.236.199.29: http://www.arin.net/cgi-bin/whois.pl?queryinput=209.236.199.29&B1=Submit+Query

```
SuperNet, Inc. (NETBLK-SUPERLINK-BLK-1)
  39 Milltown Rd.
  East Brunswick, NJ 08816
  US

  Netname: SUPERLINK-BLK-1
  Netblock: 209.236.128.0 - 209.236.223.255
  Maintainer: SUPR

  Coordinator:
    Boyes, Truman  (TB55-ARIN)  truman@SUPERLINK.NET
    732-432-5454

  Domain System inverse mapping provided by:

  DNS.SUPERLINK.NET                209.236.128.128
  EARTH.SUPERLINK.NET              209.236.128.129

  ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

  Record last updated on 22-Jun-2001.
  Database last updated on  14-Nov-2001 19:54:47 EDT.
```

This type of traffic can be indicative of reconnaissance aimed at obtaining system information that may be later used in a coordinated attack. There are no indications, however, that any of the scanned machines were later compromised by a known exploit.

It is unclear whether or not the alert generated by the internal machine 10.10.71.39 was authorized or unauthorized traffic.

Defensive Recommendation for SNMP public access

The University should make sure that any machines that have SNMP agents installed do not have the default community settings still set. It is also recommended that unless it is necessary, access to SNMP agents should be limited to the local network. For this reason it would be a good idea to block inbound and outbound traffic to port 161.

It is also recommended that if the traffic originating from the one internal machine was not allowed traffic, that machine should be checked for any inappropriate or illegal activity.

TCP SRC and DST outside network

This type of activity is caused by one of two events. Either the packets are being introduced from an outside source into the network erroneously or they are packets with spoofed source addresses being sent to off network destinations. The only other cause of this activity would be misplacement of the Snort sensor; however, it is assumed that the sensor was placed correctly for this evaluation.

Traffic analysis of these alerts indicates that most of this traffic is ICQ and Microsoft Messenger traffic. Most of the sources were shown to be from either the AOL network address range or the AT&T @Home address range. Since it is unlikely that this traffic would be introduced into the University's network, barring a serious routing misconfiguration, it is likely that the source addresses of these alerts are spoofed.

There are known vulnerabilities in the ICQ and Microsoft Messenger programs. If an attacker wanted to exploit one of these vulnerabilities it would make investigation harder if the source address did not correspond to where the packet originated.

Defensive Recommendation for TCP SRC and DST outside network

A more in-depth investigation would need to be performed to find the true source of these packets. If it is a routing issue, which has been encountered before, then the routes need to be fixed. If it is traffic originating from the University's network with spoofed packets, the University will need to perform some host based intrusion detection to determine the source of the spoofed packets.

Port 55850 tcp - Possible myserver activity - ref. 010313-1

This type of activity is caused by traffic where the source or destination port is 55850. Traffic on this port could be caused due to a compromised system in which the Trojan program myserver has been installed. The myserver program will then listen on port 55850 for commands to the compromised machine. The myserver program is usually associated with the Unix tool "RootKit."

According to a GIAC analysis on August 20, 2000 (http://www.incidents.org/archives/y2k/082200.htm), it is possible that the installation of this program precipitates system information being mailed back to the attacker. Given this explanation, it is likely that the following systems have been compromised: 10.10.253.24 and 10.10.6.7. Both of these addresses showed signs of traffic being sent to the SMTP ports of servers outside of the University's network.

Defensive Recommendation for Port 55850 tcp - Possible myserver activity - ref. 010313-1

Since port 55850 is a possible ethereal port a rule to deny traffic using this port is not possible. It is recommended that the University employ a host based intrusion detection system on those machines that are most likely to be compromised by this exploit, namely UNIX and Linux based machines.

It is also recommended that the University check each of the following machines for possible compromise: 10.10.253.24, 10.10.6.7, 10.10.253.41 (if this is not a mail server), 10.10.6.34 (if this is not a mail server), 10.10.253.43 (if this is not a mail server), and 10.10.188.100 (if this is not a news server).

High port 65535 udp - possible Red Worm – traffic
This type of activity is caused by traffic where the source or destination port is 65535. Machines may use this port as an ethereal port, but the Red Worm, or Adore Worm, also uses it. Once the machine has been compromised, the worm institutes a back-door shell program that uses port 65535 to communicate. More information on the Adore Worm may be found here: http://www.europe.f-secure.com/v-descs/adore.shtml

Analysis of the alerts logged here does not indicate any compromised machines. Compromised machines would have traffic traveling to and from port 65535 on the machine. The alerts show that none of the Internal machines had traffic on this port and there is no evidence of active scanning from external machines on this port.

Some of the traffic does suggest that some local network machines may be running a network version of a popular game called Quake3. This can be seen by traffic across ports 27960 and 27961. The possible machines that are running this program are: 10.10.226.226, 10.10.70.242, and 10.10.71.69.

Defensive Recommendation for High port 65535 udp - possible Red Worm – traffic
It is recommended that the University continue to monitor traffic across this port. It is also recommended that the University upgrade their intrusion detection sensor rule set, which probably will include a more comprehensive rule for detecting Red Worm traffic.

It is also recommended that the University review its policies regarding the use of network-based games on the University's network.

Null scan!
This type of activity is caused by TCP traffic in which none of the usage flags are set. TCP traffic uses flags to tell the destination system what type of traffic is contained within the packet. A packet without any of these flags set is considered anomalous. Many attackers often use this type of traffic in reconnaissance and OS fingerprinting.

Traffic analysis indicates that a large amount of this traffic can be attributed to scans for possible active Gnutella servers on machines on the University's network. It is unclear whether people actively looking for exploitable programs generated the other alerts since the destination ports are not well-known exploit ports.

Defensive Recommendation for Null scan!
It is recommended that the University update their intrusion detection systems and firewalls to alert on and block traffic of this nature. There is no known normal TCP traffic that has no flags set.

See Watchlist 000220 IL-ISDNNET-990517 for the recommendation on Gnutella.

ICMP SRC and DST outside network

This type of activity is caused by one of two events. Either the packets are being introduced from an outside source into the network erroneously or they are packets with spoofed source addresses being sent to off network destinations. The only other cause of this activity would be misplacement of the Snort sensor; however, it is assumed that the sensor was placed correctly for this evaluation.

Most of the sources were shown to be from either the AOL network address range or the AT&T @Home address range.

Defensive Recommendation for ICMP SRC and DST outside network

A more in-depth investigation would need to be performed to find the true source of these packets. If it is a routing issue, which has been encountered before, then the routes need to be fixed. If it is traffic originating from the University's network with spoofed packets, the University will need to perform some host based intrusion detection to determine the source of the spoofed packets.

Tiny Fragments - Possible Hostile Activity

This type of activity is caused by the fragmentation of a packet into a smaller packet, which may not be able to be scanned. This technique is most often used to get by firewall or router filters that filter based on destination port or that ignore all of the other packet fragments after the first fragment. It is also possible, as Paul Asadoorian states in his GIAC practical, that "certain operating systems will crash if the fragmented packets are sent in such a way that they overlap."[7]

An analysis of the traffic indicates that none of these alerts were an attempt to circumvent intrusion detection systems. Fragmented traffic from three of the four source addresses can possibly be explained by examining from where the traffic originated. A query for the address block shows that the traffic originated in Taiwan.

Query for 202.39.78.125: http://www.apnic.net/apnic-bin/whois.pl?search=202.39.78.125

```
inetnum:    202.39.0.0 - 202.39.255.255
netname:    TWNIC-TW
descr:      Taiwan Network Information Center
descr:      4F-2, No. 9 Sec. 2, Roosevelt Rd.,
descr:      Taipei, Taiwan, 100
country:    TW
admin-c:    SO12-AP
tech-c:     NS10-AP
mnt-by:     APNIC-HM
mnt-lower:  MAINT-TW-TWNIC
changed:    hostmaster@twnic.net 20000811
source:     APNIC

person:     Shih-Chiung Ouyang
address:    Taiwan Network Information Center
address:    4F-2, No. 9 Sec. 2, Roosevelt Rd.,
address:    Taipei, Taiwan, 100
```

[7] Asadoorian, Paul. GIAC Practical.

```
country:    TW
phone:      +886 2 2341 1313 ext. 301
fax-no:     +886 2 2396 8832
e-mail:     oyang@twnic.net
nic-hdl:    SO12-AP
notify:     hostmaster@twnic.net
mnt-by:     MAINT-TW-TWNIC
changed:    hostmaster@twnic.net 20000808
source:     APNIC

inetnum:    202.39.78.0 - 202.39.78.127
netname:    HU-JUN-JIA-TN-NET
descr:      Hu, Jun Jia
descr:      No.178, Bao An Rd., Tainan
descr:      Tainan Taiwan
country:    TW
admin-c:    JJH19-TW
tech-c:     JJH19-TW
remarks:    This information has been partially mirrored by APNIC from
remarks:    TWNIC. To obtain more specific information, please use the
remarks:    TWNIC whois server at whois.twnic.net.
mnt-by:     TWNIC-AP
changed:    network-adm@hinet.net 20010724
source:     TWNIC
```

It is quite possible that traffic from this area has gone through many different networks to arrive at the University. Any one of these networks may introduce the fragmentation.

An analysis of the other source of this traffic does not indicate any fraudulent intentions.

Defensive Recommendations for Tiny Fragments - Possible Hostile Activity

It is recommended that the University keep machine operating systems and intrusion detection systems current. This will avoid many of the problems and exploits associated with tiny fragment exploits.

High port 65535 tcp - possible Red Worm – traffic

This type of activity is caused by traffic where the source or destination port is 65535. Machines may use this port as an ethereal port, but the Red Worm, or Adore Worm, also uses it. Once the machine has been compromised, the worm institutes a back-door shell program that uses port 65535 to communicate.

Unlike the analysis performed on the udp version of this alert, analysis of the traffic of this alert suggests that several systems may have been compromised and that several others should be checked for possible compromise. Traffic patterns examined in this alert are similar to published traffic patterns of the Red Worm.

Possible compromised systems include: 10.10.253.53, 10.10.100.230, 10.10.253.42

Defensive Recommendation for High port 65535 tcp - possible Red Worm – traffic

The university should check the following systems for possible compromise: 10.10.253.53, 10.10.100.230, 10.10.253.42, 10.10.179.78 (if it is not a mail server), 10.10.6.35 (if it is not a mail server), 10.10.1.9 (if it is not a DNS server), 10.10.4.3 (if it is not a mail server)

It is recommended that the University continue to monitor traffic across this port. It is also recommended that the University upgrade their intrusion detection sensor rule set, which probably will include a more comprehensive rule for detecting Red Worm traffic.

SUNRPC highport access!

This type of activity is caused by access to port 32771 on the destination machine. While it is possible that this is an ethereal port on the machine, it is also the port that is sometimes used by UNIX machines as an alternate to the portmapper service. The portmapper service can be queried to find out what other ports on the machine have services available. Guy Bruneau, in his GIAC practical, describes it this way: "Solaris rpcbind listens on a high numbered UDP port (32771), that may not be filtered since the default port is 111, which may bypass packet filters."[8]

There were three sources of this activity. Traffic from two of the sources indicated normal network traffic. Traffic from the third source, 128.46.156.117, shows signs of an attempted intrusion. On May 3rd this source performed a Queso fingerprint against the destination machine, which would have given an attacker pertinent information about what possible exploits are available. The source then returned on May 4th and accessed port 32771 in an attempt to transfer information.

Defensive Recommendation for SUNRPC highport access!

It is recommended that 10.10.100.225 be checked for possible compromise.

The University should review its systems to determine whether or not there are any systems running the portmapper service on port 32771. If there are machines that are running the service on that port, then the University needs to review its policies concerning access to the portmapper service from external networks. It is recommended, though, that the service should not be able to be accessed from outside of the local network. If the descision is to block the portmapper service, then a firewall rule can be put in place to block inbound traffic to port 32771 that does not have a corresponding outbound session.

NMAP TCP ping!

A program called NMAP causes this type of activity. NMAP is used primarily to map the active machines on a network. In this case, the NMAP program sends a TCP packet with the ACK flag set and an acknowledgement value of 0. The acknowledgement value of a packet should never be 0. An active machine will respond to this packet, thus announcing that it is active.

Traffic shows that there were five different sources and each source contacted only a single destination machine. Three different destination ports were targeted, port 25 (SMTP), port 53 (DNS), and port 80 (HTTP). Since only the five alerts were logged, there is not enough information to make a conclusive analysis of the alerts.

[8] Bruneau, Guy. GIAC Practical.

Defensive Recommendation for NMAP TCP ping!

It is recommended that the University employ a firewall and intrusion detection system that will alert on this type of traffic and provide a temporary firewall rule that would disallow traffic from the source for a specified amount of time.

connect to 515 from inside

This type of activity is caused by a machine inside of the home network attempting to access port 515 on the destination machine. The service that usually resides on port 515 is the printer spooler port. While it is possible that this is simple printer traffic, there are also known vulnerabilities in this service that would allow an attacker to gain root access to the system.

Traffic from this alert suggests that machine 10.10.98.153 has either been compromised or is being used for illegal purposes. While it may be allowed for internal systems to send print jobs to the printer spooler port of external systems, the traffic shows that more than this is happening.

```
05/02-00:07:52.161070 [**] connect to 515 from inside [**] 10.10.98.153:1023 -> 132.250.182.60:515
05/02-00:14:05.487473 [**] connect to 515 from inside [**] 10.10.98.153:1023 -> 132.250.182.60:515
05/02-00:22:41.109434 [**] connect to 515 from inside [**] 10.10.98.153:1023 -> 132.250.182.60:515
```

The traffic shows three different sessions between the source and destination over a matter of 15 minutes. The possibility that the same source and destination port pairs could be used for all three sessions is highly improbable. Also, looking at the address block for the destination indicates that this probably wasn't normal University traffic.

Query for 132.250.182.60: http://www.arin.net/cgi-bin/whois.pl?queryinput=132.250.182.60&B1=Submit+Query

```
Naval Research Laboratory (NET-NRL-NETS)
  4555 Overlook Avenue, SW
  Washington, DC 20375-5000
  US

  Netname: NRL-NETS
  Netblock: 132.250.0.0 - 132.250.255.255

  Coordinator:
    Dowling, Marian  (MD51-ARIN)  netgroup@NRL.NAVY.MIL
    (202)767-3903 (FAX) (202)767-2129

  Domain System inverse mapping provided by:

  NET.NRL.NAVY.MIL            132.250.1.131
  GRIZZLY.NRL.NAVY.MIL                  132.250.108.12

  Record last updated on 07-Jul-1995.
  Database last updated on  14-Nov-2001 19:54:47 EDT.
```

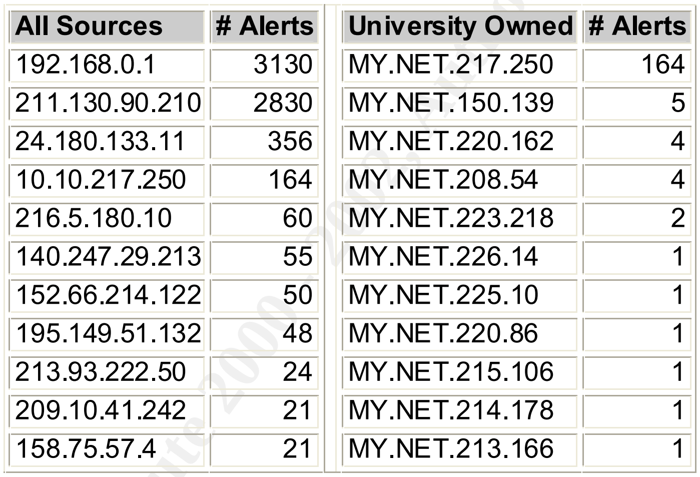
Defensive Recommendation for connect to 515 from inside

It is recommended that the University review its policies regarding access to the printer spooler port on machines outside of the University's network. It is recommended, though, that access be restricted to the local network only. If the University wishes to restrict access to the local network, then rules should be put in place on the University's routers and firewalls to block inbound and outbound traffic on port 515.

Out-Of-Spec File Analysis

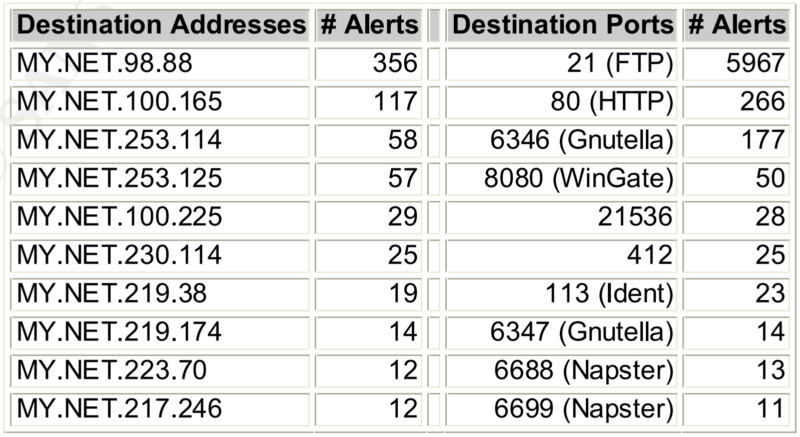
Out-Of-Spec (OOS) traffic is caused by TCP traffic that has an unusual combination of flags set. Most commonly this is one or the other of the high order flags, which are not used in normal TCP traffic, being set to a 1. This traffic can also be triggered by all of the low order bits being set to 1. OOS traffic should be considered anomalous, but shouldn't always be considered dangerous. Some OOS traffic may be caused by corruption of the packet during transit. Some OOS traffic may be due to someone trying to use the high order flag bits for their own programming purposes. Flagging OOS traffic helps to provide correlation to other intrusion detection logs.

The top 11 sources of OOS traffic from May 1st to May 7th are shown in the table below.



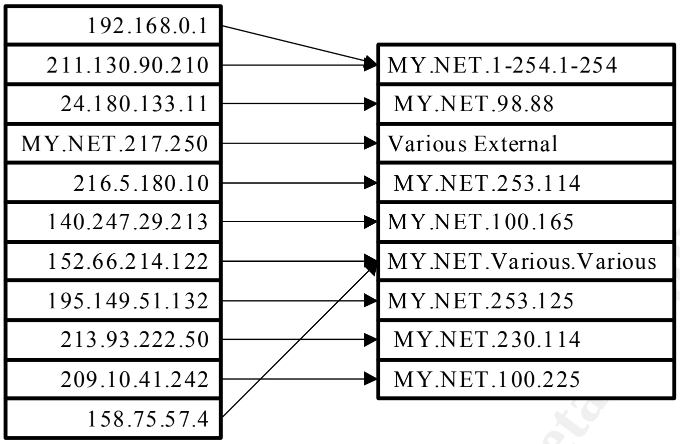
All Sources	# Alerts	University Owned	# Alerts
192.168.0.1	3130	MY.NET.217.250	164
211.130.90.210	2830	MY.NET.150.139	5
24.180.133.11	356	MY.NET.220.162	4
10.10.217.250	164	MY.NET.208.54	4
216.5.180.10	60	MY.NET.223.218	2
140.247.29.213	55	MY.NET.226.14	1
152.66.214.122	50	MY.NET.225.10	1
195.149.51.132	48	MY.NET.220.86	1
213.93.222.50	24	MY.NET.215.106	1
209.10.41.242	21	MY.NET.214.178	1
158.75.57.4	21	MY.NET.213.166	1

The top 10 destination addresses and ports of OOS traffic from May 1st to May 7th are shown in the table below.



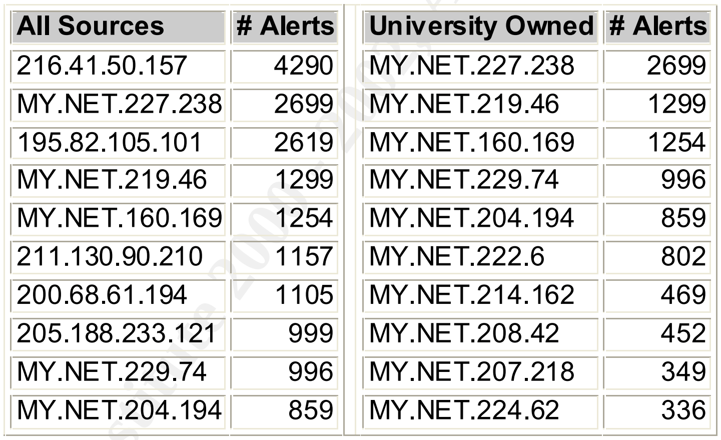
Destination Addresses	# Alerts	Destination Ports	# Alerts
MY.NET.98.88	356	21 (FTP)	5967
MY.NET.100.165	117	80 (HTTP)	266
MY.NET.253.114	58	6346 (Gnutella)	177
MY.NET.253.125	57	8080 (WinGate)	50
MY.NET.100.225	29	21536	28
MY.NET.230.114	25	412	25
MY.NET.219.38	19	113 (Ident)	23
MY.NET.219.174	14	6347 (Gnutella)	14
MY.NET.223.70	12	6688 (Napster)	13
MY.NET.217.246	12	6699 (Napster)	11

Below is a link graph that gives a graphic representation of the OOS top ten traffic.



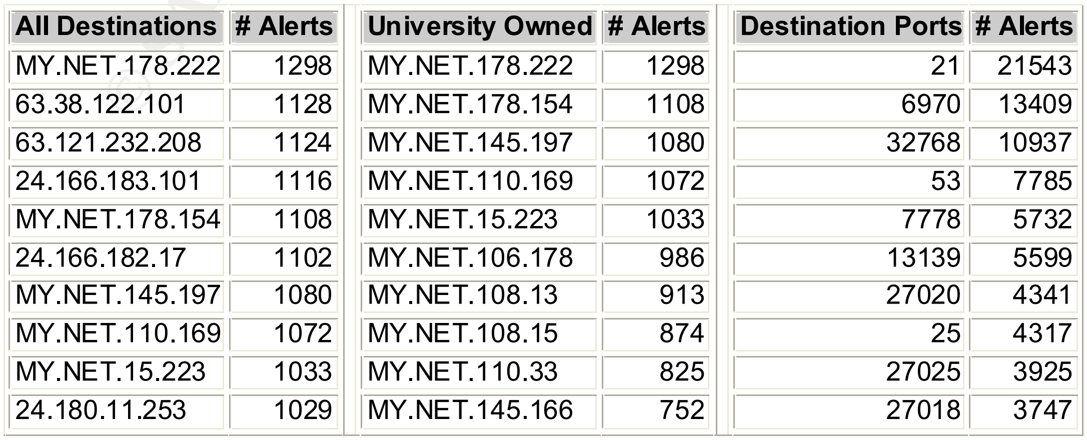
Scan File Analysis

The table below shows the top 10 sources of scans on the network from May 1st through May 7th.



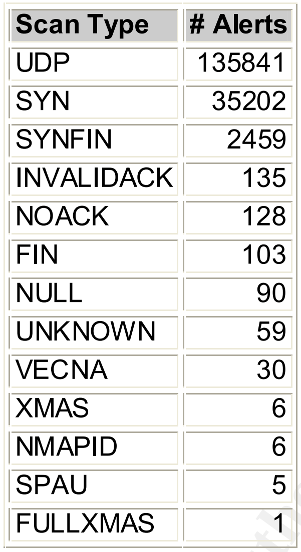
All Sources	# Alerts	University Owned	# Alerts
216.41.50.157	4290	MY.NET.227.238	2699
MY.NET.227.238	2699	MY.NET.219.46	1299
195.82.105.101	2619	MY.NET.160.169	1254
MY.NET.219.46	1299	MY.NET.229.74	996
MY.NET.160.169	1254	MY.NET.204.194	859
211.130.90.210	1157	MY.NET.222.6	802
200.68.61.194	1105	MY.NET.214.162	469
205.188.233.121	999	MY.NET.208.42	452
MY.NET.229.74	996	MY.NET.207.218	349
MY.NET.204.194	859	MY.NET.224.62	336

The table below shows the top 10 destination addresses and destination ports on the network from May 1st through May 7th.



All Destinations	# Alerts	University Owned	# Alerts	Destination Ports	# Alerts
MY.NET.178.222	1298	MY.NET.178.222	1298	21	21543
63.38.122.101	1128	MY.NET.178.154	1108	6970	13409
63.121.232.208	1124	MY.NET.145.197	1080	32768	10937
24.166.183.101	1116	MY.NET.110.169	1072	53	7785
MY.NET.178.154	1108	MY.NET.15.223	1033	7778	5732
24.166.182.17	1102	MY.NET.106.178	986	13139	5599
MY.NET.145.197	1080	MY.NET.108.13	913	27020	4341
MY.NET.110.169	1072	MY.NET.108.15	874	25	4317
MY.NET.15.223	1033	MY.NET.110.33	825	27025	3925
24.180.11.253	1029	MY.NET.145.166	752	27018	3747

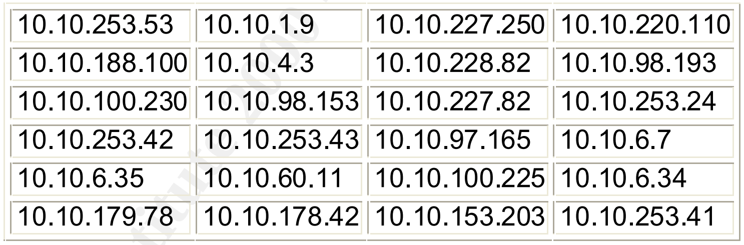
The table below shows the types of scans that were encountered on the network.



Scan Type	# Alerts
UDP	135841
SYN	35202
SYNFIN	2459
INVALIDACK	135
NOACK	128
FIN	103
NULL	90
UNKNOWN	59
VECNA	30
XMAS	6
NMAPID	6
SPAU	5
FULLXMAS	1

Possible Compromised Machines

The following table contains a list of the machines that are on the University's network that are probably compromised.



10.10.253.53	10.10.1.9	10.10.227.250	10.10.220.110
10.10.188.100	10.10.4.3	10.10.228.82	10.10.98.193
10.10.100.230	10.10.98.153	10.10.227.82	10.10.253.24
10.10.253.42	10.10.253.43	10.10.97.165	10.10.6.7
10.10.6.35	10.10.60.11	10.10.100.225	10.10.6.34
10.10.179.78	10.10.178.42	10.10.153.203	10.10.253.41

General Defensive Recommendations

It is recommended that the University should put procedures and policies in place that would provide the greatest protection for the University network, while at the same time allow for the greatest latitude in allowable behavior. I don't think that the University can reasonably expect to prevent such possible network security risks as network games, Gnutella or Napster, and ICQ or MS Messenger. What is important is to manage the risks that are known and eliminate the risks that are undesirable.

To this end, it is recommended that the University obtain and properly program routers that will only route University network traffic. These routers should not allow traffic from any network that is considered hostile or unfriendly, such as the networks that appear on the two watchlists. It would also be in the University's best interest to unsubscribe from all multicast networks unless there is a really good reason to belong to them.

The University should obtain and deploy firewalls at any connection from the local network to the outside world. These firewalls should block all inbound traffic from the external network with the following exceptions. The firewall should allow traffic in from the outside if the traffic is in response to a session started from the local network. The firewall should further only allow traffic on ports above 1024, except for the ports that become necessary to block, with the following exceptions. Ports that may be left open through the firewall are the SMTP port (25), the DNS port (53), and the HTTP port (80). Other ports under 1024 may be opened so long as there is a very good reason for them to be open.

The University should obtain and deploy a network intrusion detection system, and host based intrusion detection systems on each of the University's servers. These intrusion detection systems should be linked to the firewalls, and possibly the routers, to provide dynamic rules to block suspicious activity in near real time.

Finally, and perhaps most importantly, the University needs to make sure to keep all systems patched and current. This includes security and software patches for not only the operating systems but for all critical software on the machine, such as bios updates.

Analysis Process

When I first looked at the data set that was presented for this project I was a little taken aback thinking that I would have to analyze 3 ½ months worth of data. When I learned that I didn't have to do the entire data set, I first had to figure out what amount of data I was going to analyze. At first I thought that a months worth of data would be a good set. After an initial scan of several different months worth of data I realized that an average weeks worth of data would suffice. It was by random choice that I chose the first week in May.

Once I had chosen my data set I needed to get a "big picture" look at what I would be analyzing. For this purpose I began to look through the Internet to find tools that would help me with the data analysis. I finally chose two tools that would provide me with enough information to start. The tools were the perl script snort_stat and the program Snarf.

After I had used the above tools and knew what alerts and data I was looking at, it became clear that I needed an understanding of what the alerts were. I read through several previous practicals and looked through as many web sites as I could to get a good background on the traffic that was in the alerts.

I chose to break my analysis up by alert. As each alert presented itself I would analyze all of the traffic from most if not all of the sources, making sure to cross reference any source or destination that was in more than one alert. After I had finished analyzing one alert I would move on to the next one.

I then used some UNIX commands presented by Chris Baker in his practical to sort and analyze the OOS and Scan files.

Finally I worked on smoothing out the presentation of the paper including spelling, grammar, and font selection.

References

Asadoorian, Paul. GIAC Practical. URL: http://www.sans.org/y2k/practical/Paul_Asadoorian_GCIA.zip (15 November 2001).

Baker, Chris. GIAC Practical. URL: http://www.sans.org/y2k/practical/Chris_Baker_GCIA.zip (15 November 2001).

Bogle, Becky. GIAC Practical. URL: http://www.sans.org/y2k/practical/Becky_Bogle_GCIA.doc (15 November 2001).

Bruneau, Guy. GIAC Practical. URL: http://www.sans.org/y2k/practical/Guy_Bruneau.doc (15 November 2001).

Goodwin, PJ. GIAC Practical. URL: http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc (15 November 2001).