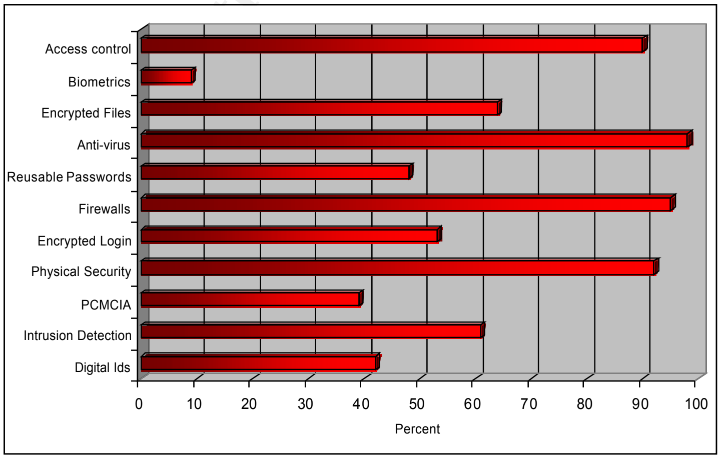
Billy Smith
GCIA Practical Assignment (v.3.1)
Network intrusion detection technologies aren't enough!

Part 1 – Describe the State of Intrusion Detection

Background

Over the past several years, there has been explosive growth in information technology due, in most part, to the Internet. Today, corporate networks are very complex. Much of this complexity is an indirect result of the Internet's rapid growth. The increased use of the Internet particularly by business has forced corporations to expand their information technology infrastructures significantly. As a result, information security incidents have grown at an even faster rate and are now a major concern globally.

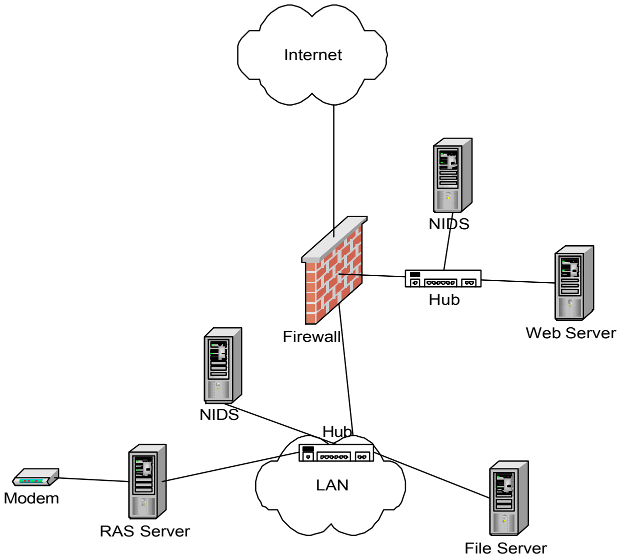
Information security incidents can be characterized as the lack of availability, integrity, and/or confidentiality of data. Software and hardware vendors have dedicated a tremendous amount of research and development resources towards insuring information availability, integrity and confidentiality. This research has led to the development of security devices such as firewalls, intrusion detection systems, strong authentication and access control mechanisms, virtual private networks and public key infrastructure. Organizations worldwide are implementing these technologies to prevent or detect an information security incident. The following chart from the Computer Security Institute/Federal Bureau of Investigation 2001 Computer Crime and Security Survey displays the prevalence of technologies among the companies surveyed [1].



Introduction

Most security devices and applications provide logging or alerting of known and possibly unknown security events that occur on an information technology infrastructure, but other operating systems and applications provide important details about the security of a corporate information technology infrastructure. These details include valid business applications, external attacks via the Internet, and internal attacks by employees.

As a part of the proliferation of security devices, network intrusion detection systems (NIDS) are the key technology that most organizations are using to monitor the security of their information assets. Generally, NIDS only detects malicious traffic that passes by on the network; therefore, many organizations do not have sufficient vision into their enterprise security. The purpose of this paper is to give arguments as to why NIDS technologies alone are not enough to truly monitor the security of most organizations. The following drawing will be used for illustrative purposes:



Argument #1 - Remote Access via Dial-in or Virtual Private Networks

Let's suppose the network above has users dialing in via a RAS server and the traffic to and from the RAS server is monitored using a NIDS. In this scenario, the NIDS will detect common security issues where trojans, worms or backdoor type applications are installed on a remote user's computer or laptop. In addition, the NIDS would catch malicious activity such as scanners that might be executed by as user that has successfully dialed in the RAS Server.

Let's assume a malicious user was dialing into the modem and guessing usernames and passwords. The NIDS would not detect this type of activity. If a malicious user found a valid username and password combination and performed valid activities such as mounting network drives or sending email, the NIDS would not detect this type of activity either. This would allow a malicious user to gain access to information without an organization knowing about it. When VPNs are used for remote access, this problem still exists.

One way to minimize the potential of this happening is through the use of strong authentication for remote access, but another way to protect against these types of threats is by monitoring the logs of the RAS server or VPN concentrator. By monitoring the logs of these servers, any excessive authentication failures can be closely analyzed for malicious activity. Even if strong authentication is used, the RAS server or VPN concentrator log monitoring provides an additional layer of security by allowing an organization to know that someone might be trying to get passed the strong authentication [2].

Argument #2 - Secure Shell

Let's suppose that the file server above has Secure Shell (SSH) listening and a disgruntled employee has SSH access to the file server and knows that the administrator uses SSH to managed the server. The disgruntled employee could SSH to the file server and use a program to try to "su root" and guess passwords. Obviously, this assumes that the normal user is allowed to "su root". Since SSH is encrypted traffic, the NIDS would not be capable of detecting these failed authentications. Also, a disgruntled employee could utilize the port tunneling capabilities of SSH to tunnel malicious connections to the file server to exploit other daemons listening on the file server. Again, the NIDS would not be aware of this activity due the SSH connection being encrypted. The only ways to detect these types of attacks would be to monitor the logs of the file server or use a host based IDS (HIDS) on the file server [3].

Argument #3 - Secure Sockets Layer (SSL) and Common Gateway Interface (CGI)

Let's suppose that the web server above uses SSL and CGI to perform some type of form transaction. Most serious attackers on the Internet try to exploit CGI on web servers. In the case where these CGI programs are access via SSL, NIDS is not able to detect this malicious activity due to the traffic being encrypted. Many NIDS are placed into DMZ where e-commerce servers reside, but many people forget that most e-commerce transactions occur over SSL. The only ways to detect this type of attack would be to monitor the logs of the web server or use a host based IDS (HIDS) on the web server.

Argument #4 – Human Element

Despite all our technological advances and the introduction of devices like firewalls and IDSs, companies' assets are being compromised every day. Many of these compromises remain unnoticed for several months or even years. One reason is simply because most companies do not utilize the information provided by their security devices.

In general, all IDS technologies provide little or no value if the human element is not applied to analyze the alerts and events generated. The reason being is that IDS do not provide an active role in protecting a network. Simply put for the sake of this argument, NIDS just watches the network ands send alerts when suspicious activity is detected. Most of the time, these alerts are in the form of Simple Mail Transfer Protocol (SMTP or email) or Simple Network Management Protocol (SNMP) Traps. If humans do not monitor the alerts, the value of a NIDS, or any IDS, is decreased tremendously [4].

Today, very few companies are monitoring events from their firewalls and network-based and host-based intrusion detection systems as well as the logs and alerts from their routers, switches, anti-virus and content scanning applications, backup applications, PBXs and critical Unix and NT servers including but not limited to web servers, FTP servers and mail servers.

Each device or application listed above can generate hundreds of lines of logs daily. A majority of the events logged are not security related so surveillance of specific security events is difficult and time consuming. For many administrators, reviewing these logs takes several hours a day and monitoring should be in real-time or near real-time so problems can receive a rapid response. For the typical system administrator, network administrator, and/or security officer, the task of reviewing logs is not a reality and monitoring events in real-time is impossible, day-to-day system maintenance demands too much time. Companies just do not have a 24 x 7 information technology staff so "off business hours" monitoring is nonexistent and internal and external hackers know this. Monitoring an entire security enterprise takes an experienced 24 x 7 staff of security analysts who have responsibility for continuously analyzing events so many companies are beginning to outsource this task to Managed Security Service Providers (MSSP)[5].

Conclusion

The arguments presented above are commonly overlooked when companies begin to investigate IDS solutions. Much of the oversight is due to the immaturity of the IDS and Managed Security Service Provider markets. The four arguments are only a small set of many examples as to why network based IDS technologies are not the complete answer. The first three arguments are technical in nature, but the last one is not. It is questionable whether or not information security will ever be solved with technology. The reason being is related to the same reason why effective automated buying and selling of stocks or investing in general has not been implemented in software. There are just too many variables and unknowns to mathematically model the problem to a level of accuracy that is accepted by the general population. Based on the number of variables and unknowns related to information security, I believe the human element will always be needed in the IDS and security monitoring space.

References

[1] Power, Richard. "CSI/FBI 2001 Computer Crime and Security Survey." Computer Security Trends & Issues. Volume VII, Number 1. Spring 2001.
[2] Schaefer, Norma Jean. "Knock Knock…Who's there? Do you know who is accessing your VPN?" 1 December 2001. URL: http://rr.sans.org/encryption/knock.php.
[3] Larrieu, Heather. "SSH and Intrusion Detection." 17 March 2002. URL: http://rr.sans.org/intrusion/SSH_ID.php.
[4] Brook, Jon-Michael C.. "Network IDS: To Tailor, or Not to Tailor." 6 March 2002. URL: http://rr.sans.org/intrusion/tailor.php.
[5] Lope-Wilkin, Esperanza. "Managed Security Services: an IDS solution." 20 May 2001. URL: http://rr.sans.org/intrusion/mss.php.

Part 2 – Network Detects

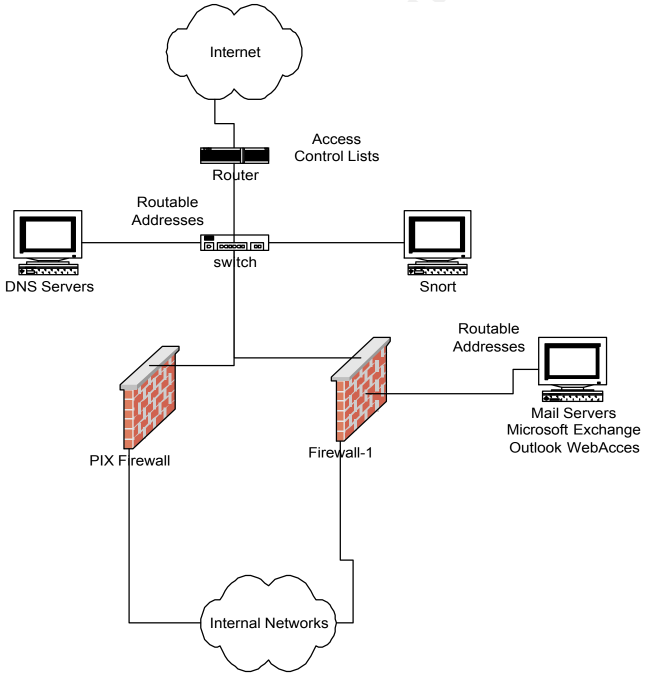
Detect #1

```
[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/28-19:37:29.411985 SAME.NET.66:80 -> VICTIM.HOST:137
TCP TTL:54 TOS:0x0 ID:5161 IpLen:20 DgmLen:40
***A**** Seq: 0x14F  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/28-19:37:19.417311 SAME.NET.34:80 -> VICTIM.HOST:137
TCP TTL:54 TOS:0x0 ID:5132 IpLen:20 DgmLen:40
***A**** Seq: 0x142  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/28-19:37:09.420750 SAME.NET.3:80 -> VICTIM.HOST:137
```

```
TCP TTL:54 TOS:0x0 ID:5102 IpLen:20 DgmLen:40
***A**** Seq: 0x138  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]
```

1. Source of Trace:

This trace came from an organization with large network and high volumes of inbound and outbound traffic. The network consists of a Snort sensor monitoring an Internet routable network that is screened from the Internet by access control lists on a Cisco router and is connected to a PIX firewall, DNS servers, and a Check Point Firewall-1. The Firewall-1 protects Microsoft Exchange email servers that also have Internet routable IP addresses. See diagram below:



2. Detect was generated by:

This detect was generated by a Snort 1.8.6 sensor running on Red Hat Linux.

This following rule trigger this alert:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap
TCP";flags:A;ack:0; reference:arachnids,28; classtype:attempted-recon; sid:628; rev:1;)
```

3. Probability the source address was spoofed:

The alerts in this trace are only an excerpt of many similar alerts that happened. Because this traffic is not associated with a connection, it is highly probable that most of the traffic associated with this trace is spoofed.

These three Snort alerts give significant evidence that these packets are spoofed. For example, it is extremely unlikely that within 20 seconds three separate hosts on the Internet would connect to the same host using destination port TCP 137 and source port TCP 80. In addition, the close incrementing packet ID numbers of 5102, 5132, and 5161 and close sequence numbers of 0x138, 0x142, and 0x14F give sufficient evidence that these packets were generated on the same host. Since they are the same amount of hops away from the destination host, the common TTL=54 gives an indication that these packets were generated on the same host. But, since these the packets have source addresses on same IP network, the TTL=54 indicator is not a strong argument.

On the other hand, since an attacker would desire response from this type of packet or stimulus, the intruder probably used his or her true IP address among a multitude of spoofed packets sent in order to potentially get a response.

4. Description of attack:

These traces are only excerpts from lots of this type of traffic. The packets generally have the following characteristics:

-Many source addresses
-Few destination addresses which are valid hosts on this organizations' network
-TCP
-Source port of 80
-Well-known destination ports such as 25, 53, 80, 137, and 443
-ACK bit set
-Strong spoofing indicators (i.e. packet ID, sequence numbers, time)

Many ingress filters allow HTTP return packets if the source port is 80 and the ACK bit is set. Someone is attempting information gathering or reconnaissance on this network by using packets that have a source port of TCP 80 and the ACK bit set to pass ingress access control lists on filtering routers. Once the packets pass the access control lists, these packets act as stimuli. Depending on the configuration of the egress access control lists on the filtering router, an attacker might be able to receive a response from these stimuli that would leak information about the operating systems and ports listening on the hosts being scanned. Due to the massive amounts of this type of traffic, it seems that this person might be spoofing traffic of this type to disguise themselves.

5. Attack mechanism:
This attack was most likely an attempt to gain information about specific hosts on this network that are protected by ingress access control lists on a filtering router. The attack mechanism utilizes the stimulus response characteristics within TCP/IP. It is very likely that an individual created the stimuli with the following nmap command:

```
nmap –sA –g 80 –D SAME.NET.34,SAME.NET.3,SAME.NET.66 –p 137
VICTIM.HOST
```

This command sends TCP packets from SAME.NET.34, SAME.NET.3, and SAME.NET.66 to VICTIM.HOST on destination port 137 with source ports of 80 and the ACK bit set. Since the access control lists are not restricting destination ports or keeping connection states, the filtering router passes these crafted packets.

Based on the information we have, it cannot be determine whether or not the attacker received any response from these stimuli. If the access control lists on the filtering router blocked the response, the attacker would learn that attacked host is protected by some type of packet filter. If the filtering router permitted the responses, the attacker would receive packets that would leak information such as window size and TTL that could indicate the operating system of the host being scanned. The attacker could then use this information to further exploit the host.

6. Correlations:

Several sources on the Internet document traffic that causes "SCAN nmap TCP" alerts. The archives at http://www.incidents.org have references to traffic generated by LinkProof, a load-balancing product by Radware. Several of the source IP addresses found in these Snort alerts have been recorded at http://www.dshield.org/ipinfo.php. Records were found for many of the most prevalent source IP addresses in these Snort alerts. This is not a surprise for a network with large amounts of outbound web browsing since many high volume web sites use these load balancers. But, many of these alerts were from packets not destined for proxy servers that the outbound web traffic would have been network address translated to. Based on these correlations and the facts presented above, it is very likely that an attacker used this type of traffic to disguise himself or herself among valid load balancer traffic. Also, some the common load balancers might have been used as decoys by the attacker as mentioned in the Spoofing and Attack Mechanism sections above.

7. Evidence of active targeting:

The volume of these Snort alerts and the spoofing indicators presented above give evidence of active targeting. Since many high volume web sites utilize load balancers, it is very likely that organizations with a lot of web browsing would see this type of probing. Even though this organization has a lot of web browsing, many of the packets that caused these Snort alerts were not destined for proxy servers that are used for all outbound web browsing. This simple fact gives a stronger argument that these packets were targeted.

8. Severity:
severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality	5	Scan targeted a system used as Internet SMTP server
Lethality	1	Scan was likely just information gathering
System Countermeasures	3	Unknown if NETBIOS Name Service is running
Network Countermeasures	1	Router access control lists do not deny these packets
Severity		(5+1) – (3+1) = 2

9. Defensive recommendation:

Restrict the destination ports to >1023 for return HTTP packets on the ingress access control lists. For a Cisco router, the configuration lines would be similar to:

```
access-list 100 permit tcp any eq 80 host VICTIM.HOST gt 1023 established
……several other permit lines……
access-list 100 deny ip any any log
```

Since the host scanned was an SMTP server, it is unlikely that this host needs to web browse at all. In general, this configuration line above could just be used to deny this scan to the firewalls or proxy servers that all browsing passes through.

10. Multiple choice test question:

When given a packet capture, which of the following sets of information give the best indications of spoofed packets?

Packet ID, Type Of Service, IP Header Length
Datagram Length, TCP Acknowledgement Numbers, TCP Sequence Numbers
Window, Time to Live, TCP Options
TCP Sequence Numbers, Packet ID, Time to Live
TCP Options, IP Header Length, Datagram Length

Answer: d

Detect #2

```
[**] [1:255:1] DNS zone transfer [**]
[Classification: Attempted Information Leak] [Priority: 3]
02/22-15:33:32.373155 ATTACKER.IP:42853 -> DNS.SVR.IP:53
TCP TTL:45 TOS:0x0 ID:53749 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0x966EE7A0  Ack: 0x3348DA67  Win: 0x4470  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS212]

............companydomain.com.....


[**] [1:255:5] DNS zone transfer [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/12-19:18:09.332969 ATTACKER.IP:42944 -> DNS.SVR.IP:53
TCP TTL:47 TOS:0x0 ID:3304 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0xF6FFCD74  Ack: 0x75750660  Win: 0x4470  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0532]
[Xref => http://www.whitehats.com/info/IDS212]

............companydomain.com.....


[**] [1:255:5] DNS zone transfer [**]
[Classification: Attempted Information Leak] [Priority: 2]
06/10-22:40:51.505269 ATTACKER.IP:37888 -> DNS.SVR.IP:53
TCP TTL:47 TOS:0x0 ID:24058 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0x95C8498  Ack: 0xB4CCF454  Win: 0x43F8  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0532]
[Xref => http://www.whitehats.com/info/IDS212]

............companydomain.com.....
```

1. Source of Trace:

These detects came from a network that has a DNS server and Snort IDS sensor on the Internet side of a Check Point Firewall-1. The DNS server is BIND 8.2.3 running on Mandrake 7.2 Linux, and the IDS sensor is Snort 1.8.1 running on Mandrake 7.2 Linux.

The Snort alerts and the DNS logs are monitored 24x7x365.

2. Detect was generated by:

This detect was generated by a Snort 1.8.1 sensor running on Mandrake 7.2 Linux. This version of Snort was compiled with a custom output plugin to print the ASCII decode of the payload portion of the packet that triggers an alert.

This following rule trigger this alert:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS zone transfer";
flags:A+;  content: "|00 00 FC|"; offset:13; reference:cve,CAN-1999-0532;
reference:arachnids,212; classtype:attempted-recon; sid:255; rev:5;)
```

3. Probability the source address was spoofed:

The ACK and PUSH bits are set in the packets the generated the alerts, and the TCP sequence and acknowledgement numbers appear to be valid. This indicates that a three-way handshake has already occurred between the attacking host and the DNS server. Since these packets were not source routed, the attacker could not be attempting a man-in-the-middle attack by spoofing these packets and routing the traffic back through his host. A man-in-the-middle attempt utilizing source routing is very difficult on the Internet because most Internet routers have source routing disabled. Since the attacker would definitely want to receive a response from this probe, it is unlikely that the source address was spoofed.

4. Description of attack:

The attacking host attempted a DNS zone transfer from the primary DNS server for the domain companydomain.com. Zone transfers are legitimate operations (AXFR) within DNS and normally occur when secondary DNS update records from a primary DNS server. Attackers attempt DNS zone transfers to map a company's network using the names and IP address information gained when successful.

The CVE information at http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0532 gives the following description for a DNS zone transfer: "A DNS server allows zone transfers". I disagree with this description and agree with Northcutt that this should be rejected as a part of the CVE list. DNS zone transfers performed between primary and secondary DNS servers are necessary and aren't necessarily malicious.

5. Attack mechanism:

Since the Domain Name System provides hostname to IP address mappings, attackers target DNS servers to gain information about the organization they are attacking. If this attack had been successful, the attacker might have gained information that could have been used to map the company's network. By mapping the network, an attacker might have been able to determine operating systems and applications used by the company. This information could have been used to further exploit the organization. An attacker could have used a command like the following to perform this probe for information:

```
dig @ns.companydomain.com axfr companydomain.com
```

This command attempts to perform a zone transfer for the domain "companydomain.com" from the DNS server "ns.companydomain.com".

6. Correlations:

The following log entries were captured from the DNS server ns.companydomain.com:

```
Feb 22 15:33:32 ns named[3621]: denied AXFR from [ATTACKER.IP].42853 for
"companydomain.com" (acl)
Apr 12 19:18:09 ns named[3621]: denied AXFR from [ATTACKER.IP].42944 for
"companydomain.com" (acl)
Jun 10 22:40:51 ns named[1347]: denied AXFR from [ATTACKER.IP].37888 for
"companydomain.com" (acl)
```

The times of these syslog entries correspond to the timestamps on the Snort alerts in this detect. Access control lists within BIND have been configured on this nameserver. These lines show that ATTACKER.IP attempted a zone transfer, but the access control lists denied the attempt.

Since zone transfers are valid DNS operations, the information at http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0532 will probably never get approved as a CVE entry. Searches of http://www.incidents.org don't give strong indications that there are any common false positives for this detect.

7. Evidence of active targeting:

There is sufficient evidence that this attack is targeted at the organization that owns "companydomain.com". The attacker attempted to perform a zone transfer for the domain "companydomain.com" from the DNS server "ns.companydomain.com" that is primary for this domain. This is strong evidence that this attack was targeted.

8. Severity:
severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality	5	External DNS is critical to public mail, www, ftp servers
Lethality	2	Attempted information gathering that could provide useful information to the attacker
System Countermeasures	5	Access control lists don’t allow zone transfers
Network Countermeasures	1	No network countermeasures blocked this traffic to the DNS server
Severity		(5 + 2) – (5 + 1) = 1

9. Defensive recommendation:

The DNS server has access control lists that deny zone transfer, but TCP packets with destination port 53 are still allowed to the server. Router access control lists could be used to deny TCP packets destined for port 53 if zone transfers are not needed or restrict these transfers to authorized secondary DNS servers. As always, the latest version of BIND should be running on the DNS server.

10. Multiple choice test question:

What protocol and port is used for DNS zone transfers?

UDP port 53
TCP port 137
UDP port 137
TCP port 53

Answer: d

Detect #3

```
Jun 16 18:33:58 fw kernel: iptables: INPUT(drop) IN=eth1 OUT=
MAC=00:a0:c9:a9:16:45:00:30:19:47:e5:38:08:00 SRC=ATTACKER.IP
DST=FIREWALL.IP LEN=35 TOS=0x00 PREC=0x00 TTL=108 ID=5355 PROTO=UDP
SPT=1024 DPT=4000 LEN=15

Jun 16 18:33:58 fw kernel: iptables: INPUT(drop) IN=eth1 OUT=
MAC=00:a0:c9:a9:16:45:00:30:19:47:e5:38:08:00 SRC=ATTACKER.IP
DST=FIREWALL.IP LEN=35 TOS=0x00 PREC=0x00 TTL=108 ID=5611 PROTO=UDP
SPT=1024 DPT=4001 LEN=15

Jun 16 18:33:58 fw kernel: iptables: INPUT(drop) IN=eth1 OUT=
MAC=00:a0:c9:a9:16:45:00:30:19:47:e5:38:08:00 SRC=ATTACKER.IP
DST=FIREWALL.IP LEN=35 TOS=0x00 PREC=0x00 TTL=108 ID=5867 PROTO=UDP
SPT=1024 DPT=4002 LEN=15

Jun 16 18:33:58 fw kernel: iptables: INPUT(drop) IN=eth1 OUT=
MAC=00:a0:c9:a9:16:45:00:30:19:47:e5:38:08:00 SRC=ATTACKER.IP
DST=FIREWALL.IP LEN=35 TOS=0x00 PREC=0x00 TTL=108 ID=6123 PROTO=UDP
SPT=1024 DPT=4003 LEN=15
```

1. Source of Trace:

This source of this detect is a network connected directly to the Internet via an iptables firewall and cable modem.

2. Detect was generated by:

This detect was generated by an iptables v1.2.2 firewall running on Mandrake 8.1 Linux. The following iptables chain drop and logged this detect:

```
iptables -A INPUT -i eth1 -j LOG --log-level warning --log-prefix "iptables: INPUT(drop) "
```

3. Probability the source address was spoofed:

There are some good indicators such as timestamps, common UDP source port and TTLs and the close incrementing packet Ids that give evidence that the packets were generated on the same host. Since an attacker would have desired responses such as ICMP port unreachable packets to these stimuli, it is unlikely that the source address was spoofed.

4. Description of attack:

This detect seems to be an information gathering attempt using a UDP port scan for ports 4000-4003. The packets had the following characteristics:

Packets were logged at the same time
Source and destination IP addresses are constant

Common UDP source port 1024
Incrementing UDP destination ports from 4000-4003
Common TTL values
Closely incrementing packet IDs

The attacker seemed to be checking for the services listening on UDP ports 4000-4003. ICQ servers normally listen on UDP port 4000.

5. Attack mechanism:

Using a port scan, an attacker tries to gain insight about the services that are listening on a server. Once this information is gain, an attacker can attempt to exploit these servers. TCP and UDP port scans utilize stimulus-response. UDP port scans are very unreliable due to the fact that UDP is a connectionless protocol meaning that there is no indication that a connection has been established. Two primary methods used for UDP port scanning are:
Send data to a UDP port and wait for a response from that port.
Send data to a UDP port and wait for an ICMP port unreachable message, indicating that this port is NOT active.

The port scan information above and more can be found at http://hq.mcafeeasap.com/vulnerabilities/vuln_data/21000.asp

6. Correlations:

In November 2001, references to similar scans were posted at http://www.incidents.org/archives/intrusions/msg01726.html. The author states that most of the scans were sourced from China and posted the following information:

Date	SourceIP	4000	4001	4002	4003
10-23	61.182.241.77	64	1	2	1
10-25	202.110.163.108	20	20	20	19
10-25	61.134.228.232		1	18	
10-26	61.167.249.201	233	233	233	248
10-27	211.97.183.67	425	238	215	200
10-28	61.182.251.89	31	13	8	16
10-29	61.182.40.85	160	159	167	159
10-30	202.111.161.129	12	9	10	13
11-01	61.184.166.11	255	190	74	75
11-04	61.180.215.2	246	246	245	245

11-09	61.156.112.13	23			
11-09	61.180.188.54	162	164	164	159
11-10	210.51.226.250	121	134	124	129

It is interesting to note that six to seven months later, ATTACKER.IP, an IP address from China, performs the same type of scan.

7. Evidence of active targeting:

There is not strong evidence of targeting in this detect. The only evidence that might give an indication of targeting is the fact that four packets with close incrementing destination ports (i.e. 4000-4003) were sent FIREWALL.IP at the same time. But, the information provided in the correlations section above gives a firm argument that this detect was a part of a broad scan.

8. Severity:
severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Criticality	5	Firewall that provides Internet connectivity
Lethality	1	Attempted information gathering that could provide useful information to the attacker
System Countermeasures	5	Iptables and Snort installed and all services disabled
Network Countermeasures	5	Iptables blocks all traffic destined for the firewall
Severity		(5 + 1) – (5 + 5) = -4

9. Defensive recommendation:

Blocking ICMP port unreachable packets, ICMP Type 3 Code 3, sourced from the firewall would help with information leak.

10. Multiple choice test question:

If UDP port scan is conducted on a host can be pinged, what type of response packets would be expected?

TCP unserved port
UDP unserved port
ICMP Host Unreachable
ICMP Port Unreachable
ICMP Protocol Unreachable

Answer: d

Part 3 – Analyze This

Executive Summary

The purpose document is to provide analysis of the university's Intrusion Detection System (IDS) logs for five consecutive days. In general, IDS produces a lot of information, and much of this information requires human analysis and research. Since the only information provided is the IDS logs, it is difficult to determine exactly what is happening in many cases. As a result, this was an intense process that required significant amount of time and energy to complete.

Analysis of these logs gives fairly thorough view of the network traffic at the university. This view is primarily a result of many of the applications being used throughout the network generating alerts that are false alarms. Some of these applications are DNS, NTP, AFS, WINS, Symantec Ghost, eDonkey2000, Gnutella, Microsoft Network Games, and KaZaa. Even though the data consisted of many false alarms, there are several interesting alarms and communications that should be looked into by the university staff. The remaining paragraphs provide the details of the findings including the false alarms and the alarms that need university staff attention.

Data Analyzed

The data analyzed was Snort Intrusion Detection System logs from March 27, 2002 through March 31, 2002. The following Scans, Alerts, and OOS files were analyzed:

scans.020327.gz
scans.020328.gz
scans.020329.gz
scans.020330.gz
scans.020331.gz
Total line count: 1,754,776

alert.020327.gz
alert.020328.gz
alert.020329.gz
alert.020330.gz
alert.020331.gz
Total line count: 745,090

oos_Mar.27.2002.gz
oos_Mar.28.2002.gz
oos_Mar.29.2002.gz
oos_Mar.30.2002.gz
oos_Mar.31.2002.gz
Total line count: 298

Process

The analysis process used is primarily based on prior experiences with the Snort Intrusion Detection System. The process consists of three steps consisting of Scan Data Analysis, Alert Data Analysis, and Out of Spec (OOS) Data Analysis.

First, the scan data is analyzed. The scan data is a result of the number of connections from a host within a time frame exceeding the port scan thresholds in the Snort configuration. Analysis of the scan data gives indications of critical servers on the network and services running on these servers. Servers that have many connections normally produce a lot of port scan data that is considered false positives. The scan data also gives clues of information gathering attempts that may be performed on the network using scanning tools such as Nmap and Nessus.

Second, analysis of the alert data is performed. The alert data gives information of packets or connections that have triggered a Snort signature or met some threshold or condition of a preprocessor. The alert data includes port scan alerts as a result of scan data. These port scan alerts where excluded from my alert data analysis. The data provides information of potentially malicious traffic on the network.

Finally, the OOS data analysis is done. The OOS data is packet level information on packets that are out of specification. This information can normally be correlated to the scan and alert data.

Scan Data Analysis

Scans Top Talkers

The criteria used for the Scans Top Talkers was the most prevalent "IP address:port" combinations. Based on my prior experience with Snort, I knew that the Scans files would most likely have a significant amount of false alarms due to heavily loaded servers. Using this knowledge and the criteria chosen allowed me to quickly determine key servers and what services were running on these servers.

```
  Count         IP address:port
----------      ------------------------------
 338709         MY.NET.11.8:1347
 336364         MY.NET.60.43:123
 135193         MY.NET.150.143:1057
 110900         MY.NET.150.113:1257
   38906        MY.NET.60.43:7000
   38418        MY.NET.1.3:53
   36007        MY.NET.6.45:7000
   26497        MY.NET.5.55:137
   19932        MY.NET.1.4:53
   18308        MY.NET.150.143:28800
```

Analysis of Scans Top Ten Talkers

MY.NET.11.8:1347

Initial research on UDP 1347 led me to believe that MY.NET.11.8 is a server running a Multi Media Conferencing application developed by BBN. After further analysis of the data in the scans files, I noticed scan data involving MY.NET.11.8:1347 looked like:

MY.NET.11.8:1347 -> MY.NET.152.x:1346 UDP

Research on UDP 1346 showed that Alta Analytics License Manager used this port, but I found very little information regarding Alta Analytics License Manager. As a result, I decided to analyze the scan file data in more detail. Finally, I searched the scan data for "MY.NET.152.x:1346" but excluded data involving MY.NET.11.8. This search revealed the scan data as follows:

MY.NET.152.158:1346 -> 229.55.150.208:1345 UDP

This was the clue that really led me in the right direction. I researched UDP 1345 and found the links:

http://lists.insecure.org/incidents/2000/Nov/0162.html mentioning Norton Ghost Client and the multicast address 229.55.150.208

and

http://service2.symantec.com/SUPPORT/ghost.nsf/docid/1999033015222425 describing how Ghost multicasting communicates over the network.

Based on this research, there are strong indications that MY.NET.11.8 is a Ghost Enterprise Console using an RML port of UDP 1347 and much of the scan data is a result of Ghost Enterprise Console communicating to the Ghost Clients.

MY.NET.60.43:123

MY.NET.60.43 is running a Network Time Protocol (NTP) server on UDP port 123. Many NTP clients are synchronizing time from MY.NET.60.43. As a result, MY.NET.60.43's responses to these client requests are triggering the Snort port scan thresholds.

MY.NET.150.143:1057

StarTron, a 3D Internet action game, is commonly associated with UDP port 1057. http://www.startron.org/support.html states that StarTron uses UDP port 1057 and TCP port 6112. After analyzing the scan data that included MY.NET.150.143 and UDP 1057, I found that a lot of this scan data was similar to:

```
MY.NET.150.143:1057 -> w.x.y.z:4665 UDP
```

After closer analysis, I also determined that UDP port 4665 seemed to be significant due to the massive amounts of scan data that involving UDP port 4665. I researched UDP port 4665 and found http://www.edonkey2000.com/faq.html#port which states that TCP port 4661 and 4662 and UDP port 4665 are the default ports used by eDonkey2000. eDonkey2000 is an Internet file sharing application and network that works much like Napster. Using these clues, I analyzed the scan data and discover data like:

```
MY.NET.150.143:4526 -> w.x.y.z:4662 SYN ******S*
MY.NET.150.143:4661 -> z.y.x.w:4662 SYN ******S*
MY.NET.150.143:1053 -> y.w.x.z:4665 UDP
```

Based on this data, MY.NET.150.143 seems to be a part of "The Donkey Network" described at http://www.thedonkeynetwork.com due to the common uses of TCP ports 4661 and 4662 and UDP port 4665. By being a part of this network, MY.NET.150.143 sends messages to many other servers on UDP port 4665. These messages trigger the port scan thresholds in Snort and causes these scan alarms.

After closer analysis, I noticed that the original clue of UDP source port 1057 was not common characteristic of the eDonkey2000 traffic. Utilizing this information, I viewed the scan data looking for MY.NET.150.143 where the source port was UDP 1057 and destination port was not UDP 4665. As a result, I found that there was a lot scan data having UDP source port 1057 and variety of destination ports greater than 1023:

```
MY.NET.150.143:1257 -> x.y.w.z:6665 UDP
MY.NET.150.143:1257 -> y.x.w.z:7665 UDP
MY.NET.150.143:1257 -> y.x.w.z:10002 UDP
```

Considering these facts, MY.NET.150.143 is likely running a service such as StarTron on UDP port 1057 that is utilized heavily or UDP port 1057 is associated with eDonkey2000 server application on MY.NET.150.143.

MY.NET.150.113:1257

http://www.iana.org/assignments/port-numbers has UDP 1257 listed as Macromedia's Shockwave 2. After some analysis, I found scan data like:

```
MY.NET.150.113:2486 -> w.x.y.z:4665 UDP
MY.NET.150.113:1257 -> w.x.y.z:4665 UDP
MY.NET.150.113:4387 -> z.y.x.w:4662 SYN ******S*
```

This is an indicator that MY.NET.150.113 is a member of "The Donkey Network" as well. After digging a little more, I analyzed scan data with UDP source port 1257 much like the scan data for MY.NET.150.143 with UDP source port 1057:

```
MY.NET.150.113:1257 -> a.b.c.d:2004 UDP
MY.NET.150.113:1257 -> e.f.g.h:6665 UDP
MY.NET.150.113:1257 -> h.i.j.k:8665 UDP
```

Similar to MY.NET.150.143, MY.NET.150.113 is likely running a service such as StarTron on UDP port 1057 that is utilized heavily or UDP port 1057 is associated with eDonkey2000 server application on MY.NET.150.143. Since the common destination ports of 6665, 7665, and 8665 are just one digit different than 4665, I believe it is more likely that the UDP ports 1057 and 1257 are associated with the eDonkey2000 server than with StarTron.

MY.NET.60.43:7000

MY.NET.60.43 is an AFS file server listening on UDP 7000. IBM's AFS documentation located at http://www-124.ibm.com/developerworks/opensource/afs/docs/html/ describes AFS as "a distributed file system that enables users to share and access all of the files stored in a network of computers as easily as they access the files stored on their local machines." http://www.faqs.org/faqs/afs-faq/ also provides a lot of good information about AFS. MY.NET.60.43 responses to many AFS client requests on UDP 7000 are causing the Snort port scan alarms to be triggered. The scan data involving MY.NET.60.43 has one of the following forms:

```
        MY.NET.60.43:7000 -> w.x.y.z:7001 UDP
Or
        z.y.x.w:7001 -> MY.NET.60.43:7000 UDP
```

AFS uses several ports. Here is an excerpt from an /etc/services file:

```
afs3-fileserver 7000/udp    # file server itself
afs3-callback   7001/udp    # callbacks to cache managers
afs3-prserver   7002/udp    # users & groups database
afs3-vlserver   7003/udp    # volume location database
afs3-kaserver   7004/udp    # AFS/Kerberos authentication service
afs3-volser     7005/udp    # volume management server
afs3-errors     7006/udp    # error interpretation service
afs3-bos        7007/udp    # basic overseer process
afs3-update     7008/udp    # server-to-server updater
afs3-rmtsys     7009/udp    # remote cache manager service
```

MY.NET.1.3:53

MY.NET.1.3 is running a Domain Name Service (DNS) server on UDP port 53. Many clients are using MY.NET.1.3 for name resolution. As a result, MY.NET.1.3's responses to these client requests are triggering the Snort port scan thresholds.

MY.NET.6.45:7000

MY.NET.6.45 is an AFS file server just as MY.NET.60.43. As MY.NET.60.43 above, the scan data involving MY.NET.6.45 is of the form:

```
        MY.NET.6.45:7000 -> w.x.y.z:7001 UDP
Or
```

```
z.y.x.w:7001 -> MY.NET.6.45:7000 UDP
```

After finding the entries for AFS in the /etc/services, I found that the connections in involving UDP 7000 and 7001 was communication between the AFS file servers MY.NET.6.45 and MY.NET.60.43 and the Cache Manager residing on the AFS client machines.

After analyzing the scan data more closely, I found other servers on MY.NET related to AFS that are producing significant amounts of scan data. MY.NET.151.70 and MY.NET.153.197 are users and groups databases using UDP port 7002. MY.NET.1.13, MY.NET.60.12, and MY.NET.6.33 are volume location databases using port UDP 7003. MY.NET.151.70 may be an AFS/Kerberos authentication server using UDP port 7004, but all of the UDP port 7004 scan data has the following form:

```
MY.NET.150.113:1257 -> 213.20.228.162:7004
```

MY.NET.151.70 might be a server-to-server updater using UDP 7008 producing scan data like:

```
MY.NET.151.70:7008 -> 205.188.228.145:15368 UDP
```

MY.NET.5.55:137

http://www.microsoft.com/windows2000/techinfo/reskit/samplechapters/cnfc/cnfc_por_simw.asp and common /etc/services files describes UDP port 137 as NetBIOS over TCP/IP name service. Most of the scan data involving MY.NET.5.55 has one of the following forms:

```
MY.NET.w.x:137 -> MY.NET.5.55:137 UDP
```

or

```
MY.NET.5.55:137 -> MY.NET.y.z:137 UDP
```

This is strong evidence that MY.NET.5.55 is a WINS server. WINS servers provide a way for Windows computer names to be mapped to IP addresses. When someone uses the Windows network browsing capability, their computer commonly uses a WINS server to translate computer names to IP addresses. Just as with DNS servers, WINS servers responding to many client requests exceeds the Snort port scan thresholds.

While analyzing the data for MY.NET.5.55, I found other hosts related to the Windows network as well as other interesting hosts. MY.NET.5.50 seems to be another WINS server. MY.NET.11.5, MY.NET.11.6, and MY.NET.11.7 appear to be Domain Controllers or Windows file servers. The following excerpts occurred frequently in the logs and are most likely due to Windows network logons or browsing:

```
00:15:37 MY.NET.w.x:137 -> MY.NET.5.55:137 UDP
00:15:37 MY.NET.w.x:137 -> MY.NET.5.50:137 UDP
00:15:40 MY.NET.w.x:137 -> MY.NET.11.6:137 UDP
00:15:40 MY.NET.w.x:1228 -> MY.NET.11.6:139 SYN ******S*
```

or

```
00:19:35 MY.NET.y.z:137 -> MY.NET.5.55:137 UDP
00:19:35 MY.NET.y.z:137 -> MY.NET.5.50:137 UDP
00:19:37 MY.NET.y.z:137 -> MY.NET.11.7:137 UDP
00:19:37 MY.NET.y.z:4028 -> MY.NET.11.7:139 SYN ******S*
```

or

```
10:16:39 MY.NET.a.b:137 -> MY.NET.5.55:137 UDP
10:16:39 MY.NET.a.b:137 -> MY.NET.5.50:137 UDP
10:16:41 MY.NET.a.b:137 -> MY.NET.11.5:137 UDP
10:16:41 MY.NET.a.b:4286 -> MY.NET.11.5:139 SYN ******S*
```

Hosts on the MY.NET.153 network seem to be trying to utilized WINS lookups from Windows NT DNS servers. These request attempts generate scan data as follows:

```
MY.NET.153.x:137 -> MY.NET.1.4:53 UDP
```

or

```
MY.NET.153.y:137 -> MY.NET.1.5:53 UDP
```

How WINS lookup works from Windows NT DNS is documented at http://support.microsoft.com/default.aspx?scid=kb;EN-US;q173161.

MY.NET.1.4:53

Just as with MY.NET.1.3, MY.NET.1.4 is running a Domain Name Service (DNS) server on UDP port 53. Many clients are using MY.NET.1.4 for name resolution. As a result, MY.NET.1.4's responses to these client requests are triggering the Snort port scan thresholds.

MY.NET.150.143:28800

All of the scan data involving UDP port 28800 has the on of the following forms:

```
        MY.NET.150.143:28800 -> w.x.y.z:28800 UDP
```

Or

```
        MY.NET.150.143:28800 -> a.b.c.d:1169 UDP
```

My research at http://support.microsoft.com/support/kb/articles/Q159/0/31.asp provides information that UDP port 28800 is associated with Microsoft Network Games that are found at http://zone.msn.com. Nslookup shows that this network has the IP range of 207.46.203.x. See below:

```
# nslookup zone.msn.com

Server:        205.218.98.2

Address:       205.218.98.2#53

Name:   zone.msn.com

Address: 207.46.203.12
```

Through further analysis of the scan data, I found scan data supports the idea that MY.NET.150.143 is involved in MSN Games. The following data is excerpts from the scan data where MY.NET.150.143 is connecting http://zone.msn.com:

```
        MY.NET.150.143:4109 -> 207.46.203.12:80 SYN ******S*
        MY.NET.150.143:4126 -> 207.46.203.96:28801 SYN ******S*
        MY.NET.150.143:4127 -> 207.46.203.19:28808 SYN ******S*
        MY.NET.150.143:4130 -> 207.46.203.22:28808 SYN ******S*
        MY.NET.150.143:4238 -> 207.46.203.23:28807 SYN ******S*
        MY.NET.153.153:4401 -> 207.46.203.50:80 SYN ******S*
        MY.NET.150.246:4502 -> 207.46.203.19:28838 SYN ******S*
```

MY.NET.150.143 is initiating or receiving many connections to or from http://zone.msn.com and other hosts participating in the games. Just like much of the other Top Ten Talkers in the scan data, MY.NET.150.143 is exceeding the Snort port scan thresholds.

While analyzing the scan data for the Top Ten Talkers, I found most of the scan alarms to be associated with valid applications being used across the network. In this analysis process, I discovered data other than the Top Talkers to be worthy of discussion.

Other Scan Data Analysis

On March 27, 2002, MY.NET.70.234 performed a port scan of TCP ports 1-6000 on MY.NET.150.114. This activity began at 09:31:07 with TCP port 1 resulting in the following scan log:

```
MY.NET.70.234:1355 -> MY.NET.150.114:1 SYN ******S*
```

And ended at 09:36:02 with TCP port 6000 generating the following scan log:

```
MY.NET.70.234:3439 -> MY.NET.150.114:6000 SYN ******S*
```

On March 28, MY.NET.151.71 performed many HTTP connections to servers outside of MY.NET and DNS connections to MY.NET.1.3 that triggered the Snort port scan thresholds. I began looking into MY.NET.151.171 because I discovered the following interesting scan data:

```
MY.NET.151.71:137 -> 209.202.218.12:137 UDP
```

I found UDP port 137 connections to an IP external to MY.NET to be unusual. After analyzing this further, MY.NET.151.71 might have been infected with Nimda or Code Red causing all the HTTP and DNS connections. The scan data involving MY.NET.151.71 ceased around 1440 on March 28.

MY.NET.70.177 also seem to be performing a lot of host and port scanning of the MY.NET.5 network. Most of this scanning involved UDP 161 indicating that MY.NET.70.177 might be a Simple Network Management Protocol (SNMP) Management station performing a discovery of MY.NET.5. TCP ports 111 and 135 were also common ports used in this scanning.

The data included a lot of scan information for TCP ports 80 and 8080's from hosts on MY.NET.5 to the external network 211.233.79.x. ARIN shows the following:

```
# whois -h whois.arin.net 211.233.79

Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
```

APNIC

AU

Netname: APNIC-CIDR-BLK2
Netblock: 210.0.0.0 - 211.255.255.255

These hosts on MY.NET.153 might be infected with Nimda or Code Red.

Within the scan data, I found the following to be interesting as well:

a.b.c.d:0 -> MY.NET.x.y:0 UDP

Some research on the web indicates that these scan alerts might have been caused by fragmented packets. Fragmented packets can be in IDS evasive techniques.

On March 31 around 2012, MY.NET.88.223 began attempting to connect to IP addresses external to MY.NET on TCP port 6346. This port is commonly associated with a fully-distributed information-sharing technology called Gnutella. I found reference to this at http://www.gnutellanews.com/information/what_is_gnutella.shtml and http://www.gnutella.com.

Throughout the scan data, MY.NET.6.49 and MY.NET.6.50 are the source addresses in many scans where the source and destination ports are high number UDP ports that are normally greater than 10000.

While analyzing the scan data, MY.NET.150.143 seemed to have a lot of HTTP connections to IP addresses external to MY.NET that triggered the port scan thresholds of Snort. This is an indicator that MY.NET.150.143 might have been infected with Nimda or Code Red.

MY.NET.150.143 creates scan data while trying to connect to MY.NET.150.1 on UDP port 1900. UDP port 1900 is associated with Universal Plug and Play (UPnP). A vulnerability with UPnP is documented in a Microsoft Security Bulletin http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms01-059.asp. MY.NET.150.113, MY.NET.60.43, and hosts on MY.NET.88 also create scan data such as this.

Alert Data Analysis

Alert Top Talkers

The most prevalent "IP address" disregarding whether it was a source or destination was the criteria used for the Alert Top Talkers. Also, since the scan data has already been analyzed, all port scan alerts were excluded from the alerts analysis.

Count	IP address
49173	MY.NET.150.198
44897	MY.NET.11.6
23626	MY.NET.11.7
20905	MY.NET.153.197
19523	MY.NET.70.177
12636	211.115.212.150
7364	MY.NET.150.195
7116	MY.NET.153.203
5175	MY.NET.152.19
5105	MY.NET.153.119

Analysis of Alert Top Ten Talkers

This analysis will consist of an explanation of each unique alert for each of the Alert Top Ten Talkers.

MY.NET.150.198

Count	Alert Summary
44298	connect to 515 from inside
4872	SNMP public access
2	NMAP TCP ping!
1	ICMP Echo Request Nmap or HPING2

connect to 515 from inside
Someone created a custom Snort signature to detect print spooler connections on TCP port 515 initiated from MY.NET. MY.NET.150.198 appears to be running a print spooler (lpd) that is being used by other hosts. When hosts connect to the spooler, the connection triggers this Snort alarm.

SNMP public access
MY.NET.150.198 seems to be polling hosts on MY.NET for SNMP agents listen on UDP 161 with a public community string. Public is the default community string from many

SNMP agents and poses a well-know security threat. MY.NET.150.198 could be an SNMP management station or someone is using MY.NET.150.198 to perform malicious information gathering.

NMAP TCP ping!
MY.NET.253.10 attempted a connection to TCP port 6112 on MY.NET.150.198. TCP port 6112 is commonly associated with the UNIX-based Common Desktop Environment (CDE) or a game called FSGS. See http://isc.incidents.org/port_details.html?port=6112. Since MY.NET.150.198 seems to be running lpd that is commonly associated with UNIX platforms, I suspect that MY.NET.253.10 is using NMAP to check for TCP port 6112 listening on MY.NET.150.198.

ICMP Echo Request Nmap or HPING2
MY.NET.253.10 pings MY.NET.150.198.

MY.NET.11.6

Count	Alert Summary
28498	SMB Name Wildcard
14280	ICMP Echo Request L3retriever Ping
2119	ICMP Echo Request Nmap or HPING2

SMB Name Wildcard
As determined in the scan data analysis above, MY.NET.11.6 is a Windows server(most likely a Domain Controller) that listens on UDP port 137. This alert is likely a result of a network logon to this server or network browsing.

ICMP Echo Request L3retriever Ping
This event may indicate that someone is scanning your network using the L3 "Retriever 1.5" security scanner. This legitimate security tool is for authorized security assessment and should not be used on unauthorized networks. Many hosts on MY.NET.152 talking to MY.NET.11.6 cause these alerts.
http://www.digitaltrust.it/arachnids/IDS311/event.html makes reference that plain Windows 2000 boxes talking to Windows 2000 Domain Controllers causes false positives with this signature.

ICMP Echo Request Nmap or HPING2
Many hosts on MY.NET.152 are pinging MY.NET.11.6.

MY.NET.11.7

Count	Alert Summary
14872	SMB Name Wildcard
7453	ICMP Echo Request L3retriever Ping
1301	ICMP Echo Request Nmap or HPING2

SMB Name Wildcard
As determined in the scan data analysis above, MY.NET.11.7 is a Windows server(most likely a Domain Controller) that listens on UDP port 137. This alert is likely a result of a network logon to this server or network browsing.

ICMP Echo Request L3retriever Ping
This event may indicate that someone is scanning your network using the L3 "Retriever 1.5" security scanner. This legitimate security tool is for authorized security assessment and should not be used on unauthorized networks. Many hosts on MY.NET.152 talking to MY.NET.11.7 cause these alerts.
http://www.digitaltrust.it/arachnids/IDS311/event.html makes reference that plain Windows 2000 boxes talking to Windows 2000 Domain Controllers causes false positives with this signature.

ICMP Echo Request Nmap or HPING2
Many host on MY.NET.152 are pinging MY.NET.11.7.

MY.NET.153.197

Count	Alert Summary
19354	spp_http_decode: IIS Unicode attack detected
858	ICMP Fragment Reassembly Time Exceeded
518	connect to 515 from inside
167	MISC Large UDP Packet
5	High port 65535 udp - possible Red Worm - traffic
3	NMAP TCP ping!

spp_http_decode: IIS Unicode attack detected
MY.NET.153.197 attempting HTTP connections to many web servers with IP address having first octets of 210 and 211 causes these alerts. Whois information at ARIN shows the following for one of the web servers:

```
#whois -h whois.arin.net 211.233.28.183
```

Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
APNIC
AU

Netname: APNIC-CIDR-BLK2
Netblock: 210.0.0.0 - 211.255.255.255

This information explains the web servers having first octets of 210 and 211. Since these IP addresses are located in Asia Pacific, there is a good chance that these web servers have Chinese. http://archives.neohapsis.com/archives/snort/2001-08/0075.html states this signature triggers falsely by users browsing to sites that use multi-byte characters such as Simplified Chinese. I believe these to be false positives due to MY.NET.153.197 browsing to Chinese web sites.

ICMP Fragment Reassembly Time Exceeded
MY.NET.153.197 is sending ICMP Fragment Reassembly Time Exceeded messages to many servers in Asia Pacific. This is an indication that the servers in the Asia Pacific have probably sent some fragmented packets to MY.NET.153.197. Fragmented packets are normal as stated at http://www.security-express.com/archives/snort/2002-01/0386.html but can be used maliciously for Denial of Service attacks and IDS evasion. Since there are so many of these alerts from Asia Pacific, I am suspicious of IDS evasion or Denial of Service attacks directed at MY.NET.153.197. Without traffic captures, it is very difficult to determine what is happening here.

connect to 515 from inside
A custom Snort signature was created to detect print spooler connections on TCP port 515 initiated from MY.NET. MY.NET.150.198 appears to be running a print spooler (lpd) that is being used by other hosts. When MY.NET.153.197 connects to the spooler, the connection triggers this Snort alarm.

MISC Large UDP Packet
All but three of these alerts occurred on March 28, 2002 and have one of the following forms:

```
    211.62.59.30:2832-> MY.NET.153.197:4013
or
    211.62.59.30:0 -> MY.NET.153.197:0
```

After exhaustive research, I couldn't determine what caused these alerts. Since this traffic is always originating from the same host (not to mention Asia Pacific again) and has unique characteristics, I would suggest that MY.NET.153.197 be analyzed closer for a possible compromise or hostile activity.

High port 65535 udp - possible Red Worm – trafficI
It is likely that MY.NET.153.197 is infected with the Adore Worm since 211.239.170.174 and 211.216.46.79 in Asia Pacific is connecting to MY.NET.153.197 on UDP port 65535. http://www.simovits.com/trojans/tr_data/y49.html states the following:

"The worm searches for known vulnerabilities in wu-ftpd, BIND, LPRng and rpc.statd. If any of them are found, the worm hacks the Linux system and becomes root. It also mails information to one of four Chinese addresses."

NMAP TCP ping!
MY.NET.253.10 attempted a connection to TCP port 6112 on MY.NET.153.197. TCP port 6112 is commonly associated with the UNIX-based Common Desktop Environment (CDE) or a game called FSGS. See http://isc.incidents.org/port_details.html?port=6112. Since MY.NET.153.197 seems to be running lpd that is commonly associated with UNIX platforms, I suspect that MY.NET.253.10 is using NMAP to check for TCP port 6112 listening on MY.NET.153.197.

MY.NET.70.177

Count	Alert Summary
19460	SNMP public access
33	SMB Name Wildcard
30	Possible trojan server activity

SNMP public access
MY.NET.70.177 seems to be polling hosts on MY.NET.5 for SNMP agents listen on UDP 161 with a public community string. Public is the default community string from many SNMP agents and poses a well-known security threat. MY.NET.70.177 could be an SNMP management station or someone is using MY.NET.70.177 to perform malicious information gathering.

SMB Name Wildcard
These alerts are likely a result of normal Windows network browsing.

Possible trojan server activity
TCP port 27374 is normally associated with the trojan Subseven 2.1.4 DefCon 8. Subseven 2.1.4 is documented at http://www.simovits.com/trojans/tr_data/y1662.html. MY.NET.70.177 seems to have a server listening on TCP port 27374 that MY.NET.5.83 is connecting to. I suspect the Subseven trojan has been installed on MY.NET.70.177.

211.115.212.150

```
  Count        Alert Summary
----------     ------------------------------
   12636       spp_http_decode: IIS Unicode attack detected
```

spp_http_decode: IIS Unicode attack detected
MY.NET.153.197 is web browsing to 211.115.212.150. http://211.115.212.150 is a Chinese web site. http://archives.neohapsis.com/archives/snort/2001-08/0075.html states this signature triggers falsely by users browsing to sites that use multi-byte characters such as Simplified Chinese. These alerts are false positives due to MY.NET.153.197 browsing to Chinese web sites.

MY.NET.150.195

```
  Count        Alert Summary
----------     ------------------------------
    7225       SNMP public access
      77       WEB-MISC Attempt to execute cmd
      38       spp_http_decode: IIS Unicode attack detected
      18       INFO FTP anonymous FTP
       4       FTP CWD / - possible warez site
       1       NMAP TCP ping!
       1       ICMP Echo Request Nmap or HPING2
```

SNMP public access
Many hosts on MY.NET are sending UDP packets to port 161 on MY.NET.150.195 that contain SNMP data using the community string public. MY.NET.150.195 could be a network management station. Normally, network management managers poll and modify information on SNMP agents running on the hosts. In the normal case, the UDP port 161 traffic would be sourced from the manager not destined for the manager as with this alert. The manager normally receives information from the agents in the form of SNMP Traps usually using UDP port 162. The traffic in these alerts seems a little abnormal, but it is quite possible that the SNMP agents could be sending SNMP UDP port 161 to the manager. See http://www.rad.com/networks/1995/snmp/snmp.htm#snmp_protocols.

WEB-MISC Attempt to execute cmd
MY.NET.150.195 seems to be running a web server on port 80 and that is being probe or exploited using Nimda or Code Red. The following servers are the host performing the probes or exploits:

172.147.15.96
192.115.135.112
194.202.147.40
194.202.147.44
209.88.103.90
211.93.8.74
212.87.23.220
216.76.16.133
217.226.144.143

spp_http_decode: IIS Unicode attack detected
MY.NET.150.195 seems to be running a web server on port 80 and that is being probe or exploited using Nimda or Code Red. The same servers listed above are causing these alerts as well.

INFO FTP anonymous FTP
MY.NET.150.195 appears to be running an anonymous FTP server. Users on 65.94.248.62 and 194.38.83.245 logged into the FTP server using the username anonymous which triggered a Snort signature to generate these alerts.

FTP CWD / - possible warez site
While the user on 194.38.83.245 was anonymously logged into the FTP server on MY.NET.150.195, he or she attempted to changed directory to the root directory – "cd /". This activity might be malicious, but it is likely a normal user since I see no other scan or alert involving 194.38.83.245 except with regard to the FTP connections.

NMAP TCP ping!
MY.NET.253.10 attempted a connection to TCP port 6112 on MY.NET.150.195. TCP port 6112 is commonly associated with the UNIX-based Common Desktop Environment (CDE) or a game called FSGS. See http://isc.incidents.org/port_details.html?port=6112. MY.NET.253.10 appears to be using NMAP to check for TCP port 6112 listening on MY.NET.150.195.

ICMP Echo Request Nmap or HPING2
MY.NET.253.10 used Nmap or Hping2 to ping MY.NET.150.195 about three minutes prior to using Nmap to check for TCP port 6112. This activity by MY.NET.253.10 seems suspicious.

MY.NET.153.203

Count	Alert Summary
5994	connect to 515 from inside
1081	spp_http_decode: IIS Unicode attack detected
28	ICMP Fragment Reassembly Time Exceeded
5	INFO Possible IRC Access
4	High port 65535 udp - possible Red Worm - traffic
3	NMAP TCP ping!
1	ICMP Echo Request Nmap or HPING2

connect to 515 from inside
A custom Snort signature was created to detect print spooler connections on TCP port 515 initiated from MY.NET. MY.NET.150.198 appears to be running a print spooler (lpd) that is being used by other hosts. When MY.NET.153.203 connects to the spooler, the connection triggers this Snort alarm.

spp_http_decode: IIS Unicode attack detected
MY.NET.153.203 is web browsing to Chinese web sites that are know to cause false positives for this signature. http://archives.neohapsis.com/archives/snort/2001-08/0075.html states this signature triggers falsely by users browsing to sites that use multi-byte characters such as Simplified Chinese.

ICMP Fragment Reassembly Time Exceeded
MY.NET.153.203 is sending ICMP Fragment Reassembly Time Exceeded messages to many servers in Asia Pacific. This is an indication that the servers in the Asia Pacific have probably sent some fragmented packets to MY.NET.153.203. Fragmented packets are normal as stated at http://www.security-express.com/archives/snort/2002-01/0386.html but can be used maliciously for Denial of Service attacks and IDS evasion. Since there are so many of these alerts from Asia Pacific, I am suspicious of IDS evasion or Denial of Service attacks directed at MY.NET.153.203. Without traffic captures, it is very difficult to determine what is happening here. The following servers are involved in this activity:

211.233.70.162
211.233.70.172

INFO Possible IRC Access
MY.NET.153.203 is attempting Internet Relay Chat connects on TCP port 6667 to servers in Asia Pacific. These servers include:

211.63.185.135
211.192.139.10
211.63.185.148

High port 65535 udp - possible Red Worm – traffic
MY.NET.153.203 may be infected with the Adore Worm since MY.NET.6.50 is connecting to MY.NET.153.203 on UDP port 65535. Without packet captures, it is difficult to determine.

NMAP TCP ping!
MY.NET.253.10 attempted a connection to TCP port 6112 on MY.NET.153.203. TCP port 6112 is commonly associated with the UNIX-based Common Desktop Environment (CDE) or a game called FSGS. See http://isc.incidents.org/port_details.html?port=6112. MY.NET.253.10 appears to be using NMAP to check for TCP port 6112 listening on MY.NET.153.203.

ICMP Echo Request Nmap or HPING2
As done to other hosts, MY.NET.253.10 used Nmap or Hping2 to ping MY.NET.153.203 about eight minutes prior to using Nmap to check for TCP port 6112.

MY.NET.152.19

```
  Count        Alert Summary
----------     ------------------------------
    3305       spp_http_decode: IIS Unicode attack detected
    1109       SMB Name Wildcard
     555       ICMP Echo Request L3retriever Ping
      85       ICMP Echo Request Nmap or HPING2
      78       High port 65535 udp - possible Red Worm - traffic
      21       SCAN Proxy attempt
      14       INFO - Possible Squid Scan
       5       INFO Possible IRC Access
       3       NMAP TCP ping!
```

spp_http_decode: IIS Unicode attack detected
MY.NET.152.19 is web browsing to Chinese web sites that are know to cause false positives for this signature. http://archives.neohapsis.com/archives/snort/2001-08/0075.html states this signature triggers falsely by users browsing to sites that use multi-byte characters such as Simplified Chinese.

SMB Name Wildcard
MY.NET.152.19 and MY.NET.11.7 are communicating bi-directionally on UDP port 137. As determined in the scan data analysis above, MY.NET.11.7 is a Windows server(most likely a Domain Controller) that listens on UDP port 137. This alert is likely a result of a network logon to this server or network browsing.

ICMP Echo Request L3retriever Ping
Many hosts on MY.NET.152 talking to MY.NET.11.7 cause these alerts. http://www.digitaltrust.it/arachnids/IDS311/event.html makes reference that plain Windows 2000 boxes talking to Windows 2000 Domain Controllers causes false positives with this signature. This alert is likely a result of a network logon to this server or network browsing.

ICMP Echo Request Nmap or HPING2
Many hosts on MY.NET.152 talking to MY.NET.11.7 cause these alerts. http://www.digitaltrust.it/arachnids/IDS311/event.html makes reference that plain Windows 2000 boxes talking to Windows 2000 Domain Controllers causes false positives with this signature. This alert is likely a result of a network logon to this server or network browsing.

High port 65535 udp - possible Red Worm – traffic
MY.NET.152.19 may be infected with the Adore Worm since the MY.NET.6 network is connecting to MY.NET.152.19 on UDP port 65535. MY.NET.152.19 is also communicating to the MY.NET.6 network on UDP port 65535. Without packet captures, it is difficult to determine what this activity is.

SCAN Proxy attempt
Many host external to MY.NET are attempting to connect to MY.NET.152.19 on TCP ports 1080 and 8080. Much I research shows that these scans are normally information gathering.

INFO - Possible Squid Scan
Many host external to MY.NET are attempting to connect to MY.NET.152.19 on TCP ports 1080 and 8080. Much I research shows that these scans are normally information gathering. They same external IP addresses that caused the SCAN Proxy attempt alerts also caused these alerts.

INFO Possible IRC Access
MY.NET.152.19 is attempting Internet Relay Chat connects on TCP port 6667 to following servers:

211.216.53.129
213.197.128.90

NMAP TCP ping!
MY.NET.253.10 attempted a connection to TCP port 6112 on MY.NET.152.19. TCP port 6112 is commonly associated with the UNIX-based Common Desktop Environment (CDE) or a game called FSGS. See http://isc.incidents.org/port_details.html?port=6112. MY.NET.253.10 appears to be using NMAP to check for TCP port 6112 listening on MY.NET.152.19.

MY.NET.153.119

Count	Alert Summary
4772	connect to 515 from inside
328	spp_http_decode: IIS Unicode attack detected
3	NMAP TCP ping!
1	ICMP Echo Request Nmap or HPING2
1	EXPLOIT x86 NOOP

connect to 515 from inside
A custom Snort signature was created to detect print spooler connections on TCP port 515 initiated from MY.NET. MY.NET.150.198 appears to be running a print spooler (lpd) that is being used by other hosts. When MY.NET.153.119 connects to the spooler, the connection triggers this Snort alarm.

spp_http_decode: IIS Unicode attack detected
MY.NET.153.119 is web browsing to Chinese web sites that are know to cause false positives for this signature. http://archives.neohapsis.com/archives/snort/2001-08/0075.html states this signature triggers falsely by users browsing to sites that use multi-byte characters such as Simplified Chinese.

NMAP TCP ping!
MY.NET.253.10 attempted a connection to TCP port 6112 on MY.NET.153.119. TCP port 6112 is commonly associated with the UNIX-based Common Desktop Environment (CDE) or a game called FSGS. See http://isc.incidents.org/port_details.html?port=6112. MY.NET.253.10 appears to be using NMAP to check for TCP port 6112 listening on MY.NET.153.119.

ICMP Echo Request Nmap or HPING2
As done to other hosts, MY.NET.253.10 used Nmap or Hping2 to ping MY.NET.153.203 about three minutes prior to using Nmap to check for TCP port 6112.

EXPLOIT x86 NOOP

This alert was generated as a result of MY.NET.153.203 browsing to the HTTP server on 216.117.135.222. The return traffic from the HTTP server contained many NOOP. NOOPs are a number of contiguous bytes that could be no-operation machine languange codes for a particular architecture. This is likely a false positive. See analysis at http://www.sans.org/y2k/practical/David_Oborn_GCIA.html#detect4.

List of detects

This is a list of all detects prioritized by number of occurrences found in the scan data. As a result of my process, I have given brief explanations of many these detects in my analysis above.

Count	Alert Summary
57675	spp_http_decode: IIS Unicode attack detected
47283	SMB Name Wildcard
44979	connect to 515 from inside
37562	SNMP public access
23126	ICMP Echo Request L3retriever Ping
7654	INFO MSN IM Chat data
3742	ICMP Echo Request Nmap or HPING2
2933	INFO Outbound GNUTella Connect request
2242	High port 65535 udp - possible Red Worm - traffic
2190	INFO Inbound GNUTella Connect request
2134	Watchlist 000220 IL-ISDNNET-990517
1735	ICMP Fragment Reassembly Time Exceeded
1727	MISC Large UDP Packet
891	WEB-IIS view source via translate header
883	WEB-MISC Attempt to execute cmd
874	ICMP Router Selection
865	NMAP TCP ping!
861	Port 55850 tcp - Possible myserver activity - ref. 010313-1
548	FTP DoS ftpd globbing
382	Null scan!
348	Watchlist 000222 NET-NCFC
219	SCAN Proxy attempt
210	INFO FTP anonymous FTP
208	Possible trojan server activity
188	WEB-FRONTPAGE _vti_rpc access
184	WEB-IIS _vti_inf access
140	INFO napster login

131	WEB-CGI scriptalias access
119	suspicious host traffic
93	INFO Possible IRC Access
90	ICMP Destination Unreachable (Communication Administratively Prohibited)
87	INFO - Possible Squid Scan
79	INFO Napster Client Data
60	Queso fingerprint
55	Incomplete Packet Fragments Discarded
54	FTP CWD / - possible warez site
53	WEB-MISC 403 Forbidden
51	High port 65535 tcp - possible Red Worm - traffic
46	spp_http_decode: CGI Null Byte attack detected
42	SCAN Synscan Portscan ID 19104
42	ICMP Echo Request Windows
24	Russia Dynamo - SANS Flash 28-jul-00
24	EXPLOIT x86 setuid 0
22	EXPLOIT x86 NOOP
19	ICMP traceroute
17	WEB-MISC compaq nsight directory traversal
12	EXPLOIT x86 setgid 0
10	ICMP Echo Request BSDtype
10	Attempted Sun RPC high port access
9	Tiny Fragments - Possible Hostile Activity
7	TCP SRC and DST outside network
7	MISC traceroute
7	Back Orifice
6	WEB-MISC http directory traversal
5	WEB-IIS Unauthorized IP Access Attempt
5	EXPLOIT NTPDX buffer overflow
4	SCAN FIN
4	ICMP Destination Unreachable (Protocol Unreachable)
4	BACKDOOR NetMetro Incoming Traffic
3	x86 NOOP - unicode BUFFER OVERFLOW ATTACK
3	WEB-MISC ICQ Webfront HTTP DOS
3	INFO Inbound GNUTella Connect accept
2	RPC tcp traffic contains bin_sh
2	Port 55850 udp - Possible myserver activity - ref. 010313-1
2	ICMP Echo Request CyberKit 2.2 Windows
2	BACKDOOR NetMetro File List
1	X11 outgoing
1	WEB-MISC webdav search access
1	TFTP - Internal UDP connection to external tftp server

1	TFTP - External UDP connection to internal tftp server
1	SYN-FIN scan!
1	SMB CD...
1	ICMP Echo Request Sun Solaris
1	EXPLOIT x86 stealth noop
1	EXPLOIT x86 NOPS

OOS Data Analysis

OOS Top Talkers

The most prevalent "IP address" disregarding whether it was a source or destination was the criteria used for the OOS Top Talkers.

Count	IP address
30	MY.NET.150.113
29	80.133.124.114
5	MY.NET.152.21
4	213.169.245.41
2	MY.NET.153.210
2	MY.NET.150.220
2	128.97.84.53
1	MY.NET.153.196
1	MY.NET.153.191
1	MY.NET.150.226

MY.NET.150.113

All thirty of the OOS packets involving MY.NET.150.113 are destined for MY.NET.150.113 on TCP port 1214. TCP port 1214 is normally associated with file sharing applications like KaZaa, Morpheous, or Grokster. More port information can be found at http://isc.incidents.org/port_details.html?port=1214. Of the thirty OOS packets involving MY.NET.150.113 and TCP 1214, twenty-nine packets have the following form:

```
03/28-06:55:40.354933 80.133.124.114:4026 -> MY.NET.150.113:1214
TCP TTL:39 TOS:0x0 ID:31318  DF
21S***** Seq: 0xBFCF6268   Ack: 0x0   Win: 0x16B0
TCP Options => MSS: 1412 SackOK TS: 71742 0 EOL EOL EOL EOL
```

At http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc, Matthew provides reference to information about hacking peer to peer file sharing applications. These could be possible attempts to exploit TCP port 1214 on MY.NET.150.113. Since all of these OOS packets have similar form (same TCP flags, TCP options, and data) and are distributed over a two day period, it likely that file sharing application software is generating these packets not a human crafting packets or performing scanning.

These packets also caused "Queso fingerprint" alert data indicating that someone might be using Queso to OS fingerprint MY.NET.150.113, but references at http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids29&view=event give that these alerts is a false positive from old reserved and unused bits, ECN and CWR, being used for Quality of Service (QoS).

80.133.124.114

This is host was sending OOS packets to MY.NET.150.113. See details above.

MY.NET.152.21

213.169.245.41 and 217.82.123.75 are sending OOS packets to MY.NET.152.21 on TCP port 6346. This port is commonly associated with a fully-distributed information-sharing technology called Gnutella. These OOS packets are very similar to the TCP port 1214 activity discussed above. The difference with the OSS packets to MY.NET.152.21 on TCP 6346 is that the have different forms each time. The TCP flags, TCP options and packet data are different for each packet. This is an indication that users on 213.169.245 and 217.82.123.75 might be sending crafted packets to MY.NET.152.21 in order to exploit Gnutella, but more likely these users are using a scanning tool to OS fingerprint MY.NET.152.21. These packets generated "Null scan!" alerts that indicate OS fingerprint activity.

213.169.245.41

This host is sending OOS packets to TCP port 6346 on MY.NET.152.21. See above.

MY.NET.153.210

128.97.84.53 sent two OOS packets to MY.NET.153.210 having the following forms:

```
        128.97.84.53:20 -> MY.NET.153.210:1320
and
        128.97.84.53:2075 -> MY.NET.153.210:113
```

Both of these packets have the same TCP flags, TCP options and packet data, but have different TCP ports. This gives an indication that someone is running a scanner at MY.NET.153.210.

These packets also caused "Queso fingerprint" alert data indicating that someone might be using Queso to OS fingerprint MY.NET.153.210, but references at http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids29&view=event give that these alerts is a false positive from old reserved and unused bits, ECN and CWR, being used for Quality of Service (QoS).

These packets are using the ECN and CWR bits so I suspect that these "Queso fingerprint" alerts are false positives. TCP port 20 is the data port for FTP and TCP port 113 is the port used by ident. POP mail, FTP, and HTTP servers can use the ident protocol to identify incoming users. See http://www.cisco.com/warp/public/110/2.html. I suspect these OOS packets are false positives due to a connection to an FTP server using ident with the ECN and CWR bits are used.

MY.NET.150.220

61.216.83.124 and 140.110.30.59 sent an OOS packet to MY.NET.150.220 on TCP port 4662. TCP port 4662 is normally associated with eDonkey2000 that is mentioned above in the scan data analysis. These two packets are very different, but are destined for the same TCP port. The packet from 61.216.83.124 has very abnormal TCP flags - 2*SFRPAU. This may be OS fingerprinting. The packet from 140.110.30.59 doesn't seem abnormal but uses the ECN and CWR bits. I suspect this is a false positive due to the information provided earlier on old reserved and unused bits.

128.97.84.53

128.97.84.53 sent two OOS packets to MY.NET.153.210. See above OSS analysis for MY.NET.153.210.

MY.NET.153.196

80.144.189.160 sent an OOS packet to MY.NET.153.196 on TCP port 6346. The packet doesn't seem abnormal but uses the ECN and CWR bits. I suspect this is a false positive due to the information provided earlier on old reserved and unused bits.

MY.NET.153.191

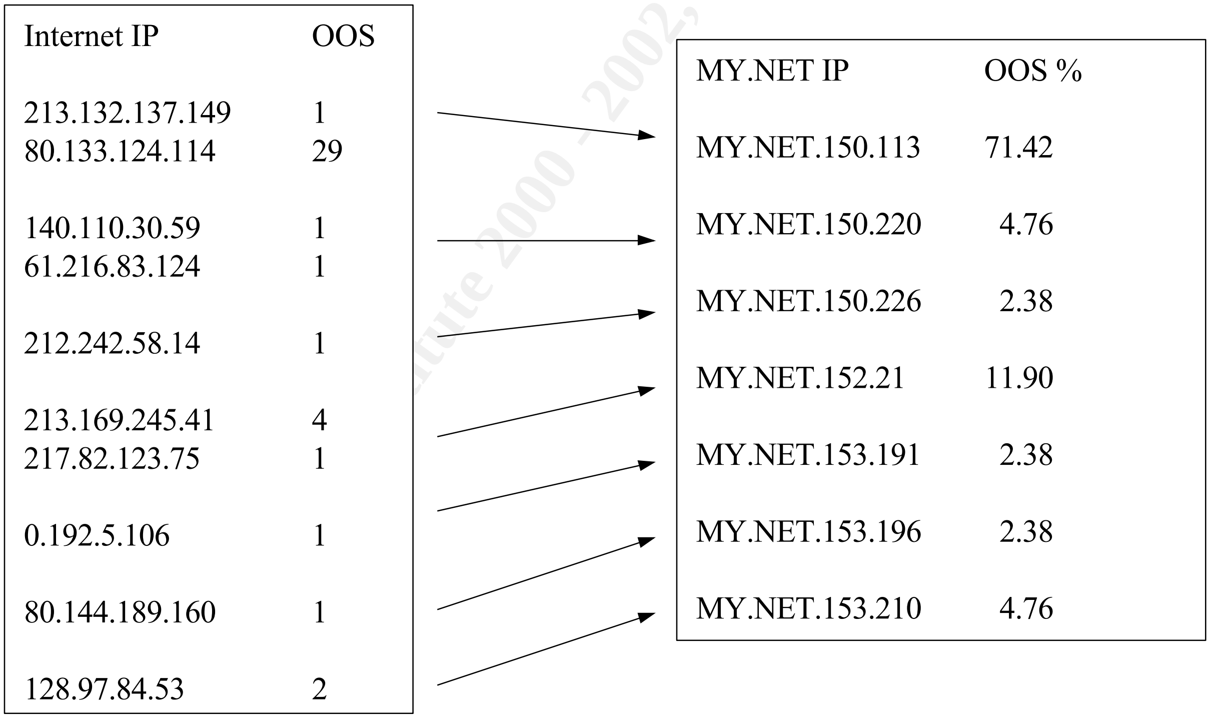
An OOS packet was sent to MY.NET.153.191 on TCP port 33376 from an odd and spoofed address of 0.192.5.106. The packet had “2*SF***U” as TCP flags. This could be an indication that someone is performing a SYN-FIN scan, but the source address indicates that the packet is spoofed. This packet was probably crafted.

MY.NET.150.226

212.242.58.14 sent an OOS packet to MY.NET.150.226 with abnormal TCP flags and TCP options. The TCP flags are “21S*R*A*” and the TCP options have server EOL characters. A user on 212.242.58.14 seems to be running some type of scanner at MY.NET.150.226.

OOS Link Graph

The following link graph provides a visual representation of the percentage of OOS packets destined for hosts on MY.NET from hosts on the Internet found.



Internet IP	OOS
213.132.137.149	1
80.133.124.114	29
140.110.30.59	1
61.216.83.124	1
212.242.58.14	1
213.169.245.41	4
217.82.123.75	1
0.192.5.106	1
80.144.189.160	1
128.97.84.53	2

MY.NET IP	OOS %
MY.NET.150.113	71.42
MY.NET.150.220	4.76
MY.NET.150.226	2.38
MY.NET.152.21	11.90
MY.NET.153.191	2.38
MY.NET.153.196	2.38
MY.NET.153.210	4.76

External Source Registration Information

I choose to gather the registration information for all the valid source addresses found in the OOS data. Based on the nature of OOS data, these hosts are normally performing information gathering or scanning of hosts on MY.NET.

213.132.137.149

```
# nslookup 213.132.137.149

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

149.137.132.213.in-addr.arpa    name = cable-213-132-137-149.upc.chello.be.

# whois -h whois.arin.net 213.132.137.149
European Regional Internet Registry/RIPE NCC (NETBLK-213-RIPE)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/perl/whois/
   NL

   Netname: RIPE-213
   Netblock: 213.0.0.0 - 213.255.255.255
   Maintainer: RIPE
```

80.133.124.114

```
# nslookup 80.133.124.114

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

114.124.133.80.in-addr.arpa     name = p50857C72.dip.t-dialin.net.

# whois -h whois.arin.net 80.133.124.114
European Regional Internet Registry/RIPE NCC (NET-80-RIPE)
   These addresses have been further assigned
   to European users. Contact information can
   be found in the RIPE database at whois.ripe.net
   NL
   Netname: 80-RIPE
   Netblock: 80.0.0.0 - 80.255.255.255
   Maintainer: RIPE
```

140.110.30.59

```
# nslookup 140.110.30.59

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

59.30.110.140.in-addr.arpa      name = hpcs009.nchc.gov.tw.

# whois -h whois.arin.net 140.110.30.59
Ministry of Education Computer Center (NETBLK-TANET) TANET-BNETS
                                            140.109.0.0 - 140.111.255.255
Ministry of Education Computer Center (NET-TANET-BNET22) TANET-BNET2
                                            140.110.0.0 - 140.110.255.255
```

61.216.83.124

```
# nslookup 61.216.83.124

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

124.83.216.61.in-addr.arpa      name = 61-216-83-124.HINET-IP.hinet.net.

# whois -h whois.arin.net 61.216.83.124
Asia Pacific Network Information Center (NETBLK-APNIC2)
   APNIC
   AU

   Netname: APNIC3
   Netblock: 61.0.0.0 - 61.255.255.255
   Maintainer: AP
```

212.242.58.14

```
# nslookup 212.242.58.14
```

```
Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

14.58.242.212.in-addr.arpa      name = port75.ds1-vbr.adsl.cybercity.dk.

# whois -h whois.arin.net 212.242.58.14
European Regional Internet Registry/RIPE NCC (NET-RIPE-NCC-)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/perl/whois/
   NL

   Netname: RIPE-NCC-212
   Netblock: 212.0.0.0 - 212.255.255.255
   Maintainer: RIPE
```

213.169.245.41

```
# nslookup 213.169.245.41

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

41.245.169.213.in-addr.arpa     name = 245.169.213-41-dial-in-dynamic.ision.nl.

# whois -h whois.arin.net 213.169.245.41
European Regional Internet Registry/RIPE NCC (NETBLK-213-RIPE)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/perl/whois/
   NL

   Netname: RIPE-213
   Netblock: 213.0.0.0 - 213.255.255.255
   Maintainer: RIPE
```

217.82.123.75

```
# nslookup 217.82.123.75

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

75.123.82.217.in-addr.arpa      name = pD9527B4B.dip.t-dialin.net.

# whois -h whois.arin.net 217.82.123.75
European Regional Internet Registry/RIPE NCC (NET-217-RIPE)
   These addresses have been further assigned
   to European users. Contact information can
   be found in the RIPE database at whois.ripe.net
   NL

   Netname: 217-RIPE
   Netblock: 217.0.0.0 - 217.255.255.255
   Maintainer: RIPE
```

80.144.189.160

```
# nslookup 80.144.189.160

Server:         DNS.SVR.IP
Address:        DNS.SVR.IP#53

160.189.144.80.in-addr.arpa     name = p5090BDA0.dip.t-dialin.net.

# whois -h whois.arin.net 80.144.189.160
European Regional Internet Registry/RIPE NCC (NET-80-RIPE)
   These addresses have been further assigned
   to European users. Contact information can
   be found in the RIPE database at whois.ripe.net
   NL

   Netname: 80-RIPE
   Netblock: 80.0.0.0 - 80.255.255.255
   Maintainer: RIPE
```

128.97.84.53

```
# nslookup 128.97.84.53
```

```
Server:        DNS.SVR.IP
Address:        DNS.SVR.IP#53

53.84.97.128.in-addr.arpa       name = ndep.seas.ucla.edu.

# whois -h whois.arin.net 128.97.84.53
University of California, Los Angeles (NET-UCLANET)
   741 Circle Dr South
   Los Angeles, CA 90095-1363
   US

   Netname: UCLANET
   Netblock: 128.97.0.0 - 128.97.255.255
```

Defense Recommendations

Normally, universities do not have rigid security policies due to the importance of sharing of information for the purposes of research and learning. As a result, many applications are used on this network that are not allow on many corporate networks. The following recommendations are a result of the findings in the analysis above. The university staff is encouraged to thoroughly understand the analysis above, further qualify any assumptions made, and use their own knowledge of the network before making any changes.

Even though universities do not have strict security policies, firewall architectures be assessed to determine if there are any configuration changes or firewall additions that need to be. Firewalls could be used to protect many of the critical servers.

Due to the difficulty of implementing tight firewall policies, real-time 24x7x365 monitoring of the IDS logs is very important. Monitoring the firewall and critical server logs could be of extreme value in the environment as well. Using a Managed Security Service Provider such as LURHQ Corporation for this monitoring is suggested.

A thorough vulnerability analysis is recommended for hosts on MY.NET that are mentioned in the analysis above. All Top Ten Talkers in the Scan, Alert, and OOS data should definitely be considered for this analysis, but the following hosts are highly encouraged:

MY.NET.70.234
MY.NET.150.114
MY.NET.151.71

MY.NET.70.177
MY.NET.150.198
MY.NET.153.197
MY.NET.150.195
MY.NET.153.203
MY.NET.152.19

Due to the many false alarms in the scan data, it is recommended that the Snort port scan preprocessors be tuned. This would significantly reduce the data that would need to be analyzed in the future. The following line in snort.conf can be used to tune the port scan thresholds:

```
preprocessor portscan: $HOME_NET 4 3 portscan.log
```

and the following line can be used to exclude heavily loaded servers from creating port scan alarms:

```
preprocessor portscan-ignorehosts: 0.0.0.0
```

Along with tuning the port scan preprocessor, the http_decode preprocessor should be considered for tuning as well. The "spp_http_decode: IIS Unicode attack detected" alerts where all generated by this preprocessor. Many of these resulted from browsing to Asian websites. Minimizing these alerts would reduce that alert data the needs to be analyzed. The following snort.conf line can be used to disable detection of UNICODE directory traversal attacks and CGI NULL code attacks:

```
preprocessor http_decode: 80 -unicode -cginull
```

Other signature tuning is recommended as well, but the following examples are encouraged. If MY.NET.150.198 is not compromised and TCP port 515 is used for lpd, then the Snort signature producing the "connect to 515 from inside" should probably be disabled or at least modified to exclude MY.NET.150.198 as a destination. Since Windows network logons generate "ICMP Echo Request L3retriever Ping" alerts, the signature that produces this message could be disabled or modified to exclude the windows servers as destinations. In general, university staff members are encouraged to use knowledge of the network to tune the Snort signatures in an effort to reduce the amount log data that needs to be analyzed.

Resources

Many resources were invaluable during this analysis. The web proved to be the most useful resource.

Websites

http://www.sans.org
http://www.giac.org
http://www.incidents.org
http://www.whitehats.com
http://www.neohapsis.com
http://www.iana.org
http://www.insecure.org
http://www.faqs.org
http://www.microsoft.com
http://zone.msn.com
http://www.symantec.com
http://www.cisco.com
http://www-124.ibm.com
http://www.startron.org
http://www.edonkey2000.com
http://www.thedonkeynetwork.com
http://www.gnutella.com
http://www.gnutellanews.com
http://www.security-express.com
http://www.simovits.com
http://www.digitaltrust.it
http://www.google.com

Two DELL Inspiron laptops were used during this exercise. One of these laptops has Mandrake Linux as the operating system. This laptop was used to parse and analyze the IDS log data. Combinations of the following commands were used on this laptop to parse and analyze the data:

grep
egrep
cut
awk
sort
uniq

Another laptop running Microsoft Windows 2000 was used to create this document and perform research on the web. Microsoft Word and Internet Explorer were used to create this document and for web browsing respectively.   OpenSSH was used to remotely access the Linux laptop for analysis.