Global Information Assurance Certification Paper

GIAC Certified Intrusion Analyst (GCIA)
Practical Assignment
Version 3.3

James Filiberto
SANS Maryland
December 7, 2002

Part 1

Describe The State Of Intrusion Detection

Intrusion Detection is a broad term that can be defined as identifying malicious or unauthorized traffic or misuse on a network or host. Intrusion detection systems can consist of several components, of which each performs a specific function. The placement of these components can differ depending on the type of IDS being deployed.

These components consist of the following general processes.

1) Data collection
2) Data Analysis
3) Response

This is a very basic overview of components in an IDS system and be aware that there are many configurations employed today that add additional components, more complex configurations and processing levels.

1) Data Collection is simply the process of collecting the data to be analyzed. In a distributed IDS environment, data is collected and processed at multiple hosts. In a centralized IDS environment the data can be collected by several hosts but is sent to a central host for processing. Data collection is an important component as incomplete data or delay of data collection will severely hamper intrusion detection.

2) Data Analysis is the classification of the collected data. This has 2 components.
   a) The database which contains signatures that the IDS will take action on
   b) Data classification is the process of comparing the data to the database.

   The database contains all the details that define an alert for the IDS. This is other wise known as a rule base. Then the classification engine determines if the data analyzed falls into the alert category. If the classification engine detects an attack, an alert is generated.

3) Response is the action to be preformed as defined by the IDS and the alert generated. This can be different actions, which can be defined by the severity rating of the alert. There are three general response categories. Passive, Reactive, and Proactive.

As previously stated, the placement of the components, addition of components and several other factors can help in determining the type of IDS. Now for a brief discussion of the generally accepted classes of an IDS.

IDS's can be broken up into 3 primary types:

1) Host Based Intrusion Detection Systems
2) Network Based Systems
3) Network Node Based systems

There are also combinations of the above, as in the case of Hybrid IDS's.

HIDS
Host Intrusion Detection Systems are systems that monitor activity on the host generally using logs. However the HIDS could reside between the application and kernel levels and monitor system and api calls to detect as well as help stop malicious behavior.

Network Based Systems are systems that monitor all traffic on the wire and look for defined signatures or irregular traffic.

Network Node Based systems are systems that apply analysis at the wire level of a node and pass events to a central console for notification and possible correlation.

While the topic of Intrusion Detection is complicated and it's deployment can take on many simple and complex configurations, the above is just meant to be a simplified overview and introduction into a more definitive and detailed explanation of one of the primary types of IDS.

Host Based Intrusion Systems.
While the world of Intrusion Detection is always in constant flux, the goal should always be static with the premise of 'Prevention is ideal, detection is a must'. I was first introduced to this important concept at a SANS conference. And when applied to the 'Defense in Depth' model, which I also first learned about at a SANS conference, it became clear to me that as the threats of intrusion grew and the points from which these attacks are mounted can originate from anywhere, that HIDS will take on a very special role in the future of Intrusion Detection. With the use of VPN's, remote access, and networks that are connected to partners and suppliers, the danger of attacks coming from within, or over encrypted connections is growing at a rapid pace. The Defense in Depth model employing HIDS can help protect sensitive or mission critical servers and workstations. The following is a discussion on HIDS.

Host Based Intrusion Detection Systems are usually employed by administrators who have mission critical, security sensitive or private systems that require a higher level of security then a Network Intrusion Detection System may provide. These important systems can be, but are not limited to, Research and Development servers, accounting servers, executive workstations, etc. If an added layer of security is warranted, a HIDS can help provide that security and more importantly help in the detection of attacks or attempted attacks and possibly an interruption of an ongoing attack. Also a HIDS can be a great asset in a switched network environment. As networks get faster and the amount of nodes grows, or too many VLANS are setup to port monitor to a NIDS, it is possible for the IDS data collection agent to get overwhelmed by too much data and that may cause it to drop packets if it cannot process them fast enough, thereby possibly missing crucial attack signatures or anomalies in protocols. HIDS systems only need to concentrate on the traffic that is destined for the host on which it resides, a much more manageable load for the intrusion detection process.

How HIDS Works

In the early days host based intrusion detection consisted generally of reviewing the system and audit logs and looking for any suspicious activity. These logs, as well as network activity consisted of a much smaller volume of data to be analyzed when compared to the amount of data produced by today's hosts. As networks and hosts saw an ever increasing volume of traffic, this tasks of manually searching through the logs became cumbersome and time consuming. HIDS today still employs some of those basic techniques as well as several new ones. But today those techniques are implemented in an automated fashion, which also employs alerting, cataloging, and occasionally reactive and proactive mechanisms.

Host based intrusion detection consists of 4 general areas of analysis. The first area we will discuss is network traffic. This includes all traffic coming into or leaving the host. The traffic component can be compared to a host based firewall in that it monitors the ports and can be set to alert on specific port accesses or attempts coming into or leaving the host. Port Sentry is an example of this type of IDS tool. This type of detection can initiate several responses related to the detection rules. In the case of a Denial of Service attack, a HIDS could generate a host based firewall rule that would drop all traffic from a specific host or a specific protocol after certain thresholds have been met. And an alert can be generated simultaneously, thereby stopping or mitigating the attack and notifying administrators who can take further protective and reactive responses. This component can also be invoked to alert an administrator to traffic anomaly's. For instance, if the host has a historical traffic pattern that is documented and data transfer amounts exceed time based thresholds, an alert can be generated as well as any reactive measures initiated by the detection system.

The second area is Log File Monitoring. As stated above, this is probably the original intrusion detection method. And is still very useful today, albeit in an automated fashion. Programs like Swatch monitor log files and look for anomaly's, patterns, or specific entries and send alerts via email, pager, or can execute a file as predetermined in the action of a defined trigger. Even though this is a very simple tool, its usefulness should not be underestimated or bypassed. The benefits of a system like this become evident in its ease of determining if a system is under attack or already compromised. The proper configuration of a Log File Monitoring System is a very important and can be customized down to a very granular point, depending on the services offered by the host giving a tremendous insight into the host's actions.

The third area is File System Integrity. File system integrity checkers main function is to determine that key system files and programs have not been altered from an established benchmark. An example of a File System Integrity Checker is Tripwire. This component of Intrusion Detection is another area that can be customized down to a very granular level depending on the host it is applied to. Taking those two statements into consideration should emphasize how vital this type of detection is to the security of key systems. Also take into consideration that root kits replace key files to mask the hackers actions and a common way to do this is to install altered system programs such as ls, top, ps, and other files commonly used by administrators. File System Integrity Checkers generally create this benchmark from a fresh install of a system not connected to any other systems. How this is done is by the File System Integrity Checker running a checksum or cryptographic hash against a defined set of key files. These checksums and hashes are kept in a database which is referenced every time the File System Integrity Checker is run. The database will need to be updated after upgrades and certain system maintenance procedures to keep the validity of the database current, but this is a small price to pay when talking about the security of a major or sensitive system.

The fourth area will be about processes and system calls. These two areas can be a key chokepoint in detecting intrusions and possible prevention of malicious processes or system calls and I believe this area will have a tremendous impact on next generation IDS's. Processes can generate system calls that can have a great impact on a system. This component of a Host Intrusion Detection System tries to define normal behaviors of privileged processes. By defining programs and their system calls and any parameters of those calls, and any expected sequence of calls, a smaller more efficient database for detection and prevention is created. Based on the relatively small database size the kernel checks each system call made by a privileged process and if it is an accepted call or sequence of calls or call with accepted parameters, then it is allowed through. However if it violates the expected behavior an alert can be generated as well as sandboxing the process. An example of this type of program is LIDS or Janus.

To conclude, HIDS or Host Intrusion Detection Systems may be coming into favor for sensitive or high priority systems because of the additional detection and protection capabilities that they can offer on a host basis. The higher level of maintenance these systems demand provide an extremely good return on the time invested. And the concentrated examination of the processes running, traffic accessing and file integrity on them can help limit or prevent malicious or aberrant behavior no matter where it originates.

Assignment 2- Network Detects

Detect 1

1. Source of the trace
   http://www.incidents.org/logs/Raw/2002.5.27
   Although the file is named 2002.5.27, the timestamps indicate that the packets are from 06/27. This has been noted in other practicals I have read that have used this source for data.

   2 Detect was generated by:
   Snort 2.0.0 (Build 72) using the rule set included in the download from http://www.snort.org/dl/binaries/win32/snort-2_0_0.exe on 5/3/2003.

[**] [1:1616:4] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
06/26-21:15:39.944488 203.155.227.98:2297 -> 46.5.160.168:53
UDP TTL:45 TOS:0x0 ID:23059 IpLen:20 DgmLen:58
Len: 30
[Xref => http://www.whitehats.com/info/IDS278][Xref => http://cgi.nessus.org/plugins/dump.php3?id=10028]

Upon further investigation, specifically, loading the log file into Ethereal to determine the extent of the suspected reconnaissance, it was noted that as I scrolled down looking for the first DNS packet, I noticed the port number 31337 in the Info field and a source address of 255.255.255.255 at line 63. I only mention this to point out that my default configuration of the installation of Snort did not see this, and it was only picked up after manual review of the log file in Ethereal.

For confirmation, the following rule was added to snort.

alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q access"; flags:A+; dsize: >1; reference:arachnids,203; sid:184; classtype:misc-activity; rev:3;)

This rule identifies any packets with the source IP address of 255.255.255.255, which should not be seen in the normal course of events. It also looks for the Ack flag and a payload size greater than 1.

Then Snort was run again with the following result. There were 41 additional packets found that are not displayed for brevity. The packets not shown are identical except for the time stamps and the destination IP address.

[**] BACKDOOR Q access [**]

```
06/26-20:51:20.024488 255.255.255.255:31337 -> 46.5.188.182:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=
```

3. Probability the source address was spoofed:
It very likely that this source address was spoofed considering the source address is a broadcast address. Also take into consideration the sequence numbers and next sequence numbers being consistent at 0 and 3 respectively , and the reset and ack bits set. This indicates that these are crafted packets.

4. Description of Attack:

The Q Trojan by Mixter. This Trojan, rather than listen on a port will look for raw IP packets that match a predetermined parameter. The initial data of these packets will contain encrypted instructions that the Q daemon will execute. The Q Trojan client generates TCP, UDP, and ICMP RawIP control packets.

CVE CAN-1999-0660
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE++CAN-1999-0660

5. Attack Mechanism

The attack utilizes RawIp packets to initiate the connection. The RawIP interface was designed to help in the implementation of new IP protocols. RawIP is different from IP in that it allows multiple endpoints to be bound to the same protocol address. SOCK_RAW sockets give the application an interface to lower layer protocols, such as IP and ICMP. This interface is often used to bypass the transport layer when direct access to lower layer protocols is needed. The connect() call from the client can be used to specify the connecting peer the server will connect to. After researching this, I have come to the conclusion that this is indeed a Trojan which is triggered when the compromised system receives a RawIp packet containing specific data which thereby triggers a connect to a system whose IP address or host name has been defined and encrypted in the payload of a crafted RawIP packet from the client. Some traits of the packets seem to be crafted to avoid detection. Specifically, the broadcast source address, which is unusual behavior and some firewalls may pass these packets. And the

fact that the ACK and RST bits were set could also bypass some firewall configurations. Then if the client packets can get to the compromised system, it includes the command to make a connection to the client. This may go unnoticed as the connection was initiated from inside. Stateful firewalls may see this as allowed traffic especially if it's destination port was 80 or 443 or some other common protocol . And there is the possibility that the trojaned system may be sent a command to do some reconnaissance of the internal network by sending out crafted packets and then log or forward any responses. It seems from the number of packets, the time between packets and the randomness of the destination IP addresses that this is a stealthy scan looking for systems that have already been compromised.

6. Correlations:

IDS203 "TROJAN-ACTIVE-Q-TCP" entry from Whitehats.com
http://www.whitehats.com/info/IDS203

Raw IP Trojans
http://lists.jammed.com/pen-test/2002/10/0027.html

A brief programming tutorial in C for raw sockets
http://mixter.warrior2k.com/rawip.html

Using RawIP
http://developer.apple.com/techpubs/mac/NetworkingOT/NetworkingWOT-56.html

CREATE A SOCKET
http://www.ucc.uconn.edu/cgi-bin/cmshelp?SOCKETS%20SOCKET

7. Evidence of Active Targeting

There is no evidence of active targeting as the randomness of the addresses and the inconsistent times of the packets actually indicate that this may be some automated tool doing a stealth type scan trying to locate compromised systems.

8. Severity

Severity will be calculated with the following formula:

severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Each value will be ranked on a scale from 1 (lowest) to 5 (highest).

Criticality = 1
There was no specific system targeted.

Lethality = 5
If this attack was successful, it would indicate that the system was compromised at the root level.

System countermeasures = 1
Unsure of any counter measures in place.

Network countermeasures = 2
There is an IDS or some other packet collecting device in place.

severity = (1 + 5) – (1 + 2)
severity = 3

9. Defensive Recommendations
Recommend border route block all ingress broadcast and multicast source addresses. Recommend firewall block all inbound packets with a source port of 31337.
Recommend IDS rule for low TTL's

10. Multiple choice Question:
What port does the Q Trojan bind to ?

A) 23
B) 80
C) none
D) all

Answer C – The Q Trojan uses RawIp, which listens to lower layer protocols.

This detect was posted to the intrusions list at incidents.org on 05/04/2003. Text above includes changes to the original version I posted based on responses I received. The following remarks from Brian Coyle on 05/04/2003 01:22:01.

Remark 1
What do the Q alerts have to do with the named version alert?
Why even mention the DNS packet if you're not going to analyze it?

I was trying to show the path the analysis took to what I considered a more crucial and pressing intrusion. Whereas I recognized the possible attempt to compromise what may be a non existent DNS issue, or possibly simple reconnaissance, I felt compelled to pursue the protocol anomaly which I felt critical and demanded further investigation. As stated above 'I only mention

this to point out that my default configuration of the installation of Snort did not see this, and it was only picked up after manual review of the log file in Ethereal.'

The following 2 remarks from Don Murdoch on 05/04/03 08:52:46.
Remark 1

For confirmation, the following rule was added to snort.

alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q access"; flags:A+; dsize: >1; reference:arachnids,203; sid:184; classtype:misc-activity; rev:3;)

Don: why did you add this rule, what does it do?

After noting this 'elite' port and the broadcast address in the source, it became evident that these were somehow crafted packets and after some searching on the address port combination, several references to Backdoor Q came up. With that information I constructed a rule that would alert on any source IP address of 255.255.255.0/24 to the internal network with the ACK flag set in any combination.

Remark 2

5. Attack Mechanism
The attack utilizes raw ip packets to initiate the connection. The RawIP interface was designed to help in the implementation of new IP protocols. RawIP is different in that it allows multiple endpoints to be bound to the same protocol address. SOCK_RAW sockets give the application an interface to lower layer protocols, such as IP and ICMP. This interface is often used to bypass the transport layer when direct access to lower layer protocols is needed. The connect() call from the client can be used to specify the connecting peer the server will connect to. After researching this, I have come to the conclusion

Don: where did you get your data from - reference and URL.

Raw IP Trojans
http://lists.jammed.com/pen-test/2002/10/0027.html

A brief programming tutorial in C for raw sockets
http://mixter.warrior2k.com/rawip.html

Using RawIP
http://developer.apple.com/techpubs/mac/NetworkingOT/NetworkingWOT-56.html

CREATE A SOCKET
http://www.ucc.uconn.edu/cgi-bin/cmshelp?SOCKETS%20SOCKET

Detect 2

1. Source of the trace
   http://www.incidents.org/logs/Raw/2002.5.26

2. Detect was generated by:
   Snort 2.0.0 (Build 72) using the rule set included in the download from http://www.snort.org/dl/binaries/win32/snort-2_0_0.exe on 5/3/2003.

Snort Rule generating detect
alert tcp any any -> any 80 (content: "scripts"; "cmd.exe"; "c+"; "dir"; msg:"Unicode";)

This rule will alert on any IP address/Any port going to any Ipaddress/port 80 with the content in the payload containing the strings “scripts”, “cmd.exe”, “c+” and “dir”. When a matching packet is found an alert message with the heading Unicode is generated.

Alerts
[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.194488 66.12.252.156:3239 -> 46.5.180.133:80
TCP TTL:110 TOS:0x0 ID:36226 IpLen:20 DgmLen:99 DF
AP Seq: 0x53533A5F Ack: 0xCB737F28 Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.194488 66.12.252.156:3241 -> 46.5.180.135:80
TCP TTL:110 TOS:0x0 ID:36228 IpLen:20 DgmLen:99 DF
AP Seq: 0x5354CE9D Ack: 0xCBFD4334 Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.194488 66.12.252.156:3240 -> 46.5.180.134:80
TCP TTL:110 TOS:0x0 ID:36230 IpLen:20 DgmLen:99 DF
AP Seq: 0x535433BB Ack: 0xCC2026DF Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.224488 66.12.252.156:3251 -> 46.5.180.145:80
TCP TTL:110 TOS:0x0 ID:36232 IpLen:20 DgmLen:99 DF
AP Seq: 0x535BB34C Ack: 0x68CC095F Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.234488 66.12.252.156:3259 -> 46.5.180.153:80
TCP TTL:110 TOS:0x0 ID:36234 IpLen:20 DgmLen:99 DF
AP Seq: 0x53623894 Ack: 0xCBD84011 Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.234488 66.12.252.156:3257 -> 46.5.180.151:80
TCP TTL:110 TOS:0x0 ID:36236 IpLen:20 DgmLen:99 DF
AP Seq: 0x5360B7DB Ack: 0xCB6370F8 Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.244488 66.12.252.156:3264 -> 46.5.180.158:80
TCP TTL:110 TOS:0x0 ID:36238 IpLen:20 DgmLen:99 DF
AP Seq: 0x5366161A Ack: 0xDE4A316E Win: 0x4470 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.794488 66.12.252.156:3239 -> 46.5.180.133:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:98
AP Seq: 0x53533A9A Ack: 0x0 Win: 0x0 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:06:59.874488 66.12.252.156:3257 -> 46.5.180.151:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:98
AP Seq: 0x5360B816 Ack: 0x0 Win: 0x0 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:07:00.234488 66.12.252.156:3259 -> 46.5.180.153:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:98
AP Seq: 0x536238CF Ack: 0xCBD84B79 Win: 0x0 TcpLen: 20

[**] [1:0:0] Unicode [**]
[Priority: 0]
06/26-15:07:06.334488 66.12.252.156:4175 -> 46.5.180.250:80
TCP TTL:110 TOS:0x0 ID:37949 IpLen:20 DgmLen:99 DF
AP Seq: 0x5628713C Ack: 0x6B475705 Win: 0x4470 TcpLen: 20

2. Probability the source address was spoofed:
Very low. It very likely that this source address was not spoofed. This type of reconnaissance/compromise requires that response be sent back.

3. Description of Attack:

Microsoft Internet Information Server (IIS) allows remote users to do a directory listing, view and delete files, and execute arbitrary commands by using the Unicode character representation of the path and command in the URL.

http://xxx.xxx.xxx.xxx/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:\
The above URL would get a command prompt and do a directory listing and return the result to the browser.

CVS CAN-2000-0884
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVS+CAN-2000-0884

http://www.securityfocus.com/bid/2909/help/
http://packetstormsecurity.nl/papers/general/IISUnicodeExplained.doc

5. Attack Mechanism:
Unicode characters can be used to craft URLs to access local resources of a Internet Information Server that would normally be denied. This happens because of a flaw in when the directory names are converted. IIS decodes the UNICODE after it checks the path.

6. Correlations:
Microsoft Security Bulletin
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp

CVS CAN-2000-0884
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884

IIS Unicode Translation Vulnerability Remediation Resources
http://www.cit.cornell.edu/computer/security/scanning/windows/iis_unicode.html

7. Evidence of Active Targeting:

The IP addresses that are being scanned seem to be in random order however the time stamps indicate that this is an automated scan which just sends the same malicious URL to any IP address within a particular subnet looking for any web server that may be vulnerable and return a directory listing. This could just be a random scan looking for any Unicode vulnerable web servers in a range if IP addresses.

8. Severity:

   Severity will be calculated with the following formula:

   severity = (criticality + lethality) – (system countermeasures + network countermeasures)

   Each value will be ranked on a scale from 1 (lowest) to 5 (highest).

   Criticality = 1
   There was no specific system targeted.

   Lethality = 3
   If this attack was successful, it could lead to a compromise at the system level thereby opening up other possible exploits to gain administrator access.

   System countermeasures = 1
   Unsure of any counter measures in place.

   Network countermeasures = 1
   Unsure of any counter measures in place.

   severity = (1 + 5) – (1 + 1)
   severity = 4

9. Defensive Recommendations:

Apply appropriate patches to all web servers. The IIS patches are located at: http://www.microsoft.com/windows2000/downloads/critical/q269862/default.asp

Only allow web traffic leaving your network that has originated from internal web servers designated to be accessible from outside the firewall.

Have an IDS to identify any packets that have contents that contain Unicode in a URL.

10. Multiple choice Question:

What command cannot be successfully run using the Unicode exploit on a vulnerable system ?

   a. dir

b. del
c. net start
d. telnet

Answer - D - Unicode commands are one time shell commands, not interactive connections.

Detect 3

1. Source of the trace

http://www.incidents.org/logs/Raw/2002.5.25
Although the file is named 2002.5.25, the timestamps indicate that the packets are from 06/24 and 06/25.

2 Detect was generated by:
Snort 2.0.0 (Build 72) using the rule set included in the download from http://www.snort.org/dl/binaries/win32/snort-2_0_0.exe on 5/3/2003.

```
[**] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
06/25-10:43:27.094488 24.206.159.155:0 -> 46.5.184.162:0
TCP TTL:49 TOS:0x0 ID:18609 IpLen:20 DgmLen:40
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x30A2  TcpLen: 12
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+=+

[**] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
06/25-10:43:27.094488 24.206.159.155:0 -> 46.5.184.162:0
TCP TTL:49 TOS:0x0 ID:18609 IpLen:20 DgmLen:40
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x30A2  TcpLen: 12
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+=+

[**] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
06/25-10:43:30.104488 24.206.159.155:0 -> 46.5.184.173:0
TCP TTL:49 TOS:0x0 ID:18610 IpLen:20 DgmLen:40
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x30A2  TcpLen: 12
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+=+

[**] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
06/25-10:43:30.104488 24.206.159.155:0 -> 46.5.184.173:0
TCP TTL:49 TOS:0x0 ID:18610 IpLen:20 DgmLen:40
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x30A2  TcpLen: 12
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+=+

[**] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
06/25-10:43:36.074488 24.206.159.155:0 -> 46.5.184.248:0
TCP TTL:49 TOS:0x0 ID:18611 IpLen:20 DgmLen:40
```

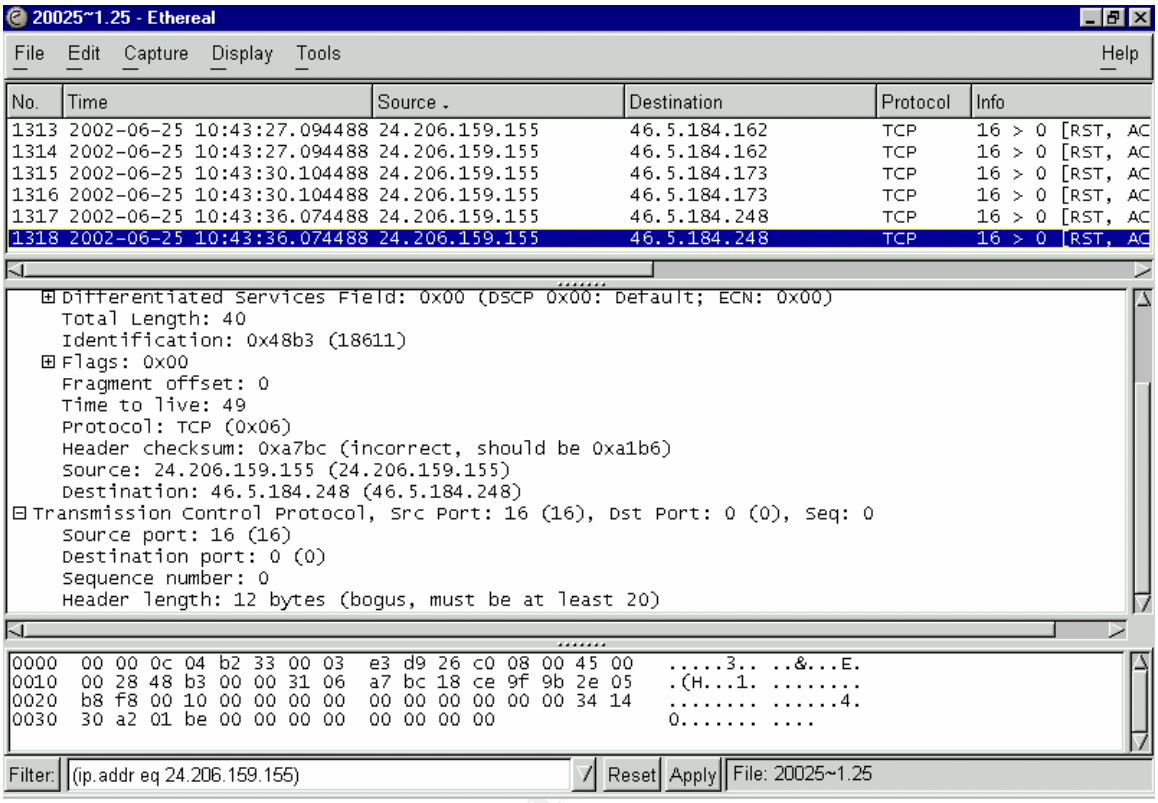
```
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x30A2  TcpLen: 12
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+=+

[**] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
06/25-10:43:36.074488 24.206.159.155:0 -> 46.5.184.248:0
TCP TTL:49 TOS:0x0 ID:18611 IpLen:20 DgmLen:40
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x30A2  TcpLen: 12
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+=+=+=+=+
```

3. Probability the source address was spoofed:
Very low. It very likely that this source address was not spoofed. This type of reconnaissance requires that response be sent back. In this case, ICMP error messages or responses from ports which will help in the reconnaissance.

4. Description of Attack:

OS fingerprinting. Commonly used to help determine the operating system. This works because there are some systems where port 0 is invalid and the response generated will be different then the response when you connect a normal closed port. Some scans set the TCP Options to different or unusual combinations. When the operating system can be determined, the attacker can now concentrate on the exploits known to that system. Also the alerts reported by Snort indicate a source port of 0, however when examining the packets in Ethereal, the packets show as source port 16. This is constant through all the packets.



5. Attack Mechanism

Traits of the packets seem to make me think they are crafted. With that in mind, one of the things that I would consider would be operating system reconnaissance. Specifically because the source port is reported by Snort as port 0 and by Ethereal as port 16 and the destination port is reported as port 0 by both programs. Certain operating systems treat port 0 differently. The port 0 probe could be a form of OS fingerprinting. Also the packets have the RST and ACK bits sent which could both illicit a particular response back to the sender, again for OS fingerprinting and/or may also aid in circumventing a firewall.

6. Correlations:

Port 0, TCP SYN/RST scans
http://honor.trusecure.com/pipermail/firewall-wizards/2001-May/010706.html

Remote OS detection via TCP/IP Stack Fingerprinting
http://www.insecure.org/nmap/nmap-fingerprinting-article.html
Monitor and inspect network activities for unexpected behavior.
http://www.cert.org/security-improvement/practices/p094.html

7. Evidence of Active Targeting

There is some evidence of active targeting as the small number of packets and small number of hosts that received these identical packets. These systems may be public servers, which may help explain why they might be targeted for reconnaissance.

8. Severity

Severity will be calculated with the following formula:

severity = (criticality + lethality) – (system countermeasures + network countermeasures)

Each value will be ranked on a scale from 1 (lowest) to 5 (highest).

Criticality = 3
There were 3 systems targeted. The systems primary functions are unknown.

Lethality = 3
There is no indication of a successful attack however the successful reconnaissance could lend itself to a more critical level.

System countermeasures = 1
Unsure of any counter measures in place.

Network countermeasures = 2
There is an IDS or some other packet collecting device in place.

severity = (3 + 3) – (1 + 2)
severity = 3

9. Defensive Recommendations

Recommend the border router block all ingress packets to port 0.
Recommend firewall block all inbound packets to packets to port 0.
Recommend IDS rule alert for packets to port 0

10. Multiple choice Question:

What tool can be used for OS fingerprinting to port 0 ?

a. Nmap
b. NBTscan
c. Ethereal
d. All of the above

Answer-A - Nmap

Part 3 Analyze This

Executive Summary

The following is a security audit that was done on 5 days worth of log files supplied by University X. These files were categorized into 3 areas, Alerts, Scans and Out of Spec.
By using different tools to analyze the vast amount of data provided in the logs(approximately 610 megabytes), by categorization of key indicators and anomalies of normal traffic patterns, we were able to drill down and identify threats that need to be addressed.

After analyzing and researching the supplied log files, it can be concluded that the perimeter defenses need to be improved to provide a proper defensive posture and lower the risk of compromise. Working with the limited supplied documentation and logs, the following overview was broken into 3 parts:

External Audit – concerning traffic and access external of the University.
DMZ Audit – concerning traffic and access to public university servers.
Internal Audit – concerning internal traffic and usage patterns.

External:
There seems to be little or no perimeter protection in place. There were several scans and active reconnaissance being initiated from outside sources against hosts on the internal network. Application of access control lists should be applied to all perimeter routers as well as firewall rules that identify and allow only authorized traffic.

DMZ:
Web servers are historically difficult to protect just by their very nature of open access. There seems to be a lot of web traffic as well as iFolder traffic that needs to have a more defined source/address path and port that a firewall can provide. This would also provide the ability to react to specific threats by monitoring defined thresholds.

Internal:
As with any University, there seems to be a lot of file sharing. Programs like Kazaa and other file sharing programs can be a security risk. An approved security policy defining accepted traffic should be distributed concerning all Internet access and software.

File List
The sets of file used in this analysis were:

Alerts

Alert.021014
Alert.021015
Alert.021016
Alert.021017
Alert.021018

OOS
OOS_Report_2002_10_14_21815.txt
OOS_Report_2002_10_15_13854.txt
OOS_Report_2002_10_16_32106.txt
OOS_Report_2002_10_17_23248.txt
OOS_Report_2002_10_18_15331.txt

Scans
scans.021014
scans.021015
scans.021016
scans.021017
scans.021018

These log files were acquired from http://www.incidents.org.
(Note: all log files were concatenated into 1 file for each log group.)

Analysis

Without having a network map provided to me, I have used the logs to try to determine the function or role that the high activity hosts seen in the initial log assessment perform and any insight to the behavior observed in those logs :

1) MY.NET is probably 130.85.

After reviewing the log files, it became clear that the scan files were indicating the obfuscated MY.NET network in the alert files were probably the class B network 130.85.0.0. This seems to be the case as all the entries in the scan files (5,256,268 rows) contain a source or destination in the 130.85.0.0 network. Also for further correlation, the following are 2 excerpts from the Alerts and Scans files that further verify this. This type of correlation was repeated often in the logs. Throughout this paper MY.NET has been changed to 130.85 except where results from a script contain MY.NET, in that case the results are not altered.
Excerpt 1
From the Scans file
Jun 2 00 - 00 - 17 130.85.219.18 - 64858 -> 219.94.89.67 - 27262 FIN *******F
Jun 2 00 - 00 - 17 130.85.219.18 - 64873 -> 68.33.98.208 - 6346 SYN ******S*

Jun 2 00 - 00 - 17 130.85.219.18 - 64874 -> 216.222.3.34 - 8112 SYN ******S*
Jun 2 00 - 00 - 18 130.85.219.18 - 64869 -> 207.184.18.15 - 6347 SYN ******S*
Jun 2 00 - 00 - 18 130.85.219.18 - 64863 -> 66.167.201.16 - 6346 SYN ******S*

From the Alerts file
06/02-00:16:46.698926 [**] spp_portscan: portscan status from MY.NET.219.18: 5 connections across 5 hosts: TCP(5), UDP(0) STEALTH [**]

Excerpt 2
From The Scans file
Jun 2 00 - 00 - 19 130.85.98.70 - 1025 -> 46.202.16.223 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.133 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.135 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1029 -> 137.158.91.165 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.139 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1025 -> 46.202.16.226 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.142 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1027 -> 26.153.71.106 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1029 -> 137.158.91.172 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.156 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1027 -> 26.153.71.111 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.157 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1029 -> 137.158.91.174 - 137 UDP
Jun 2 00 - 00 - 19 130.85.98.70 - 1026 -> 148.223.121.161 - 137 UDP
Jun 2 00 - 00 - 20 130.85.98.70 - 1026 -> 148.223.121.231 - 137 UDP
Jun 2 00 - 00 - 20 130.85.98.70 - 1026 -> 148.223.121.239 - 137 UDP
Jun 2 00 - 00 - 20 130.85.98.70 - 1029 -> 137.158.91.204 - 137 UDP

From the Alerts file
06/02-00:16:50.363371 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.98.70 (THRESHOLD 12 connections exceeded in 0 seconds) [**]

2) MY.NET.1.3 is probably a recursive name server.

As per RFC819:[1]

> "The name service at each domain is assumed to be provided by one or more name servers. There are two models for how a name server completes its work, these might be called "iterative" and "recursive".
>
> For an iterative name server there may be two kinds of responses. The first kind of response is a destination address. The second

[1] http://www.faqs.org/rfcs/rfc819.html

kind of response is the address of another name server. If the response is a destination address, then the query is satisfied. If the response is the address of another name server, then the query must be repeated using that name server, and so on until a destination address is obtained."

For a recursive name server there is only one kind of response -- a destination address. This puts an obligation on the name server to actually make the call on another name server if it can't answer the query itself.

The source port of the DNS server seems to have been set to 32832, which is a feature that was incorporated into Bind. Some regard this as a security feature. [2] Correlation of this seems to be provided by:

a) The Top Talker in the destination list in the Scans file is going to 192.26.92.30 port 53 on which is a Top Level Domain server.

b) 130.94.6.10, which is number 5 in the Top Talker destination list, which is a name server from http://www.bondedsender.org, which is an organization that provides real-time black list lookup in the form of DNS.

3) MY.NET.1.4 and MY.NET.137.7 are probably recursive name servers very similar to the first.

4) The user of host 130.85.150.101, seems to be a gamer. From the alert log we have identified 109579 instances of this host going to 20 different IP addresses but all to port 666[3] as shown in the following table. Port 666 is a known port for the game Doom.

IP Address	Count of Visits
12.222.221.8	2990
194.100.203.66	5819
206.62.130.12	5207
212.120.67.18	14607
212.19.205.97	4907
213.10.131.1	6152
213.10.131.21	394

[2] http://www.intac.com/~cdp/cptd-faq/section2.html#ports
[3] http://www.iss.net/security_center/advice/Exploits/Ports/666/default.htm

213.119.124.206	7488
213.119.3.226	15934
213.17.73.127	6978
213.201.183.6	836
213.37.125.71	1307
216.235.129.197	5912
217.120.57.127	7332
217.204.26.100	4231
217.37.14.211	1066
217.44.47.50	7187
62.173.117.178	1
62.195.123.101	4034
81.68.153.106	7197
Total Visits:	109579

Unique IP Addresses: 20

5) IRC (port 6667)[4] is active from a few hosts. The following table shows the 5 hosts with the highest IRC activity. MY.NET.190.95 seems to be under some type of attack, possibly a Denial of Service, as evidenced by the very high amount of activity.

Source	
my.net.198.221	6959
my.net.91.151	1364
my.net.97.20	808
my.net.132.24	61
my.net.105.204	32

Destination	
my.net.190.95	19433
my.net.114.116	760
my.net.105.204	228
my.net.91.151	95
my.net.83.48	50

6) The host 130.85.87.70 seems to be extremely active on ports 7674, 22321, and 445. This seems to indicate active reconnaissance with a high level of activity as shown in the tables below by the large number of destination IP addresses when compared to the small number of source addresses.

[4] http://www.seifried.org/security/ports/6000/6667.html

Source port	Number of Occurrences
7674	84238
22321	23465

Dest port	Number of Occurrences
7674	84228
22321	23179

Unique destination IP addresses 34627
Unique source IP addresses 53

7) The host at 130.85.97.160 is shown to have a lot of activity. Out of 65605 entries in the scan logs, 65170 were scanning to TCP port 17300 which would seem to indicate that the host may have been compromised by the Kuang2TheVirus.[5]

8) The host at 130.85.153.223 has a high level of activity on UDP port 6257 which is common to the P2P file sharing program WinMX.[6]

9) The host at 130.85.218.90 has a high amount of traffic to UDP port 41170 which indicate the P2P file sharing program Blubster.[7]

10) The host at 130.85.97.41 could possibly be a proxy server or may be infected with the Code Red virus.[8] The behavior that this host exhibits, is to connect to a high number of hosts on port 80. Out of 53309 connections in the scans file made by this host, 53241 were made to port 80 on 43731 unique hosts.

11) On host 130.85.70.225, out of a total of 49089 source ports extracted from the scans file, 38948 are from port 5671. Although there were no references to this port found in my research, a thorough capture and analysis of the content of the traffic to and from this host should be done. Bearing in mind it is in the top ten of the Scans file and has a high number of Unique Destination IP Addresses at 25617.

12) 130.85.100.165 is a web server. Specifically, the computer science and electrical engineering web server of University of Maryland, Baltimore County.

13) 130.85.30.4 is a Novell 6 Web server. This explains the unusual ports [9]associated with this web servers traffic. On this web server each service is given a different port when there is only one IP address for all the services.

[5] http://vil.mcafee.com/dispVirus.asp?virus_k=10213
[6] http://lists.insecure.org/lists/firewall-wizards/2001/Sep/0038.html
[7] http://www.blubster.net/php/article.php?sid=25
[8] http://www.cert.org/advisories/CA-2001-19.html
[9] http://www.tek-tips.com/gfaqs.cfm/pid/871/fid/3352

14)130.85.224.134 may be compromised. Possibly by the Sobig.a virus which installs a proxy on a non standard port.[10]

List of Detects
The following consist of all alerts that have greater than 1 occurrence.

Signature	# Alerts
SMB Name Wildcard	866729
CS WEBSERVER - external web traffic	63209
MY.NET.30.4 activity	46645
spp_http_decode: IIS Unicode attack detected	21970
[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	21169

[10] http://www.derkeiler.com/Mailing-Lists/securityfocus/incidents/2003-04/0112.html

[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	18052
EXPLOIT x86 NOOP	12388
High port 65535 tcp - possible Red Worm - traffic	11789
spp_http_decode: CGI Null Byte attack detected	11093
Queso fingerprint	6764
SYN-FIN scan!	6665
High port 65535 udp - possible Red Worm - traffic	5793
Tiny Fragments - Possible Hostile Activity	4718
Incomplete Packet Fragments Discarded	2714
TCP SRC and DST outside network	2380
MY.NET.30.3 activity	2366
CS WEBSERVER - external ftp traffic	2362
IDS552/web-iis_IIS ISAPI Overflow ida nosize	1887
[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	1477
SNMP public access	1420
Possible trojan server activity	940
Null scan!	927
SUNRPC high port access!	777
IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize	407
NMAP TCP ping!	374
[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.	341
NIMDA - Attempt to execute cmd from campus host	140
Notify Brian B. 3.54 tcp	113
SMB C access	113
Notify Brian B. 3.56 tcp	100
EXPLOIT x86 stealth noop	82
IRC evil - running XDCC	67
EXPLOIT x86 setuid 0	63
EXPLOIT x86 setgid 0	37
TFTP - Internal TCP connection to external tftp server	34
FTP passwd attempt	30
TFTP - Internal UDP connection to	19

external tftp server	
connect to 515 from outside	19
RFB - Possible WinVNC - 010708-1	14
Probable NMAP fingerprint attempt	13
Attempted Sun RPC high port access	8
External FTP to HelpDesk MY.NET.70.49	5
NETBIOS NT NULL session	5
TFTP - External UDP connection to internal tftp server	4
External FTP to HelpDesk MY.NET.70.50	4
EXPLOIT NTPDX buffer overflow	4
[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	2
ICMP SRC and DST outside network	2
TFTP - External TCP connection to internal tftp server	2

SMB Name Wildcard	# Alerts:866729

Severity Low

Summary: This is an information gathering attack. It takes advantage of the NetBios name table retrieval query. On a Windows Network, Netbios is used to help resolve workstation names, domains, and users who are currently logged in. As you can see from the following top 5 addresses table, there is a great deal of these coming from outside the network.

Top 5 Source address		Top 5 Destination address	
209.52.45.162	16533	my.net.24.34	6863
164.77.209.245	13748	my.net.12.2	3831
164.77.209.124	13729	my.net.29.11	2318
164.77.209.100	13506	my.net.218.2	2297
195.101.253.232	8803	my.net.24.44	2093

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: CANADA
NOTE: More information appears to be available at NET-209-52-45-0-1.
TELUS Communications Inc. TELAC-BLK6 (NET-209-52-0-0-1)
209.52.0.0 - 209.52.255.255
RCMP RCMP-CA (NET-209-52-45-0-1)
209.52.45.0 - 209.52.45.255
ARIN WHOIS database, last updated 2003-06-13 21:05
Enter ? for additional hints on searching ARIN's WHOIS database

Correlation
http://www.whitehats.com/info/IDS177
The whitehats.com intrusion event database entry.

http://www.cert.org/incident_notes/IN-2000-02.html
A cert advisory that briefly explains how a worm can exploit this vulnerability.

Recommendations

1) An ingress and egress filter should be set up at the border routers to drop incoming or outgoing NetBios packets with a destination port of 137 and 445.
2) A firewall should be setup to drop incoming or outgoing NetBios packets with a destination port of 137 and 445.

CS WEBSERVER - external web traffic	#alerts-63209

Severity Low

Summary: This seems to be normal web traffic destined for the Computer Science Web server at 130.85.100.165. There are 63209 references to this alert which seems to be in place, not for intrusion detection, but possibly for some type of logging of external web traffic. Below are the top 5 visitors and the number of times visited. The top user is most likely a search bot for Alta Vista to catalog the site for their search engine as referenced in the whois below..

IP Address	Count of Visits
Total Visits:	63372
216.39.48.2	6105
66.77.73.236	2492
65.49.178.17	433
64.124.5.10	315
66.196.72.110	238

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: UNITED STATES

NOTE: More information appears to be available at OA36-ARIN.

OrgName: AltaVista Company
OrgID: ALTAVI-1
Address: 1070 Arastradero Rd
City: Palo Alto
StateProv: CA
PostalCode: 94304
Country: US

NetRange: 216.39.48.0 - 216.39.63.255
CIDR: 216.39.48.0/20
NetName: NETBLK-INTERNET-BLK-1-AV
NetHandle: NET-216-39-48-0-1
Parent: NET-216-0-0-0-0
NetType: Direct Assignment
NameServer: NS1.ALTAVISTA.COM
NameServer: NS2.ALTAVISTA.COM
NameServer: NS3.ALTAVISTA.COM
Comment:
RegDate: 2002-09-09

Updated: 2002-09-09

TechHandle: OA36-ARIN
TechName: ALtaVista, Operations
TechPhone: +1-650-320-7700
TechEmail: netops@av.com

OrgAbuseHandle: ABUSE129-ARIN
OrgAbuseName: Abuse
OrgAbusePhone: +1-650-320-7700
OrgAbuseEmail: abuse@av.com

OrgTechHandle: OA36-ARIN
OrgTechName: ALtaVista, Operations
OrgTechPhone: +1-650-320-7700
OrgTechEmail: netops@av.com

ARIN WHOIS database, last updated 2003-06-14 21:05
Enter ? for additional hints on searching ARIN's WHOIS database

Correlation
http://www.giac.org/practical/Stan_Hoffman_GCIA.doc

http://www.giac.org/practical/Scott_Baird_GCIA.doc

Recommendations

This does not seem to a threat at this time. No further action needs to be taken.

MY.NET.30.4 activity	#Alerts-46645

Severity Low

Summary: This seems to be normal web traffic destined for the Novell internal Web server at 130.85.30.4. These alerts, as with the previous web server, seem to be in place, not for intrusion detection, but possibly for some type of logging of web traffic and services. Below are the top 5 visitors and the number of times visited. The whois search of the top 5 seem to indicate that these are home users in the greater Maryland and Virginia area accessing Novell iFolder. However the number 1 in the top 5 seems to be reporting the IP address as 68.49.35.0. Upon further analysis, it becomes evident that the ports this non address is accessing, are usually used by Novell for different services.

Top 5 Users

IP Address	Count of Visits
Total Visits:	13416
68.49.35.0	6100
68.33.11.236	1189
66.168.226.143	545
151.196.48.241	315
172.129.244.94	206

Top 2 ports

Port
51443
524

From Novells Website

[11]:

' The default port number for NetWare Enterprise Server is 80 for HTTP and 443 for HTTPS. If
you have NetWare Enterprise Server installed, by default the Apache Web Server will get port 51080 for HTTP and 51443 for HTTPS'

[12]:

TCP 524 - NCP Requests - Source port will be a high port (1024-65535)
UDP 524 - NCP for time synchronization - Source port will be a high port

[11] http://support.novell.com/servlet/tidfinder/2963227
[12] http://www.novell.com/coolsolutions/netware/features/a_ports_nw5_nw.html

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: UNITED STATES

NOTE: More information appears to be available at NET-68-48-0-0-1.

Comcast Cable Communications, Inc. JUMPSTART-1 (NET-68-32-0-0-1)
68.32.0.0 - 68.63.255.255
Comcast Cable Communications, Inc. DC-3 (NET-68-48-0-0-1)
68.48.0.0 - 68.49.255.255

ARIN WHOIS database, last updated 2003-06-14 21:05
Enter ? for additional hints on searching ARIN's WHOIS database.

Correlation:

http://www.novell.com/products/ifolder/

http://www.novell.com/coolsolutions/netware/features/a_ports_nw5_nw.html

Recommendations:

Although this does not seem to be a threat at this time. Action needs to be taken to explore the reason behind the ip address in the logs that ends with a zero. This may be a function of Novell services, however if that is the case, then this needs to be verified and documented and a procedure put in to place to identify these users.

spp_http_decode: IIS Unicode attack detected	#Alerts-21970

Severity Low

Summary: This attack is done by manipulation of URL encoding. By using escape and Unicode characters it is possible to have the request misinterpreted by the server and allow unauthorized access. When Snort runs this input through the HTTP_DECODE preprocessor, the decoded result is then matched against the signatures. This creates a lot of false positive and as per the Snort faq[13] "Your own internal users normal surfing can trigger these alerts in the Preprocessor" There were a total of 759 unique IP addresses that triggered this alert.
This alert is generally known to have a lot of false positives[14]. With UNICODE, there could be multiple representations of a single character. With this in mind and all the different URL's requested, it is easy to see how a legitimate web server request could be mistaken for an alert trigger. The more granular the IDS rule, there more of a chance that an actual attack may get by and the more broad based the rule will mean many more false positives.

The table that follows contains the top 5 IP addresses that caused these alerts. Note that they are all internal hosts. The number 1 destination whois lookup follows the table:

Top 5 Source hosts		Top 5 Destination Hosts	
my.net.75.107	899	202.129.15.124	388
my.net.84.216	858	217.228.142.57	233
my.net.97.79	636	61.243.175.241	193
my.net.91.2	531	211.90.88.43	60
my.net.217.102	439	66.250.68.41	32

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: THAILAND (high)
ARIN says that this IP belongs to APNIC; I'm looking it up there.
Using cached answer (or, you can get fresh results).
% [whois.apnic.net node-2]
% How to use this server http://www.apnic.net/db/
% Whois data copyright terms http://www.apnic.net/db/dbcopyright.html
inetnum: 202.129.0.0 - 202.129.31.255
netname: CAT
descr: Communication Authority of Thailand, CAT
descr: International Telecommunications Service Provider

[13] http://www.snort.org/docs/FAQ.txt
[14] http://www.mcabee.org/lists/snort-users/May-01/msg00691.html

country: TH
admin-c: TK38-AP
tech-c: SK79-AP
mnt-by: APNIC-HM
mnt-lower: MAINT-TH-THIX-CAT
changed: hostmaster@apnic.net 20000914
status: ALLOCATED PORTABLE
source: APNIC

person: Tanussit Klaimongkol
address: Data Comm. Dept.(Internet)
address: CAT Bangkok 10501
address: Thailand
country: TH
phone: +66-2-2374300
fax-no: +66-2-5063186
e-mail: ktanus@cat.net.th
nic-hdl: TK38-AP
mnt-by: MAINT-TH-THIX-CAT
changed: ktanus@cat.net.th 20000215
source: APNIC

person: Serthsiri Khantawisoote
address: Data Communication Department, CAT
address: Bangkok 10501
country: TH
phone: +66-2-237-4300
fax-no: +66-2-506-3186
e-mail: kserth@cat.net.th
nic-hdl: SK79-AP
mnt-by: MAINT-TH-THIX-CAT
changed: hostmaster@apnic.net 20000320
source: APNIC

Correlation

http://www.sans.org/resources/idfaq/anomaly_detection.php

http://www.securityfocus.com/infocus/1232

Recommendations:

1) All vulnerable web servers should be patched.
2) Consider a reverse proxy that translates all web requests to ascii and runs the through a content filter for easier identification and another layer of security. Defense in depth.

[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	#Alerts-21169

Severity High

Summary: This seems to be a denial of service attack possibly emanating from a trojaned host. Usually /kill command removes a person from an IRC server. However, that person can come back on. Usually this command comes from the server operators. However it has been noted that:[15]

> "With the advent of auto-reconnecting clients KILL is almost totally worthless as a tool for punishment. More effective methods to deal with obnoxious people are the IGNORE, KICK and various MODE's on channels, such as +i and +b."

The biggest offender here triggering these alerts by far is 66.207.164.23 and the address receiving most of this traffic is my.net.190.95 as evidenced in the following table. If this is a legitimate IRC channel, it may be receiving spoofed packets, as there seems to be no traffic coming from MY.NET.190.95 to 66.207.164.23.

Top 5 Source and Destination addresses from this alert

Source Addresses		Destination Addresses	
66.207.164.23	19463	my.net.190.95	19432
216.152.64.155	684	my.net.114.116	760
212.161.35.251	94	my.net.97.15	566
195.159.0.81	84	my.net.91.151	94
195.159.0.82	82	my.net.97.76	78

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: UNITED STATES
NOTE: More information appears to be available at JM3108-ARIN.
Using cached answer (or, you can get fresh results).
OrgName: ColoGuys
OrgID: CLGY
Address: 8101 Chapin Road
City: Fort Worth
StateProv: TX
PostalCode: 76116
Country: US
NetRange: 66.207.160.0 - 66.207.175.255
CIDR: 66.207.160.0/20
NetName: COLOGUYS-1

[15] http://vorlon.ces.cwru.edu/~tyger/irctalk/irc3.html

NetHandle: NET-66-207-160-0-1
Parent: NET-66-0-0-0-0
NetType: Direct Allocation
NameServer: NS1.COLOGUYS.COM
NameServer: NS2.COLOGUYS.COM
NameServer: NS3.COLOGUYS.COM
Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate: 2001-12-20
Updated: 2001-12-27

TechHandle: JM3108-ARIN
TechName: Montroll, Jon
TechPhone: +1-817-560-0305
TechEmail: Noc@cologuys.com

ARIN WHOIS database, last updated 2003-06-11 21:05
Enter ? for additional hints on searching ARIN's WHOIS
Correlation

http://www.valinor.sorcery.net/docs/rfc2812/3.7.1-kill-message.html

http://www.edge-zone.net/irc/bots.html

http://www.russonline.net/tonikgin/EduHacking.html

Recommendations

1) Stop all traffic from 66.207.164.23 at the border routers.

2) Stop all traffic from 66.207.164.23 at the firewalls.

3) Alert the admin of 66.207.164.23 to a possible compromised system and send cleaned logs as validation.

[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	#Alerts-18052

Severity Medium

Summary: XDCC is a file transfer mechanism that lets users download predetermined files from a IRC user or bot. The clients main purpose is to monitor file-sharing channels for XDCC offers. The interface can be similar to other file-sharing programs such as WinMX and KaZZa. However, the XDCC client will only monitor XDCC offers made in IRC channels. The software that these clients can download should be considered suspect. They can be trojanized versions of software and some of the XDCC clients themselves are known to be trojanized.

Top 5 Sources		Top 5 Destinations	
my.net.83.100	9702	208.194.163.37	7016
my.net.198.221	6976	205.188.149.12	6969
my.net.91.151	1368	212.161.35.251	1366
my.net.105.204	32	155.207.19.204	1031
my.net.83.173	9	196.38.143.228	1008

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: UNITED STATES
NOTE: More information appears to be available at NET-208-194-160-0-1.
UUNET Technologies, Inc. UUNET1996B (NET-208-192-0-0-1)
208.192.0.0 - 208.255.255.255
First Internet Alliance UU-208-194-160 (NET-208-194-160-0-1)
208.194.160.0 - 208.194.167.255
ARIN WHOIS database, last updated 2003-06-14 21:05
Enter ? for additional hints on searching ARIN's WHOIS database

Correlation
http://www.kvirc.de/docu/doc_dcc_connection.html
http://security.duke.edu/cleaning/xdcc.html

Recommendations

1) Block all ingress access to ports 6667 and 6668 at the border routers.
2) Block all incoming access to ports 6667 and 6668 at the firewall.
3) Evaluate the top 5 internal hosts for compromise and illegal software

EXPLOIT x86 NOOP	12388

Severity Low

Summary: This type of an attack tries to take advantage of services that may be coded in an unsafe, no error checking manner. When that service receives data it cannot handle, usually crafted packets padded until the buffer overflows and runs shellcode[16], This is done by overwriting the return address, and putting the address of another memory segment in and execute our code there. This alert creates a lot of false positive.[17]

Top 5 Sources		Top 5 Destinations	
80.212.2.4	3865	my.net.110.224	3435
80.178.68.208	1414	my.net.86.19	1436
213.10.134.115	1114	my.net.114.116	1341
62.178.50.12	962	my.net.198.235	575
209.216.96.136	798	my.net.106.222	544

The following Whois id from www.dnsstuff.com for the address with the most alerts, 80.212.2.4.

```
Country: NORWAY
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      80.212.0.0 - 80.212.255.255
netname:      NO-NEXTRA-ADSL-1
descr:        Telenor Business Solution AS
country:      NO
admin-c:      SI217-RIPE
tech-c:       TRR5-RIPE
tech-c:       TBS-RIPE
status:       ASSIGNED PA
remarks:      - - - - - - - - - - - - - - - - - - - - - - - - - - - -
remarks:      - - For abuse matters, mailto: abuse@telenor.net - - -
remarks:      - - - - - - - - - - - - - - - - - - - - - - - - - - - -
notify:       ripe-contacts@telenor.net
mnt-by:       AS8210-MNT
mnt-lower:    AS8210-MNT
mnt-routes:   AS8210-MNT
changed:      thk@nextra.com 20020102
changed:      thk@nextra.com 20020325
```

[16] http://www.enderunix.org/docs/en/sc-en.txt
[17] http://www.snort.org/snort-db/sid.html?id=648

```
changed:      thk@telenor.net 20020814
changed:      thk@telenor.net 20030401
source:       RIPE

route:        80.212.0.0/17
descr:        TELENOR-INTERNET
descr:        Nextra, Postboks 393 - Skoyen, N-0212 Oslo, Norway
origin:       AS2119
mnt-by:       AS8210-MNT
changed:      hso@nextra.com 20020214
source:       RIPE
role:         Telenor Routing Registry
address:      Telenor Business Solutions AS
address:      Snaroeyveien 30
address:      N-1331 Fornebu
address:      Norway
phone:        +47 22 77 19 00
fax-no:       +47 22 77 19 10
e-mail:       as-guardian@telenor.net
admin-c:      HSO3-RIPE
tech-c:       HSO3-RIPE
tech-c:       TNA4-RIPE
tech-c:       DF344-RIPE
tech-c:       THK-RIPE
nic-hdl:      TRR5-RIPE
notify:       as-guardian@telenor.net
mnt-by:       AS8210-MNT
changed:      thk@nextel.no 19990119
changed:      tna@nextel.no 19991012
changed:      tna@nextel.no 19991027
changed:      thk@nextra.com 20000411
changed:      tna@nextra.com 20000516
changed:      tna@nextra.com 20020610
changed:      tna@telenor.net 20020730
changed:      tna@telenor.net 20020731
source:       RIPE
role:         TBS AS - Customer Internet Access
address:      Telenor Business Solutions AS
address:      N-1331 Fornebu
address:      Norway
phone:        +47 67 89 00 00
e-mail:       ia-tech@telenor.net
admin-c:      RG737-RIPE
tech-c:       EAO-RIPE
nic-hdl:      TBS-RIPE
remarks:      - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

remarks: - - Please send abuse reports to abuse@telenor.net - -
remarks: -
notify: ia-tech@telenor.net
mnt-by: TNXHM-MNT
changed: eao@telenor.net 20021029
changed: eao@telenor.net 20030314
source: RIPE
person: Sigbjorn Isene
address: Telenor Networks AS
address: Snaroyveien 30
address: N-1331 Fornebu
address: Norway
phone: +47 67 89 00 00
e-mail: sigbjorn.isene@telenor.com
nic-hdl: SI217-RIPE
mnt-by: AS8210-MNT
changed: si@nextel.no 19980526
changed: thk@nextra.com 20011205
changed: tna@telenor.net 20030508
source: RIPE

Correlation
http://cert.uni-stuttgart.de/archive/intrusions/2003/03/msg00111.html

http://www.enderunix.org/docs/en/sc-en.txt

Recommendations

1) As this is an alert with a high number of false positives, it must be said that there are also a high number of exploits. The alert logs should be reviewed on a regular basis and with more emphasis on the top 5 hosts that trigger this alert.

High port 65535 tcp - possible Red Worm - traffic	#Alerts-11789

Severity High

Summary:
According to CERT[18] :

[18] http://www.cert.org/advisories/CA-2001-23.html

“The "Code Red" worm is malicious self-propagating code that exploits Microsoft Internet Information Server (IIS)-enabled systems susceptible to the vulnerability described in CA-2001-13 Buffer Overflow In IIS Indexing Service DLL. Its activity on a compromised machine is time sensitive; different activity occurs based on the date (day of the month) of the system clock. The CERT/CC is aware of at least two major variants of the worm, each of which exhibits the following pattern of behavior:”

Propagation mode (from the 1st - 19th of the month): The infected host will attempt to connect to TCP port 80 of randomly chosen IP addresses in order to further propagate the worm. Depending on the configuration of the host that receives this request, there are varied consequences

Flood mode (from the 20th - 27th of the month): A packet-flooding denial-of-service attack will be launched against a specific IP address embedded in the code.

Termination (after the 27th day): The worm remains in memory but is otherwise inactive.

From the following tables, it becomes evident that we have unusual activity involving the same 2 internal hosts and 3 external hosts in both of the top 5 lists.

Top 5 Source addresses		Top 5 Destination addresses	
my.net.24.47	4997	192.207.69.1	4996
192.207.69.1	3731	my.net.24.47	3730
my.net.70.210	1151	my.net.70.210	1226
210.194.244.45	705	210.194.244.45	687
61.120.129.119	523	61.120.129.119	463

The following Whois id from www.dnsstuff.com for the external address with the most alerts.
Country: UNITED STATES
NOTE: More information appears to be available at DA754-ARIN.
OrgName: MacNeal-Schwendler Corp.
OrgID: MACNEA
Address: 815 Colorado Blvd.
City: Los Angeles
StateProv: CA
PostalCode: 90041
Country: US
NetRange: 192.207.69.0 - 192.207.72.255
CIDR: 192.207.69.0/24, 192.207.70.0/23, 192.207.72.0/24
NetName: NETBLK-MACSCH
NetHandle: NET-192-207-69-0-1

Parent: NET-192-0-0-0-0
NetType: Direct Assignment
NameServer: ZANGARRA.MACSCH.COM
NameServer: NS2.MSCSOFTWARE.COM
Comment:
RegDate: 1992-08-20
Updated: 2001-09-18
TechHandle: DA754-ARIN
TechName: Dns Admins, Dns
TechPhone: +1-714-445-3169
TechEmail: dns.admins@mscsoftware.com

ARIN WHOIS database, last updated 2003-06-15 21:05
Enter ? for additional hints on searching ARIN's WHOIS database.

Correlation

http://www.cert.org/advisories/CA-2001-23.html

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

Recommendations

1) Take suspected systems offline until they can be cleaned and patched.
2) Patch all vulnerable systems to avoid further compromises.
   Windows NT version 4.0:
   http://www.microsoft.com/Downloads/Release.asp?ReleaseID=30833
   Windows 2000 Professional, Server and Advanced Server:
   http://www.microsoft.com/Downloads/Release.asp?ReleaseID=30800

3) The perimeter firewall should have a low threshold of traffic coming from port 65535 with proper alerts when that threshold is exceeded.

Spp_http_decode: CGI Null Byte attack detected	#Alerts-11093

Severity Low

Summary: This attack is done by manipulation of URL encoding. By using escape and Unicode characters it is possible to have the request misinterpreted by the server and allow unauthorized access. When Snort runs this input through the HTTP_DECODE preprocessor, the decoded result is then matched against the signatures. This creates a lot of false positive and as per the Snort faq[19] "Your own internal users normal surfing can trigger these alerts in the Preprocessor"

The table that follows contains the top 5 IP addresses that caused these alerts. Note that they are all internal hosts going to external servers. The number 1 destination whois lookup follows the table:

Top 5 Sources		Top 5 Destination	
my.net.97.29	3078	203.161.233.132	7402
my.net.97.92	2497	192.151.53.10	537
my.net.97.97	843	143.48.220.86	446
my.net.122.65	671	32.97.212.77	250
my.net.97.81	573	63.83.249.103	228

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: HONG KONG

ARIN says that this IP belongs to APNIC; I'm looking it up there.
Using cached answer (or, you can get fresh results).
% [whois.apnic.net node-1]
% How to use this server http://www.apnic.net/db/
% Whois data copyright terms http://www.apnic.net/db/dbcopyright.html
inetnum: 203.161.224.0 - 203.161.255.255
netname: ILINK
descr: iLink.net Limited
descr: Facility Management, Hong Kong
country: HK
admin-c: OO4-AP
tech-c: OO4-AP
mnt-by: APNIC-HM
mnt-lower: MAINT-HK-ILINK
changed: hostmaster@apnic.net 20000112

[19] http://www.snort.org/docs/FAQ.txt

status: ALLOCATED PORTABLE
source: APNIC

person: operator operator
address: 56/F The Center,
address: 99 Queen's Road Central,
address: Hongkong
country: HK
phone: +852-31231588
fax-no: +852-22182288
e-mail: ipadmin@ilink.net
nic-hdl: OO4-AP
mnt-by: MAINT-HK-ILINK
changed: ipadmin@ilink.net 19991230
source: APNIC

Correlation
http://www.giac.org/practical/GCIA/Doug_Kite_GCIA.pdf

http://www.giac.org/practical/GCIA/Brian_Coyle_GCIA.pdf

http://www.technicalinfo.net/papers/URLEmbeddedAttacks.html

Recommendations

1) All vulnerable web servers should be patched.
2) Consider a reverse proxy that translates all web requests to ascii and runs the through a content filter for easier identification and another layer of security. Defense in depth.

Queso fingerprint	#Alerts=6764

Severity Low

Summary: OS fingerprinting[20] is generally done for reconnaissance to identify the OS so that further exploits can be attempted. While generally not of a high

[20] http://www.insecure.org/nmap/nmap-fingerprinting-article.txt

severity, it does indicate active targeting and hosts perpetrating this should be identified and reported.

Top 5 sources		Top 5 Destinations	
66.117.30.14	2250	my.net.233.78	1139
193.219.55.20	187	my.net.224.134	1111
213.186.35.9	147	my.net.6.40	513
216.95.201.25	138	my.net.24.23	512
196.26.86.133	116	my.net.24.21	500

The following Whois id from www.dnsstuff.com for the address with the most alerts.
Country: UNITED STATES

NOTE: More information appears to be available at HOSTM44-ARIN.

OrgName: New Horizon Collocations
OrgID: NHC-34
Address: 603 Wilshire
Address: Suite 911
City: Los Angeles
StateProv: CA
PostalCode: 90017
Country: US

NetRange: 66.117.0.0 - 66.117.31.255
CIDR: 66.117.0.0/19
NetName: NHI-COLO
NetHandle: NET-66-117-0-0-1
Parent: NET-66-0-0-0-0
NetType: Direct Allocation
NameServer: DNS1.NHISCOLO.COM
NameServer: DNS2.NHICOLO.COM
Comment:
RegDate: 2002-09-30
Updated: 2002-11-11

OrgTechHandle: HOSTM44-ARIN
OrgTechName: HOSTMASTER
OrgTechPhone: +1-877-322-5188
OrgTechEmail: noc@nhicolo.com

ARIN WHOIS database, last updated 2003-06-15 21:05
Enter ? for additional hints on searching ARIN's WHOIS database

Correlation
http://www.insecure.org/nmap/nmap-fingerprinting-article.txt

http://www.giac.org/practical/Joe_Rayford_GCIA.doc

Recommendations
1) Use stateful inspection firewalls.

2) Alter or disable banners of accessible services.

Top Ten Of Each Group

Src

IP Address	Count of Visits
Total Visits:	5256268
130.85.1.3	926117
130.85.150.101	228199
130.85.87.80	107763
130.85.97.160	65545
130.85.153.223	63151
130.85.218.90	55512
130.85.97.41	53309
130.85.70.225	48911
218.131.136.85	47666
218.121.222.49	47067

Dst

Total Visits:	5256268
192.26.92.30	38149
213.119.3.226	33488
212.120.67.18	30585
192.148.252.171	26401
130.94.6.10	19007
194.109.6.154	17351
205.231.29.244	15862
213.119.124.206	15513
217.120.57.127	15222
81.68.153.106	14796

Alerts

Src

IP Address	Count of Visits
Total Visits:	1115830
68.170.69.138	26354
66.207.164.23	19466
209.52.45.162	16546
164.77.209.245	13751
164.77.209.124	13730

164.77.209.100	13508
my.net.83.100	9685
68.49.35.0	8850
195.101.253.232	8803
128.210.176.203	7020

Dst

IP Address	Count of Visits
Total Visits:	1115830
my.net.100.165	66054
my.net.30.4	46653
my.net.190.95	19452
203.161.233.132	7402
my.net.24.34	7180
208.194.163.37	7009
205.188.149.12	6959
192.207.69.1	5001
my.net.12.2	3891
my.net.24.47	3805

OOS

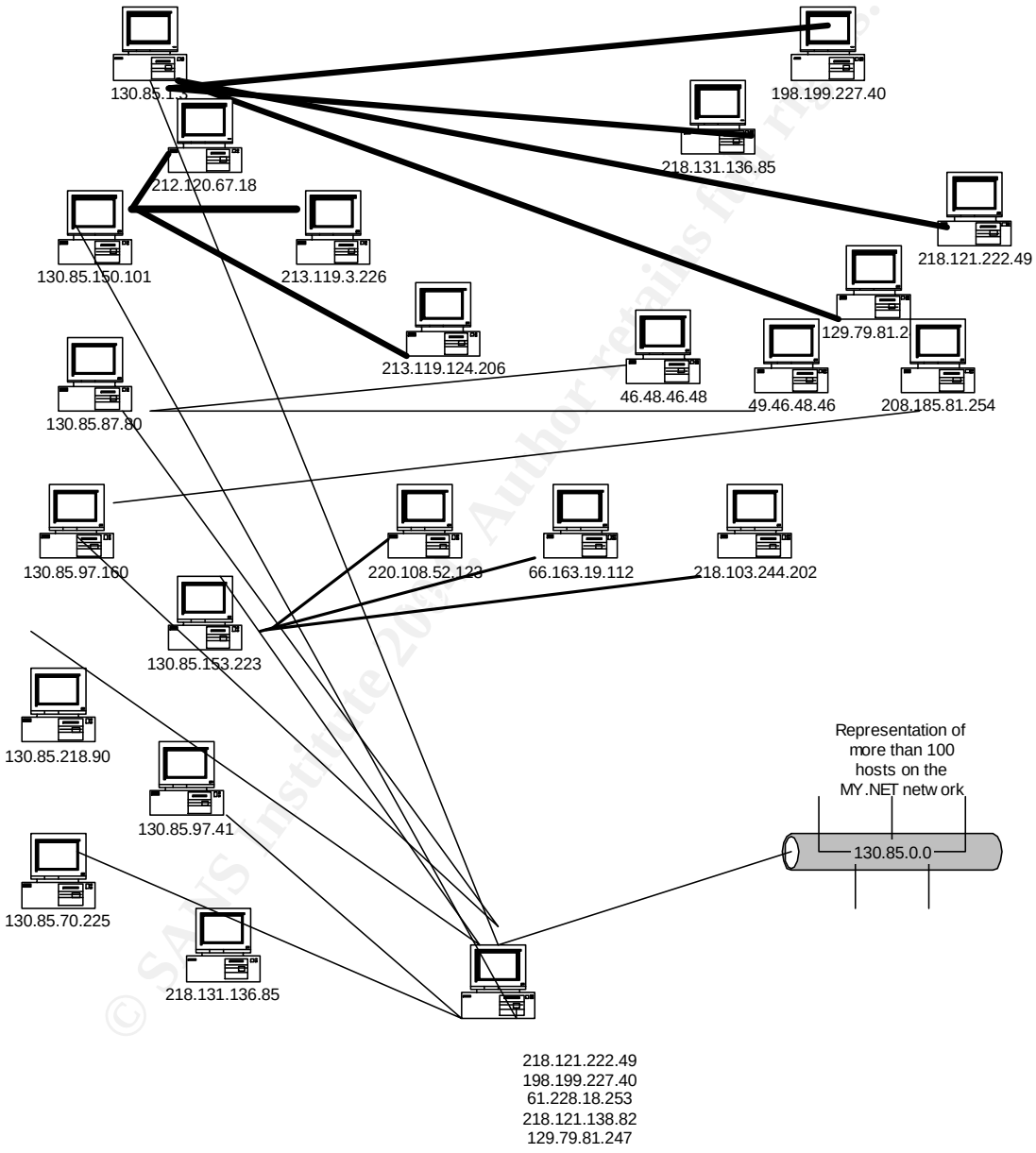
Src

IP Address	Count of Visits
Total Visits:	25577
66.117.30.14	8159
193.219.55.20	714
148.63.164.189	591
213.186.35.9	445
216.95.201.25	400
210.253.206.180	346
216.95.201.32	338
216.95.201.24	334
216.95.201.20	289
216.95.201.30	282

Dst

IP Address	Count of Visits
Total Visits:	25577
my.net.224.134	4091
my.net.233.78	4068
my.net.218.2	1752
my.net.6.47	1725

my.net.24.22	1686
my.net.6.40	1686
my.net.24.21	1656
my.net.24.23	1610
my.net.24.44	1010
my.net.240.202	591



Recommendations:
Although our recommendations have been stated above, we will take this time to summarize them.

The following IP addresses on the MY.NET network should have a traffic analysis as well as a thorough port scan to further identify or verify any anomaly's in behavior. Next to the IP address is a suggestion of possible compromise or behavior that we feel indicates this analysis.

MY.NET.190.95	Possible IRC attack
MY.NET.87.70	Unusually high activity several unknown ports
MY.NET.97.160	Possible Kuang infection
MY.NET.153.223	Possible file sharing
MY.NET.218.90	Possible file sharing
MY.NET.97.41	Possible code red
MY.NET.70.225	Unusually high activity port 5671 unknown
MY.NET.224.134	Possible sobig.a
MY.NET.24.34	Possible SMB
MY.NET.12.2	Possible SMB
MY.NET.29.11	Possible SMB
MY.NET.218.2	Possible SMB
MY.NET.24.44	Possible SMB
MY.NET.83.100	Possible XDCC
MY.NET.198.221	Possible XDCC
MY.NET.91.151	Possible XDCC
MY.NET.105.204	Possible XDCC
MY.NET.83.173	Possible XDCC
MY.NET.24.47	Possible Code red
MY.NET.70.210	Possible Code red

Also the University's perimeter needs to have a defined policy, enforced by access control lists on all of the border routers as well as a more restrictive firewall policy and a plan to restrict what seems to be a wide open policy on file sharing, irc usage, and web site access. Also rollout of content monitoring and filtering software at a proxy level should be explored.

Analysis Process

All the log files were combined into 1 file for the 3 categories. Then our analysis employed Snortsnarf for a general overview of the alerts. With the sheer volume of data that would have to be analyzed, I quickly realized that tools like grep, cat, ws, and perl would be indispensable. After quite a while of looking for scripts on the Internet, I discovered one that I would modify and use over and over. I employed this script with a program called Ultraedit, which can handle large files without too much of a problem and also offers a great expression set for data manipulation. I also made use of CYGWIN to have access to those great Unix tools on an X86 platform that Windows lacks. Using these tools in a variety of modified ways, I was able to tame the huge amount of data into a more readily usable form. The Perl script in its original state can be found at

http://216.239.39.100/search?q=cache:XrDucFY1FV4J:www.neinfo.net/newsletter/archive96/vol1_18.htm+perl+count+%22unique+ip+address%22&hl=en&ie=UTF-8

Here is the script in its original form:

```
open (LOG, "d:/web/lessons/log.txt") || die ("Cannot open log file!");
$line = <LOG>;
chop($line);
until ($line eq "")
{
if ($line =~ /^[0-9]+./) #this will make sure that the line begins with numbers.
{
$line =~ tr/A-Z/a-z/; #convert to lowercase
@parts = split(/ /, $line); #break up the line by the spaces
$IPs{$parts[0]} += 1; #create an associative array to count each address
}
$line = <LOG>;
chop($line);
}

We have now read the file. It is a simple
matter now to print out the data. Lets use a
table.

print &PrintHeader;
print "<html><head><title>Simple Counter</title></head><body>\n";
print "<TABLE BORDER=1>\n";
print "<tr><th>IP Address</th><th>Count of Visits</th></tr>\n";
foreach $address (sort keys %IPs)
{
```

```
$cnt++; # to count the Unique addresses
$visits = $visits + $IPs{$address}; #count visits
print "<tr><td>$address</td><td align=right>$IPs{$address}</td></tr>\n";
}
print "<tr><td>Total Visits:</td><td align=right>$visits</td></tr>\n";
print "</TABLE>\n";
print "Unique IP Addresses: $cnt<br>\n";
print "</body></html>\n";
```

Ultraedit can be found at
www.ultraedit.com

Cygwin can be found at

www.cygwin.com

I made extensive use of cat, ws and grep to mine certain data from the log files. And then I created files containing the mined data and modified the Perl script to extract certain fields of the massaged data files and present them in HTML format. I then cut the data from the browser and pasted it into Excel where I used the sorting feature to present the data in the order needed.

References

Data collection mechanisms for intrusion detection systems
http://www.cerias.purdue.edu/homes/zamboni/pubs/2000-08.pdf

Network- vs. Host-based Intrusion Detection
http://documents.iss.net/whitepapers/nvh_ids.pdf

Learn Host-Based Intrusion Detection
http://www.informit.com/isapi/product_id~%7BEBB19669-502C-451D-8CA8-9C1F9F5A17B5%7D/content/index.asp

Intrusion Detection FAQ
http://www.sans.org/resources/idfaq/host_based.php

Going on the Defensive: Intrusion-Detection Systems
http://www.informit.com/isapi/product_id~%7B7BE5789A-7A3D-4C63-A403-1FC262E9470E%7D/content/index.asp

Watching Your Logs
http://www.spitzner.net/swatch.html

Building an Intrusion-Detection System
to Detect Suspicious Process Behavior
http://www.raid-symposium.org/raid99/PAPERS/Wespi.pdf

Computer Immune Systems
http://www.cs.unm.edu/~immsec/papers.htm

Linux Security Tools
http://linas.org/linux/secure.html

The LIDS Project
http://www.lids.org/about.html

Could you please explain Intrusion Detection technology
http://www.itsecurity.com/asktecs/jan1402.htm

Last Updated: May 30th, 2017

Upcoming Training

Security Operations Center Summit & Training	**Washington, DC**	**Jun 05, 2017 - Jun 12, 2017**	**Live Event**
SANS Houston 2017	**Houston, TX**	**Jun 05, 2017 - Jun 10, 2017**	**Live Event**
SANS Columbia, MD 2017	**Columbia, MD**	**Jun 26, 2017 - Jul 01, 2017**	**Live Event**
SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	**Washington, DC**	**Jul 24, 2017 - Jul 29, 2017**	**vLive**
SANS San Antonio 2017	**San Antonio, TX**	**Aug 06, 2017 - Aug 11, 2017**	**Live Event**
SANS Boston 2017	**Boston, MA**	**Aug 07, 2017 - Aug 12, 2017**	**Live Event**
SANS Adelaide 2017	**Adelaide, Australia**	**Aug 21, 2017 - Aug 26, 2017**	**Live Event**
SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS vLive - SEC503: Intrusion Detection In-Depth	**SEC503 - 201709,**	**Sep 11, 2017 - Oct 18, 2017**	**vLive**
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**vLive**
SANS Baltimore Fall 2017	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS London September 2017	**London, United Kingdom**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
Community SANS Scottsdale SEC503	**Scottsdale, AZ**	**Oct 02, 2017 - Oct 07, 2017**	**Community SANS**
SANS October Singapore 2017	**Singapore, Singapore**	**Oct 09, 2017 - Oct 28, 2017**	**Live Event**
Community SANS Boston SEC503	**Boston, MA**	**Oct 09, 2017 - Oct 14, 2017**	**Community SANS**
Community SANS Ottawa SEC503	**Ottawa, ON**	**Oct 16, 2017 - Oct 21, 2017**	**Community SANS**
SANS Seattle 2017	**Seattle, WA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS San Diego 2017	**San Diego, CA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	**San Diego, CA**	**Oct 30, 2017 - Nov 04, 2017**	**vLive**
SIEM & Tactical Analytics Summit & Training	**Scottsdale, AZ**	**Nov 28, 2017 - Dec 05, 2017**	**Live Event**
SANS Cyber Defense Initiative 2017	**Washington, DC**	**Dec 12, 2017 - Dec 19, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**