Global Information Assurance Certification Paper

GIAC Certified Intrusion Analysts (GCIA) v. 3.4

IPS, IDS what is the next step?

Andrew J. Wagoner

17-May-04

Table of Contents

Part 1: The State of Intrusion Detection Today

Summary

In this paper, it is the authors intent, to discuss the various facets that surround perimeter defense/detection technologies with the purpose of dissolving the perceived gray area that separate IDS and IPS perimeter detection devices and their applied applications. It is important to distinguish the differences between IDS and IPS to ascertain proper placement and use of each device in relation to

the network environment. Due to the amount of confusion, an attempt will be made to define the functions each of these two devices performs. Additionally, the intent of this paper is to provide a brief history of IDS and IPS, and demonstrate the advantages and disadvantages of each. Furthermore, we will conclude with how perimeter detection has evolved, as well as try to determine the future of perimeter detection as it pertains to IDS/IPS.

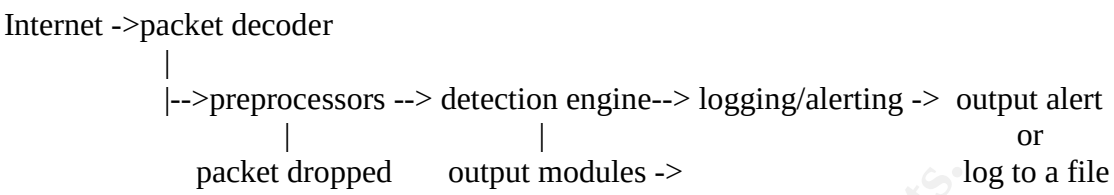
Introduction

The Internet of today is full of as much information as a person would like to know. Search engines such as Google, Yahoo, MSN and others provide avenues for beginners and advanced users a like to search this vast entity of bits and bytes that make up the information they seek. As we all know the Internet can provide colossal amounts of information for the simplest of things to the most advanced topics. Anything from how to build a bomb to how to how to get from point "A" to point "B" on the globe is available. Included in this ever-changing storehouse of information are details required to exploit the weaknesses in the technology of the Internet. It seems almost daily that more applications are created to increase productivity and enhance our digital lives. Additionally, of the new applications created, an equal amount of details are released regarding exploitation of weaknesses in these applications. As the Internet grows it becomes increasingly more difficult to protect our data from being compromised by attackers who exploit these vulnerabilities. There are so many products and services to choose from that offer the protection that we desire; how do we choose the right combination?

What is IDS and how does it work?

What is a NIDS (Network Intrusion Detection System) and how does it work? A NIDS is typically a passive device that monitors packets traversing the network. The most common way companies applied intrusion detection devices was by creating a baseline for the normal activity on the network. This would include information regarding user logins, access to internal/external services etc. The IDS will then alert based upon anything activity that goes outside of the baseline. For example, a system administrator who monitors several systems for the company he or she works for could review the logs from the previous day. The administrator may notice that a workstation normally in use during business hours was attempting to gain access to other systems across the network at 3am. This could be classified as anomaly detection as it does not fall with in the confines of "normal" activity. Other ways to apply intrusion detection is by signature matching. The IDS watches the traffic watching for known patterns of attack. For example the IDS could examine each packet that traverses the wire looking for 'codex' in the payload. This could indicate an attempted exploit of a well-known Microsoft IIS vulnerability. The IDS would trigger an alert as it matched a pre-defined signature in the rule set. IDS devices have several

signatures for the various known types of attacks so how does the IDS match the traffic to the signatures?
The chart below demonstrates how the IDS process works.[1]



An IDS is able to capture and decode packets that pass across the wire. The packet decoder prepares the packets to be preprocessed. Depending on the preprocessor, it will modify the data before the detection engine processes it, some preprocessors can find anomalies in packet headers and generate an alert. The next step is to send the packet to the detection engine to detect intrusion activity. In Snort this is done from a rule base, the rule base can be thought of as a "bulleted list". As the packet comes in it is compared to the rules in the list. If it matches any of the rules an alert is generated or it is logged, depending on the rule matched. If no rules are matched no action is taken against the packet and it passes freely.

One of the largest problems with signature based IDS's are false positives. A false positive occurs when the non-malicious data in a packet matches a pre-defined signature and triggers an alert. For example, some broadband routers often times broadcast UPnP (Simple Service Discovery Protocol) requests on port 1900. A default Snort configuration could trigger an alert on this type of traffic due to the data in the packet matches a signature in the database. While this "could" be a valid attempt to exploit a known vulnerability in several versions of the Microsoft Operating System[2] often times it is nothing more than the router advertising its services to different hosts on the network.

Another problem with signature based IDS is that the signatures must be updated regularly. Without diligent administration, the IDS could easily miss a new exploit because it does not have information regarding the new attack. In short it only detects what you tell it to detect and nothing more. It could also miss a variation on a known attack in this manner. Other issues surrounding signature based IDS is the ability for attackers to evade the signatures by changing the attack method of a known exploit. As mentioned, the attacker could put a different spin on the attack that does not match known signatures and thereby avoid generating an alert. Again, the IDS only alerts on things the administrator tells it to alert on. Additionally, the more granular the signature base is, the more processing power it takes to decode the packets. This could lead to packet loss when maximum bandwidth is reached because the IDS simply cannot keep up. Packet loss could allow for an attack slip by unnoticed.

[1] Taken from (Intrusion Detection with Snort, Rafeeq UR Rehman. Prentice Hall PTR, 2003 p. 12)
[2] http://www.microsoft.com/technet/security/bulletin/ms01-059.mspx

One proposed solution to the issues surrounding signature-based IDS's is anomaly-based IDS as mentioned briefly above, or protocol analysis. Anomaly detection grabs all of the IP headers as they come in, filters out the acceptable traffic such as acceptable web, email, DNS, etc., and triggers an alert for traffic that does not fit within the pre-defined baseline. This significantly reduces the amount of data to be analyzed resulting in fewer false positives being generated. One of the advantages of anomaly-based IDS is that it is very sensitive to recon attempts such as ping sweeps, TCP | UDP scans or operating system identification. These attempts are a common pre-cursor to exploitation of vulnerabilities.

Protocol Analysis

In protocol analysis the IDS examines the entire "conversation" of packets as opposed to single packets in the signature based IDS. This technique compares the traffic with a representation of the protocol and the way it is supposed to be used based on the RFC information and the normal traffic associated with it. Protocol analysis looks at all of fields in the datagram and matches them to known acceptable values. Traffic that is outside the confines of this model of normal traffic is considered malicious content at which point an alert is generated. Protocol analysis has its own set of problems as well. For example, not all software application vendors strictly adhere to current protocol standards. While the application may be functioning in a non-malicious way, it could be viewed as "out of spec" according to the model of acceptable traffic by the IDS. Since the traffic does not match what is considered to be legitimate or normal traffic an alert is generated.

There are many that say IDS is dead, in my opinion is that IDS not dead, only enhanced. Signature based IDS, anomaly and protocol analysis based IDS are all very valuable tools to alert to a potential or an ongoing attack. However, they only alert, often times by the time the alert is generated a host has been compromised and the damage is already done. Alerting to an attack is a positive thing. Keeping in line with the defense in depth model, simply alerting is no longer enough.

What is IPS and how does it work?

Those that say IDS is dead are aggressively pushing for the Intrusion Prevention System (IPS). The most current generation of IDS is a hybrid between anomaly- and signature-based systems. There are those that argue that to be effective, the IDS had to block attacks as they are detected. This hybrid, or enhancement of the last generation IDS is known as IPS. What is IPS and how does it work? The technical definition of IPS according to Wikipedia is as follows:

> "An **Intrusion-prevention system** (a computer security term) is used to actively drop packets of data or disconnect connections that contain

> unauthorised data. Intrusion-prevention technology is also commonly an extension of intrusion detection technology (IDS)."

As stated the majority on engineers and security professionals in this industry agree that alerting is not enough. As the alert is generated something needs to be done to prevent the potential for attack. In a recent article from R. Stiennon[3] states,

> " IDS vendors that have not introduced blocking capabilities by the end of 2004 will not be viable providers beyond the end of 2005 (0.9 probability)."

The IPS is capable of performing deep packet inspection on each packet that traverses the wire. This gives the IPS a distinct advantage in being able to detect attacks that are known and unknown as well as being able to prevent a successful attack. One of the more common ways attackers attempt to gain access to a vulnerable host is through a buffer overflow attack. A buffer overflow condition will exist when a buffer, assigned by a programmer to hold variable data, receives more variable data than what has been assigned to the buffer. A buffer is an allocated space in memory used for temporary storage of values required by the application, such as an array or a pointer. In C there are no bounds checking operations that are done automatically. The programmer must deliberately put a function in place to prevent the application from accepting more data into the buffer than has been allocated. Buffer overflows have one thing in common; execute code in writeable area of memory on a vulnerable host. In a quote from Nick Ray[4]

> "If we detect code that attempts to execute in a writable area of memory, we know it is not a process started by the system, therefore it is an attack."

IPS devices must be able to effectively use complex computing methods to detect and block these types of attacks. They must go beyond the signature-based applications such as anti-virus or IDS devices. Additionally, to be completely effective the IPS must be an inline device that does not create any latency in the network. With the development of FPGA, ASICS, and network processors; administrators now have a viable option for true intrusion prevention. These processors allow the device to process traffic with the performance of a switch. Essentially this means that there is no additional load for the network. Now the best of both worlds come together, deep packet inspection and high performance. Since the device can also be configured to fail in an open state, the issue of another point of failure is eliminated. Administrators are able to configure a stateful failover with a secondary IPS device. This means that the devices would continually synchronize state information with each other via dedicated

[3] Research Note for Gartner, Inc. by R. Stiennon April 13, 2004
[4] Nick Ray, chief executive of Prevx
http://news.zdnet.co.uk/internet/security/0,39020375,39118610,00.htm

Ethernet connection between the two IPS's. When one fails the other one kicks in exactly where the primary device left off. Again, one of the hurtles that IPS devices face is being able to tell the difference between attacks and normal traffic. An improperly tuned IPS can generate many false positives similar to an IDS, the only difference being is that an IPS can and will block what it considers to be malicious traffic where the IDS will only alert on it. An IPS can quickly create a self-inflicted denial of service condition on a network.

Future of IPS and IDS

Clearly there are several technological hoops to jump through to make this an effective reliable technology, as was the situation with the IDS. As technology evolves and be comes more advanced in the art of intrusion detection, more advanced, faster, more efficient products will be developed. The thought of leaving one technology and inventing some new "silver bullet" seems somewhat far-fetched. A more realistic approach is to continue along the path that these technologies have been traversing, which is the process of evolution.

It would be incorrect to say that IPS devices are going to make IDS devices completely obsolete. IDS devices still have a valuable function and provide an excellent view of the network and the activity therein. The age-old concept of defense-in-depth still applies. A layered defense always has been and always will be the key to success. A good firewall, an IDS, an IPS, a good Anti-virus solution and consistent adherence to security policy will always be key factors in keeping a network secure. There is not a "one appliance" solution; it is a conglomeration of several technologies and appliances working together. IPS will continue to evolve and grow to become more efficient in being able to stop malicious traffic. IDS will continue to provide analysts with an excellent source of forensic information. History has shown us that technology is an ever-evolving entity.

> "The current state of intrusion detection also includes a variety of techniques for detecting malicious traffic, including stateful pattern matching, protocol anomaly detection, and statistical anomaly detection."[5]

The state of intrusion detection is one of change, evolution and transformation.

References:

Timothy D. Wickham. "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). April 21st, 2003. URL: http://www.sans.org/rr/papers/30/1028.pdf

Brox, Arnt . "Signature Based or Anomaly Based Intrusion Detection – The Practice and Pitfalls." 02 2002. 05 May 2004 URL: http://www.itsecurity.com/papers/proseq1.htm

[5] Timothy D. Wickham GICA April 21st 2003

Graham, Robert. "FAQ: Network Intrusion Detection Systems." 21 2000. 14 May 2004 URL:
http://www.robertgraham.com/pubs/network-intrusion-detection.html#1.1

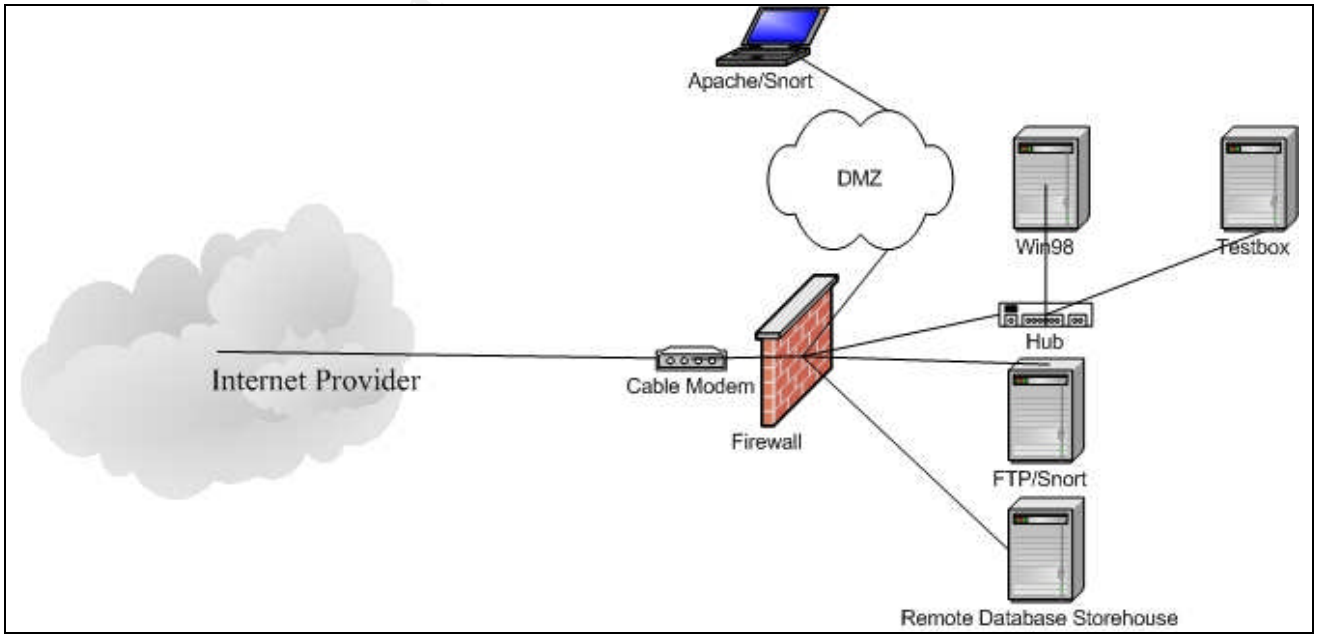
Kotadia, Munir. "Prevx intrusion detection puts agents on desktops."
17 Dec 2003. ZDNet UK. 14 May 2004 URL:
http://news.zdnet.co.uk/internet/security/0,39020375,39118610,00.htm

Desai , Neil. "Intrusion Prevention Systems: the Next Step in the Evolution of IDS." 02 2003. 15 May 2004
URL: http://www.securityfocus.com/infocus/1670

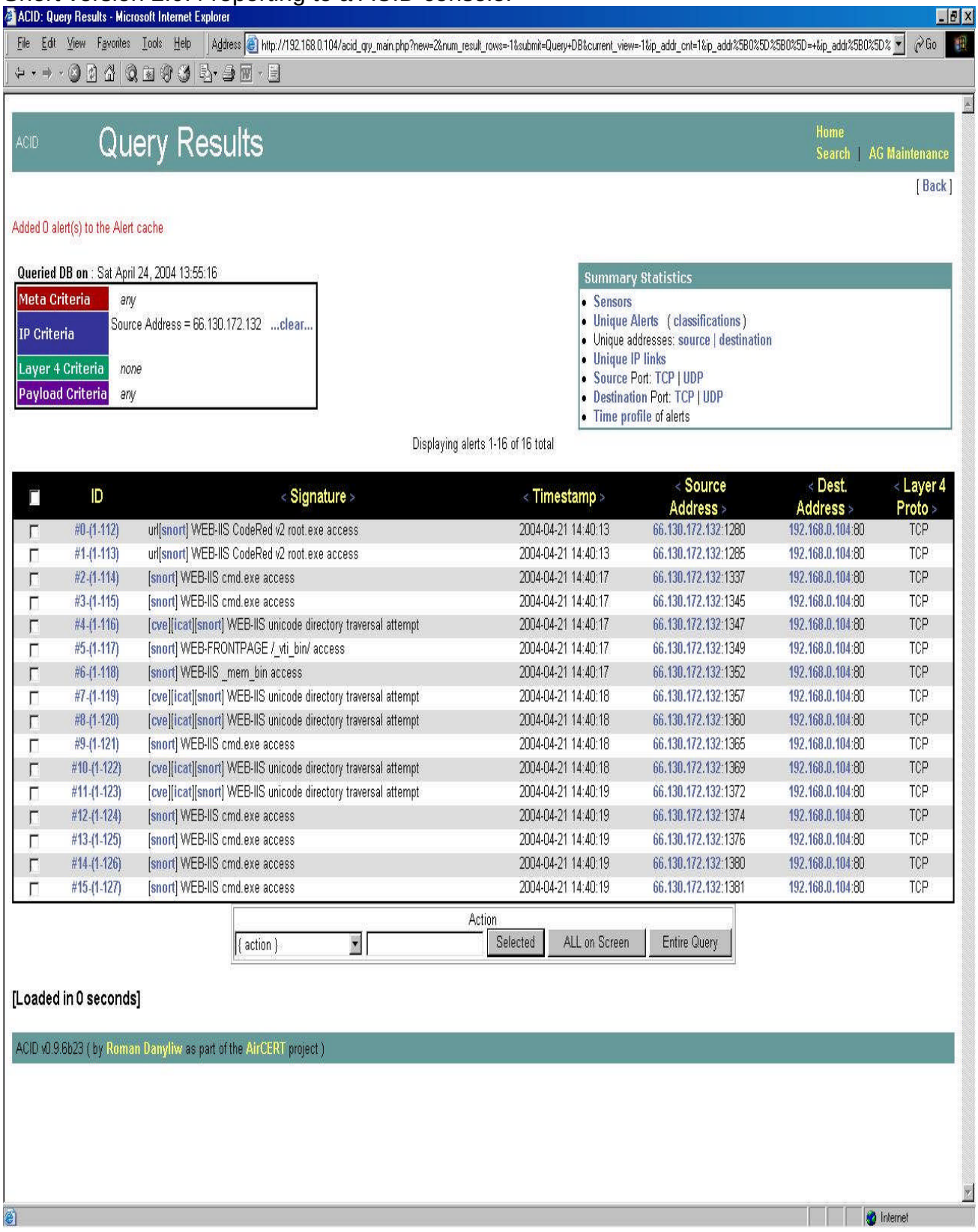
Part 2: Network Detects

Detect 1: Nimda Worm

Source of Trace:
The source of the trace is from the authors network. Method of sniffing traffic is snort. The snort sensor is also an Apache web server and is placed on the DMZ. Network Diagram is pictured below:



Detected Generated with:
Snort version 2.0.4 reporting to a ACID console.



Relevant snort alerts are:

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:13.610819 66.130.172.132:1280 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:11249 IpLen:20 DgmLen:112 DF
AP Seq: 0x206BA68F Ack: 0x207243E0 Win: 0x4470 TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:13.828710 66.130.172.132:1285 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:11303 IpLen:20 DgmLen:110 DF
AP Seq: 0x20703010 Ack: 0x2050E5F4 Win: 0x4470 TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:17.017529 66.130.172.132:1337 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:11950 IpLen:20 DgmLen:120 DF
AP Seq: 0x20A2194C Ack: 0x210FC88F Win: 0x4470 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:17.319495 66.130.172.132:1345 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:11971 IpLen:20 DgmLen:120 DF
AP Seq: 0x20A9176D Ack: 0x21114A4B Win: 0x4470 TcpLen: 20

[**] [1:1945:1] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:17.484527 66.130.172.132:1347 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:11981 IpLen:20 DgmLen:136 DF
AP Seq: 0x20AC10DF Ack: 0x21004DBA Win: 0x4470 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

[**] [1:1288:5] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentially vulnerable web application] [Priority: 2]
04/21-14:40:17.717733 66.130.172.132:1349 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:11994 IpLen:20 DgmLen:157 DF
AP Seq: 0x20AE1652 Ack: 0x204F819F Win: 0x4470 TcpLen: 20

[**] [1:1286:5] WEB-IIS _mem_bin access [**]
[Classification: access to a potentially vulnerable web application] [Priority: 2]
04/21-14:40:17.896253 66.130.172.132:1352 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12004 IpLen:20 DgmLen:157 DF
AP Seq: 0x20B09358 Ack: 0x204D8DB0 Win: 0x4470 TcpLen: 20

[**] [1:982:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:18.145244 66.130.172.132:1357 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12019 IpLen:20 DgmLen:185 DF
AP Seq: 0x20B55AC5 Ack: 0x212D8D6A Win: 0x4470 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

[**] [1:982:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:18.389446 66.130.172.132:1360 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12049 IpLen:20 DgmLen:137 DF
AP Seq: 0x20B8638F Ack: 0x204DDD3A Win: 0x4470 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:18.597674 66.130.172.132:1365 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12096 IpLen:20 DgmLen:137 DF
AP Seq: 0x20BD3A43 Ack: 0x20AB47EA Win: 0x4470 TcpLen: 20

[**] [1:981:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:18.829497 66.130.172.132:1369 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12147 IpLen:20 DgmLen:137 DF
AP Seq: 0x20C100B2 Ack: 0x20BDE0DB Win: 0x4470 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

[**] [1:983:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:19.010281 66.130.172.132:1372 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12158 IpLen:20 DgmLen:137 DF
AP Seq: 0x20C3D23E Ack: 0x210C6BC5 Win: 0x4470 TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:19.192780 66.130.172.132:1374 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12177 IpLen:20 DgmLen:138 DF
AP Seq: 0x20C64CBE Ack: 0x21010470 Win: 0x4470 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:19.355818 66.130.172.132:1376 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12184 IpLen:20 DgmLen:136 DF
AP Seq: 0x20C8D8D5 Ack: 0x20C0BFC7 Win: 0x4470 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:19.554845 66.130.172.132:1380 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12191 IpLen:20 DgmLen:140 DF
AP Seq: 0x20CC35A4 Ack: 0x20C2786A Win: 0x4470 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
04/21-14:40:19.789701 66.130.172.132:1381 -> 192.168.0.104:80
TCP TTL:109 TOS:0x0 ID:12202 IpLen:20 DgmLen:136 DF
AP Seq: 0x20CD8B12 Ack: 0x20DD994C Win: 0x4470 TcpLen: 20

Snort signatures that triggered the alerts:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS CodeRed v2 root.exe access"; flow:to_server,established; uricontent:"/root.exe"; nocase; classtype:web-application-attack; reference:url,www.cert.org/advisories/CA-2001-19.html; sid:1256; rev:7;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS cmd.exe access"; flow:to_server,established; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002; rev:5;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-FRONTPAGE /_vti_bin/ access";flow:to_server,established; uricontent:"/_vti_bin/"; nocase; classtype:web-application-activity; sid:1288; rev:5;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS _mem_bin access"; flow:to_server,established; uricontent:"/_mem_bin/"; nocase; classtype:web-application-activity; sid:1286; rev:5;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to_server,established; content:"/..%255c.."; nocase; classtype:web-application-attack; reference:cve,CVE-2000-0884; sid:1945; rev:1;)

The snort signature alerts on “WEB-IIS <content> access”. As this is a generic alert, this could be one of several other types of attacks. I used tcpdump –r alert –n –nn –X src host 66.130.172.132 –w alert_log, on the snort alert file in /var/log/snort, to get the events from the source host into a separate file. I then ran snort against the alert_log file with the following command line options: snort –r alert > snrt_alt_log.txt to output the events into a text file I could copy in to this document.

Probability the source IP address was spoofed:
Due to the fact that HTTP requests require an established TCP connection, the probability of this IP address being spoofed is low.

Description of the attack:
This attack is most likely a due to the source host 66.130.172.132 being infected with W32.nimda.a@mm or better known as the Nimda worm, it is probing the web server in attempt to propagate itself.

Attack Mechanism:
In this detect the attack was unsuccessful as it only affects Microsoft IIS, Windows 2000, Windows 95, Windows 98, Windows Me, Windows NT, Windows XP hosts. The web server being attacked is running Apache 2.0.49, which is not vulnerable to the exploit attempts, made by Nimda. This worm can spread via email, open windows network shares, and web browsers accessing a host that is infected with Nimda. Microsoft IIS is vulnerable to multiple vulnerabilities, such Extended Unicode Directory Traversal Vulnerability, and Escaped Character Decoding Command Execution Vulnerability among others. Nimda attempts to exploit these vulnerabilities to gain admin access to vulnerable hosts. If Nimda is successful in exploiting one of these vulnerabilities and makes a connection, it tries to exploit other backdoors that may have been left open by the Code Red II virus. The next step is to use TFTP to upload files from the infected host to the newly compromised system in order to perpetuate the cycle. Further more detailed information about Nimda and the Windows vulnerabilities it tries to exploit, can be found at the following links:

- http://www.microsoft.com/technet/Security/topics/virus/nimda.mspx
- http://www.cert.org/advisories/CA-2001-12.html
- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884
- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154
- http://www.cert.org/advisories/CA-2001-26.html
- http://securityresponse1.symantec.com/sarc/sarc.nsf/html/w32.nimda.e@mm.html#technicaldetails

Correlations:
Upon checking www.cert.org after an extensive google and news group search, I was able to confirm that this was indeed Nimda traffic. The following excerpt was taken from http://www.cert.org/advisories/CA-2001-26.html

> System FootPrint
> The scanning activity of the Nimda worm produces the following log entries for any web server listing on port 80/tcp:
> GET /scripts/root.exe?/c+dir
> GET /MSADC/root.exe?/c+dir
> GET /c/winnt/system32/cmd.exe?/c+dir

```
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET
/msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe?/
c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
```

Evidence of Active Targeting:
This attacked occurred only once time in a 10 day time span. The probability of this host is the target of a specific attack is low, a more likely conclusion is that this attack was part of a broad scan. The host was deemed to be unaffected by this attack, the attacker moved on.

Severity:
Severity = (criticality + lethality) - (system countermeasures + network counter measures)

Criticality	This is not a critical piece of this particular network, and the operating system is not vulnerable to the attack.	**0**
Lethality	The web server is not in danger of an attack	**0**
System	No patches are necessary for this particular host as the operating system is not affected by vulnerability.	**5**
Network	The attack was captured by the IDS and prompted further investigation	**5**
Total Score	**(Criticality = 0 + Lethality =0) – (System CM = 5 + Network CM = 5**	**10**

Defensive Recommendation:

1. Verify that all hosts, using Microsoft IIS, are completely patched before being exposed to the Internet for public use.
2. Scan internal Windows host (win98) to verify that it is not running a personal web server and that it is not currently infected with Nimda, Code Red II or any other worm/virus.
3. Verify that the firewall is filtering outbound traffic to catch any hosts that could be infected, attempting to probe for new hosts to infect.

Multiple Choice Question:

The Nimda worm propagates itself by:
A. Sending an email, to the attacker, with the user name and password of a guest account on an infected host.
B. Opening a network share and sending an email to everyone on the internal network with an executable attachment.
C. Connecting to a modem on the infected host and dialing random hosts
D. Exploiting known vulnerabilities in the www.microsoft.com website

The correct answer is A.

Detect 2: Backdoor Q Access

Source of trace:

The source of this trace was from the raw tcpdump log files at: http://www.incidents.org/logs/Raw/2002.10.14 it is unknown what the entire network configuration is.

Detect Generated with:

I generated alerts by first running Snort version 2.0.4 against all of the files from 2002.10.10-18 with the following command:
for i in 2002*; do snort -r $i -c /etc/snort/snort.conf -l alert_log/; done After looking through the alert file in alert_log, I noticed several different types of alerts. The following command allowed me to single out the event for this detect:
tcpdump -r 2002.10.14 -n -nn -X src port 31337 and dst port 515 -w 10.14. Snort was run against the file created by tcpdump with the following command:
snort –r 10.14 –c /etc/snort/snort.conf –d –l backdoor/

The relevant snort alerts I am using for this section are in the table below. The log from http://www.incidents.org indicated that data was from 2002.10.14; however, analysis revealed the dates for the alerts that were generated to be 2002.11.14-18. There were about 143 alerts that snort generated with several destination IP addresses in the 170.129.0.0 class B network in this particular file. I have included 10 of the original alerts in the interest of limiting the space to be used.

Relevant Snort Alerts:

```
[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-10:29:14.826507 255.255.255.255:31337 -> 170.129.172.186:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
```

```
11/14-10:32:53.016507 255.255.255.255:31337 -> 170.129.132.79:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-10:49:26.156507 255.255.255.255:31337 -> 170.129.129.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-11:10:22.596507 255.255.255.255:31337 -> 170.129.195.178:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-11:44:04.836507 255.255.255.255:31337 -> 170.129.30.34:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-12:44:56.156507 255.255.255.255:31337 -> 170.129.137.174:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-13:56:26.746507 255.255.255.255:31337 -> 170.129.89.87:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-14:35:08.896507 255.255.255.255:31337 -> 170.129.200.84:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]
```

```
[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-15:01:20.986507 255.255.255.255:31337 -> 170.129.23.133:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-15:04:08.966507 255.255.255.255:31337 -> 170.129.190.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]
```

Snort signature that triggered the alerts:
alert tcp 255.255.255.0/24 any -> $HOME_NET any (msg:"BACKDOOR Q access"; flags:A+; dsize: >1; stateless; reference:arachnids,203; sid:184; classtype:misc-activity; rev:4;)

Probability the source IP address was spoofed:
The probability of the source IP address being spoofed is high. It is not normal to see traffic coming from 255.255.255.255 as this is a broadcast address. Traffic shouldn't be from this source address. A small blurb from RFC 919 on broadcast addresses says:

> RFC 919 October 1984 Broadcasting Internet Datagrams
> The address 255.255.255.255 denotes a broadcast on a local hardware network, which must not be forwarded. This address may be used, for example, by hosts that do not know their network number and are asking some server for it.

Description of the attack:
The targeted port on the destination host is TCP port 515. This is the port normally used by UNIX for printing services. Given the information in the tcpdump output, the packets are crafted in an attempt to disguise the true source address. It is unclear from this traffic the true intent behind such crafting.

Attack Mechanism:
Q is a utility that functions similar in the way that netcat does such as allowing secure communications between hosts. According to the Q readme file Q is a "remote access and redirection server with strong encryption." Initially this was thought to be a stimulus packet meant for an infected host. Upon further analysis, I have determined this to be a false positive and not an attack. The following tcpdump output shows some oddities in this trace that assisted me in my decision. I did a tcpdump -nevvX -r <filename> src 255.255.255.255 on this trace and found that all of the packets contain "cko" in the payload, this could be the command to elicit a response from an already infected host. However, I was

unable to find any documentation to support this theory. It should be noted that every packet has an id of 0. This is somewhat unusual as the id should be between 1-65535.

```
10:29:14.826507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.172.186.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len
43)
0x0000          4500 002b 0000 0000 0f06 5492 ffff ffff  E..+......T.....
0x0010          aa81 acba 7a69 0203 0000 0000 0000 0000    ....zi.........
0x0020          5014 0000 09ba 0000 636b 6f00 0000              P.......cko...
10:32:53.016507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.132.79.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len 43)
0x0000          4500 002b 0000 0000 0f06 7cfd ffff ffff   E..+......|.....
0x0010          aa81 844f 7a69 0203 0000 0000 0000 0000    ...Ozi.........
0x0020          5014 0000 3225 0000 636b 6f00 0000              P...2%..cko...
10:49:26.156507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.129.188.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len
43)
0x0000          4500 002b 0000 0000 0f06 7f90 ffff ffff   E..+............
0x0010          aa81 81bc 7a69 0203 0000 0000 0000 0000    ....zi.........
0x0020          5014 0000 34b8 0000 636b 6f00 0000              P...4...cko...
11:10:22.596507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.195.178.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len
43)
0x0000          4500 002b 0000 0000 0f06 3d9a ffff ffff  E..+......=.....
0x0010          aa81 c3b2 7a69 0203 0000 0000 0000 0000    ....zi.........
0x0020          5014 0000 f2c1 0000 636b 6f00 0000              P.......cko...
11:44:04.836507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.30.34.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len 43)
0x0000          4500 002b 0000 0000 0f06 e32a ffff ffff  E..+.......*....
0x0010          aa81 1e22 7a69 0203 0000 0000 0000 0000    ..."zi.........
0x0020          5014 0000 9852 0000 636b 6f00 0000              P....R..cko...
12:44:56.156507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.137.174.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len
43)
0x0000          4500 002b 0000 0000 0f06 779e ffff ffff  E..+......w.....
0x0010          aa81 89ae 7a69 0203 0000 0000 0000 0000    ....zi.........
0x0020          5014 0000 2cc6 0000 636b 6f00 0000              P...,...cko...
13:56:26.746507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.89.87.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len 43)
0x0000          4500 002b 0000 0000 0f06 a7f5 ffff ffff   E..+............
0x0010          aa81 5957 7a69 0203 0000 0000 0000 0000    ..YWzi.........
0x0020          5014 0000 5d1d 0000 636b 6f00 0000              P...]...cko...
14:35:08.896507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.200.84.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 14, id 0, len 43)
0x0000          4500 002b 0000 0000 0e06 39f8 ffff ffff  E..+......9.....
```

```
0x0010          aa81 c854 7a69 0203 0000 0000 0000 0000    ...Tzi.........
0x0020          5014 0000 ee1f 0000 636b 6f00 0000         P.......cko...
15:01:20.986507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.23.133.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len 43)
0x0000          4500 002b 0000 0000 0f06 e9c7 ffff ffff  E..+............
0x0010          aa81 1785 7a69 0203 0000 0000 0000 0000    ....zi.........
0x0020          5014 0000 9eef 0000 636b 6f00 0000         P.......cko...
15:04:08.966507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 255.255.255.255.31337 >
170.129.190.188.printer: R [tcp sum ok] 0:3(3) ack 0 win 0 [RST cko] (ttl 15, id 0, len
43)
0x0000          4500 002b 0000 0000 0f06 4290 ffff ffff  E..+......B.....
0x0010          aa81 bebc 7a69 0203 0000 0000 0000 0000    ....zi.........
0x0020          5014 0000 f7b7 0000 636b 6f00 0000         P.......cko...
```

Through further research and investigation into the Q Trojan and the type of traffic that it generates this does appear that it could be false positive. In certain circumstances, a TCP RST packet contains the three-byte payload "cko" in several types of traffic. Consider the two packets below, the port numbers are not at all consistent with the Backdoor Q detect, yet they contain very similar information. I obtained these packets from another network and found that cko string is common in reset packets. The packets captured are from a unix host. Initially, through the guidance of a co-worker, I thought that the three-byte string was native to some function in the Windows TCP protocol. Further investigation shows that this is also common in unix as well.

```
16:57:57.697256 xxx.xxx.xxx.98.15738 > xxx.xxx.xxx.32.http: R [tcp sum ok]
448399365:448399368(3) ack 0 win 0 [RST cko] (ttl 19, id 0, len 43)
0x0000   4500 002b 0000 0000 1306 xxxx xxxx xxxx        E..+......c.B-hb
0x0010   xxxx xxxx 3d7a 0050 1aba 0805 0000 0000        B.W.=z.P........
0x0020   5014 0000 38a8 0000 636b 6f                    P...8...cko

16:57:57.725065 xxx.xxx.xxx.98.38530 > xxx.xxx.xxx.32.http: R [tcp sum ok]
1695178501:1695178504(3) ack 0 win 0 [RST cko] (ttl 19, id 0, len 43)
0x0000   4500 002b 0000 0000 1306 xxxx xxxx xxxx        E..+......c.B-hb
0x0010   xxxx xxxx 9682 0050 650a 5f05 0000 0000        B.W....Pe._.....
0x0020   5014 0000 3e4f 0000 636b 6f                    P...>O..cko
```

Correlations:

Several sources were used to confirm the analysis of this detect. The following candidate number CAN-199-0660 from the Common Vulnerabilities and Exposures website broadly covers backdoor Trojans.
This link to the Whitehats website, http://www.whitehats.com/info/ids203 , provides a brief description of Q and its functionality as a backdoor Trojan. Additionally a good explanation of the Backdoor Q access alert is found at: http://www.snort.org/snort-db/sid.html?sid=184

Evidence of Active Targeting:

As mentioned previously in this document, the destination IP addresses appear to be random, so it is highly unlikely that this attack was directed at any one particular host.

Severity:

Severity = (criticality + lethality) - (system countermeasures + network counter measures)

Criticality	The criticality of the targeted systems(s) is unknown.	**2**
Lethality	As this appears to be a false positive, however if this was a set of stimulus packets and a Trojan was activated on the target host, then the damage could be severe.	**5**
System	It is not known what defense mechanisms have been placed on the targeted hosts.	**2**
Network	The destination network layout is unknown; therefore it is not clear what countermeasures have been put in place.	**2**
Total Score	**(Criticality = 2 + Lethality = 5) – (System CM = 2 + Network CM = 2)** A more accurate analysis could be provided with detailed information for the destination host and network	**3**

Defensive Recommendations:

1. If possible block all incoming traffic, any protocol, destined for port 515. Common sense is to turn off any services not needed and close all ports not needed to prevent exposure to potential attacks.
2. Diligence on the part of the administrator in keeping up to date with anti-virus signatures on all hosts on the network.
3. Firewall rules denying any inbound broadcast traffic

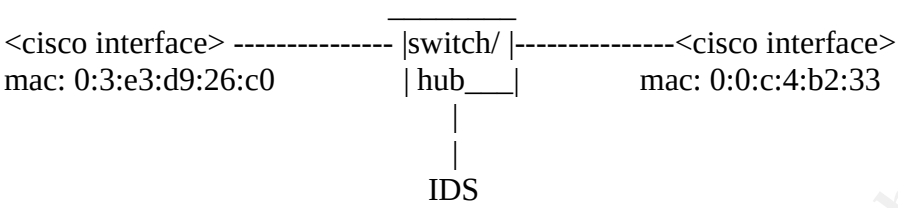
Multiple Choice Question:

What can TCP port 515 be used for:
A. Telnet
B. Syslog messages
C. Unix print services
D. POP3
The answer is C.

Correspondence from incidents.org:

Question: What does the network look like? (use mac addresses to figure out where the IDS is).

Response: The following is my best guess at the layout of this network or at least where the IDS is sitting. I apologize in advance if my poor attempt at a network map snippet didn't keep its format, it looks good in notepad. :-)



Determine where the packets are coming from:
[andy@spro 10]$ tcpdump -ner 2002.10.14 | awk '{print $3}' | sort -u
0:0:c:4:b2:33
0:3:e3:d9:26:c0

Determine where the packets are going:
[andy@spro 10]$ tcpdump -ner 2002.10.14 | awk '{print $2}' | sort -u
0:0:c:4:b2:33
0:3:e3:d9:26:c0

Two cisco nic's according to http://standards.ieee.org/regauth/oui/oui.txt

Comment: I belive this is a false positive. Here is a packet from the whitehats site that shows what Q packets look like.

01/04-02:51:15.622040 255.255.255.255:59564 -> target:15579
TCP TTL:240 TOS:0x0 ID:2323
****A* Seq: 0x5F9B8911 Ack: 0x204F1C8C Win: 0x61C4
70 69 6E 67 20 32 33 2E 32 33 2E 32 33 2E 32 33 ping 23.23.23.23
00

Response: Through further research and investigation into the Q trojan and the type of traffic that it generates this does appear that it could be false positive. In certain circumstances, a TCP RST packet contains the 3 byte payload "cko" in several types of traffic. Consider the two packets below, the port numbers are not at all consistant with the Backdoor Q detect, yet they contain very similar information.

16:57:57.697256 xxx.xxx.xxx.98.15738 > xxx.xxx.xxx.32.http: R [tcp sum ok]
448399365:448399368(3) ack 0 win 0 [RST cko] (ttl 19, id 0, len 43)
0x0000 4500 002b 0000 0000 1306 xxxx xxxx xxxx E..+.....c.B-hb
0x0010 xxxx xxxx 3d7a 0050 1aba 0805 0000 0000 B.W.=z.P.....
0x0020 5014 0000 38a8 0000 636b 6f P...8...cko

```
16:57:57.725065 xxx.xxx.xxx.98.38530 > xxx.xxx.xxx.32.http: R [tcp sum ok]
1695178501:1695178504(3) ack 0 win 0 [RST cko] (ttl 19, id 0, len 43)
0x0000   4500 002b 0000 0000 1306 xxxx xxxx xxxx        E..+.....c.B-hb
0x0010   xxxx xxxx 9682 0050 650a 5f05 0000 0000        B.W....Pe._.....
0x0020   5014 0000 3e4f 0000 636b 6f                    P...>O..cko
```

I obtained these packets earlier today in an attempt to discover what exactly "cko" is. These packets happen to come from a Unix host processing web traffic. A co-worker pointed me in the right direction to walk the path of discovery, in search of the mystical "cko" string. With that direction it was determined that certain versions of the windows operation system inserted the additional three bytes into the packet through some un-documented process. I don't claim to be an expert in this field by any stretch of the imagination, but it does appear that other hosts are capable of generating the same kind of packets as well. As mentioned, these packets are from a Unix host. The significance of the three additional bytes is probably not as important as I make it out to be, I just have a hard time letting go of things I am unable to explain. I would say, contrary to my initial assessment, that this probably not a stimulus packet.

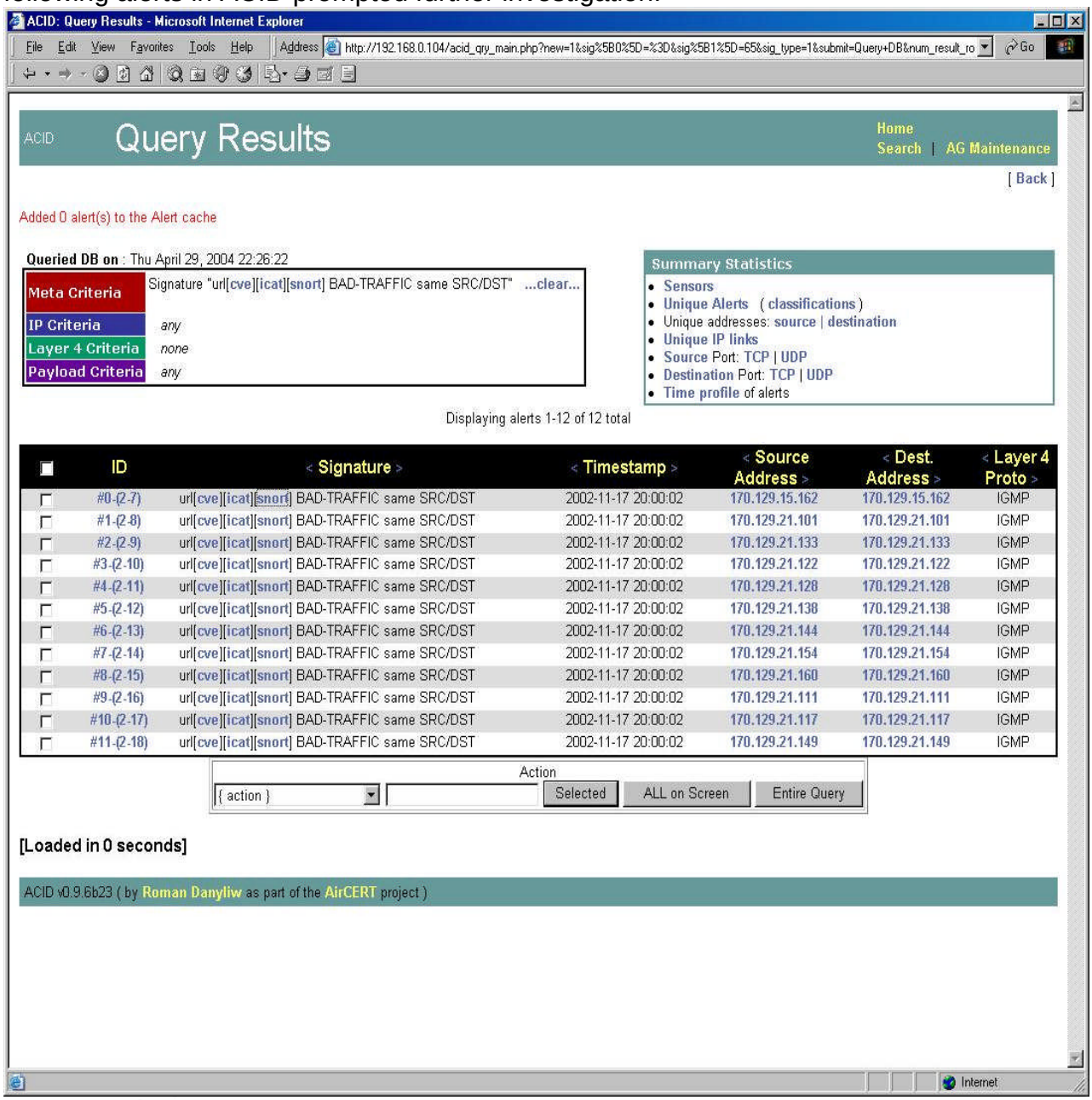
Back to Q - the packet that Donald Smith provided from the whitehats website is indeed one type of packet that can be generated by Q. On the other hand, Q is flexible enough to allow the user to craft the IP address, source and destination ports, and be send packets via TCP, ICMP or UDP protocols, so the packet could truly take just about any form. I am not sure that there is a definitive way to detect the Q Trojan. None of the documents I have read include one sure fire way to detect it. If there is any information out there that could explain this in greater detail, I think it would be wise to share it for the benefit of others who attempt to analyze this kind of detect. I see several write-ups on it in various GCIA practical assignments, but no definitive answers.

<END CORRESPONDENCE>

Detect 3: Land Attack?

Source of trace:
The source of this trace was from the raw tcpdump log files at:
http://www.incidents.org/logs/Raw/2002.10.18

Detected Generated with: Snort version 2.0.4 reporting to a ACID console. The following alerts in ACID prompted further investigation.



Relevant Snort Alerts:

```
[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.646507 170.129.15.162 -> 170.129.15.162
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
```

[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.101 -> 170.129.21.101
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.133 -> 170.129.21.133
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.122 -> 170.129.21.122
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.128 -> 170.129.21.128
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.138 -> 170.129.21.138
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.144 -> 170.129.21.144
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.154 -> 170.129.21.154
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-

```
bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.160 -> 170.129.21.160
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.111 -> 170.129.21.111
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.117 -> 170.129.21.117
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-1999-0016]

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/17-21:00:02.666507 170.129.21.149 -> 170.129.21.149
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-1999-0016]
```

Snort signature that generated an alert:
alert ip any any -> any any (msg:"BAD-TRAFFIC same SRC/DST"; sameip; reference:cve,CVE-1999-0016; reference:url,www.cert.org/advisories/CA-1997-28.html; classtype:bad-unknown; sid:527; rev:4;)

The snort signature alerts when IP packets have the same source and destination IP address.

Probability the source IP address was spoofed:
The probability is very high that this IP address has been spoofed as traffic with the same source and destination IP address should never been seen in normal traffic. Per RFC 2236

"All IGMP messages described in this document are sent with IP TTL 1"

The packets in this trace have a TTL 46, which is not consistent with IGMP Multicast Queries. IGMP messages have a TTL 1 because they are meant to be sent to from one host to multicast router and then forwarded accordingly.

Description of the attack:
According to the information provided at:
http://www.snort.org/snort-db/sid.html?sid=527 this is an attempt at a DoS attack. The reference links included in the snort alerts lead one to believe the information in the alert is correct. http://www.cert.org/advisories/CA-1997-28.html http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-00
More through analysis would indicate otherwise. The references lead an analyst to believe that this is a Land attack in the way that the source and destination IP addresses are the same.
However, the differentiating factor here is that the Land attack utilizes TCP SYN packets with the same source and destination IP addresses and ports.
In this trace the source IP and the destination IP addresses are the same, but that is where the similarities end. The packets from this trace are from a IGMP v2 Membership query. There are three types of IGMP messages:

1. Membership query – In the membership query there are two subtypes:
    a. General query – this is used to discover which groups have members on the network.
    b. Group-specific query – this is used to discover if a specific group has any members.
2. Version 2 Membership Report
3. Leave Group

The traffic detected by snort is an IGMP Group Specific query. The basis for this analysis comes from several sources, the most definitive one is RFC 2236 which states:

> 2.4. Group Address
> In a Membership Query message, the group address field is set to zero when sending a General Query, and set to the group address being queried when sending a Group-Specific Query.

Attack Mechanism:
In the table below, the tcpdump output shows that this set of queries contains an invalid Group Address in the 240.0.0.0 range of addresses. The “All Systems Multicast Group” should be 224.0.0.1 and the “All Routers Multicast Group” should be 224.0.0.2. The valid range for this type of a request should be in the 224.0.0.0-239.255.255.255 range as per RFC 1054. The [gaddr], in the tcpdump output below, is not in the valid range of addresses for this query, it must be assumed that this is malicious traffic of some type.
There are a few different IGMP attack vectors. I downloaded and compiled three different exploit code examples, to run against my test systems; I was unable to duplicate the traffic. The examples used fragmented packets to create a DoS condition on the host. The traffic detected in this trace log indicates that it is more of a probe to discover if a group has any members, possibly for some other type of attack. It is possible for this to be a DoS attack due to the fact that other IGMP DoS attacks from malformed IGMP packets exists, as mentioned previously. It is difficult to say what kind of an attack this could be or what application was used

to generate this type of traffic. tcpdump –envvX –r 2002.10.18 igmp achieved the following results. (not all packets are shown in the interest of space and readability)

```
21:00:02.646507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 170.129.15.162 > 170.129.15.162:
igmp query v2 [gaddr 240.0.3.34] (ttl 46, id 0, len 28)
0x0000   4500 001c 0000 0000 2e02 189a aa81 0fa2 E...............
0x0010   aa81 0fa2 1164 fb78 f000 0322 0000 0000 .....d.x..."....
0x0020   0000 0000 0000 0000 0000 0000 0000      ..............
21:00:02.666507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 170.129.21.101 > 170.129.21.101:
igmp query v2 [gaddr 240.0.1.21] (ttl 46, id 0, len 28)
0x0000   4500 001c 0000 0000 2e02 0d14 aa81 1565 E..............e
0x0010   aa81 1565 1164 fd85 f000 0115 0000 0000 ...e.d..........
0x0020   0000 0000 0000 0000 0000 0000 0000      ..............
21:00:02.666507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 ip 60: 170.129.21.133 > 170.129.21.133:
igmp query v2 [gaddr 240.0.1.53] (ttl 46, id 0, len 28)
0x0000   4500 001c 0000 0000 2e02 0cd4 aa81 1585 E...............
0x0010   aa81 1585 1164 fd65 f000 0135 0000 0000 .....d.e...5....
0x0020   0000 0000 0000 0000 0000 0000 0000      ..............
```

Correlations:

I could not find any references to this type of attack specifically in the CVE database, newsgroups, GOOGLE searches or otherwise. There were some attacks that were similar to some degree but the evidence in this attack did not match specifically to any of the information researched. I drew on information obtained from http://www.faqs.org/rfcs/rfc1054.html and from http://www.faqs.org/rfcs/rfc2236.html
to learn how IP Multicasting works IGMP traffic works in order to form my conclusions for this attack.
The reference for the Land attack http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0016 did not fit the description of this particular detect. Nor did the reference for the Teardrop attack found here: http://www.cert.org/advisories/CA-1997-28.html
The nearest reference to this attack I was able to find is located at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0796 This describes a DoS in IRIX and FreeBSD caused by malformed IGMP multicast packets.

Evidence of Active Targeting:

The probability of active targeting is low, as it appears that the destination hosts are picked at random in the course of a broadscan.

Severity:

Severity = (criticality + lethality) - (system countermeasures + network counter measures)

Criticality	There is no information available regarding the destination hosts.	**1**
Lethality	The traffic is malicious, and if the attack is	**3**

	successful it could cause a DoS condition.	
System	It is unknown what counter measures have been placed on the target systems.	**2**
Network	Not enough information is given to know what has been done to protect the network.	**2**
Total Score	**(Criticality = 1 + Lethality = 3) – (System CM = 2 + Network CM = 2) Low threat**	**0**

Defensive Recommendations:

1. Restrict IGMP messages at the firewall, if not needed for the business to function then block IGMP messages entirely.
2. Block packets at the firewall that contain the same source and destination IP addresses.

Multiple Choice Question:
There are three types of IGMP messages what are they:

A. Membership Query, Leave Group, Version 2 Membership Report
B. Leave Group, Version 2 Membership Query, Membership Report
C. Membership 2 Query, Join Group, Membership Report.

The Correct answer is A.

Part 3: Analyze This

Executive Summary

This is the security audit of the University. Logs from March 24, 2003 through March 28, 2003 were analyzed to provide a detailed recommendation for enhanced network security. This analysis is not entirely complete as pertinent information regarding the University network layout, and services provided. (Email, Remote Administration Applications, File shares, et. al.)
Several hours of sorting, analyzing and categorizing the data in these logs should provide a clear concise high level view of the activity on the network. The intent is to present insight into areas requiring more attention to prevent unauthorized access or exploitation of vulnerabilities on the University network. Furthermore this document should provide assistance in gaining a better overall security posture. I have concatenated all of the logs into 3 large log files (alerts.all, scans.all, oos.all) like several other students have done, for a less complicated analysis process.

Files Used

Alert Files	Scan Files	Out of Spec files
alert.030324.gz	scans.030324.gz	OOS_Report_2003_03_24_28598
alert.030325.gz	scans.030325.gz	OOS_Report_2003_03_25_4452
alert.030326.gz	scans.030326.gz	OOS_Report_2003_03_26_15824

alert.030327.gz	scans.030327.gz	OOS_Report_2003_03_27_23034
alert.030328.gz	scans.030328.gz	OOS_Report_2003_03_28_5271

Most Common Events Found In the alerts.all File

Events	Occurrences
SMB Name Wildcard	370961
TCP SRC and DST outside network	65658
CS WEBSERVER external web traffic	43574
MY.NET.30.3_activity	19363
High port 65535 tcp possible Red Worm traffic	19124
Watchlist 000220 IL-ISDNNET-990517	18007
High port 65535 udp possible Red Worm traffic	13427
Russia Dynamo - SANS Flash 28-jul-00	9677
spp_http_decode: IIS Unicode attack detected	8325
SUNRPC highport access!	5223

Analysis of Most Common Events
This portion of the document will provide a more detailed analysis of the most common events captured on the University network. Like several other students before me, I chose to analyze events that occurred more than 10k times. My analysis will include recommendations to reduce the amount of these types events for the University.

SMB Name Wildcard
Occurrences: 370769

Summary: It is very possible most of these events are of no consequence while originating from the internal network as a few of these events do. However, the largest numbers of events originate from outside the network specifically from 66.158.123.63. There is a high probability that these events are a record of an attempt to exploit the Microsoft Network SMB Buffer Overflow vulnerability. If successfully exploited, an attacker, either anonymous or an authenticated user, could execute arbitrary code on the target host by sending a malformed SMB request packet to the vulnerable host.
As stated by Microsoft:

> "By sending a specially crafted packet request, an attacker can mount a denial of service attack on the target server machine and crash the system. The attacker could use both a user account and anonymous access to accomplish this. Though not confirmed, it may be possible to execute arbitrary code."

```
03/24-17:15:12.106528  [**] SMB Name Wildcard [**] 207.6.30.34:1030 ->
MY.NET.104.209:137
03/24-17:15:12.139465  [**] SMB Name Wildcard [**] 61.60.129.96:1025 ->
MY.NET.120.213:137
03/24-17:15:12.261370  [**] SMB Name Wildcard [**] 200.165.63.135:1027 ->
```

```
MY.NET.178.241:137
03/24-17:15:12.276511  [**] SMB Name Wildcard [**] 24.185.239.110:1026 ->
MY.NET.239.63:137
03/24-17:15:12.599887  [**] SMB Name Wildcard [**] 61.60.129.96:1025 ->
MY.NET.120.216:137
03/24-17:15:12.710833  [**] SMB Name Wildcard [**] 207.6.30.34:1030 ->
MY.NET.104.213:137
03/24-17:15:12.868049  [**] SMB Name Wildcard [**] 200.165.63.135:1027 ->
MY.NET.178.245:137
03/24-17:15:12.879420  [**] SMB Name Wildcard [**] 24.185.239.110:1026 ->
MY.NET.239.67:137
```

By viewing the events shown above, it can be assumed that this is one of three things:

1. A mis-configured source host trying to reach a non-existent share.
2. An attempt by an attacker to discover an unprotected share.
3. An attempt by an attacker to exploit the vulnerability as stated by Microsoft.

Correlations:
Todd Beardsley describes this as "NetBIOS name resolution traffic." in his GCIA p.36.[6] The general consensus is that any file sharing outside of the local network is not recommended due to the lack of security and viability of this attack vector. This can be verified at Microsoft or in the Common Vulnerabilities and Exposures database under the these candidate numbers CAN-2003-0345 and CAN-2002-0724

Recommendations:
This type of traffic should be considered to be a threat and restricted as soon as possible. All inbound NetBIOS traffic should be blocked at the perimeter.

TCP SRC and DST outside network
Occurrences: 65658
Summary: There is nothing particularly unusual about this network traffic however, that is not to say that this is completely "harmless" either. A snippet of events illustrated the introduction of file sharing to the network. For instance, notice the last event in the table below. The traffic is going to port 1214, which according to **Treachery** is commonly used for file sharing applications such as: Kazaa. Morphous and Grokster.

```
03/24-17:27:17.429841  [**] TCP SRC and DST outside network [**]
192.168.0.89:1921 -> 63.146.120.73:80
03/24-18:33:01.617060  [**] TCP SRC and DST outside network [**]
192.168.1.101:1181 -> 128.91.55.23:80
03/24-20:01:09.547596  [**] TCP SRC and DST outside network [**]
```

[6] References for GCIA practicals are located in the references section at the end of this document.

```
169.254.101.152:1436 -> 205.188.48.66:5190
03/24-20:04:56.913126  [**] TCP SRC and DST outside network [**]
192.168.1.101:1689 -> 35.11.201.250:1569
03/24-20:26:50.570160  [**] TCP SRC and DST outside network [**]
192.168.1.101:3087 -> 66.75.238.241:1214
```

```
 03/27-12:28:07.242276  [**] TCP SRC and DST outside network [**]
 192.168.100.51:1830 -> 192.168.1.7:1214
 03/27-12:28:07.343937  [**] TCP SRC and DST outside network [**]
 192.168.100.51:4709 -> 12.219.195.68:1214
 03/27-18:07:20.707885  [**] TCP SRC and DST outside network [**]
 192.168.1.101:4111 -> 192.168.0.2:1214
 03/28-03:08:24.438808  [**] TCP SRC and DST outside network [**]
 192.168.1.101:2610 -> 66.68.239.83:1214
 03/28-05:37:07.711730  [**] TCP SRC and DST outside network [**]
 192.168.1.101:2411 -> 62.215.84.49:1214
 03/28-06:48:33.603853  [**] TCP SRC and DST outside network [**]
 192.168.1.101:2064 -> 24.208.6.29:1214
 03/28-07:10:55.972331  [**] TCP SRC and DST outside network [**]
 192.168.1.101:3188 -> 24.208.6.29:1214
 03/28-07:12:20.057174  [**] TCP SRC and DST outside network [**]
 192.168.1.101:3259 -> 66.26.39.92:1214
  03/28-07:41:10.823336  [**] TCP SRC and DST outside network [**]
192.168.1.101:4813 -> 66.26.39.92:1214
```

It is common knowledge that file sharing applications, such as those noted, breed infestation of worms, viruses and Trojans. Attackers commonly use this as a vector to exploit known vulnerabilities in various Internet applications as well as certain operating systems. Also, due to recent rulings on copyright infringements, users place a potential liability on the University network.

Additionally, port 5190, which is typically used for AOL Instant messaging, could be a potential attack vector to exploit vulnerabilities in other Internet applications as well. One example in particular is the Internet explorer remote code execution exploitable by AOL instant messengers file sharing capabilities. This exploitable due to the fact that AOL Instant Messenger (AIM) saves "buddy" icons for each person in the victim's buddy list in a known location. The icon comes from the attacker, when he/she is added to the victim's buddy list. It is possible for the attacker to substitute a malicious executable HTML file instead of an icon, uploading the file to the victim's hard drive. Since the path to the malicious file is hard-coded, it is possible for the attacker to craft a link to the file in an HTML instant message or email that will cause the malicious file to be run in the Local Computer security zone, bypassing normal security controls. The attacker can then execute the code of choice with no interaction from the user.

As mentioned the traffic in and of itself is not malicious but it does have the potential to provide an attacker a way in to act maliciously.

Correlations:
A quick search on the SANS website revealed a couple of articles to support the statements regarding Kazza, Morpheous and Grokster file sharing applications. They can be found at ZDNET and at Wired

The following candidate numbers have been reserved for the vulnerabilities in Kazza, Morpheous and Grokster
CAN-2002-0314
CAN-2002-0315
CAN-2003-0397

Recommendations:
Being that this is a University, the restriction of instant messaging is probably not a viable option. The network administration team could take the necessary precautions and apply all patches for the applications affected. Simultaneously, due to the danger of the file sharing applications mentioned regarding worms, copyright infringements, et al, these applications should be removed from the Universities computer systems and the ports they used should be blocked at the perimeter.

CS WEBSERVER External Web Traffic
Occurrences: 43574

Summary: This alert appears to be from a custom snort rule to watch the traffic on the CS WEBSERVER. I checked the current rule set from www.snort.org and I was unable to locate anything similar to this. There doesn't appear to be any traffic that is out of the ordinary so this should be considered a false positive.

```
 03/28-22:54:28.392511  [**] CS WEBSERVER - external web traffic [**]
66.196.72.38:6358 -> MY.NET.100.165:80
 03/28-22:54:24.485380  [**] CS WEBSERVER - external web traffic [**]
158.254.232.156:3278 -> MY.NET.100.165:80
03/28-22:54:23.173117 [**] CS WEBSERVER - external web traffic [**]
203.123.64.148:9120 -> MY.NET.100.165:80
03/28-22:54:21.356297  [**] CS WEBSERVER - external ftp traffic [**]
62.101.126.9:3516 -> MY.NET.100.165:21
03/28-22:54:17.150078  [**] CS WEBSERVER - external web traffic [**]
216.202.204.46:32897 -> MY.NET.100.165:80
03/28-22:54:15.569042  [**] CS WEBSERVER - external web traffic [**]
200.52.169.13:3438 -> MY.NET.100.165:80
03/28-22:54:08.968104  [**] CS WEBSERVER - external web traffic [**]
208.201.239.27:59257 -> MY.NET.100.165:80
03/28-22:53:01.859514  [**] CS WEBSERVER - external ftp traffic [**]
213.140.10.138:3752 -> MY.NET.100.165:21
03/28-22:53:01.411156  [**] CS WEBSERVER - external web traffic [**]
216.202.204.46:32891 -> MY.NET.100.165:80
03/28-22:52:50.356891  [**] CS WEBSERVER - external web traffic [**]
```

66.196.72.49:33319 -> MY.NET.100.165:80

Correlation:
I searched at least 10 practical assignments for one that had this kind of event analyzed and I was unable to find any. There were a few that had this type of event listed in the events that occurred most often. I was unable to find any snort signatures that applied to this kind of alert. I am certain this is a custom signature and the alerts we see here are just notifications of traffic to the MY.NET.100.165 host on port 80 and port 21.

Recommendations:
As this appears to be such an active server, then it might be wise to remove the signature that looks for "any" traffic on ports 80 and 21 and create a signature that looks for malicious traffic going to that server, specifically ports 80 and 21. This will reduce the obscurity of vision regarding the overall picture of the network.

MY.NET.30.3_activity
Occurrences: 19363

Summary: This appears to be a generic signature watching for activity on the MY.NET.30.3 network. The alert detected traffic for Netware Core Protocol traffic. NCP is used to facilitate services such as file access, printer access, name management etc. Novell Netware Remote Manager uses port 8009.

> "NetWare Core Protocol (NCP) is a series of server routines designed to satisfy application requests coming from, for example, the NetWare shell. The services provided by NCP include file access, printer access, name management, accounting, security, and file synchronization."[7]

Novell Netware 5.0, and 5.1 are known to have a vulnerability where the server will disclose system information via port 524 if properly queried.

> "Due to a combination of legacy support and default settings, Novell Netware servers using native IP will leak system information via TCP port 524 when properly queried. In mixed Novell/Microsoft environments, information regarding Microsoft devices is leaked via the Service Advertising Protocol (SAP) table. Third party products, such as those used to synchronize directory services between environments can further the problem. Essentially, a remote attacker can gather the equivalent information provided by the console command "display servers" and the DOS client command "cx /t /a /r" without authentication."[8]

Note: This vulnerability is only limited to the internal network, unless TCP 524 is allowed through the Firewall.

[7] http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/netwarep.htm
[8] http://www.securiteam.com/securitynews/6L00C0K0AE.html

Notice the source IP address in the first table below. I was unable to determine why an IP address would be considered a “legal” address with the final octet of ‘0’. The registration information for this host will appear in the Top Ten Talkers section of this document.

```
 03/28-22:03:54.243225  [**] MY.NET.30.3 activity [**] 68.49.35.0:1793 ->
MY.NET.30.3:524
 03/28-22:03:54.266656  [**] MY.NET.30.3 activity [**] 68.49.35.0:1793 ->
MY.NET.30.3:524
 03/28-22:03:54.408115  [**] MY.NET.30.3 activity [**] 68.49.35.0:1793 ->
MY.NET.30.3:524
 03/28-22:03:54.535170  [**] MY.NET.30.3 activity [**] 68.49.35.0:1793 ->
MY.NET.30.3:524
 03/28-22:03:54.556162  [**] MY.NET.30.3 activity [**] 68.49.35.0:1793 ->
MY.NET.30.3:524
```

The traffic in the next two tables demonstrates what would appear to be normal NCP and Remote Manager traffic. It is unknown what this is actually used for in the context of the University due to the lack of information concerning the network layout. As a result I cannot provide accurate results regarding this traffic as being a tactic for gathering system information or this is legitimate communication between two hosts. Further analysis of the raw binary data taken from tcpdump or snort using the –X option would be required to provide the most accurate analysis.

```
 03/24-17:39:09.995171  [**] MY.NET.30.3 activity [**] 209.158.139.5:35105 ->
 MY.NET.30.3:524
 03/24-17:39:10.615370  [**] MY.NET.30.3 activity [**] 209.158.139.5:35105 ->
 MY.NET.30.3:524
 03/24-17:39:10.621130  [**] MY.NET.30.3 activity [**] 209.158.139.5:35105 ->
 MY.NET.30.3:524
 03/24-17:39:10.641080  [**] MY.NET.30.3 activity [**] 209.158.139.5:35105 ->
 MY.NET.30.3:524
 03/24-17:39:11.012359  [**] MY.NET.30.3 activity [**] 209.158.139.5:35105 ->
 MY.NET.30.3:524
 03/24-17:39:19.010550  [**] MY.NET.30.3 activity [**] 209.158.139.5:35105 ->
 MY.NET.30.3:524
```

```
03/28-22:04:41.890833  [**] MY.NET.30.3 activity [**] 138.88.172.42:3319 ->
MY.NET.30.3:8009 03/28-22:04:41.911581  [**] MY.NET.30.3 activity [**]
138.88.172.42:3319 -> MY.NET.30.3:8009
03/28-22:04:41.917284  [**] MY.NET.30.3 activity [**] 138.88.172.42:3319 ->
MY.NET.30.3:8009
03/28-22:04:46.658223  [**] MY.NET.30.3 activity [**] 138.88.172.42:3332 ->
MY.NET.30.3:8009
03/28-22:04:46.687155  [**] MY.NET.30.3 activity [**] 138.88.172.42:3332 ->
```

MY.NET.30.3:8009

Registration for 209.158.139.5

[whois.arin.net]

OrgName: Verizon Internet Services
OrgID: VRIS
Address: 1880 Campus Commons Dr
City: Reston
StateProv: VA
PostalCode: 20191
Country: US

NetRange: 209.158.0.0 - 209.159.31.255
CIDR: 209.158.0.0/16, 209.159.0.0/19
NetName: VIS-209-158
NetHandle: NET-209-158-0-0-1
Parent: NET-209-0-0-0-0
NetType: Direct Allocation
NameServer: NSDC.BA-DSG.NET
NameServer: GTEPH.BA-DSG.NET
Comment:
RegDate:
Updated: 2002-08-22

TechHandle: ZV20-ARIN
TechName: Verizon Internet Services
TechPhone: +1-703-295-4583
TechEmail: noc@gnilink.net

OrgAbuseHandle: VISAB-ARIN
OrgAbuseName: VIS Abuse
OrgAbusePhone: +1-703-295-4583
OrgAbuseEmail: abuse@verizon.net

OrgTechHandle: ZV20-ARIN
OrgTechName: Verizon Internet Services
OrgTechPhone: +1-703-295-4583
OrgTechEmail: noc@gnilink.net

ARIN WHOIS database, last updated 2004-05-09 19:15
Enter ? for additional hints on searching ARIN's WHOIS database.

Registration for 138.88.172.42

[whois.arin.net]

```
OrgName:    Verizon Global Networks, Inc.
OrgID:      VGBN
Address:    1880 Campus Commons Drive
City:       Reston
StateProv:  VA
PostalCode: 20191
Country:    US

NetRange:   138.88.0.0 - 138.88.255.255
CIDR:       138.88.0.0/16
NetName:    VZGNI-PUB-1
NetHandle:  NET-138-88-0-0-1
Parent:     NET-138-0-0-0-0
NetType:    Direct Allocation
NameServer: NSDC.BA-DSG.NET
NameServer: GTEPH.BA-DSG.NET
Comment:
RegDate:
Updated:    2001-05-31

TechHandle: BN-ORG-ARIN
TechName:   Verizon Global Networks Inc.
TechPhone:  +1-703-295-4583
TechEmail:  noc@gnilink.net

# ARIN WHOIS database, last updated 2004-05-09 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

The hosts are registered to Verizon Global Networks, Inc. in Reston VA. As I was not clear if the University utilized NCP or Novell Netware Remote Manager, I chose to look up two of the hosts that were producing the most traffic. No conclusions can be drawn as to the true nature of this traffic without further information regarding the network and the services allowed therein.

Correlations:

I was unable to locate other practical assignments that contained specific information regarding the information I found interesting. As a result I chose to seek the vast resources of the Internet for a more focused view regarding the traffic above. There is a known issue in Novell Netware that leaks sensitive information via TCP port 524. Two very informative write-ups on this vulnerability can be found at: http://www.securiteam.com/securitynews/6L00C0K0AE.html and found here http://www.bindview.com/Support/RAZOR/Advisories/2000/adv_novellleak.cfm

Detailed information about Netware Protocols can be found here: http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/netwarep.htm

Recommendations:

- Verify the existence of Novell Remote Manager compare with security policy to ensure utilization of this application is allowed.
- Block incoming TCP 524 at the perimeter.
- Use the ncpquery tool to investigate what if any information being leaked via TCP 524

High port 65535 tcp possible Red Worm traffic
Occurrences: 19124
High port 65535 udp possible Red Worm traffic
Occurrences: 13427

Summary: The alerts generated by the traffic below are reported as Red Worm. I combined these two alerts together in this analysis due to the similarity save the protocol.
I was unable to find any signatures recent or dated that reflected this type of an alert. I will assume that this is a custom signature designed to watch for traffic that resembles the Red Worm. A Google search for this worm yielded very little information about how the worm actually works. I found that the name has been changed to Adore Worm and further information was obtained on the SANS website. The Adore Worm is designed to create a backdoor by exploiting known vulnerabilities in the LPRng, Wu-Ftpd, ISC Bind and rpc.statd applications on a vulnerable Linux hosts then sends an email to four different email addresses with information about the compromised hosts. Quoting from SANS directly:

> "Adore then runs a package called icmp. With the options provided with the tarball, it by default sets the port to listen too, and the packet length to watch for. When it sees this information it then sets a rootshell to allow connections. It also sets up a cronjob in cron daily (which runs at 04:02 am local time) to run and remove all traces of its existence and then reboots your system. However, it does not remove the backdoor."

UDP

```
03/27-02:11:05.316967  [**] High port 65535 udp - possible Red Worm - traffic
[**] MY.NET.222.194:27005 -> 66.95.149.154:65535
03/27-02:11:05.316954  [**] High port 65535 udp - possible Red Worm - traffic
[**] MY.NET.222.194:27005 -> 66.95.149.154:65535
03/27-02:11:05.306215  [**] High port 65535 udp - possible Red Worm - traffic
[**] 66.95.149.154:65535 -> MY.NET.222.194:27005
03/27-02:11:05.015158  [**] High port 65535 udp - possible Red Worm - traffic
[**] MY.NET.222.194:27005 -> 66.95.149.154:65535
03/27-02:11:04.852290  [**] High port 65535 udp - possible Red Worm - traffic
[**] MY.NET.222.194:27005 -> 66.95.149.154:65535
03/27-02:11:04.585266  [**] High port 65535 udp - possible Red Worm - traffic
[**] MY.NET.222.194:27005 -> 66.95.149.154:65535
```

TCP

```
03/26-13:55:43.899146  [**] High port 65535 tcp - possible Red Worm - traffic [**]
MY.NET.222.122:4193 -> 141.157.165.219:65535
03/26-13:55:43.896670  [**] High port 65535 tcp - possible Red Worm - traffic [**]
141.157.165.219:65535 -> MY.NET.222.122:4193
03/26-13:55:42.448568  [**] High port 65535 tcp - possible Red Worm - traffic [**]
MY.NET.87.122:3234 -> 210.77.58.167:65535
03/26-13:55:42.174082  [**] High port 65535 tcp - possible Red Worm - traffic [**]
MY.NET.222.122:4193 -> 141.157.165.219:65535
03/26-13:55:42.171651  [**] High port 65535 tcp - possible Red Worm - traffic [**]
141.157.165.219:65535 -> MY.NET.222.122:4193
03/26-13:55:42.102164  [**] High port 65535 tcp - possible Red Worm - traffic [**]
210.77.58.167:65535 -> MY.NET.87.122:3234
```

Without the raw packet data it is difficult to say if this is a valid alert or a false positive, since port 65535 is within range of acceptable ports for return traffic. Additionally if these return packets passed through a stateless firewall, this alert could have been generated.

Source	Count	Protocol
MY.NET.222.194	6098	UDP
66.95.149.154	4511	UDP
141.157.165.219	3156	TCP
210.77.58.167	2701	TCP

Correlations:

To correlate this information I used the following links and articles. Additionally, I was able to glean some information from Marcus Wu's GCIA Practical along with several GOOGLE searches. Links used for correlation are listed below.

Information regarding the LPRng User-supplied format string vulnerability can be found here: http://www.securityfocus.com/bid/1712

Information regarding Wu-Ftpd format string stack overwrite vulnerability can be found here: http://www.securityfocus.com/bid/1387

Information regarding rpc.statd format string vulnerability can be found here: http://www.securityfocus.com/bid/1480

A good article on the adore worm can be found here: http://www.sans.org/y2k/adore.htm

A script to scan for and remove the adore worm can be found here: http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm

Recommendations:
1. The four email addresses that the worm reports back to should be blocked.

adore9000@21cn.com
adore9000@sina.com
adore9001@21cn.com
adore9001@sina.com

2. William Stearns has written a script to detect and remove the worm if you have an infected system. The script is located at :
http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm
Administrators should run this script as a precautionary measure to search for and remove the worm if it exists on any of the University computers.

Watchlist 000220 IL-ISDNNET-990517
Occurrences: 18007
Summary: This signatures purpose is to match a watchlist matching hosts from Israeli networks. I can only assume that is a custom watchlist, setup by the University administrator for reasons not outlined in the Security Audit request. Some of the hosts that triggered these alerts are connecting from somewhere on the Israeli networks back to the University on port 2320. This is a Call Center application known as Siebel There is no information in the Security Audit Request to reflect that there is a call center at the University. Out of 18007 alerts there were only three hosts that connected to the MY.NET.84.244 server on port 2320. All of the hosts originated from the 212.179.x.x network and are listed in the table below. Additional traffic included peer-to-peer file sharing on port 1214. There were a total of five hosts listening on port 1214, they are listed in the table below. As mentioned in section “TCP SRC and DST outside network”, peer-to-peer file sharing is dangerous and should be stopped immediately. The rest of the traffic didn’t include anything that would be considered malicious; the bulk of it was normal web, email and ftp traffic. Although a significant amount of traffic was found connecting to port 4098. Checking all of the port lists and a plethora of websites, via GOOGLE searches, did not provide the information necessary to determine exactly what service is running on that port. The port is reserved for “drmsfsd”, but I was unable to find any information relating to that specific service or any application that uses it. Similarly, there was a large amount of traffic connecting back to the University on port 4056, it is not known what service is running on that port. Research via other GICA Practical Assignments and GOOGLE searches did not provide enough information for an accurate analysis.

Hosts connecting to port 2320

212.179.102.21
212.179.80.237

212.179.106.182

Hosts of interest listening for connections on port 1214

MY.NET.233.30
MY.NET.220.54
MY.NET.217.190
MY.NET.209.22
MY.NET.194.13
MY.NET.150.220
MY.NET.150.133
MY.NET.84.166

Snip of Trace

```
03/27-22:28:57.064047  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.127.11:4431 -> MY.NET.209.22:1214
03/28-00:35:25.385943  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.35.119:1214 -> MY.NET.219.98:2102
03/28-00:35:25.554054  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.35.119:1214 -> MY.NET.219.98:2102
03/28-02:18:04.628034  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.35.119:1214 -> MY.NET.221.186:1186
03/28-02:18:04.760328  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.35.119:1214 -> MY.NET.221.186:1186
03/28-02:18:05.001553  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.35.119:1214 -> MY.NET.221.186:1186
03/28-18:00:11.041210  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.127.16:3494 -> MY.NET.209.22:1214
```

Due to the implementation of the Watchlist signature it appears that it is important to the administrator to know what traffic is being sent or received from the Israeli Networks. The hosts connecting to port 2320 on the MY.NET addresses in the table above should be examined in more detail, as it is not clear if the University is using a call center application. I chose to display the registration information for a host that is connecting back to the University network to a host that is listening on port 1214. I chose 212.179.127.16 as it appears in the snippet of the actual displayed in the table above.

Registration information for External Host 212.179.127.16

```
[whois.ripe.net]
% 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:      212.179.0.0 - 212.179.255.255
org:          ORG-IL9-RIPE
netname:      IL-ISDNNET-990517
descr:        PROVIDER
descr:        ISDNet LTD
country:      IL
admin-c:      YK76-RIPE
tech-c:       MR916-RIPE
tech-c:       BHT2-RIPE
status:       ALLOCATED PA
remarks:      please send ABUSE complains only to abuse@bezeqint.net
notify:       hostmaster@bezeqint.net
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    AS8551-MNT
mnt-routes:   AS8551-MNT
changed:      hostmaster@ripe.net 19990517
changed:      hostmaster@ripe.net 20020912
changed:      hostmaster@ripe.net 20020926
changed:      hostmaster@ripe.net 20030508 # il.isdnnet.yuval via
https://lirportal.ripe.net
source:       RIPE

route:        212.179.120.0/21
descr:        BEZEQ-INTERNATIONAL-LTD
origin:       AS8551
notify:       hostmaster@bezeqint.net
mnt-by:       AS8551-MNT
changed:      hostmaster@bezeqint.net 20030810
source:       RIPE

organisation: ORG-IL9-RIPE
org-name:     ISDNet LTD
org-type:     LIR
descr:        BEZEQ-INTERNATIONAL-LTD
address:      Bezeq International Ltd.
address:      40 Hashacham Street,
address:      P.O Box 7097, Ramat Siv
address:      Petach-Tikva 49170
address:      Israel
phone:        +1800800110
fax-no:       +972 3 9203033
e-mail:       hostmaster@bezeqint.net
admin-c:      MR916-RIPE
```

```
admin-c:      YK76-RIPE
admin-c:      BHT2-RIPE
mnt-ref:      AS8551-MNT
mnt-ref:      RIPE-NCC-HM-MNT
mnt-by:       RIPE-NCC-HM-MNT
changed:      hostmaster@ripe.net 20040415
changed:      hostmaster@ripe.net 20040502 # il.isdnnet.mirih via
https://lirportal.ripe.net
source:       RIPE

role:         BEZEQINT HOSTMASTERS TEAM
address:      Bezeq International
address:      40 hashacham st.
address:      Petach Tikva 49170 Israel
phone:        +972 1 800800110
fax-no:       +972 3 9203033
e-mail:       hostmaster@bezeqint.net
admin-c:      YK76-RIPE
tech-c:       MR916-RIPE
nic-hdl:      BHT2-RIPE
remarks:      Please Send Spam and Abuse ONLY to abuse@bezeqint.net
mnt-by:       AS8551-MNT
changed:      hostmaster@bezeqint.net 20030204
changed:      hostmaster@bezeqint.net 20030508
source:       RIPE

person:       Yuval Keinan
address:      Bezeq International
address:      40 hashacham st.
address:      Petach Tikva 49170 Israel
phone:        +972 1 800800110
fax-no:       +972 3 9203033
remarks:      Please Send Spam and Abuse ONLY to abuse@bezeqint.net
e-mail:       hostmaster@bezeqint.net
mnt-by:       AS8551-MNT
nic-hdl:      YK76-RIPE
changed:      hostmaster@bezeqint.net 20030204
changed:      hostmaster@bezeqint.net 20030508
source:       RIPE

person:       Miri Roaky
address:      Bezeq International
address:      40 hashacham st.
address:      Petach Tikva 49170 Israel
phone:        +972 1 800800110
fax-no:       +972 3 9257021
```

remarks: Please Send Spam and Abuse ONLY to abuse@bezeqint.net e-mail: hostmaster@bezeqint.net mnt-by: AS8551-MNT nic-hdl: MR916-RIPE changed: hostmaster@bezeqint.net 20030204 changed: hostmaster@bezeqint.net 20030508 changed: hostmaster@bezeqint.net 20040203 source: RIPE

Correlations:
I was not sure what application utilized TCP port 2320, a quick search on http://www.treachery.net/security_tools/ports/ directed my attention to the Siebel call center application. I was not entirely familiar with this application so a quick visit to http://www.siebel.com/partners/portal/docs/datasheets/DatasheetAvayaEnt9-52000.pdf provided the information I needed. To learn where this host hailed from I utilized http://whois.ripe.net to gather the registration information for this host. Additionally, I was able to find correlative information in Donald Gregory's GCIA practical as well as some very useful information in Marcus Wu's GCIA practical.[9]

Recommendations:
Again, a big concern here with peer-to-peer file sharing. Due to the amount of Trojans, worms and the like that are transmitted via these types of applications as well as potential legal issues, it is strongly recommended that port 1214 should be blocked at the perimeter as soon as possible. Administrators verify that these file-sharing applications are uninstalled from the hosts on MY.NET in the table listed above. Further investigation in to port 4098 is needed to determine the need to leave this port open as well as port 4056. Additionally, further information is required regarding the Watchlist signature. It may not be completely necessary to obscure the overall vision of the network with the amount of alerts generated by this signature. Again, the reason behind this signature is unknown; as a result an accurate recommendation cannot be made at this time.

Top Ten Talkers

The table below lists the top ten source addresses from the alerts.all file created from the five consecutive days of logs from the University.

[9] References to GCIA practicals are noted in the reference section at the end of this document.

68.49.35.0	15960
MY.NET.222.194	12206
66.95.149.154	9028
212.179.48.177	5824
MY.NET.105.204	5283
194.87.6.230	4388
MY.NET.222.122	3949
128.8.10.18	3614
212.179.43.225	3326
141.157.165.219	3161

Registration information follows for the top two hosts as they are above ten thousand. Notice the first host 68.49.35.0 is from the analysis of the MY.NET.30.3_activity alert. As previously mentioned it is not clear if this host is allowed to access the network via NCP or Remote Management. I am not clear as to why a host could have an IP address with the last octet being zero. For this reason I have chosen to provide registration information to the university for this host.

Registration for 68.49.35.0

```
[ whois.arin.net ]

68.49.35.0 = [  ]
  OrgName:    Comcast Cable Communications  Inc.
  OrgID:      CMCS
  Address:    1800 Bishops Gate Blvd
  City:       Mt Laurel
  StateProv:  NJ
  PostalCode: 08054
  Country:    US
  NetRange:   68.32.0.0 - 68.63.255.255
  CIDR:       68.32.0.0/11
  NetName:    JUMPSTART-1
  NetHandle:  NET-68-32-0-0-1
  Parent:     NET-68-0-0-0-0
  NetType:    Direct Allocation
  NameServer: DNS01.JDC01.PA.COMCAST.NET
  NameServer: DNS02.JDC01.PA.COMCAST.NET
  Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
  RegDate:    2001-11-29
  Updated:    2003-11-05
  TechHandle: IC161-ARIN
  TechName:   Comcast Cable Communications Inc
  TechPhone:  1-856-317-7200
```

```
TechEmail:  cips_ip-registration@cable.comcast.com
OrgAbuseHandle: NAPO-ARIN
OrgAbuseName:   Network Abuse and Policy Observance
OrgAbusePhone:  1-856-317-7272
OrgAbuseEmail:  abuse@comcast.net
OrgTechHandle: IC161-ARIN
OrgTechName:   Comcast Cable Communications Inc
OrgTechPhone:  1-856-317-7200
OrgTechEmail:  cips_ip-registration@cable.comcast.com
CustName:   Comcast Cable Communications  Inc.
Address:    3 Executive Campus
Address:    5th Floor
City:       Cherry Hill
StateProv:  NJ
PostalCode: 08002
Country:    US
RegDate:    2003-03-19
Updated:    2003-03-19
NetRange:   68.48.0.0 - 68.49.255.255
CIDR:       68.48.0.0/15
NetName:    DC-3
NetHandle:   NET-68-48-0-0-1
Parent:     NET-68-32-0-0-1
NetType:    Reassigned
Comment:    NONE
RegDate:    2003-03-19
Updated:    2003-03-19
TechHandle: IC161-ARIN
TechName:   Comcast Cable Communications Inc
TechPhone:  1-856-317-7200
TechEmail:  cips_ip-registration@cable.comcast.com
OrgAbuseHandle: NAPO-ARIN
OrgAbuseName:   Network Abuse and Policy Observance
OrgAbusePhone:  1-856-317-7272
OrgAbuseEmail:  abuse@comcast.net
OrgTechHandle: IC161-ARIN
OrgTechName:   Comcast Cable Communications Inc
OrgTechPhone:  1-856-317-7200
OrgTechEmail:  cips_ip-registration@cable.comcast.com
 ARIN WHOIS database  last updated 2004-05-10 19: 15
 Enter ? for additional hints on searching ARIN's WHOIS database.
```

The table below contains registration information for the destination address for MY.NET.222.194. This top talker can be associated with the “High port 65535 udp - possible Red Worm – traffic alert.” As such, I felt that further investigation of the source host is required.

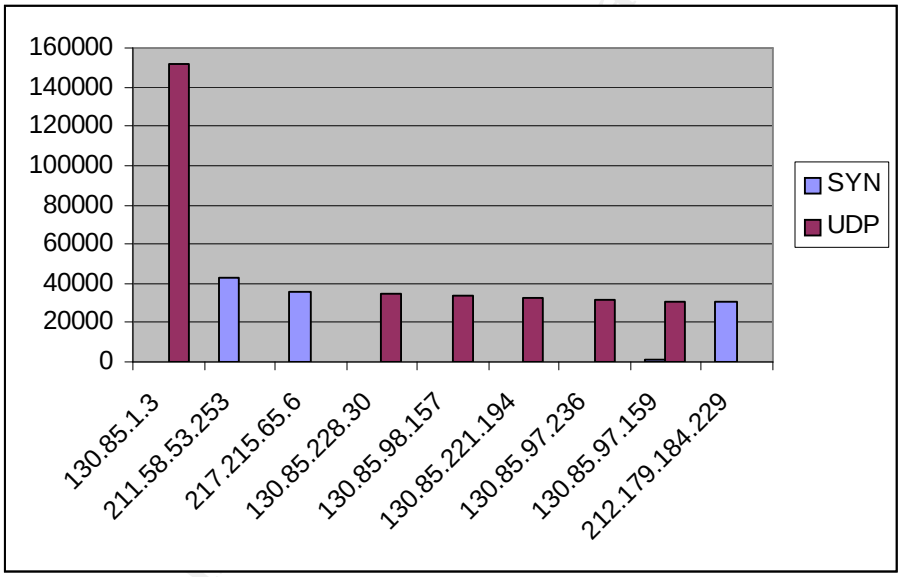
Registration for 66.95.149.154

```
[whois.arin.net]

OrgName:    DSL.net, Inc.
OrgID:      FTCI
Address:    545 Long Wharf Dr. 5th floow
City:       New Haven
StateProv:  CT
PostalCode: 06511
Country:    US

NetRange:   66.95.0.0 - 66.95.255.255
CIDR:       66.95.0.0/16
NetName:    DSL-NET-16
NetHandle:  NET-66-95-0-0-1
Parent:     NET-66-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.DSL.NET
NameServer: NS2.DSL.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2001-03-30
Updated:    2004-04-27

OrgAbuseHandle: ABUSE177-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-203-772-1000
OrgAbuseEmail:  abuse@dsl.net

OrgNOCHandle: NOC291-ARIN
OrgNOCName:   Network Operations Center
OrgNOCPhone:  +1-203-772-1000
OrgNOCEmail:  noc@dsl.net

OrgTechHandle: IPADM54-ARIN
OrgTechName:   IP Administration
OrgTechPhone:  +1-203-772-1000
OrgTechEmail:  ipadmin@dsl.net

# ARIN WHOIS database, last updated 2004-05-10 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Analysis of Scans and OOS Files

Scans Files

In the table below the top then talkers are listed. This information comes from the scans.all file. The data is sorted by UDP and SYN scans. I chose to sort it in this way since the top these are the top two flags that were counted over 10000 times. The SYN packets were counted 1305017 times and the UDP packets were counted 905327 times. You should notice that 130.85.1.3 has the highest count of UDP packets totaling 151424 and 211.58.53.253 has the highest counted SYN packets. The majority of the UDP scans appears to be scanning multiple servers in an attempt at finding a DNS server. All traffic is directed to the destination port 53 from 130.85.1.3.



Host 211.58.53.253 appears to be an attempting to connect on port 445 on multiple hosts on the 130.85.x.x network. I decided to look this up as the traffic seemed very suspicious, as there were over 1000 scans in less than two minutes. I realize that the requirements of this section require that I provide registration information for only 5 External hosts, however, I do feel that the registration information for all 6 hosts is relevant to provide the clearest picture possible, hence the sixth display of registration information.

Registration for 211.58.53.253

[whois.nic.or.kr]
ÇÑ±¹ÀÎÅÍ³ÝÁ¤º¸¼¾¾ÅÍ¿¡¼ Á¦°øÇÏ´Â µµ¸ÞÀÎÀÌ¸§ µî·ÏÁ¤º¸ Á¶È¸(WHOIS) ¼ºñ½º ÀÔ´Ï´Ù.

query: 211.58.53.253

ENGLISH

KRNIC is not ISP but National Internet Registry similar with APNIC.
Please see the following end-user contacts for IP address information.

IP Address : 211.58.53.0-211.58.53.255
Network Name : HANANET-INFRA
Connect ISP Name : HANANET
Connect Date : 20000327
Registration Date : 20031105

[Organization Information]
Orgnization ID : ORG3930
Org Name : Hanaro Telecom Inc.
State : KYONGGI
Address : 726-1 Janghang 2(i)-dong , Goyang-si Ilsan-gu
Zip Code : 411-837

[Admin Contact Information]
Name : IP Administrator
Org Name : Hanaro Telecom Inc.
State : KYONGGI
Address : 726-1 Janghang 2(i)-dong , Goyang-si Ilsan-gu
Zip Code : 411-837
Phone : +82-2-106-2
Fax : +82-2-6266-6483
E-Mail : ip-adm@hanaro.com

[Technical Contact Information]
Name : IP Manager
Org Name : Hanaro Telecom Inc.
State : KYONGGI
Address : 726-1 Janghang 2(i)-dong , Goyang-si Ilsan-gu
Zip Code : 411-837
Phone : +82-2-106-2
Fax : +82-2-6266-6483
E-Mail : ip-adm@hanaro.com

If the above contacts are not rechable, please see the following ISP contacts for relevant information or network abuse complaints.

[ISP IP Admin Contact Information]

Name : IP Administrator
Phone : +82-2-106-2
Fax : +82-2-6266-6483
E-Mail : ip-adm@hanaro.com

[ISP IP Tech Contact Information]
Name : IP Manager
Phone : +82-2-106-2
Fax : +82-2-6266-6483
E-Mail : ip-adm@hanaro.com

[ISP Network Abuse Contact Information]
Name : Network Abuse
Phone : +82-2-106-2
Fax : +82-2-6266-6483
E-Mail : abuse@hanaro.com

Speaking with several veteran security analysts and reading various newsgroups, I found that a large number of attacks come from Asia Pacific. Several companies have that entire IP address range blocked at the perimeter due to the amount of malicious activity from that area. A small sample of the scans has been included for reference. This is probably an attempt to exploit the Deloader Worm, discovered in March of 2003. A small summary from esecurityplanet.com describes in brief what the worm actually does. More information can be found at the esecurity website.

> "In order to spread, this malicious code searches across the Internet for computers to which it can connect through port 445. If a successful connection is made, it copies a file called INST.EXE in the Windows Start folder. This file is a Trojan designed to open a backdoor in the computer. Once it has done this, Deloder also copies a file called DVLDR32.EXE in the infected computer, which contains a copy of the worm."

OOS Files

The following table is a list of top talkers in the oos.all file. Relevant information will be provided for the top five hosts in the interest of readability and space consideration.

Top Ten Talkers oos.all

68.54.93.181	5103
212.186.78.246	983
216.95.201.25	739
66.140.25.157	676
209.164.33.84	664
81.218.95.86	510

148.64.20.178	494
216.95.201.32	478
62.75.157.99	440
216.95.201.31	437

During the analysis of these files I discovered that the top host 68.54.93.181 is connecting to the University mail server MY.NET.6.7 on port 110.

Host 212.186.78.246 is connecting to MY.NET.202.50 on port 6346, which is a well known port for gnutella. Gnutella is a peer-to-peer file shareing application.

Host 216.95.201.25 is connected to several hosts on the University network. It is not clear how many mail servers are in use either testing or production as I do not have a network layout. I am unable to provide a clear analysis as to this being legitimate traffic. All packets are going to the following hosts on port 25:

MY.NET.24.22
MY.NET.6.47
MY.NET.24.21
MY.NET.6.40
MY.NET.24.23

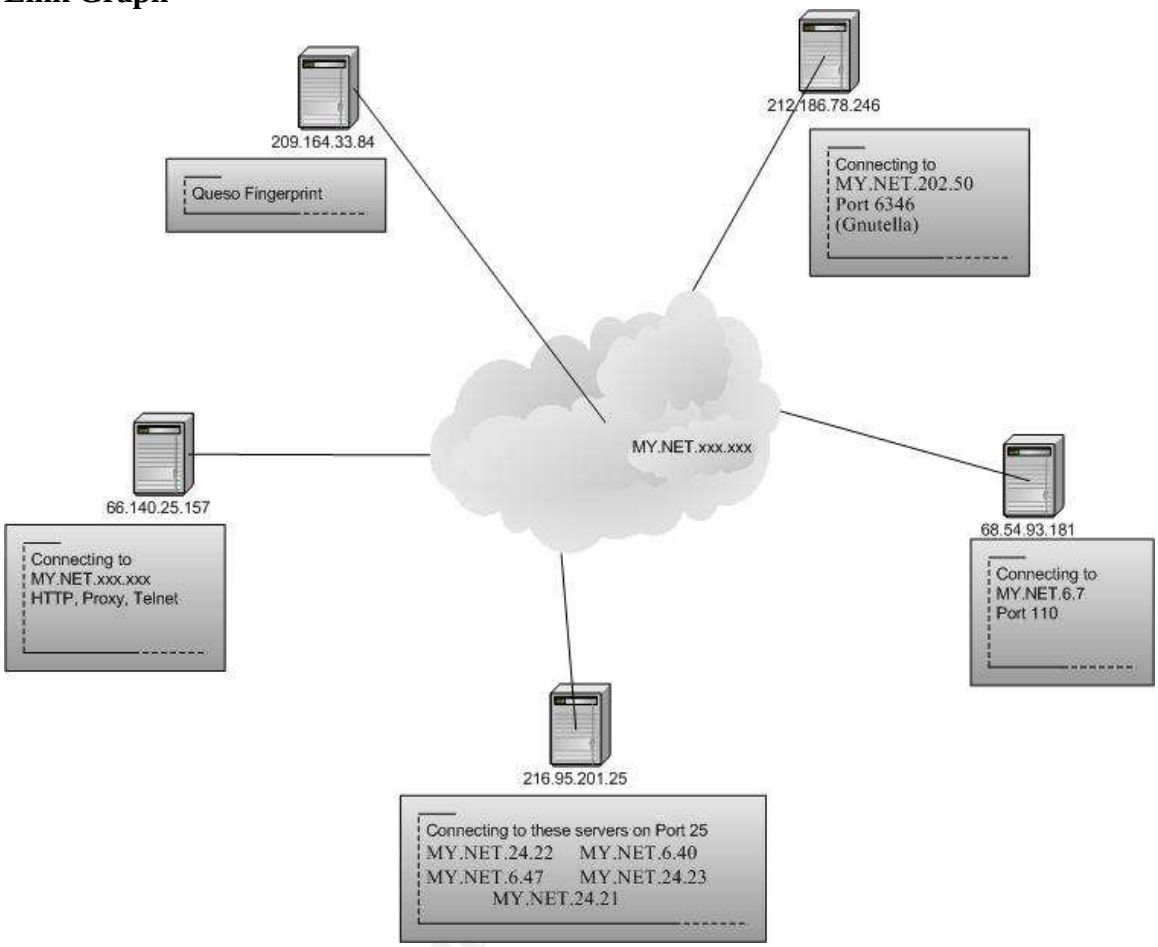
Host 66.140.25.157 is connecting to standard service ports on the MY.NET network, service ports such as telnet, proxy, and http. There didn't appear to be any thing blatantly malicious.

Host 209.164.33.84 appears to be scanning several hosts in the MY.NET network on port 25. I was able to locate some events in the alerts.all files that corresponded with this host directly. It appears this is a Queso fingerprint attempt. A brief description from ISS is as follows:

> "Queso uses a specific style of TCP fingerprinting. It sends packets that are not covered by the protocol specification. This doesn't hurt the target, but since these packets aren't standardized, everybody responds differently. By recording the responses and matching them up with a database, the fingerprinting tool is able to figure out what the operating system is."

This is usually the precursor to an attack. The attacker can zero in on the target host and identify the operating system and exploit the relevant vulnerabilities.

Link Graph



Defensive Recommendations:
Due to the fact that I have provided defensive recommendations for each of the Alerts above, this will be more of a summary regarding steps that should be taken to enhance the security posture of the University network.

SMB Name Wildcard

- This type of traffic should be considered to be a threat and restricted as soon as possible. All inbound NetBIOS traffic should be blocked at the perimeter.

TCP SRC and DST outside network

- (File Sharing and Instant Messaging Detected)
- Restrict ports used for Instant Messaging
- Block ports used for File Sharing Apps (Kazaa, Grokster, Morpheous)
- Network Admin team apply all patches relevant to vulnerabilities exploited by file sharing apps.
- Remove file sharing apps from University computers.
- Block relevant ports at the perimeter

CS WEBSERVER External Web Traffic

- Re-evaluate snort signature to decrease noise generated by this signature.

MY.NET.30.3 Activity

- Verify the existence of Novell Remote Manager compare with security policy to ensure utilization of this application is allowed.
- Block incoming TCP 524 at the perimeter.
- Use the ncpquery tool to investigate what if any information being leaked via TCP 524

High port 65535 tcp possible Red Worm traffic
High port 65535 udp possible Red Worm traffic

- Block Email addresses used by the Adore worm:
  adore9000@21cn.com
  adore9000@sina.com
  adore9001@21cn.com
  adore9001@sina.com
- Run the adorefind script, in accordance with security policy, to detect and remove adore worms from infected machines.

Watchlist 000220 IL-ISDNNET-990517

- Block port 1214 at the perimeter
- Uninstall peer-to-peer file sharing applications
- Investigate ports 4098 and 4056 to verify if this is a requirement for business or students.
- Re-evaluate snort signature to eliminate false positives and excess noise on the IDS.

Scans

- Block IP addresses that originate from Asia Pacific
- Verify all anti-virus signatures are up to date
- Scan for Deloader worm on internal network
- Block port 445 if this does not interfere with business or student requirements.

OOS

- Block port 6346 at the perimeter
- Verify the number of servers allowed to serve up email
- Block host 209.164.33.84
- Verify all hosts are patched with the most current patches to limit exposure to potential exploitation of vulnerabilites perpetuated by a Queso fingerprint scan.

Description of the Analysis Process:
Due to the overwhelming amount of data required for this analysis; I utilized several scripts from previous practical assignments. Other tools included mysql, grep, awk, sort, uniq and vi.
The first script I used was written by a co-worker, Joe Stewart, named parse.pl. This script sorted through the alerts.all file and separated out the messages from the source ip addresses into two directories. (msg, src)

parse.pl

```
#!/usr/bin/perl
# parse.pl

my $count = 0;
while (<>) {
  if (/.*\[\*\*\] (.*) \[\*\*\] (.*) -> (.*)/) {
    $count++;
    my $msg = $1; my $src = $2; my $dst = $3;
    $msg =~ s/[^A-Za-z0-9 -_]//g;
    $msg =~ s/ /_/g;
    open(OUT, ">>msg/$msg");
    print OUT;
    close OUT;
    $src =~ s/:.*//g;
    open(OUT, ">>src/$src");
    print OUT;
    close OUT;
    print "Processed $count lines\n";
  } else {
    print "Skipped $_";
    $skipped .= $_;
  }
}
print "Skipped the following lines:\n$skipped\n";
```

The next tool I used was written by another co-worker, Scott Gregory, for his practical assignment. The tool is called regex, this is a binary executable so I am unable to provide any source code. It works similar to egrep only better as it can print subexpression matchs, for example, print just one word. It was very useful in sifting through the mountains of data. Usage for regex is as follows:

```
Usage: regex [-divcn] [-a subaddr|[delim]range] <expr> [file ...]
    Options:
        -a    just print a portion of the matching line. Using regex
```

sub-address or [delim]range sub-addresses. Ex:
Match and print the 2nd subaddr:
regex -a 2 'foo.*([0-9]*)+([a-z]*)' foo.log
Match 2nd and 4th subaddrs:
regex -a ':2,4'
'(^[abc])([^0-9]*)([A-Z]*)([0-9\.)' foo.log
-d print range of subaddr matches when using -a option
-i ignore case
-v print lines not matching
-c print only a count of matched lines
-n print line number along with lines that match

The next set of tools were used to parse each of the files into a text file so that the information could be imported into a MySQL database. These tools were written in perl by a couple of co-workers, Mike Wisener and Joe Stewart. I made some slight modifications as I used them due to some of the data in the log files being unusable.

Used for Alerts files

```
#!/usr/bin/perl -w
# alerts.pl

while(<>) {
next if m/^$/;
next if m/(\S+)\s+\[\*\*\] spp_portscan: (.*)\[\*\*\]\s+([^:]+):(\S+) -> ([^:]+):(\S+)/;
chomp;
m/(\S+)\s+\[\*\*\] (.*) \[\*\*\]\s+([^:]+):(\S+) -> ([^:]+):(\S+)/;
print "insert into alerts (timestamp,summary,srcip,srcport,dstip,dstport) values
(\"$1\", \"$2\", \"$3\", \"$4\", \"$5\", \"$6\")","\n";
}
```

Used for Scans files

```
#!/usr/bin/perl -w
# scans.pl

 while(<>) {
 next if m/^$/;
$val = $_;
m/(\S+) (\S+) (\S+) ([^:]+):(\S+) -> ([^:]+):(\S+) (\S+)/;
$proto = $8;
$flags = "NULL";
if ($proto ne "UDP") {
$val = m/-> \S+ \S+ (\S+)/;
$flags = $1;
}
```

```
        if ($flags eq "NULL") {
  print "insert into scans (timestamp,srcip,srcport,dstip,dstport,scantype,flags)
values (\"$1 $2 $3\", \"$4\", $5, \"$6\", $7, \"$8\", NULL)",";\n";
        }
        else {
  print "insert into scans (timestamp,srcip,srcport,dstip,dstport,scantype,flags)
values (\"$1 $2 $3\", \"$4\", $5, \"$6\", $7, \"$8\", \"$flags\")",";\n";
        }
   }
```

Used for OOS files

```
#!/usr/bin/perl
# multiline.pl
use strict;


my ($timestamp,$srcip,$srcport,$dstip,$dstport,$ip_ttl,$ip_tos,$ip_id,
  $ip_length,$dgm_length,$tcp_flags,$tcp_seq,$tcp_ack,$tcp_win,
  $tcp_length,$tcp_urgptr,$other);

while (<STDIN>) {

  chomp(my $line = $_);

  next if /^\s*$/;
  if ($line =~ /^\=\+\=\+/) {


        print "insert into oos values (\'$timestamp\', \'$srcip\', \'$srcport\', \'$dstip\',
\'$dstport\', \'$ip_ttl\', \'$ip_tos\', \'$ip_id\', \'$ip_length\', \'$dgm_length\',
\'$tcp_flags\', \'$tcp_seq\', \'$tcp_ack\', \'$tcp_win\', \'$tcp_length\', \'$tcp_urgptr\',
\'$other\');\n" if
$timestamp,$srcip,$srcport,$dstip,$dstport,$ip_ttl,$ip_tos,$ip_id,$ip_length,$dgm
_length,$tcp_flags,$tcp_seq,$tcp_ack,$tcp_win,$tcp_length,$tcp_urgptr,$other;

    $timestamp = $srcip = $srcport = $dstip = $dstport = $ip_ttl = $ip_tos = $ip_id
=
    $ip_length = $dgm_length = $tcp_flags = $tcp_seq = $tcp_ack = $tcp_win =
    $tcp_length = $tcp_urgptr = $other = '';
  } elsif ($line =~ /(.*) (.*):(.*) -> (.*):(.*)/) {
    $timestamp = $1; $srcip = $2; $srcport = $3; $dstip = $4; $dstport = $5;
  } elsif ($line =~ /^(?:TCP|UDP|IP) TTL:(.*) TOS:(.*) ID:(.*) IpLen:(.*) DgmLen:(.*)
(.*)/) {
    $ip_ttl = $1; $ip_tos = $2; $ip_id = $3; $ip_length = $4; $dgm_length = $5;
  } elsif ($line =~ /(.*) Seq: (.*) Ack: (.*) Win: (.*) TcpLen: (.*)/) {
```

```
    $tcp_flags = $1; $tcp_seq = hex($2); $tcp_ack = hex($3); $tcp_win = hex($4);
$tcp_length = $5;
  } else {
        $line =~ s/\\/\\\\/g;
        $line =~ s/\'/\\'/g;

        $other .= $line;
  }
}
```

References:

Todd Beardsley "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). May 8, 2002 URL http://www.giac.org/practical/Tod_Beardsley_GCIA.doc

Donald Corey Merchant "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). October 7, 2002 URL: http://www.giac.org/practical/GCIA/Donald_Merchant_GCIA.doc

Craig Baltes "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). Oct 11, 2002 URL: http://www.giac.org/practical/GCIA/Craig_Baltes_GCIA.doc

Johnny Calhoun "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). January 8th 2003 URL: http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf

Marcus Wu "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). January 23 2003 URL: http://www.giac.org/practical/GCIA/Marcus_Wu_GCIA.pdf

Donald Gregory "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). January 19th 2003 URL: http://www.giac.org/practical/GCIA/Donald_Gregory_GCIA.pdf

Pete Storm "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). November 15th 2003 URL: http://www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf

Ashley Thomas "SANS Intrusion Detection & Analysis Certification." GIAC Certified Intrusion Analysts (GCIA). March 17th 2003 URL: http://www.giac.org/practical/GCIA/Ashley_Thomas_GCIA.pdf

Dzurinda, Greg. 26 2001. Nimda Explained, and What You Can Do to Protect Your Sytem(s). SANS. 5 May 2004 URL: http://www.sans.org/rr/papers/index.php?id=91

Gordan, Les. "On Q" . Whitehats.CA. 8 May 2004 URL: http://www.whitehats.ca/main/publications/external_pubs/Q-analysis/Q-analysis.html

Loveless, Mark. "Novell Netware Default settings expose sensitive system information." 11 Dec 2000. Novell. 9 May 2004 URL: http://www.securiteam.com/securitynews/6L00C0K0AE.html

Nomad, Simple. "Object Enumeration in Novell Environments." 08 Nov 2000. Bindview. 17 May 2004 URL: http://www.bindview.com/Support/RAZOR/Advisories/2000/adv_novellleak.cfm

Cisco, Systems. "NetWare Protocols." 28 2002. 15 May 2004 URL: http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/netwarep.htm

NCPQuery tool file location: http://razor.bindview.com/tools/files/ncpquery-1.2.tgz