DRAFT

GIAC Intrusion Detection In Depth
GCIA Practical Exam

Anthony K Giandomenico
Security Analyst
Honolulu, Hi

SANS Training & GIAC Certification

Hyatt-Regancy
Honolulu, Hawaii 96814

Table of Contents

I. Introduction

Successful deployment of an Intrusion Detection System

After proposing for months to management that the company needs an Intrusion Detection System (IDS), a budget is finally approved. Your efforts of researching IDS technology and devising a recommendation have paid off. The equipment is shipped and you are ready to deploy. Management eagerly waits to see a return on this investment.

Three weeks after deployment you discover that your web server has been hacked without alerting you of an intrusion. You ask yourself how this could have happened since the company purchased the best IDS on the market. You set up sensors strategically across your network and should have been alerted of this intrusion. In the process of investigating this occurrence, you discover that your network sensor is located upstream sniffing all traffic to the server but that there are two different network paths to your server.

Many companies purchase products that are capable of detecting intrusions; however, the product is improperly deployed. This improper deployment could possibly negate your efforts to effectively monitor attacks. This paper will discuss a proper deployment of Intrusion Detection Systems.

What is Intrusion Detection?

Intrusion Detection is exactly what its name implies; it is a method of detecting intrusions on your network. When the IDS is properly tuned and implemented it can give you a clear picture of what is happening on your network. As today's technology gets more advanced, the frequency and complexity of attacks becomes more apparent.

IDS inform us of the type, source and destination of an attack. IDS are not meant to secure your network, but merely inform you that an attack is taking place. There are some detection systems that may block or kill connections based on whether the rules defined are met.

Types of IDS

Like most products on the market there are varying types of detection systems available. Although, there may be some similarity between each other there are differences. There are three types of IDS available.

Network-based

A network-based IDS monitors network traffic. Basically this type of IDS will sit on the network in promiscuous mode sniffing packets. It will analyze each packet for certain traffic patterns (i.e.: anomaly-based) or signatures (i.e.: signature-based) and act accordingly. Generally when a match is found, actions can be configured. These actions may include alerts via email to scripts launched to kill connections.

Host-based

Host-based IDS monitors an individual computer system based on computer logs generated by the operating system rather than from network traffic. Some host-based systems will monitor the integrity of your file system by taking a snapshot of your file system and scanning on predefined intervals looking for changes to the files that are being monitored. Host-based systems with file integrity are often used for change management purposes as well as intrusion detection.

Hybrid-based

Hybrid-based is a combination of both network and host-based IDS. This type has the same functionality as a host-based system and will also monitor network traffic. The difference between analyses of network traffic in hybrid-based is that it only monitors traffic destined for it and that it will not operate in promiscuous mode.

Sensor and Management Console Placement

Another attribute of IDS deployment is Sensor and Management placement. This task monitors the placement of sensors and consoles ensuring that they are in areas that will allow for filtering of specific traffic.

Management Consoles

The management console should be placed behind some type of filtering device like a Demilitarized Zone (DMZ) dedicated to security architecture or behind a router with access-lists. This will provide a more granular approach to controlling access to the console.

Host Sensors

Placement of host sensors is not too difficult, all you really need to do is identify servers with sensitive information and install the sensor.

Network Sensors

On the other hand, network sensor placement is a little more difficult. As I mentioned earlier, these sensors sniff packets on the network. A good understanding of your network is required since it is not always obvious as to the flow of traffic. Once you have identified source and destinations on your network that the IDS will monitor you need to ensure that you understand the way traffic flows between the two. For example if you want to monitor traffic from one department to another identifying at what place will you be able to see all traffic from both departments or will there be multiple points.

With today's technology most companies have switches deployed in their network rather than hubs. As we know switches do not broadcast traffic to all ports, as in the case of hubs. So the first thing you must do if you are plugging into a switch is ensure that the switch is able to perform port spanning or mirroring, which is the process of configuring a port to span multiple ports or VLANs in an attempt to see all traffic. In the process of port spanning, you must not exceed the port speed limit or the speed of the sensor. An example of this is if I were spanning five 100 MB ports to one 100 MB port my aggregate bandwidth would exceed my one 100 MB port thus dropping traffic. There are some workarounds to address this problem. One product designed to address this is a Top layer switch. This switch acts as a load balancer for the sensors and is very costly. I will not discuss load balancing in detail, but if you would like to learn more about the topic the following is a good site: http://www.toplayer.com/.

Although network sensor placement varies from company to company there are generally two common locations. The first is a sensor located in front of the firewall and the second is behind your firewall. For larger companies purchasing multiple network sensors is not an issue, but for smaller to medium size companies it would become an issue. The decision of where to put your network sensor varies depending on your companies needs as there are pros and cons to both options.

If you were to place the sensor on the outside of your firewall you are electing to have what I call an "early warning" detection system. What I mean by this is that you will get a chance to see everything that is knocking on your door like port scans, vulnerability scans and penetration attempts. The pro to this approach is the fact that in some instances these reconnaissance techniques are prelude to a more serious attack. On the other hand, this will create a lot of overhead for the IDS Administrators since every time an alert is received you would need to check the firewall to see if the attack was able to go through.

The second location discussed, is placing a sensor in front of your firewall. This location will tell you when an attack went through your firewall. This will save the IDS Administrator time analyzing the firewall for penetration. This location will also allow you to see internal malicious behavior from your users, which is good since most attacks happen internally anyway.

If you have budget for only one network sensor you probably do not have budget for many host sensors as well. In this case it is a good idea to place the sensor internally so you can monitor traffic to the devices that contain sensitive information and may not have sensor software. Another practice to adopt would be to place some freeware sensors (i.e. Snort- http://www.snort.com) in areas where the company cannot afford to install commercial sensors.

Deploying IDS (A phased approach)

Phase 1: Hardening the OS

Before you begin deploying your sensors and management console you need to harden the operating system that the sensors will be installed on. Detailed hardening varies from operating system to operating system. The following is a general guideline for doing so:

- Ensure you have physical security to your devices. (i.e.: Data center with limited access, locked servers, etc.)
- Run only needed services.
- Install latest security patches.
- Create difficult passwords (i.e.: alpha numeric, minimum 6 characters, etc.).
- Delete all accounts that are not being used.
- Set appropriate file permissions.
- Ensure no network shares are created on system. If the system needs network shares ensure access is restricted.
- Ensure systems are only accessible to appropriate users and other networking devices.
- If your network has databases, ensure that it is secure.

Once hardening is complete you should run a vulnerability scanner against it to give you an idea of any other holes that may be present on your network. Although the above is a general guideline, the table below lists websites that discuss this in more detail.

Table 1- General Guidelines to Hardening OS

	Type of OS	**URL**
1)	Windows	http://www.fedcirc.gov/docs/ntlockdown.pdf
2)	Linux	http://www.infosyssec.org/infosyssec/linux1.htm
3)	Unix	http://www.cert.org/tech_tips/unix_configuration_guidelines.pdf

Phase 2: Installing the sensor software

Once the OS is hardened you can begin the installation of software. Most IDS software installs are straightforward. It does not take much to get the management console and sensors up and running and communicating to each other; however, there are some things that you need to take into consideration. For example, ensuring that the communication between the sensors and management console are encrypted as well as restricting communication to the appropriate devices. Along with encrypting the communication another safeguard to consider would be to create a management VLAN or segment where only management traffic is permitted. This will ensure that no one can see the monitoring traffic at all.

Phase 3: Testing the IDS Sensors

Once sensors and management consoles are deployed you need to verify they are working properly. IDS' testing encompasses three steps:

- *Step 1: Deployment Testing*

  The installation of the IDS is verified during Deployment Testing. The scope of this task is to validate that the sensors are placed at the proper locations on the network, the software loaded correctly on each host, validate that the consoles can push policy to the sensors, authenticate that sensors can send collection information to the consoles, and verify that the consoles are logging the information to their respective databases.

- *Host Sensor Testing (Picking up events)*

Host Sensor Testing consists of running a vulnerability scanner or penetration tool against the host sensor. After this scan is complete, verify the correct events were triggered. The following are some vulnerability scanners on the market:

Table 2- Vulnerability Scanners

	Name of Tool	**Type of Software**	**URL**
1)	Internet Scanner Vulnerability Scanner	Commercial	http://www.iss.net
2)	Nessus Vulnerability Scanner	Freeware	http://www.nessus.org
3)	Blade Penetration Tool	Commercial	http://www.blade-software.com

- *Network Sensor Testing (Picking up events and proper placement)*

By running a vulnerability scanner or penetration tool, rather than sending attack to the sensor you would send the attack through the network sensors' path so you can test network sensors. For network sensor testing, a destination target would be defined, which will be used to determine if the sensor is reporting alerts and verify their proper location.

It is important to remember if you are running a vulnerability type scanner ensure that your target has some ports open like port 80 or 21. Vulnerability scanners are designed to be efficient. So if a scan is launched and there is no port open the scanner will never send the attack and the sensor will never see one. This applies to most vulnerability type scanners; however, there are tools that would blindly send the attack (i.e. Blade Penetration Tool - Table 2/Item 3).

- *Step 2: Alarm Testing*

The response testing of the sensors is performed during the IDS Alarm Testing Task. The scope of this task is to verify that each IDS console generates an alarm and notification is sent to the designated user via email or any other notification medium that was configured. Alarm testing can be accomplished by using vulnerability scanners with configured alarms to trigger on a certain event then generate that event via the scanner.

- *Step 3: Threat simulation Testing*

The functional and stress testing of the sensors is performed during the IDS Threat Simulation Testing Task. The scope of this task is to verify that the IDS sensors correctly operate under varying loads by generating network traffic containing simulations of threats and targeting this traffic at the sensors. The testing will validate that the traffic logged by each console is representative of the threat traffic generated by the simulation software. This can be accomplished by a combination of a traffic generator and vulnerability scanners.

The traffic generator is run at different speeds while throwing a set of attacks at the sensors. By starting the traffic generator at a small load then slowly accelerating the load, you are able to define the network sensor's threshold and its' capabilities. At the threshold, the sensor will stop reporting alerts indicating that either the sensors or the ports or both are dropping packets. You may even find that the sensors may continue to pick the alerts up but they will be reported as different alerts then the ones being generated by your vulnerability scanner. Understanding how your sensors will respond under different loads and how they alert on different threats is key to the ongoing success of your IDS.

Phase 4: IDS Tuning/Baseline

The next step of a successful installation of an IDS is tuning. Properly tuning a sensor can alleviate thousands of resource hours allocated to researching false positives. The following steps should be performed to properly tune a sensor.

1. **Load Maximum Policy**

After the initial sensor is verified to be operational, the maximum policy should be set to ensure collection of all possible events. Once this is set, the sensor should be allowed to collect information for at least 48 hours.

If you are running the IDS on an internal network running either at 10 or 100 MB you may want to turn off auditing checks on communication mediums that generate common events:

a. Email (i.e.: From, to, data)
b. Http (i.e.: Gets, cookies, etc.)
c. ftp (i.e.: Get, puts, etc.)

If you do not turn some of these features (Auditing policy) off you will eventually run out of room very quickly depending on your database size and network speeds or your event channel from your sensor to collector may die due to excessive traffic.

It is always best practice to run the IDS with maximum policy loaded for a few hours to get an idea of what traffic is classified as "common occurrence."

Characterize Host Sensors

It is important to understand the role of each server on the network that has the host IDS sensor loaded. This information is used to differentiate if selected services and IDS events are actual threats or a function of the resident software applications on each host.

2. Analyze events for noise

When the collection period of event traffic is complete, the information should be exported from each IDS management system into a spreadsheet. There are several ways to analyze this information searching for common events that can be classified as either normal behavior or redundant. Once this is identified, the list should be reviewed with administrative staff responsible for each host to verify that the assumptions were correct.

3. Additional events

With the reduction in noise, the next step would be to determine the other events on the network that should be tracked by the IDS. For example, these events could include communications between specific hosts, or specific service level communications that could be captured by the IDS.

4. Validate false positives

For any event determined to be a false positive the information should be carefully checked to validate it as a mistake, which would eliminate threat condition information.

5. Identify appropriate responses to events

For events defined as valid threats, the next step is to identify the appropriate responses. These responses could be to log the information or to generate an alert. The responses should be tested for all events classified as high priority.

6. Collection and analyze again

IDS tuning is an iterative process and as such it may take significant time to collect the information and make the necessary adjustments. There should be at least 2 to 3 iterations of Steps 3 through 6 to ensure that the IDS is tuned to its most efficient configuration. Results of these tests should be documented for future reference.

Phase 5: Backup and Maintenance procedures

Creating backup and restore procedures should be preformed on all sensors and management consoles.

For all sensors and management consoles a full backup of all files should be preformed at least once a week with an emphasis on the management console. Most IDS consoles have a backend database for storing alerts so if you have a database such as SQL Server or Oracle you may want to make sure your backup software has some type of agent for these databases. (open database files) It is also recommended to use the Open File options to capture the complete state of the system. To decrease the amount of time it takes to rebuild the server in the event of hardware failure, the Disaster Recovery Option should also be used if available. This will allow the restore process to be completed quickly. It is ideal to schedule these backups at times of low traffic. Along with backing up files if you are encrypting your communications you probably have some encryption keys. These keys should be collected and stored in a separate file server in case you need to rebuild a sensor.

Maintenance consist of:

- Database Files

The IDS application's Management Console continuously generates database files. These files represent a complete list of events captured by the IDS sensors and thus should be reviewed and saved indefinitely. You may want to have some type of offsite delivery process in place to account for the possibility of a disaster in your data center.

(1) Archiving/purging

IDS Database files should be archived in the native formats on a monthly basis or a time determined by reviewing your database growth to the amount of storage space. Once files have been successfully archived you can purge them to ensure your database will not outgrow its storage space or configured size. IDS Database file archives should be indefinitely saved.

- Database sizing

The size of your database will be determined by the amount of storage space available; however, this does not mean that if you have 80 GB of storage you can have an 80 GB database. Technically you can configure your database this way, but you want to size your database to a little less then have of your total storage space. This will allow you to perform maintenance on your database locally rather than on a remote system.

- Updating IDS Signatures and patches

New signatures and security patches are released almost daily. Keeping abreast of these changes will allow you to maintain your IDS' integrity. To help you with this you may want to sign up with email lists, such as one for your IDS and the other for your operating system.

Incident Handling, Response, and Analysis

The following is a brief overview of Incident handling, response and analysis, which would be more detailed in your company's security policy.

b) Respond to Events

The IDS should be configured to automatically generate alarms and send these to the IDS Administrators via email or any other notification methods defined in the IDS configuration.

Your company should assign a minimum of two IDS Administrators to monitor the appropriate email accounts and carry pagers that the alarms are configured for. The IDS administrators should respond to events deemed critical within the time set by the Incident Response Policy.

c) Event Analysis

The IDS Administrator should be capable of performing detailed analysis of forensic events using the data collected by the IDS and other tools available on the network. The Administrator should also be trained on an annual basis on forensic analysis and IDS techniques to best prepare for the dynamic threat environment.

d) Event Correlation

Events will come from many different sources of information on the network that include the IDS, firewalls, syslogs, and other systems. The Security Administrators will be responsible to correlate events from these sensors into a logical relational flow of events for analysis. This will become more important over time as the Security Administrators become more familiar with the detailed information coming from each sensor, as they will find that a series of low security events may correlate to a single high security event.

For proper correlation of events ensure that all devices that are generating logs are time synched with each other. If the date and times are not properly synched between all systems you will have no way to determine events in chronological order and there will not be a common denominator to base analysis on these different sensors. There are also an abundance of correlation tools currently on the market. These tools will import system logs from all devices to give you an automated report that compares events, the source of the event, and the destination. Although this technology is still in its infancy and you will still have to manually process data for certain views, it helps you to initially sort through the massive amounts of data the devices provide you with. The following are two examples of these tools:

Table 3- Correlation Tools

	Name of Tool	**Type of Software**	**URL**
1)	NetForensics	Commercial	http://www.netforensics.com/
2)	Security Manager	Commercial	http://www.netiq.com/

e) ***Forensic Evaluation***

Based upon the data collected, the IDS Administrators may need to evaluate forensic information to determine the root causes and maximum impact of the security event. Depending on the severity of the event, your company may decide to take legal action against the perpetrator. Therefore, properly training the IDS Administrators on the methods of collecting and processing forensic information is imperative.

f) ***Incident Handling***

The timing nature of an IDS results in the need for rapid decisions and reactions to each event. For this reason, it is important to prepare incident handling policies and procedures that define how to deal with the information presented by the IDS. The IDS administrator is typically responsible for developing and maintaining these policies and procedures. Approval of these policies and determining the call to action for events are the responsibility of corporate management.

Conclusion

The task of successfully deploying IDS is an ongoing process that encompasses constant awareness of security threats and network events, researching call to actions on events, and minimizing resource hours by integrating automated tools for data collection and analysis. As a security administrator, it's your responsibility to select a good IDS as well as ensuring that you have a proper understanding of deployment efforts and maintaining the IDS. Although budgets will allow for acquiring the best of breed IDS, if you do not implement or constantly maintain the system it will prove to be a useless implementation.

References:

Author *James Mendelsohn*, "Crime, Security, and Privacy", Netsecurity.tqn.com,
Url: http://dcb.sun.com/practices/howtos/intrusion.jsp

Author *Paul Innella*, *Oba McMillan*, *and David Trout*, *with assistance from Rebecca Bace* , " **Managing Intrusion Detection Systems in Large Organizations, Part One** ", Online.securityfocus.com,
Url: http://online.securityfocus.com/infocus/1564

Author By Patrick Mueller and Greg Shipley "To catch a thief", networkcomputing.com,
Url: http://www.networkcomputing.com/1217/1217f1.html

Author John McHugh, Alan Christie and Julia Allen , "Intrusion Detection: Implementation and Operational Issues", www.stsc.hill.af.mil, Url: http://www.stsc.hill.af.mil/crosstalk/2001/jan/mchugh.asp

Author *Unknown Author*, "State of the Practice of Intrusion Detection Technologies", http://www.sei.cmu.edu,
Url:
http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028chap04.html

II. Network Detects

Detect # 1 RPC Exploit Statd - CVE-2000-0666

```
18:03:44.926325 203.XXX.XXX.XXX.766 > 66.XXX.XXX.XXX.111:  udp 56
0x0000 4500 0054 eb4a 0000 3011 9e1c cb8d 82e7E..T.J..0.......
0x0010 425b 7062 02fe 006f 0040 c783 7c6d a969B[pb...o.@..|m.i
0x0020 0000 0000 0000 0002 0001 86a0 0000 0002................
0x0030 0000 0003 0000 0000 0000 0000 0000 0000................
0x0040 0000 0000 0001 86b8 0000 0001 0000 0011................
0x0050 0000 0000                              ....
18:03:45.536325 203.XXX.XXX.XXX.767 > 66.XXX.XXX.XXX.1024:  udp 1076
0x0000 4500 0450 edc7 0000 3011 97a3 cb8d 82e7E..P....0.......
0x0010 425b 7062 02ff 0400 043c 26e5 3d78 e19fB[pb.....<&.=x..
0x0020 0000 0000 0000 0002 0001 86b8 0000 0001................
0x0030 0000 0001 0000 0001 0000 0020 3c74 7022...........<tp"
0x0040 0000 0009 6c6f 6361 6c68 6f73 7400 0000....localhost...
0x0050 0000 0000 0000 0000 0000 0000 0000 0000................
0x0060 0000 0000 0000 03e7 18f7 ffbf 18f7 ffbf................
0x0070 1af7 ffbf 1af7 ffbf 2538 7825 3878 2538........%8x%8x%8
0x0080 7825 3878 2538 7825 3878 2538 7825 3878x%8x%8x%8x%8x%8x
0x0090 2538 7825 3632 3731 3678 2568 6e25 3531%8x%62716x%hn%51
0x00a0 3835 3978 2568 6e90 9090 9090 9090 9090859x%hn.........
0x00b0 9090 9090 9090 9090 9090 9090 9090 9090................
0x00c0 9090 9090 9090 9090 9090 9090 9090 9090................
0x00d0 9090 9090 9090 9090 9090 9090 9090 9090................
0x00e0 9090 9090 9090 9090 9090 9090 9090 9090................
0x00f0 9090 9090 9090 9090 9090 9090 9090 9090................
0x0100 9090 9090 9090 9090 9090 9090 9090 9090................
0x0110 9090 9090 9090 9090 9090 9090 9090 9090................
0x0120 9090 9090 9090 9090 9090 9090 9090 9090................
0x0130 9090 9090 9090 9090 9090 9090 9090 9090................
0x0140 9090 9090 9090 9090 9090 9090 9090 9090................
0x0150 9090 9090 9090 9090 9090 9090 9090 9090................
0x0160 9090 9090 9090 9090 9090 9090 9090 9090................
```

```
0x0170 9090 9090 9090 9090 9090 9090 9090 9090................
0x0180 9090 9090 9090 9090 9090 9090 9090 9090................
0x0190 9090 9090 9090 9090 9090 9090 9090 9090................
0x01a0 9090 9090 9090 9090 9090 9090 9090 9090................
0x01b0 9090 9090 9090 9090 9090 9090 9090 9090................
0x01c0 9090 9090 9090 9090 9090 9090 9090 9090................
0x01d0 9090 9090 9090 9090 9090 9090 9090 9090................
0x01e0 9090 9090 9090 9090 9090 9090 9090 9090................
0x01f0 9090 9090 9090 9090 9090 9090 9090 9090................
0x0200 9090 9090 9090 9090 9090 9090 9090 9090................
0x0210 9090 9090 9090 9090 9090 9090 9090 9090................
0x0220 9090 9090 9090 9090 9090 9090 9090 9090................
0x0230 9090 9090 9090 9090 9090 9090 9090 9090................
0x0240 9090 9090 9090 9090 9090 9090 9090 9090................
0x0250 9090 9090 9090 9090 9090 9090 9090 9090................
0x0260 9090 9090 9090 9090 9090 9090 9090 9090................
0x0270 9090 9090 9090 9090 9090 9090 9090 9090................
0x0280 9090 9090 9090 9090 9090 9090 9090 9090................
0x0290 9090 9090 9090 9090 9090 9090 9090 9090................
0x02a0 9090 9090 9090 9090 9090 9090 9090 9090................
0x02b0 9090 9090 9090 9090 9090 9090 9090 9090................
0x02c0 9090 9090 9090 9090 9090 9090 9090 9090................
0x02d0 9090 9090 9090 9090 9090 9090 9090 9090................
0x02e0 9090 9090 9090 9090 9090 9090 9090 9090................
0x02f0 9090 9090 9090 9090 9090 9090 9090 9090................
0x0300 9090 9090 9090 9090 9090 9090 9090 9090................
0x0310 9090 9090 9090 9090 9090 9090 9090 9090................
0x0320 9090 9090 9090 9090 9090 9090 9090 9090................
0x0330 9090 9090 9090 9090 9090 9090 9090 9090................
0x0340 9090 9090 9090 9090 9090 9090 9090 9090................
0x0350 9090 9090 9090 9090 9090 9090 9090 9090................
0x0360 9090 9090 9090 9090 9090 9090 9090 9090................
0x0370 9090 9090 9090 9090 9090 9090 9090 9090................
0x0380 9090 9090 9090 9090 9090 9090 9090 9090................
0x0390 9090 9090 9090 9090 9090 9090 9090 9090................
0x03a0 9090 9090 9090 9090 9090 9090 9090 9090................
0x03b0 9090 9090 9090 9090 9090 9090 9090 9090................
0x03c0 9090 9090 9090 9090 9090 31c0 eb7c 5989..........1..|Y.
0x03d0 4110 8941 08fe c089 4104 89c3 fec0 8901A..A....A.......
0x03e0 b066 cd80 b302 8959 0cc6 410e 99c6 4108.f.....Y..A...A.
0x03f0 1089 4904 8041 040c 8801 b066 cd80 b304..I..A.....f....
0x0400 b066 cd80 b305 30c0 8841 04b0 66cd 8089.f....0..A..f...
0x0410 ce88 c331 c9b0 3fcd 80fe c1b0 3fcd 80fe...1..?.....?...
0x0420 c1b0 3fcd 80c7 062f 6269 6ec7 4604 2f73..?..../bin.F./s
0x0430 6841 30c0 8846 0789 760c 8d56 108d 4e0chA0..F..v..V..N.
```

0x0440 89f3 b00b cd80 b001 cd80 e87f ffff ff00

Source of the Detect:

This trace was found on my home network (cable modem). Snort was installed on my linux box directly connected to the cable modem.

Detect was generated by:

This is a snort 1.8.3 detect using the snort rule set that is available for download at www.snort.org. The following signature caught the attack:

alert UDP $EXTERNAL any -> $INTERNAL any (msg: "IDS362/shellcode_shellcode-x86-nops-udp"; 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|";)

Probability the source address was spoofed:

The likelihood of the source being spoofed is slim thought it could be possible do to the traffic being udp. It seems like the attacker took action (although it seems automated due the time between packets from the response he got from the port scanning. But it is likely that the source could be a compromised system. Also after researching this detect I found some information backing that the source address was not spoofed. Excerpt from www.whitehats.com

[redacted]

The packet that caused this event is normally a part of an established TCP session, indicating that the source IP address has not been spoofed. If you are using a firewall that supports stateful inspection, and are not vulnerable to sequence number prediction attacks, then you can be fairly certain that the source IP address of the event is accurate. Also, it has been noted that the due to the nature of this event the attacker does not normally require response traffic. In most cases this means that the event should be analyzed along with other supporting data before acting on the event.

Description of the attack:

This is an attack targeted at a format string vulnerability within the rpc.statd program. This daemon is part of the nfs-utils packages that are distributed with a number of popular Linux distributions. (Versions 6.0, 6.1, 6.2.) The specific CVE entry is CVE-2000-0666.

What is rcp.statd?

As stated in the BSD man pages rpc.statd is a daemon that co-operates with rpc.statd daemons on other hosts to provide a status monitoring service. The daemon accepts requests from programs running on the local host (typically, rpc.lockd the NFS file locking daemon) to monitor the status of specified hosts. If a monitored host crashes and restarts, the remote daemon will notify the local daemon, which in turn will notify the local program(s) which requested the monitoring service. Conversely, if this host crashes and restarts, when rpc.statd restarts, it will notify all of the hosts, which were being monitored at the time of the crash. .

Attack mechanism:

This attack works by first connecting to a machine listening on port 111 to determine what port statd is listening on. (rpcinfo –p "machine ip") If you know the exact port you could connect directly to the port running statd. Statd is usually running if you are using NFS as mentioned above. In the above detect the port was found to be 1024 and that is where the exploit is directed. Once connected (as mentioned on the security focus website) the attacker will send a format string that adds executable code (statd.c) into the process address space and overwrites a function's return address, thus forcing the program to execute the code. Now the attacker can execute code with root privileges.

Correlations:

This detect was documented in 2000. All vulnerable versions of the OS have been patched to date.

While searching the Internet, I found multiple exploits dealing with the rpc.statd. Another one was an exploit dealing with an input validation problem.
http://www.ciac.org/ciac/bulletins/k-069.shtml

Scripts are a good way to attack these vulnerabilities. In April of 2001 a Linux worm called Adore was detected. (hybrid of Ramen and the Lion). This worm variant would scan for systems that were vulnerable to the following: LPRng, rpc-statd, wu-ftpd, and BIND.
Once exploited this worm would gain root access and install a backdoor. The worm would also send email with system information to certain addresses. Sans did release a utility to find all backdoors installed by Adore. This utility is named adorefind.
http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm

For more information see
As stated above the CVE number is 2000-0666.
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666

Evidence of active targeting:

This machine does not have any signs of active targeting. Due to the fast times between packets it is probably someone is running a script against network IP ranges, possibly some type of worm or script.

Severity:

Criticality - This machine was not vulnerable to this exploit due to later version of Linux that has been patched and it being my test machine. However, if I was running an un-patched system with rpc-statd loaded my box in a production environment depending on the use the criticality level would be extremely high. **Criticality = 2**

Lethality - This exploit if successful is very lethal. A successful attempt would give the attacker full access on the box . **Lethality = 5**

System Countermeasures - System countermeasures were very high. The box was patched a the highest level. This box was hardened.
System Countermeasures = 5

Network Countermeasures – The box was connected directly to the Time Warner network with no external or internal firewall.
Network Countermeasures = 1

(Criticality + Lethality) – System Countermeasures + Network Countermeasures) = Severity

(2+2) – (5+1) = -2

Defensive recommendation:

Patches/Services: All machines should have up to date security patches and all unused services turned off.
Sensors: An IDS of some sort of Network sensor should be installed configured to look for rpc connects with that particular string.
File Integrity: If your machine is compromised having file integrity software loaded would give you an idea to when and what was compromised on the machine.
I firewall would also be good blocking all incoming ports that were not needed for business functionality.

Multiple choice test question:

What command can you use to see what certain rpc's are listening on?

a) rcpinfo -p

b) rcp
c) netstat
d) inforpc

Answer - The best answer is **a**. Rpcinfo can tell you a lot of information about the rcp daemon.

Detect # 2 Port Scan (FIN)

```
 18:02:21.756325 XXXXXXX.hawaii.rr.com > YYYYYYY.hawaii.rr.com: icmp: echo 
request
18:02:21.776325 YYYYYYY.hawaii.rr.com > XXXXXXX.hawaii.rr.com: icmp: echo 
reply
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.32775: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.1650: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.73: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.1376: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.185: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.525: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.235: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.480: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.6141: F 
0:0(0) win 2048
18:02:22.086325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.1349: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.447: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.644: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.314: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.547: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.1505: F 
```

```
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.138: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.1448: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.802: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.184: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.828: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.801: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.62: F 
0:0(0) win 2048
18:02:22.136325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.261: F 
0:0(0) win 2048
18:02:22.156325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.809: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.7100: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.323: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.437: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.96: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.354: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.929: F 
0:0(0) win 2048
18:02:22.166325 XXXXXXX.hawaii.rr.com.39861 > YYYYYYY.hawaii.rr.com.638: F 
0:0(0) win 2048
```

Source of the Detect:
This data was gathered at my home network (Cable modem). Snort was loaded on my linux box directly connected to the cable modem.

Detect was generated by:
This detect was found by snort and is in tcpdump format.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flags: F;
reference:arachnids,27; classtype:attempted-recon; sid:621; rev:1;)
```

Probability the source address was spoofed:
The probability of the source address being spoofed is very low. This detect is for reconnaissance. The attacker would need to see the returned information. There is a remote possibility that the address might be spoofed. In this case the attacker might be using passive reconnaissance. He could be on the same net sniffing the return traffic which is possible with cable modem networks.

Description of the attack:
Port scanning is one of the most popular ways to tell what ports or services are running on a machine. When identifying these ports you are locating possible open doors to a system. There are many types of scans this particular one is a fin scan. A fin scan is a packet that is sent to a particular port with the fin bit set. Usually these packets are sent when you are closing a connection. In this case when you are sending a fin packet first you are attempting to close a connection that has not been opened. Based on the response you can tell if the port is open or not. If the port is not listening, the system will send back an error (rst) message. If the port is listening, the system will not respond at all.

Attack mechanism:
The attacker is performing reconnaissance on this particular machine. The next steps for the attacker would be based on the information he receives on what ports are open. If a well-known port is open like say port 25 the attacker would concentrate on smtp exploits.

This scan could also be used as decoy for a more harmful attack. Often attackers will flood the network with port scans to hide what they really intend to do. Though the attack is eventually found it gave the attacker a head start.

This particular detect was probably running a type of script or scanning tool due to the time of events being very close and the static source port. It also looks like the attacker preformed a ping first to possibly determine the OS or to see if the target is alive.
There are many port scanners out there today. A popular scanning tool and easiest to use is nmap. This tool can be used with all flavors of unix and there is also a version for Windows. Namp can be downloaded at http://www.insecure.org/nmap.

If you would like to found more about other port scanners please visit:
http://www.apocalypseonline.com/security/tools/tools.asp?exp_category=Scanners

Correlations:
This detect is well known and documented throughout the security community. For more information on port scanning the following site can be useful:

Overall information on port scans
http://www.cs.wright.edu/~pmateti/Courses/499/Probing/

Port scanning information with Namp
http://www.insecure.org/nmap/nmap_doc.html

General information on port scanning (Q&A)
http://www.abuse-guidance.com/portscanfaq.htm

Evidence of active targeting:
This detect is definitely active targeting. After researching the attackers IP address I found out that this attacker was a co-worker of mine testing my home IDS.

Severity:
Criticality – This machine is a test machine with no important data on it. **Criticality = 1**

Lethality – This attempt to gather information is not very harmful. **Lethality = 1**

System Countermeasures – This box was hardened so system countermeasures were high. **System Countermeasures = 5**

Network Countermeasures - IDS detected the scan alerting me to a possible future attack. From here I could monitor the source address or I could even block the address by installing a personal firewall.
Network **Countermeasures = 5**

(Criticality + Lethality) – System Countermeasures + Network Countermeasures) = Severity

(1+1)-(5+5) = -8

Defensive recommendation:
Detecting that someone might be planning an attack is the one of the main defenses along with keeping up to date with all patches.

To detect the port scans you would need an IDS or some type of port scan logging. Keeping up with service patches can become time consuming especially with windows machines. A useful tool to incorporate in your network is Shavlik. This utility will scan all windows machines for missing service patches. You may also want to register with your operating systems emailing list for the latest service releases. Having a perimeter firewall that has the capability of state full inspection should block all fin scan traffic as well.

Multiple choice test question:
When sending a fin port scan what response would you receive if the port were open?

a) You would receive errors. (rst)

b) If sending on a windows machine you would receive a doctor Watson error.

c) You would receive a SYN-ACK

d) You would receive nothing.

Answer – The answer is **d**. When a port is open a packet with a fin is of the norm so an open port will do nothing otherwise if the port is closed it will send a icmp port unreachable.

Detect # 3 255.255.255.255 Source Address (Incident Log 2002.4.14)

```
08:58:16.134488 255.255.255.255.31337 > 78.37.83.81.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum 529f!)
0x0000          4500 002b 0000 0000 0e06 529f ffff ffff         E..+......R.....
0x0010          4e25 5351 7a69 0203 0000 0000 0000 0000        N%SQzi..........
0x0020          5014 0000 06c7 0000 636b 6f00 0000             P.......cko...
09:01:31.004488 255.255.255.255.31337 > 78.37.177.36.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum f2cc!)
0x0000          4500 002b 0000 0000 0e06 f2cc ffff ffff         E..+...........
0x0010          4e25 b124 7a69 0203 0000 0000 0000 0000        N%.$zi..........
0x0020          5014 0000 a6f4 0000 636b 6f00 0000             P.......cko...
09:14:09.984488 255.255.255.255.31337 > 78.37.185.215.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum eb17!)
0x0000          4500 002b 0000 0000 0e06 eb17 ffff ffff         E..+...........
0x0010          4e25 b9d7 7a69 0203 0000 0000 0000 0000        N%..zi..........
0x0020          5014 0000 9f3f 0000 636b 6f00 0000             P....?..cko...
09:56:28.124488 255.255.255.255.31337 > 78.37.179.19.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum f0dd!)
0x0000          4500 002b 0000 0000 0e06 f0dd ffff ffff         E..+...........
0x0010          4e25 b313 7a69 0203 0000 0000 0000 0000        N%..zi..........
0x0020          5014 0000 a505 0000 636b 6f00 0000             P.......cko...
09:57:49.014488 255.255.255.255.31337 > 78.37.219.95.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum c891!)
0x0000          4500 002b 0000 0000 0e06 c891 ffff ffff         E..+...........
0x0010          4e25 db5f 7a69 0203 0000 0000 0000 0000        N%._zi..........
0x0020          5014 0000 7cb9 0000 636b 6f00 0000             P...|...cko...
10:11:31.004488 255.255.255.255.31337 > 78.37.204.213.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum d819!)
0x0000          4500 002b 0000 0000 0e06 d819 ffff ffff         E..+...........
0x0010          4e25 ccd5 7a69 0203 0000 0000 0000 0000        N%..zi..........
0x0020          5014 0000 8c41 0000 636b 6f00 0000             P....A..cko...
11:17:25.054488 255.255.255.255.31337 > 78.37.199.239.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum dcff!)
0x0000          4500 002b 0000 0000 0e06 dcff ffff ffff         E..+...........
0x0010          4e25 c7ef 7a69 0203 0000 0000 0000 0000        N%..zi..........
0x0020          5014 0000 9127 0000 636b 6f00 0000             P....'..cko...
11:24:40.034488 255.255.255.255.31337 > 78.37.97.191.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum 452f!)
0x0000            4500 002b 0000 0000 0e06 452f ffff ffff          E..+......E/....
0x0010            4e25 61bf 7a69 0203 0000 0000 0000 0000          N%a.zi..........
0x0020            5014 0000 f956 0000 636b 6f00 0000               P....V..cko...
11:27:52.074488 255.255.255.255.31337 > 78.37.185.180.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum eb3a!)
0x0000            4500 002b 0000 0000 0e06 eb3a ffff ffff          E..+.......:....
0x0010            4e25 b9b4 7a69 0203 0000 0000 0000 0000          N%..zi..........
0x0020            5014 0000 9f62 0000 636b 6f00 0000               P....b..cko...
11:41:25.024488 255.255.255.255.31337 > 78.37.218.87.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum c999!)
0x0000            4500 002b 0000 0000 0e06 c999 ffff ffff          E..+............
0x0010            4e25 da57 7a69 0203 0000 0000 0000 0000          N%.Wzi..........
0x0020            5014 0000 7dc1 0000 636b 6f00 0000               P...}...cko...
11:51:46.034488 255.255.255.255.31337 > 78.37.239.85.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum b49b!)
0x0000            4500 002b 0000 0000 0e06 b49b ffff ffff          E..+............
0x0010            4e25 ef55 7a69 0203 0000 0000 0000 0000          N%.Uzi..........
0x0020            5014 0000 68c3 0000 636b 6f00 0000               P...h...cko...
12:21:22.044488 255.255.255.255.31337 > 78.37.88.65.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum 4daf!)
0x0000            4500 002b 0000 0000 0e06 4daf ffff ffff          E..+......M.....
0x0010            4e25 5841 7a69 0203 0000 0000 0000 0000          N%XAzi..........
0x0020            5014 0000 01d7 0000 636b 6f00 0000               P.......cko...
12:53:04.054488 255.255.255.255.31337 > 78.37.192.216.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum e416!)
0x0000            4500 002b 0000 0000 0e06 e416 ffff ffff          E..+............
0x0010            4e25 c0d8 7a69 0203 0000 0000 0000 0000          N%..zi..........
0x0020            5014 0000 983e 0000 636b 6f00 0000               P....>..cko...
13:15:19.044488 255.255.255.255.31337 > 78.37.136.101.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum 1d8b!)
0x0000            4500 002b 0000 0000 0e06 1d8b ffff ffff          E..+............
0x0010            4e25 8865 7a69 0203 0000 0000 0000 0000          N%.ezi..........
0x0020            5014 0000 d1b2 0000 636b 6f00 0000               P.......cko...
13:17:52.094488 255.255.255.255.31337 > 78.37.13.122.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum 9876!)
0x0000            4500 002b 0000 0000 0e06 9876 ffff ffff          E..+.......v....
0x0010            4e25 0d7a 7a69 0203 0000 0000 0000 0000          N%.zzi..........
0x0020            5014 0000 4c9e 0000 636b 6f00 0000               P...L...cko...
13:24:01.084488 255.255.255.255.31337 > 78.37.240.118.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum b37a!)
0x0000            4500 002b 0000 0000 0e06 b37a ffff ffff          E..+.......z....
0x0010            4e25 f076 7a69 0203 0000 0000 0000 0000          N%.vzi..........
0x0020            5014 0000 67a2 0000 636b 6f00 0000               P...g...cko...
13:25:58.104488 255.255.255.255.31337 > 78.37.197.119.515: R 0:3(3) ack 0 win 0 (ttl 14,
id 0, bad cksum de79!)
0x0000          4500 002b 0000 0000 0e06 de79 ffff ffff         E..+.......y....
0x0010          4e25 c577 7a69 0203 0000 0000 0000 0000         N%.wzi..........
0x0020          5014 0000 92a1 0000 636b 6f00 0000              P.......cko...
13:26:52.084488 255.255.255.255.31337 > 78.37.232.34.515: R 0:3(3) ack 0 win 0 (ttl 14, id
0, bad cksum bbce!)
0x0000          4500 002b 0000 0000 0e06 bbce ffff ffff         E..+............
0x0010          4e25 e822 7a69 0203 0000 0000 0000 0000         N%."zi..........
0x0020 5014 0000 6ff6 0000 636b 6f00 0000     P...o...cko...
```

Source of Trace:
This trace was downloaded from the incidents.org/logs/Raw site.

Detect was generated by:
This detect was generated by a Snort Rule set and logged in tcpdump format.

Possibility the source IP address was spoofed:
The source IP address was spoofed. It is not of the norm for the source IP address to be a broadcast address. This packet was probably crafted.

Description of the Attack:
I am not sure to the intent of this detect. With the source address as a broadcast address and the source port the same on every packet it is safe to say this is some type of packet craft. At first glance this attack seems to be probing for devices with port 515 open. If the attacker were probing you would need a response, these packets have the RST and ACK bits set. This will get you no response. Even if the SYN packet were set, the attacker would have to be on that broadcast domain to receive a response. Possible a cable or DSL modem network. Another interesting item is the source port, which is usually associated with the Back Orifice program, but if the attacker were searching for Back Orifice is would be destination port. The check sums all seem to be invalid as well. So sending packets without intending to receive a response could indicate some type of remote control command like program or covert channel. A covert channel is a communication channel designed to bypass a security policy (firewall device) to go undetected.

Attack Mechanism:
I am not fully sure of the exact classification of the so a will address to possibilities, one being a covert channel and the other being some type of decoy. A covert channel as I mentioned before is way to send information back and forth in a stealthy way to avoid firewalls and ID systems. To create a covert channel you must have control of a remote system to install client software. So the system would need to be compromised before client software can be installed. This client software, once installed, will be used to initiate a connection to a specified control server with specified ports to communicate through.

The port selected is usually a port that is permitted through most firewalls like port 80 or 53. In this detect there is a bunch of traffic coming in on port 80 (https) and port 1080 (proxy). Some ways the data can be embedded in the packet are within the ip packet ident field, tcp initial sequence number field and in the tcp acknowledge sequence number field. If it were a decoy, the attack works by creating some type of packet that will trigger an alert from an ID System. The analysis would spend most of his or her time analyzing it that some other attack would go unnoticed.

Correlations:
This detect was sighted by a variety of people. Some of the detects had different flags set but source port and address were the same along with the destination port (515).

http://lists.jammed.com/incidents/2001/07/0025.html
http://lists.jammed.com/incidents/2001/04/0092.html
http://www.incidents.org/archives/intrusions/msg00020.html

Evidence of Active Targeting:
This detect has no evidence of active targeting to a specific IP address. The destination IP addresses are random, thought the source port is the same. The attack does have evidence of active targeting to a specific port open.

Severity:
Criticality – This detect itself is not very harmful, though it could be evidence of something more to come. **Criticality = 1** (If it were a covert channel the lethality would equal 5)

Lethality – The detect is not very harmful. **Lethality = 1** (though if it were a covert channel it would be very lethal equaling a 5.

System Countermeasures –Iit is not clear as to the level of protection
I would ensure all patches are installed and only open ports that are needed to perform server function properly. **System Countermeasures = 3**

Network Countermeasures – It is not clear as to the level of protection. Even from the detect I can not determine the level of protection though I do notice a hitbox gateway in existents. This hitbox does web traffic analysis including ssl control. I also see no response so there may be some filters blocking broadcast address as source address.
Countermeasures = 3

(Criticality + Lethality) – System Countermeasures + Network Countermeasures) = Severity

(1+1)-(3+3) = -1

Defensive recommendation:
Implementing Anti-spoofing at your perimeter would eliminate this detect from penetrating you network. Also ensure that you have only ports that are needed open. As well as having a knowledgeable staff to analysis this detect fully.

Questions from Incidents.org

1. What service is associated with port 515?
    a. LPR daemon (printers)
2. Could this be a response?
    a. Though the flags (RST and ACK) are set to indicate a response to a RST I do not know of any device that will send an IP packet with the source address as a broadcast address.
3. Is there any other way you might see the source address be a full broadcast like this? (think about routers here!!)
    a. After further research I still do not know of any device that will send a packet with the source address as a broadcast address. Certain devices will use 0.0.0.0 for some implementations of DHCP but no broadcast address.

Multiple choice test question:
How could an attacker use a broadcast address 255.255.255.255 as a source and receive a response from that packet:

a) If the attacker was on the same broadcast domain.
b) If the attacker was on a cable modem network
c) If the attacker was less then 3 hops away
d) It the attacker created a ricochet broadcast network

Answer: Choice **a** and **b** are correct. The attacker would need to be on the same broadcast network due to routers not being able to forward broadcast. Cable modem networks are created as flat network. It is usually one big broadcast domain.

III. "Analyze This" Scenario

I have conducted an analysis of 5 consecutive days worth of snort IDS logs. The dates that were chosen are April 02, 2002 through April 06, 2002. This report contains a list of all events with a brief description within the 5 days describe above, a top talkers list and a list of 5 external IP addresses with registration information. This report also includes a detailed description and correlation of the top 10 events along with insights to the internal network. I have also provided general defense recommendations that should be followed for good security practice. This report only gives you an assessment of the findings and general recommendations for those 5 days worth of logs. For a more in depth security assessment you should perform vulnerability scans on all hosts contained in your network as well as an analysis of your network security architecture, war dialing to detect modem lines, interviews with personal, collecting documents, penetration testing and examining system configurations on key systems.

Alert Files	Scan Files	Out of Spec Files
alert.020402	scans.020402	oos_Apr.2.2002
alert.020403	scans.020403	oos_Apr.3.2002
alert.020404	scans.020404	oos_Apr.4.2002
alert.020405	scans.020405	oos_Apr.5.2002
alert.020406	scans.020406	oos_Apr.6.2002

Over a period of five days the IDS has detected 82 events. Some of these events are informational while others pose serious threats to the security of your network. There are quite a few systems that will need to be investigated to determine if they were compromised. With the data analyzed it is clear that the security of the network needs improvement. As you read this document you will see the seriousness of these events as well as an idea on where to start your remediation.

The following is a listing of all alerts within the 5 days. Most of the descriptions are taken from the www.Whitehats.com website. Visit this site if you would like to learn more about these events.

Signature	Alerts
Connect to 515 from inside	537343
Connections from internal networks to the LPR port, 515/tcp. This is potentially serious as it could be an attempt to located vulnerable hosts running LRP	
SNMP public access	226476
An SNMP community name is the default (e.g. public), null, or missing. This is potentially serious as devices are now exposed to the many vulnerabilities of SNMP.	
SMB Name Wildcard	153860
This signature detects attempts to enumerate shares on a system running SMB file sharing (usually Microsoft Windows platforms). This can be a precursor to an attack, as it can potentially reveal a great deal about the target system, such as usernames and poorly protected shared resources	
ICMP Echo Request L3retriever Ping	77144
This event indicates that someone may be scanning your network using the L3 "Retriever 1.5" security scanner. This legitimate security tool is for authorized security assessment and should not be used on unauthorized networks.	
INFO MSN IM Chat data	48664
This signature detects data from the popular MSN program by Microsoft. This is an instant message program that will allow you to chat with someone using the same program	

INFO Inbound GNUTELLA Connect request 38814

This signature will detect inbound connection requests of the GNUTELLA program that is a popular peer-to-peer network program.

High port 65535 udp- possible Worm -traffic 29934

This signature detects udp connections to port 65535, which is a good indication of some type of worm traffic.

MISC Large UDP Packet 23212

This event indicates that an abnormally large UDP packet was sent to your server. This may indicate a denial of service attack or the use of a covert channel

ICMP Echo Request Nmap or HPing 13032

This signature detects an icmp request packet that is usually used by Nmap or Hping used to fingerprint a networking device.

Watchlist 000220 IL-isdnnet-990517 9034

This signature looks for IP addresses coming from Israel and China.

FTP DoS ftpd globbing 8286

Wu-Ftpd allows for clients to organize files for ftp actions based on file globing patterns. The implementation of file globing included in Wu-Ftpd contains a heap corruption vulnerability that may allow for an attacker to gain remote root access.

ICMP Fragment Reassembly Time Exceeded 4342

This signature detects packets with the ICMP fragment reassembly time exceeded set. This packet is usually a sign of possible attempt to avoid an IDS system.

Web-IIS view source via translate header 3124

This event indicates that a remote intruder has attempted to exploit the default IIS functionality to view the source of scripts on a server

icmp router selection 3220

This event indicates an attempt to add a default route to your Windows machine, possibly for denial of service attacks. An attacker can add default route entries into a remote system

NMAP TCP ping! 2808

This signature detects the presents of Nmap sending a TCP ping packets used for some type of reconnaissance of your network.

Web-MISC Attempt to execute cmd 1780

This signature detects the cmd.exe file in a get request of a web page. Many worms today use the cmd.exe command to exploit a Windows IIS server.

INFO Outbound GNUTella Connect request 1580

This signature will detect outbound connection requests of the GNUTELLA program that is a popular peer-to-peer network program.

Watchlist 000222 NET-NCFC 826

This signature looks for traffic originating from a specific network rancge in china. (159.226.x.x)

ICMP Echo Request Windows 768

This signature detects a icmp echo request packet coming from a Windows type operating system.

WEB-IIS_vti_inf access	694

This signature detects a reconnaissance attempt on possible vulnerable IIS servers.

WEB-Frontpage_vti_rpc access	664

This signature detects a possible attempt to create a denial of service on a Web server running Frontpage. If the attack is successful the server would require a reboot to gain normal functionality.

Null scan!	460

This signature detects a type of reconnaissance where the packet send has 0 flags set in an attempt to identify open ports. According to RFC 793 a system should send back an RST for all TCP ports closed when they receive a packet without any specified IP flags for a specific port. This scan will not work with Microsoft 95/NT machines because Microsoft did not follow the RFC.

Info Napster Client Data	294

This signature detects a Napster client on your network. Napster is a popular mp3 file-sharing program.

Scan Proxy Attempt	*270*

This signature detects a scan which is looking for any type of proxy server. This is a form of reconnaissance.

ICMP Destination Unreachable (Communication Administratively Prohibited)	244

This signature detects a packet was sent by a networking device telling the source address that destination is not reachable do to an administrator specifically denying the packet.

Signature	**Alerts**

ICMP traceroute 208

This signature detects an icmp packet which is being used to run the traceroute command. This is a form a reconnaissance as it is capable of mapping a route a packet takes to get to a certain destination. This information can be helpful in mapping your network.

INFO Napster Login *194*

This signature detects Napter login traffic. Napster is a popular peer to peer program used to share files. (MP3)

Possible trojan server activity 184

This signature detects packets that indicate some sort of Trojan activity. This event does not guarantee you have a Trojan just a possibility.

INFO Possible IRC Access 152

This signature detects possible Internet Relay Chat traffic. IRCs are used in many distributed attacks.

INFO FTP anonymous FTP *128*

This signature detects any ftp traffic using the anonymous account to gain access. It is usually best security practice to not configure anonymous ftp unless you have to.

ICMP Echo Request BSD *120*

This signature detects any icmp echo request packets (Ping) coming from a BSD source that might be running some type of packet crafting software.

Queso fingerprint *112*

This signature detects an attacker using a tool called Queso to fingerprint the OS of a system.

INFO- Possible Squid Scan *96*

This signature detects a proxy scan specifically a Squid proxy with runs on linux.

MISC traceroute *94*

The signature detects a traceroute being performed on a network. A traceroute can be used as a type of reconnaissance to map out a network.

WEB-MISC compaq nsight directory traversal 80

This signature detects Web access trying to expoit Compaqs Insight manager software via a directory traversal. If successful the attack could take over the machine.

IDS552/web-iis_IIS ISAPI Overflow ida nosize 78

This signature detects a vulnerability used by a variety of worms attacking IIS.

SUNRPC Highport access! 72

This signature detects possible RPC (Remote Procedure Call) is a protocol that allows a program on one computer to use a service from a program located in another computer. RPC has a long history of exploits

Expoilt NTPDX buffer overflow 58

This signature detects packets trying to exploit the buffer overflow vulnerability in the NTP daemon running on unix type systems.

Scan Sysnscan Portscan ID 19104 56

This signature detects port scans coming from a tool called synscan which is used a type of reconnaissance.

Back Orifice 56

This signature detects a probe to UDP port 31337 looking for the existence of the Back orifice Trojan which allows the attacker to take over Windows hosts.

Attempted Sun RPC high port access 56

This signature detects an attempt to exploit Sun Remote procedure call. There are many vulnerability associated with the rpc protocol.

INFO napster upload request 44

This signature detects an upload request for the napster program. Napster is a popular mp3 file sharing program.

MYPARTY- Possible My Party infection 44

This signature detects the my party virus. When this virus is executed it will compromise a machine installing a backdoor then will email itself out via outlook address book.

Exploit x86 NOOP 38

This signature detects a string of the character 0x90. Depending on the context, this usually indicates the NOP operation in x86 machine code. Many remote buffer overflow exploits send a series of NOP (no-operation) bytes to pad their chances of successful exploitation.

Web-MISC http directory traversal *34*

This signature detects a directory traversal directed at web server. This attack will try to exploit a vulnerability in the web server daemon or a cgi script. If successful the attacker could take over your machine.

Exploit X86 setuid 0 *34*

This signature detects data resembling the x86 assembly code to change the group identity to 0.

WEB-CGI scriptalias *32*

This signature detects an attempt to exploit the scriptalias bug to view the source of CGI scripts that are normally only executable.

High port 65535 tcp- possible Red Worm -traffic *30*

This signature detects the high port 65535 port which is used by the red worm virus. This virus attacks a vulnerability in an IIS indexing server causing a buffer overflow.

WEB-MISC 403 Forbidden *30*

This signature detects someone trying to access a website that they do not have access to. This may be a type of reconnaissance.

Signature **Alerts**

Port 55850- Possible myserver activity –ref. 010313-1 *26*

This signature detects a possible attack from the myserver DDOS. This attack listens on port 55850.

Exploit x86 stealth noop *26*

This signature detects an that an attacker attempted to overflow one of your daemons with jmp 0x02 "stealth nops" This may be triggered by several possible exploits.

Backdoor NetMetro File List *22*

This signature detects a backdoor Trojan may be operating on a host.

ICMP Destination Unreachable (Protocol Unreachable) *20*

This signature detects a icmp destination unreachable packet responding to a ping. This may be an indication someone is doing reconnaissance on your network.

Exploit x86 setgid 0 *18*

This signature detects a shellcode trying to set a group identity to 0. In unix seting this to 0 give a person root level privileges.

Scan Fin *14*

This signature detects a scan with the only the fin flag set. This could be a type of reconnaissance being done on your network.

Incomplete Packet Framents Discarded *12*

This signature detects fragmented packets that have been discarded. Fragmented packets could be used to bypass an Intrusion Detection System.

Virus- Possible scr Worm *10*

This signature detects mail carrying a src extension which can possibly contain some type of worm virus.

Probable NMAP fingerprint attempt *10*

This signature detects NMAP traffic. NMAP is a popular port scanning tool used for reconnaissance. Fingerprinting will try to figure out your operating system.

WEB-IIS encoding access *10*

This signature detects that an intruder has sent an invalid hex sequence. This may be an attempt tp circumvent access control. IIS allows for invalid hex sequences. Example: %1u%1u translates to ".."

Port 55850 tcp – Possible myserver activity –ref. 010313-1 *10*

This signature detects the myserver program. This program is a DDOS which binds to UDP 55850.

MISC PCAnywhere Startup *8*

This signature detects traffic pertaining to PCAnywhere. PCAnywhere is a popular remote control program used to gain access remotely as if you where at the machine itself. If passwords are not set an attacker could use this program maliciously.

RPC tcp traffic contains bin_sh) 8

This signature detects someone trying to open a root shell on system.

RFB- Possible WinVNC – 010708-1 8

This signature detects WinVNC traffic. VNC is a free remote control program similar to PCAnywhere.

INFO Outbound GNUTella Connect accept 6

This signature detects outbound connection accepts from GNUTELLA program. GNUTELLA is a popular peer-to-peer network program.

TFTP- Eternal UDP connection to internal tfpt server 6

This signature detects a connection originating from the outside to a TFTP server inside. TFTP is Trivial File Transport Protocol usually used with Cisco devices to download software. The tftp protocol has also been used by worms to transport itself. TFTP traffic should not be originating from the outside.

x11 outgoing 6

This signature detects that an XTERM session was initiated, sending the output to an external x-server. This is considered insecure traffic and it is often a sign of compromise.

WEB-MISC ICQ Webfront HTTP DOS 6

This signature detects a web application attack on a http server.

INFO Inbound GNUTella Connect accept 6

This signature detects inbound connection accepts from the GNUTELLA program. GNUTELLA is a popular peer-to-peer network program.

WEB-CGI redirect access 6

This signature detects malicious traffic to ColdFusion , a web application. This attack will append stale query string arguments to a URL during HTML redirection, which may provide sensitive information to the redirected site.

MISC source port 53 to < 1024 *6*

This signature detects that an attacker is making a connection to a privileged port using the source port 53 (dns). This should not normally occur. Old or misconfigured packetfilters may allow the connection if they allow all dns traffic.

WEB-IIS Unathorized IP Access Attempt *4*

This signature detects an unauthorized IP access attempt on a Microsoft IIS server. This could show a possible attack to an IIS server.

MISC Invalid PCAnywhere Login *4*

This signature detects someone failing to login to a PCAnywhere device

TELNET Access *4*

This signature detects someone connecting to a device via telnet. Telnet is a very unsecured protocol. You might want to use ssh which is encrypted.

TFTP – Internal UDP connection to external tftp server *4*

This signature detects a connection originating from the inside to a TFTP server outside. TFTP is Trivial File Transport Protocol usually used with networking devices to download software. The tftp protocol has also been used by worms to transport itself. TFTP traffic should not be leaving your internal network.

WEB-IIS asp-dot attempt *4*

This signature detects an attempt to attack a web application running on Microsoft IIS server using an asp exploit.

Suspicious host traffic *4*

This signature detects traffic coming from a host that is not of the norm. It might be worth looking into the host or hosts generating this traffic.

IDS475/web-iis_web-webdav-propfind

4

This signature detects that a remote user has attempted to use the webdav PROFIND directive to retreive a directory listing on the web server. This may allow an attacker to gain knowledge about the web server that could be useful in an attack.

TCP SMTP Source Port traffic

4

This signature detects someone connecting to a device with the source port 25. This is unusual traffic. A source port should usually be above 1024. This could be an attempt to bypass some packet filters.

WEB-CGI formmail access

4

This signature detects that a query was made to the formmail CGI program that could allow an attacker to execute arbitrary commands on the server. A vulnerability exists because shell metachars are not properly quoted in the form field parameters.

WEB-MISC webdav search access

2

This signature detects that a remote user has attempted to use the SEARCH directive to retreive a list of directories on the web server. This may allow an attacker to gain knowledge about the web server that could be useful in an attack.

x86 NOOP – unicode Buffer Overflow Attack

2

This signature detects that a string of the character 0x90 was detected. Depending on the context, this usually indicates the NOP operation in x86 machine code. Many remote buffer overflow exploits send a series of NOP (no-operation) bytes to pad their chances of successful exploitation.

ICMP Router Selection (Undefined Code!)

2

This signature detects an icmp packet with the router selection type set along with some undefined code. This could be a crafted packet send by an attacker to bypass a security defense.

List of Top 10 Conversations (Top Talkers Events)

The following is the top conversions during the 5-day period of analysis.

# of Conversions	Source IP	Destination IP
283228	10.1.150.83	10.1.151.77
70636	10.1.153.164	10.1.150.198
24987	10.1.153.126	10.1.150.198
17900	10.1.153.119	10.1.150.198
10522	10.1.153.136	10.1.150.198
9683	10.1.88.203	10.1.150.195
9601	10.1.88.207	10.1.150.195
9565	10.1.88.145	10.1.150.195
9557	10.1.88.181	10.1.150.195
9516	10.1.88.159	10.1.150.195

List of Top 10 Sources Scans

# of Scans	Source IP
445185	10.1.60.43
250064	10.1.150.143
160523	10.1.6.45
150858	10.1.6.49
145262	10.1.6.52
142804	10.1.6.48
125157	10.1.11.8
115735	10.1.6.50
63952	10.1.6.53
60208	10.1.6.60

Details of source port scans:

1. 10.1.60.43- Due to most of the traffic generated by this IP address having a source port of 123 it is highly likely this is an NTP server. NTP which stands for Network time protocol is a time sync program. It is usually used to synch all networking devices in your network. These port scans are normal traffic if ntp is running in your network. NTP

programs are often exploited so implementing some type of access-list only allowing only your ntp server is recommended.

2. 10.1.150.143- The majority of port scan traffic from this ip address is port 4665 and 4662 which are ports usually used by peer to peer file sharing programs specifically edonkey. This machine should be investigated to file sharing programs installed on it. Most of the traffic is destined to outside IP addresses which would created bypassing of a firewall for unauthorized data.

3-6. 10.1.6.45, 49, 48, 52, 50 and 53 and 48- Most of the traffic generated by these IP address is to port 7000 and 7001. There are numerous programs that use this port to communicate. This traffic could possibly be initiated by a chat program or some type of IRC (AOL) . It could also be from a java program called Weblogics. This program listens on port 7001. The above is possible but a more sound reason for this traffic is it is being generated by IBM' AFS file servers. Port 7000 and 7001 are well-known ports for these file servers. Port 7000 the file server itself and 7001 is for callbacks to cache managers. This traffic is probably legitimate traffic.

7. 10.1.11.8- This IP address is the source of port scan traffic to port 1346 and 1347. The well-known port for 1346 is Alta Analytics license manager and for port 1347 is multi media conferencing. This source did not trigger an event during the 5-day period of analysis. Most of this traffic is directed to the .152 network.

Out of Spec Data

The following is a summary of Out of spec data. This traffic is traffic that was not expected. Most of the traffic that was generated was from some kind of file sharing program.

The majority of oos traffic have destination ports of:
Port 4662 – This is traffic is usually from the file sharing program called edonkey.
Port 6346 – This is traffic is usually from the file sharing program called gnutella.
Port 1214 - This is traffic is usually from the file sharing program called KaZaA.
Port 80 – This is web traffic.

Most of the out of spec data consisted of bogus tcp flags set (sf, sfrp, sfrpu) , TCP sequence number the same.

Example of oos with destination port 4662.

The packets all had the reserved bits and SYN flag set. The other characteristics of the packet appeared to be normal.

```
04/03-09:15:20.229595 217.80.78.17:52113 -> MY.NET.150.143:4662
TCP TTL:53 TOS:0x0 ID:17778  DF
21S***** Seq: 0x8B7122E   Ack: 0x0   Win: 0x16B0
TCP Options => MSS: 1412 SackOK TS: 88037212 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
04/03-10:15:54.729481 217.80.78.17:53167 -> MY.NET.150.143:4662
TCP TTL:53 TOS:0x0 ID:49260  DF
21S***** Seq: 0xED99234A   Ack: 0x0   Win: 0x16B0
TCP Options => MSS: 1412 SackOK TS: 88400678 0 EOL EOL EOL EOL
```

Example of oos with destination port 6346.

The packets all had the reserved bits and SYN flag set. The other characteristics of the packet appeared to be normal.

```
04/02-11:17:49.303666 217.96.21.210:51309 -> MY.NET.153.143:6346
TCP TTL:148 TOS:0x0 ID:37329  DF
21S***** Seq: 0xD0F8FFB7   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 618086 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
04/02-11:43:57.594927 24.232.140.16:40355 -> MY.NET.153.143:6346
TCP TTL:45 TOS:0x0 ID:49401  DF
21S***** Seq: 0x44AA2194   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 151839085 0 EOL EOL EOL EOL
```

Example of oos data to destination port 1214

The packets have all tcp bits set, which is an indication of packet craft. This could be a reconnaissance attempt.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
04/04-01:54:39.727093 142.51.44.123:1900 -> MY.NET.88.162:1214
TCP TTL:115 TOS:0x0 ID:10560  DF
*1SFRPAU Seq: 0x12CA55B   Ack: 0x33C0AC   Win: 0x5010
01 2C A5 5B 00 33 C0 AC 1E BF 50 10 22 38 65 D9  .,.[.3....P."8e.
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
04/04-02:04:34.504903 142.51.44.123:1900 -> MY.NET.88.162:1214
```

```
TCP TTL:115 TOS:0x0 ID:18889  DF
**SFR*AU Seq: 0x11012C   Ack: 0xA55BC191   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
```

Example of oos data to destination port 80

The packets all had the reserved bits and SYN flag set. The other characteristics of the packet appeared to be normal.

```
04/04-22:32:50.433282 202.153.244.62:46488 -> MY.NET.150.83:80
TCP TTL:44 TOS:0x0 ID:61500  DF
21S***** Seq: 0xFC4559FA   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 55224774 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
04/04-22:32:54.014058 202.153.244.62:46509 -> MY.NET.150.83:80
TCP TTL:44 TOS:0x0 ID:177  DF
21S***** Seq: 0xFC5BAB9F   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 55225134 0 EOL EOL EOL EOL
```

List of Top 10 Most Numerous Detects

Signature	# Alerts	# Sources	# Dests	Description

Connect to 515 from inside	537343	160	5	Port 515 is a common port that printer daemons listen on. This is a very normal event to see in networks that use LPR for printing. Having said that there is also some vulnerabilities connected to the LPRng daemon in some versions of linux systems. One of these vulnerabilities is the ramen worm. The ramen worm exploits the LPRng daemon vulnerability then will take over the machine and configure it to scan for other vulnerable machines. If you start to see a large number of this event it is highly likely that one of your systems is compromised. There only seems to be 5 destination addresses. There is some unusual traffic that will be explained in insights to internal machines below. Correlations: http://service2.symantec.com/SARC/sarc.nsf/html/Linux.Ramen.Worm.html http://www.sans.org/y2k/ramen.htm http://www.sans.org/y2k/practical/David_Singer_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description

SNMP Public access	90084	25	151	SNMP is a protocol used to remotely monitor and manage networking devices. This protocol is very useful for network administrators to perform their daily routines. Unfortunately SNMPv1 is susceptible to a variety of vulnerabilities. Specifically vulnerabilities were found in the way many SNMP managers decode and process SNMP trap messages. This event is triggered when the default community string is being used. (public) There is no traffic originating from the outside. All traffic is internal. 10.1.150.195 is the destination of most of this traffic which would indicate it might be some type of network management device. What is interesting is that traffic for this event stopped at 04/02 5:37 then started back up at 04/03 17:42. The ip address 10.1.150.195 should be investigated to see if it is some type of management device and why traffic stopped for that period of time. Correlations: http://securityresponse.symantec.com/avcenter/security/Content/2002.02.13.html http://www.cert.org/advisories/CA-2002-03.html

Signature	# Alerts	# Sources	# Dests	Description
SMB Name Wildcards	63467	282	287	This signature detects attempts to enumerate shares on a system running SMB file sharing (usually Microsoft Windows platforms). This can be a precursor to an attack, as it can potentially reveal a great deal about the target system, such as usernames and poorly protected shared resources. All this traffic has originated from the inside so it is likely to be normal traffic. It could be nbtstat lookupsor some browsing to 137. If this were to originate from the outside then you could speculate some type of reconnaissance. My.NET.11.6 is the top source address of this event. Correlations: http://www.sans.org/newlook/resources/IDFAQ/port_137.htm http://www.sans.org/y2k/practical/Teri_Bidwell_GCIA.doc http://www.giac.org/practical/Tamara_Bowman_GCIA.doc

Signature	# Alerts	# Sourc es	# Dests	Description

ICMP Echo Request L3retriever Ping	71696	163	14	This event indicates that someone may be scanning your network using the L3 "Retriever 1.5" security scanner. This legitimate security tool is for authorized security assessment and should not be used on unauthorized networks. If this scanner is used for unauthorized use the IP address would probably be crafted. There are known false positives with this signature. when Win2K computers are talking to domain controllers. In this case it looks like most of the traffic is destined to 10.1.11.7,10.1.11.6 and 10.1.11.5. These machines also generated lots of netbios traffic which would indicate they are NT machines probably domain controllers which would explain the number of alerts to these machines which seem to be false positives. Correlations: http://www.giac.org/practical/Edward_Peck_GCIA.doc www.giac.org/practical/Dennis_Ruck_GCIA.doc

Signature	# Alerts	# Sources	# Dests	Description
INFO MSN IM Chat data	19852	113	113	This signature detects data from the popular MSN program by Microsoft. This is an instant message program that will allow you to chat with someone using the same program. There has been numerous vulnerabilities with these types of programs. But the main issue is that these programs by pass the firewall rules making it hard for administrators to lock this traffic down. All external addresses look legitimate because they all seem to belong to a block of addresses by Hotmail which is a Microsoft company. Correlations: http://www.giac.org/practical/jeffrey_widom_GSEC.doc http://documents.iss.net/whitepapers/X-Force_P2P.pdf

Signature	# Alerts	# Sources	# Dests	Description
INFO Inbound GNUTella Connect request	16739	12485	13	This signature will detect inbound connection requests of the GNUTELLA program that is a popular peer-to-peer network program similar to the popular Napster but this program can share more than just mp3 files. This program lets you share any type of file. These programs can by pass the firewall rules allowing unauthorized data to pass through. It would be a good idea to block this sort of traffic at the firewall along with locking down your workstations not allowing the installation of any type of sharing program. The following internal ip addresses were involved with this alert. 10.1.153.211, 153.194, 153.175, 153.170, 153.164, 153.160, 153.143, 152.19, 152.185, 152.164 and 150.209. Correlations: http://securityresponse.symantec.com/avcenter/security/Content/2002.02.13.html http://www.cert.org/advisories/CA-2002-03.html

Signature	# Alerts	# Sources	# Dests	Description

High port 65535 udp - possible Red Worm - traffic	12169	215	147	This signature detects udp connections to port 65535, which at first glance is a good indication of some type of worm traffic. IA further look into this traffic indictates that most traffic both source and destination is internal. Typically worm traffic would not just spread to internal machines but would also spread to external IP addresses. Most of this traffic is with the .6 network. This would lead me to believe some other type of traffic causing these alerts. Also most of the .6 addresses also were found to have port scanned to udp port 7001. Machines connecting to port 7001 have been found to have a program called freak2k possibly installed. But this could also be a false positive as to some applications listen on port 7001 such as a program called weblogics. Please check these addresses for applications installed which would generate this type of traffic. 10.1.6.48 is the number one source of this signature. As well as the following: 10.1.6.49, 10.1.6.52, 10.1.6.50, 10.1.6.51, 10.1.6.53, 10.1.6.60, 10.1.6.45, 10.1.60.43. All these machines IP addresses should be investigated further. Most of these scans to port 7000 and 7001 are probably false positives but should still be investigated further. For proper security. Correlations: http://www.giac.org/practical/James_Conz_GCIA.doc http://www.**symantec**.com/avcenter/venc/data/codered.**worm**.html https://www.javaworld.com/javaworld/jw-0223-servletweblogic-p3.html

Signature	# Alerts	# Sources	# Dests	Description
MISC Large UDP Packet	9148	16	12	This event indicates that an abnormally large UDP packet was sent to your server. This may indicate a denial of service attack or the use of a covert channel. All the sources of this attack are external. Some these alerts look like false positives due to the sources are streaming music sites. But one address that is suspicious is 163.239.2.31. This machine is located in Korea and numerous connections were tried via 0 as the source port. The destination is 10.1.153.110 and should be looked at futher. Correlations: http://www.giac.org/practical/Victor_Maseda_GCIA.doc http://www.zeltser.com/sans/idic-practical/

DRAFT

Signature	# Alerts	# Sources	# Dests	Description

ICMP Echo Request Nmap or HPING2	5199	62	303	This signature detects an icmp request packet that is usually used by Nmap or Hping used to fingerprint a networking device. In this case most of the traffic is coming from .11 network specifically 11.6 and 11.7. This two ip addresses are probably windows machines do to the amount of port scans to udp port 137 and 139. There is also quite a few ldap (389) scans indicting a possible active directory installation. The scan source and ip addresses seem to match the source and destination ip addresses for this alert. In many windows networks I have seen pings being initiated by domain controllers periodically so it is highly likely these are false positives. Correlations: http://www.sans.org/y2k/practical/David_Singer_GCIA.doc http://www.giac.org/practical/John_McReynolds_GCFW.doc http://csrc.nist.gov/organizations/fissea/presentations/2000/internet-threat.ppt

Signature	# Alerts	# Sources	# Dests	Description
Watchlist 000220 IL-ISDNNET-990517	4288	18	14	This signature looks for IP addresses coming from Israel and China. Obviously all sources are external and from another country. This traffic is very suspicious due to a lot of connections trying to connect with a source port of 80. These 14 destinations should be investigated further. Correlations: http://ouah.sysdoor.net/George_Bakos.html http://www.giac.org/practical/Nelson_Carter_GCIA.doc

IV. Insight to Internal Machines

Above I have listed some insights to various internal machines. Below the list continues:

10.1.5.83 looks to be compromised with the subseven trojan. Should be investigated. This machine might have been compromised by 10.1.5.43 at about 04/02 12:44:30. Once compromised 10.1.5.83 started scanning for other machines to effect. 10.1.5.55 was one machine scanned and should also be investigated for compromise. There is also a lot of snmp public access events to 10.1.5.83 from 10.1.70.177.

```
04/02-12:44:30.610858   Possible trojan server activity   MY.NET.5.83:7938 ->
MY.NET.5.43:27374
04/02-12:44:30.610570   Possible trojan server activity   MY.NET.5.43:27374 ->
```

```
MY.NET.5.83:7938
04/02-12:44:30.610014   Possible trojan server activity   MY.NET.5.43:27374 ->
MY.NET.5.83:7938
04/02-12:44:30.609917   Possible trojan server activity   MY.NET.5.83:7938 ->
MY.NET.5.43:27374
04/02-12:44:30.609772   Possible trojan server activity   MY.NET.5.43:27374 ->
MY.NET.5.83:7938
```

```
04/04-07:52:40.513859   Possible trojan server activity   MY.NET.5.42:27374 ->
MY.NET.5.83:8139 MY.NET.5.42:
04/04-07:52:40.502086   Possible trojan server activity   MY.NET.5.42:27374 ->
MY.NET.5.83:8139 MY.NET.5.42:
04/04-07:52:40.480643   Possible trojan server activity   MY.NET.5.83:8139 ->
MY.NET.5.42:27374 MY.NET.5.83:
04/04-07:52:40.471388   Possible trojan server activity   MY.NET.5.42:27374 ->
MY.NET.5.83:8139 MY.NET.5.42:
04/04-07:52:40.470927   Possible trojan server activity   MY.NET.5.83:8139 ->
MY.NET.5.42:27374 MY.NET.5.83:
04/04-07:52:40.457888   Possible trojan server activity   MY.NET.5.42:27374 ->
MY.NET.5.83:8139 MY.NET.5.42:
04/04-07:52:40.457595   Possible trojan server activity   MY.NET.5.42:27374 ->
MY.NET.5.83:8139 MY.NET.5.42:
04/04-07:52:40.457528   Possible trojan server activity   MY.NET.5.83:8139 ->
MY.NET.5.42:27374 MY.NET.5.83:
04/04-07:52:40.457394   Possible trojan server activity   MY.NET.5.42:27374 ->
MY.NET.5.83:8139 MY.NET.5.42:
```

10.1.153.171 should be investigated. This machine is the source of a lot of traffic with a destination port of 515. Traffic to port 515 is usually normal traffic but because the time between events it is a good possibility that this machine is infected with a worm. The traffic to 515 seem to stop after 04/03. There was a lot of possible red worm high port events as well as ftp globbing. This machine should definitely be investigated for compromise.

```
04/02-08:22:16.905457   connect to 515 from inside   MY.NET.153.171:2651 ->
MY.NET.150.198:515
04/02-08:22:16.905377   connect to 515 from inside   MY.NET.153.171:2651 ->
MY.NET.150.198:515
04/02-08:22:16.904166   connect to 515 from inside   MY.NET.153.171:2651 ->
MY.NET.150.198:515
04/02-08:22:16.904095   connect to 515 from inside   MY.NET.153.171:2651 ->
MY.NET.150.198:515
04/02-08:22:16.904023   connect to 515 from inside   MY.NET.153.171:2651 ->
```

```
MY.NET.150.198:515
04/02-08:22:16.903956   connect to 515 from inside  MY.NET.153.171:2651 ->
MY.NET.150.198:515
04/02-08:22:16.903199   connect to 515 from inside  MY.NET.153.171:2651 ->
MY.NET.150.198:515
04/02-08:22:16.903133   connect to 515 from inside  MY.NET.153.171:2651 ->
MY.NET.150.198:515

04/05-15:32:24.798920   High port 65535 udp - possible Red Worm - traffic
MY.NET.6.48:61695 -> MY.NET.153.171:65535
04/05-09:57:41.804701   High port 65535 udp - possible Red Worm - traffic
MY.NET.6.52:65535 -> MY.NET.153.171:65535
04/04-21:05:54.124116   High port 65535 udp - possible Red Worm - traffic
MY.NET.6.52:65535 -> MY.NET.153.171:65535

04/05-18:10:40.759516    FTP DoS ftpd globbing   155.69.8.243:1293 -> MY.NET.15
.171:21  155.69.8.243:  -> MY.NET.153.171:      :1293 - :21
04/05-18:12:13.586929    FTP DoS ftpd globbing   155.69.8.243:1293 -> MY.NET.15
.171:21  155.69.8.243:  -> MY.NET.153.171:      :1293 - :21
04/05-18:11:51.688339    FTP DoS ftpd globbing   155.69.8.243:1293 -> MY.NET.15
.171:21  155.69.8.243:  -> MY.NET.153.171:      :1293 - :21
04/05-18:10:52.974823    FTP DoS ftpd globbing   155.69.8.243:1293 -> MY.NET.15
.171:21  155.69.8.243:  -> MY.NET.153.171:      :1293 - :21
04/05-18:12:38.890156    FTP DoS ftpd globbing   155.69.8.243:1293 -> MY.NET.15
.171:21  155.69.8.243:  -> MY.NET.153.171:      :1293 - :21
```

10.1.88.162 looks like it has been compromised as well since it is the source of the Backdoor Metro signature. (port 5032 = Netmetro backdoor)

```
04/06-19:49:32.727980   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:49:21.455120   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:49:15.856500   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:49:12.735182   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:49:07.472905   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:48:59.206884   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:47:55.738679   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
04/06-19:47:53.068279   BACKDOOR NetMetro File List  MY.NET.88.162:1214 ->
68.47.111.45:5032
```

10.1.6.50 / 10.1.6.49 / 10.1.6.48 These ip addresses which are probably AFS servers (IBM) can possibly be compromised as they are sources of the Back orifice events. Further investigation is necessary.

10.1.153.46 should be investigated. It has connections from outside via TFTP. It is also involved with some Trojan activity as well as port scanning.

10.1.153.45 should also be investigated due to TFTP connections. This IP address also was the source of many port scans and Large UDP Packets/MSN Chat Data.

```
04/06-13:32:32.598375    TFTP - Internal UDP connection to external tftp server
        160.79.2.66:69 -> MY.NET.152.19:51836   160.79.2.66:   -> MY.NET.152.19
:      :69 -   :51836
04/05-17:10:54.201645    TFTP - Internal UDP connection to external tftp server
        64.124.157.10:69 -> MY.NET.153.45:71    64.124.157.10: -> MY.NET.153.45
:      :69 -   :71
04/03-13:30:54.909734    TFTP - External UDP connection to internal tftp server
        63.250.219.189:16495 -> MY.NET.153.45:69       63.250.219.189:
-> MY.NET.153.45:       :16495 -        :69
04/03-11:21:28.527684    TFTP - External UDP connection to internal tftp server
        63.250.205.36:256 -> MY.NET.153.46:69   63.250.205.36: -> MY.NET.153.46
:      :256 -  :69
04/02-11:41:05.826050    TFTP - External UDP connection to internal tftp server
        63.250.219.189:0 -> MY.NET.153.46:69    63.250.219.189:        -> MY.NE
T.153.46:      :0 -    :69
```

V. External Sources

These five external IP addresses were chosen because they were the top 5 external talkers.

1. 216.106.173.150

This IP address was the top talker for the Misc_Large_UDP_Packets event. After looking up the address is looks like these were false positives because this address does streaming media. Streaming has a tendency to send large udp packets.

From ARIN Who is Server: iBEAM Broadcasting Corporation (NETBLK-IBEAM)
645 Almanor Ave., suite 100
Sunnyvale, CA 94085
US

Netname: IBEAM

```
 Netblock: 216.106.160.0 - 216.106.175.255
 Maintainer: BEAM

 Coordinator:
   Le, Stewart  (SL895-ARIN)  stle@ibeam.com
   408-830-3572

 Domain System inverse mapping provided by:

 NS1.IBEAM.COM                    204.233.70.15
 NS2.IBEAM.COM                    204.247.99.125

 ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

 Record last updated on 22-Jan-2002.
 Database last updated on  1-Jul-2002 20:10:52 EDT.
```

```
From Amnesia Who is Server: iBEAM Broadcasting Corporation (NETBLK-IBEAM)
 645 Almanor Ave., suite 100
 Sunnyvale, CA 94085
 US

 Netname: IBEAM
 Netblock: 216.106.160.0 - 216.106.175.255
 Maintainer: BEAM

 Coordinator:
   Le, Stewart  (SL895-ARIN)  stle@ibeam.com
   408-830-3572

 Domain System inverse mapping provided by:

 NS1.IBEAM.COM                    204.233.70.15
 NS2.IBEAM.COM                    204.247.99.125

 ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

 Record last updated on 22-Jan-2002.
 Database last updated on  1-Jul-2002 20:10:52 EDT.
```

2. 210.94.0.146

This IP Address was the second largest Top Talker of Misc_Large_UDP_Packets.

Unlike 216.106.173.150 this IP originated in Korea. These events from this IP should be investigated.

```
From ARIN Who is Server:  Asia Pacific Network Information Center
(NETBLK-APNIC-CIDR-BLK)
  APNIC
  AU

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

  Coordinator:
    Administrator, System  (SA90-ARIN)  [No mailbox]
    +61 7 3858 3100

  Domain System inverse mapping provided by:

  NS.APNIC.NET                          203.37.255.97
  SVC00.APNIC.NET                       202.12.28.131
  NS.TELSTRA.NET                        203.50.0.137
  NS.RIPE.NET                           193.0.0.193

  Asia Pacific Network Information Centre
   This IP address range is not registered in the ARIN database.
   For details, refer to the APNIC Whois Database via
   WHOIS.APNIC.NET or http://www.apnic.net/apnic-bin/whois2.pl
  IMPORTANT NOTE: APNIC is the Regional Internet Registry for
   the Asia Pacific region. APNIC does not operate networks using
   this IP address range and is not able to investigate spam
   or abuse reports relating to these addresses. For more help,
   refer to http://www.apnic.net/info/faq/abuse/

  Record last updated on 03-May-2000.
  Database last updated on  3-Jul-2002 20:04:49 EDT.

      From APNIC Who is Server:  inetnum:     210.92.0.0 -
      210.95.255.255
      netname:     KRNIC-KR
      descr:      KRNIC
      descr:      Korea Network Information Center
      country:     KR
      admin-c:     HM127-AP
      tech-c:     HM127-AP
      remarks:
      ******************************************
```

```
remarks:    KRNIC is the National Internet Registry
remarks:    in Korea under APNIC. If you would like
to
remarks:    find assignment information in detail
remarks:    please refer to the KRNIC Whois DB
remarks:    http://whois.nic.or.kr/english/index.html
remarks:
****************************************
mnt-by:     APNIC-HM
mnt-lower:  MNT-KRNIC-AP
changed:    hostmaster@apnic.net 19981001
changed:    hostmaster@apnic.net 20010606
source:     APNIC

person:     Host Master
address:    11F, KTF B/D, 1321-11, Seocho2-Dong, Seocho-
Gu,
address:    Seoul, Korea, 137-857
country:    KR
phone:      +82-2-2186-4500
fax-no:     +82-2-2186-4496
e-mail:     hostmaster@nic.or.kr
nic-hdl:    HM127-AP
mnt-by:     MNT-KRNIC-AP
changed:    hostmaster@nic.or.kr 20020507
source:     APNIC

inetnum:    210.94.0.0 - 210.94.2.255
netname:    HANANET-KR
descr:      HANARO Telecom
descr:      1445-3 Seocho-Dong Seocho-Ku
descr:      SEOUL
descr:      137-728
country:    KR
admin-c:    IS48-KR
tech-c:     SH270-KR
remarks:    This IP address space has been allocated to
KRNIC.
remarks:    For more information, using KRNIC Whois
Database
remarks:    whois -h whois.nic.or.kr
remarks:    This information has been partially mirrored by
APNIC from
remarks:    KRNIC. To obtain more specific information,
please use the
```

```
remarks:     KRNIC whois server at whois.krnic.net.
mnt-by:      MNT-KRNIC-AP
changed:     hostmaster@nic.or.kr 20020701
source:      KRNIC

person:      Inyup Sung
country:     KR
phone:       +82-2-106
fax-no:      +82-2-6266-6483
e-mail:      info@hananet.net
nic-hdl:     IS48-KR
remarks:     This information has been partially mirrored by
APNIC from
remarks:     KRNIC. To obtain more specific information,
please use the
remarks:     KRNIC whois server at whois.krnic.net.
mnt-by:      MNT-KRNIC-AP
changed:     hostmaster@nic.or.kr 20020701
source:      KRNIC
```

3. 163.239.2.31

This IP address is the Third Top Talker for the event Misc_Large_UDP_Packets. Once again the IP address is originating from Korea.. Though the server appears to be a streaming service there is a lot of connections with source port and destination port being 0. This is suspicious traffic and should be investigated.

```
From Who is Server:
Sogang University (NET-SOGANG-NET)
  Seoul
  KR

  Netname: SOGANG-NET
  Netblock: 163.239.0.0 - 163.239.255.255

  Coordinator:
    Villarreal, Felix M.  (FMV3-ARIN)  [No mailbox]
    (82)(2) 705-8492

  Domain System inverse mapping provided by:

  CCS.SOGANG.AC.KR                   163.239.1.1
  NS.HANA.NM.KR                      203.232.127.1

  Record last updated on 08-Oct-1992.
  Database last updated on  3-Jul-2002 20:04:49 EDT.
```

From Amnesi Who is Server: Sogang University (NET-SOGANG-NET)
Seoul
KR

Netname: SOGANG-NET
Netblock: 163.239.0.0 - 163.239.255.255

Coordinator:
Villarreal, Felix M. (FMV3-ARIN) [No mailbox]
(82)(2) 705-8492

Domain System inverse mapping provided by:

CCS.SOGANG.AC.KR	163.239.1.1
NS.HANA.NM.KR	203.232.127.1

Record last updated on 08-Oct-1992.
Database last updated on 3-Jul-2002 20:04:49 EDT.

4. 212.179.40.132

This IP address was a Top Talker for the Watchlist-000220IL-ISDNNET-990517 event. Obviously this IP originated from Israel since that is what the Watchlist signature looks for. What is interesting is that all connections have a destination port of 4662, which is a standard port for the edonkey2000 file-sharing program similar to a Kazaa or Napster. This traffic is should be investigated further.

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

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

Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN) nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

```
NS.RIPE.NET                193.0.0.193
AUTH03.NS.UU.NET           198.6.1.83
NS2.NIC.FR                 192.93.0.4
SUNIC.SUNET.SE             192.36.125.2
MUNNARI.OZ.AU              128.250.1.21
NS.APNIC.NET               203.37.255.97

To search on arbitrary strings, see the Database page on
the RIPE NCC website at http://www.ripe.net/perl/whois/

Record last updated on 16-Oct-1998.
Database last updated on  3-Jul-2002 20:04:49 EDT.
```

From Amenesi Who is Server:

```
inetnum         212.179.40.128 - 212.179.40.255
Origin          KIBBUTZ-GADOT
descr           KIBBUTZ-GADOT-LAN
country         IL
Admin. Contact  ZV140-RIPE
Tech. Contact   NP469-RIPE
status          ASSIGNED PA
Notify          hostmaster@isdn.net.il
mnt-by          RIPE-NCC-NONE-MNT
changed         hostmaster@isdn.net.il 20001015
source          RIPE
route           212.179.0.0/18
descr           ISDN Net Ltd.
Origin          AS8551
Notify          hostmaster@bezeqint.net
mnt-by          AS8551-MNT
changed         hostmaster@bezeqint.net 20020618
source          RIPE
person          Zehavit Vigder
address         bezeq-international
address         40 hashacham
address         petach tikva 49170 Israel
phone           +972 52 770145
fax-no          +972 9 8940763
e-mail          hostmaster@bezeqint.net
NIC Handle      ZV140-RIPE
changed         zehavitv@bezeqint.net 20000528
source          RIPE
person          Nati Pinko
```

address	Bezeq International
address	40 Hashacham St.
address	Petach Tikvah Israel
phone	+972 3 9257761
e-mail	hostmaster@isdn.net.il
NIC Handle	NP469-RIPE
changed	registrar@ns.il 19990902
source	RIPE

5. 63.240.15.207

This IP address was the 4th Top Talker in the Misc_Large_UDP_Packets event. This IP is registered with AT&T. What is interesting about this traffic is that all source port and destination ports are identical. The destination port is 2109 which is registered to Ergolight. Eroglight is a type of software producing website diagnostic reports. This could be legitimate traffic. The destination IP address 10.1.153.171 should be investigated to insure the use of this program or something similar.

From ARIN Who is Server: AT&T CERFnet (NETBLK-CERFNET-BLK-5)
P.O. Box 919014
San Diego, CA 92191
US

Netname: CERFNET-BLK-5
Netblock: 63.240.0.0 - 63.242.255.255
Maintainer: CERF

Coordinator:
AT&T Enhanced Network Services (CERF-HM-ARIN)
notify@attens.com
(858) 812-5000

Domain System inverse mapping provided by:

DBRU.BR.NS.ELS-GMS.ATT.NET	199.191.128.106
CBRU.BR.NS.ELS-GMS.ATT.NET	199.191.128.105
DMTU.MT.NS.ELS-GMS.ATT.NET	12.127.16.70
CMTU.MT.NS.ELS-GMS.ATT.NET	12.127.16.69

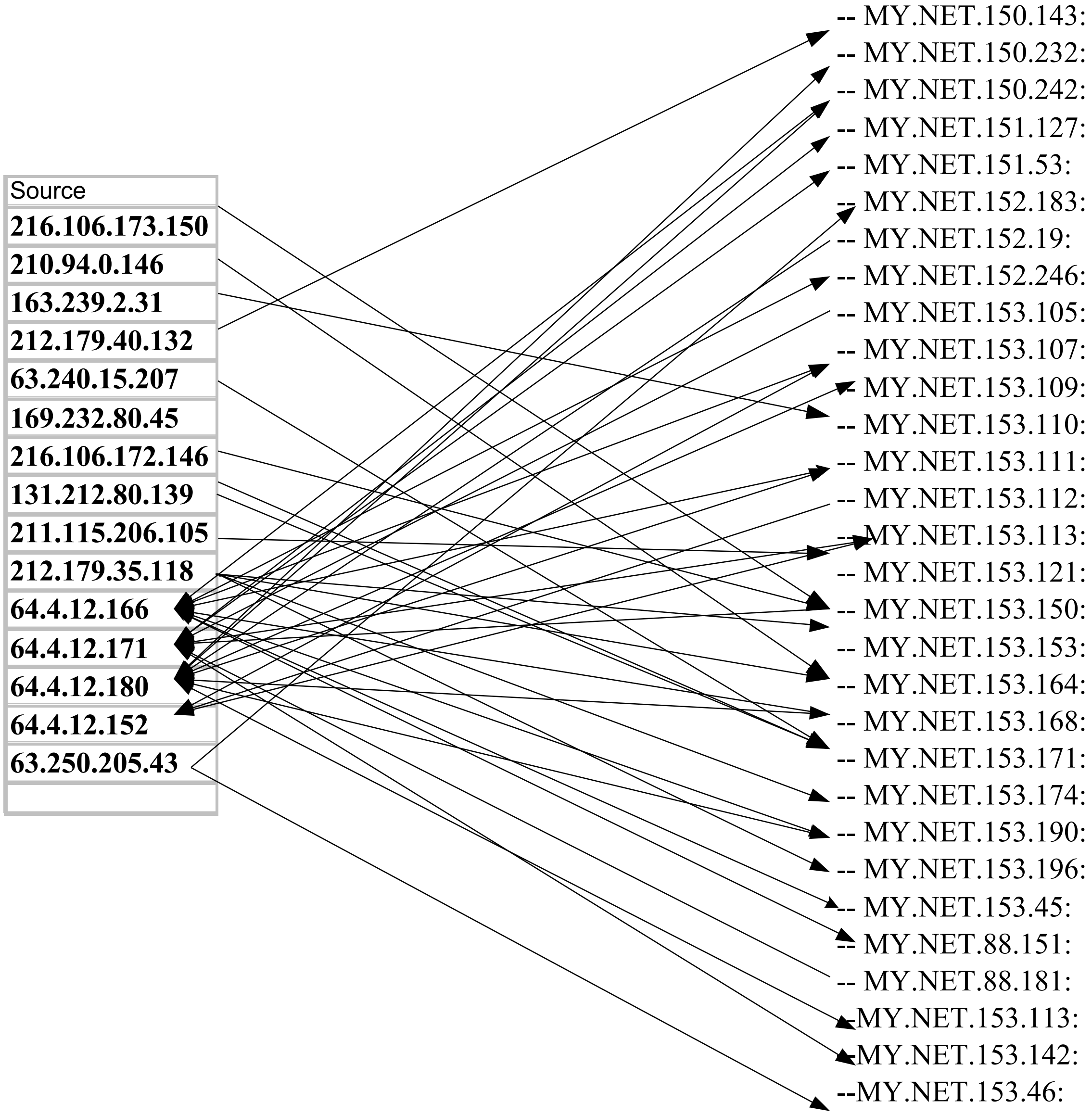
ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 06-Aug-2001.
Database last updated on 3-Jul-2002 20:04:49 EDT.

```
From Amenesi Who is Server:  AT&T CERFnet (NETBLK-CERFNET-BLK-5)
  P.O. Box 919014
  San Diego, CA  92191
  US

  Netname: CERFNET-BLK-5
  Netblock: 63.240.0.0 - 63.242.255.255
  Maintainer: CERF

  Coordinator:
    AT&T Enhanced Network Services  (CERF-HM-ARIN)  notify@attens.com
    (858) 812-5000

  Domain System inverse mapping provided by:

  DBRU.BR.NS.ELS-GMS.ATT.NET199.191.128.106
  CBRU.BR.NS.ELS-GMS.ATT.NET199.191.128.105
  DMTU.MT.NS.ELS-GMS.ATT.NET            12.127.16.70
  CMTU.MT.NS.ELS-GMS.ATT.NET            12.127.16.69

  ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

  Record last updated on 06-Aug-2001.
  Database last updated on  3-Jul-2002 20:04:49 EDT.
```

Link Graph of Top 15 External IP addresses

Most of the traffic to external ip addresses were to the network 64.4.12.0 which is registered to hotmail.com. This is probably massager traffic. (MSN)



VI. Defensive Recommendations

Currently there does not seem to be much security in place.

The first thing to do is react to the systems that are compromised. Ensure these machines are cleaned and hardened following the security best practices guidelines. Each operating system has a security checklist that can give you insight to a solid security baseline. Along with cleaning compromised systems you should patch all systems to current revision levels. This will help cut down future compromises while you plan your network security design. Once you have completed the reaction step you should start developing a security policy. This will give you a foundation to build your security design. Some things to consider for your security design are firewalls blocking the perimeter, intrusion detection placed strategically, log collection and secure builds for your servers and workstations. These are just a few of the many things to consider when designing a policy. It might help to hire a security consultant to provide guidance along the way. When configuring your firewall make sure you are only allowing services in and out that are absolutely necessary. Another thing to consider is informing your user community about the importance of security. The best technology in the world can be circumvented by human error.

Some other more specific recommendations are noted below.

Snmp is a very useful tool for managing your network however it is very exploitable. If you are not using snmp for managing your network you should turn off this service. If you are using it you might want to use version 2 where you have more security options. Changing default strings is also imperative.

There was a lot of chat and peer to peer traffic on your network. These programs susceptible to vulnerability and if they are not required they should be blocked at a firewall and you should lock down the workstations to allow nothing to be installed without administrative privileges. This would also take care of all the scanning tools and remote control applications being run on your network as well. Implementing this should begin by defining a acceptable use policy for your company.

.

VII. Analysis

The alerts log was initially started with snorfsnarf but due to the amount of data the perl script was not able to complete due to memory issues. I then tried again on a linux machine with 1 gigabyte of ram and snortsnarf still failed. I then decided to brake up the data by event. I first aggregated all 5 files to one file then replaces MY.NET with 10.1.
The following commands were executed to accomplish this:
Cat alert1 alert2 alert3 alert4 alert5 >Alerts, sed 's/MY.NET/10.1/g' Alerts.10
I then grep'd out individual events just to get a feel for snortsnarf.
Cat Alerts.10 |grep "event name"
After a saw what snortsnarf produced I then took that text file (Alerts.10) and did a bulk insert to SQL creating fields for each record. After successfully inserting data, I parsed the data into separate fields for source IP, source Port, destination IP and destination Port.

VIII. References

Snort - The Open Source Network IDS
URL: http://www.snort.org

Asia Pacific Network Information Centre

URL: http://www.apnic.net

American Registry for Internet Numbers
URL: http://www.arin.net

SANS Institute Resources, Global Incident Analysis Center, Detects Analyzed 9/1/00
URL: http://www.sans.org/y2k/090100.htm

MusicCity Morpheus
URL: http://www.musiccity.com

WhiteHats WebSite
URL: http://www.whitehats.com

Amnesi Who is Server
URL: http://www.amnesi.com

George Bakos Giac Practical
URL: http://ouah.sysdoor.net/George_Bakos.html

- All other references are located throughout the paper.