Global Information Assurance Certification Paper

GIAC Certified Intrusion Analyst

Practical Assignment

Version 3.3

Submitted by:

Denis E. Brooker, GCFA, CISSP

April 16, 2003

Abstract

This is the GIAC Certified Intrusion Analysis (GCIA) certification paper submitted by Denis E. Brooker to GIAC for consideration as required for the GCIA Certification. The paper was written under the guidelines for the GIAC Certified Intrusion Analyst Practical Assignment Version 3.3 and consists of three parts as required by the assignment.

Part 1 meets the requirements for "Describe the State of Intrusion Detection" as required by the assignment. This section, entitled "Intrusion Prevention Systems" looks as the emerging technology of Intrusion Prevention Systems as they relate to Intrusion Detection Systems. The paper discusses the concepts of Intrusion Detection and Prevention and looks at three specific flaws that affect the functionality of Intrusion Prevention Systems. This section of the document is 9 pages long and has 12 references.

Part 2 of the assignment meets the requirements for "Network Detects" and includes three separate sets of alerts with detailed analysis of each. Detects one and two were downloaded from the logs at http://www.incidents.org/logs/Raw as required by the assignment. Detect three was taken from a private network of a Fortune 500 company. One detect contains comments received from a user of Incidents.Org as required in the assignment.

Part 3 of the paper is the "Analyze This" section written to meet the requirements outlined in the assignment. It is a scenario based security audit for a University based upon log files downloaded from the incidents.org website. It includes all of the requirements of the assignment.

This paper was written to clearly demonstrate mastery of the course material and to help improve the state of practice of information security and is hereby submitted to GIAC for approval.

Table of Contents

Assignment Part 1 - Describe the State of Intrusion Detection

Intrusion Prevention Systems

Abstract

Many of the commercial products in the Intrusion Detection genre are touting the concept of "Intrusion Prevention" as a superior alternative to plain "Intrusion Detection". The major difference seems to be the capability of Intrusion Prevention to take proactive actions to safeguard protected systems. This paper will explore the issues involved in intrusion prevention, specifically flaws in the development and deployment of the systems and how these flaws may be addressed.

The Intrusion Prevention Concept and the Flaws

The first thing that must be covered is a definition of Intrusion Prevention Systems. The actual definition is, apparently, a problem in the Information Security field. According to Andy Briney, "The point is that if 'intrusion prevention' can refer to everything, it can't mean anything-that is, it can't mean any one thing."[1] Mr. Briney is referring to the over-usage of the term Intrusion Prevention. There are many tools that are available as free software or commercially packaged products that are designed to prevent intrusions. Anti-virus systems are one example of this as they prevent viruses from attacking network systems. Vulnerability assessment tools may be considered as intrusion prevention systems. The scope of this white paper will be limited to either Host Based or Network Based Intrusion Prevention Systems that monitor activity in the same manner as Intrusion Detection Systems, but have the additional capability to take action to mitigate or eliminate detected threats. Further references to Intrusion Prevention Systems after this point will be made from within this scope.

The concept behind intrusion prevention is to simply react to hostile traffic in a proactive manner that will prevent that traffic from harming protected systems and to ensure that confidentiality, integrity, and accessibility are maintained. While the concept is fairly simple to describe, it is unfortunately much more difficult to implement.

Intrusion prevention is based upon solid intrusion detection. It is, therefore, necessary that the reader understand intrusion detection before proceeding. Intrusion detection is the process of monitoring traffic and activities to determine when an attack is taking place. There are two basic types of Intrusion Detection Systems (IDS); Network based (NIDS) and Host based (HBIDS).

---

[1] Briney, Andy. "What Isn't Intrusion Prevention" April 2002, URL: http://www.infosecuritymag.com/2002/apr/note.shtml (February 23, 2003)

NIDS is comprised of network sensors that monitor all traffic traversing the network. They are operating in what is known as "promiscuous" mode, meaning they are able to read all traffic regardless of whether or not it was addressed to that interface.

HBIDS is software that actually runs on the hosts on the network. Instead of monitoring traffic traversing the network, HBIDS monitors files, process, memory, log files, or other internal system functions for activity that matches a signature.

Developing a "signature" of malicious traffic, used in both NIDS and HBIDS, and then using that signature to detect activity and traffic that match it is the most commonly used method to accomplish the detection.

The other method of detecting attacks is termed "anomaly based" intrusion detection. In this method, a baseline of activity over a period of time is developed. The longer the time period for baseline development, the more accurate the system will be. Once the baseline is determined, then the system alerts on any traffic or activity that does not fall within the parameters of the baseline.

There are many weaknesses and flaws in the two detection process methods that will directly affect using the system actively to prevent intrusions. The three primary weaknesses will be discussed in this paper. The first is the problem of false positives, the second is the threat of the system being used against the network it is intended to protect, and the third is the problem of false negatives.

Flaw 1 – False Positives

The first and foremost flaw in signature based intrusion detection systems is the problem of false positives. False positives, attack alerts without an attack, occur when a non-malicious or normal packet matches the signature. This causes an alert to be generated and in the case of intrusion prevention, it causes the system to take an action that would be inappropriate for the situation. It is a foregone conclusion that false positives will occasionally happen due to the complexities of Internet Protocol (IP) traffic and the lack of sufficient checks and balances. In order to demonstrate how easy it is to achieve a false positive let us look at the following signature and break it down to see exactly what causes a match.

Alert tcp any any -> any 25 (msg: "Virus – Possible QAZ Worm Calling Home"; content : "nongmin_cn"; reference: MCAFEE, 98775; sid: 733; classtype:misc-activity; rev:3;)

This is a rule for the Snort Intrusion Detection system that was downloaded from the Snort website.[2] This signature is for QAZ Worm making a connection using port 25. If you look at the circled section of the signature, you can discern exactly what has to match for an alert to be generated. The section that says

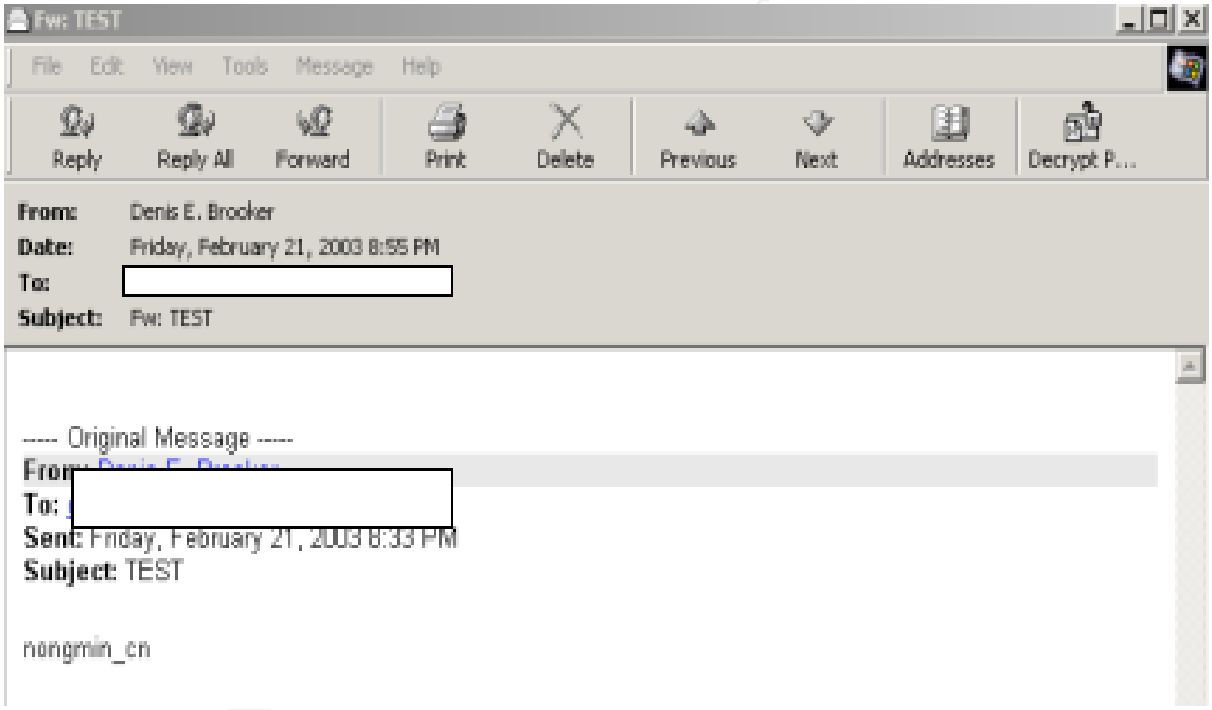
[2] Snort Signature Database, SID 733, Virus. Rules, Snort Signature Database, http://www.snort.org/cgi-bin/needed.cgi?offset (February 23, 2003)

“content: nongmin_cn” states that the packet must have this exact phrase located anywhere in the content area of the packet. If this phrase is in the content area of any message where the destination port is 25, regardless of what the packet is really used for, the alert will be generated.

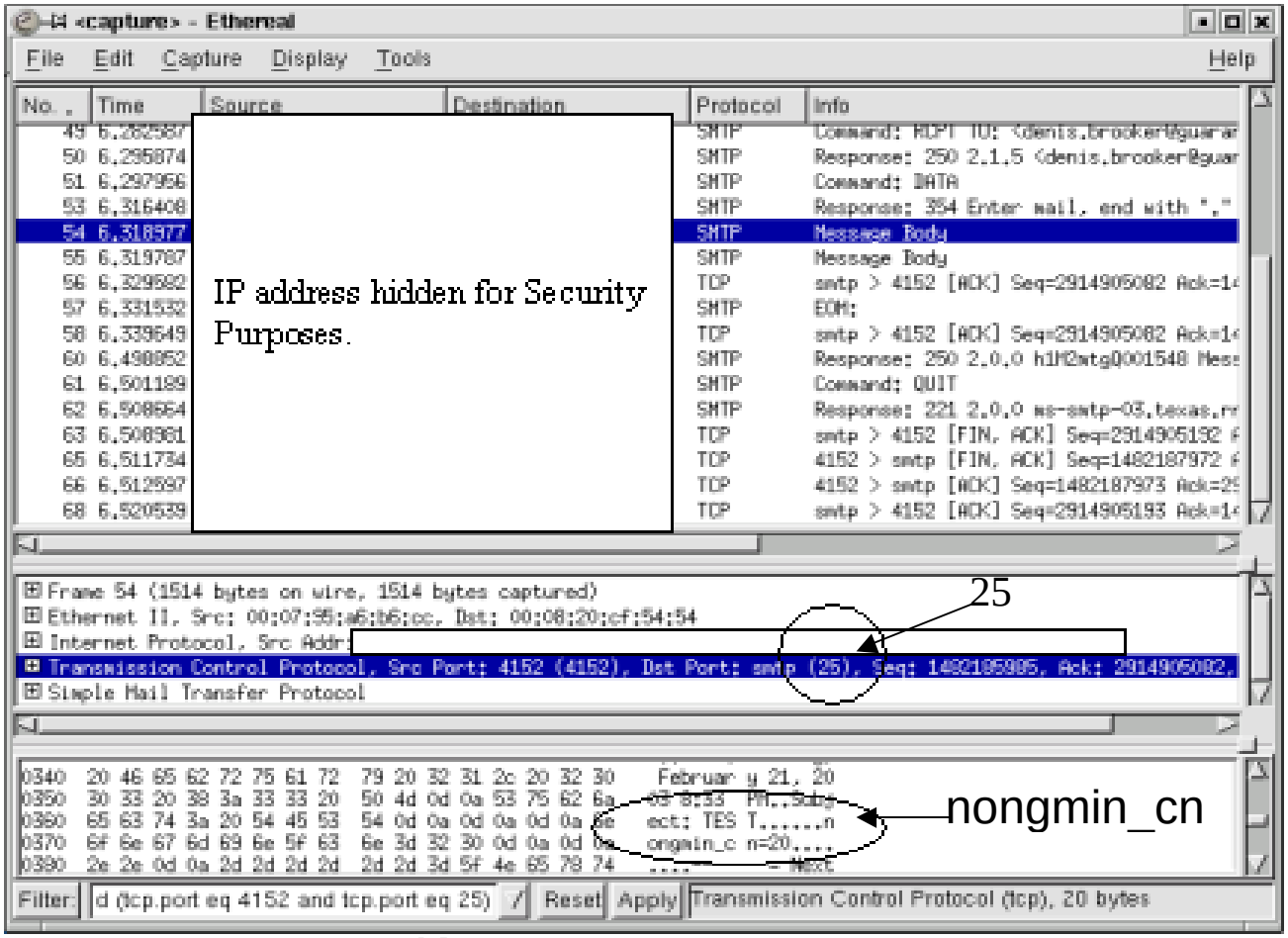
Now, let’s look at innocent traffic that would cause a match to this signature and a resultant alert.

In this case, a simple email (recipient has been blocked from view for privacy purposes) was constructed and sent with the “nongmin_cn” statement in the body of the text. Since email uses the SMTP port, 25, and the required text is in the content area, it should cause an alert to be generated.

It should be noted at this point that, while this is a setup demonstration, it takes place on an active production network. The email shown above was real and actually sent between two accounts across the Internet.

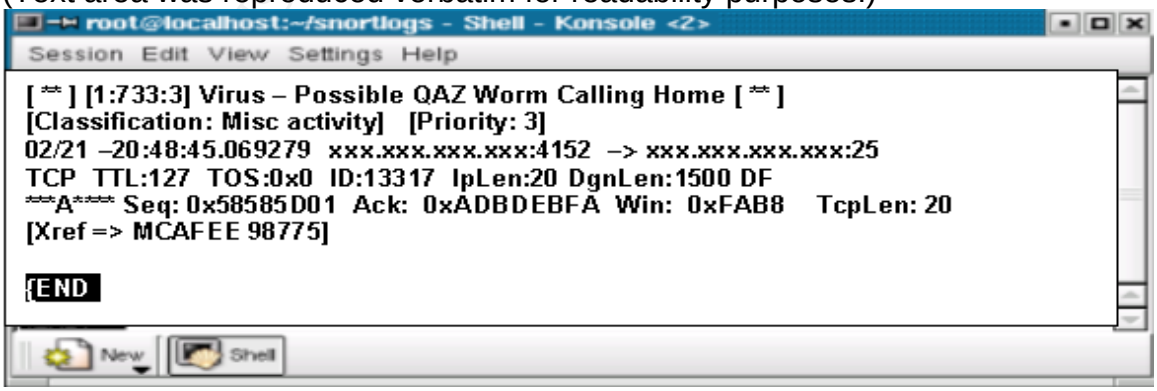


While you may think the likelihood of something like this happening in the real world is very small, it actually occurs very frequently right after a new security flaw or virus, and resultant signature has been released. Many different security vendors will send out mass emails warning of the problem, which will many times cause alerts as the technical details of the problem at hand are discussed. In our example above, an actual email may describe this virus and let the recipient know that “nongmin_cn” would be somewhere in the virus text. Since it is an email and is sent to destination port 25 it will, like the demonstration message above or the actual virus, trigger an alert. On a system with numerous rules to be deployed, it is extremely difficult to foresee all of the situations that may occur to cause a false positive.



This screenshot shows the packets that were generated when the email was sent and is used as further proof of the preceding test. This particular traffic is being displayed by the Windows version of Ethereal.[3]

Note in the middle frame circled area that the destination port was 25 and note in the lower frame circled area the "nongmin_cn" text. The parameters required for the alert were both met and the alert was generated as shown below. (Text area was reproduced verbatim for readability purposes.)



[3] Ethereal Network Protocol Analyzer, URL: http://www.ethereal.com/ (February 25, 2003)

At the writing of this document, there were over 1700 default Snort rules available on the Snort website.[4] If even a small portion of these cause false positives on a fairly active network, you can see the sheer volume of false positives could be overwhelming.

The problem of false positives is one that is well known. In her Network World article, Joanne Cummings stated, "Intrusion-prevention vendors have to find a way to eliminate false positives,"[5] referring to the topic being discussed. She brings up yet another false positive problem when she points out: "False positives are thorns in the sides of so many traditional IDSs because, if improperly configured, they will register attacks as legitimate even if those attacks have no bearing on the network. For example, an IDS on a network of Apache web servers must be told not to register attacks to Microsoft Internet Information Server, otherwise it will issue an alarm when it sees an IIS attack".[6]

False positives can cause great consternation from a Network Security perspective from three different aspects. First, excessive false positives can bog down an Intrusion Detection System (IDS), taking up limited resources such as memory, processor utilization, and even hard-drive storage space. Second, excessive false positives can obfuscate authentic positives rendering all alerts ineffective. Third, excessive false positives desensitize the network security staff prompting them to ignore alerts because they see so many of them. False positives are a big problem when it comes to Intrusion Detection, but are a "deal stopper" when it comes to intrusion prevention.

Intrusion Prevention Systems take the detection process to the next logical step; they take action to prevent the intrusion from occurring. Methods used to accomplish this task may include having the system reset the access control lists on the firewall to block all traffic from the "attacking" source, send TCP resets to both source and destination IP addresses, and/or reconfigure the web server to reject the hostile traffic. Therefore, false positives could cause a system to inappropriately block legitimate traffic, even traffic from your best customers.

"The Review is using the product to block only a modest portion of known attacks because of concern about dropping legitimate traffic for the web sites the publication manages."[7] Ellen Messmer concisely and precisely sums up the problems of false positives as they relate to intrusion prevention in her Computerworld article. The problem then, having been adequately defined, requires a solution or at least mitigation.

---

[4] Snort Signature Database, Snort Signature Database, http://www.snort.org/cgi-bin/needed.cgi?offset (February 23, 2003)

[5] Joan Cummings, From Intrusion Detection to Intrusion Prevention A New Breed of Security Tools for Stopping Intruders Shows Promise, But Can't Be Entirely Trusted Yet, Network World, 9/23/2002 , Cummings, Joanne. "Intrusion detection to Intrusion prevention." Network World. Volume 19, No.38 (2002): 72-82. (May also be found at URL: http://www.nwfusion.com/buzz/2002/intruder.html (February 23, 2003))

[6] Ibid

[7] Messmer, Ellen. "Intrusion prevention systems raise hopes, concerns", URL: http://www.computerworld.com/securitytopics/security/story/0,10801,75630,00.html

Overcoming False Positives

Unfortunately, the problem of false positives is not one that is easy to overcome. Certainly, it is not possible to overcome the problem with a "Turn-Key" system as promised by some of the Intrusion Prevention vendors.

In order to overcome the problem of false positives, the Intrusion Detection/Prevention System must be tuned to exactly meet the requirements of the system(s) and network that it is protecting and in the environment where it resides. Again, this is very easy to say, but a bit more difficult to accomplish.

When tuning the system, the security engineer must have an in-depth knowledge of how the network operates. A baseline of operations must be well established and understood. Signatures can then be adjusted to address only traffic that could possibly affect the network.

Another tuning technique is to accurately identify the home network to the detection system. This will prevent false positives from traffic that could not possibly be the source of an attack.

Next, the system must be thoroughly tested in the detection only mode in order to determine what false positives are likely to be received. It would be a major error, even after extensive tuning, to immediately have the system take proactive action in response to alerts. This would certainly lead to an unintended restriction of traffic.

Overcoming false positives is time-consuming, but is very important to any Intrusion Detection System. It is absolutely critical to an Intrusion Prevention System.

Flaw 2 – Use Against the Protected Network

The second major flaw with intrusion protection systems is the possibility that they could be used against the network by a pre-meditated attack designed to cause a Denial of Service (DoS). If hostile parties were to discover an active Intrusion Protection System on a network, they could very easily use the capabilities of the system to systematically lock out legitimate users. Let's further explore this possibility by a fictional scenario.

> An Intrusion Prevention System has been installed on a network that has the enabled ability to proactively change the ACL's on the main firewall to disallow traffic from sources it sees as hostile. A hostile packet comes in that attempts to perform a Buffer Overflow on a web site. The system recognizes this by a matching signature pattern and changes the firewall ACL so that the firewall drops all packets from IP address 192.x.z.34 (this address is for discussion purposes only).
>
> The attacker (he/she) at the source IP address recognizes that his attack has been thwarted and wonders what happened. He attempts a port scan, duplicating one that had been successful earlier, and determines that he is being stopped at the border firewall. He then uses a different IP address,

perhaps from a system that he has already compromised and sends another attack only to be foiled again by the Intrusion Prevention system. At this point, he becomes suspicious that the system is being protected by an automated system due to the speed that he is being blocked. He realizes that his only chance at compromising this system is to get the Intrusion Prevention System turned off.

He sets up several "zombies" that he controls to send spoofed IP packets to the victim network. These zombies will send packets to the victim network that have the same characteristics of the Buffer Overflow that had caused the Intrusion Protection system to react. The systems are also setup to run through a series of IP addresses from different major ISP's including AOL and Earthlink. Once they are placed into operation, they send thousands of packets to the network and each one results in the ACL being changed and that address being blocked. Soon, many of the company's customers can no longer access the network. In order to maintain productivity, the network administrators are forced to turn off the Intrusion Prevention System, leaving the intruder to attempt the illegal activity.

While the preceding scenario is fictitious, it is a realistic look at what could actually happen. An actual instance of this type of exploitation could not be found, but is easy to visualize. The problem of using a security system against the network is not a new one, but Intrusion Prevention takes the problem to a new level much more hazardous than any found in the past. Intrusion Prevention systems are designed to shutdown access to the protected network based upon traffic it receives from the outside world. This makes it trivial for hostile actions to be directed at the system and have it do what it was designed to do.

Overcoming its use against the protected network.

It is beyond the scope of this paper to discuss methods for defeating a Denial of Service attack against a network. However, an Intrusion Prevention System must have the ability to monitor its own activities.

Some mechanism must be in place that will alert to the fact that an abnormal number of intercepts and reactions have taken place within a very short period of time and across a wide range of IP addresses. This will allow network administration to work with their ISPs to deal with the attack without the resulting denial of service.

Flaw 3 – False Negatives

False negatives are simply the failure of the Intrusion Prevention system to recognize an attack and take the appropriate actions. This may occur on a signature or anomaly based system.

Keeping in mind that Intrusion Prevention begins with Intrusion Detection, you must realize that Signature based systems have a big problem with false negatives. Simply stated, a signature must be available for a particular attack or

it will never be detected. Regardless of the number or complexity of the systems, a signature is absolutely necessary for detection and the system is worthless without it. New attacks that have not been identified will not have the prerequisite signature and will not be detected until the attack pattern is known and a signature is developed.

Even known attacks that have a signature may be missed if the attacker modifies the attack to be outside the signature parameters. Evasion of Intrusion Detection systems has been successfully accomplished using several different methods. In his article, "Are There Limitations of Intrusion Signatures"[8], published on the SANS Intrusion Detection FAQ, Matthew Richard does a fine job of describing possible evasions of IDS signatures. He describes a process by which a simple changing of the first byte of the payload can alter the entire payload rendering it invisible to the IDS based on the signature available.

Another problem encountered in the Network Based IDS technology is that of data encryption. Once encryption technology is introduced, the traffic is unreadable by the IDS and is, therefore, rendered useless. A good example of how encryption can be used against an Intrusion Detection/Prevention System is attacks against SSL encrypted websites on port 443. Network based intrusion detection systems have no hope of alerting to such attacks.

The last problem that will be discussed concerning false negatives is the issue of fast networks with more traffic than can be monitored by a normal system. When traffic arrives at a sensor faster than the sensor can read it, the packets are merely dropped by the sensor, but they will arrive at the destination normally. Attacks may be successful because they are not detected on the busy network due to excess traffic.

Overcoming False Negatives

Overcoming false negatives is as difficult as overcoming false positives, maybe even more so. Certainly, false negatives can be more damaging to the network than false positives.

The first issue that must be addressed in overcoming false positives is that of a lack of signature for a particular attack. Anomaly detection systems do not have this problem, but are also in the minority of deployed systems. As far as signature based systems, the only thing that can be done to mitigate the issue is to make sure that the systems are always up to date with the most current signature files applicable to the network. Choosing a system where the signature files are rapidly produced after an attack or vulnerability is discovered will aid in this quest.

Unfortunately, there is no answer for the problem of signature shifting until another signature has been developed for the shifted attack. The problem can best be addressed by the use of "defense in depth". Properly configured firewalls, different types of Intrusion Detection/Prevention systems, use of input

[8] Richard, Matthew. SANS Intrusion Detection FAQ, "Are There Limitations of Intrusion Signatures", April 5, 2001 URL: http://www.sans.org/resources/idfaq/limitations.php (February 23, 2003)

validation, and ensuring all systems are updated to prevent the known exploits is the best method of safeguarding systems.

The problem of encryption is overcome by the use of host-based systems. As most host-based systems depend on other sources of information and not the network traffic, the effect of the traffic will trigger the alert and corresponding response and not the traffic itself.

A relatively new technology that will also mitigate the encryption problem is the "Inline" Intrusion Detection/Prevention System. In this case, a network style system that monitors and alerts/reacts to network traffic is built into the host. The traffic would be decrypted by the host and then compared to the signatures. Tim Slighter has produced an intriguing white paper entitled "Configuring IPTables for Snort Inline"[9] in which he details the process of installing Snort as an inline system.

There are other systems that are said to be inline, but are not a part of the host. These systems act much like a firewall, intercepting traffic, scanning for hostile content, and then forwarding it on to the appropriate host. This type of system is sometimes called Gateway IDS or GIDS.[10]

Fast networks causing dropped packets by the Intrusion Prevention System can be overcome by the use of hardware. There is enough information and discussion to author a paper on this topic alone. Monitoring gigabit Ethernet is one of the major topics of discussion presently in the Information Security world. Additional sensors on separate network segments may be all that is required to mitigate the problem. Larger systems may require the use of hardware such as Top Layers IDS Balancer, which purports to monitor "multiple network segments simultaneously at network speeds ranging up to multi-Gigabit configurations"[11]

Conclusion

Intrusion Prevention Systems are the wave of the future in Information Security. The proliferation of hacking, cracking, viruses, and other tools that may be used by any "Script Kiddie" to perform sophisticated attacks, plus the recent outbreak of politically motivated attacks, mean that more progressive methods of protection must be employed. Unfortunately, there are some flaws, only a few of which have been addressed by this paper, that limit the ability of the systems to protect the network.

There are ways to mitigate some of the flaws, but those employed still fail to completely resolve the problems. When all is said and done, the most effective answer is to build "Defense in Depth" and not rely on any one method to protect your valuable resources.

---

[9] Slighter, Tim. "Configuring IPTables for Snort Inline", January 23, 2003. URL: http://www.snort.org/dl/contrib/patches/inline/ (February 26, 2003)

[10] Liesen, Detmar. "Requirements for Enterprise-Wide Scaling Intrusion Detection Products. A Criteria Catalog for IT Executives, IDS Users, and Vendors. (Version 2002-06-19 Rev 3)". URL: http://www.snort.org/docs/IDS_criteria.pdf (February 26, 2003)

[11] Top Layer Products, "Products and Solutions", URL: http://www.toplayer.com/content/products/index.jsp (February 26, 2003)

White Paper References

Richard, Matthew. SANS Intrusion Detection FAQ, "Are There Limitations of Intrusion Signatures", April 5, 2001 URL: http://www.sans.org/resources/idfaq/limitations.php (February 23, 2003)

Cummings, Joanne. "Intrusion detection to Intrusion prevention." Network World. Volume 19, No.38 (2002): 72-82. (May also be found at URL: http://www.nwfusion.com/buzz/2002/intruder.html (February 23, 2003))

Snort Signature Database, URL: http://www.snort.org/cgi-bin/needed.cgi?offset (February 23, 2003)

Briney, Andy. "What Isn't Intrusion Prevention" April 2002, URL: http://www.infosecuritymag.com/2002/apr/note.shtml (February 23, 2003)

Piscitello, David. "Intrusion Detection … or Prevention?", URL: http://www.bcr.com/bcrmag/2002/05/p42.asp (February 23, 2003)

Sequeira, Dinesh. SANS InfoSec Reading Room, "Intrusion Prevention Systems – Security's Silver Bullet?" URL: http://www.sans.org/rr/intrusion/silver_bullet (February 23, 2003)

Messmer, Ellen. "Intrusion prevention systems raise hopes, concerns", URL: http://www.computerworld.com/securitytopics/security/story/0,10801,75630,00.html (February 25, 2003)

Ethereal Network Protocol Analyzer, URL: http://www.ethereal.com/ (February 25, 2003)

Brindley, Adrian. "Denial of Service Attacks and the Emergence of Intrusion Prevention Systems" November 1, 2002. URL: http://www.sans.org/rr/firewall/prevention.php (February 25, 2003)

Slighter, Tim. "Configuring IPTables for Snort Inline", January 23, 2003. URL: http://www.snort.org/dl/contrib/patches/inline/ (February 26, 2003)

Liesen, Detmar. "Requirements for Enterprise-Wide Scaling Intrusion Detection Products. A Criteria Catalog for IT Executives, IDS Users, and Vendors. (Version 2002-06-19 Rev 3)". URL: http://www.snort.org/docs/IDS_criteria.pdf (February 26, 2003)

Top Layer Products, "Products and Solutions", URL: http://www.toplayer.com/content/products/index.jsp (February 26, 2003)

Assignment Part 2 – Network Detects

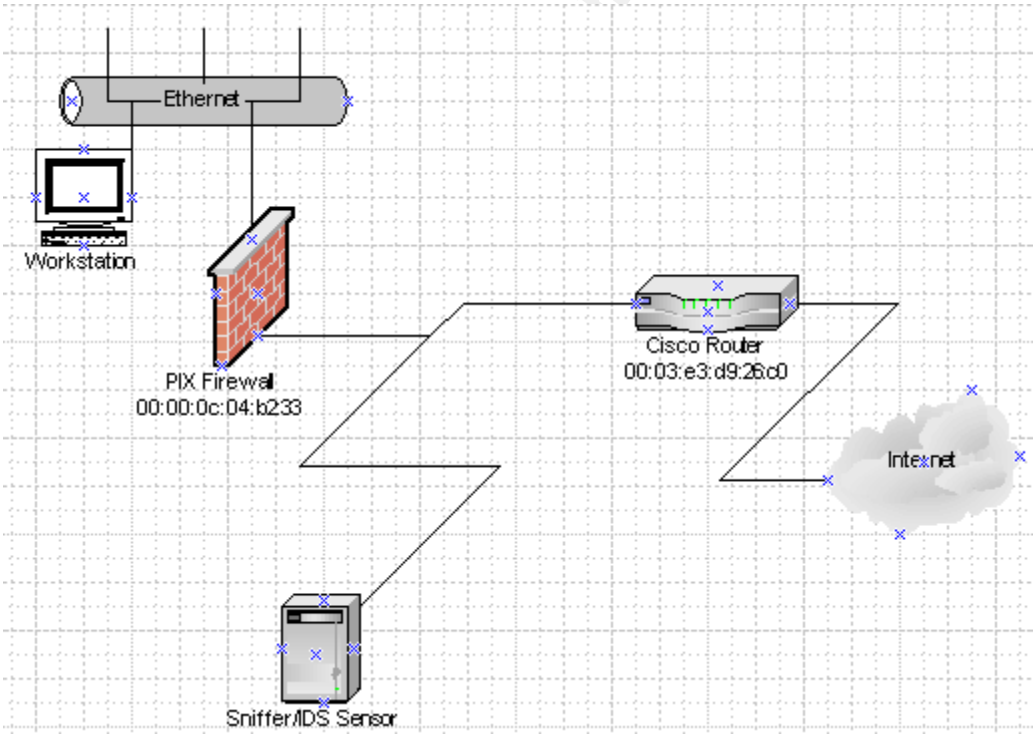
Detect #1 – Looking for a Trojan

1. Source of trace:

The trace file used for this investigation is file "2002.9.31" downloaded from the log files on the Incidents.Org website.[12] It was generated by Snort, but the rules used were not provided on the website.

In order to accurately assess and evaluate the meaning of the traffic, it is necessary to analyze the network where the trace originated. This requires a broad view of the network traffic. A full dump of the traffic was accomplished by using Windump with the command line syntax of "windump -ner c:\logs\2002.9.31 > c:\logs\tcpdump7.txt". This starts windump, turns off the Name Address Resolution, Includes the Data Link Headers, reads the appropriate log file, and dumps the results to a text file. Once the text file was generated, it was imported into a Microsoft Access® database. This allows for advanced queries to be run against the data.

Based on the information available in the data, the following network diagram was developed.



Here is how the diagram was developed. First to note is that there are only two MAC addresses showing in the data, 00-00-0c-04-b2-33 and 00-03-e3-d9-26-c0, while there are multiple IP addresses coming from each MAC address.

[12] Incidents.Org Log Files, URL: http://www.incidents.org/logs/Raw/. (February 28, 2003)

Second, when you research the MAC addresses running the 24bit company id section of the MAC address through the "IEEE OUI and Company ID Database",[13] the results are as follows:

```
00-03-E3    (hex)                Cisco Systems, Inc.
0003E3      (base 16)            Cisco Systems, Inc.
                                 170 West Tasman Dr.
                                 San Jose CA 95134
                                 UNITED STATES
```

The other MAC address also results in Cisco Systems as the manufacturer. This screenshot has been left off in the interest of conserving space. Since all traffic was coming from two Cisco devices, it only makes sense that the sensor gathering the data is between them. Unfortunately, the MAC addresses do not differentiate between router devices and firewall devices, so there could be any mix of the two on this network. The diagram provided was an assumption based upon other factors such as the frequency and success of SYN packets and the number of different IP subnets originating at the devices. There were 459 SYN packets sent through the network during the time period the traffic was monitored and captured. All 459 SYN packets came through the device with MAC address 00-03-e3-d9-26-c0. All the packets coming through MAC address 00-00-0c-04-b2-33 were in the 207.166.x.x subnet. In addition, all outbound traffic from the 207.166.x.x subnet had the destination MAC address of 00-03-e3-d9-26-c0, establishing it as the gateway to the Internet. I believe the inside device is a firewall because of the number of SYN packets that were inbound versus the single SYN-ACK packet that was returned. This would seem to indicate a dead network, a firewall, or a router with ACL's restricting traffic.

2. Detect was generated by:

When the trace file was read by Snort in order to generate detects, the following alert was generated and will be the focus of this investigation:

```
[**] [1:0:0] IDS175/misc_socks-probe [**]
[Classification: relay attempt] [Priority: 9]
10/31-08:55:14.6507 204.94.58.44:44196 -> 207.166.87.157:1080
TCP TTL:110 TOS:0x0 ID:59308 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE6BB33C7  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[Xref =>  arachnids 175]
```

This alert was generated by Windows version 1.9 of Snort as shown in the screenshot below:

[13] IEEE. "IEEE OUI and Company ID Database", URL: http://standards.ieee.org/regauth/oui/index.shtml (February 28, 2003)

```
C:\Snort>snort -V
Initializing Output Plugins!

-*> Snort! <*-
Version 1.9.0-ODBC-MySQL-WIN32 (Build 209)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
1.7-WIN32 Port By Michael Davis (mike@datanerds.net, www.datanerds.net/~mike)
1.8-1.9 WIN32 Port By Chris Reid (chris.reid@codecraftconsultants.com)
```

The rules used for the detect were downloaded from the file vision18.conf.gz at Whitehats, URL: http://www.whitehats.com/ids/index.html .[14] The specific rule that generated the alert is as follows[15]:

```
alert TCP $EXTERNAL any -> $INTERNAL 1080 (msg:
"IDS175/misc_socks-probe"; ack: 0; flags: S; classtype: relay-
attempt; reference: arachnids,175;)
```

In order to break this rule down and the alert it generates, they will be compared next to each other as shown below:

```
alert TCP $EXTERNAL any -> $INTERNAL 1080 (msg:
"IDS175/misc_socks-probe"; ack: 0; flags: S; classtype: relay-
attempt; reference: arachnids,175;)
```

[**] [1:0:0] IDS175/misc_socks-probe [**]
Taken Directly from the Rule
[Classification: relay attempt] [Priority: 9]
Taken Directly from the Rule
10/31-08:55:14.6507 204.94.58.44:44196 -> 207.166.87.157:1080
Source is External *Destination is Internal Port Matches*
TCP TTL:110 TOS:0x0 ID:59308 IpLen:20 DgmLen:48 DF
Data taken from traffic.
******S* Seq: 0xE6BB33C7 Ack: 0x0 Win: 0x4000 TcpLen: 28
SYN only *Ack is 0*
TCP Options (4) => MSS: 1460 NOP NOP SackOK
Data taken from traffic
[Xref => arachnids 175]
Taken Directly from the Rule

[14] WhiteHats.Com, "ArachNIDS Database", URL: http://www.whitehats.com/ids/index.html (February 28, 2003)
[15] Whitehats.com , "IDS175 SOCKS PROBE", Research Tab, URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids175&view=research (March 1, 2003)

3. Probability the source address was spoofed:

It is highly unlikely the source address of this traffic was spoofed as the sender would need a reply in order to be effective. While there is not a connection required and the source IP could be spoofed, there is no logical reason to do so in this case.

4. Description of attack:

According to Whitehats concerning this attack, "hackers scour the Internet religiously looking for systems they can bounce their attacks through. This intrusion signature indicates that somebody scanned your system looking for SOCKS." One could very easily read this section and decide that as long as there are no SOCKS servers running there is not a problem. As this investigation will reveal, the alerts are not always what they seem to be.

When the trace file is read by Windump using the command syntax of "Windump -nvvr c:\logs\2002.9.31 'host 204.94.58.44 and host 207.166.87.157' > c:\logs\tcpdump6.txt", the following data is produced and will be used for the investigation:

```
08:55:07.466507 IP (tos 0x0, ttl 110, id 59210, len 48) 204.94.58.44.44183 > 207.166.87.157.8080: S
3868902610:3868902610(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41f9 (->f7ae)!

08:55:08.076507 IP (tos 0x0, ttl 110, id 59214, len 48) 204.94.58.44.44183 > 207.166.87.157.8080: S
3868902610:3868902610(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41f5 (->f7aa)!

08:55:08.626507 IP (tos 0x0, ttl 110, id 59219, len 48) 204.94.58.44.44183 > 207.166.87.157.8080: S
3868902610:3868902610(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41f0 (->f7a5)!

08:55:08.796507 IP (tos 0x0, ttl 110, id 59221, len 48) 204.94.58.44.44185 > 207.166.87.157.3128: S
3869327733:3869327733(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41ee (->f7a3)!

08:55:09.316507 IP (tos 0x0, ttl 110, id 59224, len 48) 204.94.58.44.44185 > 207.166.87.157.3128: S
3869327733:3869327733(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41eb (->f7a0)!

08:55:09.816507 IP (tos 0x0, ttl 110, id 59229, len 48) 204.94.58.44.44185 > 207.166.87.157.3128: S
3869327733:3869327733(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41e6 (->f79b)!

08:55:13.486507 IP (tos 0x0, ttl 110, id 59285, len 48) 204.94.58.44.44196 > 207.166.87.157.1080: S
3871028167:3871028167(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41ae (->f763)!

08:55:14.046507 IP (tos 0x0, ttl 110, id 59292, len 48) 204.94.58.44.44196 > 207.166.87.157.1080: S
3871028167:3871028167(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 41a7 (->f75c)!

08:55:14.6507 IP (tos 0x0, ttl 110, id 59308, len 48) 204.94.58.44.44196 > 207.166.87.157.1080: S
3871028167:3871028167(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 4197 (->f74c)!

08:55:14.706507 IP (tos 0x0, ttl 110, id 59311, len 48) 204.94.58.44.44201 > 207.166.87.157.1080: S
3871614967:3871614967(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 4194 (->f749)!

08:55:15.226507 IP (tos 0x0, ttl 110, id 59322, len 48) 204.94.58.44.44201 > 207.166.87.157.1080: S
3871614967:3871614967(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 4189 (->f73e)!

08:55:15.736507 IP (tos 0x0, ttl 110, id 59333, len 48) 204.94.58.44.44201 > 207.166.87.157.1080: S
3871614967:3871614967(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)bad cksum 417e (->f733)!
```

The first things of significance to note are the destination ports of this traffic. While we were only alerted to port 1080, there are two other ports that the source

attempted at the same time; port 8080 and 3128. The second item to note is that the target of the scan did not respond to any of the traffic, that is the good news. The third analysis of the data to note is that while there are nine packets shown here, there were only three attempts, one attempt with three identical packets for each port number. The second three are the second attempt and the third three are the third attempt. This is evident because the sequence numbers match each other within the sets, but the timing is not right for them to be retries.

Next, it is prudent to determine what else, besides a SOCKS probe, if indeed that is what this was, was being attempted. A lookup of the well-known port numbers was accomplished on the IANA website[16] for the three ports with the following results:

```
ndl-aas       3128/tcp   Active API Server Port
ndl-aas       3128/udp   Active API Server Port

socks         1080/tcp   Socks
socks         1080/udp   Socks

http-alt  8080/tcp     HTTP Alternate (see port 80)
http-alt  8080/udp     HTTP Alternate (see port 80)
```

This confirms 1080 may be used for Socks and 3128 for API, and 8080 for an HTTP alternate. This does not shed any light on the subject, so a general word search went out for these ports on the Internet using Google[17]. This led to an interesting site about Trojans at Simovits.Com[18] that shed some light on the possible uses of these ports:

port 3128 Reverse WWW Tunnel Backdoor , RingZero
port 1080 SubSeven 2.2, WinHole
port 8080 Reverse WWW Tunnel Backdoor , RingZero, Screen Cutter

While these ports all have legitimate uses that are not relevant to one another, they also have illegitimate uses for Trojans. It is interesting to note that ports 3128 and 8080 are both used for the same Trojans, while port 1080 is used for different ones.

Based on this evidence, it is most likely that the originator of this traffic was probing for a response from one of the Trojans listed above. While the possibility that a query was being made to a Socks server still exists, the fact that these uses coincide with each other is strong evidence to the contrary.

[16] IANA, "Well Known Ports Database", (February 26, 2003) URL : http://www.iana.org/assignments/port-numbers (March 1, 2003)
[17] Google Search Engine, URL: www.google.com (March 1, 2003)
[18] Simovits Consulting, "Ports Used by Trojans (2002-10-15)", URL: http://www.simovits.com/nyheter9902.html (March 2, 2003)

5. Attack mechanism:

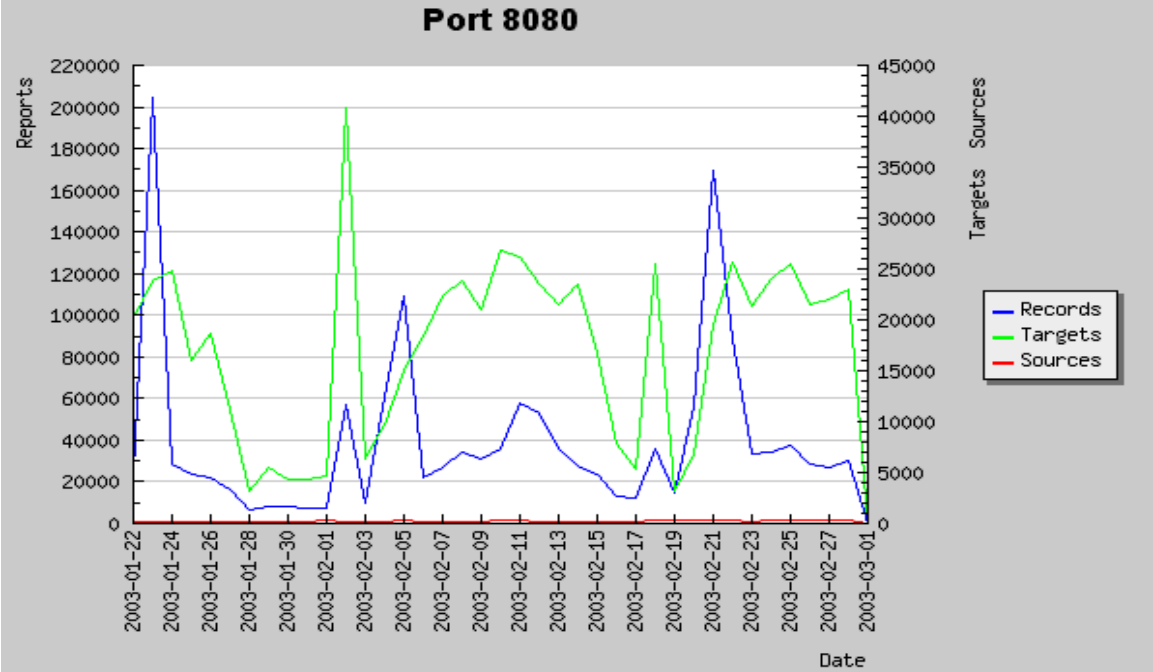
Trojans are programs that are generally considered to be part of the virus family. According to the Cert Coordination Center, Trojans may be installed by tricking or enticing users to install the program from an email attachment, by hiding the Trojan in a legitimate program, or placing it within compromised web sites. [19] Once the Trojan is installed, the next step is for the attacker to be able to contact the Trojan. The Trojan program does this by sending a connection notification to a specified email address. This notification would include the IP address of the compromised system. Another method is to scan for systems that have the Trojan installed. Once contact is made, the attacker can then do whatever the Trojan is designed to do. This can be quite extensive up to and including full control of the compromised machine.[20]

In the current investigation, we saw the external IP address attempting to contact a Trojan or Trojans across three different ports. In order to determine exactly what is occuring, it is necessary to perform a forensic analysis on the target machine. It is highly likely that this machine has been compromised by a Trojan and is available to answer on one or more of the suspect ports. Multiple ports are used to increase the chances that the traffic will pass through any firewalls encountered.

6. Correlations:

Incidents.Org is reporting activities on Port 8080 as shown in the screenshots from their website below:[21]

[19] CERT Coordination Center, "Cert Advisory CA-1999-02 Trojan Horses", URL: http://www.cert.org/advisories/CA-1999-02.html (March 2, 2003)
[20] ibid.
[21] Internet Storm Center, "Port Reports", URL: http://isc.incidents.org/port_details.html?port=8080 (March 2, 2003)



Port 8080 activity between Jan 22, 2003 and March 1, 2003.

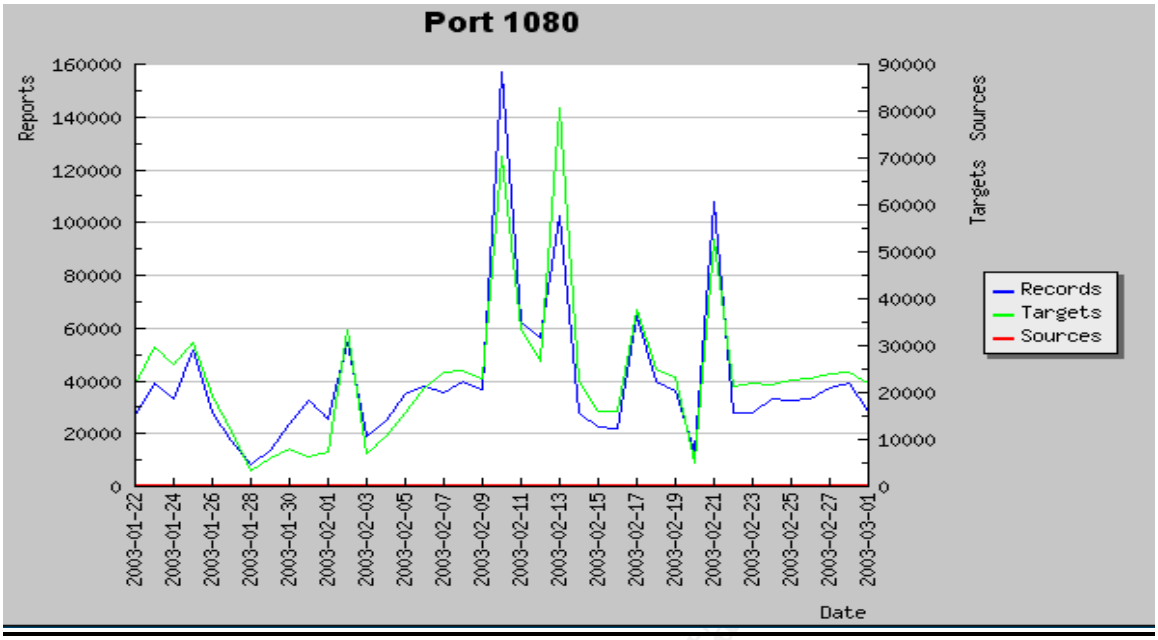
Services registered for this port (from Neohapsis)

Protocol	Service	Name
tcp	http-alt	HTTP Alternate (see port 80)
udp	http-alt	HTTP Alternate (see port 80)
tcp	BrownOrifice	[trojan] Brown Orifice
tcp	BrownOrifice	[trojan] Brown Orifice
tcp	Genericbackdoor	[trojan] Generic backdoor
tcp	RemoConChubo	[trojan] RemoConChubo
tcp	ReverseWWWTunnel	[trojan] Reverse WWW Tunnel Backdoor
tcp	RingZero	[trojan] RingZero

Port 8080 Services Registered at Neohapsis (from the Incidents.Org Website)

Incidents.Org is reporting activities on Port 1080 as shown in the screenshots from their website below:[22]

[22] Internet Storm Center, "Port Reports", URL: http://isc.incidents.org/port_details.html?port=1080 (March 2, 2003)



Port 1080 activity between Jan 22, 2003 and March 1, 2003.

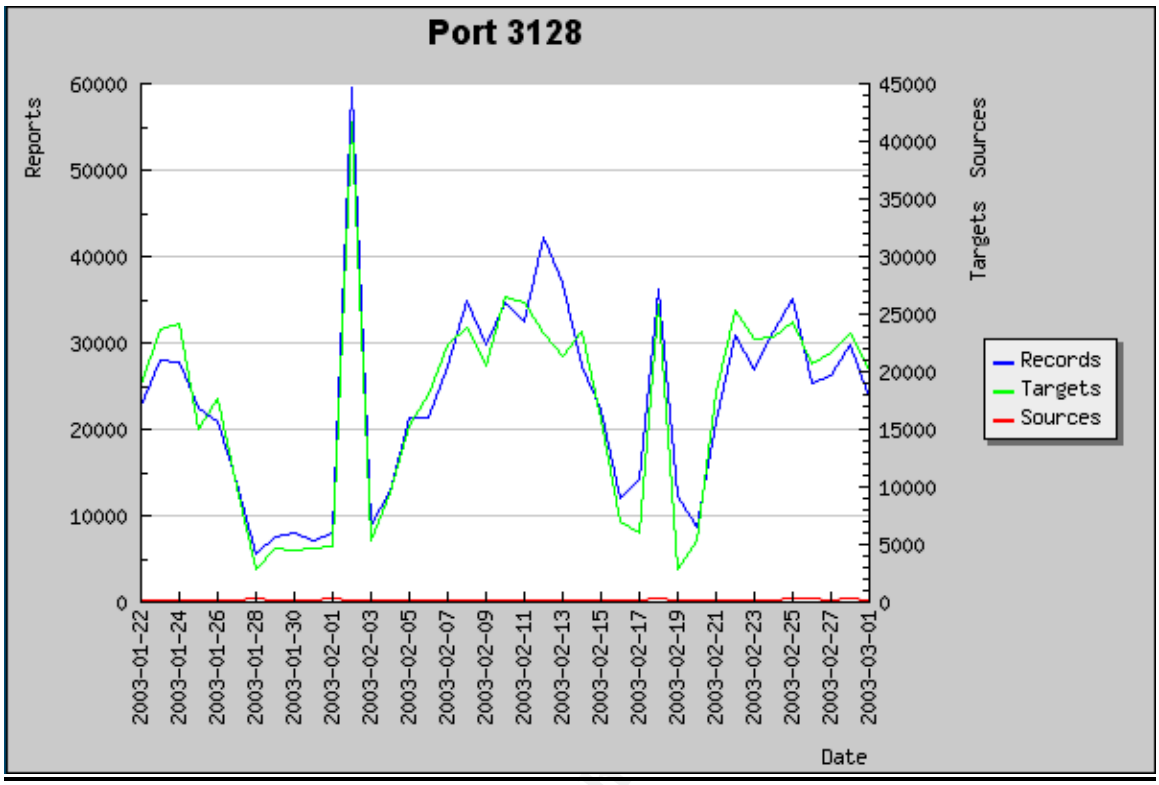
Services registered for this port (from Neohapsis)

Protocol	Service	Name
tcp	socks	Proxy Server
udp	socks	Proxy Server
tcp	SubSeven2.2	[trojan] SubSeven 2.2
tcp	WinHole	[trojan] WinHole
tcp	WinHole	[trojan] WinHole

Port 1080 Services Registered at Neohapsis (from the Incidents.Org Website)

Incidents.Org is reporting activities on Port 3128 as shown in the screenshots from their website below:[23]

[23] Internet Storm Center, "Port Reports", URL: http://isc.incidents.org/port_details.html?port=3128 (March 2, 2003)



Port 3128 activity between Jan 22, 2003 and March 1, 2003.

Services registered for this port (from Neohapsis)

Protocol	Service	Name
tcp	squid-http	Proxy Server
tcp	ReverseWWWTunnel	[trojan] Reverse WWW Tunnel Backdoor
tcp	RingZero	[trojan] RingZero
tcp	RingZero	[trojan] RingZero

Port 3128 Services Registered at Neohapsis (from the Incidents.Org Website)

As you can see from the preceding charts, all three of these ports are currently very active on the Internet.

7. Evidence Of active targeting:

It is very obvious that this attack was the result of active targeting. On this entire network, the only traffic that was sent from the attackers IP address was directed at this one server. It is not likely a wrong number as the source sent packets on three different port numbers.

The reason for the active targeting should be discovered if at all possible. This will take a forensic analysis of the targeted machine, which is beyond the scope of this paper. However, the likely scenario is that the target has been compromised by a Trojan. It sent out notification to the attacker, who then attempted to make contact with the Trojan.

```
....&... ...3..E.
.ns.@.|. ....w...
.N..... 6.[q7XP.
Dp+c..GN UTELLA C
ONNECT/0 .6..User
-Agent:  Morpheus
 2.0.1.8 ..X-Ultr
apeer: F alse..PO
NG-CACHI NG: 0.1.
.X-MY-AD DRESS: X
XXXXXX.5 0.120:79
88..X-Tr y: 12.22
0.36.242 :6346,20
7.159.11 1.111:63
47,24.19 7.179.22
9:7795,4 .0.0.3:6
685,155. 68.71.24
9:6346,1 69.254.1
85.2:644 9,169.22
9.82.91: 5715,147
.124.52. 82:9934,
132.241. 209.176:
5821,207 .210.XXX
.88:6346 ....
```

Another possible scenario concerns the use of other Internet programs. The targeted machine has been very active on the Internet including the use of Gnutella, a Peer-to-Peer networking program that shares its IP address with other users as evidenced by the contents of this Gnutella packet. (The IP address has been obfuscated in this example for security reasons.)

8. Severity:

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

Criticality – The preponderance of evidence would suggest the targeted system is an end-workstation. The evidence includes a varying type of traffic that is seen originating from the system, including normal web browsing as shown below:

```
2004 2002-10-31 04:41:03.596507 207.166.87.157   140.184.85.171   TCP   63575 > 9671 [PSH, ACK] Seq=3038635680 A
2006 2002-10-31 04:42:35.506507 207.166.87.157   210.65.1.187     HTTP  GET /UserData/ukyo9/2002/%E9.gif B.gif H
2007 2002-10-31 04:45:32.266507 207.166.87.157   24.242.241.133   TCP   61530 > 8406 [PSH, ACK] Seq=3108436013 A
```

Notice the HTTP Get command. Also, the HTTP traffic content shows us that the browser is Windows Internet Explorer 5.5 compatible. This is probably a Windows 2000 system. This would rate 2 points as a desktop system according to the criticality charts.[24]

```
: Mozill a/4.0 (c
ompatibl e; MSIE
5.5; win dows NT
5.0; HQ1 0818)..H
```

[24] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-8 (Severity:Criticality Chart).

Lethality – Trojans have the capability to give full control access to the attacker. It therefore meets the chart criteria for 5 points as the attacker can gain root access across the net[25].

System Countermeasures – It appears that there are no system countermeasures in place, but that is really only a guess until a forensic analysis is accomplished. According to the chart, the score for this metric is 1.[26]

Network Countermeasures – It appears the network countermeasures were totally effective in blocking the inbound traffic. There is no indication that there is another method of entering or exiting this network than through the firewall. According to the chart, the score is 5.[27]

Severity = (2 + 5) – (1 + 5) = 1 (Low)

9. Defensive Recommendation:

Based on the information available, here are recommendations to further enhance network security:

- Ensure the target workstation is clean from Trojans and install an anti-virus system that will detect future Trojans.
- Prohibit the use of Gnutella and other Peer-to-Peer programs on the networks. Enforce with custom IDS Signatures such as:

  Alert tcp any any -> any any (msg: "Gnutella Traffic Detected"; flags: A+; Content: "Gnutella")

[25] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-9 (Severity:Lethality Chart).
[26] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-14 (Severity:System Countermeasures Chart).
[27] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-15 (Severity:Network Countermeasures Chart).

10. Multiple choice test question:

```
08:55:07.466507 IP 204.94.58.44.44183 > 207.166.87.157.8080: S 3868902610:3868902610(0) win 16384 <mss
1460,nop,nop,sackOK> (DF)
08:55:08.076507 IP 204.94.58.44.44183 > 207.166.87.157.8080: S 3868902610:3868902610(0) win 16384 <mss
1460,nop,nop,sackOK> (DF)
08:55:08.626507 IP 204.94.58.44.44183 > 207.166.87.157.8080: S 3868902610:3868902610(0) win 16384 <mss
1460,nop,nop,sackOK> (DF)
```

Which of the following statements about the trace shown above is correct?

A. The source IP address in the trace is absolutely spoofed, otherwise, there would not be three packets with the same sequence number.
B. This traffic represents an original packet and then TCP retries.
C. This traffic represents three original packets sent with identical information.
D. The Window size of these packets is incorrect.

Answer: C

Explanation:

A is not correct as it can't be absolutely proven by what is shown here that the source IP address is spoofed. The sequence numbers have nothing to do with it.

B is not correct as the subsequent two packets are not retries. TCP retries occur at specific intervals of three seconds for the first retry and then six seconds for the second retry. These packets were sent all at once.

C is the correct answer as it is opposite of B above.

D is just a distracter as there is nothing wrong with this window size.

Detect #2 – Directory Traversal

1. Source of trace:

The trace file used for this investigation is file "2002.10.17" downloaded from the log files on the Incidents.Org website.[28] It was generated by Snort, but the rules used were not provided on the website.

In order to accurately assess and evaluate the meaning of the traffic, it is necessary to analyze the network where the trace originated. This requires a broad view of the network traffic. A full dump of the traffic was accomplished by using Windump with the command line syntax of "windump -ner c:\logs\2002.10.17 > c:\logs\tcpdump.txt". This starts windump, turns of the Name Address Resolution, Includes the Data Link Headers, reads the appropriate log file, and dumps the results to a text file. Once the text file was generated, it was imported into a Microsoft Access® database. This allows for advanced queries to be run against the data.

Based on the information available in the data, this appears to be the same network as described in Detect #1 in this paper. The IP addresses are different and have probably been obfuscated for security purposes. The MAC addresses of the router and firewall are identical, however, so the network must be the same.

2. Detect was generated by:

When the trace file was read by Snort in order to generate detects, the following alert was generated and will be the focus of this investigation:

```
[**] [1:0:0] IDS297/web-misc_http-directory-traversal1 [**]
[Classification: system integrity attempt] [Priority: 11]
11/16-20:27:37.476507 211.87.212.36:1393 -> 170.129.130.226:80
TCP TTL:100 TOS:0x0 ID:41944 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xDF5A3BBD  Ack: 0x1235  Win: 0x4470  TcpLen: 20
[Xref =>  arachnids 297]
```

This alert was generated by Windows version 1.9 of Snort as shown in the screenshot below:

```
C:\Snort>snort -V
Initializing Output Plugins!

-*> Snort! <*-
Version 1.9.0-ODBC-MySQL-WIN32 (Build 209)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
1.7-WIN32 Port By Michael Davis (mike@datanerds.net, www.datanerds.net/~mike)
1.8-1.9 WIN32 Port By Chris Reid (chris.reid@codecraftconsultants.com)
```

[28] Incidents.Org Log Files, URL: http://www.incidents.org/logs/Raw/. (February 28, 2003)

The rules used for the detect were downloaded from the file vision18.conf.gz at Whitehats, URL: http://www.whitehats.com/ids/index.html .[29] The specific rule that generated the alert is as follows[30]:

```
alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS297/web-
misc_http-directory-traversal1"; flags: A+; content: "../";
classtype: system-attempt; reference: arachnids,297;)
```

In order to break this rule down and the alert it generates, they will be compared next to each other as shown below:

```
alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS297/web-
misc_http-directory-traversal1"; flags: A+; content: "../";
classtype: system-attempt; reference: arachnids,297;)
```

[**] [1:0:0] IDS297/web-misc_http-directory-traversal1 [**]
Taken Directly from the Rule
[Classification: system integrity attempt] [Priority: 11]
Taken Directly from the Rule
11/16-20:27:37.476507 211.87.212.36:1393 -> 170.129.130.226:80
Source is External *Destination is Internal Port Matches*
TCP TTL:100 TOS:0x0 ID:41944 IpLen:20 DgmLen:136 DF
Data taken from traffic.
AP Seq: 0xDF5A3BBD Ack: 0x1235 Win: 0x4470 TcpLen: 20
Ack Plus Another Flag
[Xref => arachnids 297]
Taken Directly from the Rule

3. Probability the source address was spoofed:

It is highly unlikely the source address of this traffic was spoofed as the sender would need a reply in order to be effective. This particular attack also requires a TCP Connection to be established making the possibility of spoofing the source address very minimal.

4. Description of attack:

This is a Web URL Encoding Attack[31] used to accomplish a Directory Traversal[32] against a web server. It uses URL encoding of Unicode representation of characters to replace standard characters that would not normally be allowed by the web server. For example, a slash (/) or a backslash (\) may be rejected as invalid characters by the web operating system in order to

[29] WhiteHats.Com, "ArachNIDS Database", URL: http://www.whitehats.com/ids/index.html (February 28, 2003)
[30] Whitehats.com , "IDS175 SOCKS PROBE", Research Tab, URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids175&view=research (March 1, 2003)
[31] @Stake, "Application Security Principles Course Book", 2002, Pg 58
[32]

prevent directory traversal (moving through the directory). Using URL encoding, the invalid characters are not passed to the web operating system rather the codes are converted to unicode representation, url encoded, passed to the web server, and then interpreted by the system. This works because all characters have a unicode representation.[33] The Unicode values are then url-encoded by replacing the leading zeros with the % symbol.[34] For example, the unicode representation for "/" is 0025 and the url-encoded unicode is %25.[35]

5. Attack mechanism:

In order to understand this attack, you must first closely examine the contents of the packets that were involved in the attack. The following screenshot from Ethereal shows just the HTTP contents:

```
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir HTTP/1.0
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir c+dir HTTP/1.0
GET /msadc/..%5c../..%5c../..%5c/..55../..c1../../.../winnt/system32/cmd.exe?/c+dir 32/cmd.exe?/c+dir HTTP/1.0
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir c+dir HTTP/1.0
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir r HTTP/1.0
```

Looking at the first line that reads:

Get /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r http/1.0

The "Get" here is a standard HTTP get command. Since the scripts directory is normally one that is accessible to the web users, the attacker will start there. After that it become more interesting. The next section says "/scripts/..%5c../", which is URL encoding. The %5c is an encoded "\"[36] making that section "/scripts/..\../". This is beginning to make a little more sense. It means that the system is to look for the files by going to the scripts folder and then changing the directory upward two times, which would take it to the root of C drive. Then it would walk down the directory structure to "/winnt/system32/" and execute the cmd (command) executable. Once running successfully, it will list the contents of the directory.

That must not have worked, as the remaining packets carry a variation of the same attempt. All of the attempts have the same goal, to traverse through the directory to a different directory that has the command line or perhaps other tools that could be used to gain advantage or access on the server. You will notice

[33] @Stake, "Application Security Principles Course Book", 2002, Pg 58
[34] ibid
[35] ibid
[36] @Stake, "Application Security Principles Course Book", 2002, Pg 60

%2f in some of the attempts, which decodes to "/". The representation can be double-encoded in order bypass safeguards against encoding.

It appears by the continued attempts at the same attack that the attacker is not successful. Had the attacker been successful, the same technique could be used to run virtually any command on the server.

6. Correlations:

There are several resources to correlate this attack from both the directory traversal and the url-encoding attack perspective.

Internet Security Systems intrusion number 2000645[37] describes double-url encoding, similar to what was seen in this case.

According to The Open Web Application Security Project: "The query portion of the URL is often used to submit data to the server. URL-encoding is a technique defined in the URL/URI specifications for mapping 8-bit data to the subset of the US-ASCII character set allowed in a URL/URI. Without proper validation, URL-encoded input can be used to disguise malicious code for use in a variety of attacks." [38]

The Cert Coordination Center at Carnegie Mellon University issued Vulnerability Note VU#111677[39] concerning a combination of the url-encoding and directory traversal problem that was seen here.

7. Evidence of active targeting:

This appears to be an incident of active targeting. The source IP address did not send traffic to any other destination address on this network. It was obviously not a random scan, though a scan not seen in this traffic may have set up the attack.

8. Severity:

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

Criticality – There is not enough traffic to confirm the function of the victim system. The available traffic does not contain any response or other outbound traffic from the victim system. However, all traffic to the victim is on port 80 so it

[37] Internet Security Systems, "HTTP URL with double encoded ../", URL: http://www.iss.net/security_center/advice/Intrusions/2000645/default.htm (March 4, 2003)
[38] The Open Web Application Security Project, "ASAC", URLL: http://www.owasp.org/asac/canonicalization/url.shtml (March 4, 2003)
[39] Cert Coordination Center, "Vulnerability Note VU#111677", URL: http://www.kb.cert.org/vuls/id/111677 (March 4, 2003)

is probably safe to assume it is a web server. According to the point scale system, the criticality score would be 4.[40]

Lethality – If this attack were successful, it would allow the attacker to have root access across the net. According to the severity chart, the Lethality score is 5.[41]

System Countermeasures – There is not enough information available to determine system countermeasures in this case. While there was no response from the server, it could have been the result of network countermeasures, system countermeasures, or the fact a rule was not present that would have caused the response to be logged. For the purposes of this paper, I will assume a system countermeasure level of 4. This would be interpreted as a modern operating system with all patches, but no other countermeasures.[42]

Network Countermeasures – There is not enough information available to determine network countermeasures in this case. While there was no response form the server, it could have been the result of system countermeasures as easily as network countermeasures. For the purposes of this paper, I will assume a network countermeasure level of 5. This would be interpreted as a "validated restrictive firewall, only one way in or out".[43]

severity = (4 + 5) – (4 + 5) = 0 (Low) This is borne out by the fact the server did not respond to the attack.

9. Defensive recommendation:

The primary method of defending against a Directory Traversal or Web URL Encoding Attack or both is a method termed "Input Validation"[44] This is a method whereby the web administrator installs patches, third party software, or other code to check each and every input by the client (user) to the web server to make sure it is a valid request and does not contain hostile intent. Authors must develop their code so it does not require the use of URL encoding to make it easier for the web administrators to employ Input Validation.

Using network security measures such as firewalls for this purpose is not realistic. In order for a web server to be effective, ports 80 and 443, the well-

[40] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-8 (Severity:Criticality Chart).
[41] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-9 (Severity:Lethality Chart).
[42] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-14 (Severity:System Countermeasures Chart).
[43] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-15 (Severity:Network Countermeasures Chart).
[44] @Stake, "Application Security Principles Course Book", 2002, Section 2-2, Slide 3

known ports for web traffic[45] must be open. All other ports can be closed, but these attacks will always take place on port 80 or 443.

On the host side, one defensive recommendation is to change the default setup of the web server directory structure. Web pages, including all of the supporting directory, should be placed on another server drive. This eliminates the possibility of getting to system files through Directory Traversal. In the event a separate drive is not available, the default directory structure should not be used, rather a completely new structure should be developed. Another defensive recommendation is to ensure that directory rights, in this case NTFS rights, are the minimum required to run the server.

10. Multiple choice test question:

```
211.87.212.36    170.129.130.226    HTTP    GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0
211.87.212.36    170.129.130.226    HTTP    GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
211.87.212.36    170.129.130.226    HTTP    GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
211.87.212.36    170.129.130.226    HTTP    GET /msadc/..%5c../..%5c../..%5c/..55../..c1../../.../winnt/system3
211.87.212.36    170.129.130.226    HTTP    GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir dir HTTP/1.0
211.87.212.36    170.129.130.226    HTTP    GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir r HTTP/1.0
211.87.212.36    170.129.130.226    HTTP    GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir c+dir HTTP/1.0
211.87.212.36    170.129.130.226    HTTP    GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir r HTTP/1.0
```

What information can you gather from looking at the traffic above, taken from an Ethereal Capture.

A. This is a normal request to a website as it is HTTP traffic.
B. The % sign indicates a url-encoding is being used.
C. This is an attack against a Microsoft Windows system.
D. Both B and C Above.

Answer: D

A is incorrect because, while this is HTTP traffic, it is not normal. It is using url-encoding and it is against a Microsoft Windows system.

Responses from Incidents.Org

The text that is shown in blue in this section is the text that is pasted directly from the response received. The text shown in green is my reply to the points made in the response.

Response 1:

Good analysis! Just a couple of little things.

[45] IANA, "Well Known Ports Database", (February 26, 2003) URL : http://www.iana.org/assignments/port-numbers (March 1, 2003)

> System Countermeasures - There is not enough information available to
> determine system countermeasures in this case. While there was no
> response from the server,

Don't forget that You only have the packets in the logs that generated alerts. Would a response have generated an alert?

Considering the Snort rule is looking for "../" in the content of the packet, I don't believe that a response to such a request would have generated an alert. However, a response would have given an indication, by how the system responded, as to whether or not system countermeasures were effective. The responder was saying I would not see the response based upon the logs that I had available to me. After this response, I clarified that point in the paper.

> 9. Defensive recommendation:
>
>
> The primary method of defending against a Directory Traversal or Web
> URL Encoding Attack or both is a method termed "Input Validation"[17]
> This is a method whereby the web author develops code to check each
> and every input by the client (user) to the web server to make sure
> it is a valid request and does not contain hostile intent.

I think the phrase "web author" disturbs me, because that phrase to me means someone who develops Web sites, not someone who develops a Web server (e.g. Apache, IIS, etc.). There is nothing a Web author can do about this specific attack, since the attack is against the Web server software itself. Your point is well taken, though, and certainly belongs here.

This is a good point. While input validation may be the responsibility of a web author in many cases, not in the case of Directory Traversal or Web URL Encoding Attack. Other means of input validation must be put in place, depending upon the type of web server involved. After this input, the paper was changed to reflect this.

> Using network security measures such as firewalls for this purpose is
> not realistic. In order for a web server to be effective, ports 80
> and 443, the well-known ports for web traffic[18] must be open. All
> other ports can be closed, but these attacks will always take place
> on port 80 or 443.

Yes, but many modern firewalls are capable of blocking certain URLs or patterns in URLs. Might that not be appropriate? How about a reverse proxy?

Can You think of additional host security measures that would defend against this attack? I can. Hint: You said that the root directory of the file system is two directories up from the scripts directory.

A good point was made here. Text was added to the paper concerning directory structure of web servers.

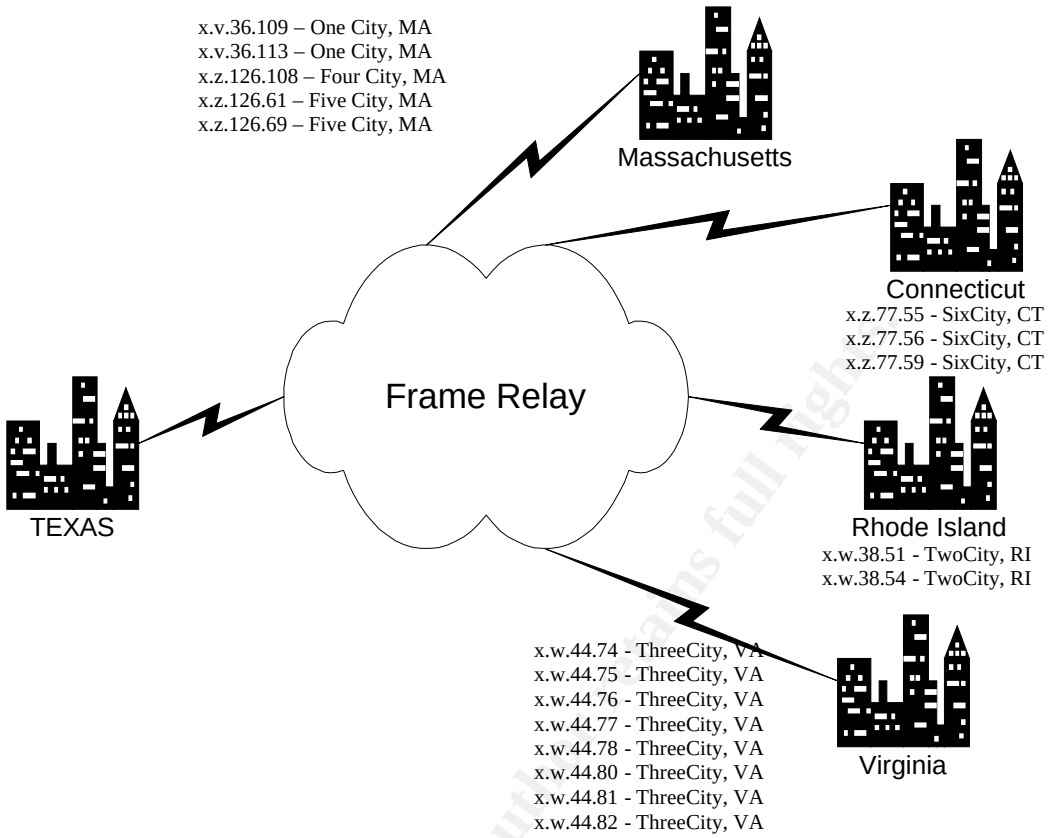
Detect #3 – Mystery Traffic

```
17:48:07.922888 IP (tos 0x0, ttl 128, id 19119, len 75) x.y.25.20.2034 > x.y.25.21.: [udp sum ok]  1+ A? CFB_NTS_S28.mylan.com.
(47)
17:48:07.923071 IP (tos 0x0, ttl 128, id 31445, len 78) x.y.25.21. > x.y.25.20.137: [udp sum ok]  61997+ NIMLOC?
EHEGECFPEOFEFDFPFDDCDICACACACAAA. (50)
17:48:07.923673 IP (tos 0x0, ttl 128, id 19375, len 90) x.y.25.20.137 > x.y.25.21.: [udp sum ok]  61997* [0q] 1/0/0
EHEGECFPEOFEFDFPFDDCDICACACACAAA. NIMLOC (62)
17:48:07.924376 IP (tos 0x0, ttl 128, id 31701, len 91) x.y.25.21. > x.y.25.20.2034: [udp sum ok]  1* 1/0/0
CFB_NTS_S28.mylan.com. A x.y.25.128 (63)
17:48:07.937127 IP (tos 0x0, ttl 126, id 26329, len 78) x.z.77.56.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:07.938477 IP (tos 0x0, ttl 126, id 1981, len 64) x.z.126.69 > x.y.25.21: icmp 44: echo request seq 0
17:48:07.968252 IP (tos 0x0, ttl 126, id 28789, len 64) x.w.40.85 > x.y.25.21: icmp 44: echo request seq 0
17:48:08.012162 IP (tos 0x0, ttl 126, id 2237, len 71) x.z.126.69.4485 > x.y.25.21.: [udp sum ok]  1+ PTR? 21.25.y.x.in-addr.arpa.
(43)
17:48:08.041651 IP (tos 0x0, ttl 125, id 1297, len 75) x.v.109.59.1220 > x.y.25.21.: [udp sum ok]  1+ A? gimc-online.mylan.com. (47)
17:48:08.058034 IP (tos 0x0, ttl 126, id 29045, len 71) x.w.40.85.4902 > x.y.25.21.: [udp sum ok]  1+ PTR? 21.25.y.x.in-addr.arpa.
(43)
17:48:08.072780 IP (tos 0x0, ttl 126, id 2493, len 78) x.z.126.69.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:08.141756 IP (tos 0x0, ttl 126, id 29301, len 78) x.w.40.85.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:08.176366 IP (tos 0x0, ttl 126, id 54992, len 71) x.z.113.75.3329 > x.y.25.21.: [udp sum ok]  1+ PTR? 21.25.y.x.in-addr.arpa.
(43)
17:48:08.217977 IP (tos 0x0, ttl 126, id 55248, len 78) x.z.113.75.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:08.422890 IP (tos 0x0, ttl 126, id 2749, len 64) x.z.126.69 > x.y.25.21: icmp 44: echo request seq 0
17:48:08.487112 IP (tos 0x0, ttl 126, id 3005, len 71) x.z.126.69.4486 > x.y.25.21.: [udp sum ok]  1+ PTR? 21.25.y.x.in-addr.arpa.
(43)
17:48:08.517910 IP (tos 0x0, ttl 126, id 29557, len 64) x.w.40.85 > x.y.25.21: icmp 44: echo request seq 0
17:48:08.547772 IP (tos 0x0, ttl 126, id 3261, len 78) x.z.126.69.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:08.607572 IP (tos 0x0, ttl 126, id 29813, len 71) x.w.40.85.4903 > x.y.25.21.: [udp sum ok]  1+ PTR? 21.25.y.x.in-addr.arpa.
(43)
17:48:08.666225 IP (tos 0x0, ttl 126, id 33808, len 72) x.w.90.61.1118 > x.y.25.21.: [udp sum ok]  1+ A? FAUSS022.mylan.com. (44)
17:48:08.691454 IP (tos 0x0, ttl 126, id 30069, len 78) x.w.40.85.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:08.897388 IP (tos 0x0, ttl 126, id 3517, len 64) x.z.126.69 > x.y.25.21: icmp 44: echo request seq 0
17:48:08.962556 IP (tos 0x0, ttl 126, id 3773, len 71) x.z.126.69.4487 > x.y.25.21.: [udp sum ok]  1+ PTR? 21.25.y.x.in-addr.arpa.
(43)
17:48:08.966978 IP (tos 0x0, ttl 126, id 26585, len 64) x.z.77.56 > x.y.25.21: icmp 44: echo request seq 0
17:48:09.002366 IP (tos 0x0, ttl 123, id 52927, len 69) x.z.6.3.3184 > x.y.25.21.: [udp sum ok]  3354+ A? www.anotherlan.com. (41)
17:48:09.023340 IP (tos 0x0, ttl 126, id 4029, len 78) x.z.126.69.137 > x.y.25.21.137: [udp sum ok] udp 50
17:48:09.026329 IP (tos 0x0, ttl 125, id 3089, len 74) x.v.109.59.1222 > x.y.25.21.: [udp sum ok]  2+ A? crl-online.mylan.com. (46)
```

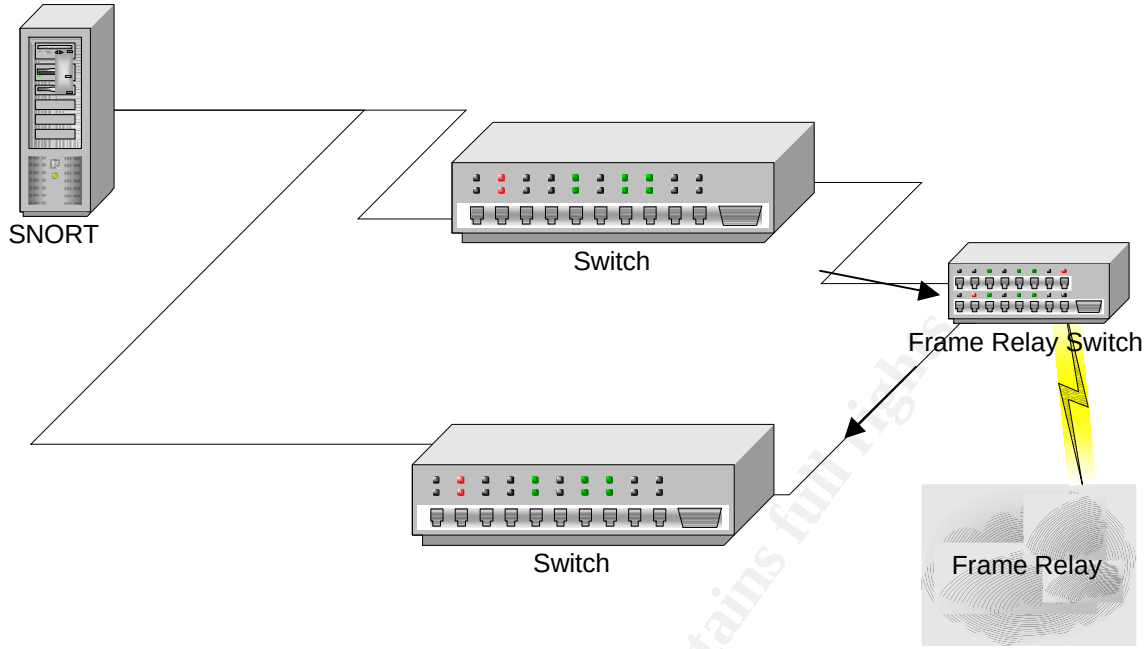
Very small sample of the mystery traffic that was detected.

1. Source of trace:

This trace came from the operational network of a Fortune 500 company with multiple lines of business and offices in 40 different states. The traffic was detected, monitored, and intercepted in the corporate headquarters located in Texas. The suspect traffic originated in Maine, Connecticut, Rhode Island, and Virginia.



The Wide Area Network is connected via Frame Relay from all remote locations to the main corporate offices in Texas. The Frame Relay lines are connected through a series of Cisco Switches to form the Frame Relay backbone. One of the interesting factors to note in this configuration is that the incoming packets, from the remote location to the corporate office, was detected on one switch, while the reply traffic was seen on a second switch. Both were being monitored by Snort running on the same server machine with two separate network interface cards (sensors).



While the overall network of this company has over 6000 workstations spread across 40 states, all of the suspect traffic was coming from 18 different workstations in 6 East Coast cities.

It should be noted that the IP addresses depicted in this detect as well as the actual cities have been changed to protect the security integrity of the actual network. The remaining information contained in the packets is authentic.

2. Detect was generated by:

Part of the Information Security infrastructure of this organization includes Snort IDS version 1.9 running on the Linux OS. The particular system that was used to detect the anomalous traffic was running two instances of Snort on different network interface cards set to promiscuous mode and no IP address set. The traffic was saved to a binary file and transferred to a Windows system in order to process it for this paper.

The traffic was analyzed by version 1.9 of Snort for Windows with the Whitehats signature file[46]. No alerts resulted.

3. Probability the source address was spoofed:

There is no chance that the source address of this traffic was spoofed. It originated on the internal network structure. There are infrastructure security methods in place that would make spoofing all but impossible.

[46] WhiteHats.Com, "ArachNIDS Database", URL: http://www.whitehats.com/ids/index.html (March 7, 2003)

4. Description of attack:

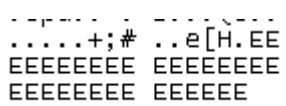
This is a particularly exciting aspect of this paper as the investigation of the "attack" is underway as the paper is being written. We have found no external sources to describe what we are seeing on the network as an attack. However, there are several factors that indicate that this may be some sort of malicious activity.

The Pattern – In reviewing the traffic, a certain pattern quickly emerges that tells a big part of the story. In this case, we will isolate our investigation to a single workstation at IP address "x.w.40.85". The traffic is arriving in a series of three packets in very short intervals. The three packets are as shown below from Ethereal:

0.045364	9.40.85	.25.21	ICMP	Echo (ping) request
0.135146	9.40.85	.25.21	DNS	Standard query PTR 21.25.20.168 in-addr.arpa
0.218868	9.40.85	.25.21	NBNS	Name query NBSTAT *<00><00><00><00><00><00><00><00>

There are a few things that are noteworthy in this trace. First, the timing of the three packets. They are all within a fraction of a second of each other. Second, they are all to the same DNS server. Third, the packets that are coming across are always in this same order. Fourth the packets are streaming constantly with three full sets or nine packets arriving per second.

Looking at the individual packets. The first to arrive appears to be a standard ICMP Echo Request. It is a type 8 ping request, code 0. One of the possible identifying features of this ICMP packet is the payload looks like the image below:



A search of the Internet indicates that this particular type of packet may be the result of the Grims ping tool.[47] James C Slora, Jr. reports seeing similar types of traffic in an online discussion where he expresses the opinion "The ping portion looks like a Grims Ping (http://grimsping.cjb.net/) scan I think - ID:1, EEEEEEEE... data." [48]

The second packet to arrive is a DNS query. The interesting aspects of this query on port is that it is querying the same DNS server as the ping and it is a reverse DNS lookup on the DNS servers IP address.

The third packet to arrive is a netbios name query. This appears to be a standard netbios name query for type nbstat and class inet as shown in the Ethereal screenshot below:

*<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>: type NBSTAT, class inet

Nothing out of the ordinary was found in the packet. According to Slora, the Grims Ping can produce NetBios queries identical to the queries found in our trace.[49]

---

[47] GrimsPing, "Programming for the Free World", URL: http://grimsping.cjb.net/ (March 7, 2003)
[48] Slora, James C. Jr. "Grims Ping Targeted Recon Probe?", URL: http://cert.uni-stuttgart.de/archive/intrusions/2002/08/msg00292.html (March 7, 2003)
[49] ibid

Netbios Query found with the Grims Ping.

```
08/28/02-12:15:38.292012 a.b.198.176:1025 -> host36:137 UDP
TTL:112 TOS:0x0 ID:39 IpLen:20 DgmLen:78
Len: 58
0x0000: 00 xx xx xx xx xx 00 xx xx xx xx xx xx 00 45 00  ..............E.
0x0010: 00 4E 14 DB 00 00 70 11 D6 AB 50 0E C6 B0 xx xx  .N....p...P.....
0x0020: xx xx 04 01 00 89 00 3A D7 77 86 16 00 10 00 01  .......:.w......
0x0030: 00 00 00 00 00 00 20 43 4B 41 41 41 41 41 41 41  ...... CKAAAAAAA
0x0040: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
0x0050: 41 41 41 41 41 41 41 00 00 21 00 01              AAAAAAA..!..
```

5. Attack mechanism:

Is this an attack? What type of attack is this? Could it be innocuous traffic or the product of malfunctioning software? These are critical questions that must be answered. The first step in discovering the answers will be to state the facts as they are known at this point:

1. There are 18 systems that are streaming packets in a particular pattern of ICMP Echo Request to the DNS Server, DNS Reverse Lookup of the DNS Server, and Netbios Name Query of the DNS Server.
2. The DNS server that is targeted by the traffic is the default DNS server for all of the systems affected.
3. All 18 of the systems are Windows 98 machines.
4. All 18 of the systems are in the same geographical area of the country with a large percentage concentrated in one office.
5. They are all members of a very large enterprise wide network. Of 6000 systems on the network, only these 18 are displaying the unusual behavior.
6. The Ping packets and the Netbios queries resemble the Grims Ping packets. This program should never appear on any of the corporate systems for any legitimate reason.
7. No unusual traffic is being returned as a result of this traffic.

The best hypothesis is that this is a new worm virus that is designed as a denial of service against an internal network by incapacitating the network DNS server with excessive traffic. This hypothesis will be proven or disproved once a forensic analysis is completed on one or more of the suspect systems.

6. Correlations:

This appears to be the first time this exact pattern of traffic has been seen. James C Slora, Jr. reports seeing similar types of traffic, but with the inclusion of SQL (Port 1433) and web (Port 80) traffic scans[50]. This traffic was also a scan of

[50] Slora, James C. Jr. "Grims Ping Targeted Recon Probe?", URL: http://cert.uni-stuttgart.de/archive/intrusions/2002/08/msg00292.html (March 7, 2003)

a network and not a probe or query against a single system. The similarities are in the ICMP packets and Netbios traffic.

7. Evidence of active targeting:

It is without doubt that the traffic seen was targeted at the internal DNS server. The traffic from all 18 suspect systems was directed to this single point.

8. Severity:

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

Criticality – According to the Criticality Severity Chart[51], the DNS server is worth 5 points.

Lethality – If only 18 systems are affected by this worm, then the Lethality factor would be 1 point as the attack is very unlikely to succeed. However, if this were to be a fast spreading worm and spread to hundreds or thousands of computer systems, that value would be raised to 4 as a total lockout by denial of service. At this point, it does not seem clear that the latter possibility exists, so the value will be calculated as 1 point.[52]

System Countermeasures – At this point in time, there are no system countermeasures in place that can thwart this activity.[53]

Network Countermeasures – At this point in time, there are no network countermeasures in place that can thwart this activity.[54]

Severity = (5 + 1) – (0 + 0) = 6

9. Defensive Recommendation:

Based on the severity analysis in section 8 above, the defensive actions that must be taken are in the form of system and network countermeasures. System countermeasures would include the ability to detect and stop the program(s) that are causing the traffic to be sent. The easiest way to do that would be to develop an appropriate anti-virus signature file to detect the worm, if that is what the

[51] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-8 (Severity:Criticality Chart).
[52] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-9 (Severity:Lethality Chart).
[53] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-14 (Severity:System Countermeasures Chart).
[54] SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-15 (Severity:Network Countermeasures Chart).

forensic investigated concludes is the source. Network countermeasures would be in the form of detection systems to notify maintenance personnel that the problem has re-surfaced.

10. Multiple choice test question:

```
17:48:07.937127 IP (tos 0x0, ttl 126, id 26329, len 78) x.z.77.56.137 > x.y.25.21.137: [udp sum ok] udp 50
                4500 004e 66d9 0000 7e11 174d a817 4d38
                ac14 1915 0089 0089 003a a68f 5b26 0000
                0001 0000 0000 0000 2043 4b41 4141 4141
                4141 4141 4141 4141 4141 4141 4141 4141
                4141 4141 4141 4141 4100 0021 0001
17:48:07.938477 IP (tos 0x0, ttl 126, id 1981, len 64) x.z.126.69 > x.y.25.21: icmp 44: echo request seq 0
                4500 0040 07bd 0000 7e01 457a a817 7e45
                ac14 1915 0800 a593 f182 0000 6794 a500
                4545 4545 4545 4545 4545 4545 4545 4545
                4545 4545 4545 4545 4545 4545 4545 4545
17:48:07.968252 IP (tos 0x0, ttl 126, id 28789, len 64) x.w.40.85 > x.y.25.21: icmp 44: echo request seq 0
                4500 0040 7075 0000 7e01 32b6 a813 2855
                ac14 1915 0800 bb2b 3b23 0000 655b 4801
                4545 4545 4545 4545 4545 4545 4545 4545
                4545 4545 4545 4545 4545 4545 4545 4545
17:48:08.012162 IP (tos 0x0, ttl 126, id 2237, len 71) x.z.126.69.4485 > x.y.25.21.: [udp sum ok]  1+ PTR?
21.25.y.x.in-addr.arpa. (43)
                4500 0047 08bd 0000 7e11 4463 a817 7e45
                ac14 1915 1185 0035 0033 b65c 0001 0100
                0001 0000 0000 0000 0232 3102 3235 0232
                3003 3137 3207 696e 2d61 6464 7204 6172
                7061 0000 0c00 01
```

In reviewing the trace above, what about the traffic seems to be unusual?

A. ICMP traffic should always precede port 137 traffic.
B. The data of ICMP traffic is unusual.
C. The data of the Netbios traffic is unusual.
D. The DNS traffic is in-addr.arpa traffic, which should never happen normally.

Answer: B

Answer A is nonsense and put there only as a distracter.
Answer B is correct. Data in ICMP is not normally EEEEE, rather it is abcd. Etc.
Answer C is not correct. This is normal looking Netbios traffic.
Answer D is not correct. In-addr.arpa traffic is normal.

Conclusion to this Detect

After this detect was written for this paper, further investigation revealed that the traffic was being caused by an authorized program that network administrators use to inventory software on all systems belonging to the company. While using the software to determine if any new executable programs had been installed on the "infected" systems, it was determined that each and every system that was sending this abnormal traffic was missing from the software inventory listing.

Further investigation revealed that the stopping of the inventory program processes on the system caused cessation of the traffic and restarting the program restarted the traffic.

It is unknown why the application caused a traffic pattern such as this or why the ping packet was so similar to Grims Ping. The issue has been taken up with the software vendor.

Assignment Part 3 – Analyze This

Executive Summary

An extensive analysis of intrusion detection log files provided by the University was accomplished in order to determine the extent of Information Security problems that are present on the University network. The log files were taken from the Snort Intrusion Detection systems and reflect Alerts, Scans, and Out of Spec log files over a five-day period between March 4, 2003 and March 8, 2003. An emphasis was placed on determining which systems, if any, were likely compromised. The full report follows this executive summary.

It was determined during the course of the audit that numerous University systems appear to be compromised or have configuration problems as determined by malicious, hostile, or anomalous traffic that appeared in the logs. It also became apparent that the security policies that have been in place up to this point have been totally ineffective in safeguarding University information system assets.

The recommendations to the governing body of the University are as follows:

1. A complete review and revision of network security policies needs to be undertaken with increased security as the objective. In addition, the legal liabilities that may be incurred as the result of ineffective security should be investigated and included in the security policies.
2. Implementation of the security policies should be scheduled to take place as soon as feasible. Continued operations of the network with current security constitutes a hazard to the systems and a liability to the University.
3. Security training should be included in the ongoing education of IT personnel. This is critical as the complexity of information security continues to increase and the ability required of hackers lessens.

The full audit following this executive summary includes a list of the alerts that were received during the period and a short explanation of each, an analysis of the computer relationships within the detects, and a link-graph showing relationships of some of the traffic.

List of Files Analyzed

All of the following files were downloaded from http://www.incidents.org/logs/ [55] for University's Security Audit. They represent log files for five consecutive days between March 4th and March 8th, 2003.

	Scan Logs	Alert Logs	Out of Spec Logs
March 4, 2003	scans.030304.gz	alert.030304.gz	OOS_Report_2003_03_05_29589.txt
March 5, 2003	scans.030305.gz	alert.030305.gz	OOS_Report_2003_03_06_20831.txt
March 6, 2003	scans.030306.gz	alert.030306.gz	OOS_Report_2003_03_07_19899.txt
March 7, 2003	scans.030307.gz	alert.030307.gz	OOS_Report_2003_03_08_17114.txt
March 8, 2003	scans.030308.gz	alert.030308.gz	OOS_Report_2003_03_09_9640.txt

There are three different types of log files used in this audit. Each of the log types are used for a specific purpose.

Scan Logs show port scans that are taking place on the network. According to Lawrence Teo, "A port scan is a method used by intruders to discover the services running on a target machine."[56] Knowledge of port scans is important as they are used as a means of reconnaissance, that is getting to know the target before an attack. The port scans available in the log files are in the following format:

```
Date/Time        Source IP    Port      Dest. IP        Port   Flags   .
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:8 SYN ******S*
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:443 SYN ******S*
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:1458 SYN ******S*
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:27 SYN ******S*
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:1467 SYN ******S*
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:372 SYN ******S*
Mar  4 14:33:00 68.50.16.210:35421 -> 130.85.195.17:1450 SYN ******S*
Mar  4 14:33:01 68.50.16.210:35421 -> 130.85.195.17:7070 SYN ******S*
```

In the example above, you see the same Source IP Address and port that is port scanning a single destination address on numerous different ports. The S indicates a SYN or Synchronization flag meaning a TCP connection to that port is being requested. Any response to the SYN flag indicates a possible available service. This information can then be used to attack that service.

Alert Logs are used to record alerts generated by the Snort ruleset. The logs are in the following format:

---

[55] Internet Storm Center, "Log Files", URL: http://www.incidents.org/logs/ (Mar 11, 2003)

[56] Lawrence Teo, "Network Probes Explained: Understanding Port Scans and Ping Sweeps" Linux Journal Online March 7, 2003, URL: http://www.linuxjournal.com/article.php?sid=4234 (March 14, 2003)

Source IP Port Dest IP Port
```
MY.NET.105.204:3140 -> 194.87.6.77:3366
```
Date/Time Alert Name / Reference
```
03/04-12:33:01.451093  [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
```

Finally, the OOS logs are the Out of Spec logs. This means that the packets that do not meet the TCPIP RFC for flags. They have unusual or illegal combinations of flags set. The OOS logs are in the following format:

```
03/04-09:01:52.7838 66.140.25.156:56436 -> MY.NET.60.16:8001
TCP TTL:44 TOS:0x0 ID:57139 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x1E00ED2  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 820564748 0 NOP WS: 0
```

Note the log information for the OOS logs is more extensive than the Scan and Alert logs. More information is required to determine what is happening. In most, if not all cases, the packets seen in the OOS logs are also going to be found in the Alert Logs.

Computer Relationships

We can determine a basic network topology by counting the port numbers in the IP traffic on the network and then relating them back to the IP addresses to determine the function of the individual devices.

Looking at the number of individual IP addresses in the “MY.NET” range, almost 37,000, combined with the 3rd octet ranges from 1 through 253, it is apparent that we are looking at a very large , Class B, network.

Beginning with the DNS Servers by looking for port 53 traffic:

Port 53 Traffic to MY.NET Servers - Alert Logs

MY.NET Server	Count
MY.NET.1.200	42
MY.NET.1.3	2
MY.NET.1.4	2
MY.NET.1.5	1

There are no logs where Port 53 traffic from MY.NET servers was present.

Port 53 Traffic from MY.NET Servers - the Scan Logs (Count > 100)

MY.NET Server	Destination IP	Count
MY.NET.1.200	206.21.107.142	101
MY.NET.1.200	206.117.30.102	102
MY.NET.1.200	131.118.254.35	103
MY.NET.1.200	198.41.0.4	105
MY.NET.1.200	152.163.159.232	106

MY.NET.1.200	63.164.70.3	107
MY.NET.1.200	149.174.54.3	108
MY.NET.1.200	193.0.14.129	109
MY.NET.1.200	63.175.146.7	117
MY.NET.1.200	64.81.84.165	119
MY.NET.1.200	216.82.120.196	122
MY.NET.1.200	194.109.6.154	123
MY.NET.1.200	205.231.29.243	124
MY.NET.1.200	66.45.120.62	127
MY.NET.1.200	193.190.198.10	128
MY.NET.1.200	209.98.98.115	132
MY.NET.1.200	64.142.16.36	133
MY.NET.1.200	128.194.254.5	137
MY.NET.1.200	192.31.80.30	137
MY.NET.1.200	192.5.6.36	139
MY.NET.1.200	130.94.6.10	140
MY.NET.1.200	66.197.162.2	143
MY.NET.1.200	64.0.0.134	144
MY.NET.1.200	64.58.77.85	145
MY.NET.1.200	193.0.0.193	150
MY.NET.1.200	63.250.206.138	153
MY.NET.1.200	130.94.244.139	159
MY.NET.1.200	205.231.29.245	163
MY.NET.1.200	128.8.10.90	171
MY.NET.1.200	205.231.29.244	187
MY.NET.1.200	62.242.234.100	193
MY.NET.1.200	192.41.162.30	205
MY.NET.1.200	131.118.254.35	209
MY.NET.1.200	140.186.128.222	223
MY.NET.1.200	192.41.162.30	234
MY.NET.1.200	212.100.230.160	242
MY.NET.1.200	131.118.254.33	248
MY.NET.1.200	212.242.41.170	275
MY.NET.1.200	63.164.70.2	287
MY.NET.1.200	66.33.98.17	309
MY.NET.1.200	128.63.2.	349
MY.NET.1.200	192.52.178.30	359
MY.NET.1.200	194.109.6.152	364
MY.NET.1.200	205.231.29.245	378
MY.NET.1.200	205.231.29.243	394
MY.NET.1.200	192.5.6.30	423
MY.NET.1.200	131.211.28.48	455
MY.NET.1.200	194.109.6.154	473
MY.NET.1.200	194.109.6.1	486
MY.NET.1.200	64.142.16.36	516
MY.NET.1.200	192.26.92.30	653

MY.NET.1.200	205.231.29.244	591
MY.NET.1.200	209.98.98.115	598
MY.NET.1.200	131.118.254.34	621
MY.NET.1.200	130.94.6.10	640
MY.NET.1.200	192.26.92.30	1544

The 1544 count for 192.26.92.30 requires further investigation. The registry information for the IP address from ARIN[57] is shown below. This could be a "root" DNS server.

```
OrgName:    VeriSign Global Registry Services
OrgID:      VGRS
Address:    21345 Ridgetop Circle
City:       Dulles
StateProv:  VA
PostalCode: 20166
Country:    US
NetRange:   192.26.92.0 - 192.26.92.255
CIDR:       192.26.92.0/24
NetName:    VGRSGTLD-3
NetHandle:  NET-192-26-92-0-1
Parent:     NET-192-0-0-0-0
NetType:    Direct Assignment
NameServer: L2.NSTLD.COM
NameServer: D2.NSTLD.COM
NameServer: E2.NSTLD.COM
NameServer: C2.NSTLD.COM
Comment:
RegDate:    2000-11-30
Updated:    2001-03-20
TechHandle: ZV22-ARIN
TechName:   VeriSign Global Registry Services
TechPhone:  +1-703-318-6444
TechEmail:  nstld@verisign-grs.com
```

By running an nslookup on the IP address, it is confirmed by the name that it is a root DNS server.

```
C:\>nslookup 192.26.92.30
Server:  ns1.austin.rr.com
Address:  24.93.35.62

Name:    c.gtld-servers.net
Address:  192.26.92.30
```

Based upon the accumulated evidence MY.NET.1.200 is the primary DNS server for the network. This is deduced by the fact that there are so many queries to port 53 to so many different IP addresses. The sequence of the traffic indicates that a client is requesting the resolution of a domain name from the

[57] "WhoIs", URL: http://www.arin.net/ (March 22, 2003)

DNS server (MY.NET.1.200). The DNS server is querying the root DNS server (192.26.92.30) first to determine the Top Level DNS and then queries that DNS server for the IP address, down through DNS servers until the address required by the client is resolved. Alerts from the Alert log indicate that the DNS server also responds to queries from outside the internal network.

Next, we determine the Web Servers by reviewing Port 80 and 443 traffic:

Port 80/443 Traffic to MY.NET Servers - Alert Logs (Count > 100)

MY.NET Server	Port	Count
MY.NET.24.34	80	228
MY.NET.218.26	80	247
MY.NET.30.4	80	669
MY.NET.100.165	80	10582

Port 80/443 Traffic to MY.NET Servers - Scan Logs (Count >20)

MY.NET Server	Source IP	Count
MY.NET.100.165	147.91.173.31	29

Based upon the traffic shown above, there is little doubt that "MY.NET.100.165" is a primary Web Server. Based on the numbers of alerts, the other three addresses shown in the Alert Logs are probably Web Servers as well.

Reviewing Port 137, 138, and 139 traffic, most of the network appears to be primarily Windows based. There are over 36,000 systems that have alerts on these three ports. The servers in the following tables would appear, by sheer volume of alerts, to be primary file servers.

Port 137/138/139 Traffic to MY.NET Servers – Alert Logs (Count>100)

MY.NET Server	Port	Count
MY.NET.235.222	137	103
MY.NET.195.3	137	106
MY.NET.204.102	137	116
MY.NET.6.55	137	117
MY.NET.6.7	137	128
MY.NET.218.26	137	129
MY.NET.236.230	137	131
MY.NET.249.94	137	131
MY.NET.204.30	137	140
MY.NET.202.214	137	160
MY.NET.208.174	137	175
MY.NET.219.6	137	180
MY.NET.29.3	137	187
MY.NET.210.238	137	188

MY.NET.249.134	137	217
MY.NET.24.44	137	315
MY.NET.12.2	137	425
MY.NET.29.11	137	478
MY.NET.194.13	137	710
MY.NET.24.34	137	887

Next, a check of traffic for Port 21, FTP Services, indicates there are at least three FTP servers in the network.

Port 21 Traffic to MY.NET Servers (Count > 10)

MY.NET Servers	Port	Count
MY.NET.24.27	21	14
MY.NET.100.165	21	244
MY.NET.211.98	21	1413

Port 25 was checked in order to detect the Simple Mail Transport Protocol servers:

Port 25 Traffic to MY.NET Servers (Count > 100)

MY.NET Servers	Port	Count
MY.NET.24.21	25	177
MY.NET.145.9	25	235
MY.NET.6.47	25	364

The count of the port 25 traffic is fairly low, showing that all of the traffic is not present in the logs. Certainly, a network of this size would have significant email traffic.

In summary, the University network is a large, Class B, network made up of over 36,00 primarily Microsoft Windows® systems. Services found include email services, web services, SMB services, ftp services, and DNS services.

Detect List by Frequency of Occurrence (Occurrence > 500)

Alert	Count
SMB Name Wildcard	**109775**

Snort Rule: misc-lib:alert udp any any -> $HOME_NET 137 (msg:"SMB Name Wildcard"; content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";)

"Translating the content string back out of the strange Netbios encoding (see RFC 1001 page 25), it becomes the name "*". This special name is used for any broadcast name service requests (RFC 1001 page 57).[58]

[58] "IDS: Source Port of Samba Scans", Daniel Swan, URL: http://www.shmoo.com/mail/ids/mar00/msg00065.shtml (March 23, 2003)

Basically, this is standard NetBios name traffic that is broadcast to the network. According to Robert Graham, "Incoming connections to this port are trying to reach NetBIOS/SMB, the protocols used for Windows "File and Print Sharing" as well as SAMBA. People sharing their hard disks on this port are probably the most common vulnerability on the Internet."[59]

The thing that is most disconcerting about these alerts is that the Netbios traffic does not originate from within the internal environment, but are all external. This means the firewall/router systems is allowing this type of traffic through the firewall.

Alert	Count
Watchlist 000220 IL-ISDNNET-990517	**37635**

"This is a custom alert signature based on previous suspicious activity for the source netblock. Watchlists are created to trigger on follow-up traffic and must be investigated."[60] In this case, the watchlist is monitoring traffic from the "212.179" network. Registration information on this network was checked through the RIPE whois database with the results showing in the "Five External Registration Information Examples" section following this section.

Alert	Count
TCP SRC and DST outside network	**22183**

This alert indicates that the Source and Destination IP address of the traffic are both outside of the network. This is highly unusual and could only occur if the network were misconfigured in some fashion or if the source IP address were spoofed and the traffic actually originated within the network. It is more likely that the latter is the case. This would require some additional investigation with logs that are more extensive than what are available at this time.

Alert	Count
spp_http_decode: IIS Unicode attack detected	**20792**

This is basically the use of Unicode characters to bypass input validation for IIS servers. This would allow hostile action such as directory traversal. This attack is explained in detail earlier in this paper in the section titled "Detect #2 – Directory Traversal".

Alert	Count
High port 65535 tcp - possible Red Worm - traffic	**12994**

[59] "Netbios File and Print Sharing", Graham, Robert, URL: http://www.robertgraham.com/pubs/firewall-seen.html#port137 (March 23, 2003)
[60] "GCIA Practical", Coyle, Brian URL: http://www.linuxwidows.com/mirror/bucket/Brian_Coyle_GCIA.pdf (March 23, 2003)

High port 65535 udp - possible Red Worm - traffic	**2396**

Also known as the Adore Worm, Red Worm is a Trojan that when activated, makes an outbound connection on TCP or UDP port 65535.[61] This indicates that connection may have occurred. The numbers of connections would indicate that malicious activity is under way.

Alert	Count
CS WEBSERVER - external web traffic	**10452**

This alert simply states that external web traffic has been detected coming into the network.[62]

Alert	Count
Russia Dynamo - SANS Flash 28-jul-00	**5338**

This appears to be another Trojan that is sending data outbound to addresses in Russia. The destination IP address is 194.87.6.77. Registration information on this address was checked through the RIPE whois database with the results showing in the "Five External Registration Information Examples" section following this section. It confirms that the address is indeed in Moscow, Russia.

Alert	Count
Incomplete Packet Fragments Discarded	**4772**

This alert indicates that packet fragments were detected, but the remaining fragments were not found. The cause of this anomaly could be a misconfigured network or it may indicate hostile traffic.[63]

Alert	Count
TFTP - Internal TCP connection to external tftp server	**3805**
TFTP - External UDP connection to internal tftp se	**2110**

This alert indicates that an internal resource has connected to an external tftp server. This is a concern as tftp is often used for unattended software and file downloads. This would be an ideal way for a hacker to gain access to his/her hacker tools without attracting a lot of attention. When we break down the traffic a bit more, there are some interesting patterns that emerge.

---

[61] "Adore Worm Version 0.8 – April 12, 2001", URL: http://www.sans.org/y2k/adore.htm (March 26, 2003)
[62] "Intrusion Detection Indepth Version 3.0", Baird, Scott, URL: http://www.giac.org/practical/Scott_Baird_GCIA.doc (March 26, 2003)
[63] "Intrusion Detection In Depth GCIA Practical Assignment Version 3.0 (revised August 13, 2001)", Jenkins, David, URL: http://www.giac.org/practical/David_Jenkins_GCIA.doc (March 26, 2003)

Internal Address	TFTP Server/Port		Count
MY.NET.226.22	202.156.52.27	69	10
MY.NET.242.42	205.188.1.27	69	4
MY.NET.242.42	205.188.6.52	69	20
MY.NET.237.90	205.188.6.53	69	8
MY.NET.242.42	205.188.6.53	69	36
MY.NET.236.126	205.188.6.54	69	36
MY.NET.206.130	205.188.7.59	69	30
MY.NET.203.2	63.231.14.237	69	572
MY.NET.242.42	64.12.161.153	69	4
MY.NET.242.42	64.12.161.185	69	2
MY.NET.223.114	64.12.25.1	69	50
MY.NET.223.114	64.12.25.150	69	239
MY.NET.242.42	64.12.25.240	69	2
MY.NET.206.130	64.12.26.144	69	123
MY.NET.206.130	64.12.26.145	69	71
MY.NET.206.130	64.12.26.23	69	1
MY.NET.242.42	64.12.26.249	69	17
MY.NET.242.42	64.12.26.250	69	4
MY.NET.242.42	64.12.26.251	69	55
MY.NET.242.42	64.12.26.45	69	3
MY.NET.206.130	64.12.27.84	69	38
MY.NET.206.130	64.12.27.85	69	8
MY.NET.206.130	64.12.27.86	69	10
MY.NET.206.130	64.12.28.52	69	45
MY.NET.206.130	64.12.28.55	69	36
MY.NET.223.114	64.12.29.1	69	3
MY.NET.206.130	64.12.29.64	69	1
MY.NET.236.126	64.12.30.136	69	70
MY.NET.237.90	64.12.30.136	69	13
MY.NET.242.42	64.12.30.136	69	228
MY.NET.242.42	64.12.30.224	69	62

An initial investigation reveals that the 205.188 network and the 64.12 network are both America Online. It is unlikely that there are that many active tftp servers on America Online or that they would be setup to be used in this manner. It is more likely that there is an America Online service accessible from the Internet that uses port 69. This caused the false positives seen here. There are two other address ranges, however, that are cause for concern and are highlighted above.

Alert	Count
connect to 515 from outside	**3236**

Port 515 is a well-known port used for printer connections.[64] It would be highly unusual for printers to be available to users and systems outside the internal network, yet it appears by this alert that it is occurring. This could be false positive traffic if port 515 was being used by some other program other than printers. This is highly suspicious traffic and should be investigated further.

Alert	Count
External RPC call	**2807**

The destination port for the traffic that triggered this alert was port 111. This is a well-known port for SUN Remote Procedure Call[65] According to Silotto "Portmapper is a service that keeps a directory of all the RPC services running on that machine. Here, we can see some queries to it. Probably the attacker was trying to find out what services are running and in what port."[66]

Alert	Count
spp_http_decode: CGI Null Byte attack detected	**2778**

Basically, if the http decoding routine finds a %00 in an http request, it will alert with this message.[67] This alert appears to have originated as part of Snort's HTTP pre-processor.

Alert	Count
MY.NET.30.4 activity	**2520**

This alert would apparently be setup to alert when the server at MY.NET.30.4 were accessed. The activity on this server is extensive and all alerts have the IP address as the destination address.

Alert	Count
Null scan!	**2024**

The best description of a Null scan that I found was from Neil Warner on the Honeynet project. Neil describes null scan traffic he saw as: "A NULL scan is when no flags are set. A NULL scan attack is looking for a RST from the target when the port is closed or no response which might mean the port is open. At packet #148067 the attacker sends a NULL scan to port 80 of the target. The target did not respond which indicates that port 80 is open on the target

[64] IANA, "Well Known Ports Database", (February 26, 2003) URL : http://www.iana.org/assignments/port-numbers (March 1, 2003)
[65] ibid.
[66] **"**GIAC Intrusion Detection Curriculum", Silotto, Claudio R. G. , URL: http://www.giac.org/practical/Claudio_Silotto_GCIA.doc (March 27, 2003)
[67] "Neohapsis Archives", from Joe Stewart. URL: http://archives.neohapsis.com/archives/snort/2000-11/0244.html (March 27, 2003)

system."[68] Besides the fact that the traffic seen on the University's network were Null Scans, a preponderance of them were directed to a destination port of "0" and many had source ports of "0". According to an anonymous contributor to the Honeynet Project, this may be the work of Hping2.[69]

Alert	Count
Watchlist 000222 NET-NCFC	**1799**

This is another watchlist, alerts designed to identify traffic from a specific source. In this case, it appears to be the "159.226" network, which is registered to "The Computer Network Center Chinese Academy of Sciences".[70] Registration information on this address was checked through the RIPE whois database with the results showing in the "Five External Registration Information Examples" section following this section.

Alert	Count
FTP DoS ftpd globbing	**1413**

As the name implies, this attack is an FTP denial of service attack against the ftpd service. It works by using a process termed "globbing". According to the NCSA archives, "*Globbing* is the process by which the csh handles wildcards in file names"[71] The wildcard characters are used to "confuse" the OS and cause it to crash.

Alert	Count
Possible Trojan server activity	**1388**

This alert indicates a host may be infected with a Trojan. Trojans can have several different functions, but one of them is to gather information from an infected machine. Therefore, alerts of this nature that have the local network as the source address are of special concern.

SrcIP	SrcPort	DstIP	DstPort	Alert
MY.NET.179.77	80	168.171.25.60	27374	Possible trojan server activity
MY.NET.208.106	1214	203.177.33.61	27374	Possible trojan server activity
MY.NET.208.174	4662	80.135.244.50	27374	Possible trojan server activity
MY.NET.208.26	2861	66.98.36.118	27374	Possible trojan server activity
MY.NET.210.238	3162	81.17.193.95	27374	Possible trojan server activity
MY.NET.218.26	80	217.157.187.165	27374	Possible trojan server activity
MY.NET.220.66	1214	12.221.200.27	27374	Possible trojan server activity
MY.NET.221.102	27374	68.55.4.114	8152	Possible trojan server activity

[68] "Scan 23 –South Florida Honeynet Project",Warner, Neil, URL: http://project.honeynet.org/scans/scan23/sol/Neil.html (March 29, 2003)
[69] "SCAN OF THE WEEK #1 - 28 May - 3 June", The Honeynet Project, URL: http://project.honeynet.org/scans/arch/scan1.txt (March 29, 2003)
[70] "ARIN Whois Database", URL: http://ws.arin.net/cgi-bin/whois.pl (March 27, 2003)
[71] "Globbing", NCSA Archives, URL: http://archive.ncsa.uiuc.edu/General/Training/InterUnix/csh/glob.html (March 29, 2003)

MY.NET.223.182	80	12.107.16.28	27374	Possible trojan server activity
MY.NET.238.106	2570	208.138.28.114	27374	Possible trojan server activity
MY.NET.24.47	1431	132.189.76.10	27374	Possible trojan server activity
MY.NET.240.226	3431	200.184.127.60	27374	Possible trojan server activity
MY.NET.249.134	1214	168.103.147.165	27374	Possible trojan server activity
MY.NET.249.134	1214	65.41.84.184	27374	Possible trojan server activity
MY.NET.250.126	2286	208.42.95.244	27374	Possible trojan server activity
MY.NET.250.22	1853	128.11.61.146	27374	Possible trojan server activity
MY.NET.6.47	25	129.250.156.247	27374	Possible trojan server activity
MY.NET.70.231	80	63.97.240.1	27374	Possible trojan server activity

The majority of the traffic seems to be one of the following Trojans: port 27374 Bad Blood, Fake SubSeven, li0n, Ramen, Seeker, SubSeven , SubSeven 2.1 Gold, Subseven 2.1.4 DefCon 8, SubSeven 2.2, SubSeven Muie, The Saint[72] None of these bring good news.

Alert	Count
Queso fingerprint	**834**

Queso is a scanning utility similar to NMAP that may be used to scan a network for various purposes. Attempts to access the website for queso, http://www.apostols.org/projectz/, met with a "Cannot find server or DNS Error" from the browser.

Top Talkers List

In determining the top talkers list, extensive analysis was performed on the data available. It was broken down into source and destination IP addresses and the counts of alerts, OOS entries, and scans that appeared. The listings of the top five external and internal IP addresses and their counts for each of the log types in order to look for patterns of traffic. The top ten list was extracted from this and is shown below.

ALERTS

External IP Destination	Count
216.173.214.13	21713

Internal IP Destination	Count
MY.NET.100.165	10899

External IP Source	Count
212.179.61.220	4839
212.179.126.3	3460

[72] **"**Ports used by trojans (2002-10-15)", Simovits Consulting, URL: http://www.simovits.com/nyheter9902.html (March 29, 2003)

Internal IP Source	Count
MY.NET.194.125	4362
MY.NET.105.204	4000

SCANS

Internal IP Source	Count
MY.NET.150.210	135906
MY.NET.195.155	72039

OOS

Internal IP Destination	Count
MY.NET.207.2	3383

External IP Source	Count
148.64.22.79	3247

An interesting fact in this analysis is that the same IP address did not appear in both source and destination charts indicating the traffic patterns are very much one way. Another interesting fact is that there was no alerts, scans, or OOS between internal servers indicating that any compromised servers are not being used to attack other internal servers. One last interesting fact is that the top talker in the alerts category had 21713 alerts that were "TCP SRC and DST outside network". This is highly unlikely and the source address is probably being spoofed and is originating inside the network. In order to determine the victim of this attack, it's registration information was looked up and shown in the "Five External Registration Information Examples" section.

Five External Registration Information Examples

The first IP address checked had a large number of alerts that were triggered by the watchlist "Watchlist 000220 IL-ISDNNET-990517" shown in an earlier section of this paper. This is traced back through the RIPE Whois database[73] to:

inetnum: 212.179.0.0 - 212.179.0.255
netname: REDBACK-EQUIPMENT
mnt-by: INET-MGR
descr: BEZEQINT-EQUIPMENT
country: IL
admin-c: MR916-RIPE
tech-c: ZV140-RIPE
status: ASSIGNED PA

[73] "Ripe Whois Database", URL: http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=212.179.0.0&do_search=Search (March 23, 2003)

```
remarks:      please send ABUSE complains to abuse@bezeqint.net
remarks:      INFRA-AW
notify:       hostmaster@bezeqint.net
changed:      hostmaster@bezeqint.net 20021020
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:       Miri Roaky
address:      bezeq-international
address:      40 hashacham
address:      petach tikva 49170 Israel
phone:        +972 1 800800110
fax-no:       +972 3 9203033
e-mail:       hostmaster@bezeqint.net
mnt-by:       AS8551-MNT
nic-hdl:      MR916-RIPE
changed:      hostmaster@bezeqint.net 20021027
changed:      hostmaster@bezeqint.net 20030204
source:       RIPE
person:       Zehavit Vigder
address:      bezeq-international
address:      40 hashacham
address:      petach tikva 49170 Israel
phone:        +972 1 800800110
fax-no:       +972 3 9203033
e-mail:       hostmaster@bezeqint.net
mnt-by:       AS8551-MNT
nic-hdl:      ZV140-RIPE
changed:      hostmaster@bezeqint.net 20021027
changed:      hostmaster@bezeqint.net 20030204
source:       RIPE
```

The second IP address checked for its registry information was done to confirm that the numerous "Russia Dynamo - SANS Flash 28-jul-00" alerts were legitimate. IP address 194.87.6.77 was checked through the Ripe Whois Database[74] as follows confirming that the destination is in Moscow, Russia.

```
% This is the RIPE Whois server.
```

[74] "Ripe Whois Database", URL: http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=194.87.6.77&do_search=Search (March 26, 2003)

% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum: 194.87.6.0 - 194.87.6.255
netname: DEMOS-DOL-DIALUP
descr: DEMOS-Online Dialup
descr: Demos-Internet Co.
descr: Moscow, Russia
country: RU
admin-c: DNOC-ORG
tech-c: DNOC-ORG
status: ASSIGNED PA
mnt-by: AS2578-MNT
remarks: **********
remarks: Please send abuse reports to abuse@demos.su
remarks: **********
changed: rvp@demos.net 20020911
source: RIPE
route: 194.87.0.0/19
descr: DEMOS
origin: AS2578
notify: noc@demos.net
mnt-by: AS2578-MNT
changed: noc@demos.net 20000927
source: RIPE
role: Demos Internet NOC
address: Demos Company Ltd.
address: 6-1 Ovchinnikovskaya nab.
address: Moscow 115035
address: Russia
phone: +7 095 737 0436
phone: +7 095 737 0400
fax-no: +7 095 956 5042
e-mail: ncc@demos.net
trouble: ----------
trouble: NOC working hours:
trouble: 09am-09pm MSK/MSD (GMT+3/+4) workdays
trouble: ----------
trouble: Contact addresses by category:
trouble: Routing/DNS/IP delegation: ncc@demos.net
trouble: SPAM/UCE: abuse@demos.net
trouble: Scans/Hacking attempts: security@demos.net
trouble: Mail: postmaster@demos.net
trouble: ----------
admin-c: KEV-RIPE

admin-c: RPS-RIPE
admin-c: GVS-RIPE
admin-c: VOX19-RIPE
tech-c: KEV-RIPE
tech-c: RPS-RIPE
tech-c: GVS-RIPE
tech-c: VOX19-RIPE
nic-hdl: DNOC-ORG
notify: hm-dbm-msgs@ripe.net
notify: ncc@demos.net
notify: ip-reg@ripn.net
mnt-by: AS2578-MNT
changed: evgeny@demos.su 20021021
changed: gvs@demos.su 20030207
source: RIPE

The third IP range that was checked was from the alert "TFTP - Internal TCP connection to external tftp server". A good portion of the connections were to different IP addresses on the 205.188.0.0 network, which is the America Online network as shown below[75]. The second lookup in this section was on the 65.12.0.0 network, which as you can also see was America Online as well[76]. The third lookup in this section shows the tftp traffic is at an IP address on US West.[77] The fourth lookup in this section shows an address in the Asian Pacific region.[78]

OrgName: America Online, Inc
OrgID: AMERIC-59
Address: 22080 Pacific Blvd
City: Sterling
StateProv: VA
PostalCode: 20166
Country: US

NetRange: 205.188.0.0 - 205.188.255.255
CIDR: 205.188.0.0/16
NetName: AOL-DTC
NetHandle: NET-205-188-0-0-1
Parent: NET-205-0-0-0-0
NetType: Direct Assignment
NameServer: DNS-01.NS.AOL.COM
NameServer: DNS-02.NS.AOL.COM
Comment:

[75] "ARIN Whois Database", URL: http://ws.arin.net/cgi-bin/whois.pl (March 27, 2003)
[76] ibid
[77] ibid
[78] "APNIC Whois Database", URL: http://www.apnic.net/apnic-bin/whois2.pl (March 27, 2003)

RegDate: 1998-04-18
Updated: 1998-04-27

TechHandle: AOL-NOC-ARIN
TechName: America Online, Inc.
TechPhone: +1-703-265-4670
TechEmail: domains@aol.net

OrgName: America Online, Inc.
OrgID: AMERIC-158
Address: 10600 Infantry Ridge Road
City: Manassas
StateProv: VA
PostalCode: 20109
Country: US

NetRange: 64.12.0.0 - 64.12.255.255
CIDR: 64.12.0.0/16
NetName: AOL-MTC
NetHandle: NET-64-12-0-0-1
Parent: NET-64-0-0-0-0
NetType: Direct Assignment
NameServer: DNS-01.NS.AOL.COM
NameServer: DNS-02.NS.AOL.COM
Comment:
RegDate: 1999-12-13
Updated: 1999-12-16

TechHandle: AOL-NOC-ARIN
TechName: America Online, Inc.
TechPhone: +1-703-265-4670
TechEmail: domains@aol.net

OrgName: U S WEST Internet Services
OrgID: USW
Address: 950 17th Street
Address: Suite 1900
City: Denver
StateProv: CO
PostalCode: 80202
Country: US

NetRange: 63.224.0.0 - 63.231.255.255
CIDR: 63.224.0.0/13
NetName: USW-INTERACT99
NetHandle: NET-63-224-0-0-1

Parent: NET-63-0-0-0-0
NetType: Direct Allocation
NameServer: NS1.USWEST.NET
NameServer: NS2.DNVR.USWEST.NET
Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate: 1999-06-07
Updated: 2002-08-12

TechHandle: ZU24-ARIN
TechName: U S WEST ISOps
TechPhone: +1-612-664-4689
TechEmail: abuse@uswest.net

OrgAbuseHandle: QIA2-ARIN
OrgAbuseName: Qwest IP Abuse
OrgAbusePhone: +1-703-363-3001
OrgAbuseEmail: abuse@qwest.net

OrgNOCHandle: QIN-ARIN
OrgNOCName: Qwest IP NOC
OrgNOCPhone: +1-703-363-3001
OrgNOCEmail: support@qwestip.net

OrgTechHandle: QIA-ARIN
OrgTechName: Qwest IP Admin
OrgTechPhone: +1-888-795-0420
OrgTechEmail: ipadmin@qwest.com

inetnum: 202.156.0.0 - 202.156.95.255
netname: SCVCABLENET-AP
descr: SINGAPORE CABLE VISION LTD
descr: SINGAPORE CABLE NETWORK PROVIDER
country: SG
admin-c: FK6-AP
tech-c: FK6-AP
mnt-by: APNIC-HM
mnt-lower: MAINT-SG-SCV
changed: hostmaster@apnic.net 19990929
changed: apnic-dbm@apnic.net 20000905
status: ALLOCATED PORTABLE
source: APNIC

The fourth lookup of registration information is to confirm the Watchlist 000222 NET-NCFC Alerts. The returned information is shown below.[79]

OrgName: The Computer Network Center Chinese Academy of Sciences
OrgID: CNCCAS
Address: P.O. Box 2704-10,
Address: Institute of Computing Technology Chinese Academy of Sciences
Address: Beijing 100080, China
City:
StateProv:
PostalCode:
Country: CN

NetRange: 159.226.0.0 - 159.226.255.255
CIDR: 159.226.0.0/16
NetName: NCFC
NetHandle: NET-159-226-0-0-1
Parent: NET-159-0-0-0-0
NetType: Direct Assignment
NameServer: NS.CNC.AC.CN
NameServer: GINGKO.ICT.AC.CN
Comment: The information for POC handle QH3-ARIN has been reported to
Comment: be invalid. ARIN has attempted to obtain updated data, but has
Comment: been unsuccessful. To provide current contact information,
Comment: please email hostmaster@arin.net.
RegDate: 1992-06-11
Updated: 2002-10-08

TechHandle: QH3-ARIN
TechName: Xiqiong, Zhang
TechPhone: 10 82616000
TechEmail: zxq@cstnet.net.cn

The fifth lookup of registration information is to determine the possible victim on an attack from the Universities internal network. The returned information is shown below.[80]

[79] "ARIN Whois Database", URL: http://ws.arin.net/cgi-bin/whois.pl (March 29, 2003)

[80] "ARIN Whois Database", URL: http://ws.arin.net/cgi-bin/whois.pl (March 29, 2003)

Olympia Networking Services OLYWA-OLYW (NET-216-173-192-0-1) 216.173.192.0 - 216.173.223.255
Level Seven L7-PLAZMA-WEBHOSTING (NET-216-173-214-8-1) 216.173.214.8 - 216.173.214.15

Correlations

One of the interesting things about the preparation of this paper was the correlations that were found in the practicals from GCIA candidates. In many cases, Internet web searches for alerts resulted in numerous practicals as the only source of information. Here are some specific correlations.

Brian Coyle encountered the alerts "Watchlist 000220 IL-ISDNNET-990517" as he outlined in his GCIA practical:

> "GCIA Practical", Coyle, Brian URL: http://www.linuxwidows.com/mirror/bucket/Brian_Coyle_GCIA.pdf (March 23, 2003)

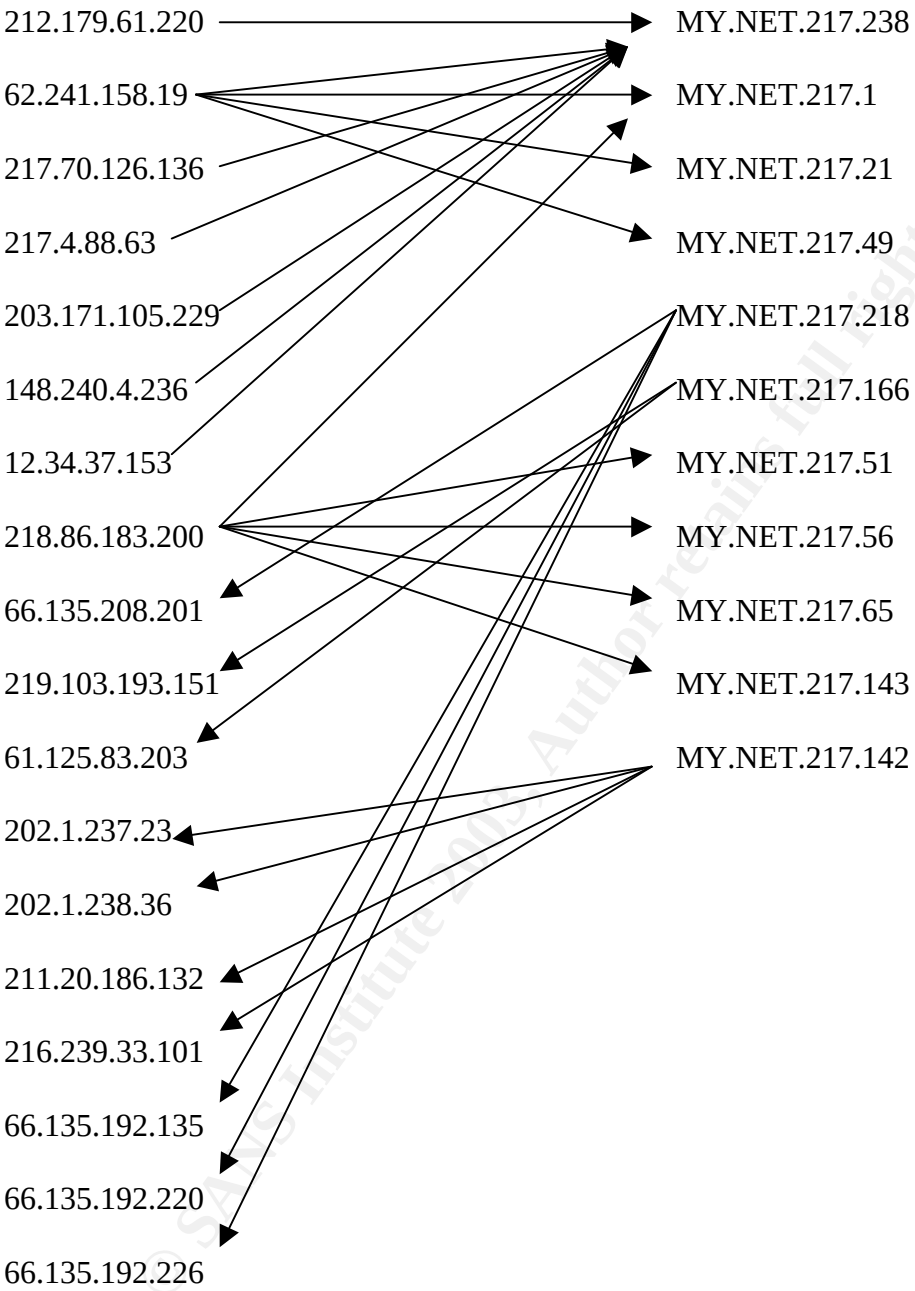
Scott Baird encountered the alerts "CS WEBSERVER - external web traffic" as outlined in his practical.

> "Intrusion Detection Indepth Version 3.0", Baird, Scott, URL: http://www.giac.org/practical/Scott_Baird_GCIA.doc (March 26, 2003)

Claudio Silotto detected the "External RPC call" alerts and did an excellent job of explaining what they meant.

> **"**GIAC Intrusion Detection Curriculum", Silotto, Claudio R. G. , URL: http://www.giac.org/practical/Claudio_Silotto_GCIA.doc (March 27, 2003)

Link Graph



The interesting fact about the above Link Graph is the amount of traffic that is either from or destined to the "MY.NET.217" subnet. This particular subnet of the University should be carefully evaluated for misconfiguration and compromised systems. There are three systems in particular that need to be addressed: MY.NET.217.238, which is the destination IP address of numerous alerts from different sources. The second is MY.NET.217.218, which is the source address

for several alerts to different destinations. The third is MY.NET.217.142, which is also a source address for several alerts to different destinations.

Internal Compromises

There are several systems that merit further investigation as alerts, scans, and OOS captures indicate a compromise or possible dangerous or anomalous activity.

The first system I would investigate is the system that is causing the alerts "TCP SRC and DST outside network". This is most likely a compromised system(s) that is sending out traffic with a spoofed source IP address. This is certainly suspicious and most likely malicious in nature. There is no reason to use spoofed source addresses otherwise. The source addresses listed, with few exceptions, are different. The vast majority of them are purported to come from the 13.x.x.x network. Some of them are reporting a source address of 0.0.0.0. A recommended procedure would be to capture traffic using TCPDump with the -e option. This would provide the MAC address of the offending computer(s) allowing further forensic investigation to determine the source and nature of the traffic.

The second system that should be investigated is IP address MY.NET.194.125. This system has over 4300 alerts for "Incomplete Packet Fragments Discarded" to an external system at IP address 67.65.70.129 alone. As stated in an earlier section of this document the cause of this anomaly could be a misconfigured network or it may indicate hostile traffic. Either way, the system needs attention.

The next two systems that require further investigation are systems that have a high number of "High port 65535 tcp - possible Red Worm – traffic" alerts. While there are 66 other systems that show this same alert, there were few alerts per address indicating that it is likely a false positive. The following two machines, however, had over 2000 alerts per machine making it unlikely that it is a false positive. MY.NET.88.193 had 3245 such alerts and MY.NET.208.14 had 2391 alerts. Both should be investigated for the Red Worm infection.

The fifth system that should be investigated shows a large amount of the alert "Russia Dynamo - SANS Flash 28-jul-00". It is the only internal system that shows that alert. MY.NET.105.204 has been conversing with 194.87.6.77 causing the alert in both directions.

Next, MY.NET.150.210 needs to be checked as it is the stated source of a huge number of port scans, almost all to destination port 1080, destined to over 130 thousand different IP addresses. This may be an indication of a Trojan infection such as SubSeven or WinHole[81].

Another system that needs attention is IP address MY.NET.195.155. It has been involved in Port 2303 scans to 107 different destination IP addresses. Port 2303 scans are not indicated as supporting Trojan activity. Port 2303 is assigned

[81] "Ports Used by Trojans", 10-15-2002, URL: http://www.simovits.com/nyheter9902.html (April 11, 2003)

to “Proxy Gateway”[82], but is unlikely that this system is talking to 107 different Proxy Gateways.

Defensive Recommendations

The University network appears to be wide open with few, if any, security measures in place to protect network resources. Universities strive to protect freedom of speech and to avoid appearances of censorship. This attitude is extended to the Internet and is in direct conflict with Internet security. Information Security is essentially a balancing act between free access to information and the protection of information assets. The University is skewed to the free access to information side of the scale. Information security policies must be put in place to define exactly where the governing body of the University wants that balance to sit.

The first and most obvious recommendation for the University is to put in place a Firewall or IP filtering router. While best practices would have the device allow that which is good and deny all else, it should at a minimum be used to control traffic once hostile or anomalous traffic has been detected. This one recommendation, if employed, would go a tremendous way in controlling the security of the network. It is not an end all in security but certainly a quick fix to many of the problems that are found here.

Once you have at least a minimum means of controlling traffic, an in-depth traffic analysis needs to take place to determine the extent of the problems on the network. Our short analysis has shown that the problems are extensive with hundreds of University systems likely compromised by Trojans or other hostile programs. An in-depth traffic analysis would allow the University to first control traffic by the firewall, and then take actions on individual systems to correct the problems.

Finally, a means of preventing further security problems needs to be employed. Email seems to be the prevalent means of transmitting Trojans and other viruses and must be controlled to protect a network. This, however, may be seen as censorship and may meet stiff resistance.

The Analysis Process

This analysis was hindered by the lack of extensive logging. It was not an in-depth traffic analysis as more than alerts, scans, and OOS are required for that. However, the data provided does allow a good look at the likely problems that exists on the network.

[82] IANA, “Well Known Ports Database”, (February 26, 2003) URL : http://www.iana.org/assignments/port-numbers (March 1, 2003)

The problem in analyzing even the relatively small volume of traffic available for this analysis is finding a way to make the data easy to read and easy to manipulate. In order to do this, the data was imported into a Microsoft Access® Database. Once the data was in the database, the individual lines were parsed to separate out the specific pieces of information into fields in a table. The alerts were parsed into a table as shown below:

Alert2 : Table

Field Name	Data Type	
DT	Text	Date/Time of Alert
Alert	Text	Alert Name
SrcIP	Text	Source IP Address
SrcPort	Text	Source Port
DstIP	Text	Destination IP Address
DstPort	Text	Destination Port

The Scans were parsed into a table as shown below:

Scans2 : Table

Field Name	Data Type	
Tme	Date/Time	Time of the Scan
Src	Text	Source IP Address
SrcPort	Text	Source Port
Dst	Text	Destination IP Address
DstPort	Text	Destination Port
Options	Text	TCP Options
Flags	Text	TCP Flags
Options2	Text	Other Options
Dte	Date/Time	Date of the Scan

The OOS logs were parsed into a table as shown below:

OOS_Parsed : Table

Field Name	Data Type	
Source	Text	Source IP Address
Sourceport	Text	Source Port
dest	Text	Destination IP Address
destport	Text	Destination Port

The parsing was done by taking the logs on a line by line basis and using code that would pull out the applicable data. As an example, the following is the code used to parse the alerts.

```
Private Sub Command5_Click()
On Error GoTo err_c5

Dim cnt, chk, Tme, alert, lt, Src, ln, srcport, Dst, dstport

parse:
```

```
If Me![Field3] Like "SPP_Portscan*" Then GoTo Nxt
If Me![Field3] Like "Tiny Fragments*" Then GoTo Nxt
If Me![Field3] Like "*possible myserver activity*" Then GoTo Nxt

cnt = 1
chk = Null
Tme = Me![Field1]

Do Until chk = "[**]" Or cnt = 255

   chk = Mid(Me![Field3], cnt, 4)
   cnt = cnt + 1
Loop

If cnt = 255 Then GoTo Nxt

alert = Left(Me![Field3], cnt - 3)

lt = cnt + 4

chk = Null

Do Until chk = ":" Or cnt = 255

   chk = Mid(Me![Field3], cnt, 1)
   cnt = cnt + 1

Loop

If cnt = 255 Then GoTo Nxt

ln = cnt - lt - 1

Src = Mid(Me![Field3], lt, ln)

lt = cnt
chk = Null

Do Until chk = "-" Or cnt = 255

   chk = Mid(Me![Field3], cnt, 1)
   cnt = cnt + 1

Loop

If cnt = 255 Then GoTo Nxt

ln = cnt - lt - 2

srcport = Mid(Me![Field3], lt, ln)

chk = Null
cnt = cnt + 2
lt = cnt
```

```
Do Until chk = ":" Or cnt = 255

   chk = Mid(Me![Field3], cnt, 1)
   cnt = cnt + 1

Loop

If cnt = 255 Then GoTo Nxt

ln = cnt - lt - 1

Dst = Mid(Me![Field3], lt, ln)
dstport = Mid(Me![Field3], cnt, 6)

DoCmd.SetWarnings False
sqlquerystr = "Insert into alert2 (DT, alert, srcip, srcport, dstip, dstport) values ('" & Tme &
"', '" & alert & "', '" & Src & "', '" & srcport & "', '" & Dst & "', '" & dstport & "')"
DoCmd.RunSQL sqlquerystr
DoCmd.SetWarnings True

Nxt:

DoCmd.GoToRecord , "alert_parse", acNext
GoTo parse

exit_c5:
Exit Sub

err_c5:
MsgBox Err.Description
Resume Next

End Sub
```

The code for parsing out the OOS and Scans files is slightly different, but employs the same techniques. They will not be included in this paper in the interest of saving space.

Once the files were parsed into the aforementioned tables, numerous different queries were formulated in order to show the data in different ways. Examples of the SQL queries used are shown below:

```
SELECT TOP 5 Alert2.DstIP, Count(Alert2.DstIP) AS CountOfDstIP
FROM Alert2
GROUP BY Alert2.DstIP
HAVING (((Alert2.DstIP) Not Like "my*"))
ORDER BY Count(Alert2.DstIP) DESC;

SELECT TOP 5 OOS_Parsed.Source, Count(OOS_Parsed.Source) AS
CountOfSource
FROM OOS_Parsed
GROUP BY OOS_Parsed.Source
HAVING (((OOS_Parsed.Source) Like "my*"))
```

```
ORDER BY Count(OOS_Parsed.Source) DESC;
```

Again, this is only an example of the different queries that were employed. There were over 25 different queries used to break the data down into meaningful information. Using this methodology provided the ability to manipulate the data in a meaningful way to show relationships and counts of detected traffic.

References

Richard, Matthew. SANS Intrusion Detection FAQ, “Are There Limitations of Intrusion Signatures”, April 5, 2001 URL: http://www.sans.org/resources/idfaq/limitations.php (February 23, 2003)

Cummings, Joanne. “Intrusion detection to Intrusion prevention.” Network World. Volume 19, No.38 (2002): 72-82. (May also be found at URL: http://www.nwfusion.com/buzz/2002/intruder.html (February 23, 2003))

Snort Signature Database, URL: http://www.snort.org/cgi-bin/needed.cgi?offset (February 23, 2003)

Briney, Andy. “What Isn’t Intrusion Prevention” April 2002, URL: http://www.infosecuritymag.com/2002/apr/note.shtml (February 23, 2003)

Piscitello, David. “Intrusion Detection … or Prevention?”, URL: http://www.bcr.com/bcrmag/2002/05/p42.asp (February 23, 2003)

Sequeira, Dinesh. SANS InfoSec Reading Room, “Intrusion Prevention Systems – Security’s Silver Bullet?” URL: http://www.sans.org/rr/intrusion/silver_bullet (February 23, 2003)

Messmer, Ellen. “Intrusion prevention systems raise hopes, concerns”, URL: http://www.computerworld.com/securitytopics/security/story/0,10801,75630,00.html (February 25, 2003)

Ethereal Network Protocol Analyzer, URL: http://www.ethereal.com/ (February 25, 2003)

Brindley, Adrian. “Denial of Service Attacks and the Emergence of “Intrusions Prevention Systems” November 1, 2002. URL: http://www.sans.org/rr/firewall/prevention.php (February 25, 2003)

Slighter, Tim. “Configuring IPTables for Snort Inline”, January 23, 2003. URL: http://www.snort.org/dl/contrib/patches/inline/ (February 26, 2003)

Liesen, Detmar. “Requirements for Enterprise-Wide Scaling Intrusion Detection Products. A Criteria Catalog for IT Executives, IDS Users, and Vendors. (Version 2002-06-19 Rev 3)”. URL: http://www.snort.org/docs/IDS_criteria.pdf (February 26, 2003)

Top Layer Products, “Products and Solutions”, URL: http://www.toplayer.com/content/products/index.jsp (February 26, 2003)

Incidents.Org Log Files, URL: http://www.incidents.org/logs/Raw/. (February 28, 2003)

WhiteHats.Com, "ArachNIDS Database", URL: http://www.whitehats.com/ids/index.html (February 28, 2003)

IEEE. "IEEE OUI and Company ID Database", URL: http://standards.ieee.org/regauth/oui/index.shtml (February 28, 2003)

Whitehats.com , "IDS175 SOCKS PROBE", Research Tab, URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids175&view=research (March 1, 2003)

IANA, "Well Known Ports Database", (February 26, 2003) URL : http://www.iana.org/assignments/port-numbers (March 1, 2003)

Google Search Engine, URL: www.google.com (March 1, 2003)

CERT Coordination Center, "Cert Advisory CA-1999-02 Trojan Horses", URL: http://www.cert.org/advisories/CA-1999-02.html (March 2, 2003)

SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-8 (Severity:Criticality Chart).

SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-9 (Severity:Lethality Chart).

SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-14 (Severity:System Countermeasures Chart).

SANS Institute, "Track 3 Intrusion Detection In-Depth, IDS Signatures and Analysis, Parts 1 and 2", page 4-15 (Severity:Network Countermeasures Chart).

GrimsPing, "Programming for the Free World", URL: http://grimsping.cjb.net/ (March 7, 2003)

Slora, James C. Jr. "Grims Ping Targeted Recon Probe?", URL: http://cert.uni-stuttgart.de/archive/intrusions/2002/08/msg00292.html (March 7, 2003)

Sans Institute, "Global Incidents Analysis Center – Detects Analyzed 5/13/00 –", URL: http://www.sans.org/y2k/051300.htm (March 9, 2003)

Internet Storm Center, "Log Files", URL: http://www.incidents.org/logs/ (Mar 11, 2003)

Lawrence Teo, “Network Probes Explained: Understanding Port Scans and Ping Sweeps” Linux Journal Online March 7, 2003, URL: http://www.linuxjournal.com/article.php?sid=4234 (March 14, 2003)

“WhoIs”, URL: http://www.arin.net/ (March 22, 2003)

“IDS: Source Port of Samba Scans”, Swan, Daniel, URL: http://www.shmoo.com/mail/ids/mar00/msg00065.shtml (March 23, 2003)

“Netbios File and Print Sharing”, Graham, Robert, URL: http://www.robertgraham.com/pubs/firewall-seen.html#port137 (March 23, 2003)

“GCIA Practical”, Coyle, Brian URL: http://www.linuxwidows.com/mirror/bucket/Brian_Coyle_GCIA.pdf (March 23, 2003)

“Ripe Whois Database”, URL: http://www.ripe.net/perl/whois?form_type=simple&full_query_string=&searchtext=212.179.0.0&do_search=Search (March 23, 2003)

“Adore Worm Version 0.8 – April 12, 2001”, URL: http://www.sans.org/y2k/adore.htm (March 26, 2003)

“Intrusion Detection Indepth Version 3.0”, Baird, Scott, URL: http://www.giac.org/practical/Scott_Baird_GCIA.doc (March 26, 2003)

“Intrusion Detection In Depth GCIA Practical Assignment Version 3.0 (revised August 13, 2001)”, Jenkins, David, URL: http://www.giac.org/practical/David_Jenkins_GCIA.doc (March 26, 2003)

“ARIN Whois Database”, URL: http://ws.arin.net/cgi-bin/whois.pl (March 27, 2003)

“APNIC Whois Database”, URL: http://www.apnic.net/apnic-bin/whois2.pl (March 27, 2003)

“Incident Report Date: February 14, 2001 – 0900”, SANS Global Incident Analysis Center, URL: http://www.sans.org/y2k/021401.htm (March 27, 2003)

“GIAC Intrusion Detection Curriculum”, Silotto, Claudio R. G. , URL: http://www.giac.org/practical/Claudio_Silotto_GCIA.doc (March 27, 2003)

“Neohapsis Archives”, from Joe Stewart. URL: http://archives.neohapsis.com/archives/snort/2000-11/0244.html (March 27, 2003)

"Scan 23 –South Florida Honeynet Project",Warner, Neil, URL: http://project.honeynet.org/scans/scan23/sol/Neil.html (March 29, 2003)

"SCAN OF THE WEEK #1 - 28 May - 3 June", The Honeynet Project, URL: http://project.honeynet.org/scans/arch/scan1.txt (March 29, 2003)

"Globbing", NCSA Archives, URL: http://archive.ncsa.uiuc.edu/General/Training/InterUnix/csh/glob.html (March 29, 2003)

"Ports used by trojans (2002-10-15)", Simovits Consulting, URL: http://www.simovits.com/nyheter9902.html (March 29, 2003)

Last Updated: May 27th, 2017

Upcoming Training

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