Global Information Assurance Certification Paper

Assignment I - Honeypots and Honeynets: Their Place in the Current State of Intrusion Detection

A white paper for the SANS GCIA certification

David Manley
Intrusion Detection In-Depth
GCIA Practical Assignment, Version 3.0
SANS Cyber Defence Initiative
Washington, DC – Nov/Dec 2001

Table of Contents

1 Introduction

Over the past few years, the Intrusion Detection field has grown rapidly, and with it has spread the awareness of the concepts of honeypots and honeynets. While these concepts are hardly new, the growing recognition of the need for Intrusion Detection systems in industry today has caused honeypots and honeynets to come increasingly under consideration for their potential security benefit.

While currently they are not as widely deployed as other Intrusion Detection systems, the discussion of concepts and designs, as well as the debate over the value and drawbacks of honeypots and honeynets is becoming more widespread. This paper will define the concepts surrounding these tools and will attempt to clarify the debate over their advantages and disadvantages so that the reader can make an informed judgement.

2 Honeypots and Honeynets

This section will define the terms *honeypot* and *honeynet,* and the concepts surrounding the terms. It will also discuss the reasons one might have for deploying these intrusion detection technologies.

2.1 What is a honeypot?

Martin Roesch, creator of the Snort intrusion detection system, uses the synonym "deception systems" for honeypots and defines them as dedicated hosts that entice intruders to probe and attack them. [1] Lance Spitzner of the Honeynet Project, one of the leading proponents for the use of honeypots, defines a honeypot as "a resource whose value is in being attacked or compromised". [2]

From a security perspective, the concept of designing a system for the purpose of being breached at first seems to be somewhat counterintuitive. However, a honeypot's sole purpose in design and placement in a network is to be an attractive target for attack, and indeed the success of a honeypot is measured upon it. The reasons for wanting a system to be attacked will be discussed in the following section.

Honeypots come in an extremely diverse array, but they all contain vulnerabilities and should therefore be attractive targets for hackers. For example, a honeypot could be commercial software such as Back Officer Friendly from NFR (no longer available for purchase) installed on a Windows machine that emulates standard services such as http or telnet or the Back Orifice Trojan. [3] It may even be the operating system itself that is emulated within a sandbox, a mechanism that controls or prevents the interaction of the intruder with the actual underlying operating system. A honeypot can even be just a spare machine that is taken off of the shelf, installed with a basic operating system (of any variety), and put on the network without any hardening.

The tools for building a honeypot are as varied as the security vulnerabilities that currently exist today. What makes for a good honeypot design depends in large part upon what results the administrator hopes to gain.

2.2 Why would I want a system to be attacked?

The reasons and objectives for deploying a honeypot are varied, but most experts agree that they fall into two main categories: r esearch and production. [4]

2.2.1 Research Honeypots

Research honeypots are an information -gathering endeavour which is purely academic. Their creators design and deploy a system that is an attractive target for a ttack in order to watch and learn how the system is discovered, what reconnaissance met hods and tools are used, how it is breached, and what activities occur after the system has been compromised. The hope is that during the attack and p otentially after the intrusion, information about the process and meth odology of hacking can be gleaned. Some additional goals may be to learn generally the motivations that hackers have for attacking systems, broad or specific methods and tools that are employed, and to observe the use of new exploits for known vulnerabil ities.

Probably the most well known example of a research honeypot in use today is the Honeynet Project. The goal of this non -profit group is "to learn the tools, tactics, and motives of the Blackhat communi ty and share these lessons learned". [5] By sharing the knowledge gained through their research, the creators of the project hope that the entire security community will benefit from the increased awareness of the current dangers on the Internet and the to ols and exploits that are currently in use, and to spread the knowledge with which the security comm unity can better arm itself against attack.

2.2.2 Production Honeypots

Production honeypots are designed to be a decoy to lure hackers away from real data that is secured. This diversionary tactic is achieved by creating a more attractive ta rget that is easier to compromise than the legitimate systems housing the a ctual corporate resources or information. Goals of production honeypot creators might be to gain ti me to track hackers or to help in determining their identities, to act as an early warning sy stem by alerting administrators to hacking activity, as well overlapping with the goals of r esearch honeypots, e.g., to learn the motivation and characteristics of attacks in an effort to improve defences by u sing this knowledge.

An example of a production honeypot in use may well have been a well -publicized break-in by Russian hackers of the Microsoft corporate network in October of 2000. It was widely reported that during a period of several days, hackers possibly had access to Microsoft source code for Windows and Office software. However, one analyst with the Gartner group was quoted as saying, "There is a s trong possibility that the hacker really did not get into anything more than what a well -designed security system based on a honeypot network would allow". [6]

Whether or not it was actually a production hone ypot that was compromised may not ever be publicized, but using false source code as a lure in a designed -to-be-loosely-secured network by Micr osoft would fit the bill of a production Honeypot. However, it would be

counterproductive for Microsoft to release information about the successful application of its security methodology to the pu blic.

2.3 What is a honeynet?

A honeynet is the honeypot concept expanded to include multiple networked hosts and the additional systems used for controlling and logging the activities of an i ntruder (such as a syslog server and firewall). The advantage of a honeynet over a single honeypot system is that it more convincingly simulates a production environment, i.e., separate systems with different operating systems and services available on the network. This means tha t systems more accurately reflecting those used in production can be placed within a honeynet. Another advantage is that since multiple types of operating systems and services are available, the chances of attack i ncrease. [7]

The goals and reasons for dep loying a honeynet are the same as for a honeypot, but with the added aim of more closely simulating an actual production env ironment.

2.3.1 Virtual Honeynets

One of the latest innovations in this arena is the virtual honeynet. A virtual honeynet is created u sing software called Vmware which allows virtual machines to be created within a single computer. Each of the components of a honeynet, including potentially the firewall and network IDS, are simulated in virtual ma chines and a virtual network ru nning on one computer. An entire network simulated on one computer, in itself an exciting innov ation, has the additional benefit of eliminating two of the logistical problems associated with deploying a honeynet – the need for both space and hardware for each of the servers. It is also arguably cheaper, because the software required for running multiple virtual m achines is the main expense, although individual operating systems may still have to be purchased, and the hardwar e used must be robust enough to support the load. [8]

3 Logging

The method by which an intruder and his or her actions are tracked is as critical as the honeypot or honeynet itself. Some commercial honeypot systems include a facility for logging the intruder s' activities while they are visiting the honeypot host. However, it may be preferable to include a separate, stand -alone packet-sniffer such as Snort, tcpdump, or Ethereal, because once the honeypot host is compromised, the logs on that host may not be trustworthy.

4 Pros and Cons

This section will discuss the advantages and disadvantages of the use of honeypot and honeynet technologies.

4.1 Pros

- **Education and research**. Through the use of these technologies, a great deal can be learned about the typical be haviour, methods, tools, habits, motives, and exploits used by hackers. This knowledge can be shared, leading to overall improvements in the security of the Internet.

- **Discovery of new exploits**. If an unknown attack is used to attack a honeypot, the traffic can be examined and new signatures for IDS products created to d etect the attack. This benefit was shown in January 2002 when a honeypot captured the first known attack of the Unix CDE dtspcd service, a known vulnerabi lity but one for which previously there had been no known exploit. [9]

- **Fewer false positives**. False positives are a typical problem with other intr usion detection systems, but one that is reduced with honeypots. Beca use the honeypot is not a production system, there is no reason for valid network traffic or activity to take place on the system or network. This also provides an anal ysis benefit, as there is very little "chaff," but plenty of "wheat".

- **Early Warning Sy stem**. The fact that traffic should not be going to a honeypot normally means that triggers can be set to alert administrators as soon as any traffic or activity is detected. This is especially useful with production hone ypots or honeynets, as security s taff can be alerted during the early stages of an a ttack, and tracking, monitoring, and defence of resources can be performed before critical systems come under attack.

- **Diversion**. As hackers generally do reconnaissance before attacking a ne twork, the results of this reconnaissance would hopefully point them in the dire ction of the honeypot or honeynet rather than toward crucial production sy stems.

4.2 Cons

- **Time and resources**. In many IT security departments, time and resources are scarce, and honeypots, honeynets, and virtual honeynets require a significant amount of time and effort to be set up, monitored, and maintained. Whether they are to be used as a research tool or in production, it would be reasonable to perform a cost/benefit analysis before em ploying these technologies. One estimate suggests that for every 30 minutes an intruder spends on a honeypot system, between 30 and 40 hours will be required to analyse the information captured. [10]

- **Exposure**. A honeypot system is designed to be atta cked, generally by someone who does not respect normal legal or ethical boundaries. The da nger exists that the system, once compromised, could be used to attack other sy stems, and

potentially leave the administrator of the honeypot responsible for the hac ker's actions.

- **Not a total solution**. These technologies compliment other security measures taken in an organization, but cannot replace standard security practices and devices.

- **Legal concerns**. Honeypots have sometimes been compared to electronic wiretapping, and critics have questioned whether or not they constitute entrapment. L egal experts commenting about this have suggested that honeypots fall short of entrapment because they are not causing the hackers to perform any actions that they would not otherwise do. However, there is very little legal precedence with respect to hone ypots, and care must be taken in this regard.

- **Ethical concerns**. Administrators of honeypots often feel conflicted about "spying" on the intruder, even though the intruder b eing watched lacks the acknowledgement of normal privacy or other ethical bound aries. According to an article in Wired News, one of the original Honeynet Project team members, J. D. Glaser of Foundstone, r esigned from the project because he felt that electronic wiretapping is wrong even if used for r esearch and that honeypots may "[tramp] on others' rights, even criminals' rights." [11]

Conclusion

This paper endeavoured to give an over view of the concepts and uses of current hone ypot intrusion detection technology, as well as to present a balanced view of their potential benefits as well as their possible drawbacks. It is hoped that armed with this knowledge, organizations can make an informed decision regarding these technologies and whether they have a place in their enterprises.

Further Reading

- Overview of commercial honeypot products and "homemade honeypots" – Lance Spitzner, The Value of Honeypots, Part Two (http://www.securityfocus.com/infocus/1498)

- Overview of Honeynets – The Honeynet Project, Know Your Enemy: Honeynets (http://www.securityfocus.com/infocus/1209)

- Building a Vmware Honeynet – Michael Clark, Virtual Honeynets (www.securityfocus.com/infocus/1506

- Kurt Seifried, Honeypotting with Vmware – basics (http://seifreid.org/security/ids/20020107 -honeypot-vmware-basics.html

- Vmware – (http://www.vmware.com)

References:

1 – Roesch, Martin, course material for SANS Intru sion Detection Snort Style

2 – Spitzner, Lance – The Value of Honeypots, Part 1 (www.securityfocus.com/infocus/1498)

3 – Spitzner, Lance – The Value of Honeypots, Part 2 (www.securityfocus.com/infocus/1498) (You can download Back Officer Friendly from http://www.enteract.com/~lspitz/bof.zip)

4 – ibid, 2

5 – Honeynet Project, The - About the Project - http://project.honeynet.org/project.html

6 - Ticehurst, Jo - Microsoft 'set hacker trap' th eory - http://www.vnunet.com/News/1 113504

7 – Spitzner, Lance – Know Your Enemy: Honeynets - http://www.securityfocus.com/infocus/1209

7 - ibid, 1

7 - Clark, Michael, Virtual Honeynets - http://www.securityfocus.com/infocus/1506

8 – Clark, Michael, Virtual Honeynets - http://www.securityfocus.com/infocus/1506

8 – Siefried, Kurt, Honeypotting with Vmware – basics - http://seifreid.org/security/ids/20020107 -honeypot-vmware-basics.html

9 – Honeynet Project, The – *untitled* - http://project.honeynet.org/scans/dtspcd/dtspcd.txt

10 – Honeynet Project, The – Know Your Enemey: Honeynets - http://www.securityfocus.com/infocus/1209

11 – Delio, Michelle – Honeypots: Bait for the Cracker - http://www.wired.com/news/culture/0,1284,42233,00.html

---

Assignment II - Intrusion Detection In-Depth: Network Detects

2 Assignment 2 – Network Detects

2.1 Detect One

Date	Source IP Address: Port	Destination IP Address: Port
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.132:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.133:6112 SYN ****** S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.134:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.136:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.137:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6 112	xxx.xxx.xxx.138:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.139:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.151:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.152:6112 SYN ******S *
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.153:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.154:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.155:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:61 12	xxx.xxx.xxx.156:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.158:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.159:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.160:6112 SYN ******S*
Feb 11 01:19:59	204.192.116.243:6112	xxx.xxx.xxx.162:6112 SYN ******S*

Fig 2.1 Scan excerpt from http://www.incidents.org/archives/intrusions/msg03765.html

2.1.1 Source of Trace:

A detect taken from the Incidents.org archives, posted 11 February 2002, which can be found at the following URL:

http://www.incidents.org/archives/intrusions/msg03765.html

2.1.2 Detect Generated By:

Unknown. It appears to be a Snort Scan log, a very basic firewall log, or another type of log that has been sanitized and trimmed.

2.1.3 Probability of Spoofed Source Address:

Slim. At first glance, this appears to be a scan for a remotel y exploitable buffer overflow within the Unix Common Desktop Environment. Because this is probably a scan for a port and service with a known vulnerability that allows the attacker to potentially take control of the system, the attacker would have to know which systems are vulnerable in order for this information to be useful. Therefore, it would not make sense to scan from a spoofed IP address because the information would not be returned to the attacker.

2.1.4 Description of attack:

The excerpted scan a bove is a very fast scan of 17 hosts for port 6112/tcp. A CERT advisory (CA-2001-31) for a buffer overflow of the dtspcd service, which typically runs on port 6112/TCP, was released on November 12, 2001 and updated on January 10, 2002. The vulnerability was assigned the CVE candidate number CAN -2001-0803.

The vulnerability described in the advisory is a buffer overflow within the Common Desktop Environment, or CDE, which is the graphical user interface used on many Unix and Linux systems. Because the C DE comes fully installed and enabled by default on many systems, the potential of the host initiating the scan finding this vulnerability on the Internet is fairly high.

Although the vulnerability was first reported in 1999, t he timeframe of the recently released CERT advisory corresponds to the date of the post to incidents.org. Furthermore, vendors released patches between 7 Nov 2001 and 21 Feb 2002, so there was likely to be a number of un-patched systems on the Internet at the time of the scan (See Fi g. 2.1.2).

Vulnerability Note VU#172583 – Vendor Patch Information		
Vendor	Status	Date Updated
IBM	Vulnerable	17-Dec-2001
SGI	Vulnerable	30-Nov-2001
Compaq Computer Corporation	Vulnerable	12-Nov-2001
Hewlett Packard	Vulnerable	21-Feb-2002
Sun	Vulnerable	10-Jan-2002
The Open Group	Vulnerable	12-Nov-2001
Cray	Not Vulnerable	31-Oct-2001
Fujitsu	Not Vulnerable	31-Oct-2001
Caldera	Vulnerable	7-Nov-2001
Data General	Unknown	31-Oct-2001
Xi Graphics	Vulnerable	15-Nov-2001
TriTeal	Unknown	12-Nov-2001

Fig. 2.1.2 Table Taken From: http://www.kb.cert.org/vuls/id/172583

2.1.5 Attack mechanism:

This remotely exploitable buffer overflow exists in the CDE Subprocess Control Service called dtspcd. dtspcd is a network daemon that is designed to execute commands and launch applications upon request from remote clients. As stated previously, dtspcd usually runs on port 6112/tcp, and, most importantly, runs with root privileges. According to the CERT advisory, on the un-patched systems, "…dtspcd accepts a length value and subsequent data from the client without performing adequate input validation." If this data is manipulated in a particular way it can cause a buffer overflow, and potentially cause code to be executed with root privileges. Of course, this means that a skilled attacker could potentially use this exploit to gain control of the target system.

2.1.6 Correlations:

The first detect was submitted by Ernie Pritchard on 11 February 2002. A subsequent post was submitted later in the day by Michael Dwyer, and the sample scan he submitted contained an entry from the same host as Mr. Pritchard's original post. There were three similar scans reported by three other individuals - one more on the same day and two the next day. The additional posts can be found at the following URLs:

http://www.incidents.org/archives/intrusions/msg03766.html
http://www.incidents.org/archives/intrusions/msg03768.html
http://www.incidents.org/archives/intrusions/msg03774.html
http://www.incidents.org/archives/intrusions/msg03780.html

Further correlation is the first observed use of an exploit for this vulnerability by the Honeynet Project in January of 2002. Details of the Honeynet Project compromise are located at the following URL: http://project.honeynet.org/scans/dtspcd/dtspcd.txt . This scan was included because of its relationship to text included in Assignment I of this practical, in which the Honeynet Project discovery of the first use of this exploit is mentioned.

2.1.7 Evidence of Active Targeting:

Multiple scans of large parts of network ranges conducted on the same day, and in one case two networks scanned from the same host, all for a known vulnerability, point to active targeting. This appears to be the "reconnaissance phase" of the attack, and it can be assumed that the results of this scan will be used in an attempt to compromise the vulnerable host(s).

2.1.8 Severity:

The post did not mention what systems were involved, whether CDE was running, or what firewall security policy was in place. Therefore, determining the severity from the information given is difficult.

Using the formula (Criticality + Lethality) – (System + Network Countermeasures) = Severity, the following analysis was made:

Criticality: Depending on importance of target host, criticality = 4.
Lethality: If this exploit works, they can gain root access, and therefore, lethality = 5.

Total = 9

System Countermeasures: If patches are applied, and CDE not running if unnecessary = 5
Network Countermeasures: Firewall blocks port 6112 = 5

Total = 10

Severity: **(9) – (10) = -1**

2.1.9 Defensive Recommendation:

Defensive recommendations wou ld be to find and apply the patches for this vulnerability to the systems affected. The necessity and use of CDE should be questioned if the system is accessible from the Internet, and should be turned off in almost all cases if the system is a firewall, IDS, or DNS system. Furthermore, port 6112/tcp, as well as all other unused ports, should be blocked by the firewall.

2.1.10 Multiple Choice Test Question:

You notice a number of drops in your firewall logs to every host on your network on a port you do not recognize. What is the most likely explanation for this activity?

A) It is most likely a web crawler looking for web pages to add to its search engine database.

B) It is likely a reconnaissance scan for hosts running a service with a known vulnerability.

C) It is probably your ISP checking for active hosts on its IP range.

D) It is probably an attack against the port in question on any server it can find.

Correct Answer: B

2.2 Detect Two

[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Inform ation Leak] [Priority: 2]
02/08-00:25:19.032255 61.18.133.100:4485 -> x.y.z.102:8080
TCP TTL:50 TOS:0x0 ID:15500 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FAEF265 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[**] [1:61 5:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.032255 61.18.133.100:4482 -> x.y.z.101:1080
TCP TTL:50 TOS:0x0 ID:15501 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FAD19EB Ack: 0x0 Win: 0x2000 TcpLen : 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.032255 61.18.133.100:4488 -> x.y.z.104:1080
TCP TTL:50 TOS:0x0 ID:15503 IpLen:20 DgmLen:4 8 DF ******S* Seq: 0x3FB0F879 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.032255 61.18.133.100:4481 -> x.y.z.100:8080
TCP TTL:50 TOS:0x0 ID:15506 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FAC1CBE Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.032255 61.18.133.100:4487 -> x.y.z.103:8080
TCP TTL:50 TOS:0x0 ID:15508 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FB062E6 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[**] [1:615:1] S CAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.032255 61.18.133.100:4484 -> x.y.z.102:1080
TCP TTL:50 TOS:0x0 ID:15509 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FAE6764 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[**] [1:620:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.042255 61.18.133.100:4483 -> x.y.z.101:8080
TCP TTL:50 TOS:0x0 ID:15513 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FADB471 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/08-00:25:19.042255 61.18.133.100:4480 -> x.y.z.100:1080
TCP TTL:50 TOS:0x0 ID:15514 IpLen:20 DgmLen:48 DF
******S* Seq: 0x3FAB3F67 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

Fig. 2.2.1 Detect Taken from http://www.incidents.org/archives/intrusions/msg03748.html

2.2.1 Source of Trace:

A detect taken from the Incidents.org archives, posted 8 February 2002, which can be found at the following URL:

http://www.incidents.org/archives/intrusions/msg03748.html

The table above (Fig 2.2.1) is an excerpt of the posted message.

2.2.2 Detect Generated By:

Unknown. The submission is probably Snort packet dumps. Fig. 2.2.2 is an explanation of the fields in the log file. The last scan entry from above is used for illustrative purposes, and the explanation of each field is in red.

<table>
<tr><td colspan="6">[**] [1:615:1] SCAN Proxy attempt [**] – Information about the signature/rule</td></tr>
<tr><td colspan="6">[Classification: Attempted Information Leak] [Priority: 2] – Classification and Priority Information</td></tr>
<tr><td>02/08-
0:25:19.042255 -
Time Stamp
TCP – Protocol</td><td>61.18.133.100:4480
– Source IP
Address: Port
TTL: 50 – Time To
Live</td><td colspan="2">-> x.y.z.100:1080 -
Destination IP Address (Sanitized): Port
TOS: 0x0 – Type
of Service</td><td>ID: 15514 - IP
Fragment
Identification</td><td>IpLen:
20 – IP
Header
Length</td><td>DgmLen: 48
- Datagram
Length</td></tr>
<tr><td>DF ******S* – TCP
Flag – (Syn and
don't fragment in
this case.)</td><td>Seq: 0x3FAB3F67
- Packet sequence
number (hex)</td><td>Ack: 0x0 - TCP
Acknowledgement
number (hex)</td><td>Win: 0x2000 –
TCP window
size</td><td colspan="2">TcpLen: 28 –
TCP header length</td></tr>
<tr><td colspan="6">TCP Options (4) MSS: 1460 NOP NOP => SackOK – TCP options: max. segment size, 2 no operations to pad TCP options to make them fall on 4 -byte boundaries, sel ective acknowledgements permitted</td></tr>
</table>

Fig. 2.2.2 – Explanation of the log file fields

2.2.3 Probability of Spoofed Source Address:

Unlikely. A quick analysis of this indicates that it appears to be a scan for a potential vulnerability within a proxy server or a scan for Winhole (1080) and RingZero (8080), which are both Trojan Horse applications. The source IP address is from a range of addresses allocated to a cable provider in Hong Kong, and probably indicates a potential attacker with cable high -speed Internet access. In order for this scan to be effective, the results of the scan would have to be returned to the person initiating the scan, and therefore it is probably from a valid IP address.

2.2.4 Description of Attack:

The ports in question are 8080/tcp and 1 080/tcp. Typically, port 8080/tcp is used for web proxy services and port 1080/tcp is a socks proxy port. However, as mentioned above, both of these are also ports used by lesser -known Trojan Horse applications. The fact that the scan was limited to thes e two particular ports, which are typically used for proxy services, and that other, more well -known Trojan Horse ports were not scanned, leads me to believe that this is not a scan for Trojan Horses.

A search of the CERT current activity page revealed an entry for port 1080/tcp and a link to the vulnerability note numbered VN-98:03. The CERT vulnerability note is entitled "WinGate IP Laundering", and discusses a vulnerability in WinGate software from Deerfield. WinGate software allows the sharing of a single Internet connection, while, according to th eir website, "…protecting the valuable information on the network with the integrated **proxy server/firewall**."

Proxy servers can be used for many different purposes, including use as a firewall or as a web-caching server, but, most importantly in our anal ysis, they have the effect of hiding the IP addresses of the computers using the proxy server to access the Internet behind the IP address of the proxy itself. This provides an obvious benefit to a hacker or someone wishing to conduct any sort of activity calling for his or her identity to be concealed. Therefore, this probably is a scan for WinGate proxy servers or other proxy servers having a vulnerability that could be exploited and used by the attacker to conceal their true location. For our analysis , we will concentrate on WinGate.

An additional reason for seeking out proxies of this sort would be to use the exploit of any vulnerability to gain access to the server and use that as a point of entry into an internal network environment.

2.2.5 Attack Mechanism:

Further reading and investigation explains the vulnerability found in older versions of WinGate software. The CERT vulnerability note indicates that "[t] he default configuration for WinGate allows an intruder to use a WinGate server to conceal his or her true location without the need to forge packets." The Deerfield WinGate page also has a link that contains security information, from which the following was excerpted:

> **WinGate Security Information**
>
> **Version 1.3 / 2.0** (1995-1998)
> WinGate 1.3 and 2.0 versions offered efficient Internet sharing abilities; however, these versions did not offer a bindings tab to allow you to specify what interfaces you were accepting or not accepting connections on. This allowed the WinGate server to listen on both interfaces (internally and externally), which did raise some security issues. Since WinGate was the first Internet sharing solution to market, the developers were quick to enhance WinGate, and add the ability to specify which interfaces should be internally or externally accessible.

It appears that the attack mechanism used to exploit WinGate is simply taking advantage of sites that used the default configura tion when installing their WinGate server. The default configuration allows external "users" to access the WinGate proxy as if they were located internally.

2.2.6 Correlations:

No follow-up posts were submitted to Incidents.org, however, there is ample evidence of an exploitable vulnerability on the ports being scanned.

2.2.7 Evidence of Active Targeting:

This scan is clearly the prelude to active targeting, as it was a scan of multiple hosts on a single network for two similarly vulnerable ports. Additionally, the total of 930 alerts detected according to the submission indicates it was not an accidental scan. It can be assumed that if a vulnerable server was found, it would have been exploited.

2.2.8 Severity:

The post did not mention what systems were involved o r whether a WinGate proxy was being used. Therefore, determining the severity from the information given is based upon certain assumptions.

Using the formula (Criticality + Lethality) – (System + Network Countermeasures) = Severity, the following analy sis was made:

Criticality: In this case, the proxy server would probably be the firewall and perhaps border router/Internet access point, and therefore, criticality = 5.
Lethality: If this exploit worked, internal network access could have been gained or the proxy server used as a launching point for further hacking, and therefore, lethality = 5.

Total = 10

System Countermeasures: Most likely, the organization submitting this scan is not using WinGate as its proxy or for sharing Internet access = 5
Network Countermeasures: Likely using Snort and probably not using WinGate = 5

Total = 10

Severity: **(10) – (10) = 0**

2.2.9 Defensive Recommendation:

WinGate provides information about securing a WinGate installation here. The website provides easily understood methods of securing WinGate. The two main recommendations are to use the WinGate rule base mechanism to create access rules and to employ a security policy defining specific users or machines allowed to use the proxy. The second recommendation is to bind WinGate to a specific interface (obviously the internal one), thereby eliminating the access to the proxy from the external network (e.g., the Internet).

2.2.10 Multiple Choice Test Question:

Which of the following two ports are commonly used for proxy servers?

A) 25/tcp and 75/tcp
B) 9008/tcp and 8008/tcp
C) 8080/tcp and 1080/tcp
D) 16/tcp and 17/tcp

Correct answer: C

2.3 Detect Three

02/01-12:34:42.100938 208.1.82.9:22227 -> www.xxx.yyy.2:22227 TCP TTL:119 TOS:0x0 ID:12147 IpLen:20 DgmLen:40
******S* Seq: 0x6345036 Ack: 0x686A6B36 Win: 0xFCA4 TcpLen: 20
02/01-12:34:42.116733 208.1.82.9:22227 -> www.xxx.yyy.4:22227
TCP TTL:119 TOS:0x0 ID:12147 IpLen:20 DgmLen:40
******S* Seq: 0x6345036 Ack: 0x686A6B36 Win: 0xFCA4 TcpLen: 20
02/01-12:34:42.143022 208.1.82.9:22227 -> www.xxx.yyy.6:22227
TCP TTL:119 TOS:0x0 ID:12147 IpLen:20 DgmLen:40
******S* Seq: 0x6345036 Ack: 0x686A6B36 Win: 0xFCA4 TcpLen: 20
02/01-12:34:42.143351 208.1.82.9:22227 -> www.xxx.yyy.5:22227
TCP TTL:119 TOS:0x0 ID:12147 IpLen:20 DgmLen:40
******S* Seq: 0x6345036 Ack: 0x686A6B36 Win: 0xFCA4 TcpLen: 20
02/01-12:34:42.177385 208.1.82.9:22227 -> www.xxx.yyy.8:22227
TCP TTL:119 TOS:0x0 ID:12147 IpLen:20 DgmLen:40
******S* Seq: 0x6345036 Ack: 0x686A6B36 Win: 0xFCA4 TcpLen: 20
02/01-12:34:42.180084 208.1.82.9:22227 -> www.xxx.yyy.10:22227
TCP TTL:119 TOS:0x0 ID:12147 IpLen:20 DgmLen:40
******S* Seq: 0x6345036 Ack: 0x686A6B36 Wi n: 0xFCA4 TcpLen: 20

Fig. 2.3.1 – Excerpt from http://www.incidents.org/archives/intrusions/msg03726.html

2.3.1 Source of Trace:

Incidents.org, a post from 6 February 2002, which can be found at the following URL:

http://www.incidents.org/archives/intrusions/msg03726.html

An excerpt of the submission is in the table above (Fig 2.3.1).

2.3.2 Detect Generated By:

The method used to extract the packets was not mentioned in the submission to Incidents.org. The last packet from Fig 2.3.1 is used to explain each field (Fig. 2.3.2).

02/01-12:34:42.180084 - Time Stamp	208.1.82.9:22227 – Source IP Address: Port	-> www.xxx.yyy.10:22227 - Destination IP Address (Sanitized): Port			
TCP - Protocol	TTL: 119 – Time To Live	TOS: 0x0 – Type of Service	ID: 12147 - IP Fragment Identification	IpLen: 20 – IP Header Length	DgmLen: 40 - Datagram Length
******S* – TCP Flag – (Syn in this case.)	Seq: 0x6345036 - Packet sequence number (hex)	Ack: 0x686A6B36 - TCP Acknowledgement number (hex)	Win: 0xFCA4 – TCP window size	TcpLen: 20 – TCP header length	

Fig 2.3.2 – Explanation of log file fields

2.3.3 Probability of Spoofed Source Ad dress:

Unlikely, however, it is an unknown. This scan does not lend itself to an obvious quick interpretation, and therefore a spoofed IP address cannot be ruled out.

2.3.4 Description of Attack:

The thing that jumps out when first glancing at these packet s is the fact that the source and destination ports are the same, and they are an unusual port (22227). A quick check with the Internet Assigned Naming Authority list of port numbers for the port 22227 and 22224 (mentioned in the text of the post to Incidents.org but no packets with this port were posted) reveals that both ports are currently unassigned. A check of two lists of popular Trojan Horse ports revealed only that several Trojans use port 22222 and that some of them allow the port to be modified.

Moving beyond the obvious source -destination port equivalen ce and delving deeper into these packets, it becomes apparent that each packet in this submission is in fact identical except for the time stamp and destination IP address. This indicates that this is almost certainly a scan using crafted packets. There are several “Black Hat” tools that can be used to craft packets for automated scanning. Two of the most popular ones are SynScan and t0rnscan.

Unfortunately, without further information such as more packets or payload, it is nearly impossible to know e xactly what this particular scan was targeting.

2.3.5 Attack Mechanism:

Unknown. It is not possible to determine from the data exactly what the intent of this scan is. However, crafted packets and a scan for an unusual port most likely indicate bad traffic rather than good.

2.3.6 Correlations:

No follow-up posts were submitted. The post did have an excerpt incorporated within the submission from another individual who came to a similar conclusion that these appear to be crafted packets.

2.3.7 Evidence of Active Tar geting:

The scan for an unusual port and the evidence of crafted packets indicate that this is a search for some sort of vulnerability that can potentially be exploited.

2.3.8 Severity:

Using the formula (Criticality + Lethality) – (System + Network Countermea sures) = Severity, the following analysis was made:

Criticality: In this case, the target is unknown = 5
Lethality: Again, an unknown = 5

Total = 10

System Countermeasures: Most likely not running a service on port 22227 = 5
Network Countermeasures: Por t likely blocked by firewall = 5

Total = 10

Severity: **(10) – (10) = 0**

2.3.9 Defensive Recommendation:

Defensive measures are obvious: apply patches to systems as soon as they are released by the vendor, turn off unused services on servers, and block all unne cessary ports at the firewall and border routers.

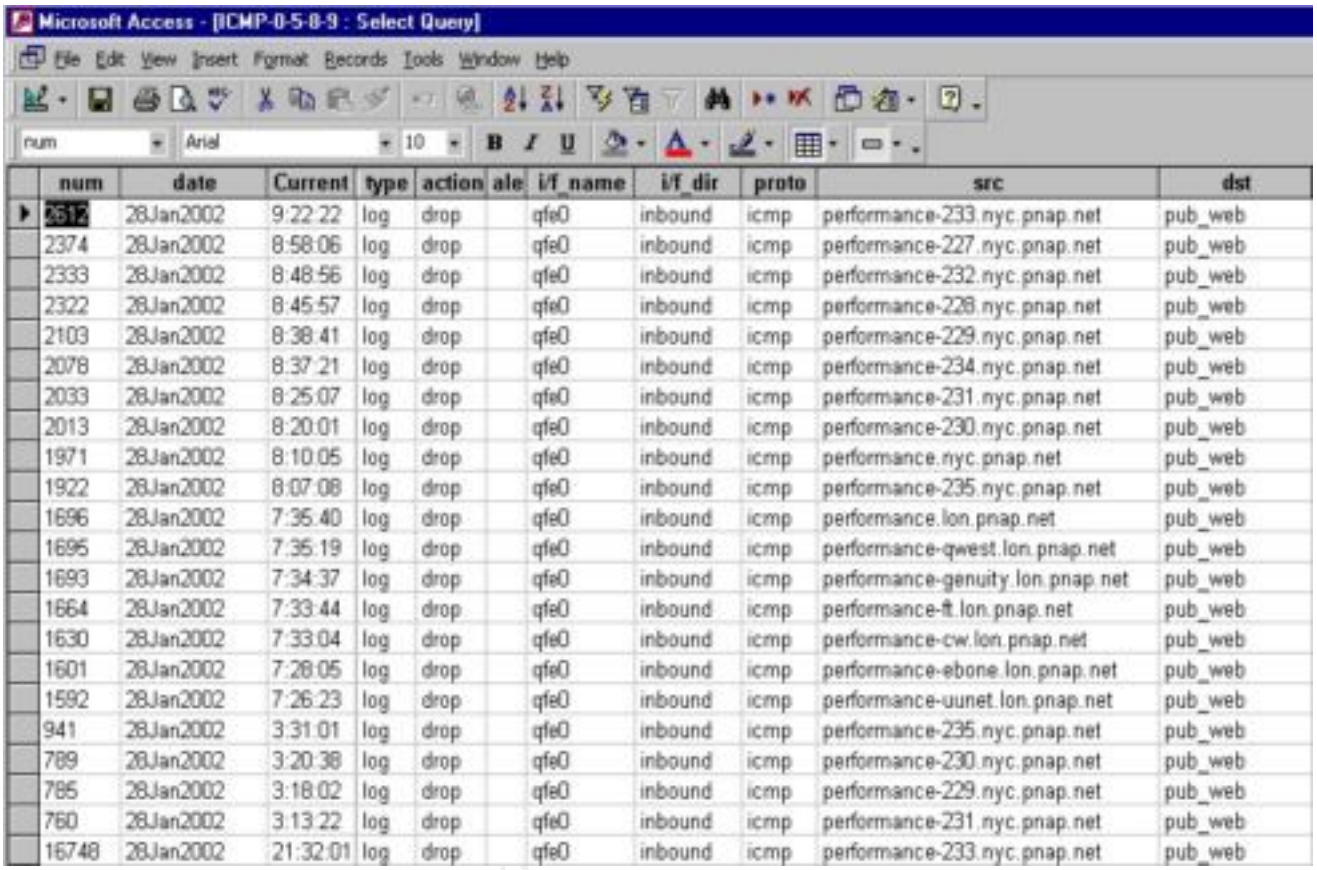
2.3.10 Multiple Choice Test Question:

Your logs show a number of packets that have the same source IP to multiple destination IPs on your network, but all having the same *source* and *destination* ports that are not well-known ports. What would be the likely cause of this?

A) This is normal TCP/IP traffic, the duplicate source and destination ports is a fluke.
B) A duplication of source and destination ports is sometimes caused by fragmented packets.
C) Duplicate source and destination ports in multiple packets is a common indication of forged packets.
D) Duplicate source and destination ports is typical of Trojan Horse traffic.

Correct Answer: C

2.4 Detect Four



num	date	Current	type	action	ale	i/f_name	i/f_dir	proto	src	dst
3812	28Jan2002	9:22:22	log	drop		qfe0	inbound	icmp	performance-233.nyc.pnap.net	pub_web
2374	28Jan2002	8:58:06	log	drop		qfe0	inbound	icmp	performance-227.nyc.pnap.net	pub_web
2333	28Jan2002	8:48:56	log	drop		qfe0	inbound	icmp	performance-232.nyc.pnap.net	pub_web
2322	28Jan2002	8:45:57	log	drop		qfe0	inbound	icmp	performance-228.nyc.pnap.net	pub_web
2103	28Jan2002	8:38:41	log	drop		qfe0	inbound	icmp	performance-229.nyc.pnap.net	pub_web
2078	28Jan2002	8:37:21	log	drop		qfe0	inbound	icmp	performance-234.nyc.pnap.net	pub_web
2033	28Jan2002	8:25:07	log	drop		qfe0	inbound	icmp	performance-231.nyc.pnap.net	pub_web
2013	28Jan2002	8:20:01	log	drop		qfe0	inbound	icmp	performance-230.nyc.pnap.net	pub_web
1971	28Jan2002	8:10:05	log	drop		qfe0	inbound	icmp	performance.nyc.pnap.net	pub_web
1922	28Jan2002	8:07:08	log	drop		qfe0	inbound	icmp	performance-235.nyc.pnap.net	pub_web
1696	28Jan2002	7:35:40	log	drop		qfe0	inbound	icmp	performance.lon.pnap.net	pub_web
1695	28Jan2002	7:35:19	log	drop		qfe0	inbound	icmp	performance-qwest.lon.pnap.net	pub_web
1693	28Jan2002	7:34:37	log	drop		qfe0	inbound	icmp	performance-genuity.lon.pnap.net	pub_web
1664	28Jan2002	7:33:44	log	drop		qfe0	inbound	icmp	performance-ft.lon.pnap.net	pub_web
1630	28Jan2002	7:33:04	log	drop		qfe0	inbound	icmp	performance-cw.lon.pnap.net	pub_web
1601	28Jan2002	7:26:05	log	drop		qfe0	inbound	icmp	performance-ebone.lon.pnap.net	pub_web
1592	28Jan2002	7:26:23	log	drop		qfe0	inbound	icmp	performance-uunet.lon.pnap.net	pub_web
941	28Jan2002	3:31:01	log	drop		qfe0	inbound	icmp	performance-235.nyc.pnap.net	pub_web
789	28Jan2002	3:20:38	log	drop		qfe0	inbound	icmp	performance-230.nyc.pnap.net	pub_web
785	28Jan2002	3:18:02	log	drop		qfe0	inbound	icmp	performance-229.nyc.pnap.net	pub_web
760	28Jan2002	3:13:22	log	drop		qfe0	inbound	icmp	performance-231.nyc.pnap.net	pub_web
16748	28Jan2002	21:32:01	log	drop		qfe0	inbound	icmp	performance-233.nyc.pnap.net	pub_web

Fig. 2.4.1 Sample Log Entry

2.4.1 Source of Trace

This detect was mad e in a network with two web servers within a DMZ. The perimeter is defended by Check Point Firewall -1. The log entries above are a sample of recurring log entries detected over a period of one week.

2.4.2 Detect Generated By:

The log entry exported above (Fig. 2.4.1) is from a Check Point Firewall -1 log that has been exported into a text file using the Check Point “fw logexport” utility. The resulting text file was then imported into a Microsoft Access database used for log file analysis. A query was run to select ICMP traffic from the logfile, and further to select records having a source of “performance*”.

The important fields in the log file above are **date**, which is self-explanatory, **current**, which is the time, **proto**, the protocol, in this case ICMP, **src**, the source address which has been resolved, and **dst**, the destination address, which has been sanitized.

2.4.3 Probability of Spoofed Source Address:

A first analysis does not lend itself to a conclusion one way or another in this regard, but further investigation produced a definite decision that this is not a spoofed address.

2.4.4 Description of Attack:

Over a period of one week, between 500 and 750 ICMP echo requests were sent from one domain to these web servers and dropped by the firewall. The ICMP dro ps originated from hosts in the pnap.net domain and were observed in the firewall logs on a daily basis. Since ICMP is dropped at the firewall, the amount of packets was not enough for this to be considered a DoS attack, and there was no other traffic fro m this domain, this was not considered to be a top priority or very harmful traffic.

2.4.5 Attack Mechanism:

One ICMP echo request was sent approximately every 20 minutes (averaged over the period of one day) to the web servers and each was dropped by the f irewall. Firewall-1 does not allow an examination of the packet itself. Because the ICMP echo requests were sent fairly regularly, and no evidence of additional scanning or other types of traffic from this domain appeared in the logs, it did not appear t o be particularly hostile traffic. However, it did deserve follow-up attention, and as soon as time was available an investigation was conducted.

2.4.6 Correlations:

No correlation was available.

2.4.7 Evidence of Active Targeting:

The ICMP packets were directed a t the web servers and no scanning activity of the other servers occurred. It was determined that the traffic was directed intentionally at the web servers.

2.4.8 Severity

Using the formula (Criticality + Lethality) – (System + Network Countermeasures) = Severity, the following analysis was made:

Criticality: Potential for DoS (if a significant increase in this sort of traffic were to occur, causing web services to be unavailable) = 4
Lethality: Potential DoS or reconnaissance = 4

Total = 8

System Counterme asures: Nothing more can be done to protect the host system = 5
Network Countermeasures: ICMP traffic dropped by the firewall = 5

Total = 10

Severity: **(8) – (10) = - 2**

2.4.9 Defensive Recommendation:

This turned out to be harmless traffic, but if it had not been, the defensive countermeasure recommended to protect against harmful ICMP traffic would be to drop all unnecessary ICMP traffic at the border routers and firewalls, and employ IDS systems to detect harmful ICMP traffic. All ICMP traffic is dropped by the firewall and a network IDS is in place with a rule set to trigger an event for harmful/unusual ICMP traffic.

A further investigation was conducted, and a query made using the whois service at Network Solutions. This revealed that the administrator of the domain pnap.net contact could be reached at an email address with the suffix "@internap.com". Fig. 2.4.9 displays the results of the pnap.net name lookup. The website http://www.internap.com was contacted, and revealed that Internap provided a service to map the quickest route to certain hosts on the Internet for its clients. Below is an excerpt from their web page under the "About Internap" section:

> We have created a platform to intelligently route data over the Internet's major backbones from a single connection to one of our Service Points. We achieve this by connecting directly to each of the major backbones, thereby avoiding the congested traffic exchange process, while using our intelligent routing technology to find the most direct path across the public infrastructure.

As the traffic was not directed at an Internap customer, and the IC MP echo requests they were sending were being dropped and therefore probably were not revealing any pertinent information to Internap, it was decided that Internap should be contacted. Internap's contact information web page had an email address for getting in touch with them about security issues. A quick, polite email was sent, enquiring whether there was any way that the web servers be removed from their list of hosts being pinged. An equally polite email was received promptly, with details about the removal process. An appropriate request was submitted, and within 48 hours the offending traffic no longer appeared in the firewall log files.

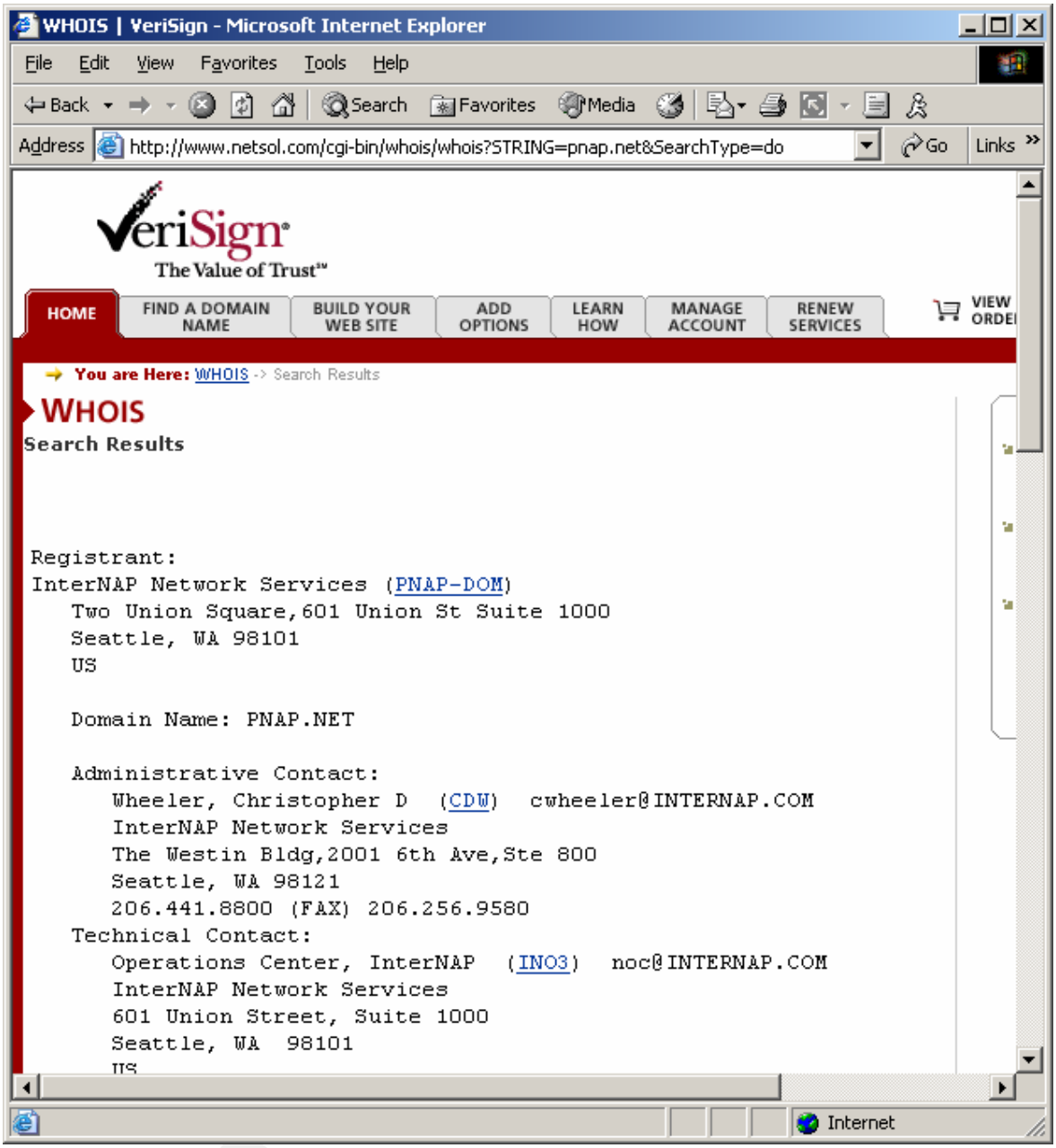


Fig. 2.4.9 Network Solutions whois registratio n for pnap.net

2.4.10 Multiple Choice Test Question:

Which of the following would NOT be useful for guarding against harmful ICMP traffic?

A) Dropping unnecessary ICMP traffic at the firewall
B) Allowing only PING traffic through the firewall
C) Employing an IDS that is able to detect malicious ICMP traffic
D) Dropping unnecessary ICMP traffic at the border routers

Correct Answer: B

2.5 Detect Five

```
Dec 22 00:18:12 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 00:58:13 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 01:38:14 ho sty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 02:18:14 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification :
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 02:58:15 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 2 2 03:38:15 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 04:18:15 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [C lassification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 04:58:16 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34

Dec 22 05:38:17 hosty snort: [ID 702911 local0.alert] [1:522:1] MISC Tiny Fragments [Classification:
Potentially Bad Traffic] [Priority: 2]: {ICMP} 211.13.231.126 > z.y.x.34
```

Fig. 2.5 Excerpt from Incidents.org archive post http://www.incidents.org/archives/intrusions/msg03048.html

2.5.1 Source of Trace

A detect taken from the Incidents.org archives, posted 26 Dec 2001, which can be found at the following URL:

http://www.incidents.org/archives/intrusions/msg03048.html

2.5.2 Detect Generated By:

The submission appears to be from Snort logs.

2.5.3 Probability of Spoofed IP Address

Unlikely. Tiny fragments activity in this amount (the traffic continued for days) is an indication of possible hostile activity.

2.5.4/5 Description of Attack and Attack Mechanism

RFC 1858 gives the following description of Tiny Fragment Attack:

> With many IP implementatio ns it is possible to impose an unusually small fragment size on outgoing packets. If the fragment size is made small enough to force some of a TCP packet's TCP header fields into the second fragment, filter rules that specify patterns for those fields wil l not match. If the filtering implementation does not enforce a minimum fragment size, a disallowed packet might be passed because it didn't hit a match in the filter.

In other words, the Tiny Fragment Attack is a method of bypassing a firewall rule base by sending a malicious packet with an unusually small fragment size. When the firewall or other filter drops the initial connection attempt, the next part of the fragmented packet will be ignored by some systems. This only need be successful on the firs t communication attempt, as once communication has been established it is viewed as legal communication by the firewall.

2.5.6 Correlations

Request for Comments (RFC) 1858, dated Oct 1995, discusses this attack in detail. The full RFC can be found here: http://www.faqs.org/rfcs/rfc1858.html . Further discussion regarding Protection Against a variant of the Tiny Fragment Attack can be found in RFC 3128, dated June 2001, found here: http://www.faqs.org/rfcs/rfc3128.html .

2.5.7 Evidence of Active Targeting

This would appear to be active targeting, because the traffic was all destined for one host. However, it is unknown if this is indeed malicious traffic, and it is not possible to tell based on the submission, although this traffic is certainly suspicious.

2.5.8 Severity

Using the formula (Criticality + Lethality) – (System + Network Countermeasures) = Severity, the following analysis was made:

Criticality: Target is unknown = 3
Lethality: Potentially could allow malicious communication to appear legitimate = 5

Total = 8

System Countermeasures: Systems likely patched = 5
Network Countermeasures: IDS detected malicious fragments = 5

Total = 10

Severity: **(8) – (10) = - 2**

2.5.9 Defensive Recommendations

Defensive countermeasures in this case would be to apply patches to systems as soon as they are released by the vendor, turn off unused services on servers, and block all unnecessary ports at the firewall and bor der routers.

2.5.10 Multiple Choice Test Question:

Which of the following is an example of malicious TCP/IP traffic using fragmentation?

A) Tiny Fragment Attack
B) Queso Attack
C) Frick and Fragment Attack
D) Christmas Tree Attack

Correct Answer: A

Intrusion Detection In-Depth - Assignment Three:

"Analyze This" Scenario - University Security Audit

3 Assignment 3 - "Analyze This" Scenario – University Audit

As requested, an analysis was made of five days worth of data collecte d from a University network's Snort Intrusion Detection System. The analysis was based on data contained in the following files taken from among the files provided by the University:

Data Collected On:	Alert Logs	File Size	Out of Spec (OOS) Logs	File Size	Scan Logs	File Size
22 Dec 2001	alert011222.gz	6 MB	oos_dec.22.2001.gz	80 KB	scans.011222.gz	8 MB
23 Dec 2001	alert011223.gz	4 MB	oos_dec.23.2001.gz	20 KB	scans.011223.gz	5 MB
24 Dec 2001	alert011224.gz	3 MB	oos_dec.24.2001.gz	19 KB	scans.011224.gz	4 MB
25 Dec 2001	alert011225.gz	6 MB	oos_dec.25.2001.gz	2 MB	scans.011225.gz	3 MB
26 Dec 2001	alert011226.gz	11 MB	oos_dec.26.2001.gz	1 KB	scans.011226.gz	10 MB

Fig. 3 Files used for analysis in Analyze This assignment

The files listed above are a repre sentative sample of 5 consecutive days within the last 60 days on the date this assignment was begun, and covers the period including Christmas Day and a good portion of the surrounding holidays. It is common knowledge that suspicious network traffic incr eases during this time because of a combination of school holidays for students and the fact that most IT departments are short -staffed as systems administrators and security personnel go on vacation.

It is assumed therefore that this is a relatively go od sample of data that should contain a low degree of normal network traffic and an increased amount of "bad" traffic that will help to identify areas needing improvement within the University's network.

3.1 Overview

It is interesting to note that one of the most prevalent issues confronting Intrusion Detection today is illustrated in the table (Fig 3) above, namely the amount of log data produced by Intrusion Detection Systems. This log data must be reviewed, sorted through, and analyzed on a regular and rec urring basis in order for the data to have any value.

During the five-day period, during a time of presumably low "normal" network traffic, the Snort IDS generated nearly 62 MB of log files. The file sizes in the table are estimated (rounded downward in each case). The Snort log data generated is, of course, in addition to log files created by other systems on the network such as routers, firewalls, web servers, etc., and during a period in which staff and students' activity on the network should have be en considerably less than normal. It is quite easy to see how an understaffed IT department could easily be overwhelmed by information, and how it would be difficult to dig out from under the log files for long enough to choose a place to begin to improve the security of the network.

The goal of the analysis undertaken and the resulting document is to present a comprehensive summary of the data analyzed with the objective of pointing out areas of vulnerability and identify potential threats in the current state of the University's network, to discuss the steps taken to identify these areas, and to present recommendations to help improve the overall security of the University's network environment.

4 Detects and Statistics

To begin the analysis, the Snort log files were run through a tool called SnortSnarf, available for free from Silicon Defense. This tool is very useful for analyzing data created by the Snort Intrusion Detection System, and runs on many platforms, including Windows and most flavors of Unix, with the only additional requirement being a prior installation of the Perl programming language. A further caveat in the use of SnortSnarf: because of the large size of the log files, SnortSnarf requires a very generous amount of RAM and disk space to analyze the data and store the output files (e.g., this analysis required 512 MB RAM to process the logs, and approximately 1.5 GB of disk space to store the resulting output, and took many hours to complete).

It is highly recommended that SnortSnarf or another analysis tool be incorporated into use in the regular examination of the University's Snort IDS log data. Analyzing such a tremendous amount of data by hand is overwhelming, and undoubtedly leads to missing import information or details. Furthermore, a tool such as SnortSnarf allows a view of the "big picture" that otherwise would not be possible.

In order to obtain a comprehensive overview from SnortSnarf, the five days worth of Snort data were concatenated into one file, respectively, for both the alert log data as well as the scan log data. The log files were modified further, changing the first two octets of the University's network IP addresses from "MY.NET." (as found in the log data provided by the University), to "10.253.", after a scan of the logs to make sure that this address prefix did not occur within the log files prior to the modification. This change was made so that SnortSnarf could perform a more accurate analysis of IP addresses. Therefore, in the analysis that follows, it can be assumed that any "10.253.x.x" IP address belongs to the University's network.

Below is an overview of alert signatures generated by the Snort IDS during the period in question, prioritized from least to most number of alerts. The snort alert log files identified 127 unique alert signatures with a total of 221,217 alerts from 10,424 source hosts to 13,805 destination hosts. This is a rather large number of unique alerts for such a short time span, and the number of alerts generated should be diminished considerably through defensive countermeasures.

Following is a summary of the log data and the events detected prioritized by their frequency. After this summary section will be an analysis of some of the information from the logged data. This analysis should be used as a sample of analysis methodology and this process can then be extended and applied by University personnel to further secure the network environment. Where appropriate, e.g., to provide further insight into motive, an external source IP address may be looked up in an appropriate whois database. Additionally, defensive countermeasures will be suggested and correlating data will be offered where possible.

4.1 Alert Log Data

The following table is a summary of the alerts log data generated by Snort, organized from least number of alerts to greatest:

No.	Signature	No. of Alerts	No. of Sources	No. of Destinations
1	ICMP Redirect (Undefined Code!)	1	1	1
2	FTP RETR 1MB possible warez site	1	1	1
3	SCAN XMAS	1	1	1
4	SCAN - wayboard request - allows reading of arbitrary files as http service	1	1	1
5	WEB-FRONTPAGE shtml.dll	1	1	1
6	INFO – Web Dir listing	1	1	1
7	ICMP Photuris (Undefined Code!)	1	1	1
8	External FTP to HelpDesk 10.253. 83.197	1	1	1
9	FTP passwd attempt	1	1	1
10	ICMP Reserved for Security (Type 19) (Undefined Code!)	1	1	1
11	External FTP to HelpDesk 10.253.70.50	1	1	1
12	EXPLOIT x86 stealth noop	1	1	1
13	WEB-MISC guestbook.cgi access	2	2	1
14	DDOS mstream handler to client	2	1	1
15	WEB-CGI glimpse access	2	1	1
16	WEB-IIS .cnf access	2	2	2
17	WEB-CGI tsch access	2	2	1
18	TFTP - External UDP connection to internal tftp server	2	1	2
19	WEB-CGI ksh access	2	2	2
20	External FTP to HelpDesk 10.253.70.49	2	1	1
21	INFO - Web Command Error	2	1	2
22	WEB-CGI survey.cgi access	2	2	2
23	FTP CWD / - possible warez site	2	1	1
24	x86 NOOP - unicode BUFFER OVERFLOW ATTACK	2	2	2
25	Attempted Sun RPC high port access	3	1	1
26	MISC solaris 2.5 backdoor attempt	3	2	1
27	WEB-CGI csh access	3	3	3
28	Virus - Possible MyRomeo Worm	3	3	3
29	WEB-CGI finger access	3	3	1
30	RFB - Possible WinVNC - 010708-1	4	2	2
31	INFO napster login	4	1	4
32	IDS50/trojan_trojan-active-subseven [arachNIDS]	4	1	1
33	ICMP Destination Unreach able (Network Unreachable)	4	1	1
34	WEB-MISC /....	4	1	1
35	MISC PCAnywhere Startup	4	2	3
36	WEB-MISC compaq nsight directory traversal	5	2	2
37	EXPLOIT x86 setuid 0	5	4	4
38	WEB-FRONTPAGE fpcount.exe access	5	2	2
39	spp_http_decode: CGI Null Byte at tack detected	5	3	3
40	WEB-CGI scriptalias access	5	2	2
41	IDS475/web-iis_web-webdav-propfind [arachNIDS]	5	1	1
42	WEB-MISC Lotus Domino directory traversal	6	4	2
43	WEB-CGI archie access	6	4	2

No.	Signature	No. of Alerts	No. of Sources	No. of Destinations
44	WEB-FRONTPAGE shtml.exe	7	3	1
45	X11 outgoing	7	3	5
46	SMTP chameleon overflow	7	7	4
47	WEB-FRONTPAGE posting	7	2	1
48	EXPLOIT x86 setgid 0	7	4	4
49	Virus - Possible pif Worm	8	2	2
50	WEB-IIS File permission canonicalization	8	2	2
51	SNMP public access	10	2	8
52	beetle.ucs	11	4	6
53	DNS zone transfer	11	3	3
54	SCAN Synscan Portscan ID 19104	11	11	7
55	Virus - Possible scr Worm	12	5	6
56	High port 65535 udp - possible Red Worm - traffic	12	4	3
57	Port 55850 udp - Possible myserver a ctivity - ref. 010313-1	13	1	1
58	SUNRPC highport access!	14	2	2
59	ICMP redirect (Host)	15	1	1
60	TELNET access	16	1	12
61	MISC Large ICMP Packet	17	13	6
62	WEB-CGI rsh access	17	6	3
63	WEB-CGI formmail access	18	13	6
64	INFO - Web Cmd completed	19	3	6
65	SCAN FIN	19	1	6
66	Possible trojan server activity	20	6	7
67	DDOS shaft client to handler	25	1	1
68	ICMP Echo Request L3retriever Ping	31	6	6
69	INFO Inbound GNUTella Connect request	37	21	5
70	WEB-CGI redirect access	38	20	5
71	WEB-IIS Unauthorized IP Access Attempt	38	2	22
72	EXPLOIT x86 NOOP	49	6	7
73	High port 65535 tcp - possible Red Worm - traffic	54	12	13
74	INFO Napster Client Data	55	26	40
75	WEB-FRONTPAGE _vti_rpc access	55	24	7
76	WEB-IIS _vti_inf access	60	27	7
77	TFTP - Internal TCP connection to external tftp server	62	2	2
78	NMAP TCP ping!	65	16	11
79	ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)	65	48	5
80	connect to 515 from inside	66	1	1
81	WEB-MISC count.cgi access	77	39	2
82	WEB-IIS view source via translate header	88	11	7
83	Port 55850 tcp - Possible myserver activity - ref. 010313-1	93	16	17
84	WEB-MISC http directory traversal	109	45	3
85	CS WEBSERVER - external ftp traffic	109	30	1
86	connect to 515 from outside	110	3	107
87	ICMP Echo Request CyberKit 2.2 Windows	158	44	5
88	INFO Outbound GNUTella Connect accept	169	151	19
89	INFO - Possible Squid Scan	211	9	11
90	Null scan!	211	64	21
91	TELNET login incorrect	222	10	149
92	INFO Possible IRC Access	263	41	50
93	FTP DoS ftpd globbing	278	10	10

No.	Signature	No. of Alerts	No. of Sources	No. of Destinations
94	ICMP Echo Request Windows	336	73	49
95	ICMP traceroute	351	98	192
96	spp_http_decode: IIS Unicode attack detected	395	88	35
97	TCP SRC and DST outside network	412	42	170
98	Tiny Fragments - Possible Hostile Activity	491	5	4
99	ICMP Echo Request Sun Solaris	496	11	451
100	WEB-MISC 403 Forbidden	511	11	264
101	INFO Inbound GNUTella Connect accept	534	14	461
102	Incomplete Packet Fragments Discarded	544	15	5
103	SMB Name Wildcard	573	83	228
104	WEB-MISC Attempt to execute cmd	632	73	32
105	SMTP relaying denied	838	11	22
106	ICMP Destination Unreachable (Protocol Unreachable)	920	14	56
107	INFO FTP anonymous FTP	1054	183	107
108	BACKDOOR NetMetro Incoming Traffic	1097	2	2
109	ICMP Echo Request Nmap or HPING2	1218	18	13
110	External RPC call	1256	3	824
111	Watchlist 000222 NET -NCFC	1980	21	16
112	ICMP Fragment Reassembly Time Exceeded	2249	12	37
113	ICMP Destination Unreachable (Host Unreachable)	3447	305	27
114	BACKDOOR NetMetro File List	3586	1	1
115	ICMP Destination Unreachable (Communication Administratively Prohibited)	4681	56	50
116	SYN-FIN scan!	5026	1	5026
117	ICMP Source Quench	5111	25	93
118	Queso fingerprint	5132	34	26
119	SCAN Proxy attempt	5753	61	4669
120	MISC Large UDP Packet	7748	27	4
121	WEB-MISC prefix-get //	9644	571	3
122	INFO MSN IM Chat data	10305	145	195
123	ICMP Echo Request BSDtype	11550	19	9
124	MISC source port 53 to <1024	16955	4019	8
125	CS WEBSERVER - external web traffic	18080	3438	1
126	MISC traceroute	32793	67	7
127	Watchlist 000220 IL -ISDNNET-990517	62318	22	13
127	Total	221217	10424	13805

Fig. 4.1 Summary of Alert Log Data

4.1.1 Alert Log Data "Top 10 Talkers"

The following tables (Fig 4.1.1a and 4.1. 1b) are the top ten source and destination IP addresses from the Snort alert log information provided by the University:

Top 10 Source IP addresses

No.	*Number of Alerts*	*IP Address*	*Number of signatures*	*Destination IP(s)*
1	61327 alerts	212.179.35.118	1 signatures	(3 destination IPs)
2	5648 alerts	216.106.172.149	2 signatures	10.253.153.210
3	5027 alerts	24.0.28.234	2 signatures	(5027 destination IPs)
4	5026 alerts	10.253.5.13	1 signatures	(90 destination IPs)
5	4908 alerts	206.65.191.129	3 signatures	10.253.98.177, 10.25 3.98.187
6	4668 alerts	65.165.14.43	3 signatures	(4632 destination IPs)
7	3667 alerts	10.253.60.11	4 signatures	(44 destination IPs)
8	3661 alerts	65.207.94.30	1 signatures	10.253.137.7
9	3610 alerts	128.223.4.21	3 signatures	10.253.70.148
10	3460 alerts	141.213.11.120	2 signatures	10.253.70.148

Fig. 4.1.1a Top 10 Source IP Addresses

Top 10 Destination IP addresses

No.	*Number of Alerts*	*IP Address*	*Number of signatures*	*Source IP(s)*
1	62875 alerts	10.253.70.70	12 signatures	(383 source IPs)
2	34204 alerts	10.253.140.9	4 signatures	(66 source IPs)
3	18947 alerts	10.253.100.165	21 signatures	(3471 source IPs)
4	11085 alerts	10.253.253.114	16 signatures	(573 source IPs)
5	10322 alerts	10.253.70.148	10 signatures	(45 source IPs)
6	8010 alerts	10.253.153.210	3 signatures	(3 source IPs)
7	6768 alerts	10.253.1.3	5 signatures	(2064 source IPs)
8	4871 alerts	10.253.1.5	3 signatures	(1490 source IPs)
9	4642 alerts	10.253.98.177	6 signatures	(32 source IPs)
10	4516 alerts	10.253.1.4	2 signatures	(1518 source IPs)

Fig. 4.1.1b Top 10 Destination IP Addresses

4.2 Scan Data

The log data provided by the University's Snort IDS indicated that scanning activity triggered over 500,000 alerts. On two single days, Christmas Day and the day after, there were 142,427 UDP scan alerts from 105 sources to over 20,000 destination hosts. The following were the types of scans that triggered the most alerts:

- UDP scan
- TCP *******F scan
- TCP *2U***SF scan
- TCP 12****S* scan
- TCP ******S* scan
- TCP ****P*** scan
- TCP ******** scan
- TCP ******SF scan

The following tables list the top 10 source (Fig. 4.2a) and destination (Fig. 4.2b) hosts, in terms of alerts triggered, discovered during the analysis of the scan logs.

No	Number of Alerts	Source IP Address	Signatures	Destinations
1	331649 alerts	10.253.87.50	1	Multiple Destination IPs
2	27085 alerts	10.253.98.203	1	Multiple Destinati on Ips
3	9876 alerts	211.248.231.10	1	Multiple Destination Ips
4	9508 alerts	65.165.14.43	1	Multiple Destination Ips
5	7952 alerts	210.77.145.30	1	Multiple Destination Ips
6	7680 alerts	210.58.102.86	1	Multiple Destination Ips
7	5412 alerts	24.44.21.206	1	Multiple Destination Ips
8	5072 alerts	24.0.28.234	2	Multiple Destination Ips
9	5483 alerts	204.152.184.75	1	10.253.70.148
10	4045 alerts	10.253.84.185	2	Multiple Destination IPs

Fig. 4.2a Top 10 Source hosts ordered by number of alerts

No.	Number of Alerts	Destination IP Address	Signatures	Originating Source IP
1	20604 alerts	24.164.41.2 10	1 signature	10.253.87.50
2	11066 alerts	216.33.98.254	1 signature	10.253.98.203
3	7144 alerts	194.251.249.182	1 signature	10.253.98.203
4	6583 alerts	10.253.70.148	1 signature	(4 source IPs)
5	4942 alerts	24.23.140.185	1 signature	10.253.87.50
6	4428 alerts	24.197.48.74	1 signature	10.253.98.203
7	3377 alerts	24.254.241.95	1 signature	10.253.87.50
8	3453 alerts	10.253.98.177	4 signatures	206.65.191.129, 207.71.92.221
9	2640 alerts	24.203.36.254	1 signature	10.253.87.50
10	2433 alerts	168.73.245.58	1 signature	10.253.60.38

Fig. 4.2b Top 10 Destination hosts ordered by number of alerts

5 Detailed Analysis of Alert Log Data

This section will discuss a portion of the alerts generated by the Snort IDS system during the period in question, 22-26 Dec 2001. As outlined in section 4 above, 127 unique alert signatures were triggered by the IDS. Because of the sheer number and variety of alerts, a detailed analysis of each would require a tremendous amount of time to thoroughly compose and would require an almost equal effort to read and digest by the recipient of this report.

Instead of analyzing each alert, a representative sampling will be taken and discussed in-depth, and correlations and countermeasures will be discussed for each. It is hoped that this will provide methodology for internal University personnel to further analyze data from their day-to-day Snort logs and that the recommended countermeasures can be adapted to other threats detected during this regular and frequent analysis.

5.1 Watchlist 000220 IL-ISDNNET-990517 and Watchlist 000222 NET-NCFC

With 62,318 alerts triggered, the Watchlist 000220 signature takes the grand prize. These "Watchlist" signatures contain IP addresses that are known for initiating suspicious activity, and it is therefore troubling that such a large amount of traffic from potentially hostile sources is bound for the University's network.

By far, the number one source of traffic triggering this alert was the IP address 212.179.35.118, with a total of 61,327 individual alerts (also listed in the "pole position" of the "Top 10 Source IP Addresses" section). Below is an excerpt taken from the RIPE (RIPE Network Coordination Centre) database entry for this Israeli IP address:

inetnum:	***212.179.35.96 - 212.179.35.127***	***person:***	***Zehavit Vigder***
netname:	EPLICATION-LTD	address:	bezeq-international
mnt-by:	INET-MGR	address:	40 hashacham
descr:	EPLICATION-LTD-HOSTING	address:	petach tikva 49170 Israel
country:	IL	phone:	+972 52 770145
admin-c:	ZV140-RIPE	fax-no:	+972 9 8940763
tech-c:	MZ4647-RIPE	e-mail:	hostmaster@bezeqint.net
status:	ASSIGNED PA	nic-hdl:	ZV140-RIPE
notify:	hostmaster@isdn.net.il	changed:	zehavitv@bezeqint.net 20000528
changed:	hostmaster@isdn.net.il 20020312	source:	RIPE
source:	RIPE		

Fig. 5.1 RIPE whois database search results for IP address 212.179.35.11 8

There were two destination IP addresses on the University's network targeted by this source address, both with the same destination port.

Destination IP Addresses:	Port
10.253.70.70	1214
10.253.99.39	1214

Fig. 5.1 Destination IP addresses from 212.179.35.118

The IP address 10.253.70.70 had far more "hits" than did the other IP address, and also holds the top spot on the "Top 10 Destination IP Address Talkers" list.

Port 1214 is listed on the Internet Assigned Numbers Authority website of port listings as the port for KAZAA, the popular peer-to-peer file-swapping service. Similar to the currently-offline service called Napster, KA ZAA is primarily used as a music -swapping utility, but can also be used to swap mpegs, software, and other files.

This would indicate that the machines with the IP addresses listed above are certainly part of a peer-to-peer file-sharing network. What sort of files are being shared is unknown. While the courts are still debating the legality of music file sharing in this manner, the files being shared may be other files in which the copyright is not in question and could potentially present a legal iss ue for the University. Additionally, applications such as KAZAA create a high amount of network traffic, potentially presenting issues with regard to bandwidth resources available for legitimate activity.

The University's policy on this sort of activit y should be reviewed, and it is recommended that these machines be taken off -line and examined thoroughly for the content that is being shared as well as for other potential security vulnerabilities.

The Watchlist 000222 NET -NCFC signature contained en tries from 21 source IP addresses in the Class B range of 159.226.x.x. Below is an excerpt from the ARIN (American Registry for Internet Numbers) whois database for the net block 159.226.0.0 – 159.226.255.255:

The Computer Network Center Chinese Academy of Sciences (NET -NCFC)
P.O. Box 2704 -10,
Institute of Computing Technology Chinese Academy of Sciences
Beijing 1000 80, China
Netname: NCFC Netblock: 159.226.0.0 - 159.226.255.255
Coordinator: Qian, Haulin (QH3 -ARIN) hlqian@NS.CNC.AC.CN
+86 1 2569960

Fig. 5.1.1 ARIN whoi s database registration for netblock 159.226.0.0 – 159.226.255.255

While there were a total of 16 target destinations for traffic from IP addresses on this watchlist, by far the most traffic was bound for the University's internal IP address of 10.253.253.114, port 80 (http). This IP address is also number four on the list of Top 10 Destination IP addresses.

It would appear that this is a web server running IIS, as a search for correlating information for this as a destination address revealed that 16 different attack signatures were detected, among them instances of an IIS Unicode attack, possible Red Worm traffic, as well as attempts to execute commands.

It is possible that this web server has been compromised. It is recommended that, at a minimum, this machine be taken off -line and procedures recommended by Microsoft for securing IIS web servers be undertaken. This server should also be checked for further security vulnerabilities. The IIS lockdown tool is available from Microsoft at the follow ing URL:

http://www.microsoft.com/downloads/release.asp?ReleaseID=33961&area=search&ordinal=2

And information about the Code Red Worm and patch is av ailable here:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/codealrt.asp?frame=true

Additionally, due to the extremely high number of visits to the University's network by suspect source addresses, blocking addresses known for malicious activity should be taken into consideration.

5.2 Miscellaneous Traceroute and ICMP Echo Request BSDType

Traceroute is a tool that is used to determine routes taken by packets on their journey to a particular host system. According to an article by Christopher Low in the System Administration, Network and Security (SANS) reading room (http://rr.sans.org/threats/ICMP_attacks.php), the traceroute command can be very useful for mapping out a potential target network because it not only gives information regarding the path a packet takes to reach a host, it illum inates information regarding the network topology as well. Information such as the IP addresses of border routers and other intermediate hosts between the source and destination IP addresses can be gleaned using this tool.

The host 10.253.140.9 is the t op destination for this event (34,204 alerts) and is the number two destination host in the Top 10 Talkers list. The second on the list of destination hosts for the Miscellaneous Traceroute alert, the host 10.253.70.148, is the fifth most popular destinat ion host in the Top 10 Talkers list.

Echo Requests such as those generated by ping, a tool generally used to verify that a host is available by sending a request that is answered at a very low IP level, can also be used in reconnaissance activities. Th is reconnaissance process can also be automated using tools such as nmap or superscan. Over 11,000 alerts were generated by ICMP echo request activity, and three of the top -ten initiators of this activity were located on the University's internal network: 10.253.60.39, 10.253.101.142, and 10.253.60.11. 10.253.60.11 is also number 7 on the list of source addresses of the Top 10 Talkers.

Blocking ICMP incoming echo requests such as ping and traceroute is recommended in the SANS "How To Eliminate The Ten Most Critical Internet Security Threats" document (http://www.sans.org/topten.htm) in the Appendix entitled "Perimeter Protection For An Added Layer of Defense In Depth."

It is recommended that, in order t o guard against potential malicious activity, blocking ICMP requests at the border routers/firewalls of the University network seriously be taken into consideration. Because internal hosts triggered some of these events, egress filtering (i.e., blocking outbound traffic) for ICMP should also be considered. There are, of course, legitimate uses for the traceroute and other ICMP tools and the consequences of removing the ability to use them should be weighed against the security benefit that would be gained.

5.3 INFO MSN IM Chat Data

Microsoft Network Messenger is a tool used for instant messaging, but also can be used to share files. This traffic is completely normal network traffic unless it is prohibited by the University's security policy. Multiple hosts on the University's internal network are heavy MSN Messenger users, particularly 10.253.98.200 and 10.253.98.19. The host 10.253.98.177 is also a significant MSN Messenger user and is number 9 on the Top 10 Destination IP address list.

Further examination of the host 10.253.98.177 as a destination indicates that this host was singled out for Queso fingerprinting (see section 5.6).

5.4 MISC Large UDP Packet and Incomplete Packet Fragments Discarded

There were almost 8000 alerts generated by the miscellaneo us large UDP packet signature, and although there were 27 individual sources, the overwhelming majority of the alerts were from two specific hosts. There were 4 destination hosts targeted, all located within the University's network. The following table (Fig. 5.1.4) lists the two hosts generating nearly all of this traffic, and the single most popular destination host and the UPD destination ports:

No.	Source Hosts	Destination Host	UDP Destination Ports
1	216.106.172.149	10.253.153.210	3816, 3872, 38 88, 1434
2	61.219.53.135		

Fig. 5.4 Misc Large UDP Packet source and destination hosts

Host 216.106.172.149 is number two on our source Top Ten talkers, and host 10.253.153.210 is number six on our list of Top Ten destination hosts. Following is an excerpt of a search of the ARIN (American Registry for Internet Numbers) whois database for the host 216.106.172.149, and a search of the APNIC (Asian-Pacific Network Information Center) for the host 61.219.53.135.

216.106.172.149 (Source: ARIN)	***61.219.53.135 (Source: APNIC)***
iBEAM Broadcasting Corporation	Chunghwa Telecom Co., Ltd.
Netname: IBEAM	netname HINET-TW
Netblock: 216.106.160.0 - 216.106.175.255	Netblock: 61.216.0.0 - 61.219.255.255
645 Almanor Ave., suite 100	Data-Bldg.6F, No.21, Sec.21, Hsin-Yi Rd.
Sunnyvale, CA 94085 US	Taipei Taiwan

Fig. 5.4a: ARIN And APNIC Whois Database Information

Traffic from source host number two to the destination host number one in Fig. 5.1.4 began on 22 Dec at 17:32 and 20 seconds, and continued with multiple packets (as many as 11 per second) directed at port 3818 until 17:42 and 02 seconds. The traffic stopped for less than one minute, and then the traffic continued from source host one listed in Fig. 5.1.4, directed at the same target but toward UDP port 3872. This traffic continued for approximately 5 minutes, at which time the hosts remained the same but the source and destination ports changed again. The traffic continued for another 5 minutes and then ceased after a total of 20 minutes at approximately 17:52.

On the following day, 23 Dec, at approximately 16:00, traffic from the same source to the same destination resumed, this time with a destination port of 1434. This traffic continued with a few small breaks until 17:21.

Interspersed within this traffic are approximately 500 packets that triggered the Incomplete Packet Fragments Discarded alert. These discarded packets have the same source and destination IP addresses as the other traffic and occurred within the same timeframes. It is assumed that this traffic is a side effect of the large UDP packet traffic.

A search of the IANA (Internet Assigned Numbers Authority) website of TCP and UDP port listings shows all of the destination ports contained in this traffic are currently unassigned. A general search of the Internet using WebFerret and Google also did not turn up any information regarding these ports. It can therefore be assumed that the particular port being targeted was irrelevant. A search of the CERT® Coordination Center (originally the Carnegie-Mellon Emergency Response Team) website also did not reveal any information pertaining to these particular ports or to this type of traffic.

An advisory at the National Information Protection Center (http://www.nipc.gov/warnings/advisories/2001/01-012.htm) addresses potential Denial of Service attacks using large, fragmented UDP packets, although in the advisory this traffic is destined for port 80. The advisory mentions that "… certain major routing equipment manufacturer's products will block the first fragment of a large UDP packet, but may not block subsequent packets, thereby permitting the denial of service to continue" and that "…inbound packets of this type indicate that a denial of service to the network in question may be underway."

If indeed this is a Denial of Service (DoS) attempt, the IP addresses researched using ARIN and APNIC above may have been spoofed. It is recommended that the payload of further instances of this sort of traffic be captured in order to determine what may be

intended by this traffic, a nd that the offending IP addresses be put on a watch list (Steve Lukacs, SANS Practical, http://www.giac.org/GCIA.php).

5.5 Scan Proxy Attempt

There were 5753 alerts with the SCAN proxy attempt signature. The source IP address to which the majority of the attempts can be attributed is 65.165.14.43, with 4668 alerts. This IP address is also number six on our Top Ten source address list. This IP address was researched using the ARIN (American Registry for Internet Numbers) whois database, and the following information was obtained:

SYSTEMS SOLUTIONS INC (NETBLK -FON-110133555275610)
2108 E THOMAS RD
PHOENIX, AZ 85016
US
Netname: FON-110133555275610
Netblock: 65.165.12.0 - 65.165.15.255

Fig. 5.5 ARIN query for 65.165.14.43

For about 20 minutes on 26 Dec (06:47 – 07:06), the host in question scanned 4668 host s on the University's network for the port 1080. Port 1080/tcp is listed at IANA (the Internet Assigned Numbers Authority) as a socks proxy port.

The CERT® Coordination Center contains a vulnerability note numbered VN-98:03 and entitled "WinGate IP Laundering" which discusses a vulnerability on port 1080 in WinGate software from Deerfield. WinGate software allows the sharing of a single Internet connection, while, according to their website, "… protecting the valuable information on the network with the integrated **proxy server/firewall** ."

Proxy servers can be used for many different purposes, but they have the effect of hiding the IP addresses of the computers using the proxy server behind the IP addre ss of the proxy itself. This provides an obvious benefit to a hacker or someone wishing to conduct any sort of anonymous activity. It would appear that this probably is a scan for WinGate proxy servers or other proxy servers having a vulnerability that c ould be exploited and used by the attacker to conceal their true location. Additionally, proxies of this sort could be used to gain access to the server and use that as a point of entry into an internal network environment.

Older versions of WinGate sof tware, in its default configuration, contained a vulnerability that allowed external clients to use it as a proxy server. This meant that hosts on the Internet would appear as if they were behind the proxy server on the internal network. It is possible that this scan was for a WinGate proxy server or some other proxy server that contains a similar vulnerability.

It is not known whether this product is in use in the University's network. If so, the Wingate website offers easily understood information o n securing their product here. If other types

of proxy servers are in use, they should be researched to determine if any a similar vulnerability exists.

5.6 Queso Fingerprint

Queso is a tool used to "fingerprint" a host; it is used to determine what operating system is installed on a system. According to David Leach (SANS Practical, http://www.giac.org/GCIA.php), "TCP packets with the SYN flag and reserved bits 1 and 2 (i.e. S12) trigger this alert. " Fingerprinting is done during the reconnaissance phase of an attack, so that once the operating system on a host has been identified, the attacker may then choose the most effective means of attacking a host system.

Of the 5132 total alerts generated with this signature, 4908 can be attributed to one host, number five on our Top Ten source IP addresses: 206.65.191.129. The troubling aspect of this particular event is that all of this traffic was directed at two hosts: 10.253.98.187 and 10.253.98.177. For some reason, these particular machines have been singled out from the entire class B of the University's network and chosen as targets for further probing.

As some time has passed since this probing occurred, it is unknown if the Queso tool was successful in identifying the host operating system or whether a further attempt was made to compromise these hosts. Further research of the logs of the time frame analyzed only indicate that host 10.253.98.177 is a minor user of MSN Messenger and the GNUTella file-sharing application.

5.7 SYN-FIN Scan!

There were 5026 alerts for the SYN-FIN Scan! signature, all originating from our third most popular source address, 24.0.28.234. On Christmas Day, beginning just before 10 PM, the source address began scanning the University's network. The scan continued until about 10:06 PM, when a series of ICMP Destination Unreachable (Communication Administratively Prohibited) alerts were triggered, originating from University network hosts with the destination address listed above.

A SYN-FIN scan! alert is triggered by TCP packets with both the SYN (synchronization) and FIN (finish) flags set. This configuration is abnormal in TCP traffic because the SYN flag is used in the beginning of communication and the FIN flag at the end and they should not be found together in the same packet. Therefore, it is likely that these packets have been crafted.

The scanning host used a source port of TCP/22 and a destination port of TCP/22, which is used for SSH or the Secure Shell protocol. SSH is a widely used and accepted method of providing a more secure alternative to Telnet and FTP. While it is indeed recommended to use SSH rather than less-secure alternatives, there are multiple vulnerabilities in some versions of SSH, especially version I. SSH (the company) has a good discussion of the vulnerabilities found in SSH at the following two links:

http://www.ssh.com/products/ssh/cert/

http://www.ssh.com/products/ssh/cert/vulnerability.cfm

The CERT® Coordination Center also details multiple vulnerabilities involving SSH, including CERT VU#565052, CERT VU#665372, and CERT VU#25309, among others. Defensive recommendations would be to apply patches or upgrade to the latest version of SSH if it is in use in the University's network.

5.8 BACKDOOR NetMetro File List

Backdoor NetMetro is a Trojan horse - a malicious application that allows a remote computer to control actions on an infected computer. Some of the typical things Trojans allow are the viewing of the remote computer's screen, the viewing of files on the remote computer, shutting down the remote computer, and retrieving information from the remote system (e.g., username, computer name, etc.). Backdoor NetMetro runs on TCP ports 5031 or 5032.

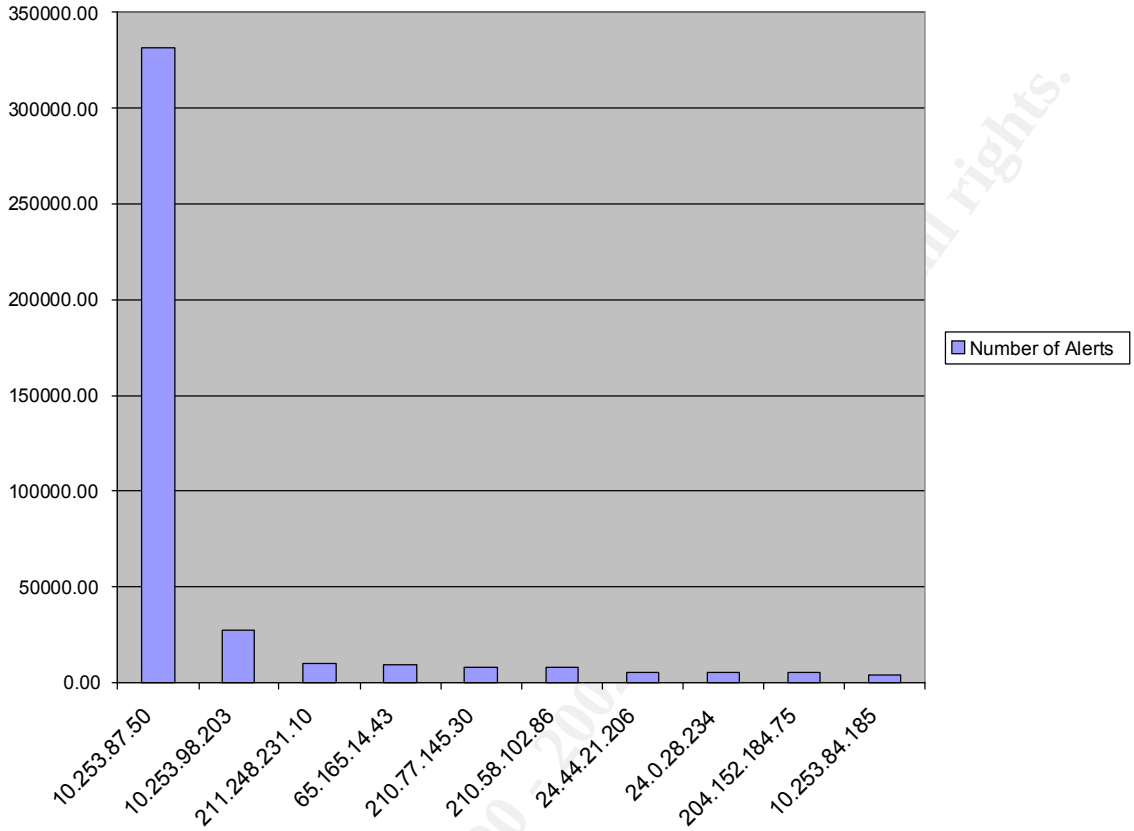
All of the Trojan activity was initiated from within the University network, from one source host, 10.253.60.11, to one destination host, 209.49.12.32. The source host is also number seven of our Top 10 source talkers. This undoubtedly means that the host 209.49.12.32 is infected with the Backdoor.NetMetro Trojan, and that the University's internal host is accessing this computer remotely. This can potentially be a legal issue for the University and should be investigated further. The host in question has also been both the target and source for additional suspicious activity (see section 5.2) and should certainly be examined further. Antivirus software offers protection against known Trojan horses and should be employed on the University's network to protect against them.

6 Analysis of Snort Scan Log Data

The analysis of the Snort scan logs indicated a wide variety and large amount of scanning activity. This activity was both conducted by hosts on the University's network as well as destined for the University's network. In the Detects and Statistics section of this document, the Top Ten sources and destination hosts involved in scanning activity are enumerated.

One host on the University's network, 10.253.87.50, is the source of over 330,000 Snort scan alerts. The second-most frequent source of scan alerts (10.253.98.203) is also on the University's network. The following chart (Fig.6) illustrates the relationship of the number of scans to source hosts.

Fig 6 Number of Alerts by Source



Host 10.253.87.50 scanned hundreds of thousands of different hosts for unassigned UDP ports, for example UDP/27005, UDP/24230, UDP/1869, and UDP/27500. These also do not appear to be active Trojan ports. While it is unclear exactly what the intent of this sc an was, it is definitely highly irregular network traffic. It is recommended that this host, as well as host 10.253.98.203, be removed from the network and thoroughly examined for possible compromise.

On the scan-receiving side, the host 10.253.70.148 on the University network received a high number of scans as well. This host is also on the list of Top Ten destination addresses from the alert logs, most of which were the result of ICMP echo request and trace routes. However, there were also instances of FTP server activity and Red Worm traffic. So much suspicious activity surrounding this host warrants further investigation. The host 10.253.98.177 also merits examination, as it has a similar scan and alert history.

7 Summary and Recommendations

There were a tremendous number of alerts of a remarkable variety generated by the University's Snort IDS. It is understood, and has been taken into consideration, that a University's network has to be very open to allow for a certain amount of academic freedom. However, some of the alerts generated appear to indicate possibly compromised hosts on the University's network or malicious and/or potentially illegal activity originating from the University's network. As a result, the following recommendations be low are given to help further secure the University's network against potential malicious and/or illegal activity. The recommendations should be considered within the framework of the University's existing security and access policies.

- Examine the hosts on the University network that were mentioned in this report for compromise or other security issues and perform the defensive recommendations as noted in the analysis section.

- Employ egress filtering (blocking outbound traffic) where appropriate and possible within the confines of the University's access policy.

- Modify the firewall rule base to restrict access to only those servers and services that are absolutely required, again, if allowed within the framework of the University's access and security po licies.

- Conduct a vulnerability assessment or hire a third party to assess potential vulnerabilities within the University network and apply appropriate remedies for any vulnerabilities discovered.

- Regularly apply vendor patches to all operating syst ems and applications.

- Employ regularly updated virus -scanning software on hosts and servers.

- Add hosts originating suspicious network traffic to the Watchlist signatures (which appear to be already within Snort).

- Maintain a list of top -talkers and inves tigate suspicious traffic generated by these hosts, especially hosts on the University's network.

7.1 False Positives

It is also highly recommended that the IDS rule set be modified to attempt to lessen the number of false positives encountered. For in stance, one of the Top Ten destination hosts (number three) appears to be a legitimate web and ftp server. If this is the case, normal, acceptable traffic destined for this host triggered unnecessary alerts and potentially skewed the Snort alert log data, and as a result may have distracted the security staff from alerts which should take a higher precedence.

Appendix A: Brief Overview of Alert Signatures

Jamil Farshchi, a GIAC Certified Intrusion Analyst, obvious ly spent a tremendous amount of time researching many of the common alert signatures found within Snort. He created a brief overview of these, many of which triggered alerts on the University's network during the five-day span of our analysis. Below is t he table (Fig. Appendix-1) he created with the brief summaries. (Some of the signatures he describes were not found during our analysis.) His full document can be found here:

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

Name of alert	Description of alert
WEB-MISC Lotus Domino directory traversal	Lotus Domino is a multiplatform web server which integrates messaging and various interactive web applications. It is possible for a remote user to gain access to any known file residing on the Lotus Domino Server 5.0.6 and previous. A specially crafted HTTP request comprised of '.nsf' and '../' along with the known filename, will display the contents of the particular file with read permissions. It should be noted that when making this malformed request Internet Explorer removes '.nsf' portion of the URL, obstructing the exploitation of this vulnerability. Successful exploitation of this vulnerability could enable a remote user to gain access to systems files, password files, etc. This could lead to a complete compromise of the host.
WEB-CGI w3-msql access	Under certain versions of Mini SQL, the w3 -msql CGI script allows users to view directories which are set for private access via .htaccess files. W3 -mSQL converts any form data passed to a script into global Lite variables and these variables can then be accessed by your script code. When an HTML form is defined a field name is given to each

	element of the form. When the dat a is passed to W3-mSQL the field names are used as the variable names for the global variables. Once a set of variables has been created for each form element, the values being passed to the script are assigned to the variables. This is done automatically during start-up of the W3-mSQL program.
SNMP public access	Insufficient access control, and allow reading/writing of MIB data with any community password
Virus – Possible pif Worm	Numerous worms use this .pif extension including Sircam.
WEB-MISC Compaq buffer overflow vulnerability	The administration tool is vulnerable to buffer overflow attack techniques employing maliciously -formed user-supplied input. Properly exploited, this vulnerability can allow a remote attacker to execute arbitrary code on the affected system, with the privilege level of the system administrator.
IDS50/ nicod_ nicod-active-subseven	SubSeven is a nicod for the windows platform. It comes at least in two parts a client and a server. The client is used by the hacker to connect to the victim' s machine. Once the server.exe is installed on the victim's machine the hacker has full access to the victim's machine. http://www.sans.org/newlook/resources/IDFAQ/subseve n.htm
WEB-CGI ksh access	Korn shell access, this may or may not be malicious.
RFB – Possible WinVNC – 010708-1	There may be a VNC server or client on the network. WinVNC has multiple exploits associated with it
RPC tcp traffic contains bin_sh	Bin_sh is string which is common to many exploits, it is usually an attempt for an intruder to get a shell
spp_http_decode: CGI Null Byte attack detected	If the http decoding routine finds a %00 in an http request, it will alert with this message. Sometimes you m ay see false positives with sites that use cookies with URL encoded binary data, or if you're scanning port 443 and picking up SSL encrypted traffic.
Back Orifice	Back Orifice is not a virus. It is in essence a ***remote administration tool***. It gives "system admin" type privileges to a remote user by way of the computer's Internet link. http://www.nwinternet.com/~pchelp/bo/bo.html
SCAN FIN	Portscan that sets only the TCP FIN flag. This scan can produce varied results based on the type of operating system the victim is utilizing/
X11 outgoing	Client inside the network, server (display) outside. This could be an exploited box that is serving an X terminal to an attacker.

INFO Inbound GNUTella Connect request	The mp3 service is requesting a connection. There may be a system with GNUtella on the network.
Tiny Fragments – Possible Hostile Activity	Many small fragments. This is a method that attackers use to bypass firewalls as well as confuse certain TCP/IP stacks.
WEB-CGI redirect access	CGI script that will redirect browsers to a new URL. It can display a page telling the user they are about to be redirected as well as log the redirect. This program can be used maliciously to redirect users to a sp ecific web location.
ICMP Timestamp Reply	The data received (a timestamp) in the message is returned in the reply together with an additional timestamp. The timestamp is 32 bits of milliseconds since midnight UT. The ID and Seq # fields returned to the se nder should be unaltered. Type: 14 Code: 0
WEB-IIS view source via translate header	It is possible to force the IIS server to send back the source of known scriptable files to the client if the HTTP GET request contains a specialized header with 'Translat e: f' at the end of it, and if a trailing slash '/' is appended to the end of the URL. The scripting engine will be able to locate the requested file, however, it will not recognize it as a file that needs to be processed and will proceed to send the file source to the client.
X86 NOOP – nicode BUFFER OVERFLOW ATTACK	NOOP's are common to find in the payload of packets that are attempting to perform a buffer overflow. In this case, they are masked in Unicode. This is a fairly reliable signature which is rarely a false positive.
WEB-IIS scripts-browse	IIS may return content specified by a malicious third party back to a client through the use of specially formed links.
EXPLOIT x86 setgid 0	Exploit that attempts to utilize the root group id (0).
WEB-CGI upload.pl access	CGI script which allows the uploading of files. This perl script allows directory traversal.
WEB-FRONTP AGE fpcount.exe access	Fpcount.exe is an exploitable program that should not be used. This may be a false positive because the signatur e simply looks for the "fpcount.exe" string in the payload of the packet.
INFO napster upload request	Napster is an mp3 file trading service.
WEB-CGI rsh access	Rsh is a service which does password authentication through plain text. All passwords are vi sible to anyone who is sniffing on the network. Highly advisable to disable this service.
WEB-CGI files.pl access	The toolkit contained a script called "FILES.PL" that could be used to view the contents of files or directories on the server by a remote at tacker. This is done by passing the parameter "file=<file-or-directory-to-view>" to the script. An attacker could gain information useful in conducting subsequent attacks, or retrieve personal or proprietary information.
WEB-MISC whisker head	Web scanner that has many anti-IDS features. This signature means that this scanner may have been used against the network.

NMAP TCP ping!	Nmap is a popular port scanner and this is one of the methods that it can scan. It is a simple TCP ping. (as opposed to the usual ICMP echo request/reply).
WEB-MISC L3retriever HTTP Probe	Scanner that probes web servers. It may have been used against the network.
Beetle.ucs	A CD burning web site? Unknown.
ICMP Echo Request BSDtype	A BSD O.S. Echo Request. Arachnids 152. Type: 8 Code: 0
WEB-MISC count.cgi access	Wwwcount (count.cgi) is a very popular CGI program used to track website usage. In particular, it enumerates the number of hits on given webpages and increments them on a 'counter'. In October of 1997 two remotely exploit able problems were discovered with this program. The first problem was somewhat innocuous in that it only allowed remote users to view .GIF files they were not supposed to have access to. This may be dangerous if the site contains sensitive data in .GIF files such as demographic/financial data in charts etc. The second and most serious problem is a buffer overflow in QUERY_STRING environment variable handled by the program. In essence a remote user can send an overly long query to the program and overflow a buffer in order to execute their own commands as whatever privilege level the program is running as.
Connect to 515 from inside	This is a connection to the LPD service from within the network.
MISC Large ICMP Packet	This signature can be cause by a vari ety of sources. It is primarily triggered upon an MTU discovery attempt.
Queso fingerprint	Queso is an operating system fingerprint program. This means that this program was detected while it scanned a host on the network.
INFO FTP anonymous FTP	There was an anonymous login to an FTP server. This signature is informative more than it shows a specific hacking attempt. This should only cause alarm is there should not be an anonymous FTP server on the network.
WEB-MISC http directory traversal	A web server has been used to traverse the hosts' directories. This may or may not be an incident to investigate.
High port 65535 udp – possible Red Worm – traffic	The "Code Red" worm is self -replicating malicious code that exploits a known vulnerability in Microsoft IIS servers (CA-2001-13).
ICMP Destination Unreachable (Protocol Unreachable)	This is an administrative alert. It may or may not be worthy of investigation. Type: 3 Code: 2
spp_http_decode: IIS Unicode attack detected	A flaw exists in the handling of .asp requests. Typically when a request is made for an .asp file, IIS will identify that it is a script and run it as such. However if the host is formatted with a FAT file system and a request i s made with an .asp Unicode encoded file extension, IIS may not handle the request

	properly and return the source code of the file.
TELNET login incorrect	Multiple telnet incorrect logins could mean that an intruder is attempting to brute force the passw ord of an account on a telnet server.
INFO Possible IRC Access	Internet Relay Chat program access. This means that IRC has been detected, merely informational.
Null scan!	This is a TCP scan that has no flags set. This can cause some TCP/IP stack implemen tations to disclose information about open ports on the system.
ICMP Echo Request Windows	A ping from a Windows machine.
WEB-MISC 403 Forbidden	Attempt to access a web page that is "Forbidden" by the administrator. This may or may not be harmful. It may be accidental.
FTP DoS ftpd globbing	Globbing generates pathnames from file name patterns used by the shell, eg. Wildcards denoted by * and ?, multiple choices denoted by {}, etc. The vulnerable FTP servers can be exploited to exhaust system resources if per-user resource usage controls have not been implemented.
INFO Outbound GNUTella Connect accept	Another GNUtella signature, this time it is for an outbound request.
INFO Napster Client Data	Another Napster signature, this time it is for a Napster cli ent.
TCP SRC and DST outside network	The IDS should only capture data that is coming to or from the local network. If data is neither originating nor destined for the local network, the data must be spoofed, which is a tell -tale sign of malicious activity or a misconfigured router.
ICMP Fragment Reassembly Time Exceeded	This may be informational and not malicious. There is a 60 second grace period for fragment reassembly and this alerts us to that scenario. Type: 11 Code: 1
External RPC call	Attempted use of an RPC service from a remote location. The RPC services are listed as one of the top 10 SANS most vulnerable services.
ICMP Echo Request Sun Solaris	Sun specific ping. This is an informational alert. Type: 8 Code: 0
ICMP Source Quench	This alert is triggered when one of the hosts in a connection cannot handle the amount of data being sent to it from another host. A source quench tells the offending host to reduce the amount of traffic it is sending. Type: 4 Code: 0
ICMP traceroute	Informational. Traceroute traces the route a packet will take to a particular destination. Traceroute will initially send a packet

	with a TTL of 1 to the ultimate destination and await an error response from the host the packet timed out at. It will continue to increment the TTL by 1 until it finally reaches the destination host.
ICMP Destination Unreachable (Host Unreachable)	Informational. ICMP error message saying that the destination cannot be reached. Type: 3 Code: 1
ICMP Destination Unreachable (Network Unreachable)	Informational. The router cannot reach the desired network. Type: 3 Code: 0
SMTP relaying denied	An attempt to relay mail from an SMTP server failed and the server replied with this message. This is usually a good message because open mail relaying will lead to the blacklisting of the particular SMTP server.
WEB-MISC prefix-get //	This string is associated with numerous exploits including: ht://dig Remote Command Execution Vulnerability ht://dig Arbitrary File Inclusion Vulnerability AOL Instant Messenger 'aim://' Buffer Overflow Vulnerability Trend Micro Interscan Applet Trap '//' Bypass Vulnerability
MISC source port 53 to <1024	Port 53 is the reserved address for nameserver activity. A DNS server should send data (source) on port 53 to a port above 1024 (destination). Sometimes on older BIND implementations, both the source and destination are 53 and therefore this leads to a false positive.
ICMP Destination Unreachable (Communication Administratively Prohibited)	Informational. This is an alert that is generated when the network has specific restrictions on the traffic. This ICMP message is returned to a host who attempts to direct traffic to a restricted location. RFC 1812: http://sunsite.dk/RFC/rfc/rfc1812.html Type: 3 Code: 13
MISC Large UDP Packet	This could be a sign of a UDP flood. If many large UDP packets are sent to a host it can cause a DOS. Another possibility is that the UDP session is actually a covert channel used by an attacker to communicate with a compromised host. This warrants investigation.

Fig. Appendix -1 Jamil Farschchi, GCIA, description of many Snort alert signatures

Appendix B: References and Analysis Process

1. Silicon Defense. - http://www.silicondefense.com
2. Internet Assigned Numbers Authority – http://www.iana.org/assignments/port - numbers
3. KAZAA File Sharing Service - http://www.kazaa.com/en/index.htm
4. American Registry for Internet Numbers – ARIN - http://www.arin.net/
5. Microsoft Corporation – IIS Security Information

   http://www.microsoft.com/downloads/release.asp?ReleaseID=33961&area=search&ordinal=2

   http://www.microsoft.com/technet /treeview/default.asp?url=/technet/security/topics/codealrt.asp?frame=true
6. System Administration, Network and Security Organization (SANS) –

   http://www.sans.org

   http://rr.sans.org/threats/ICMP_attacks.php

   http://www.sans.org/topten.htm
7. Asian-Pacific Network Information Center - APNIC – http://www.apnic.net
8. Web Ferret – Search Engine - http://www.ferretsoft.com
9. Google – Search Engine – http://www.google.com
10. CERT® Coordination Center (originally the Carnegie -Mellon Emergency Response Team) – http://www.cert.org
11. National Information Protection Center –http://www.nipc.gov/
12. Deerfield - Wingate software – http://www.deerfield.com
13. Gnutella – http://www.gnutella.com
14. SSH – http://www.ssh.com
15. Open SSH – http://www.openssh.com

Multiple GCIA practical papers were consulted for formatting, general approach, structure, etc. The ones used for direct correlation or information are cited directly within this paper, but I would like to cite them again, as their work was extremely valuable. They are:

- Lukacs, Steve – http://www.giac.org/practical/steve_lukacs_gcia.doc
- Leach, David – http://www.giac.org/practical/david_leach_gcia.doc
- Farschchi, Jamil - http://www.giac.org/practical/jamil_farschchi_gcia.doc

Additionally, the following practical papers were downloaded and read for a general sense of direction, formatting, and so on:

- Kneppers, Mark – http://www.giac.org/practical/mark_kneppers_gcia.doc
- Hoover, James – http://www.giac.org/practical/james_hoover_gcia.doc
- Holland, Jeffrey – http://www.giac.org/practical/jeffrey_holland_gcia.doc
- Shinberg, Scott – http://www.giac.org/practical/scott_shinberg_gcia.doc
- Baker, Chris – http://www.giac.org/practical/chris_baker_gcia.doc
- Woodroffe, Alan – http://www.giac.org/practical/alan_woodroffe_gcia.doc
- Lethaby, Chris – http://www.giac.org/practical/chris_lethaby_gcia.doc
- Pitts, Donald – http://www.giac.org/practical/donald_pitts_gcia.doc
- Lajon, Gregory – http://www.giac.org/practical/gregory_lajon_gcia.doc
- Jenkinson, John – http://www.giac.org/practical/john_jenkinson_gcia.doc

The approach taken was similar to that of the others who have gone before me, and pretty much follows the outline of this document. First, the Snort data was (briefly) analysed by hand. Next, the Snort data was modified as mentioned in the body of this paper. Then it was run through the SnortSnarf analysis tool.

This took quite a while (most of two days) to run, and failed during the scan log portion. The scan logs were eventually broken into 3 sections and run individually, and the data was re-assimilated once the SnortSnarf results were complete. Perl, Excel, MSDOS, and several UNIX tools were used in the modification and analysis of the data. The resulting output was used to draw correlations and to do further research.

As there was so much alert data (for this assignment, a preferable situation to the alternative), I chose to limit the scope to the most frequent alerts and the top ten talkers (both source and destination). Each alert was examined for anything unique, alarming, interesting, and so on, and further references, correlations, and other information was searched for and included in the analysis. The results are contained within the body of this paper.