Intrusion Detection in Depth
GCIA Practical Assignment
Version 3.1

Lorna J. Hutcheson
Orlando SANS 2002
# TABLE OF CONTENTS

**Part 1: Describe the State of Intrusion Detection**  
3

- Introduction  
3
- Bounce Attack Overview  
3
- Stimulus and Response  
3  
  - ICMP Traffic  
3  
  - TCP Traffic  
5
- Malicious Usage  
5  
  - Packet Bounce Scans  
5  
  - Packet Bounce Attacks  
6
- Distributed Reflection Denial Of Service  
6  
  - Source of Trace:  
6  
  - Detect was generated by:  
7  
  - Probability the source address was spoofed:  
8  
  - Description of the attack:  
8  
  - Attack Mechanism  
8  
  - Correlations  
9  
  - Evidence of active targeting  
9
- Conclusion  
10

**Citation of Sources**  
11

**Part 2: Network Detects**  
12

- Detect 1  
12  
  - Source of Trace:  
19  
  - Detect was generated by:  
19  
  - Probability the source address was spoofed:  
20  
  - Description of the attack:  
21

- Query the APNIC Whois Database  
25  
  - Attack Mechanism  
27  
  - Correlations  
27  
  - Evidence of active targeting  
28  
  - Severity  
28  
  - Defensive recommendations  
29  
  - Multiple choice test question  
29

- Detect 2  
29  
  - Source of Trace:  
31  
  - Detect was generated by:  
31  
  - Probability the source address was spoofed:  
31

Table of Contents  
Page 2 of 73
Part 1: Describe the State of Intrusion Detection

Introduction
The focus of this paper is to analyze the Distributed Reflection Denial of Service attack presented on Steve Gibson’s website at [http://grc.com/dos/drdos.htm](http://grc.com/dos/drdos.htm). We will use the analysis format provided in the SANS Intrusion Detection practical to do the examination the attack based on the information provided on Mr. Gibson’s website. Before doing the analysis, we will first look at the anatomy of a packet bounce and some basic network functionality.

Bounce Attack Overview
Before we begin looking at any specific bounce attacks, we need to define a bounce attack in a generic manner. A bounce attack simply consists of sending some sort of packet (TCP, UDP etc) to an unsuspecting system that is up and running on the Internet. The unsuspecting system then sends a response back to what is believed to be the requesting system. The following example will help to clarify the point:

![Packet Bounce Diagram](image.png)

Once again, speaking in generic terms for a moment, this attack is usually carried out by an attacker that is sending packets to make the unsuspecting host believe it is the victim who wants to talk with him. Depending on the type of packet sent, it will elicit different responses from the victim.

Stimulus and Response
A quick review of stimulus and response concepts is necessary to understand the intent of the packet bounce in question. It is necessary to have a clear understanding of what is considered “normal” network traffic in order to understand what might be gained from an
activity and even if the activity is normal. “Correct traffic is consistent with the
specifications of the Request for Comment (RFC) documents that define the IP protocols.
Incorrect traffic violates these protocols.” (Northcutt, 133) We are going to look at some
different aspects of “normal” network traffic and then apply it to a given packet bounce
scenario. This is by no means a comprehensive analysis of “normal” network traffic.

ICMP Traffic

ICMP is a connectionless protocol with no ports associated with it. It is basically the
Internet messenger. If there trouble getting a packet from point A to point B, it is ICMP
that lets you know what is going on with the transmission and tells you if it is fixable or
not. If you’re talking to fast to another host, it’s ICMP that sends the source quench
message to throttle back the transmission. ICMP helps to keep this running smoothly on
the Internet. It is amazing what this one protocol can do. Here is a chart taken from page
71 of Dr. Stevens’s book called *TCP/IP Illustrated, Volume 1, The Protocol* that
illustrates all the many functions that ICMP can provide.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
<th>Query</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Echo reply</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>destination unreachable;</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Network unreachable</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Host Unreachable</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Protocol unreachable</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Port unreachable</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Fragmentation needed but don’t fragment bit set</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Source route failed</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Destination network unknown</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Destination host unknown</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Source host isolated</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>Destination network administratively prohibited</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>Destination host administratively prohibited</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Network unreachable for TOS</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>Host unreachable for TOS</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>Communication administratively prohibited</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>Precedence cutoff in effect</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>Source quench</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Redirect:</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Redirect for network</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Redirect for host</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Redirect for type-of-service and network</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Redirect for type of service and host</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>Echo request</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>Router advertisement</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>Router solicitation</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Time exceeded:</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Time-to-live equals 0 during transit</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Time-to-live equals 0 during reassembly</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Parameter problem:</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>IP header bad (catchall error)</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Required option missing</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>Timestamp request</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>Timestamp reply</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>Information request (obsolete)</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>Information reply (obsolete)</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>

Part 1: Describe the State of Intrusion Detection
Page 5 of 73
As you can see, ICMP provides for the ICMP echo request and echo reply. If ICMP echo request is sent, you can see all of the possible expected answers that you could receive other than an echo reply that tells you the host is up. You might get a type 3, code 1 back telling you that the host is unreachable. These are all “normal” network traffic stimulus and responses.

**TCP Traffic**

Transmission Control Protocol (TCP) is a connection oriented protocol. This means that it requires an established connection before data is exchanged. (Stevens, 223) A quick look at the three way handshake will help to clarify what is expected. The computer that wishes to talk to another computer must first send a SYN (synchronization) packet which is like a “hello” in human language. This really isn’t a different packet, but the SYN flag is set in the TCP packet. The receiving computer might accept the connection by sending an ACK (acknowledge) packet as well as setting the SYN flag and establishing its own connection with its own sequence number to the originating system. Each side of the connection uses their own sequence numbers to ensure they track the packets as they come in and in the right order and that all the data is received. The initiating system will send its own ACK packet and the three way handshake is complete. The systems can now talk.

What if the receiving computer didn’t want to talk? One option is to send a RST (reset). This is a TCP packet with the RST flag set. This lets the initiating computer know the other computer doesn’t want to talk now. Maybe it is not even up and running and ICMP steps up and delivers the message from a router. Or maybe the packet is just silently dropped. These are all normal behavior with TCP traffic. Chapters 17-24 of Dr. Richard Stevens book *TCP/IP Illustrated Volume 1* addresses the usage of TCP.

**Malicious Usage**

The defining of the characteristics and behavior of network traffic is essential for all the devices on the Internet to be able to talk. As with everything, someone will come up with a way to use its prescribed behavior in a way that it was not intended. Maybe it is using the standards to perform a scan or maybe launch an attack. Let’s look at a couple of quick examples of how a packet bounce could be used.

**Packet Bounce Scans**

There are ways to scan a network by bouncing a packet. For an example we will look at the Hping pattern discussed in the (SANS 3.5/3.6, Section 7-3). Based upon the normal characteristics of IP, the IP packet has a field called the identification field (IP ID). This field identifies each IP datagram and increments by 1 with every packet. Different OSs generate their IP IDs differently, so you will need to ensure how the system is
incrementing theirs. If there is no response, the IP ID will not increment, but remain the same. (Stevens, 36) How could you use this as a scan? Well, it gives the ability to conceal the identity of originator by bouncing the packet off of a system. Let’s look at it.

To do this, you need to find a host that is up on the network and confirm it is NOT talking to anyone. This will be called our idle host. If this machine is not idle, this will not work. Our target host is the host(s) from which more information is to be obtained. The attacker sends a packet to the idle machine and verifies the IP ID. A SYN packet is then crafted and sent to the target host with source address of the idle host. If the host exists and is listening on the port, it will send a SYN/ACK to the idle host. If it is not, it will send a RST/ACK. The idle host will respond with a RST to the SYN/ACK since it did not originate the SYN packet and increment its IP ID. It will not respond to the RST/ACK. The attacker would then send another packet to the idle host and check the IP ID. If it is incremented by 1, this means the target machine is not listening because the idle host did not respond to it. If it is incremented by 2, the target host is listening on the port.

This is a good example of how the normal characteristics of TCP were used in a way not intended to gain information.

**Packet Bounce Attacks**

The same thing applies to a packet bounce attack. We’ll look at a simple one such as a Smurf attack. In this attack, ICMP is used to launch an attack by bouncing it off an amplifying network. An attacker crafts an ICMP echo request packet with the source address of the target host and sends it to a broadcast address. All of the hosts on that broadcast address respond with an echo reply and uses up the available bandwidth causing a denial of service attack. (Northcutt, Novak, 242-243) Once again, the normal characteristics of ICMP were used in a way not intended.

**Distributed Reflection Denial Of Service**

On his website, at www.grc.com/dos/drdos.htm, Mr. Gibson states, “At 2:00 AM, January 11th 2002, grc.com was blasted off the Internet by a more advanced malicious packet flood. This new style of DDoS attack could be called a Distributed Reflection Denial of Service attack—DRDoS.” The essence of the attack as recorded by Mr. Gibson was that “We appeared to be under attack by more than TWO HUNDRED of the Internet’s core infrastructure routers.” He later states that it was SYN/ACK packets, with a source port of 179, flooding grc.com and provides only a list of IP addresses and resolved names. There was a second wave recorded that originated from different servers with various well known source ports as their origin.

We will look at the attack as recorded by Mr. Gibson and do an analysis of the attack to determine if it is a new attack or if it is possible an old attack with a different twist. We will apply as possible the analysis format that will be used in Part II of the practical as a means of applying a logical look at the attack.

Part 1: Describe the State of Intrusion Detection

Page 7 of 73
Source of Trace:
The data for the attack was provided by Mr. Gibson on his web site at www.grc.com/dos/drdos.htm. From Mr. Gibson’s description of his web server it is behind his ISP’s aggregation router and he has two T1 trunk lines.

Detect was generated by:
The data was provided in the following manner (see table 1) with this being only a small representation of the table showing the routers involved. The second wave was documented in a like manner, but with much less IP addresses (see table 2). It is important to note that there was no mention of what source the data was collected from only that the packets were captured. The type of data given will make it difficult to do a proper analysis of what occurred. There is much information missing that could be helpful such as log files, packet dumps, destination ports etc.

Table 1

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Machine Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>129.250.28.1</td>
<td>ge-6-2-0.r03.sttlwa01.us.bb.verio.net</td>
</tr>
<tr>
<td>129.250.28.3</td>
<td>ge-1-0-0.a07.sttlwa01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.20</td>
<td>ge-0-1-0.a12.sttlwa01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.33</td>
<td>ge-0-0-0.r00.bcrtfi01.us.bb.verio.net</td>
</tr>
<tr>
<td>129.250.28.49</td>
<td>ge-1-1-0.r01.bcrtfi01.us.bb.verio.net</td>
</tr>
<tr>
<td>129.250.28.98</td>
<td>ge-1-2-0.r00.sfldmi01.us.bb.verio.net</td>
</tr>
<tr>
<td>129.250.28.99</td>
<td>ge-1-0-0.a00.sfldmi01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.100</td>
<td>ge-1-1-0.a01.sfldmi01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.113</td>
<td>ge-1-2-0.r01.sfldmi01.us.bb.verio.net</td>
</tr>
<tr>
<td>129.250.28.116</td>
<td>ge-1-1-0.a00.sfldmi01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.117</td>
<td>ge-1-0-0.a01.sfldmi01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.131</td>
<td>ge-0-3-0.a00.scrmca01.us.ra.verio.net</td>
</tr>
<tr>
<td>129.250.28.142</td>
<td>ge-0-2-0.r00.scrmca01.us.bb.verio.net</td>
</tr>
<tr>
<td>129.250.28.147</td>
<td>ge-1-2-0.a00.scrmca01.us.ra.verio.net</td>
</tr>
</tbody>
</table>

Table 2
Source IP

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Machine Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>64.152.  4.  80</td>
<td></td>
</tr>
<tr>
<td>128.121.223.161</td>
<td></td>
</tr>
<tr>
<td>131.103.248.119</td>
<td></td>
</tr>
<tr>
<td>164.109. 18.251</td>
<td></td>
</tr>
<tr>
<td>171. 64. 14.238</td>
<td></td>
</tr>
<tr>
<td>205.205.134. 1</td>
<td></td>
</tr>
<tr>
<td>206.222.179.216</td>
<td></td>
</tr>
<tr>
<td>208. 47.125. 33</td>
<td></td>
</tr>
<tr>
<td>216. 34. 13.245</td>
<td></td>
</tr>
<tr>
<td>216.111.239.132</td>
<td></td>
</tr>
<tr>
<td>216.115.102. 75</td>
<td></td>
</tr>
</tbody>
</table>

www.wwfsuperstars.com  
veriowebsites.com  
www.cc.rapidsite.net  
whalenstoddard.com  
www4.stanford.edu  
s111.novalinktech.net  
forsale.txic.net  
gary7.nsa.gov  
channelserver.namezero.com  
www.jeah.net  
w3.snv.yahoo.com

Probability the source address was spoofed:
There are two different conclusions you could arrive at based on the information provided by Mr. Gibson. All we know from Mr. Gibson’s write-up are the following:

1. SYN/ACK packets from Internet Routers
2. SYN/ACK packets from Servers

Based on this, it is possible it could be spoofed addresses or crafted packets.

Spoofed IP address
Based on the information given, it is possible that the packets were sent from spoofed IP addresses. The following URL, http://online.securityfocus.com/archive/1/37272, documents a tool called Idlescan. This gives you the ability to conduct port scans that appear to becoming from numerous IP addresses, similar to the capabilities of Nmap, Queso and other tools used for spoofing IP address. Based on the given information, you can not rule out the source IP addresses as being spoofed using a tool capable of spoofing.
Description of the attack:

According to Mr. Gibson, the attack began at 0200 on 11 January 2002. Grc.com was flooded by SYN/ACK packets originating from “more than TWO HUNDRED of the Internet’s core infrastructure routers.” (www.grc.com/dos/drdos.htm) The destination port was TCP 179 which is used as the port for Border Gateway Protocol (BGP). At 0400, Mr. Gibson contacted his service provider Verio and requested a block of all incoming traffic with a source port of 179. The second wave of the attack immediately followed. It was also SYN/ACK packets, this time originating from source ports 22, 23, 53, 80, 4001 and 6668 and coming from many different Internet web servers. There is no information as to the total time period of the attack. Mr. Gibson did say that “Verio’s router had discarded more than one billion (1,072,519,399) malicious SYN/ACK packets.” (www.grc.com/dos/drdos.htm)

Attack Mechanism

Let’s look at the attack mechanism and see if we can determine what happened. There is not much concrete evidence provided in Mr. Gibson’s analysis. As such, we can only draw from what he writes for the possible answers.

Is this a Stimulus or Response?

Based on the information provided by Mr. Gibson, it is difficult to make a definite determination as to what is taking place. It is possible that all of the IPs used were generated as decoys for the real IP address and the SYN/ACKs were direct stimulus of the GRC.com website. But what was the purpose? The sheer number of packets that Mr. Gibson talks about makes it unlikely that this is a stimulus. Also, the length of time that this took place lends itself to this conclusion.

In order to get the sheer number of packets that Mr. Gibson said was flooding Verio’s router, it makes more sense that this was a response that grc.com was seeing. The stimulus was unseen and the devices were responding to a SYN packet.

Keep in mind that what is missing are packet captures that show many different items. The TTL would be useful in determining how many hops away all of the devices are. If the packets have the same or very close TTL, it would lend more support that these are crafted packets. The Internet routers listed are from all over the place. There should be a varying TTL for the routers in different locations. It also never mentions the destination address or the destination port. It always mentions the source port. What were these packets aimed at? Many other items of a packet capture would have been useful in determining whether these are stimulus or response.

What service is being targeted?

This is not completely known based on the write-up. The source port is the only one mentioned for all of the packets. If these are indeed responses, then the ports listed in Mr.
Gibson’s write up would be what are being targeted. He lists the following ports: 179 (BGP), 22(Secure Shell), 23(Telnet), 53(DNS), 80(HTTP/Web), 4001 (possible proxy) and 6668 (IRC chat). ([www.grc.com/dos/drdos.htm](http://www.grc.com/dos/drdos.htm))

**Does the service have known vulnerabilities or exposures?**
All of these services have known vulnerabilities. It does not appear however that vulnerability was being sought. It appears from the information known that the normal characteristics of network traffic were being used in a malicious manner.

**Is this benign, an exploit, denial of service, or reconnaissance?**
This appears to be a denial of service attack against Mr. Gibson’s website. It was obviously not benign if it knocked grc.com off the air for a time period greater than two hours. If it was a reconnaissance attempt on all of those unsuspecting routers and servers, it was a poorly designed and noisy attempt. As for an exploit, based on the above discussion on vulnerabilities, it does not appear that vulnerability was being sought.

**Correlations**
A search of BugTrap, Incidents.org, Cert.org, SecurityFocus and many of different web sites as well as Internet searches did not reveal any increased activity on 11 January 2002. It is hard to image that something as large scaled as described by Mr. Gibson and involving as many different IPs went unnoticed. Mr. Gibson did mention some posts on BugTrap after the event, but they discussed mild SYN flood reports and were not the night of the event. I also tried contacting Verio about information pertaining to that night, but they did not have information they could provide without permission.

**Evidence of active targeting**
It would appear that grc.com was actively targeted. The SYN/ACKs from multiple sources were destined for grc.com, although the destination address was not given. That would lend credence to it being active targeting. If SYN packets were crafted to multiple routers and servers, then the source address of grc.com was selected. This would cause the SYN/ACKs to be sent back. Once again, there are no packets to analyze to look for other clues such as time gaps in the packets which could indicate someone else was being hit during that time, or, the time between the SYN/ACKs. Was it following the normal TCP retries of three, six and nine seconds? A possibility exists, without knowing the exact destination address, that this was a DOS against Verio and Mr. Gibson’s website was just a means of accomplishing this. If Mr. Gibson was shut down with two T1 lines, then maybe the attack was against Verio instead.

**Conclusion**
The lack of information provided by Mr. Gibson made it difficult to do a thorough analysis based on actual data. The analysis had to be based upon information provided. The question still remains. Is this a new DoS? Here are the summarized conclusions based upon Mr. Gibson’s write-up.

Part 1: Describe the State of Intrusion Detection
Page 11 of 73
The attack started at 0200 in the morning and went for an undetermined amount of time. The attack appears to be a response to an unknown stimulus. It also appears to be active targeting because Mr. Gibson’s website was shut down. It was SYN/ACK packets and over one billion of them hit Verio’s router. We were told the source port and there were several listed.

What is not known is the exact destination address and port. There are no packet captures provided in the write-up although Mr. Gibson says that he captured them. Everything is based on Mr. Gibson’s write-up.

What kind of attack actually occurred on 11 January 2002? Mr. Gibson says it is a “new style of DDOS attack could be called a Distributed Reflection Denial of Service attack –DRDoS.” (www.grc.com/dos/drdos.htm) As an analyst, it does not appear from the information given to be a new type of attack at all. Remember the packet bounce we discussed in the earlier section? Packets were sent to an idle host, while the true victim was somewhere else and the idle host performed the dirty work? It was a way of using the normal characteristics of network traffic in a malicious way. The Smurf attack using ICMP is a prime example of this. When you look at Mr. Gibson’s attack, the characteristics are very similar. SYN packets were sent to an unsuspecting host and they replied with SYN/ACKs as was expected. This is normal stimulus and response behavior. I do not believe this was a new type of attack at all, but a common packet bounce used in a DDOS. As a final note, there is an excellent paper dated June 26, 2001 by Vern Paxson titled “An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks.” (Paxson, http://www.icir.org/vern/papers/reflectors.CCR.01/index.html) In this paper Mr. Paxson looks at how to defend against a packet bounce, but he refers to them as reflectors. I have to conclude based on information given, that this not a new type of attack at all. It is a packet bounce using the normal characteristics of the protocol for a malicious intent.
Citation of Sources


Northcutt, Steven. IDS Signatures and Analysis, Parts 1 and 2. 2002.


Works Cited

Page 13 of 73
Part 2: Network Detects

Detect 1

1 06:53:31.383133  xxx.xxx.xxx.219 > 10.0.0.1: icmp: echo request (wrong icmp csum) (ttl 208, id 36441, len 48)
   0x0000  4500 0030 8e59 0000 d001 d5a4 cff2 acdb E..0.Y........
   0x0010  0a00 0001 0845 2457 209b ba10 0000 0000 ........SW......
   0x0020  5018 1e8d 6f75 0000 0000 0000 0000 0000 P..ou........

2 06:53:38.3449  xxx.xxx.xxx.220 > 10.0.0.1: icmp: host 0.0.0.0 unreachable - admin prohibited (wrong icmp csum) (ttl 246, id 20967, len 48)
   0x0000  4500 0030 51e7 0000 f601 ec15 cff2 acdc E..0Q...........
   0x0010  0a00 0001 030a 2457 209b ba10 0000 0000 ......$W........
   0x0020  5018 1e8d 6f75 0000 0000 0000 0000 0000 P..ou........
   0x0030  5018 1e8d 6f75 0000 0000 0000 0000 0000 P..ou........

3 06:53:38.4069  xxx.xxx.xxx.220 > 10.0.0.1: icmp: ip reassembly time exceeded (wrong icmp csum) (ttl 200, id 48005, len 48)
   0x0000  4500 0030 bb85 0000 c801 b077 cff2 acdf E..0.......w....
   0x0010  0a00 0001 0b01 0000 0000 0000 0000 0000 ................
   0x0020  5018 1e8d 6f75 0000 0000 0000 0000 0000 P..ou........

4 06:53:39.0306  xxx.xxx.xxx.221 > 10.0.0.1: icmp: type-#20 (wrong icmp csum) (ttl 233, id 2044, len 48)
   0x0000  4500 00b4 3916 0000 f701 03ba cff2 ace0 E...9...........
   0x0010  0a00 0001 0837 44e0 7b2e 30a8 0a00 0001 ......t{.0.......
   0x0020  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........
   0x0030  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........
   0x0040  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........
   0x0050  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........
   0x0060  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........
   0x0070  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........
   0x0080  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........

   0x0000  4500 00b4 3916 0000 f701 0361 cff2 acde E...9......a....
   0x0010  0a00 0001 0837 44e0 7b2e 30a8 0a00 0001 ......t{.0.......
   0x0020  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........

   0x0000  4500 00b4 3916 0000 f701 0361 cff2 acde E...9......a....
   0x0010  0a00 0001 0837 44e0 7b2e 30a8 0a00 0001 ......t{.0.......
   0x0020  5010 4733 b651 0000 0000 0000 0000 0000 P.G3.Q........

7 06:53:39.5705  xxx.xxx.xxx.224 > 10.0.0.1: icmp: type-#31 (wrong icmp csun) (ttl 232, id 47353, len 48)
   0x0000  4500 0030 b8f9 0000 e801 4300 cff2 acdb E..0............
   0x0010  0a00 0001 14d0 74d5 1192 b5e2 0a00 0001 ......t{.0.......
   0x0020  5018 5eb3 9845 0000 0000 0000 0000 0000 P.^..E........

8 06:53:40.2633  xxx.xxx.xxx.226 > 10.0.0.1: icmp: parameter problem - code 2 (wrong icmp csun) (ttl 199, id 14141, len 48)
   0x0000  4500 0030 373d 0000 c701 35c0 cff2 acdc E..07=.......u....
   0x0010  0a00 0001 0c02 b667 14cc 0a00 0001 ......Z.a........
   0x0020  5018 5eb3 9845 0000 0000 0000 0000 0000 P.^..E........

   0x0000  4500 00b4 3916 0000 c701 35c0 cff2 acdc E..07=.......u....
   0x0010  0a00 0001 0c02 b667 14cc 0a00 0001 ......Z.a........
   0x0020  5018 5eb3 9845 0000 0000 0000 0000 0000 P.^..E........

10 06:53:42.3107  xxx.xxx.xxx.229 > 10.0.0.1: icmp: type-#40 (wrong icmp csun) (ttl 246, id 62067, len 48)
   0x0000  4500 0030 f273 0000 f601 48b0 cff2 ace5 E..0.s.....K....
   0x0010  0a00 0001 2801 106b 3109 3ef5 0a00 0001 ......(k1>....
   0x0020  5018 5eb3 9845 0000 0000 0000 0000 0000 P.^..E........

11 06:53:42.8923  0:d0:58:43:38:80 0800 194: xxx.xxx.xxx.231 > 10.0.0.1: icmp: echo reply (wrong icmp csun) (ttl 233, id 19911, len 180)

Page 14 of 73

© SANS Institute 2000 - 2002. As part of GIAC practical repository. Author retains full rights.
icmp: echo reply (wrong icmp csum) (ttl 222, id 3578, len 180)
0x0000 4500 00b4 00fa 0000 de01 4774 cff2 ace7 E...........Gt....
0x0010 0a00 0001 0015 0019 00ff 0a00 0001 ..............F2........
0x0020 5000 e66c cf76 0000 0000 0000 0000 0000 P..F.V...........
0x0030 0000 0000 0000 0000 0000 0000 0000 4141 4E46 ..........AAA
0x0040 4141 4141 4141 0000 0000 0000 0000 0000 0000 EY"............

12 06:53:31.430783  xxx.xxx.xxx.232 > 10.0.0.1: icmp: type=110 (wrong icmp csum) (ttl 211, id 7080, len 48)
0x0000 4500 0030 1b40 0000 d301 4774 cff2 ace8 E............E1....
0x0010 0a00 0001 0047 0050 0000 0000 0000 0000 ...........P........
0x0020 5018 2441 e05c 0000 0000 0000 0000 0000 P.$A............
0x0030 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
(6 Lines of zeros deleted)

0x0000 4500 0030 ffae 0000 e301 4774 cff2 acec E............QN....
0x0010 0a00 0001 005d 007b 0000 0000 0000 0000 ...........{........
0x0020 5038 2d7e 0a5b 0000 0000 0000 0000 0000 P8-~.

0x0000 4500 0030 ffa7 0000 e301 4774 cff2 acec E............QN....
0x0010 0a00 0001 005d 007b 0000 0000 0000 0000 ...........{........
0x0020 5038 2d7e 0a5b 0000 0000 0000 0000 0000 P8-~.

15 06:53:31.435075  xxx.xxx.xxx.235 > 10.0.0.1: icmp: echo request (wrong icmp csum) (ttl 207, id 20202, len 48)
0x0000 4500 0030 4eea 0000 e401 4774 cff2 aceb E............X.C....
0x0010 0a00 0001 0078 007b 0000 0000 0000 0000 ...........{........
0x0020 5018 0774 2d00 0000 0000 0000 0000 0000 P.t-...........

16 06:53:31.435373  xxx.xxx.xxx.236 > 10.0.0.1: icmp: time stamp reply id 666 seq 0 : org 0xa000001 recv 0x50180774 xmit 0x2d000000 (wrong icmp csum) (ttl 239, id 49620, len 48)
0x0000 4500 0030 c1d4 0000 ef01 107d cff2 acee E............}....
0x0010 0a00 0001 0078 007b 0000 0000 0000 0000 ...........{........
0x0020 5018 0774 2d00 0000 0000 0000 0000 0000 P.t-...........

17 06:53:31.440461  xxx.xxx.xxx.237 > 10.0.0.1: icmp: echo reply (wrong icmp csum) (ttl 227, id 22537, len 48)
0x0000 4500 00b4 5809 0000 e301 4774 cff2 acef E............X....
0x0010 0a00 0001 0055 007b 0000 0000 0000 0000 ...........{........
0x0020 5038 f6ec e3ec 0000 0000 0000 0000 0000 P8..............

20 06:53:31.476674  xxx.xxx.xxx.240 > 10.0.0.1: icmp: router solicitation (wrong icmp csum) (ttl 252, id 10092, len 48)
0x0000 4500 0030 276c 0000 e001 0000 0000 0000 P.w.o...........
0x0010 0a00 0001 0123 0019 00ff 0a00 0001 ..............V........
0x0020 5002 57f6 6f09 0000 0000 0000 0000 0000 P.w.o...........

0x0000 4500 0030 dd5e 0000 e001 1f9a 0876 0a00 0001 P.w.o...........
0x0010 0a00 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
(7 Lines of zeros deleted)

22 06:53:31.495096  xxx.xxx.xxx.241 > 10.0.0.1: icmp: echo request (wrong icmp csum) (ttl 246, id 48371, len 48)
0x0000 4500 00b4 00fa 0000 de01 4774 cff2 aceb E............X.C....
0x0010 0a00 0001 0015 0019 00ff 0a00 0001 ..............F2........
0x0020 5000 e66c cf76 0000 0000 0000 0000 0000 P..F.V...........
0x0030 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
(7 Lines of zeros deleted)

Part 2: Network Detects
Page 15 of 73
Part 2: Network Detects

Page 17 of 73

Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46

0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

34 06:31:36.07036  xxx.xxx.xxx.220 > 10.0.0.1: icmp: redirect-tos 0.0.0.0 to net 0.0.0.0
  (wrong icmp csum) (ttl 249, id 29171, len 48)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

35 06:31:36.09822  xxx.xxx.xxx.252 > 10.0.0.1: icmp: echo reply (wrong icmp csum) (ttl 237, id 2475, len 180)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

38 06:31:36.23101  xxx.xxx.xxx.220 > 10.0.0.1: icmp: 0.0.0.0 protocol 151 unreachable
  (wrong icmp csum) (ttl 249, id 29171, len 48)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

39 06:31:36.25531  xxx.xxx.xxx.220 > 10.0.0.1: icmp: 0.0.0.0 unreachable -need to frag
  (mtu 16045) (wrong icmp csum) (ttl 241, id 24120, len 180)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

40 06:31:36.42962  xxx.xxx.xxx.220 > 10.0.0.1: icmp: source quench (wrong icmp csum)
  (ttl 248, id 33889, len 180)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

41 06:31:36.45742  xxx.xxx.xxx.50 > 10.0.0.1: icmp: type-#116 (wrong icmp csum)
  (ttl 215, id 45405, len 48)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

42 06:31:36.58886  xxx.xxx.xxx.220 > 10.0.0.1: icmp: net 0.0.0.0 unreachable - tos prohibited
  (wrong icmp csum) (ttl 199, id 8018, len 48)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

43 06:31:36.60281  xxx.xxx.xxx.51 > 10.0.0.1: icmp: type-#36 (wrong icmp csum)
  (ttl 213, id 47412, len 48)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

  request (wrong icmp csum) (ttl 215, id 19036, len 180)
  0x0000 4500 0030 71f3 0000 f901 c909cff2 acdc E..0g..........
  0x0010 0a00 0001 0503 0000 0000 0000 0a00 0001 ........................
  0x0020 5030 f615 3b79 0000 0000 0000 0000 0000 P0..;y..........

(6 Lines of zeros deleted)

(6 Lines of zeros deleted)
0x0050 5556 5741 4243 4445 4647 4849 0000 0000 UVWBCDEFGH...
Part 2: Network Detects

Page 19 of 73
0x0000  4500 0030 b917 0000 e001 9ae5 cff2 acdc  E..0...........
0x0010  0a00 0001 030c 614c 1778 2f74 0a00 0001 ...2L.x/t....
0x0020  5038 2250 0838 0000 0000 0000 0000 0000 P@#F............
65 06:53:31.883052  xxx.xxx.xxx.72 > 10.0.0.1: icmp: type-#36 (wrong icmp csum) (ttl 207, id 49866, len 48)
0x0000  4500 0030 c2ca 0000 cf01 a2c6 cff2 ac48 E..0...........H
0x0010  0a00 0001 0900 63da 11fd eb14 0a00 0001 ...........t....
0x0020  5038 488f 86d4 0000 0000 0000 0000 0000 P8H............
66 06:53:31.889054  xxx.xxx.xxx.220 > 10.0.0.1: icmp: redirect-tos 0.0.0.0 to net 203.237.226.116 (wrong icmp csum) (ttl 254, id 26556, len 48)
0x0000  4500 0030 67bc 0000 fe01 ce40 cff2 acdc E..0g......@....
0x0010  0a00 0001 0502 e217 cbed e274 0a00 0001 ...t.............
0x0020  5018 a59b 6ebe 0000 0000 0000 0000 0000 P...n...........
67 06:53:31.893877  xxx.xxx.xxx.73 > 10.0.0.1: icmp: type-#7 (wrong icmp csum) (ttl 244, id 36175, len 48)
0x0000  4500 0030 0279 0000 cd01 6516 cff2 ac4a E..0.y....e....J
0x0010  0a00 0001 0000 0000 0000 0000 0a00 0001 ................
0x0020  5018 1b16 7096 0000 0000 0000 0000 0000 P...p...........
70 06:53:31.904951  xxx.xxx.xxx.78 > 10.0.0.1: icmp: echo reply (wrong icmp csum) (ttl 205, id 633, len 48)
0x0000  4500 0030 0279 0000 cd01 6516 cff2 ac4a E..0.y....e....J
0x0010  0a00 0001 0000 0000 0000 0000 0a00 0001 ................
0x0020  5018 1b16 7096 0000 0000 0000 0000 0000 P...p...........
71 06:53:31.904951  xxx.xxx.xxx.75 > 10.0.0.1: icmp: type-#32 (wrong icmp csum) (ttl 244, id 38341, len 48)
0x0000  4500 0030 95c5 0000 f901 aac8 cff2 ac4f E..0...........K
0x0010  0a00 0001 2072 f10d 03c2 85f0 0a00 0001 ..........F........
0x0020  5018 fafc 6c6e 0000 0000 0000 0000 0000 P...o...........
72 06:53:31.905247  xxx.xxx.xxx.77 > 10.0.0.1: icmp: type-#33 (wrong icmp csum) (ttl 249, id 43089, len 48)
0x0000  4500 0030 a851 0000 f901 933a cff2 ac4d E..0.Q.....:...M
0x0010  0a00 0001 210f f10d 03c2 85f0 0a00 0001 ..........F........
0x0020  5018 fafc 6c6e 0000 0000 0000 0000 0000 P...o...........
0x0000  4500 0030 0279 0000 cd01 6516 cff2 ac4a E..0.y....e....J
0x0010  0a00 0001 0000 0000 0000 0000 0a00 0001 ................
0x0020  5018 1b16 7096 0000 0000 0000 0000 0000 P...p...........
74 06:53:31.911133  xxx.xxx.xxx.220 > 10.0.0.1: icmp: host 0.0.0.0 unreachable - precedence cutoff (wrong icmp csum) (ttl 249, id 48567, len 48)
0x0000  4500 0030 b917 0000 e001 9ae5 cff2 acdc  E..0...........E
0x0010  0a00 0001 030c 614l 1778 2f74 0a00 0001 ...2L.x/t....
0x0020  5038 2250 0838 0000 0000 0000 0000 0000 P@#F............
75 06:53:31.912329  xxx.xxx.xxx.79 > 10.0.0.1: icmp: address mask request (wrong icmp csum) (ttl 241, id 8255, len 48)
0x0000  4500 0030 203f 0000 f101 234b cff2 ac4f E..0.7....K...O
0x0010  0a00 0001 1100 f18e cb9d 400f 0a00 0001 ..........F0......
0x0020  5018 b62c 6373 0000 0000 0000 0000 0000 P...c...........
76 06:53:31.913588  xxx.xxx.xxx.80 > 10.0.0.1: icmp: echo reply (wrong icmp csum) (ttl 208, id 40099, len 48)
0x0000  4500 0030 9ca3 0000 d001 c7e5 cff2 ac50 E..0...........P

Part 2: Network Detects

Page 20 of 73
Part 2: Network Detects

Page 21 of 73

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.
Source of Trace:
This trace was found at http://lists.jammed.com/incidents/2002/05/ and was submitted by Robert Buckley. The IPs used as the source are from Mr. Buckley’s external IP addresses. The packets have been numbered with a red number for clarity in referencing during the analysis.

Detect was generated by:
The detect was generated by SHADOW (Secondary Heuristic Analyses for Defensive Online Warfare) and the packet dump was displayed using TCPdump. Mr. Buckley also tells us that SHADOW identified this as Stacheldraht. You will find spoofworks in the ICMP packet and the Arachnids database points to an ICMP ID of 666. This will be discussed in the Analysis portion.

Probability the source address was spoofed:
The probability of this being spoofed is 100%. Upon analyzing the TCPdump output, you will find several things of interest.

1. The biggest indicator that first drew my attention was the lack of MAC addresses in the majority of the packets. However, some of the packets had the source MAC address listed, but there was no destination MAC address. For example:

As you can see from this, there is only a source MAC address, 0:d0:58:43:83:80, but no destination MAC address. All of those packets containing a source MAC address, all contained the exact same MAC address, but different IP addresses (see packets 5, 6, 9, 11, 14, 17, 25, 26, 27, 36, 37, 44, 45, 47, 49, 51, 54, 55, 56, 59, 60, 61, 63, 73, 77, 78, 80)

2. Another interesting characteristic of the trace was that all of the packets were going to the same IP address of 10.0.0.1, yet the TTL of each of the packets varied wildly between 199 and 254. You would expect to see the TTLs closer together if going from the same subnet to the same IP address. Especially those coming from the same machine and operating system which is evident by the MAC address. This MAC address will play an important role in future analysis. The source IP range went from .219 to .254 and then

Part 2: Network Detects
Page 22 of 73
from .50 to .96 with most of the increments being by one.

3. An additional indicator is that almost all of the packets have the wrong ICMP checksums. All packets in the trace, except packet #55 contained wrong ICMP checksums. Many of the spoofing programs are unable to calculate correct ICMP checksums or choose to calculate them incorrectly. Also notice that the entire trace is ICMP. You will find echo request and replies do not add up. There is a reason this could occur. One possibility is that we are not seeing both sides of the conversation. Maybe Mr. Buckley did not post a full capture of the detect, or the IDS was unable to handle all of the traffic and dropped some of the packets.

4. Another characteristic that lends support to this being packet spoofing is packet #56 shown below:

   56 06:53:31.813043 0:d0:58:43:38:80 0800 70: xxx.xxx.xxx.65 > 10.0.0.1:
icmp: host xxx.xxx.xxx.65 unreachable (ttl 255, id 3281, len 56)
0x0000 4500 0038 0cd1 0000 ff01 28bf cff2 ac41 E..8.....(....A
0x0010 0a00 0001 0301 d6a2 0000 0000 4500 00b4 ............E...
0x0020 eb9d 0000 d806 7071 0a00 0001 cff2 ac41 ....pq........A
0x0030 006e 1bed 0a00 0001 .n......

This is the only packet which contains the packet header that caused the ICMP unreachable to be sent. Dr. Richard Stevens book TCP/IP Illustrated Volume 1 The Protocols states on page 70 that “When an ICMP error message is sent, the message always contains the IP header and the first 8 bytes of the IP datagram that caused the ICMP error to be generated.” Notice first that this packet was sent from IP xxx.xxx.xxx.65 to 10.0.0.1 stating that host xxx.xxx.xxx.65 is unreachable. This should not be seen. A computer should not tell another computer that it is unreachable. Also notice this is one of the packets containing the source MAC address. Let’s look at the trace and specifically at the IP header in bold contained within the ICMP reply. I assume the hex conversion process to be understood by the reader. Everything appears normal: 4500 (IP version 4, Internet Header Length (IHL) 5 and no Type of Service (ToS)), 00b4 (Total packet length is 180), eb9d (ID field is 60317), 0000 (no fragments its 0), d806 (TTL is 216 and the protocol is TCP), 7071 (Header checksum is 28785), 0a00 0001 (source IP: 10.0.0.1), cff2 ac41 (destination IP: 207.242.172.65), 006e (source port: 110), 1bed (Destination Port: 7149). I am not sure why this would occur; nothing seems abnormal except the host unreachable being sent.

5. The final aspect of this being packet spoofing is packet #16 which contains a timestamp reply with the following data passed:

   timestamp reply id 666 seq 0 : org 0xa00000001 recv 0x50180774 xmit 0x2d000000

At first glance, nothing appears abnormal till you convert the data: org = 167,772,161; recv = 1,343,752,052; and xmit = 754,974,720. According to Dr. Richard Stevens book TCP/IP Illustrated Volume 1 The Protocols on page 75 states “Since the timestamp values are the number of milliseconds past midnight, UTC they should always be less than 86,400,000 (24 x 60 x 60 x 1000).” As you can see from the above timestamps, we are just a little over.

**Description of the attack:**
According to Mr. Buckley, the entire duration of the activity was for one minute and it was nothing but ICMP. The initial diagnosis of Stacheldraht at first was misleading. Mr. Buckley thought it looked strange, but another individual responded that it was Stacheldraht by virtue of “You can see the ECHO REPLY packet containing the passphrase of "sicken."” However, after dumping the packet fields into an Excel spreadsheet, it appeared to be missing some important characteristics. Mr. David Dittrich wrote an excellent paper entitled “The “stacheldraht” distributed denial of service attack tool” and it can be found at http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt. According to his analysis Stacheldraht uses TCP as well as ICMP. There should be traffic seen to/from TCP port 16660 or 65000 depending on whether this was an agent/handler/client. None of this was found in the trace. Also, Stacheldraht uses Blowfish for its encryption of the traffic. However, several of the packets contained payloads that were not encrypted and will be discussed further in the attack Mechanism.

It appears that this is a scan of some sort, but as of now I have not been able to identify the exact tool in use. The following table shows a breakdown of the ICMP usage. The first column is the red number used earlier to help identify which packet was being discussed.

<table>
<thead>
<tr>
<th>Packet Seq</th>
<th>SRC MAC Address</th>
<th>Source IP</th>
<th>Protocol</th>
<th>ICMP Type/Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>xxx.xxx.xxx.219</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: host 0.0.0.0 unreachable- admin prohibited</td>
<td>Type 3, Code 10: Router selection</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: ip reassembly time</td>
<td>Type 11, Code 1: Fragment Reassembly Time Exceeded</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>xxx.xxx.xxx.221</td>
<td>icmp: type-#20</td>
<td>Type 20: Reserved</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0:d0:58:43:38:80</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0:d0:58:43:38:80</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>xxx.xxx.xxx.224</td>
<td>icmp: type-#31</td>
<td>Type 31: Datagram Conversion Error</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: parameter problem – code 2</td>
<td>Type 12, Code 2: Bad Length</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0:d0:58:43:38:80</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>xxx.xxx.xxx.229</td>
<td>icmp: type-#40</td>
<td>Type 40, Code 1: Authentication Failed</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0:d0:58:43:38:80</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>xxx.xxx.xxx.232</td>
<td>icmp: type-#110</td>
<td>Type 110: Unassigned????</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: source quench</td>
<td>Type 4: Source Quench</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0:d0:58:43:38:80</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>xxx.xxx.xxx.234</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>xxx.xxx.xxx.236</td>
<td>icmp: time stamp reply id 666 seq 0 : org 0xa000001 recv 0x50</td>
<td>Type 14: Timestamp Reply</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0:d0:58:43:38:80</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: redirect-#34 0.0.0.0 to net 114.101.45.225</td>
<td>Type 5: Redirect</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>xxx.xxx.xxx.238</td>
<td>icmp: type-#39</td>
<td>Type 39: SKIP</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>xxx.xxx.xxx.240</td>
<td>icmp: router solicitation</td>
<td>Type 10: Router Selection</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: parameter problem – code 2</td>
<td>Type 12, Code 2: Bad Length</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>xxx.xxx.xxx.241</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>xxx.xxx.xxx.242</td>
<td>icmp: type-#40</td>
<td>Type 40, Code 0: Bad SPI</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: time exceeded-#110</td>
<td>Type 11: TTL exceeded in transit</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.243</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>26</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.244</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>27</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.246</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
</tr>
<tr>
<td>28</td>
<td>xxx.xxx.xxx.247</td>
<td>icmp: type-#35</td>
<td>Type 35: Mobile Registration Request</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>xxx.xxx.xxx.248</td>
<td>icmp: router solicitation</td>
<td>Type 10: Router Selection</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>xxx.xxx.xxx.249</td>
<td>icmp: type-#39</td>
<td>Type 39: SKIP</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: host 0.0.0.0 unreachable- admin prohibited</td>
<td>Type 3, Code 10: Communication with Destination Host is Administratively Prohibited</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: 0.0.0.0 unreachable -source host isolated</td>
<td>Type 3, Code 8: Source Host Isolated</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: redirect-tos 0.0.0.0 to net 0.0.0.0</td>
<td>Type 5, Code 3: Redirect Datagram for the Type of Service and host</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: redirect-tos 0.0.0.0 to net 0.0.0.0</td>
<td>Type 5, Code 3: Redirect Datagram for the Type of Service and host</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>xxx.xxx.xxx.252</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.253</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>37</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.254</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>38</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: 0.0.0.0 protocol 151 unreachable</td>
<td>Type 3, Code 2: Protocol Unreachable</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: 0.0.0.0 unreachable -need to frag</td>
<td>Type 3, Code 4: Fragmentation Needed and Don't Fragment was Set</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: source quench</td>
<td>Type 4: Source Quench</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>xxx.xxx.xxx.50</td>
<td>icmp: type-#116</td>
<td>Type 116: Unassigned</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: net 0.0.0.0 unreachable -tos prohibited</td>
<td>Type 3, Code 11: Destination Network Unreachable for Type of Service</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>xxx.xxx.xxx.51</td>
<td>icmp: type-#36</td>
<td>Type 36: Mobile Registration Reply</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.52</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>45</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.53</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
</tr>
<tr>
<td>46</td>
<td>xxx.xxx.xxx.57</td>
<td>icmp: type-#30</td>
<td>Type 30: Traceroute</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.58</td>
<td>icmp: echo reply</td>
<td>Type 0: Echo Reply</td>
</tr>
<tr>
<td>48</td>
<td>xxx.xxx.xxx.60</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.61</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>50</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: xxx.xxx.xxx.220 protocol 17 unreachable</td>
<td>Type 3, Code 2:</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.62</td>
<td>icmp: echo request</td>
<td>Type 8, Echo Request</td>
</tr>
<tr>
<td>52</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: source quench</td>
<td>Type 4: Source Quench</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>xxx.xxx.xxx.220</td>
<td>icmp: parameter problem -</td>
<td>Type 12, Code 0: Pointer indicates the error</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.63</td>
<td>icmp:echo reply</td>
<td>Type 0: Echo Reply</td>
</tr>
<tr>
<td>55</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.64</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
</tr>
</tbody>
</table>

Part 2: Network Detects
Page 25 of 73
<table>
<thead>
<tr>
<th></th>
<th>Time Stamp</th>
<th>Source Address</th>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>0:d0:58:43:38:80</td>
<td>xxx.xxx.xxx.65</td>
<td>icmp:host</td>
<td>xxx.xxx.xxx.65 unreachable</td>
<td>Type 3, Code 1: Host unreachable</td>
</tr>
<tr>
<td>57</td>
<td></td>
<td>xxx.xxx.xxx.66</td>
<td>icmp:router solicitation</td>
<td>Type 10: Router selection</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td></td>
<td>xxx.xxx.xxx.70</td>
<td>icmp:echo reply</td>
<td>Type 0: Echo Reply</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td></td>
<td>xxx.xxx.xxx.71</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>xxx.xxx.xxx.72</td>
<td>icmp:echo request</td>
<td>Type 40, Code 0: Bad SPI</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td></td>
<td>xxx.xxx.xxx.73</td>
<td>icmp:echo request</td>
<td>Type 12: Destination Host Unreachable for Type Of Service</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td>xxx.xxx.xxx.74</td>
<td>icmp:echo request</td>
<td>Type 13: Time Stamp Query</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td></td>
<td>xxx.xxx.xxx.75</td>
<td>icmp:echo request</td>
<td>Type 32: Mobile Host Redirect</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
<td>xxx.xxx.xxx.76</td>
<td>icmp:echo request</td>
<td>Type 33: IPV6 Where-Are-You</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>xxx.xxx.xxx.77</td>
<td>icmp:echo request</td>
<td>Type 3: Source Route Failed</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td></td>
<td>xxx.xxx.xxx.78</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td></td>
<td>xxx.xxx.xxx.79</td>
<td>icmp:echo request</td>
<td>Type 15: Precedence Cutoff in Effect</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td></td>
<td>xxx.xxx.xxx.80</td>
<td>icmp:echo request</td>
<td>Type 0: Echo Reply</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td></td>
<td>xxx.xxx.xxx.81</td>
<td>icmp:echo request</td>
<td>Type 11: Code 0: Time to Live Exceeded in Transit</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>xxx.xxx.xxx.82</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td></td>
<td>xxx.xxx.xxx.83</td>
<td>icmp:echo request</td>
<td>Type 7: Unassigned</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td></td>
<td>xxx.xxx.xxx.84</td>
<td>icmp:echo request</td>
<td>Type 36: Mobile Registration Reply</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td></td>
<td>xxx.xxx.xxx.85</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td></td>
<td>xxx.xxx.xxx.86</td>
<td>icmp:echo request</td>
<td>Type 32: Mobile Host Redirect</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td></td>
<td>xxx.xxx.xxx.87</td>
<td>icmp:echo request</td>
<td>Type 3: Source Route Failed</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td></td>
<td>xxx.xxx.xxx.88</td>
<td>icmp:echo request</td>
<td>Type 13: Time Stamp Query</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td></td>
<td>xxx.xxx.xxx.89</td>
<td>icmp:echo request</td>
<td>Type 4: Source Quench</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td></td>
<td>xxx.xxx.xxx.90</td>
<td>icmp:echo request</td>
<td>Type 7: Unassigned</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td></td>
<td>xxx.xxx.xxx.91</td>
<td>icmp:echo request</td>
<td>Type 12: Code 0:</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>xxx.xxx.xxx.92</td>
<td>icmp:echo request</td>
<td>Type 32: Mobile Host Redirect</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td></td>
<td>xxx.xxx.xxx.93</td>
<td>icmp:echo request</td>
<td>Type 13: Time Stamp Query</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td></td>
<td>xxx.xxx.xxx.94</td>
<td>icmp:echo request</td>
<td>Type 12: Code 2: Bad Length</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td></td>
<td>xxx.xxx.xxx.95</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td></td>
<td>xxx.xxx.xxx.96</td>
<td>icmp:echo request</td>
<td>Type 2: Unassigned</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td></td>
<td>xxx.xxx.xxx.97</td>
<td>icmp:echo request</td>
<td>Type 32: Mobile Host Redirect</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td></td>
<td>xxx.xxx.xxx.98</td>
<td>icmp:echo request</td>
<td>Type 12: Code 2: Bad Length</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td></td>
<td>xxx.xxx.xxx.99</td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td></td>
<td></td>
<td>icmp:echo request</td>
<td>Type 9: Router Advertisement</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td></td>
<td></td>
<td>icmp:echo request</td>
<td>Type 8, Echo Request</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td></td>
<td>icmp:echo request</td>
<td>Type 2: Unassigned</td>
<td></td>
</tr>
</tbody>
</table>

As you can see from the above table, many different types of ICMP packets were sent. It
appears that this is an aggressive, but brief scan mechanism. I believe it is directed against
the gateway router to determine its configuration. There are several things which appear
in the above trace which lends support of this. If you notice, the IP address
XXX.XXX.XXX.220 appears many times throughout the trace. It is only IP address
which duplicates itself. Almost all of the ICMP messages being sent are specific to
routers. There are several redirects sent by XXX.XXX.XXX.220 and from Dr. Richard
Stevens book TCP/IP Illustrated Volume 1 The Protocols on page 122 we know that
"redirects are generated only by routers, not by hosts." (see packets 18, 33, 34, 66 and 69)
We also find Type 3, Code 15: Precedence cutoff in effect; Type 3, Code 11: Destination
Network Unreachable for Type of Service; icmp: host 0.0.0.0 unreachable-admin
prohibited, etc. allow coming from XXX.XXX.XXX.220. By virtue of the number of
times it appears and the ICMP used, I would say it is a router. However, there is always
the possibility this not the case. There are other packets that contain router specific
information such as Packet 87 sent a router advertisement, but it was not IP
XXX.XXX.XXX.220. An interesting characteristic found in all of the packets except for
three of them (1, 2, and 56) is as follows:
0x0000  4500 0030 f0a3 0000 ed01 56d5 cff2 ac60
0x0010  0a00 0001 0200 0019 22da 9090 0a00 0001
All packets contain the IP address of 10.0.0.1 in hex in the exact same location regardless
of the ICMP type. I have not be able to determine why the start of the ICMP datagram is
set to this, unless it is to ensure communication or some sort of IP tunneling.
There are several other things about this trace that offer clues, but not necessarily answers.
In packet #18, we find a redirect to net 114.101.45.225. Looking this up at www.arin.net
provided the following information.

Search results for: 114.101.45.225
IANA (RESERVED-8)
 Internet Assigned Numbers Authority
 4676 Admiralty Way, Suite 330
 Marina del Rey, CA 90292-6695
 US
 Netname: RESERVED-8
 Netblock: 96.0.0.0 - 126.255.255.255
Unfortunately, this does not tell us a lot. However we find another redirect in packet #66
to 203.237.226.116 and another search of the IP reveals better results seen below:

Query the APNIC Whois Database
% [whois.apnic.net node-2]
% How to use this server  http://www.apnic.net/db/
% Whois data copyright terms http://www.apnic.net/db/dbcopyright.html

inetnum:  203.237.216.0 - 203.237.231.255
netname:  DKUNET-KR
descr:  Dankook University
descr:  San 8 Hannam-dong Yongsan-gu
descr:  SEOUL

Part 2: Network Detects
Page 27 of 73
Part 2: Network Detects

Page 28 of 73

A redirect to Korea? It is possible, but Mr. Buckley’s email is rbuckley@synapsemail.com. A quick lookup of synapsemail.com using nslookup reveals:

> synapsemail.com
Server: dialcache040.ns.uu.net
Address: 198.6.1.140
Name: synapsemail.com

This IP address in an Arin lookup resolves to:

UUNET Technologies, Inc. (NETBLK-UUNETCBLK6)
3060 Williams Drive, Suite 601
Fairfax, VA 22031
US

By function of a redirect, a machine sitting off a router requests an IP which has a shorter route using another gateway. The router will send a redirect to the host, if on the same network, telling it to use a new gateway router which is the closest router in the path to its destination. It is hard to believe that Korea would be the next closest router. We also know since the redirect was sent, that 10.0.0.1 is on the same subnet as this router. The effect of this would be to update the routing table of the host to this new route for all requests to this network with this new gateway.

We also find several packets that need to be looked at more in depth. All of those packets with the same source MAC address, all had one thing in common. Each of these packets was either an echo request or an echo reply. Also, each of them had data with in the packets. Look at the following table constructed:

<table>
<thead>
<tr>
<th>Packet Seq</th>
<th>Source IP</th>
<th>ICMP Type/Code</th>
<th>Data Contained in Packet</th>
<th>Signature from SNORT Rule Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>xxx.xxx.xxx.222</td>
<td>Type 8, Echo Request</td>
<td>ISSPNGRQ.s.you</td>
<td>ISS Pinger</td>
</tr>
<tr>
<td>9</td>
<td>xxx.xxx.xxx.226</td>
<td>Type 0: Echo Reply</td>
<td>ficken.exe?about</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>xxx.xxx.xxx.231</td>
<td>Type 0: Echo Reply</td>
<td>AAAAAAAA.FBORF W.EXE&quot;</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>xxx.xxx.xxx.234</td>
<td>Type 8, Echo Request</td>
<td>(From Hex values)</td>
<td>Flowpoint 2200DSL Router</td>
</tr>
<tr>
<td>No.</td>
<td>Source IP Address</td>
<td>Type</td>
<td>Data</td>
<td>Comment</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>--------</td>
<td>-----------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>17</td>
<td>xxx.xxx.xxx.237</td>
<td>Type 0</td>
<td>Echo Reply</td>
<td>Shell.bound.to.port</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TFN server response</td>
</tr>
<tr>
<td>25</td>
<td>xxx.xxx.xxx.243</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>Sustainable.So</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IP NetMonitor Macintosh</td>
</tr>
<tr>
<td>26</td>
<td>xxx.xxx.xxx.244</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>(From Hex values)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cisco Type.x</td>
</tr>
<tr>
<td>27</td>
<td>xxx.xxx.xxx.246</td>
<td>Type 0</td>
<td>Echo Reply</td>
<td>sicken</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Stacheldraht server-response-gag</td>
</tr>
<tr>
<td>36</td>
<td>xxx.xxx.xxx.253</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>abcdefghijklmnop.EXE\</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Microsoft Windows: Same as 14 and 26</td>
</tr>
<tr>
<td>37</td>
<td>xxx.xxx.xxx.254</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>bcdefghijklmnop.EXE\</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Microsoft Windows: Same as 14 and 26</td>
</tr>
<tr>
<td>44</td>
<td>xxx.xxx.xxx.52</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>ABCDEFGHIJKLMNOPQRSTUWABCDEFGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PING-SCANNER-L3RETRIEVER</td>
</tr>
<tr>
<td>45</td>
<td>xxx.xxx.xxx.53</td>
<td>Type 0</td>
<td>Echo Reply</td>
<td>gesundheit!</td>
</tr>
<tr>
<td>47</td>
<td>xxx.xxx.xxx.58</td>
<td>Type 0</td>
<td>Echo Reply</td>
<td>spoofworks</td>
</tr>
<tr>
<td>49</td>
<td>xxx.xxx.xxx.61</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>(From Hex values)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Flowpoint 2200DSL Router</td>
</tr>
<tr>
<td>51</td>
<td>xxx.xxx.xxx.62</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>WhatsUp.-.A.Netw</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WhatsupGold Windows</td>
</tr>
<tr>
<td>54</td>
<td>xxx.xxx.xxx.63</td>
<td>Type 0</td>
<td>Echo Reply</td>
<td>spoofworks.ls/getdrvs.exe</td>
</tr>
<tr>
<td>59</td>
<td>xxx.xxx.xxx.67</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>(From Hex values)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Flowpoint 2200DSL Router</td>
</tr>
<tr>
<td>60</td>
<td>xxx.xxx.xxx.68</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>ISSPNGRQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ISS Pinger</td>
</tr>
<tr>
<td>61</td>
<td>xxx.xxx.xxx.69</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>(From Hex values)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Seer Windows</td>
</tr>
<tr>
<td>63</td>
<td>xxx.xxx.xxx.71</td>
<td>Type 0</td>
<td>Echo Reply</td>
<td>spoofworks.i</td>
</tr>
<tr>
<td>73</td>
<td>xxx.xxx.xxx.78</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>20&amp;CiRestriction=none &amp;CiHiliteType=Full.HTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P/1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Attempt to retrieve ASP contents</td>
</tr>
<tr>
<td>77</td>
<td>xxx.xxx.xxx.63</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>OMeterObeseArmad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ping-O-MeterWindows</td>
</tr>
<tr>
<td>78</td>
<td>xxx.xxx.xxx.85</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>Sustainable.So</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IP NetMonitor Macintosh</td>
</tr>
<tr>
<td>80</td>
<td>xxx.xxx.xxx.86</td>
<td>Type 8</td>
<td>Echo Request</td>
<td>Pinging.from.Del</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Delphi-Piette Windows</td>
</tr>
</tbody>
</table>

The majority of these had to be figured out by searching for the hex values in SNORT rule sets when looking at the data contained with in them. Why would you see so many prominent signatures? At first I was wondering about the source MAC address only listed in certain packets. I believe this was to ensure a response back from the receiving system. If two systems have the same IP address, the MAC address will determine the delivery, especially if you update the routing table to reflect an IP is at a different MAC address. It seems they were looking for something.

**Attack Mechanism**

After looking at the above analysis, I cannot say exactly what is going on. There are...
many many different possibilities. One that seems the most plausible is that this is a scan to determine a router configuration. It is more difficult not knowing the network configuration; however Mr. Buckley says that 10.0.0.1 is not a valid host. Given the information from the above analysis, it appears that a local host off of the router is attempting to scan the router for configuration information. The IP addresses used are spoofed and some undetermined program would be generating the ICMP traffic we are seeing due to the speed of the scan and the spoofing IPs. I am unable to figure out exactly which program. It has characteristics of many of them, but I don’t find one in particular that has this signature. It is possible that this is scripted ICMPush or a SING (Send ICMP Nasty Garbage) scan which can be found at http://hisphack.ccc.de/. However, I don’t find they support the IPV6 Where are you and some of the others that are found. The scan starts by spoofing IP XXX.XXX.XXX.219 and spoofs through the IP range to .XXX.XXX.XXX.254 then restarting at XXX.XXX.XXX.50 and goes to XXX.XXX.XXX.96. Keep in mind this occurs for one minute and is all ICMP traffic. I believe the different signatures are used because the router would respond differently to different type ICMP traffic from different type programs. These are the packets which contain a source MAC address, which I believe are used to ensure they receive the response to the packets. It is possible with the 10.0.0.1 appearing in the data portion of the ICMP packet in almost everyone that this is a covert channel in use and the redirect to 203.237.226.116 and to 114.101.45.225 would ensure the host of 10.0.0.1 could pass the traffic. I believe this to be an initial reconnaissance. The signatures found in the in the packets could also be used as a decoy to throw an IDS off of what is really happening.

**Correlations**

There are correlations to different aspects of the attack, but I am unable to find one that fits this pattern of traffic. Each of the individual signatures has been found in different cases:

1. ISS Pinger: [http://www.digitaltrust.it/arachnids/IDS158/event.html](http://www.digitaltrust.it/arachnids/IDS158/event.html)
   adVICE: [http://www.iss.net/security_center/advice/Intrusions/2001508/default.htm](http://www.iss.net/security_center/advice/Intrusions/2001508/default.htm)
2. Flowpoint 2200DSL Router: [http://www.digitaltrust.it/arachnids/IDS158/event.html](http://www.digitaltrust.it/arachnids/IDS158/event.html)
5. Cisco Type.x: [http://www.digitaltrust.it/arachnids/IDS153/event.html](http://www.digitaltrust.it/arachnids/IDS153/event.html)
6. Stacheldraht server-response-gag:
   [http://www.digitaltrust.it/arachnids/IDS195/event.html](http://www.digitaltrust.it/arachnids/IDS195/event.html)
7. Microsoft Windows: [http://www.digitaltrust.it/arachnids/IDS159/event.html](http://www.digitaltrust.it/arachnids/IDS159/event.html)
Evidence of active targeting

This would be active targeting. We see all packets going to 10.0.0.1. We also see the contents of the packet causing the ICMP host unreachable message to be sent in packet #56 from 10.0.0.1. Almost all of the ICMP requests or ICMP replies were router specific. Even if 10.0.0.1 were not an active host, the router would still send some responses back to the originator. The identical source MAC address would help to ensure that the information was returned.

Severity

The severity is calculated with information available. This could change if more information about the network were known.

\[
\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures})
\]

\[
\text{Criticality} = 5, \text{ the router is a critical piece of the infrastructure}
\]

\[
\text{Lethality} = 4, \text{ because we don’t know the router’s configuration, this would be a higher number due to the information that can be received by whoever is doing the reconnaissance.}
\]

\[
\text{System Countermeasures} = 3, \text{ because we don’t know what the router is; how it’s configured; or what vulnerabilities exist with it.}
\]

\[
\text{Network Countermeasures} = 2, \text{ the IDS picked up the scan, but it does not stop the scan.}
\]

\[
\text{Severity} = (5+4) - (3+2) = 4:
\]

Defensive recommendations

Since the router is outside the firewall, the IDS is a good start. Make sure the IDS has the most current rule set and that it has been properly configured. Ensure the router has all of the latest patches applied. Review the ACL list to ensure it has been configured correctly.
is and blocking dangerous ICMP. This will have to be determined by the router owner and the networks being serviced. Block all ICMP traffic going in and out of the router that is not needed. Review the logs daily and watch for malicious behavior.

**Multiple choice test question**

**Question:** Time Stamp replies should always be
a. greater than 86,400,000
b. less than 86,400,000
c. less than 60,400,000
d. greater than 60,400,000

**Answer:** B, the way to calculate the timestamp is: (24 x 60 x 60 x 1000)

**Detect 2**

May 04 15:13:54.192847 213.114.155.74.10363 > A.B.24.105.32320: R 0:0(0) ack
2093292673 win 0
May 10 10:32:02.907545 202.96.170.175.23132 > A.B.24.105.16147: R 0:0(0) ack
2119353641 win 0 (DF)
May 10 10:33:02.244385 202.96.170.175.28393 > A.B.24.105.27350: R 0:0(0) ack
2093292673 win 0 (DF)
May 11 17:41:25.668000 195.159.0.90.25787 > A.B.24.105.50026: R 0:0(0) ack
2093292673 win 0 (DF)
May 12 20:57:40.114036 195.159.0.90.17655 > A.B.24.105.42560: R 0:0(0) ack
2093292673 win 0 (DF)
May 13 02:43:49.277926 210.51.195.242.30405 > A.B.24.105.55321: R 0:0(0) ack
2093292673 win 0
May 13 02:47:42.141686 210.51.195.242.13712 > A.B.24.105.13470: R 0:0(0) ack
2119353641 win 0
May 13 03:08:44.392753 210.51.195.242.14624 > A.B.24.105.25786: R 0:0(0) ack
2119353641 win 0
May 13 03:09:02.581235 210.51.195.242.21772 > A.B.24.105.55043: R 0:0(0) ack
2093292673 win 0
May 13 03:10:47.108680 210.51.195.242.16260 > A.B.24.105.50721: R 0:0(0) ack
2093292673 win 0
May 13 03:23:01.695751 210.51.195.242.24690 > A.B.24.105.43529: R 0:0(0) ack
2093292673 win 0
May 13 03:30:40.841510 210.51.195.242.20326 > A.B.24.105.32961: R 0:0(0) ack
2119353641 win 0
May 13 03:53:25.418298 195.159.0.90.28711 > A.B.24.105.54951: R 0:0(0) ack
2093292673 win 0 (DF) [tos 0x60]
May 13 19:23:30.740548 202.103.196.69.5890 > A.B.24.105.55141: R 0:0(0) ack
2093292673 win 0
May 14 09:14:44.181069 202.108.58.52.18598 > A.B.24.105.19788: R 0:0(0) ack
2119353641 win 0
May 14 16:53:22.218980 195.159.0.90.14934 > A.B.24.105.42941: R 0:0(0) ack
Source of Trace:

Part 2: Network Detects

Page 33 of 73
The trace was taken from http://cert.uni-stuttgart.de/archive/intrusions/2002/05/msg00402.html and posted by Mr. Michael Scott. He does not list what the network configuration looked like.

**Detect was generated by:**
The log format is TCPdump, but what captured the trace is not given.

**Probability the source address was spoofed:**
The probability the IP addresses are spoofed is low. There are 14 different IP addresses used over a 13 day period. As you can see below, the IPs break down as follows: one from Norway, one with AT&T, one from Italy and 11 from China. If the resets were deliberate, then the originator would want to see a reply back. This definitely not a DoS (Denial of Service). If the destination IP A.B.24.105 was being used against another system, the IP we are responding to with a reset would be the IP being scanned/attacked and would still be a legitimate IP.

```
195.159.0.90
inetnum: 195.159.0.0 - 195.159.6.63
netname: POWERTECH-CORE-NETS
descr: PowerTech, Oslo, Norway
country: NO
```

```
202.103.196.61, 202.103.196.69
inetnum: 202.103.192.0 - 202.103.255.255
netname: CHINANET-GX
descr: CHINANET Guangxi province network
descr: Data Communication Division
descr: China Telecom
country: CN
```

```
202.108.58.52
inetnum: 202.108.58.0 - 202.108.58.255
netname: REDSAIL-INFOR-TECH-CO
descr: Beijing Telecom Red Sail Information
descr: Technology Co.Ltd
country: CN
```

```
202.96.170.175
inetnum: 202.96.128.0 - 202.96.191.255
netname: CHINANET-GD
descr: CHINANET Guangdong province network
descr: Data Communication Division
descr: China Telecom
country: CN
```

```
210.51.195.242
netname: ZHENJIANG-JUYOU-NETBAR
descr: zhenjiang city
```

Part 2: Network Detects
Page 34 of 73
country: CN

211.144.65.118
inetnum: 211.144.65.1 - 211.144.65.255
netname: XUHUI1POPNET
descr: Cable Online Network XUHUI1POPNet
descr: Internet Service Provider
descr: Shanghai China
country: CN

211.155.241.86
inetnum: 211.155.241.80 - 211.155.241.95
netname: BEI-YONG
descr: BEIJING BEI-YOUG-KE-JI CO.LTD
descr: Co.Ltd
descr: Beijing
country: CN

213.114.155.74
inetnum: 213.112.0.0 - 213.115.255
netname: SE-CYBER-20000314
descr: Provider Local Registry
country: SE

213.156.32.125
inetnum: 213.156.32.0 - 213.156.32.255
netname: FASTWEB-DATACENTER
descr: Streaming and gaming public subnet
descr: Infrastructure for Fastweb's main location
country: IT

218.1.1.158
inetnum: 218.1.0.0 - 218.1.255.255
netname: CHINANET-SH
descr: CHINANET Shanghai province network
descr: Data Communication Division
descr: China Telecom
country: CN

32.97.166.142
OrgName: AT&T Global Network Services
OrgID: ATGS
NetRange: 32.0.0.0 - 32.255.255.255
CIDR: 32.0.0.0/8

61.139.77.80
inetnum: 61.139.77.0 - 61.139.77.255
netname: CHENGDU-SCINFO-IDC
descr: Sichuan Public Information Industry Co.Ltd IDC
descr: ChengDu, Sichuan
descr: PR China
country: CN

Part 2: Network Detects
Page 35 of 73
61.144.236.154

| inetnum:  | 61.144.0.0 - 61.144.255.255 |
| netname:  | CHINANET-GD |
| descr:    | CHINANET Guangdong province network |
| descr:    | Data Communication Division |
| descr:    | China Telecom |
| country:  | CN |

**Description of the attack:**

Before we start, it is important to note that the same two ack numbers of 2093292673 and 2119353641 are used in all of the packets except one. The packet from Italy had an ack number of 1702151370. Upon looking at this packet, the IP is 213.156.32.125 and it is registered to Streaming and gaming public subnet. The source port was checked on a [www.google.com](http://www.google.com) query and returned a hit on [http://216.239.51.100/search?q=cache:Kik6rn_w9KM:runarena.com/stats/cs27015%4e1/pbc/81265/+%22+Port+20404%22&hl=en&ie=UTF-8](http://216.239.51.100/search?q=cache:Kik6rn_w9KM:runarena.com/stats/cs27015%4e1/pbc/81265/+%22+Port+20404%22&hl=en&ie=UTF-8) which is a gaming site. This appears to be legitimate traffic of someone looking for a gaming port.

The other traffic is all resets. This does not appear to be a scan, because they are all going to one IP address. You wouldn’t learn much from this. However, the IP address of 195.159.0.90 resolves to:

| inetnum:  | 195.159.0.0 - 195.159.6.63 |
| netname:  | POWERTECH-CORE-NETS |
| descr:    | PowerTech, Oslo, Norway |
| country:  | NO |

Another interesting characteristic of this IP is that it is an IRC server for Norway and is listed on many sites such as [http://www.frenzy.com/~dougmc/irc-stats/server-lists/server-list.990325](http://www.frenzy.com/~dougmc/irc-stats/server-lists/server-list.990325) as: irc.homelien.no 195.159.0.90 Norway. An nslookup also provides the following results:

C:\> nslookup
Default Server: dialcache040.ns.uu.net
Address: 198.6.1.140

> 195.159.0.90
Server: dialcache040.ns.uu.net
Address: 198.6.1.140

Name: irc.homelien.no
Address: 195.159.0.90

It is unlikely that 13 different IP addresses over a 13 day period would randomly choose the same IP to use for whatever purpose, especially since the majority of this is from China. In addition to this, all spoof the same two ack numbers. This appears to be a coordinated effort if indeed these are not spoofed IPs. The IRC server’s IP appearing poses some interesting possibilities as hackers use the IRC quite frequently. It could be
efforts were coordinated over IRC. If so, these resets could be to verify the host is up and available for some purpose. Keep in mind this is only a theory, but one I would pursue if I were looking at it and had all of the facts.

**Attack Mechanism**

There are two ways this traffic could occur. The first is that a crafted syn packet was sent to the source IP address with the destination IP address of A.B.24.105 and we are seeing the rst/ack being sent back. This does not make much sense, because nothing would be gained by an attacker as they would not see the reset. Unless we don’t see they bigger reset scan, or they are sniffing the traffic before it arrives. However, a reset sent to A.B.24.105 can be useful in determining if the machine is a valid host and especially if it is up and running. If the IP is not valid or not alive, the router would send a host unreachable ICMP error message. A host that is up would not respond to a reset. In an inverse scan of this nature, you would discard the host unreachable messages and those from which you did not receive a reply would be considered alive. With the crafted packets and the few packets received on any given day, it appears they could be checking to see if the host is up. The most packets received on a given day are seven and they were spaced over a period of 1-1/2 hours. It is also possible that this A.B.24.105 is participating in IRC chats or the IP is behind a firewall or proxy and someone is trying to knock them offline.

**Correlations**

It is hard to determine exact correlations because many analysts ignore resets as being harmless. [http://archives.neohapsis.com/archives/incidents/2000-11/0115.html](http://archives.neohapsis.com/archives/incidents/2000-11/0115.html) has an example of more resets coming in for an undetermined reason. I also searched [www.google.com](http://www.google.com) for IP 195.159.0.90 and found another incident in July 2002 of an intrusion attempt from this IP address. This can be found at: [http://tyholt.uninett.no/pipermail/ripe-notify/2002-July/034431.html](http://tyholt.uninett.no/pipermail/ripe-notify/2002-July/034431.html).

There are noted vulnerabilities associated with using resets:
   
   BUGTRAQ: 19981005 New Windows Vulnerability
   
   URL:[http://www.securityfocus.com/archive/1/10789](http://www.securityfocus.com/archive/1/10789) “(Public chat connections such as IRC have been found to be susceptible to this attack. These are particularly fun as you get to see them being reset (again and again :))).”
   
   XG:nt-brkill(1383)
   
   URL:[http://xforce.iss.net/static/1383.php](http://xforce.iss.net/static/1383.php)

**Evidence of active targeting**

Part 2: Network Detects
Page 37 of 73
This appears to be active targeting. Given the nature that the packets are from all over China, one from AT&T and one from Italy, it appears that this is active targeting since it is unlikely they all chose the same IP. Also, the two ack numbers used are used by all of the IPs.

**Severity**
The severity is calculated with information available. This could change if more information about the network were known.

\[
\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures})
\]

- **Criticality** = 4, the individual who submitted these detects was concerned and it was found in the logs. Since we don’t know the system, but it was actively used by a wide spread range of IPs, we will give it a 4.
- **Lethality** = 3, because we don’t 100% we are the direct receivers of the resets, I give this a 3 because they will gain information on whether the system is up and we also don’t know what will follow.
- **System Countermeasures** = 3, we don’t know what the system is; how it’s configured; or what vulnerabilities exist with it.
- **Network Countermeasures** = 2, the IDS picked up the scan, but it does not stop the scan and we don’t know what the system is.

\[
\text{Severity} = (4+3) - (3+2) = 2:
\]

**Defensive recommendations**
Ensure the system is properly patched and the correct security features for it in place.
Ensure it is watched for anything unusual since it was targeted. If possible, protect it with a firewall or if outside the firewall, make sure that a good ACL list is on the router.

**Multiple choice test question**

*Question:* A reset is handled in what fashion when received by a host?

a. Respond with an ack
b. Respond with a reset
c. If a router receives it and the host is unavailable respond with an ICMP Time exceeded in Transit
d. Silently drop the packet.

*Answer:* D, A host should never respond to a reset. It should always drop the packet without a reply.

**Detect 3**

12:59:34.427801 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: icmp: echo request (frag 44560:1480@0+)
12:59:34.427801 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@50320+)
12:59:34.427801 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@51800+)
12:59:34.427801 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@53280+)
12:59:34.427801 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@54760+)
12:59:34.427801 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@56240+)
12:59:34.437800 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@57720+)
12:59:34.437800 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@59200+)
12:59:34.437800 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@60680+)
12:59:34.437800 > d226-19-71.home.cgocable.net >
ct299951-b.edgewd1.ky.home.com: (frag 43565:1480@62160+)
12:59:34.437800 > d226-19-71.home.cgocable.net >
d226-19-71.home.cgocable.net: (frag 44560:1480@63640)
12:59:34.457799 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@1480+)
12:59:34.477797 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@2960+)
12:59:34.507795 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@4440+)
12:59:34.537793 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@5920+)
12:59:34.557791 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@7400+)
12:59:34.587789 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@8880+)
12:59:34.617787 < port90.dsl-vj.adsl.cybercity.dk >
d226-19-71.home.cgocable.net: (frag 44560:1480@10360+)
12:59:35.087752 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: icmp: echo request (frag 58961:1480@0+)
12:59:35.267739 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@1480+)
12:59:35.317735 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@2960+)
12:59:35.377731 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@4440+)
12:59:35.467724 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@5920+)
12:59:35.557717 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@7400+)
12:59:35.657710 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@8880+)
12:59:35.747703 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@10360+)
12:59:35.847696 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@11840+)
12:59:35.937689 < D5E02291.kabel.telenet.be >
d226-19-71.home.cgocable.net: (frag 58961:1480@13320+)
d226-19-71.home.cgocable.net: icmp: echo request (frag 56714:1480@0+)
d226-19-71.home.cgocable.net: (frag 56714:1480@1480+)
d226-19-71.home.cgocable.net: (frag 56714:1480@2960+)

Part 2: Network Detects
Page 39 of 73
Source of Trace:

Located at http://lists.jammed.com/incidents/2002/01/0168.html. This trace was submitted by Sebastian Ip. We have no information of the network configuration.

Detect was generated by:

The output is TCPdump, but I don’t know what detected the traffic. We do know that it was directed to Mr. Ip’s network. However, one set of traces appear to be from Mr. Ip’s network. www.arin.net and www.ripe.net has the IP addresses registered as follows:

d226-19-71.home.cgocable.net (IP address from nslookup: 24.226.19.71)
Customer Name: Cogeco Cable Solutions
Address: 950 Syscon Drive Burlington, ON L7R 4S6
Country: CA

port90.ds1-vj.adsl.cybercity.dk (IP address from nslookup: 212.242.123.157)
inetnum: 212.242.96.0 - 212.242.127.255
netname: DK-CYBERCITY-POPS1
descr: CyberCity POPs
country: DK

12-248-194-107.client.attbi.com (IP address from nslookup: 12.248.194.107)
OrgName: AT&T WorldNet Services
OrgID: ATTW
Address: 400 Interpace Parkway Parsippany, NJ 07054
Country: US

D5E02291.kabel.telenet.be (IP address from nslookup: 213.224.34.145)
inethnum: 213.224.0.0 - 213.224.51.255
netname: Telenet Operaties N.V.
descr: Telenet Operaties N.V.
country: BE

The two destination addresses are:
ct299951-b.edgewd1.ky.home.com (IP address from nslookup: Non-existent domain)

*** dialcache040.ns.uu.net can't find ct299951-b.edgewd1.ky.home.com: Non-existent domain

d226-19-71.home.cgocable.net (IP address from nslookup: 24.226.19.71)
CustName: Cogeco Cable Solutions
Address: 950 Syscon Drive Burlington, ON L7R 4S6
Country: CA

**Probability the source address was spoofed:**
The probability that these are spoofed is very high. There are only four source IPs however, d226-19-71.home.cgocable.net is also a destination IP and receives the majority of the traffic. The fragmentation we see appears to be malicious and a response back would not be the intention of the sender. The host d226-19-71.home.cgocable.net (IP 24.226.19.71) also appears on the active proxy list at www.lachuleta.org which has tools for mIRC.

**Description of the attack:**
What we see occurring is malicious fragmentation. Fragmentation is dangerous because it can pass through many firewalls, IDSs, routers and other devices that are designed to provide network security. Mr. Id stated he believed that the ICMP echo request was causing his systems to respond. This does not appear to be the case. When looking at fragmentation, it is important to look at the fragmentation (frag) ID. This ID is the IP identification number taken from the IP header. Fragmentation packets do not necessarily arrive in the order they were sent. As a quick overview, fragmentation appears in the following format:

```
12:59:36.507647 < D5E02291.kabel.telenet.be > d226-19-71.home.cgocable.net: (frag 58961:1480@22200+)
```

The frag ID is 58961 and all fragments that relate to this packet will have that frag ID so they can be reassembled. The 1480 is the length of the data contained. The @22200+ is the offset in the original packet and the + means more fragments follow. The first packet in the fragmentation is the only one that has the protocol header. You will not be able to tell from the rest of the fragments what protocol is being used. This will be key later on in the analysis.

If you sort the packets according to the frag ID, you get a different picture than you see in the original trace as it is sorted by the time. Here is what it looks like sorted by the frag ID:

<table>
<thead>
<tr>
<th>Time</th>
<th>Source Address</th>
<th>Destination Address</th>
<th>Protocol</th>
<th>Fragmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:59:34.427801</td>
<td>d226-19-71.home.cgocable.net</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>(frag 43565:1480@50320+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.427801</td>
<td>d226-19-71.home.cgocable.net</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>(frag 43565:1480@51800+)</td>
<td></td>
</tr>
</tbody>
</table>

Part 2: Network Detects
Page 41 of 73
<table>
<thead>
<tr>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
<th>Event Description</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:59:34.42780</td>
<td>d226-19-71.home.cgocable.net</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>(frag 43565:1480@53280+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.42780</td>
<td>d226-19-71.home.cgocable.net</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>(frag 43565:1480@54760+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.42780</td>
<td>d226-19-71.home.cgocable.net</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>(frag 43565:1480@56240+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.43780</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>(frag 43565:1480@57720+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.43780</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>(frag 43565:1480@59200+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.43780</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>(frag 43565:1480@60680+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.43780</td>
<td>ct299951-b.edgewd1.ky.home.com:</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>(frag 43565:1480@62160+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.43780</td>
<td>port90.ds1-vj.adsl.cybercity.dk</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>icmp: echo request</td>
<td></td>
</tr>
<tr>
<td>12:59:34.45779</td>
<td>port90.ds1-vj.adsl.cybercity.dk</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>(frag 44560:1480@10360+)</td>
<td></td>
</tr>
<tr>
<td>12:59:34.47779</td>
<td>port90.ds1-vj.adsl.cybercity.dk</td>
<td>d226-19-71.home.cgocable.net:</td>
<td>(frag 44560:1480@1480+)</td>
<td></td>
</tr>
<tr>
<td>12:59:36.03768</td>
<td>D5E02291.kabel.telenet.be</td>
<td>D5E02291.kabel.telenet.be:</td>
<td>icmp: echo request</td>
<td></td>
</tr>
<tr>
<td>12:59:36.12767</td>
<td>D5E02291.kabel.telenet.be</td>
<td>D5E02291.kabel.telenet.be:</td>
<td>(frag 58961:1480@11840+)</td>
<td></td>
</tr>
<tr>
<td>12:59:36.21766</td>
<td>D5E02291.kabel.telenet.be</td>
<td>D5E02291.kabel.telenet.be:</td>
<td>(frag 58961:1480@17760+)</td>
<td></td>
</tr>
<tr>
<td>12:59:36.31766</td>
<td>D5E02291.kabel.telenet.be</td>
<td>D5E02291.kabel.telenet.be:</td>
<td>(frag 58961:1480@18240+)</td>
<td></td>
</tr>
<tr>
<td>12:59:36.40765</td>
<td>D5E02291.kabel.telenet.be</td>
<td>D5E02291.kabel.telenet.be:</td>
<td>(frag 58961:1480@17200+)</td>
<td></td>
</tr>
<tr>
<td>12:59:36.50764</td>
<td>D5E02291.kabel.telenet.be</td>
<td>D5E02291.kabel.telenet.be:</td>
<td>(frag 58961:1480@22200+)</td>
<td></td>
</tr>
</tbody>
</table>

Part 2: Network Detects
Page 42 of 73
Only three of the four source IP addresses contain the first fragment. From these you see they are ICMP echo requests. This in itself is unusual due to the massive size of the ICMP echo request packets. However, none of the packet totals exceed the threshold of 65,535 bytes allowed in an ICMP packet. There are a couple of other things wrong with what we see. The first fragmented packet, with a frag ID of 43565, does not have the first fragment with an offset of 0, but it does have the final fragment. The other three fragmented packets are just the opposite. They all have the first fragment, but none of them have the final fragment. They all have the more fragments follows flag set. It is unknown what data was contained in these packets.

### Attack Mechanism

The attack is using malicious fragmentation to cause a DoS or at least degradation in service. This according to the times recorded by the capture, the entire attack took place in two seconds. We are going to focus initially on the three packets that have the initial fragments. The spoofed source address sends three large fragmented ICMP echo request packets (using programs such as Fragrouter, Packet Shell, etc) to the destination host of d226-19-71.home.cgocable.net. The packets are large in size and they do not contain the final fragment. As such, the host would try to reassemble the fragments and be waiting for the final fragment to arrive, which is never does. The massive size of the packets combined with them arriving almost simultaneously would cause a DoS to the host or a severe **degradation** in service.

The last packet to look at is the one without an initial fragment, but with a final fragment. Our not seeing the initial fragment could be because it passed through a network device that did not allow that type of protocol. It would drop that initial fragment, but allow the others to pass through. This could be normal traffic from d226-19-71.home.cgocable.net.

### Correlations

The lists of vulnerabilities associated with fragmentation on the CVE website were numerous. They ranged from vulnerabilities in firewalls, IDSs, operating systems, etc. Here are a few of them listed.

Evidence of active targeting

This is active targeting because of the malicious fragments were directed at destination host of d226-19-71.home.cgocable.net. The attacker was deliberately trying to achieve a DoS or degradation in service.

Severity

The severity is calculated with the information available. This could change if more information about the network were known.

\[
\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures})
\]

- \text{Criticality} = 4, If the device is a proxy server providing service to many customers, this would be a critical piece of the infrastructure
- \text{Lethality} = 4, this attack would cause a denial of service or a severe \text{degradation} in service.
- \text{System Countermeasures} = 3, because we don’t know what the host configuration is; or what vulnerabilities exist with it, we will give it an average number. One would hope that if it is a proxy server for an IRC channel, it would be properly hardened.
- \text{Network Countermeasures} = 2, the packets were detected, but they were allowed to pass.

\[
\text{Severity} = (4+4) - (3+2) = 3:
\]

Defensive recommendations

Ensure that all of your systems are up to date on all of their patches. Many vendors have patches available that help to prevent malicious fragmentation from having an effect. In cases like this, stateful security devices such as firewalls, routers, etc are your friend. By maintaining state, they can help defend against malicious fragmentation.

Multiple choice test question

Question: With fragmentation, all fragments should always contain:

- a. overlapping offsets
- b. a IP ID
- c. a final fragment
- d. the protocol

Answer: C
Part 3: Analyze This

Executive Summary:
We have been asked to provide a security audit for GIAC University and they have provided us with five days worth of logs to analyze. The data was collect using Snort however; they did not provide us with a copy of the rule set in use at the time. As such, we downloaded the latest rule set from www.snort.org and are using this as our basis for analysis.

Here is a listing of five days of consecutive files obtained from GIAC University. There are three types of data files to analyze and they are Scans, OOS (Out of Spec) and Alerts. Here are the files that will be analyzed for this security audit:

<table>
<thead>
<tr>
<th>Scans</th>
<th>OOS</th>
<th>Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>scans.020706.gz</td>
<td>oos_Jul.6.2002.gz</td>
<td>alert.020706.gz</td>
</tr>
<tr>
<td>scans.020707.gz</td>
<td>oos_Jul.7.2002.gz</td>
<td>alert.020707.gz</td>
</tr>
<tr>
<td>scans.020708.gz</td>
<td>oos_Jul.8.2002.gz</td>
<td>alert.020708.gz</td>
</tr>
<tr>
<td>scans.020709.gz</td>
<td>oos_Jul.9.2002.gz</td>
<td>alert.020709.gz</td>
</tr>
<tr>
<td>scans.020710.gz</td>
<td>oos_Jul.10.2002.gz</td>
<td>alert.020710.gz</td>
</tr>
</tbody>
</table>

We will analyze the logs above as a complete five day set and not individual logs. This will help with correlations and make sure we see the big picture, not just isolating one day of events. Several tools were used to complete the analysis. As a quick overview, I will list the tools used and what purpose they serve. Several different tools were used to ensure a good look at the data was accomplished. The complete description of their usage will be provided at the end of the security audit.

✓ SnortSnarf: Used to analyze the alert files against the current Snort rule set and summarize them into a web based output.
✓ Snort_Sort: Breaks the alerts down into a web based output. Lists the alerts and those packets that generated them.
✓ WinGrep: Used to generate the OOS logs into a format that could be exported into excel. Also used to look for certain pieces of information within the files.
✓ CSV.pl: Converts the alert file into a CSV format. (From Tod Beardsley’s practical found at http://www.giac.org/GCIA.php)
✓ Summarize.pl: Summarizes the data from generated from the CSV.pl into a summary looking at different aspects of the data. (From Tod Beardsley’s practical found at http://www.giac.org/GCIA.php)
✓ Alertcount.pl: Used to total the alerts. Used to compare against the snort_snarf output, since snort_snarf would not process a concatenated file of all of the logs due to a lack of memory. (From Chris Kuethe’s practical found at http://www.giac.org/GCIA.php)
✓ Scanalyze.pl: Used to process the scan logs (with the flag set not to exclude anything) into a usable format this is then passed to scancount. (From Chris Kuethe’s practical found at http://www.giac.org/GCIA.php)
✓ Scancount.pl: Used to total up the scans of the different scan types found in the Scan logs. (From Chris Kuethe’s practical found at http://www.giac.org/GCIA.php)
✓ Excel.exe: Used to organize the OOS logs into a more usable format.

The analysis will be completed by looking at the detects occurring most frequently. For our purposes, this will be those occurring greater than 1500 times over the five day period for a total of 18 detects. MY.NET was replaced with 10.0 for purposes are parsing the data. Each detect will be analyzed providing the following: a descriptions of the attack, correlations of the attack (if available) and recommendations for improving the University’s defensive posture. Snort_sort will be used in conjunction with alertcount.pl since snort_snarf could not be used to process all of the alert logs. Snort_snarf was done for each day and will be used to determine the rules and help with the correlations. Here the analyzed scan results from GIAC University's alert logs:

<table>
<thead>
<tr>
<th>Scan Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>789224</td>
<td>TFTP - Internal TCP connection to external tftp server</td>
</tr>
<tr>
<td>290278</td>
<td>Incomplete Packet Fragments Discarded</td>
</tr>
<tr>
<td>90505</td>
<td>SUNRPC highport access!</td>
</tr>
<tr>
<td>69486</td>
<td>SNMP public access</td>
</tr>
<tr>
<td>63279</td>
<td>IDSR52/web-iis IIS ISAPI Overflow ida INTERNAL nosize</td>
</tr>
<tr>
<td>53555</td>
<td>Watchlist 000220 IL-ISDNNET-990517</td>
</tr>
<tr>
<td>28533</td>
<td>SMB Name Wildcard</td>
</tr>
<tr>
<td>26191</td>
<td>NIMDA - Attempt to execute cmd from campus host</td>
</tr>
<tr>
<td>13459</td>
<td>spp http decode: IIS Unicode attack detected</td>
</tr>
<tr>
<td>11429</td>
<td>External RPC call</td>
</tr>
<tr>
<td>9210</td>
<td>Watchlist 000222 NET-NCFC</td>
</tr>
<tr>
<td>7531</td>
<td>UDP SRC and DST outside network</td>
</tr>
<tr>
<td>7135</td>
<td>TFTP - External UDP connection to internal tftp server</td>
</tr>
<tr>
<td>4334</td>
<td>High port 65535 udp - possible Red Worm - traffic</td>
</tr>
<tr>
<td>3152</td>
<td>spp http decode: CGI Null Byte attack detected</td>
</tr>
<tr>
<td>2636</td>
<td>SYN-FIN scan!</td>
</tr>
<tr>
<td>2096</td>
<td>AFS - Off-campus activity</td>
</tr>
<tr>
<td>1577</td>
<td>beetle.ucs</td>
</tr>
<tr>
<td>1241</td>
<td>Attempted Sun RPC high port access</td>
</tr>
<tr>
<td>883</td>
<td>IDSR52/web-iis IIS ISAPI Overflow ida nosize</td>
</tr>
<tr>
<td>776</td>
<td>IRC evil - running XDCC</td>
</tr>
<tr>
<td>655</td>
<td>Null scan!</td>
</tr>
<tr>
<td>325</td>
<td>IDSR452/web-iis http-iis-unicode-binary</td>
</tr>
<tr>
<td>278</td>
<td>Queso fingerprint</td>
</tr>
<tr>
<td>213</td>
<td>High port 65535 tcp - possible Red Worm - traffic</td>
</tr>
<tr>
<td>200</td>
<td>SMB C access</td>
</tr>
<tr>
<td>183</td>
<td>MYPARTY - Possible My Party infection</td>
</tr>
<tr>
<td>137</td>
<td>Possible trojan server activity</td>
</tr>
<tr>
<td>133</td>
<td>SCAN Proxy attempt</td>
</tr>
<tr>
<td>92</td>
<td>IDSR475/web-iis web-wbdav-propfind</td>
</tr>
<tr>
<td>84</td>
<td>STATDX UDP attack</td>
</tr>
<tr>
<td>61</td>
<td>EXPLOIT x86 NOOP</td>
</tr>
<tr>
<td>51</td>
<td>SMTP relaying denied</td>
</tr>
<tr>
<td>46</td>
<td>Port 55850 udp - Possible myserver activity - ref. 010313-1</td>
</tr>
<tr>
<td>44</td>
<td>TFTP - Internal UDP connection to external tftp server</td>
</tr>
<tr>
<td>39</td>
<td>Back Orifice</td>
</tr>
<tr>
<td>31</td>
<td>INFO - Possible Squid Scan</td>
</tr>
<tr>
<td>29</td>
<td>IDS305/web-iis http-iis translate f</td>
</tr>
<tr>
<td>14</td>
<td>EXPLOIT NTDPDX buffer overflow</td>
</tr>
</tbody>
</table>
### Prioritized detects/Analysis

#### Detect:
TFTP - Internal TCP connection to external tftp server: Occurrence = 789,224

#### Detect Description:
TFTP (Trivial File Transfer Protocol) uses UDP and does not provide for security of any sort. [http://www.webopedia.com/TERM/T/TFTP.html](http://www.webopedia.com/TERM/T/TFTP.html) This detect is of concern because there is no security available, it is using TCP instead of UDP, and there are multiple vulnerabilities for this type of attack. There are also a high number of these occurring in a five day period.

#### Correlations:
- Alcatel ADSL modems grant unauthenticated TFTP access via Bounce Attacks: [http://www.kb.cert.org/vuls/id/211736](http://www.kb.cert.org/vuls/id/211736)
- Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability: [http://216.239.35.100/search?q=cache:dWh3qkHmLhMC:online.securityfocus.com/bid/1806/exploit/+%22TFTP%22+vulnerabilities&hl=en&ie=UTF-8](http://216.239.35.100/search?q=cache:dWh3qkHmLhMC:online.securityfocus.com/bid/1806/exploit/+%22TFTP%22+vulnerabilities&hl=en&ie=UTF-8)
- Other detects of the same kind can be found at: [www.giac.org/practical/Edward_Peck_GCIA.doc](http://www.giac.org/practical/Edward_Peck_GCIA.doc) [www.giac.org/practical/Mike_Poor_GCIA.doc](http://www.giac.org/practical/Mike_Poor_GCIA.doc)

#### Defensive Recommendations:
GIAC University should carefully monitor the TFTP traffic and block it if necessary. They should also ensure that all of their primary systems such as switches, modems etc are protected by the recommendations of the vendor. Many more vulnerabilities can be found by doing a search on [www.google.com](http://www.google.com)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>FTP DoS flpd globbing</td>
</tr>
<tr>
<td>14</td>
<td>SCAN FIN</td>
</tr>
<tr>
<td>14</td>
<td>TCP SRC and DST outside network</td>
</tr>
<tr>
<td>13</td>
<td>NMAP TCP ping!</td>
</tr>
<tr>
<td>12</td>
<td>EXPLOIT x86 setuid 0</td>
</tr>
<tr>
<td>11</td>
<td>EXPLOIT x86 setgid 0</td>
</tr>
<tr>
<td>8</td>
<td>RFB - Possible WinVNC - 010708-1</td>
</tr>
<tr>
<td>8</td>
<td>SMB D access</td>
</tr>
<tr>
<td>6</td>
<td>SCAN Synscan Portscan ID 19104</td>
</tr>
<tr>
<td>6</td>
<td>connect to 513 from outside</td>
</tr>
<tr>
<td>4</td>
<td>EXPLOIT x86 stealth noop</td>
</tr>
<tr>
<td>4</td>
<td>SMTP chameleon overflow</td>
</tr>
<tr>
<td>4</td>
<td>Tiny Fragments - Possible Hostile Activity</td>
</tr>
<tr>
<td>3</td>
<td>External FTP to HelpDesk 130.85.70.49</td>
</tr>
<tr>
<td>2</td>
<td>BACKDOOR NetMetro Incoming Traffic</td>
</tr>
<tr>
<td>2</td>
<td>IDS50/trojan_trojan-active-subseven</td>
</tr>
<tr>
<td>2</td>
<td>IDS553/web-iis_IIS ISAPI Overflow idq</td>
</tr>
<tr>
<td>2</td>
<td>TFTP - External TCP connection to internal tftp server</td>
</tr>
<tr>
<td>1</td>
<td>FTP passwd attempt</td>
</tr>
<tr>
<td>1</td>
<td>HelpDesk 130.85.70.50 to External FTP</td>
</tr>
<tr>
<td>1</td>
<td>IDS433/web-iis_http-iis-unicode-traversal-optyx</td>
</tr>
</tbody>
</table>
Detect:
Incomplete Packet Fragments Discarded: Occurrence = 290,278

Detect Description:
Fragmentation can be a nightmare for intrusion detection. Fragmentation is dangerous because it can pass through many firewalls, IDSs, routers and other devices that are designed to provide network security. This does not mean that is the case. There are two possible reasons for seeing this. It may be malicious traffic or Mr. Martin Roesch answered a question on this message with the Snort version 1.8.2 at http://archives.neohapsis.com/archives/snort/2001-11/0822.html and recommended ensuring the individual was using the frag2 processor. Without knowing the Snort configuration, it is difficult to answer this one.

Correlations:
- Other detects of the same kind can be found at:
  - www.giac.org/practical/Matthew_Fiddler_GCIA.doc
  - www.giac.org/practical/Edward_Peck_GCIA.doc

Defensive Recommendations:
GIAC University should carefully monitor all fragmentation for possible malicious usage. Ensure all of their servers and primary network devices have tight security lockdowns and all of the latest patches and fixes. Also, ensure that Snort is using the frag2 processor.

Detect:
SUNRPC highport access! Occurrence = 789,224

Detect Description:
SUNRPC (Remote Procedure call) detect is looking to connect to port 32771 tcp/udp. This can be an attempt to hide communication.

Correlations:
- Other detects of the same kind can be found at:
  - http://www.sans.org/capsans/snort/SnortA34.txt
  - www.giac.org/practical/dana_mclaughlin_gcia.doc
  - www.giac.org/practical/Dennis_Davis_GCIA.doc

Defensive Recommendations:
GIAC University should ensure that all of the Sun servers are properly configured and locked down according to the proper procedures. Also, block access to this port if not needed on the network.

Part 3: Analyze This
Page 48 of 73
Detect:
SNMP public access Occurrence = 69,486

Detect Description:
SNMP public access is an attempted to gain access as an authorized user to a network device running SNMP. The community string is set by default, and if not changed can provide a way for hackers to gain access. The use of “public” can be an attempt to gain access to one of these devices.

Correlations:

❖ Good article on the attack:
  
  http://ki.sei.cmu.edu/idar/drill_attack.cfm?attack=SNMP%20Grabbing
  
  
  
  
❖ Other detects of the same kind can be found at:
  
  http://www.sans.org/giactc/snort/SnortA48.txt
  
  http://www.sans.org/y2k/051200.htm
  
  www.giac.org/practical/dana_mclaughlin_gcia.doc
  
  www.giac.org/practical/Dennis_Davis_GCIA.doc

Defensive Recommendations:
The number of vulnerabilities that exist with SNMP is growing large. This is just one of them. GIAC University should ensure that the SNMP community string has been changed. If it is not necessary, do not run SNMP. Ensure all systems are locked down, patched and up to date. Identify all devices running SNMP and ensure they have no known vulnerabilities left unpatched. If not running it, consider block access at the router (I know it’s difficult in an university environment)

Detect:
IDS552/web-iis_IIS_ISAPI_Overflow ida INTERNAL nosize: Occurrence = 63,279

Detect Description:
This means that an attempt has been made to compromise or recon an IIS server. If this occurs, system level access can be gained. In this event, the IP address is usually not spoofed since it requires a TCP connection to be established. It would warrant further investigation. It could also be Code Red or some other similar worm. The packets would need to be looked at closer to determine what the intent of the attack.

Correlations:

❖ IDS552/IIS ISAPI OVERFLOW IDA: http://www.whitehats.com/info/IDS552
  
❖ Other detects of the same kind can be found at:
  
  www.giac.org/practical/Stan_Hoffman_GCIA.doc
  
  www.giac.org/practical/Matthew_Fiddler_GCIA.doc
Defensive Recommendations:
GIAC University should carefully monitor this traffic and determine if it is Code Red or Nimda attempts. They should also ensure that all of their web servers are patched to defend against this. If it is Code Red or Nimda, http://www.cert.org has advisories for how to create ingress and egress filtering to defend against this. If it is not one of these, further investigation is warranted as to the activities of the attacker.

Detect:
Watchlist 000220 IL-ISDNNET-990517: Occurrence = 53,555

Detect Description:
This detect is one that watches IL-ISDNNET-990517 for activity. This IP range from Israel has been known for malicious activity and to such an extent that a watch was created. In looking at the data from GIAC University we see several activities originating from 212.179.XXX.XXX subnet. These are some of the ones producing the most traffic:

- NMSD listens on Port 1239, but I am unable to determine what that is or if something else is occurring.
  07/07-02:46:55.698426 212.179.43.225:17040 -> 10.0.111.130:1239

- This appears to be the CD Database Protocol (CDDBP) which uses port 888. It is database storage for music CD and allows access and downloads.
  (www.giac.org/practical/Edward_Peck_GCIA.doc)
  07/07-04:20:56.505706 212.179.105.44:2958 -> 10.1.163.240:888

- Kazaa uses port 1214. It is possible someone has a machine configured as a Supernode to be able to allow others to upload and download files that are shared. This is concerning with the IP coming from Israel.
  07/07-11:15:07.307631 212.179.126.3:18014 -> 10.1.88.162:1214

- IANA has port 1057 registered to STARTRON which is an Internet game. More information can be found at http://www.startron.org/support_main.html
  07/07-15:09:26.522975 212.179.35.119:1214 -> 10.0.150.209:1057

- There were various connections from port 80 to numerous destination ports.
  07/07-15:09:47.820529 212.179.66.17:80 -> 10.0.150.209:1072
  07/07-21:51:49.294236 212.179.66.17:80 -> 10.0.110.224:1059
  07/10-09:47:58.782895 212.179.35.128:80 -> 10.0.84.191:1149

- Port 1037 is unassigned, but many Microsoft Operating systems use it for communications including NBT. It is worth watching.
  07/07-21:51:40.353047 212.179.35.119:1214 -> 10.0.150.209:1037

- Multiple attempts to port 80. This could be numerous things especially if port 80 allows for unrestricted access.
  07/08-06:06:20.036178 212.179.42.189:15532 -> 10.0.99.174:80

- The Remote USB System Port is listening on 3422.
  07/10-01:18:31.854357 212.179.32.130:54435 -> 10.0.110.92:3422

Correlations:
- Other detects of the same kind can be found at:
  www.sans.org/y2k/051900.htm
  www.giac.org/practical/Rick_Yuen_GCIA.doc

Part 3: Analyze This
Page 50 of 73
Defensive Recommendations:
GIAC University should carefully monitor this traffic and determine if this is authorized traffic for their network. If not, I would advise blocking it at the router and/or firewall for all IP addresses from this range for the unsolicited traffic. Part of this could be legitimate traffic. They should also ensure that all of their key network devices are patched to defend against this.

Detect:
SMB Name Wildcard: Occurrence = 28,533

Detect Description:
This detect is one that watches for NetBIOS traffic. You will see this alert on normal activities of Windows systems, especially when file sharing is enabled. This should be watched closely when it originates from an external network to an internal network as this is used as a preattack probe. There is some good information on this found at http://www.finchhaven.com/pages/incidents/030102_udp_137.html The majority of this traffic looks like normal NetBIOS traffic on Port 137. There are a couple of exceptions of traffic originating from outside GIAC University. These should be followed up on and flag the IPs for future activity. These are some of those that are from outside sources:

07/07-02:27:10.043732 203.218.7.171:3016 -> 10.0.82.2:137
inetnum: 203.218.0.0 - 203.218.255.255
netname: NETVIGATOR
descr: PCCW Limited
descr: PO Box 9896 GPO Hong Kong
country: HK

netname: CHINANET-NM
descr: CHINANET Neimenggu province network
descr: Data Communication Division
descr: China Telecom
country: CN

07/07-03:01:33.809627 216.78.248.247:137 -> 130.85.85.97:137
OrgName: BellSouth.net Inc.
OrgID: BELL

07/07-02:40:05.001892 209.158.44.22:137 -> 130.85.111.130:137
OrgName: Integrity Total Systems, Inc.
OrgID: ITS-36

07/07-02:40:17.744012 63.183.192.115:137 -> 130.85.111.130:137
OrgName: Sprint
OrgID: SPDN

Part 3: Analyze This
Page 51 of 73
Correlations:
- Other detects of the same kind can be found at:
  - www.chrisgrout.com/data/chrisgrout_gcia.pdf
  - www.giac.org/practical/Robert_Nine_GCIA.doc
  - www.giac.org/practical/chris_kuethen_gcia.html

Defensive Recommendations:
GIAC University should block all incoming and outgoing NetBIOS traffic at the border router or firewall as it is not needed for the functionality of the network. Ensure all systems are locked down and patched appropriately.

Detect:
NIMDA - Attempt to execute cmd from campus host: Occurrence = 26,191

Detect Description:
NIMDA is a worm that propagates itself via email, web services and file sharing. The alert triggered on an internal host that was infected and trying to look for other IIS servers to infect. Instructions for removal can be found at CERT® Advisory CA-2001-26 Nimda Worm at URL http://www.cert.org/advisories/CA-2001-26.html. Here are the IP addresses found. These are compromised systems and need to be fixed.

07/06-00:16:34.142358 10.0.105.120:4044 -> 63.79.65.244:80
07/06-00:16:34.358029 10.0.117.27:3792 -> 0.71.160.76:80

Correlations:
- Other detects of the same kind can be found at:
  - www.giac.org/practical/Rick_Yuen_GCIA.doc
  - www.giac.org/practical/Gregory_Lajon_GCIA.doc

Defensive Recommendations:
GIAC University should immediately patch all systems running IIS and ensure they configured correctly and securely. Some defense can be provided by using ingress filters and blocking traffic originating from outside the network. This may not be practical for the University. Egress filtering can be done on port 69, however this will impact TFTP. (http://www.cert.org/advisories/CA-2001-26.html) Neither of these filters can stop the propagation of the NIMDA totally since it propagates itself by many means. Firewalls that filter can block .eml extensions and help as well.
**Detect:**
spp_http_decode: IIS Unicode attack detected: Occurrence = 13,459

**Detect Description:**
This attack is carried out by passing Unicode to the IIS server in an attempt to gain access. There are also other false positives and can cause this alert. User’s normal outbound traffic as well as Netscape can produce false positives. ([http://www.snort.org/docs/faq.html#4.17](http://www.snort.org/docs/faq.html#4.17)) It is impossible to tell what all of these are with out knowing the network configuration and what is considered normal network traffic at the University to say what each of these are. We have Unicode alerts on both inbound and outbound traffic. Some of the traffic however appears to be Unicode scans and they are from internal hosts to external host:

```
07/06-00:38:58.458879 10.0.84.220:1923 -> 60.101.11.42:80
07/06-00:38:58.459721 10.0.84.220:1926 -> 129.236.112.105:80
07/06-00:38:58.461105 10.0.84.220:1928 -> 27.223.85.101:80
07/06-00:38:58.463307 10.0.84.220:1931 -> 188.124.219.11:80
07/06-00:38:58.464588 10.0.84.220:1930 -> 40.167.88.35:80
07/06-00:38:58.465887 10.0.84.220:1932 -> 147.122.173.183:80
07/06-00:38:58.468449 10.0.84.220:1934 -> 178.9.121.214:80
07/06-00:38:58.472294 10.0.84.220:1936 -> 93.21.24.206:80
07/06-00:38:58.473428 10.0.84.220:1935 -> 142.166.122.37:80
07/06-00:38:58.478384 10.0.84.220:1907 -> 97.76.188.70:80
```

**Correlations:**
- [http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884](http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884)
- [http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0709](http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0709)
- Other detects of the same kind can be found at:
  - [www.giac.org/practical/Jeff_Zahr_GCIA.doc](http://www.giac.org/practical/Jeff_Zahr_GCIA.doc)
  - [www.giac.org/practical/Matthew_Fiddler_GCIA.doc](http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc)
  - [www.sans.org/y2k/practical/Miika_Turkia_GCIA.html](http://www.sans.org/y2k/practical/Miika_Turkia_GCIA.html)

**Defensive Recommendations:**
GIAC University should immediately patch all systems running IIS and ensure they configured correctly and securely. It is important to learn what normal network traffic is. This will aid in determining if it is a Unicode attack/scan or if it is normal network traffic.

**Detect:**
External RPC call: Occurrence = 11,429

**Detect Description:**
This attack is carried out by looking for a listening RPC port. Typically this is port 111. Portmapper is a well known service running at port 111 for both TCP and UDP connections. “However, security personnel should know that under some versions of Unix, and Solaris rpcbind not only listens on the TCP/UDP port 111, but it also listens on UDP ports greater than 32770.”

Part 3: Analyze This
Page 53 of 73
If the attacker is able to find a listening RPC port, then they may be able to determine the services running on that machine or even gain root access to make calls to those services. GIAC University was being actively scanned for port 111. Here is a list of the IP addresses, some were resolved to show who the IP belonged to, but not all of them.

**Correlations:**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/07-06:15</td>
<td>08:834069</td>
<td>61.185.139.2:4767</td>
<td>10.0.253.17:111</td>
<td>111</td>
</tr>
<tr>
<td>07/07-08:09</td>
<td>48:513611</td>
<td>212.45.32.75:2407</td>
<td>10.0.1:2:111</td>
<td>111</td>
</tr>
<tr>
<td>07/07-11:30</td>
<td>09:190873</td>
<td>203.239.155.2:60117</td>
<td>10.0.159.29:111</td>
<td>111</td>
</tr>
<tr>
<td>07/07-17:00</td>
<td>53:044927</td>
<td>210.119.9.16:2790</td>
<td>10.0.28:3:111</td>
<td>111</td>
</tr>
<tr>
<td>07/07-17:15</td>
<td>17:058009</td>
<td>195.117.179.12:2879</td>
<td>10.0.111.21:111</td>
<td>111</td>
</tr>
<tr>
<td>07/08-12:01</td>
<td>15:373028</td>
<td>195.116.95.216:3320</td>
<td>10.0.28:8:111</td>
<td>111</td>
</tr>
<tr>
<td>07/08-15:15</td>
<td>15:59.770334</td>
<td>210.119.58.4:47226</td>
<td>10.0.28:3:111</td>
<td>111</td>
</tr>
<tr>
<td>07/08-21:00</td>
<td>00:638189</td>
<td>80.49.3:86:4783</td>
<td>10.0.10.174:111</td>
<td>111</td>
</tr>
<tr>
<td>07/09-07:33</td>
<td>47:244785</td>
<td>202.172.46.43:3516</td>
<td>10.0.15.178:111</td>
<td>111</td>
</tr>
<tr>
<td>07/09-08:30</td>
<td>22:397409</td>
<td>203.48.91.12:4190</td>
<td>10.0.28:13:111</td>
<td>111</td>
</tr>
<tr>
<td>07/10-09:16</td>
<td>21:442046</td>
<td>203.231.125.187:3556</td>
<td>10.0.5.95:111</td>
<td>111</td>
</tr>
<tr>
<td>07/10-17:15</td>
<td>15:819605</td>
<td>211.118.11.219:3072</td>
<td>10.0.80.69:111</td>
<td>111</td>
</tr>
</tbody>
</table>

Part 3: Analyze This
Page 54 of 73
Vulnerability Note VU#638099 http://www.kb.cert.org/vuls/id/638099
Other detects of the same kind can be found at: www.giac.org/practical/James_Conz_GCIA.doc www.giac.org/practical/dana_mclaughlin_gcia.doc

Defensive Recommendations:
GIAC University should immediately patch all systems running portmapper and RPCbind. If it is not being used, then port 111 should be blocked with the egress and ingress filters. Keep in mind; this will not eliminate the vulnerability in its entirety.

Detect:
Watchlist 000222 NET-NCFC: Occurrence = 9,210

Detect Description:
This watchlist is for the subnet 159.226.XXX.XXX and is registered to:
OrgName: The Computer Network Center Chinese Academy of Sciences
OrgID: CNCCAS
These were detected at GIAC University. Here is a list of the IP addresses and a list of possible activity.

- Port 4230 is registered in IANA to VRML Multi User Systems which are “systems which support distributed virtual worlds in which objects can be shared by different users”. (http://www.c-lab.de/vrml99/courses.html)

  07/06-00:29:41.411926 159.226.210.220:80 -> 10.0.84.220:4230

- This was an interesting one. Port 4160 TCP/UDP is registered to Jini Discovery. Sun describes it in this fashion: “Jini technology provides a flexible infrastructure for delivering services in a network and for creating spontaneous interactions between clients that use these services regardless of their hardware or software implementations.” (http://wwws.sun.com/software/jini/faqs/index.html#1)

  07/06-00:45:25.442036 159.226.119.3:80 -> 10.0.84.220:4160

- Port 3785 is unassigned and no other information was available about what activity may be occurring.

  07/06-00:50:37.310954 159.226.67.196:80 -> 10.0.198.199:3785

- Port 80 is http and as such it is difficult to know what was going on. These packets would require a closer look and monitoring. A connection from the outside to internal hosts on port 80 is not a good security practice.

  07/07-03:13:30.308645 159.226.100.51:3094 -> 10.0.252.23:80
  07/07-05:02:50.495320 159.226.49.157:19043 -> 10.0.111.140:80
  07/07-06:44:56.547291 159.226.47.236:1818 -> 10.0.198.199:80
  07/07-08:54:07.783066 159.226.221.122:4819 -> 10.0.146.97:80
  07/08-00:30:11.184703 159.226.4.142:1232 -> 10.0.111.140:80
  07/08-16:01:42.705235 159.226.110.142:2602 -> 10.0.158.2:80
  07/10-00:37:17.310790 159.226.39.251:64743 -> 10.0.111.140:80
  07/10-03:11:01.972951 159.226.165.70:3694 -> 10.0.111.140:80
Part 3: Analyze This

07/10-05:01:39.776616 159.226.92.118:1408 -> 10.0.145.18:80
07/10-20:47:19.391527 159.226.100.203:3526 -> 10.0.139.230:80
  - Port 25 is SMTP and is used for mail services. This could be spam or some other
    attack on an email system or a scan for an email system. It would require further
    investigation.

07/10-10:35:14.215434 159.226.64.138:1662 -> 10.0.6.40:25

**Correlations:**
- Other detects of the same kind can be found at:
  - www.giac.org/practical/Dennis_Davis_GCIA.doc
  - www.stearns.org/doc/william_stearns_gcia.html

**Defensive Recommendations:**
GIAC University should carefully monitor this traffic and determine if this is authorized
traffic for their network. If not, I would advise blocking it at the router and/or firewall for
all IP addresses from this range for the unsolicited traffic. Part of this could be legitimate
traffic. They should also ensure that all of their key network devices are patched to
defend against this.

**Detect:**
UDP SRC and DST outside network: Occurrence = 7,531

**Detect Description:**
This detect is concerned with the source and destination IP address both being external
sources. This should cause a concern that someone is spoofing that IP address or crafting
packets or a system is participating in some malicious activity. I do not believe these are
reconnaissance scans because they would not get a response back with a spoofed IP.
What ever is going on, they do not care to get an answer in return. All of these should be
followed up on to ensure malicious activity is not leaving the university network. Here
are some of the packets and activity that it is associated with:

- This activity appears to be NetBIOS which is port 137. They could be trying
different attacks on NetBIOS. There are several vulnerabilities.
  Destination IP resolves to:
    CustName: RIO MOTOR SPORTS, INC
    Address: 25 Broadway New York, NY 10004
    Country: US

- Port 53 is used for DNS (Domain Name Services). There are many known attacks
  against DNS. Ironically, this IP address is part of the private address space so I
  am unsure what would be gained from this.
  07/09-10:09:35.91263 169.254.236.55:137 -> 172.25.0.51:53

- At port 1900 resides SSDP (Simple services discover protocol) and this one seems
  pretty clear what is going on. The traffic is multicast and there is vulnerability in
  SSDP that takes advantage of the multicast traffic and can force a windows box
  into high CPU and memory utilization causing it to hang or forcing a reboot.

Part 3: Analyze This
Page 56 of 73
“The DDoS exploit uses this same vulnerability, taking advantage of the broadcast and multicast nature of SSDP to direct an attack from multiple “devices” against a single victim or against a range of victims.”

VPJP (Virtual Place Java Port). I am unable to determine the exact nature of this port, but the name tends to describe the function it is used for.

Here is our multicast destination IP again. There is vulnerability for Novell clients with this particular port and SLP. Apparently when scanning a network with NMAP using a half open scan across port 427, it will instantly blue screen.

Correlations:
- DoS vulnerability in Novell Intranetware Client 3.0.0.0: [http://packetstormsecurity.nl/9901-exploits/novell-iwc-DoS.txt](http://packetstormsecurity.nl/9901-exploits/novell-iwc-DoS.txt)
- Vulnerability Note VU#411059 (SSDP): [http://www.kb.cert.org/vuls/id/411059](http://www.kb.cert.org/vuls/id/411059)
- CERT® Incident Note IN-2001-03 (port 53, DNS): [http://www.cert.org/incident_notes/IN-2001-03.html](http://www.cert.org/incident_notes/IN-2001-03.html)
- Other detects of the same kind can be found at: [www.giac.org/practical/Rick_Yuen_GCIA.doc](http://www.giac.org/practical/Rick_Yuen_GCIA.doc) [www.giac.org/practical/Dennis_Davis_GCIA.doc](http://www.giac.org/practical/Dennis_Davis_GCIA.doc)

Defensive Recommendations:
GIAC University should install egress filters on their border router that drops all traffic not originating from the internal network with a source IP address of the internal network. Do not allow NetBIOS traffic to leave the internal network. If you are running your own DNS servers, you can ensure users use these and allow only the DNS server to traffic to leave the network. If not, you are just going to have to monitor the network. Ensure all of your network devices drop private address spaces and do not route them.

Detect:
TFTP - External UDP connection to internal tftp server: Occurrence = 7,135

Detect Description:
This detect is concerned with the source IP address being external sources. The TFTP traffic is coming from an IP address of the University and destined for an IP address that is a private address space. Also, it always to same four IP addresses listed below and to numerous ports at the destination IP address. It should be looked at for what is occurring here. Here are some of the packets and activity that it is associated with:
Correlations:
- CERT® Advisory CA-1991-18 Active Internet tftp Attacks:
- Alcatel ADSL modems grant unauthenticated TFTP access via Bounce Attacks:
  [Link](http://www.kb.cert.org/vuls/id/211736)
- Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability:
  [Link](http://216.239.35.100/search?q=cache:dWh3qkHmLhMC:online.securityfocus.com/bid/1806/exploit/%22TFTP%22+vulnerabilities&hl=en&ie=UTF-8)
- Other detects of the same kind can be found at:
  [Link](www.giac.org/practical/Matthew_Fiddler_GCIA.doc)
  [Link](www.giac.org/practical/Mike_Poor_GCIA.doc)
  [Link](www.giac.org/practical/Karim_Merabet_GCIA.doc)

Defensive Recommendations:
GIAC University should carefully monitor the TFTP traffic and block it if necessary. It would be key to identify where the traffic is coming from within the University. There is an awful lot of TFTP traffic entering and leaving the University. They should also ensure that all of their primary systems such as switches, modems etc are protected by the recommendations of the vendor. Many more vulnerabilities can be found by doing a search on [www.google.com](http://www.google.com)

Detect:
High port 65535 udp - possible Red Worm - traffic: Occurrence = 4,334

Detect Description:
This detect is concerned with the source IP address being external sources. The TFTP traffic is coming from an IP address of the University and destined for an IP address that is a private address space. Also, it always to same four IP addresses listed below and to numerous ports at the destination IP address. It should be looked at for what is occurring here. Here are some of the packets and activity that it is associated with:

<table>
<thead>
<tr>
<th>Time</th>
<th>Source IP</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/10-23:31:33.218013</td>
<td>10.0.111.231:69</td>
<td>192.168.0.216:3320</td>
</tr>
</tbody>
</table>

Correlations:
- Alcatel ADSL modems grant unauthenticated TFTP access via Bounce Attacks:
  [Link](http://www.kb.cert.org/vuls/id/211736)
- Other detects of the same kind can be found at:
  [Link](www.giac.org/practical/Tyler_Schacht_GCIA.doc)
  [Link](http://www.giac.org/practical/Christof_Voemel_GCIA.txt)
  [Link](www.giac.org/practical/Karim_Merabet_GCIA.doc)
Defensive Recommendations:
GIAC University should carefully monitor the TFTP traffic and block it if necessary. It would be key to identify where the traffic is coming from within the University. There is an awful lot of TFTP traffic entering and leaving the University. They should also ensure that all of their primary systems such as switches, modems etc are protected by the recommendations of the vendor. Many more vulnerabilities can be found by doing a search on www.google.com

Detect:
spp_http_decode: CGI Null Byte attack detected: Occurrence = 3,152

Detect Description:
This detect is alerted by the SNORT rule looking for a null value in the http traffic. There are many cases of false positives that occur with this, however Martin Roesch created a patch for the http_decode processor to ensure what was being handed to it was valid and thus to eliminate some of the false positives. (http://archives.neohapsis.com/archives/snort/2001-03/0425.html) This traffic, for the alert, needs to be examined for null values and see if they are false positives or actual alerts. The ones listed appear to be normal web traffic resulting in a false positive.

OrgName: The Cobalt Group, Inc
OrgID: THECOB

07/10-14:46:44.164834 10.0.137.35:4478 -> 199.104.95.15:80
OrgName: Deseret News
OrgID: DESERE-1

07/10-15:15:39.330715 10.0.163.125:1460 -> 128.167.120.48:80
OrgName: Genuity
OrgID: GNTY

Correlations:
* http://cert.uni-stuttgart.de/archive/incidents/2001/12/msg00006.html
* Other detects of the same kind can be found at:
  www.giac.org/practical/Mike_Poor_GCIA.doc
  www.giac.org/practical/Karim_Merabet_GCIA.doc

Defensive Recommendations:
GIAC University should carefully monitor all web traffic closely for malicious activity. In this particular circumstance, they need to make sure SNORT has the latest patches to help eliminate the false positives.

Detect:
SYN-FIN scan!: Occurrence = 2,636

Detect Description:
This detect is alerted by having both the SYN and FIN flags set on TCP connections. The purpose is to get the packets passed some firewalls and IDSs. Most of these are done as part of OS fingerprinting and are very loud and noisy for most systems today. There are many different type of scanners that produce this combination. Here are the sources using SYN/FIN scans that are looking for port 21 which is an FTP port.

netname:      BERGKEMPER-NET
descr:      Ursula Bergkemper EDV-Engineering
country:      DE

07/07-02:17:55.221415 166.104.219.69:21 -> 10.0.88.114:21
OrgName:    Hanyang University
OrgID:      HANYAN

07/08-03:32:02.739547 211.171.149.164:21 -> 10.0.1.203:21
inetnum:      211.168.0.0 - 211.171.255.255
netname:      KRNIC-KR
descr:      KRNIC
descr:      Korea Network Information Center
country:      KR

Correlations:
muş Symantec Norton Personal Firewall 2002 SYN/FIN scan issue:
http://securityresponse.symantec.com/avcenter/security/Content/2002.05.16.html
muş http://www.sans.org/PH2000/snort/SnortAle.txt
muş Other detects of the same kind can be found at:
www.giac.org/practical/Alex_Stephens_GCIA.htm
www.giac.org/practical/chris_kuethe_gcia.html

Defensive Recommendations:
GIAC University needs to ensure that all systems are patched and to test their key network devices such as routers, firewalls and IDSs to see if they are allowing them to pass through. Symantec says “Although a Microsoft Windows 2000 computer can be detected through the SYN/FIN scan, Symantec Norton Personal Firewall 2002 continues to protect the computer from an actual intrusion by blocking connections to the computer.” (http://securityresponse.symantec.com/avcenter/security/Content/2002.05.16.html) They did however come up with a patch. It is critical to test your security devices and know what is getting through!

Detect:
AFS - Off-campus activity: Occurrence = 2,096

Detect Description:
This detect appears to be looking for an AFS vulnerability. “By scanning port 7001 and sending malicious packets the attacker was able to crash AFS servers. Reports have shown that at least Solaris 5.6 and 5.7 machines and AIX 4.3.3 machines are affected.” (https://lists.openafs.org/pipermail/openafs-info/2002-June/004784.html) If they are not attacking it, then it would be important to find out if someone has setup file sharing with
AFS on these boxes. There is also vulnerability on this port for BEA Weblogic’s Proxy, however, I would not expect to see this many proxies on the University. Also, this not from Port 80, but port 7000 which is part of AFS. These are the IP addresses specifically targeted by all IP addresses in question. It is important to note they did not scan for these, but all went directly to them. They were known targets! Here is a partial list of IP addresses hitting these machines.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Service</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/07-00:42:58</td>
<td>58.963654</td>
<td>63.250.205.49:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>07/07-12:18:42</td>
<td>52.2629</td>
<td>63.250.205.17:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>07/07-13:19:15</td>
<td>323416</td>
<td>63.250.219.185:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>07/07-14:19:04</td>
<td>790704</td>
<td>63.250.219.187:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>07/07-14:33:17</td>
<td>724876</td>
<td>63.250.205.39:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>07/07-18:39:38</td>
<td>063368</td>
<td>63.250.205.42:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>07/07-19:52:41</td>
<td>828299</td>
<td>63.250.205.35:7000</td>
<td>TCP</td>
<td>7000</td>
<td></td>
</tr>
</tbody>
</table>

OrgName: Yahoo! Broadcast Services, Inc.
OrgID: YAHO

Correlations:
- Vulnerability Report for BEA Weblogic’s Proxy: http://security-archive.merton.ox.ac.uk/bugtraq-200008/0241.html
- Other detects of the same kind can be found at: www.sans.org/y2k/practical/David_Singer_GCIA.doc

Defensive Recommendations:
GIAC University needs to investigate this further, especially with the locations of the visitors from overseas countries. These IP address were known in advance and were

Part 3: Analyze This
Page 61 of 73
specifically targeted. All target systems should be identified and checked for possible compromise. Appropriate steps should be taken to ensure they are patched properly and secured. If possible, block access from external sources on port 7001.

**Detect:**

beetle.ucs: Occurrence = 1,577

**Detect Description:**

This detect appears to be watching two machines at the University with a CD-R. ([http://www.gl.umbc.edu/root/common.shtml](http://www.gl.umbc.edu/root/common.shtml)) The following seems to be normal traffic to and from these two machines:

```
07/06-00:20:00.827483 10.0.70.69:841 -> 10.0.60.11:782
07/06-00:20:00.827837 10.0.60.11:782 -> 10.0.70.69:841
```

However, they have visitors as evident from below. The concern would be the external IP addresses were looking for data that had been stored for burning, but not yet taken off of the system. Also, malicious code could be placed on the machine and accidentally burned onto an unsuspecting individual’s CDs. We see multiple ports from port 1433:

Microsoft SQL, Port 80: HTTP, Port 21: FTP etc.

```
07/07-08:00:48.037443 206.168.112.119:4945 -> 10.0.70.69:1433
07/07-08:00:48.037617 10.0.70.69:1433 -> 206.168.112.119:4945
```

```
OrgName: NeTrack
OrgID:   NTRK
Address: PO BOX 17700 Boulder, CO 80308-0700
Country: US
```

```
07/07-09:03:51.466922 140.131.114.155:4953 -> 10.0.70.69:80
07/07-09:03:51.467204 10.0.70.69:80 -> 140.131.114.155:4953
```

```
OrgName: Ministry of Education Computer Center
OrgID:   MOEC
Address: 12th Floor No. 106
         Section 2, Ho-Ping East Road
         Taipei, Taiwan, ROC ,
Country: TW
```

```
07/07-18:27:45.562518 68.39.7.45:22 -> 10.0.70.69:22
07/07-18:27:45.566512 10.0.70.69:22 -> 68.39.7.45:22
```

```
OrgName: Comcast Cable Communications, Inc.
OrgID:   CMCS
Address: 3 Executive Campus
         5th Floor Cherry Hill, NJ 08002
```

```
07/08-04:05:59.617258 80.140.10.148:4038 -> 10.0.70.69:21
07/08-04:05:59.617503 10.0.70.69:21 -> 80.140.10.148:4038
```

```
07/09-23:00:32.960886 62.253.226.1:3749 -> 10.0.70.69:80
07/09-23:00:32.961129 10.0.70.69:80 -> 62.253.226.1:3749
```

**Correlations:**

- [http://www.gl.umbc.edu/root/common.shtml](http://www.gl.umbc.edu/root/common.shtml)
- Other detects of the same kind can be found at: www.giac.org/practical/Jeff_Zahr_GCIA.doc
  www.giac.org/practical/Edward_Peck_GCIA.doc

Part 3: Analyze This

Page 62 of 73
Defensive Recommendations:
GIAC University needs to investigate this further, especially with the locations of the visitors from overseas countries. These IP addresses were known in advance and were specifically targeted. All target systems should be identified and checked for possible compromise. Appropriate steps should be taken to ensure they are patched properly and secured. If possible, block access from external sources on port 7001.

"Top Talkers for OOS and Scan logs"
In order to look at the OOS and the Scan logs in a logical manner with the other logs, the top 5 talkers from each were extracted and then searched for against the other logs to determine possible activity of that particular IP address. This will not take into account if the IP address was spoofed as there is no way to determine this from a big perspective.

OOS Logs Top Five Talkers
Here are the top five talkers for the OOS logs:

<table>
<thead>
<tr>
<th>Source IP address</th>
<th>Number of Times Appearing</th>
</tr>
</thead>
<tbody>
<tr>
<td>68.32.126.64</td>
<td>230</td>
</tr>
<tr>
<td>209.116.70.75</td>
<td>92</td>
</tr>
<tr>
<td>65.210.154.210</td>
<td>37</td>
</tr>
<tr>
<td>211.110.13.28</td>
<td>19</td>
</tr>
<tr>
<td>141.161.105.226</td>
<td>17</td>
</tr>
</tbody>
</table>

Here is a look at the flags for each of the top IP addresses and their destination IP:

<table>
<thead>
<tr>
<th>Source IP Address</th>
<th>Destination IP Address</th>
<th>Destination Port</th>
<th>Flags Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>68.32.126.64</td>
<td>10.0.6.7</td>
<td>110</td>
<td>21S*****</td>
</tr>
<tr>
<td>209.116.70.75</td>
<td>Multiple IPs</td>
<td>25</td>
<td>21S*****</td>
</tr>
<tr>
<td>211.110.13.28</td>
<td>Multiple IPs</td>
<td>21</td>
<td><strong>SF</strong>**</td>
</tr>
<tr>
<td>141.161.105.226</td>
<td>10.0.253.114</td>
<td>80</td>
<td>21S*****</td>
</tr>
</tbody>
</table>

We will look at each IP address and see what activity they appeared to be looking for and if there is any correlations with the other logs.

1. **68.32.126.64**: This was the top talker in the OOS logs and below is an example of the traffic:

   07/10-13:52:58.741859 68.32.126.64:13369 -> 10.0.6.7:110
   TCP TTL:47 TOS:0x0 ID:10510 DF
   21S***** Seq: 0xC5B9F5DC Ack: 0x0 Win: 0x16D0
   TCP Options => MSS: 1460 SackOK TS: 39574738 0 EOL EOL EOL EOL

   The IP address was not found in the other log files. They are hitting one IP address on Port 110 which is POP3 for mail. This scan started just shortly before 1400 and ended at 0004. The packets were sent anywhere from 1 to 15 minutes apart. This does not appear to be a SYN scan as they are only hitting one target; however this can be used as a DoS against another box. It is possible the attacker has identified the box as listening and is...
using it in an attack against another system. This would be determined by looking at network traffic to see the whole conversation and what is taking place. This is a smart port to use, as it is the mail port and denying access to it would not be feasible. As such, it is critical that the email server be locked down tight and patched accordingly.

2. **209.116.70.75**: This IP address is sending SYN packets to port 25, which is SMTP, on multiple destinations IPs. Here is a look at one of the packets.

07/10-13:53:40.732900 209.116.70.75:55580 -> 10.0.100.217:25
TCP TTL:51 TOS:0x0 ID:1257 DF
21S**** Seq: 0xD4120012 Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 757794043 0 EOL EOL EOL EOL

This IP address belonged to a busy individual and was found in the OOS, Scan and Alert Logs. Here is some of the activity found:

**Alert Logs**

323372: **07/07-13:42:12.058561** [**] Queso fingerprint [**] 209.116.70.75:55136 -> 10.0.100.217:25 15609: **07/08-01:37:53.731651** [**] Queso fingerprint [**] 209.116.70.75:59672 -> 10.0.100.217:25
00956: **07/09-00:00:58.760429** [**] Queso fingerprint [**] 209.116.70.75:52267 -> 10.0.100.217:25
48160: **07/10-03:16:56.301686** [**] Queso fingerprint [**] 209.116.70.75:36265 -> 10.0.100.217:25

**Scan Logs**


It is interesting to note that 10.0.6.40 keeps responding with SYN packets to 209.116.70.75 on port 113. This could be because 10.0.6.40 is running an IDENT Service or Auth service which listens on this port. If the attacker probes that box on FTP, HTTP, SMTP etc, the IDENT services attempts to connect back to the target for some information. ([http://www-h.eng.cam.ac.uk/help/jpmg/CUED_Probed_Me.html](http://www-h.eng.cam.ac.uk/help/jpmg/CUED_Probed_Me.html)) However, Invisible IDENT Deamon and Kazimas are two trojans who also listen on this port. It is important check this system for signs of possible compromise.

Apparently this individual doing the scanning was not concerned with noise, or was scanning from a spoofed IP and sniffing in the middle. As such, noise would not be an issue to them. The IP in question is from:

**OrgName**: Inflow  
**OrgID**: NFLO  
**Address**: 1860 Lincoln Street, Suite 305 Denver, CO 80295  
**Country**: US

This traffic needs to be examined further and checked for other possible signs of compromise.

3. **65.210.154.210**: This IP was only found in the OOS and appears to be the part of a file sharing group. Port 4662 is associated with Edonkey and can be found at [http://www22.brinkster.com/edonkeyhq/faq.htm](http://www22.brinkster.com/edonkeyhq/faq.htm). The IP in question is from:

**OrgName**: Massachusetts Institute of Technology  
**OrgID**: MIT-2

Part 3: Analyze This  
Page 64 of 73
It appears that some files sharing is going on between GIAC University and MIT. If this is acceptable for GIAC University security policy, just monitor for usage to make sure nothing changes. If not, put an ingress filter on that blocks incoming port 4662 connections.

4. **211.110.13.28**: This IP was the only deviant from the SYN packets we were seeing. This attacker chose to use SYN/FIN scans. There were no other correlations between this IP address and the other logs. Using a SYN/FIN combination allows some packets to slide past firewalls and IDSs. See the above discussion on SYN/FIN scans in the Alert Analysis. This attacker is scanning for a listening FTP server on port 21. Here is where this IP originates from:

   **inetnum:** 211.104.0.0 - 211.119.255.255
   **netname:** KRNIC-KR
   **descr:** Korea Network Information Center
   **country:** KR

5. **141.161.105.226**: This IP address was interested in port 80 and only one destination IP. It is difficult to determine what activity they were up to. There were no correlations for this IP address and the other Alert and Scan logs. You would need to look at the logs of the destination IP and see if they logged any malicious activity. Here is a look at one of the packets:

   TCP TTL:59 TOS:0x0 ID:8557 DF
   21S***** Seq: 0x157A96C9   Ack: 0x0   Win: 0x16D0
   TCP Options => MSS: 1460 SackOK TS: 9615268 0 EOL EOL EOL EOL

   It is not something you can block (life can function without the Internet on a University Campus), but it is important to closely watch all traffic on port 80. Ensure the SNORT rules are kept up to date. If this is not a web server and this traffic is not to originate from external IP addresses, then put an ingress filter on to block traffic from outside to port 80 except for authorized web servers. This IP address is from:

   **OrgName:** Georgetown University
   **OrgID:** GEORGE-8

**Scan Logs Top Five Talkers**

Here are the top five talkers for the Scan logs. Notice that two of them are from the internal network.

<table>
<thead>
<tr>
<th>Source IP address</th>
<th>Number of Times Appearing</th>
</tr>
</thead>
<tbody>
<tr>
<td>211.171.149.164</td>
<td>2628</td>
</tr>
<tr>
<td>10.0.70.183</td>
<td>207</td>
</tr>
<tr>
<td>10.0.186.16</td>
<td>155</td>
</tr>
<tr>
<td>207.69.221.121</td>
<td>15</td>
</tr>
<tr>
<td>200.221.179.255</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scan Type</th>
<th>Number of packets passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNFIN</td>
<td>2639</td>
</tr>
<tr>
<td>NULL</td>
<td>391</td>
</tr>
</tbody>
</table>

Part 3: Analyze This
Page 65 of 73
### Part 3: Analyze This

<table>
<thead>
<tr>
<th>Source IP Address</th>
<th>Destination IP Address</th>
<th>Destination Port</th>
<th>Scan Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>211.171.149.164</td>
<td>Multiple IPs</td>
<td>21</td>
<td>SYN/FIN</td>
</tr>
<tr>
<td>10.0.70.183</td>
<td>Multiple IPs</td>
<td>Multiple Ports</td>
<td>Null</td>
</tr>
<tr>
<td>10.0.186.16</td>
<td>Multiple IPs</td>
<td>Multiple Ports</td>
<td>Null</td>
</tr>
<tr>
<td>207.69.221.121</td>
<td>Multiple IPs</td>
<td>21</td>
<td><strong>SF</strong>**</td>
</tr>
<tr>
<td>200.221.179.255</td>
<td>10.0.253.114</td>
<td>80</td>
<td>21S*****</td>
</tr>
</tbody>
</table>

1. **211.171.149.164**: This IP ran lots of noisy SYN/FIN scans against port 21. They hit all ranges of the MY.NET Subnet looking for a listening FTP server. This IP address showed up in both the Alert and Scan logs for 8 Jul 02. For more information on the SYN/FIN scan see the Alert section.

   07/08-03:47:37.917714  [**] SYN-FIN scan! [**] 211.171.149.164:21 -> 10.0.185.48:21

2. **10.0.70.183**: This is an internal IP address and was very busy. It appeared in the following logs: alert.020707, alert.020708, alert.020709, alert.020710, scans.020706, scans.020707, scans.020708, scans.020709, and scans.020710. It was always hitting 10.0.1.4 on port 37 (which is another internal box and the time service) and 150.254.64.64 and on multiple ports. Here are what some of the packets look like:

   07/07-08:03:14.064226  [**] Null scan! [**] 10.0.70.183:53974 -> 10.0.1.4:37
   Jul 9 22:01:01 10.0.70.183:48121 -> 150.254.64.64:5825 UDP
   Jul 9 22:01:02 10.0.70.183:33252 -> 150.254.64.64:6324 UDP
   Jul 9 22:01:02 10.0.70.183:12037 -> 150.254.64.64:6920 UDP
   Jul 9 22:01:03 10.0.70.183:32304 -> 150.254.64.64:7673 UDP

In addition to this on port 37 resides the time service. “Linux Time Bomb - The inetd running the TCP time services, daytime (port 13) and time (port 37) will crash if you send excessive SYN packets. Once inetd crashes, all other services running through inetd no longer will work.” ([http://www.attrition.org/security/denial/w/den-list.dos.html](http://www.attrition.org/security/denial/w/den-list.dos.html)) It could also be that this is legitimate network traffic from looking at the time stamps. GIAC University needs to determine if this is indeed a time server. This also identified as a null scan, meaning there were no flags set. In this case, a listening port will always reply with a reset. (RFC 793)

As for the UDP traffic to 150.254.64.64, this could be anything. It always hits the same IP address and on random ports. It could be checking which ports are responding? The address alone leaves one to wonder what is going on. Here is the address information for the destination IP address:

**inetnum**: 150.254.64.0 - 150.254.64.255

---

**Part 3: Analyze This**

Page 66 of 73
GIAC University needs to look farther into this one and see what is going on.

3. **10.0.186.16**: This internal IP address is hitting multiple internal hosts on many different ports. It is recorded as a null scan, meaning that no flags were set and open ports should respond with a reset as we stated above. All of these scans were done from port 23 on the source host. Port 23 is where you would find telnet. It appears that someone compromised this box and logged onto it through a telnet session. From there, they proceeded to scan the internal network. If the box wasn’t compromised, then the owner of the box needs to be found and determine what was being scanned for. Here is an example of part of the scan:

```
07/08-07:50:04.704616  [**] Null scan! [**] 10.0.186.16:23 -> 10.0.177.55:1260
```

This IP address was found in the following logs: alert.020708, alert.020709, alert.020710, scans.020708, scans.020709 and scans.020710.

4. **207.69.221.121**: This IP address certainly was creative. The first time it appeared was in alert.020709 log and it appeared to be a null scan from port 0 to port 0. Here is one of the packets:

```
07/09-19:46:45.625135  [**] Null scan! [**] 207.69.221.121:0 -> 10.0.115.236:0
```

However, this was not all that was going on. In the scans.020709 log, we find the following recorded:

```
Jul  9 19:46:46 207.69.221.121:0 -> 10.0.115.236:0 NULL ********
Jul  9 19:46:50 207.69.221.121:1929 -> 10.0.115.236:3796 INVALIDACK 12UAP*SF RESERVEDBITS
Jul  9 19:47:04 207.69.221.121:53545 -> 10.0.115.236:3487 UNKNOWN 12*A**** RESERVEDBITS
Jul  9 19:47:04 207.69.221.121:18244 -> 10.0.115.236:14433 NOACK *2**PRS* RESERVEDBITS
Jul  9 19:47:04 207.69.221.121:0 -> 10.0.115.236:0 NULL ********
```

Here is the IP information on 207.69.221.121:

```
OrgName: EarthLink, Inc.
OrgID: ERMS
Address: 3100 New York Drive Pasadena, CA 91107
```

The destination IP was always 10.0.115.236. This attacker was certainly interested in this host and tried several combinations of flags against it. Also, notice the null scans are from port 0 to port 0. I am unsure what tool caused the scan. I cannot find one that duplicates this pattern. GIAC University should look at the destination IP and ensure it has not been compromised and what for future traffic.

5. **200.221.179.255**: This IP address was alternating between two different hosts. There was one SYN packet and the rest had the Push flag set. It was always to port 1214 and the source port was 1988 for 10.0.150.133 and port 1938 for 10.0.150.220. The following is a look at the traffic:
Port 1214 is used for Kazaa Lite and may be nothing more looking for someone running Kazaa. However, it would be interesting to know what was in the payload on the packets with the Push flag set. The IP address is registered to:

- **Owner**: Comite Gestor da Internet no Brasil
- **Owner ID**: BR-CGIN-LACNIC
- **Responsible**: Frederico A C Neves
- **Address**: Av. das Nações Unidas, 11541, 7° andar
- **City**: São Paulo
- **Country**: BR

GIAC University needs to look further at this one. If Kazaa is not allowed by their security policy, block it with an ingress filter.

### External Source Addresses

The IP addresses chosen to lookup for those that participated in what could be Trojan activity or one that appears to have a system compromised. Here are the IP addresses with the top talker of each alert being looked up.

1. 211.161.112.18: SubSeven

#### Query the APNIC Whois Database

```
% [whois.apnic.net node-1]
% How to use this server    http://www.apnic.net/db/
% Whois data copyright terms http://www.apnic.net/db/dbcopyright.html

inetnum: 211.152.0.0 - 211.163.255.255
netname: CNNIC
descr: No.4, Zhongguancun No.4 South Street,
descr: Haidian District, Beijing
descr: P.O.Box: No.6 Branch-box of No.349 Mailbox, Beijing
country: CN
admin-c: MW1-AP
techn-c: IPAS1-AP
mnt-by: APNIC-HM
mnt-lower: MAINT-CNNIC-AP
changed: hostmaster@apnic.net 20000627
status: ALLOCATED PORTABLE
source: APNIC
role: CNNIC IPAS CONFEDERATION
descr: No.4, Zhongguancun No.4 South Street, Haidian District, Beijing
country: CN
phone: +86-10-62553604
fax-no: +86-10-62559892
e-mail: ipas@cnnic.net.cn
admin-c: LW152-AP
techn-c: LY220-AP
nic-hdl: IPAS1-AP
mnt-by: MAINT-CNNIC-AP
changed: ipas@cnnic.net.cn 20020910
```
source: APNIC
person: Mao Wei
address: China Internet Information Center (CNNIC) No. 4 of South street, Zhongguancun, Haidian District
address: Beijing, 100080
address: P.R.China
country: CN
phone: +86-10-62619750
fax-no: +86-10-62559892
e-mail: mao@cnnic.net.cn
nic-hdl: MW1-AP
mnt-by: MAINT-CNNIC-AP
changed: IPAS@CNNIC.NET.CN 20010319

2. 67.201.32.129: Backdoor NetMetro:

Output from ARIN Whois
OrgName: UUNET Technologies, Inc.
OrgID: UUDA

NetRange: 67.192.0.0 - 67.255.255.255
CIDR: 67.192.0.0/10
NetName: UUNET01DU
NetHandle: NET-67-192-0-0-1
Parent: NET-67-0-0-0-0
NetType: Direct Allocation
NameServer: DIALDNS1.UU.NET
NameServer: DIALDNS2.UU.NET
Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate: 2001-09-13
Updated: 2002-03-25

TechHandle: OA12-ARIN
TechName: UUNet, Technologies
TechPhone: +1-800-900-0241
TechEmail: help@uu.net

OrgAbuseHandle: ABUSE3-ARIN
OrgAbuseName: abuse
OrgAbusePhone: +1-800-900-0241
OrgAbuseEmail: abuse-mail@wcom.com

OrgNOCHandle: NAG-ARIN
OrgNOCName: GridNet International, Net
OrgNOCPhone: +1-800-998-5520
OrgNOCEmail: netadmin@ao.wcom.net

OrgTechHandle: NAG-ARIN
OrgTechName: GridNet International, Net
OrgTechPhone: +1-800-998-5520
OrgTechEmail: netadmin@ao.wcom.net

# ARIN Whois database, last updated 2002-09-18 19:05
# Enter ? for additional hints on searching ARIN's Whois database.
3. 207.38.1.201: EXPLOIT x86 stealth noop

Output from ARIN Whois

Search results for: ! NET-207-38-0-0-2

OrgName: GameSpy Industries
OrgID: GAMESP-3
NetRange: 207.38.0.0 - 207.38.1.255
CIDR: 207.38.0.0/23
NetName: ICI-GAMESPY-1
NetHandle: NET-207-38-0-0-2
Parent: NET-207-38-0-0-1
NetType: Reassigned
NameServer: NS3.INTELENET.NET
NameServer: NS4.INTELENET.NET
NameServer: NS.GAMESPY.COM
NameServer: NS2.GAMESPY.COM
NameServer: NS3.GAMESPY.COM
Comment:
RegDate: 2002-04-11
Updated: 2002-04-11

TechHandle: SB1687-ARIN
TechName: Berrigan, Stephen
TechPhone: +1-949-798-4200
TechEmail: admin@gamespy.com

# ARIN Whois database, last updated 2002-09-18 19:05
# Enter ? for additional hints on searching ARIN's Whois database.

4. 195.130.152.11: FTP DoS FTPd globbing

Query the Ripe Whois Database

inetnum: 195.130.150.0 - 195.130.159.255
netname: TELENET
descr: Telenet Operaties N.V.
country: BE
admin-c: PS396-RIPE
tech-c: PS396-RIPE
status: ASSIGNED PA
mnt-by: TELENET-DBM
mnt-lower: TELENET-DBM
changed: tech@telenet-ops.be 20010315
source: RIPE
route: 195.130.128.0/19
descr: TELENET
origin: AS6848
mnt-by: TELENET-OPS-MNT
changed: tech@telenet-ops.be 20010523
source: RIPE
role: Technical Internet
address: Telenet Operaties N.V.
address: Liersesteenweg 4
address: B-2800 Mechelen
address: Belgium
e-mail: tech@telenet-ops.be
trouble: IMPORTANT: To report intrusion attempts, hacking,
trouble: IMPORTANT: spamming, or other unaccepted behavior
trouble: IMPORTANT: by a Telenet/Pandora customer, please
trouble: IMPORTANT: send a message to abuse@pandora.be
trouble: IMPORTANT: Voor het rapporteren van inbraakpogingen,
trouble: IMPORTANT: hacking, spamming, of ander onaanvaardbaar
trouble: IMPORTANT: gedrag van een Telenet/Pandora klant,
gelieve
trouble: IMPORTANT: een bericht te zenden naar abuse@pandora.be
admin-c: TI346-ORG
tech-c: TI346-ORG
nic-hdl: PS396-RIPE
mnt-by: TELENET-DBM
changed: tech@telenet-ops.be 20000630
source: RIPE

- **Bold**: Object type.
- **Underlined**: Primary key(s).
- **Hyperlinks**: Searchable Attributes.

3 records found for '195.130.152.11'

5. 202.166.2.62: SMB C access
**Query the APNIC Whois Database**

% [whois.apnic.net node-2]  
% How to use this server  
% Whois data copyright terms  
http://www.apnic.net/db/dbcopyright.html

**inetnum:** 202.166.0.0 - 202.166.31.255  
**netname:** MAGIX  
**descr:** Magix Broadband Network  
**descr:** Singapore Telecommunications LTD  
**country:** SG  
**admin-c:** MH213-AP  
**tech-c:** MH213-AP  
**mnt-by:** APNIC-HM  
**mnt-lower:** MAINT-SG-MAGIX  
**changed:** hostmaster@apnic.net 19981103  
**changed:** hostmaster@apnic.net 20010117  
**changed:** hostmaster@apnic.net 20011029  
**status:** ALLOCATED PORTABLE  
**source:** APNIC  
**person:** Magix Hostmaster  
**address:** Singapore Telecommunications Ltd.  
**address:** 10 Eunos Road 8  
**address:** Singapore Post Centre  
**address:** #13-03  
**address:** Singapore, 408600  
**country:** SG  
**phone:** +65-6-848-4052  
**fax-no:** +65-6-848-4052  
**e-mail:** hostmaster@magix.com.sg  
**nic-hdl:** MH213-AP  
**remarks:** Spam and Security Issues: abuse@magix.com.sg  
**remarks:** Network Issues: noc@magix.com.sg  
**notify:** hostmaster@magix.com.sg  
**mnt-by:** MAINT-SG-MAGIX  
**changed:** raymondh@singtel.com 20011111  
**source:** APNIC

- **Bold:** Object type.  
- **Underlined:** Primary key(s).  
- **Hyperlinks:** Searchable Attributes.

2 records found for '202.166.2.62'

**Machines to investigate further**

There are several machines which are in need of further investigation. These machines are listed below by IP address and why they should be looked at for possible worm, Trojan, or suspicious activity. It is important that these machines are looked at immediately and steps taken to fix any issues that may exist.

1. **Back Orifice:** A backdoor Trojan giving access to your system. These IP addresses participated in both sides of a conversation:
2. Possible Trojan server activity: Once again we have possible Trojan activity and these IP addresses were actively communicating outside of our network:

<table>
<thead>
<tr>
<th>10.0.253.124</th>
<th>10.0.6.50</th>
<th>10.0.6.52</th>
<th>10.0.6.53</th>
<th>10.0.6.62</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.60.10</td>
<td>10.0.99.120</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Nimda: Nimda is a worm that was discussed in the alerts analysis above. Appropriate steps need to be taken to ensure the system is cleaned correctly. Here are the IP addresses:

<table>
<thead>
<tr>
<th>10.0.111.140</th>
<th>10.0.111.21</th>
<th>10.0.111.41</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.6.40</td>
<td>10.0.70.231</td>
<td>10.0.158.24</td>
</tr>
</tbody>
</table>

4. Possible Red Worm traffic: Red worm traffic is detected from the following IP addresses. They need to be followed up on and cleaned if necessary.

<table>
<thead>
<tr>
<th>10.0.8.8</th>
<th>10.0.5.74</th>
<th>10.0.6.40</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.1.15</td>
<td>10.0.85.97</td>
<td></td>
</tr>
</tbody>
</table>

5. Suspicious Traffic: This traffic triggered and alert and the IP addresses below need to be checked for possible system compromise.

<table>
<thead>
<tr>
<th>10.0.162.90</th>
<th>10.0.158.53</th>
</tr>
</thead>
</table>

6. Possible My Party infection: This is a virus that needs to be cleaned from the infected system:

| 130.85.109.47 |

**Link Graph**

The following is a link graph looking at the OOS logs. The logs were totaled and then graphed on the Y axis by number of hits and the X axis by port and flag setting. This was a little difficult so I had to turn off the values on the Y axis or you could not see the data due to the outliers. The graphed value has a letter which corresponds to the flags used and a number which corresponds to the port. By this we are able to see several things:

1. The destination ports our attackers were interested in.
2. An idea of which were the most popular destination ports.
3. The different flag combinations in use and which ones were most used.
OOS Flags

Flag Legend

| A=**SF**** | B=21S***** | C=*1SF***** | D=21**R**U | E=21**RP*U |
| F=**SF***U | G=2*SF*PAU | H=21**RPA* | I=21S**P** | J=2*SFR**U |
| K=21**R*** | L=*1SFR**U | M=2*SFRPA* | N=21*F***U | O=21SF**AU |
| P=21*FR*** | Q=21SFR**U | R=2*SFR*A* | S=2*SF***U |

Description of the Analysis Process

The analysis process took a little while to figure out how to accomplish. The track did an excellent job of how to do the analysis, but not how to model the data in a usable form and the tools available. Many of the tools were UNIX oriented and I am on a Windows platform. Never having used any of these before, it took experimenting with many different ones to find the right tools. Here are the tools used and how they were used.

- **SnortSnarf**: Used to analyze the alert files against the current Snort rule set and summarize them into a web based output. This sounded easy until I had to run it on windows. I soon found there were not clear cut directions on how to do this and finally accomplished it through trial and error. First you need to put the include directory’s contents into the perl\site\lib folder. Then you need to ensure that the snort_snarf.pl is there as well. The time modules folder had to be placed in the folder into the same directory. You do not have to compile them under windows to make them work. You also have to use a program to get rid of the MY.NET for the IP address or it dies when running it. I chose WinGrep for this. It was fast. Do not use Notepad it takes a LONG time. I also found that
SnortSnarf is a memory and resource hog. It took a long time to process the logs and kept dying for lack of memory and I have 512 MB in my system. I finally moved it to a system at work with a 1 GB of memory and it did four of the five days well. The fifth day is still working right now. I am leaving it just to see how long it takes. The output is very useful and friendly. I mainly compared it with my other output. SnortSnarf did not become the primary tool as I had anticipated.

✓ Snort_Sort: Breaks the alerts down into a web based output. Lists the alerts and those packets that generated them. I found that this tool was easy to use and was not as resource intensive. I concatenated my logs together by using the command at the command line:

    Copy file1+file2+file3+file4+file5 allfiles

This gave me a concatenated list which I then passed to Snort_Sort for processing. It also gives you the ability to pass it a rule set which I told it to process the results against the latest Snort rules. It did create a big html file, which by the way you have to redirect the output to a file in order to see it.

✓ WinGrep: Used to generate the OOS logs into a format that could be exported into excel and to replace MY.NET with 10.0. Also used to look for certain pieces of information within the files. This was very useful for the OOS files so that they could be imported into excel. I found I had to do it by line of the packet and then export that to a .txt file and then import it into excel. Not too graceful, but it worked.

✓ CSV.pl: Converts the alert file into a CSV format. (From Tod Beardsley’s practical found at http://www.giac.org/GCIA.php). This did exactly what it says very efficiently.

✓ Summarize.pl: Summarizes the data from generated from the CSV.pl into a summary looking at different aspects of the data. (From Tod Beardsley’s practical found at http://www.giac.org/GCIA.php). I did this, however it was not as useful as I had hoped it would be and I ended up just referencing the data on some of it.

✓ Alertcount.pl: Used to total the alerts. Used to compare against the snort_snarf output, since snort_snarf would not process a concatenated file of all of the logs due to a lack of memory. (From Chris Kuethe’s practical found at http://www.giac.org/GCIA.php). This worked great. All I did was combine the results into a spreadsheet and I had a great picture of the alerts.

✓ Scanalyze.pl: Used to process the scan logs (with the flag set not to exclude anything) into a usable format this is then passed to scancount. (From Chris Kuethe’s practical found at http://www.giac.org/GCIA.php) Worked great and was easy to use.

✓ Scancount.pl: Used to total up the scans of the different scan types found in the Scan logs. (From Chris Kuethe’s practical found at http://www.giac.org/GCIA.php) This gave me a good overview of the scans and their types. I combined all the results into an excel spreadsheet.

✓ Excel.exe: Used to organize the OOS logs and aspects of the scan logs into a more usable format.

Once I had my data processed I chose to analyze it as a whole so that I didn’t miss

Part 3: Analyze This
Page 75 of 73
anything between the days. I looked at the five days together for each log type and
analyzed the alerts based on their numbers. I chose to look at the ones with the highest
totals. I wanted to look at the scan logs and OOS logs, but not in a vacuum. So I chose
the five top talkers in each and passed each IP address individually to WinGrep and had it
search each of the log files for its occurrence. The results for each IP address were then
combined into one file and saved by the IP address as a .txt. You now had all of that IP
address’s activity in one file from all three types of logs. This way, you could see if these
outliers played a part in the alerts whether as a prescan or active in the alert itself.

References
Kuethe, Chris “GCIA Practical Assignment” February 22, 2001. URL:
http://www.giac.org/GCIA.php (3 August 2002)

Northcutt, Stephen; Cooper, Mark; Fearnlow, Matt; Frederick, Karen. Intrusion

Northcutt, Steven. IDS Signatures and Analysis, Parts 1 and 2. 2002.


Stevens, Richard. TCP/IP Illustrated, Volume1, The Protocol. Reading. Addison-Wesley

http://www.arin.net

http://cert.uni-stuttgart.de (throughout whole practical)

http://www.google.com (throughout whole practical)

http://www.ripe.net/nicdb.html (throughout whole practical)

www.cert.org (throughout whole practical)

www.cve.mitre.org (throughout whole practical)


www.whitehats.com (throughout whole practical)
# Upcoming Training

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS October Singapore 2020 - Live Online</td>
<td>Singapore, Singapore</td>
<td>Oct 12, 2020 - Oct 24, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS London October 2020</td>
<td>London, United Kingdom</td>
<td>Oct 12, 2020 - Oct 17, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Dallas Fall 2020</td>
<td>Dallas, TX</td>
<td>Oct 19, 2020 - Oct 24, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS London November 2020</td>
<td>, United Kingdom</td>
<td>Nov 02, 2020 - Nov 07, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS San Diego Fall 2020</td>
<td>San Diego, CA</td>
<td>Nov 16, 2020 - Nov 21, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Frankfurt November 2020</td>
<td>, Germany</td>
<td>Nov 30, 2020 - Dec 05, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Cyber Defense Initiative 2020</td>
<td>Washington, DC</td>
<td>Dec 14, 2020 - Dec 19, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>