Global Information Assurance Certification Paper

SANS GCIA Practical version 3.3

Kurt C. Anderson

SANS Network Security

Submitted: February 19th, 2004

Table Of Contents:

Assignment 1 - Intrusion Detection For Remote Users. 1
Abstract. 1
Introduction. 1
Issues of Intrusion Detection Systems. 2
The Remote User and the Network Connection. 3
The Packaged Product For The Remote Client. 3
The Remote IDS 4
Results and Discussion 5
Conclusions 6
Assignment Part 2 – Detect Analysis 7
Detect 1 7
1. Source of Trace. 7
2. Detect was generated by: 9
3. Probability the source was spoofed. 11
4. Description of attack. 11
5. Attack mechanism. 15
6. Correlations. 15
7. Evidence of active targeting. 17
8. Severity. 17
9. Defensive recommendations. 18
10. Multiple choice question. 19
Detect 2 19
1. Source of Trace. 20
2. Detect was generated by: 21
3. Probability the source was spoofed. 21
4. Description of attack. 23
5. Attack mechanism. 24
6. Correlations. 25
7. Evidence of active targeting. 27
8. Severity. 27
9. Defensive recommendations. 28
10. Multiple choice question. 28
Detect 3 29
1. Source of Trace. 29
2. Detect was generated by: 29
3. Probability the source was spoofed. 31
4. Description of attack. 31
5. Attack mechanism. 32
6. Correlations. 32
7. Evidence of active targeting. 34
8. Severity. 34
9. Defensive recommendations. 35
10. Multiple choice question. 35
Assignment Part 3 - Analyze This. 37
Executive Summary 37
Analyzed Logs 38

Most Frequent Events. 38
1) Events Of Interest: Alert - Incomplete Packet Fragments Discarded 39
Sources triggering this attack signature 40
Destinations receiving this attack signature 40
Conclusions and recommendations: 41
2) Events Of Interest: Alert - 10.0.30.3 activity 42
Top Sources triggering this attack signature 42
Conclusions and recommendations: 44
3) Events Of Interest: Alert - 10.0.30.4 activity 44
Destinations receiving this attack signature 45
Conclusions and recommendations: 46
4) Events Of Interest: Alert - EXPLOIT x86 NOOP 46
Sources triggering this attack signature 47
Destinations receiving this attack signature 48
Conclusions and recommendations: 49
5) Events Of Interest: Alert - connect to 515 from inside 50
Destinations receiving this attack signature 51
Conclusions and recommendations: 51
Most Frequent Scan Events 52
Events Of Interest: SYN Scans 55
Conclusions and recommendations: 56
Events Of Interest: UDP Scans 56
Conclusions and recommendations: 56
Events Of Interest: Other Scans 57
Conclusions and recommendations: 57
Events Of Interest: OOS 58
Events Of Interest: OOS – Packet Corruption 58
Events Of Interest: OOS - ECN Bits Set 59
Top External Sources 59
Top External Talker With Suspicious Alerts 60
Conclusions and recommendations: 60
Link Diagram: 62
Appendix A: Tools and Methods 65
Appendix B: SnortSnarf All Alerts 68

Table Of Figures:

Table 1 - Five days of logs. 38
Table 2 - Top 10 alerts by quantity. 39
Table 3 - Top sources for EOI number1 40
Table 4 - Top destinations for EOI number 1 40
Table 5 - Top sources for EOI number2 42
Table 6 - Top destination ports for EOI number2 42
Table 7 - Top sources for EOI number3 45
Table 8 - Top destinations for EOI number3 45
Table 9 - Top destination ports for EOI number3 45
Table 10 - Top sources for EOI number 4 47
Table 11 - Top destinations for EOI number4 48
Table 12 - Top destination ports of EOI number 4 48
Table 13 - Top destinations for EOI number5 51
Table 14 - Top sources for EOI number5 51
Table 15 - Sources not conforming with RFC1179 51
Table 16 - Most frequent TCP flag combinations 53
Table 17 - Top scan type by name 53
Table 18 - Top scan sources 55
Table 19 - Top scan destinations 55
Table 20 - Top Destination Ports 55
Table 21 - Top Destination ports for SYN scans 55
Table 22 - Top Destination ports for UDP scans 56
Table 23 - Hosts scanning for Backdoor Q hosts 57
Table 24 - TCP flags in OOS packets 58
Table 25 - Top sources of OOS packets 58
Table 26 - Top destinations of OOS packets 58
Table 27 - Top external sources 59

Assignment 1 - Intrusion Detection For Remote Users.

Abstract.

With telecommuting and road-warriors, the enterprise network goes well beyond the physical boundaries of the company's walls as soon as a remote user connects to the company network. Corporate defense must go with these users – the perimeter defenses must include the remote resources on the network. These users need good anti-virus protection and a firewall, but what about intrusion detection capabilities?

Many enterprises that currently deploy strong perimeter defenses that include intrusion detection should raise some questions:

"Are intrusion detection capabilities needed by our remote users?".

"Are the returns worth the costs in software, licensing, administration and infrastructure?"

"Will there be enough gains in defensive capabilities to justify the costs?"

Introduction.

The cost burden of expanding the remote defenses must be considered by the company's security group. The cost consideration is not only software prices and licensing, but a very large cost is the resources, both hardware and human to add-on to an IDS.

While large corporations may be perceived to have large budgets to spend on its IT perimeter defenses, it is most often a matter of scale. The larger the corporation, the larger the security budget for defenses, but also, there is will be many more resources to defend. With a corporate network that could include various e-commerce initiatives and multi-site WAN connection and extranets to business partners, their external exposure can be extensive. It is not uncommon that a company could have multiple T1 or OC-3 connections and each of these must run perimeter filter routers and firewalls.

Intrusion detection is one of the fastest growing markets for the 'layered defense' of the corporate network. From http://www.itsecurity.com/tecsnews/sep2003/sep123.htm :

Date posted in ITsecurity.com: 21 September, 2003

> FRAMINGHAM, Mass., Sept. 18 -- According to IDC's Worldwide Quarterly Security Server Appliance Tracker, the worldwide security appliance market grew 10 percent in the second quarter of 2003 when compared to the second quarter of 2002. Factory revenues totaled $333 million which marked an additional $30 million in sales gained from 2Q02. The intrusion detection market experienced strong 29.2 percent year-over-year growth while the firewall/virtual private network (VPN) security appliance market underwent an increase of 7.5 percent.
>
> "In contrast to statements that intrusion detection software is dead, the growth in intrusion detection appliances show that many organizations still see the value in monitoring their networks," said Charles Kolodgy, research director for IDC's Security Products service. "Not surprisingly, the intrusion detection appliance market has been growing as many more intrusion detection software appliances are available compared to a year ago."

Intrusion detection is not an active defense mechanism for the corporation, but rather it is an information gathering capability on the activities of those who probe for vulnerabilities and run attacks. So with IDS systems doing what they do best by monitoring the network, alerting the analyst and logging large amounts of data, do the capabilities of an IDS need to be extended to the far reaches of the corporate network? How much benefit does the company gain from an end user based IDS?

Issues of Intrusion Detection Systems.

Any conversation about intrusion detection software will focus on the very high percentage false alarming and large amounts of data that they tend to create. This raises the issue of the resources necessary to monitor the huge number of events that come out of IDS systems. These resources include servers with high powered processor, large amounts of memory and huge disk storage devices. When accessing system costs, make certain to include the human resources for IDS. Whether it be company employees or a managed security service, there is still a huge cost involved.

Once installed an IDS will require a great deal of time in fine tuning the signatures. Tuning involves the customization of the signature database to fit within its environment. If an e-commerce architecture is running Apache web servers inside of a DMZ, then signatures that trigger for attacks against Microsoft IIS servers are pointless and can be tuned out. This is a simple example and seems intuitive enough, but what about other traffic that is expected on your network that cannot be as easily defined as the web server example. For instance, the company's networking group may have monitoring tools using the SNMP protocols. Again quite simple, but in this case filtering rules must be created to not trigger with legitimate traffic but yet provide

the correct alerts from sources other then those expected.

The Remote User and the Network Connection.

The ability for a remote user to connect to the corporate networks has been around for a long time. Companies would set up a modem pool so their remote users could dial into the modem pool, establish a point to point connection and connect directly into the company's network. Although legacy systems of this type are still in use, availability to the Internet from virtually anywhere has many firms switching to VPN systems. The 'always-on' broadband connection for the remote user has changed many requirements about the remote computer and the necessary protections.

It is not argued that the very minimum security needs are anti-virus software and a firewall on the remote computer. Desktop anti-virus is available from many vendors and has proved to be reliable and inexpensive. With the cost of electronic hardware constantly droping, an inexpensive home router can provide a great deal of firewall like protections through simple network address translation. Despite the low cost and simple setup of small home routers, the software based firewalls are more suited to the mobile remote users, the road warrior.

The Packaged Product For The Remote Client.

Internet Security Systems, a security software leader from Atlanta Georgia, packages an extensive product suite which is typically deployed and controlled through a central management and reporting location and can support many sensing points. For this discussion of remote users and IDS capabilities, the ISS product line, which includes a desktop firewall/ intrusion detection module, will be used as the architecture model. Product specifics and features will not be highlighted other then to help provide a better understanding of a discussion point. There are other vendors, of security products, which provide similar functionality, but this discussion is for current IDS technology and is not a product review. Figure 1, from ISS (http://www.iss.net) is an example of a corporate network that includes various remote connections. It is a 'typical' architecture for today's enterprise. If the focus is on the mobile and the wireless user, the number of remote users could actually the hundreds, thousands or even tens of thousands.

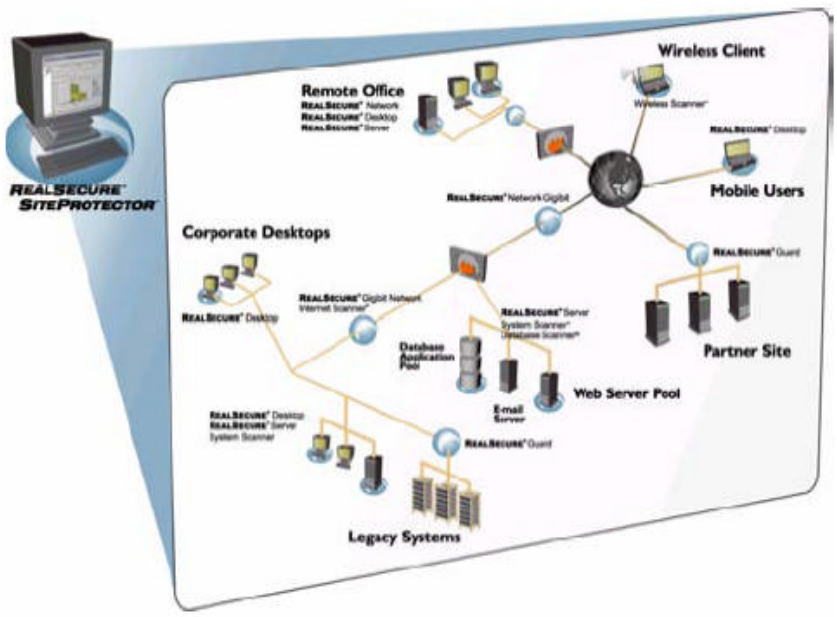


Figure 1- The" New" Enterprise Perimeter

With the desire of security software vendors to become the one stop shop for their corporate customer, many of these security companies try to combine multiple defense capabilities into their products. It is very common to see complimentary types of products bundled into one 'feature rich' bundle. This model, while combining complimentary functions and adding convenience, creates 'bloatware' which can consume a portion of the computing power of even the newest systems. The company that typically has large investments and contractual license agreements, is often left without much choice but to accept the vendor's 'new and improved' product evolutions. It is rare that a company can 'break out' features that it feels it does not need for their requirements. They are then faced with either accepting the product offered by the current vendor or to begin a search for a product that better matches their expectations. Doing this would almost definitely require new capital outlays with long evaluation and test phases.

The Remote IDS

Desktop Protector, from ISS, is an example of the type of software that grows from product mergers and development. As a marketer of IDS products for the medium to large enterprise, ISS acquired a desktop firewall product and merged it with their complimentary desktop products. Rolling the different products into an enterprise desktop security solution, it is a

desktop firewall with integral IDS. After desktop installation and configuration, the central management server will push policy to the remote client, while doing other functions such as version patching and updateing. All very desirable functions and a very big selling point for a system with centralized management. This is very vital for firewall software, just as it is for any anti-virus software to maintain up to date virus pattern files. But what are the benefits or drawbacks to maintaining a remote IDS?

The enterprise that is considering using remote user access to their network and that recognizes the importance of securing that remote asset, is also very likely to be a larger compnay. They will likely have an investment in IDS technologies in their current network infrastructure. This would include both the human and hardware resources. There is a very good chance that the analysts who run this system are stretched to capacity. They are watching their network and host sensors alerting on events in both the internal network and outside the perimeter firewalls. Take a few minutes to discuss the attacker and what an IDS is seeing in way of alerts and events.

When a network IDS sensor is placed outside of a firewall it will quickly become the 'busiest' of all the sensors that a company may have. With the alerting that will come from that sensor a very big portion of it is scanning activity and script kiddies releasing the latest worm. This should be somewhat of a concern, but through strict firewall policy and maintaining and patching the systems, a company can maintain a very secure and stable environment. Ideally, a company can have an IDS sensor inside its firewall and focus their attention to that sensor. The attack that gets through the perimeter is the kind of event that requires immediate response.

The hacker that directs their efforts at a specific site or company are somewhat rare, most attacks are follow-ups to service and vulnerability scans. They are likely to be scripted and directed at IP address ranges and not directed at a particular company. Then most of the events that are happening at a company's perimeter are not likely to be directed attacks, we can easily assume that our remote users will very likely have the same probes and scans launched against their computers.

Results and Discussion

Intrusion detection systems require a great deal of attention, if they are expected to perform their best. It can take a great deal time to tune them to minimize the number of false alarms and to maximize the return of actual events of interest. If a company is running an IDS at their

perimeter, they will inevitably 'see' all the suspicious traffic and attacks on the Internet. The remote user connected to the Internet will certainly see this traffic, also.

It goes without argument that remote users need a very good anti-virus and personal firewall. If these two products can be used with an enterprise ready central management server to make certain that the anti-virus pattern files are current and if the firewall policies can be maintained across the network, then those system make for an ideal perimeter protection system. But is intrusion detection data logging needed at these remote systems?

When the analyst is monitoring what's happening across the Internet from outside the company's perimeter, information from one or two sensors will be very indicative of the current state of the attacks and scans. On an external sensor, a typical default rule base will generate a huge number of detects, many will be false alarms but then there will be large numbers of scanner and worm activity. This will give the analyst a great deal of information for formulating a judgement of the companies defenses.

Conclusions

The information gathered outside the company firewalls is providing a very good idea of the general status of attaker activity.

If a company is looking to add remote user access to the corporate network, there will be many architectures and products to evaluate. Anti-virus and firewalls on the remote client computers are a must. But when those products come with an integral intusion detection software client, the evaluation team should consider all of the issues of IDS capabilities, before selecting a product.

Ask the question: How much real benefit will there be from hundreds, or perhaps thousands, more IDS sensors?

Assignment Part 2 – Detect Analysis

Detect 1

```
[**] BAD-TRAFFIC bad frag bits [**]
08/24-04:52:42.184488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
192.9.100.88 -> 138.97.10.219 TCP TTL:232 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
MF
Frag Offset: 0x0458   Frag Size: 0x0014
0E F4 00 50 02 67 19 28 02 67 19 28 3E 04 00 00  ...P.g.(.g.(>...
81 A0 00 00                                      ....

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

1. Source of Trace.

The source file for this detect was from http://www.incidents.org/logs/Raw/2002.7.24 . Per the README file in the logs/Raw directory, we are told the following about these files:

```
The log files are the result of a Snort instance
running in binary logging mode.  This means that only the packets that
violate the ruleset will appear in the log.

The logs themselves have been sanitized.  All of the IP addresses of
the protected network space have been "munged".  Additionally, the
checksums have been modified to prevent clever people from discovering
the original IP addresses.  You will find that certain keywords
within the packets have been replaced with "X"s.  All ICMP, DNS, SMTP and
Web traffic has also been removed.
```

All additional files, with timestamps close to this file, where downloaded and used to determine if there was other traffic that may provide insight into the alert in question. Using the other files it was also possible to gain some insight regarding the network topology. The two hardware addresses of all captured packets are `0:3:E3:D9:26:C0 and 0:0:C:4:B2:33`, both hardware addresses are registered to Cisco system. We can speculate that the intrusion detection sensor is located between a border router for the private network side and the other router could likely be the Internet service providers. As we make the assumptions that follow, be reminded of the information in the README file regarding the log files. This is from Snort running in binary logging mode so that only packets that violate the rule set are contained in the logs. All of the traffic from the protected network and the Internet were not captured.

The only outbound detects are from the single source address, 138.97.18.88, and all of these packets where to destination ports 80 and 1863. There could be a few reasons for this. It could

```
Frame 3 (196 bytes on wire, 196 bytes captured)
    Arrival Time: Aug 24, 2002 00:22:52.234488000
<SNIP>
Ethernet II, Src: 00:00:0c:04:b2:33, Dst: 00:03:e3:d9:26:c0
    Destination: 00:03:e3:d9:26:c0 (Cisco_d9:26:c0)
    Source: 00:00:0c:04:b2:33 (Cisco_04:b2:33)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 138.97.18.88 (138.97.18.88), Dst Addr: 64.4.12.158
(64.4.12.158)
<SNIP>
Transmission Control Protocol, Src Port: 61924 (61924), Dst Port: 1863 (1863), Seq:
1190209470, Ack: 899290941, Len: 142
    Source port: 61924 (61924)
    Destination port: 1863 (1863)
<SNIP>
MSN Messenger Service
    MSG 3 N XXX\r\n
    MIME-Version: 1.0\r\n
    Content-Type: text/plain; charset=UTF-8\r\n
    X-MMS-IM-Format: FN=MS%20Shell%20Dlg; EF=; CO=0; CS=0; PF=0\r\n
    \r\n
    salaam
```

Illustration 1- Ethereal view of MSN Instant Messenger packet

```
0030  30 2f 2f 45 4e 22 3e 0a 3c 48 54 4d 4c 3e 3c 48   0//EN">.<HTML><H
0040  45 41 44 3e 0a 3c 54 49 54 4c 45 3e 34 30 33 20   EAD>.<TITLE>403
0050  46 6f 72 62 69 64 64 65 6e 3c 2f 54 49 54 4c 45   Forbidden</TITLE
0060  3e 0a 3c 2f 48 45 41 44 3e 3c 42 4f 44 59 3e 0a   >.</HEAD><BODY>.
0070  3c 48 31 3e 46 6f 72 62 69 64 64 65 6e 3c 2f 48   <H1>Forbidden</H
0080  31 3e 0a 59 6f 75 20 64 6f 6e 27 74 20 68 61 76   1>.You don't hav
0090  65 20 70 65 72 6d 69 73 73 69 6f 6e 20 74 6f 20   e permission to
00a0  61 63 63 65 73 73 20 2f 6d 61 69 6e 2f 63 61 74   access /main/cat
00b0  61 6c 6f 67 2f 65 6e 68 61 6e 63 65 64 2e 68 74   alog/enhanced.ht
00c0  6d 6c 0a 6f 6e 20 74 68 69 73 20 73 65 72 76 65   ml.on this serve
00d0  72 2e 3c 50 3e 0a 3c                       r.<P>.<
```

be that internal addresses are hidden by being NAT'ed to one address behind a firewall or by the router. It is also possible that internal users access the Internet through a proxy server. A proxy server can be configured to control what the internal users are permitted to access, thus the explanation for only two different services in the captured packets outbound to the Internet.

Port 80 is usually HTTP web server protocols and port 1863 is typical of MSN Instant Messenger. These two protocols where confirmed using Ethereal to display the dump from file 2002.7.24. Most of this frame is snipped for brevity and only pertinent information was shown here. Note the destination port and the MSN Instant Messenger payload at the end of this output.

From the Ethereal views, Illustration 1 and Illustration 2, we see that these are valid TCP connections. Since we are only attempting to speculate on the network topology we will discuss only a few packets that do not pertain to the detect being discussed. Other outbound traffic to external web servers is very typical and will not be discussed in depth here.

Illustration 2 is another interesting packet that helps determine that we have a web server at

138.97.18.225, and it also gives us a great deal more information about the web server. Again we are only trying to determine the network topology, but it is interesting to look at what we can find from this particular frame.

You can see that this is a reply from a web server on our network and we get information about the type of web server, the operating system it is running on and several of the web server modules in use.

```
Server: Apache/1.3.12 (Unix)  (Red Hat/Linux) mod_jk mod_ssl/2.6.6
OpenSSL/0.9.5a PHP/4.0.1pl2 mod_perl/1.24 FrontPage/4.0.4.3\r\n
```

In the payload is the "`403 Forbidden`" web server error message, which is typical of a user that has entered an incorrect user name and/or password when trying to access a secured web page. In order for the user to have viewed a web page that requires that they enter authentication

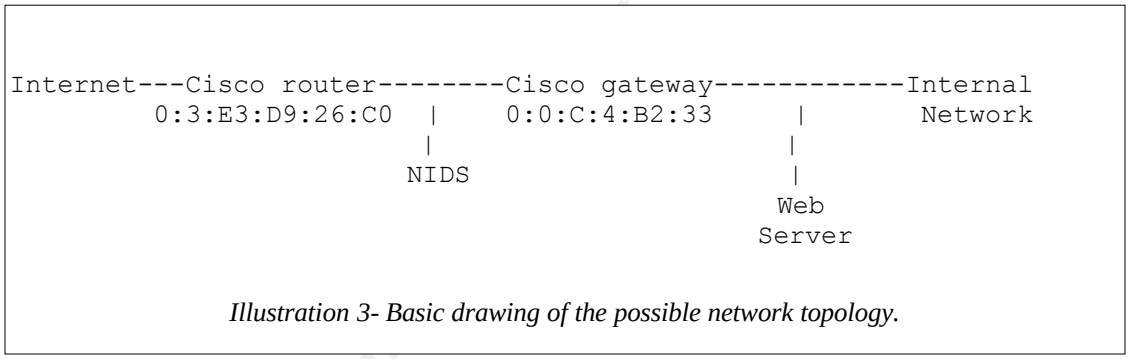


Illustration 3- Basic drawing of the possible network topology.

information, we know that a valid TCP connection had been made.

With this information, we can construct the diagram as shown in Illustration 3. In this illustration, the web server is shown in the internal network. But a network architecture that shows enough sophistication to be running an IDS and firewall, it is very likely that the web server is in a DMZ environment.

2. Detect was generated by:

The Windows port of Snort version 2.0.2 (Build 92) was used for this detect. The current stable ruleset, last modified Thu Nov 6 21:15:10 2003 GMT, was download from http://www.snort.org/dl/rules/snortrules-stable.tar.gz . The raw log files from www.incidents.org

have been sanitized to obfuscate the network address, so the packet checksums are broke. To run Snort against these log files, it is necessary to disable the checksum mode with the option '-k none'.

```
\snort\bin\snort -k none -c \snort\etc\snort.conf -deh 138.97.0.0/16 -r
2002.7.24
```

`-k none`	turn off checksums
`-c`	specify te Snort configuration file
`-d`	display application layer data
`-e`	display link layer data
`-h`	sets the $HOME_NET variable
`-r`	specify the file to read data from

The Snort rule which triggered this alert is from `/etc/snort/bad-traffic.rules` :

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"BAD TRAFFIC bad frag bits";
fragbits:MD; sid:1322; classtype:misc-activity; rev:4;)
```

To understand the format of the alert we look to describe each part, beginning with the `[**]` that start and end the first line. These symbols are characteristics of Snort output and make them easily identifiable as being generated by Snort.

The format for the first line of the alert output is the SID number and the event name.

```
[**] [1:1322:4] BAD TRAFFIC bad frag bits [**]
```

The second line is the event classification and it's priority.

```
[Classification: Misc activity] [Priority: 3]
```

Line three is the date and time stamp followed by the hexadecimal source address and port, and then the destination address and port. The source is always listed first and the destination comes after the arrow symbol. It's direction never points anything but to the right but it helps to improve the ease of reading the output. Next on the first line is the packet type and the packet length

```
08/24-04:52:42.184488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x3C
```

Line four is again the source and destination address, in dotted decimal format, seperated by the direction arrow. After the address information is the protocol type, time to live (TTL), type of service (TOS), then the session ID number, the IP header length and the datagram length. Last on the fourth line is the fragment flags. In this case the 'don't fragment' and 'more fragment' bits are both set.

```
192.9.100.88->138.97.10.219 TCP TTL:232 TOS:0x0 ID:0 IpLen:20 DgmLen:40  DF
MF
```

Line five is the fragment information, which are the fragment offset and the size of the fragment.

```
Frag Offset: 0x0458   Frag Size: 0xFFFFFBBC
```

3. Probability the source was spoofed.

As will be discussed later, there is no evidence that this is active targeting. Although it is not certain if there are any completed TCP connections between hosts on our network and the source addresses found in this detect, if the attacker is trying to elicit responses from the targets then the sender is depending upon getting a response back to the source. Considering that this is stimulus for reconnaissance type of activity, then spoofing the source IP is doubtful.

4. Description of attack.

Using the source address, from the Snort detect above, windump was run with the log file to see other traffic this host was generating and if there was any response from our network. Log analysis was done using Windump version current-cvs.tcpdump.org, which is based on tcpdump version current-cvs.tcpdump.org with WinPcap version 3.0 alpha, based on libpcap version current-cvs.tcpdump.org

```
D:\>windump -Xx -r 2002.7.24 host 192.9.100.88
05:52:42.184488 IP 192.9.100.88 > 138.97.10.219: tcp (frag 0:20@8896+)
0x0000   4500 0028 0000 6458 e806 ff99 c009 6458        E..(..dX......dX
0x0010   8a61 0adb 0ef4 0050 0267 1928 0267 1928        .a.....P.g.(.g.(
0x0020   3e04 0000 81a0 0000 0000 0000 0000             >.............
```

Windump has a great many options and output formats, so for brevity, the format of only the output used here will be described. Note that two output options where selected for this dump that effect the output, -X and -x. These are combined as -Xx, but the result is the same if stated separately. The -x option prints each packet in hexadecimal, while the -X prints the ASCII text. For this run of windump these two options could have been left out and the output would still divulge the most interesting part of this packet, the fragment information.

The most pertinent information is on the first line of the output. It contains the time stamp, source address, destination address and the fragment information. Since this is similar to the Snort alert from earlier, its format will not be described. Only the last part of this line will be explained, as it is the most interesting:

```
05:52:42.184488 IP 192.9.100.88 > 138.97.10.219: tcp (frag 0:20@8096+)
```

This is a identified as a TCP packet and inside the parenthesis you have four pieces of information; The fragment ID, the size of the fragment, the location of the fragment for re-assembly, and an indicator if more fragments are to follow. In this packet the fragment ID is 0, it's size is 20 bytes, it is to be placed at the 8896-th byte and the plus sign tells us that more fragments are to come.

First off, the fragment ID number of 0, this is suspicious as this number is assigned by the sender and is usually randomly generated. Although, the ID of 0 could be random, it is extremely unlikely. Some Linux kernels are known to use the ID of 0, but if you take this and find it in the same packet with both the More Fragment and the Don't Fragment bits set this becomes a packet that is almost certainly crafted.

To determine if this was an isolated packet, the windump command described above, was run against the logs, in Table 1. The filenames of these logs are somewhat deceiving with regards to the date/time stamps and Figure 2 shows the timestamp range of entries in the log files. All of these logs also came from http://www.incidents.org/logs/Raw/ website.

File name	1st packet	last packet
2002.7.24	8/23/03 ~23:00	8/24/03 < 19:00
2002.7.25	8/24/03 > 19:00	8/25/03 < 19:00
2002.7.26	8/25/03 > 19:00	8/26/03 < 19:00
2002.7.27	8/26/03 > 19:00	8/27/03 < 19:00
2002.7.28	8/27/03 > 19:00	8/28/03 < 19:00
2002.7.29	8/28/03 > 19:00	8/29/03 < 19:00
2002.7.30	8/29/03 > 19:00	8/30/03 < 19:00
2002.7.31	8/30/03 > 19:00	8/31/03 < 19:00
2002.8.1	8/31/03 > 19:00	9/01/03 < 19:00
2002.8.2	9/01/03 > 19:00	9/02/03 < 19:00
2002.8.3	9/02/03 > 19:00	9/03/03 < 19:00
2002.8.4	9/04/03 > 19:00	9/04/03 < 19:00
2002.8.5	9/05/03 > 19:00	9/05/03 < 19:00
2002.8.6	9/06/03 >19:00	9/05/03 < 19:00

Figure 2 - Timestamps of log data.

Because the source address may come from a host whose IP address has been dynamically assigned and therefore the exact address may not appear more then once, the BPF filters where modified to check for traffic that may come from any other sources within the same address range. In the previous windump command, a specific host qualifier was used - `host 192.9.100.88`. In these next dumps this filter was changed to specify the network and the the network mask –

```
net 192.9.100.0 mask 255.255.255.0.
```

The result of these dumps are the following:

```
C:\>windump -Xx -r d:\2002.7.24 net 192.9.100.0 mask 255.255.255.0
04:52:42.184488 IP 192.9.100.88 > 138.97.10.219: tcp (frag 0:20@8896+)
0x0000   4500 0028 0000 6458 e806 ff99 c009 6458        E..(..dX......dX
0x0010   8a61 0adb 0ef4 0050 0267 1928 0267 1928        .a.....P.g.(.g.(
0x0020   3e04 0000 81a0 0000 0000 0000 0000             >.............

C:\>windump -Xx -r d:\2002.7.25 net 192.9.100.0 mask 255.255.255.0
21:45:09.524488 IP 192.9.100.88 > 138.97.222.130: tcp (frag 0:20@9000+)
0x0000   4500 0028 0000 6465 e806 ffe4 c009 6458        E..(..de......dX
0x0010   8a61 de82 135d 0050 0606 0c42 0606 0c42        .a...].P...B...B
0x0020   0a04 0000 f01d 0000 0000 0000 0000             ..............

C:\>windump -Xx -r d:\2002.7.26 net 192.9.100.0 mask 255.255.255.0
23:17:38.144488 IP 192.9.100.69 > 138.97.137.248: tcp (frag 0:20@10200+)
0x0000   4500 0028 0000 64fb e806 ffe9 c009 6445        E..(..d.......dE
0x0010   8a61 89f8 04d3 0050 0159 ed1c 0159 ed1c        .a.....P.Y...Y..
0x0020   2c04 0000 79e7 0000 0000 0000 0000             ,...y.........
02:55:26.444488 IP 192.9.100.69 > 138.97.171.46: tcp (frag 0:20@8744+)
0x0000   4500 0028 0000 6445 e806 ff6b c009 6445        E..(..dE...k..dE
0x0010   8a61 ab2e 0d7a 0050 0221 55e6 0221 55e6        .a...z.P.!U..!U.
0x0020   4c04 0000 5be9 0000 0000 0000 0000             L...[.........
05:12:45.864488 IP 192.9.100.88 > 138.97.77.28: tcp (frag 0:20@8896+)
0x0000   4500 0028 0000 6458 e806 ff58 c009 6458        E..(..dX...X..dX
0x0010   8a61 4d1c 11f4 0050 029f 0fe0 029f 0fe0        .aM....P........
0x0020   4c04 0000 407f 0000 0000 0000 0000             L...@.........
11:40:27.064488 IP 192.9.100.88 > 138.97.144.41: tcp (frag 0:20@34232+)
0x0000   4500 0028 0000 70b7 ef06 ffeb c009 6458        E..(..p.......dX
0x0010   8a61 9029 069f 0050 0402 014e 0402 014e        .a.)...P...N...N
0x0020   c004 0000 af24 0000 0000 0000 0000             .....$........

C:\>windump -Xx -r d:\2002.7.27 net 192.9.100.0 mask 255.255.255.0
19:59:52.364488 IP 192.9.100.69 > 138.97.81.233: tcp (frag 0:20@9840+)
0x0000   4500 0028 0000 64ce e806 ff26 c009 6445        E..(..d....&..dE
0x0010   8a61 51e9 08c4 0050 05cb 3e82 05cb 3e82        .aQ....P..>...>.
0x0020   c504 0000 6956 0000 0000 0000 0000             ....iV........

C:\>windump -Xx -r d:\2002.7.28 net 192.9.100.0 mask 255.255.255.0
09:37:00.984488 IP 192.9.100.157 > 138.97.188.164: tcp (frag 0:20@38760+)
0x0000   4500 0028 0000 72ed e806 fff5 c009 649d        E..(..r.......d.
0x0010   8a61 bca4 102f 0050 0130 3e90 0130 3e90        .a.../.P.0>..0>.
0x0020   c004 0000 03f4 0000 0000 0000 0000             ..............
11:56:22.274488 IP 192.9.100.157.1314 > 138.97.222.108.80: R
28300520:28300528(8) win 0
0x0000   4500 0028 0000 0000 e806 ff1c c009 649d        E..(..........d.
0x0010   8a61 de6c 0522 0050 01af d4e8 01af d4e8        .a.l.".P........
0x0020   3e04 0000 3f8b 0000 0000 0000 0000             >...?.........

C:\>windump -Xx -r d:\2002.7.29 net 192.9.100.0 mask 255.255.255.0
18:21:52.884488 IP 192.9.100.88 > 138.97.87.190: tcp (frag 0:20@9352+)
0x0000   4500 0028 0000 6491 e806 ff7d c009 6458        E..(..d....}..dX
```

```
0x0010   8a61 57be 1088 0050 0058 b484 0058 b484        .aW....P.X...X..
0x0020   2f04 0000 0f8e 0000 0000 0000 0000             /.............

C:\>windump -Xx -r d:\2002.7.30 net 192.9.100.0 mask 255.255.255.0
20:51:08.954488 IP 192.9.100.88 > 138.97.129.193: tcp (frag 0:20@10104+)
0x0000   4500 0028 0000 64ef e806 ff1b c009 6458        E..(..d.......dX
0x0010   8a61 81c1 04fa 0050 0015 b7ec 0015 b7ec        .a.....P........
0x0020   da04 0000 3fce 0000 0000 0000 0000             ....?.........
21:04:17.864488 IP 192.9.100.88 > 138.97.126.77: tcp (frag 0:20@8360+)
0x0000   4500 0028 0000 6415 e806 ff6a c009 6458        E..(..d....j..dX
0x0010   8a61 7e4d 1327 0050 0021 c186 0021 c186        .a~M.'.P.!...!..
0x0020   0104 0000 fac9 0000 0000 0000 0000             ..............
03:06:18.274488 IP 192.9.100.88 > 138.97.196.88: tcp (frag 0:20@32952+)
0x0000   4500 0028 0000 7017 e806 ff5c c009 6458        E..(..p....\..dX
0x0010   8a61 c458 1291 0050 002b 1a10 002b 1a10        .a.X...P.+...+..
0x0020   c004 0000 452d 0000 0000 0000 0000             ....E-.........
04:22:22.104488 IP 192.9.100.144 > 138.97.185.166: tcp (frag 0:20@8480+)
0x0000   4500 0028 0000 6424 e806 ffc9 c009 6490        E..(..d$......d.
0x0010   8a61 b9a6 0eb3 0050 0070 bec6 0070 bec6        .a.....P.p...p..
0x0020   0c04 0000 bd8e 0000 0000 0000 0000             ..............
11:40:39.074488 IP 192.9.100.88 > 138.97.44.183: tcp (frag 0:20@8896+)
0x0000   4500 0028 0000 6458 e806 ffbd c009 6458        E..(..dX......dX
0x0010   8a61 2cb7 0974 0050 0202 0376 0202 0376        .a,..t.P...v...v
0x0020   0104 0000 ce72 0000 0000 0000 0000             .....r........

C:\>windump -Xx -r d:\2002.7.31 net 192.9.100.0 mask 255.255.255.0
C:\>windump -Xx -r d:\2002.8.1 net 192.9.100.0 mask 255.255.255.0
C:\>windump -Xx -r d:\2002.8.2 net 192.9.100.0 mask 255.255.255.0
20:53:38.464488 IP 192.9.100.88 > 138.97.212.200: tcp (frag 0:20@10200+)
0x0000   4500 0028 0000 64fb ec06 ff08 c009 6458        E..(..d.......dX
0x0010   8a61 d4c8 047b 0050 00c5 2fa6 00c5 2fa6        .a...{.P../.../.
0x0020   1e04 0000 b873 0000 0000 0000 0000             .....s........
21:33:55.034488 IP 192.9.100.144 > 138.97.117.153: tcp (frag 0:20@8680+)
0x0000   4500 0028 0000 643d ec06 ffbe c009 6490        E..(..d=......d.
0x0010   8a61 7599 10ed 0050 00ea 0e02 00ea 0e02        .au....P........
0x0020   5504 0000 16f7 0000 0000 0000 0000             U.............
01:49:40.304488 IP 192.9.100.88 > 138.97.2.40: tcp (frag 0:20@9160+)
0x0000   4500 0028 0000 6479 ec06 ff2b c009 6458        E..(..dy...+..dX
0x0010   8a61 0228 0d88 0050 01d4 3716 01d4 3716        .a.(...P..7...7.
0x0020   3004 0000 5f09 0000 0000 0000 0000             0..._.........

C:\>windump -Xx -r d:\2002.8.3 net 192.9.100.0 mask 255.255.255.0
C:\>windump -Xx -r d:\2002.8.4 net 192.9.100.0 mask 255.255.255.0
C:\>windump -Xx -r d:\2002.8.5 net 192.9.100.0 mask 255.255.255.0
C:\>windump -Xx -r d:\2002.8.6 net 192.9.100.0 mask 255.255.255.0
```

From the log file information in Figure 2 we can see that there is more than thirteen days of consecutive log data. Packets sourced from the 192.9.100.0 network ended after 9/02/2003, but without files prior to the first log used for these detects, it is not possible to know if other fragmented packets had been sent to this network. Nor, do we know if any other packets of interest had come from this source network.

5. Attack mechanism.

There was only one packet in the initial detect using the log file 2002.7.24, so using only this log file it appeared that this could possibly be a single, incorrectly configured packet. A broken router could mistakenly fragment a packet despite its DF bit being set. In this case all but one of the packets received from these addresses had the same type of fragmentation anomalies and the other single packet from the source network, 192.9.100.0, was a reset packet. This single reset packet, to destination port 80, does not appear to be a response to any session with a host inside the network. There are no other packets in the logs from the source address of 138.97.222.108 and we could not determine if this is a web server, nor if it is even an active host. It is not certain what type of gateway is on the perimeter of our network, but if this gateway is a firewall with stateful packets capabilities then the reset packet would be dropped since it does not appear to be from an existing session maintained by a stateful firewall.

It is also a point of interest to look at the timestamps of these packets. Illustration 4, in Part 7 of this detect, is a list of only the first line of each packet captured with windump. This makes it easier to read the relationships of the source addresses and destination addresses. From this, it's easy to see the time that elapsed between each packet, with some timestamps between the suspicious packets arriving many hours apart. If this is an attempt to gather information then it can definitely be called a low and slow reconnaissance.

6. Correlations.

There where no direct correlations found that closely match these detects. The Internet Storm Center has no reports of suspicious traffic from neither the detected source address nor the network.

Nmap could be used to do a scan such as this, but nmap is more the type of tool to do wider based scans and at the same time it will often trip other signatures that identify it as being an nmap scan. It would be more likely that a tool such as SendIP was used. At http://freshmeat.net/projects/sendip/ we can see from the description of this tool makes it an interesting possibility.

> **About:**
> SendIP is a command-line tool to send arbitrary IP packets. It has a large number of options to specify the content of every header of a RIP, RIPng, BGP, TCP, UDP, ICMP, or raw IPv4/IPv6 packet. It also allows any data to be added to the packet. Checksums can be calculated automatically, but if you wish to send out wrong checksums, that is supported too.

Checking the source of the fragmented packets sent to this network yields the following from an ARIN whois search.

Search results for: 192.9.100.88

OrgName: Sun Microsystems, Inc
OrgID: SUN
Address: 4150 Network Circle
City: Santa Clara
StateProv: CA
PostalCode: 95054
Country: US

NetRange: 192.9.10.0 - 192.9.199.255
CIDR: 192.9.10.0/23, 192.9.12.0/22, 192.9.16.0/20, 192.9.32.0/19, 192.9.64.0/18, 192.9.128.0/18, 192.9.192.0/21
NetName: SUN3
NetHandle: NET-192-9-10-0-1
Parent: NET-192-0-0-0-0
NetType: Direct Assignment
NameServer: NS1.SUN.COM
NameServer: NS2.SUN.COM
NameServer: NS7.SUN.COM
NameServer: NS8.SUN.COM
Comment:
RegDate: 1983-10-17
Updated: 2003-10-13

TechHandle: IS189-ARIN
TechName: Sun Microsystems, Inc.
TechPhone: +1-303-272-7000
TechEmail: Netmaster@sun.com

OrgTechHandle: IS189-ARIN
OrgTechName: Sun Microsystems, Inc.
OrgTechPhone: +1-303-272-7000
OrgTechEmail: Netmaster@sun.com

ARIN WHOIS database, last updated 2003-11-28 19:15
Enter ? for additional hints on searching ARIN's WHOIS database.

```
19:59:52.364488 IP 192.9.100.69 > 138.97.81.233: tcp (frag 0:20@9840+)
09:37:00.984488 IP 192.9.100.157 > 138.97.188.164: tcp (frag
0:20@38760+)
11:56:22.274488 IP 192.9.100.157.1314 > 138.97.222.108.80: R
28300520:28300528(8) win 0
18:21:52.884488 IP 192.9.100.88 > 138.97.87.190: tcp (frag 0:20@9352+)
20:51:08.954488 IP 192.9.100.88 > 138.97.129.193: tcp (frag
0:20@10104+)
21:04:17.864488 IP 192.9.100.88 > 138.97.126.77: tcp (frag 0:20@8360+)
03:06:18.274488 IP 192.9.100.88 > 138.97.196.88: tcp (frag 0:20@32952+)
04:22:22.104488 IP 192.9.100.144 > 138.97.185.166: tcp (frag
0:20@8480+)
11:40:39.074488 IP 192.9.100.88 > 138.97.44.183: tcp (frag 0:20@8896+)
20:53:38.464488 IP 192.9.100.88 > 138.97.212.200: tcp (frag 0:20@10200+)
21:33:55.034488 IP 192.9.100.144 > 138.97.117.153: tcp (frag
0:20@8680+)
01:49:40.304488 IP 192.9.100.88 > 138.97.2.40: tcp (frag 0:20@9160+)
04:52:42.184488 IP 192.9.100.88 > 138.97.10.219: tcp (frag 0:20@8896+)
21:45:09.524488 IP 192.9.100.88 > 138.97.222.130: tcp (frag 0:20@9000+)
23:17:38.144488 IP 192.9.100.69 > 138.97.137.248: tcp (frag
0:20@10200+)
02:55:26.444488 IP 192.9.100.69 > 138.97.171.46: tcp (frag 0:20@8744+)
05:12:45.864488 IP 192.9.100.88 > 138.97.77.28: tcp (frag 0:20@8896+)
11:40:27.064488 IP 192.9.100.88 > 138.97.144.41: tcp (frag 0:20@34232+)
```

Illustration 4- First line only of windump output

7. Evidence of active targeting.

When looking at the destination addresses of the fragmented packets, you can see that no destination address is repeated. With that we should believe that this is not active targeting.

8. Severity.

To calculate the severity of this detect the formula, prescribed by the GIAC GCIA instructions, is used. From http://www.giac.org/GCIA_assignment.php :

severity = (criticality + lethality) (system countermeasures + network countermeasures)
Each value should be ranked on a scale from 1 (lowest) to 5 (highest).

Criticality is a measure of how critical the targeted system is.
Lethality is a measure of how severe the damage to the targeted system would be if the attack succeeded.
System countermeasures is a measure of the strength of the defensive mechanisms in place on the host itself.
Network countermeasures is a measure of the strength of the defensive mechanisms in place on the network.

This trace's severity:

Criticality = 4

We do not know anything of the targets in these traces, so we should assume a high level of importance. Based on the Snort logs, there where no other packets with the same destination addresses and it may be assumed that there may not be active hosts at the destination. Therefore, we could take a slightly paranoid criticality of 5 and reduce it to a 4.

Lethality = 3

Again, the unknowns may have us assume too much about the destinations. In this case we have judged this is not active targeting and more likely reconnaissance. A somewhat higher rating of 3 here is sufficient, but in reality, it is probably a 2 or less.

System countermeasures = 2

Not knowing the destination's operating system, its version and/or patch levels, nor if they are active hosts, a 2 is an adequately low rating for the system countermeasure.

Network countermeasures = 4

This is the one area that we can make better determinations as to the network defenses that are in place. With perimeter routers, gateway firewalls and possible use of proxy servers, the defenses for this detect are strong.

Plugging the above values into our formula:

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

severity = (4 + 3) - (2+ 4) = 1

This rating would be worthy of the analyst's time in confirming the circumstances of the parameters used in this calculating the severity. The numbers here are somewhat conservative and an investigation would probably lower this to a 0 or less. A revised severity level that would not warrant an increased urgency in reporting to a CIRT team.

9. Defensive recommendations.

Due to the source from an private IP address assigned to a well known corporation, blocking this address and/or address range at the network perimeter could deny legitimate network traffic. If the packet is indeed crafted, it will not be part of an existing TCP session, therefore a stateful firewall will drop it, so the minimum of a stateful firewall along with continuing the use of the

current intrusion detection system, is recommended. An IDS requires on-going analysis from skilled personal and the network owners should have processes and procedures in place for incident response. The handling of IDS findings should be part of a security policy and reported to a CIRT team.

10. Multiple-choice question.

In this line from the detect

05:52:42.184488 IP 192.9.100.88 > 138.97.10.219: tcp (frag 0:20@8096+)

The fragment identification number, used for re-assembly at the destination is:

A) 184488
B) 0
C) 20
D) 8096

Answer: B)

In this packet the fragment ID is 0, it's size is 20 bytes, it is to be placed at the 8896-th byte and the plus sign tells us that more fragments are to come.

Detect 2

One of the goals of the practical is for the GCIA candidate to show their skills using different tools used for intrusion detection. This detect was made using a commercial product, which will be explained more in depth later in this document. For now, here is the detect that was exported from an ISS Real Secure event report.

Event Number : 1
Date/Time : 2003-11-03 11:02:05 CST
Tag Name : DNS_Hostname_Overflow
Alert Name : DNS_Hostname_Overflow
Severity : High
Tag Brief Description :
Observance Type : Intrusion Detection
Combined Event Count : 1
Cleared Flag : No
Target DNS Name :
Target IP Address : ccc.yyy.138.122
Target Object Name : 53
Target Object Type : Target Port
Target Service :
Source DNS Name :
Source IP Address : 80.38.19.30
SourcePort Name : 26779
Sensor DNS Name : dsvpsu83.ourdomain.com
Sensor IP Address : bbb.zzz.88.21

Sensor Name : network_sensor_1

Attribute Value Pairs for Event Number : 1
Attribute Name : :length
Attribute Value : 218
Attribute Name : :dnsname
Attribute Value : <empty>
Attribute Name : :victim-ip-addr
Attribute Value : ccc.yyy.138.122
Attribute Name : :victim-port
Attribute Value : 53
Attribute Name : :intruder-ip-addr
Attribute Value : 80.38.19.30
Attribute Name : :intruder-port
Attribute Value : 26779
Attribute Name : algorithm-id
Attribute Value : 2000403
Attribute Name : Packet DestinationAddress
Attribute Value : ccc.yyy.138.122
Attribute Name : Packet SourcePort
Attribute Value : 26779
Attribute Name : Packet DestinationPort
Attribute Value : 53
Attribute Name : Packet DestinationPortName
Attribute Value : domain
Attribute Name : Packet SourceAddress
Attribute Value : 80.38.19.30

1. Source of Trace.

The source of this detect is from an actual corporate e-commerce environment. The detection point is outside of five Checkpoint firewalls. For the e-commerce production environment, two of the firewalls are configured in a high availability configuration. Two more of the firewalls are the gateway to two separate silos, one each for pre-production and user testing. The fifth firewall handles only outbound traffic from the internal private networks. All traffic into and out of these silos passes the network intrusion detection sensing device.

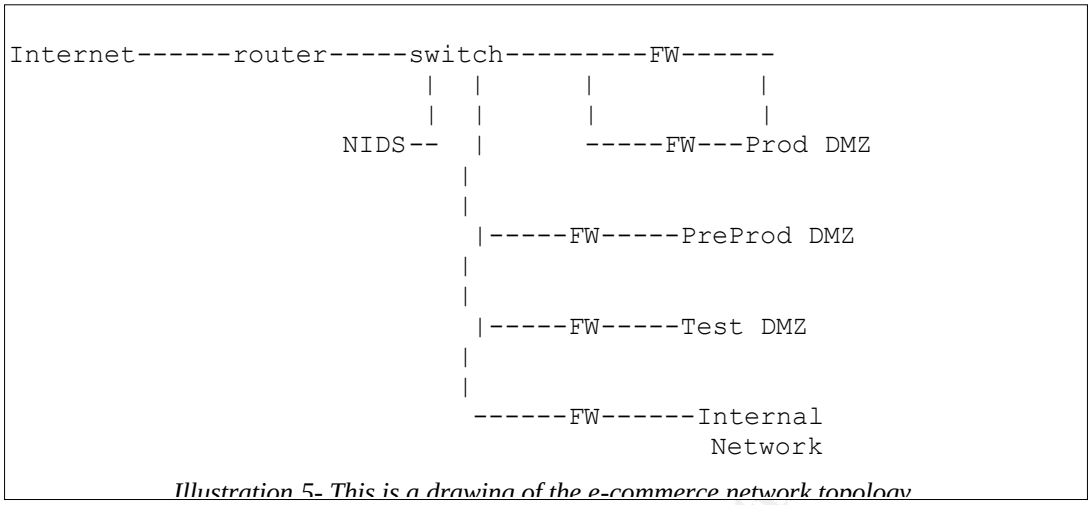


Illustration 5- This is a drawing of the e-commerce network topology

2. Detect was generated by:

This detection was made by ISS Network Sensor version 7.0, with the latest XPU, on a Sun server running the Solaris 2.8 platform. A Cisco switch mirrors the traffic on a VLAN to the network interface of the Network Sensor. The Sensor is dual homed with the interface for detection in the promiscuous mode, which prevents its detection on the network. The second interface is connected to an out of band network for the ISS front-end tools, reports and databases. Events from the Network Sensor devices are collected and monitored with the ISS Site Protector suite of products. The Site Protector product is GUI based and the above log is the text export of the detected event. The format is not widely known but the labels for each data field are rather self-explanatory. The source address is as actually detected and only the destination and sensor address have been obfuscated.

3. Probability the source was spoofed.

The source is a valid IP address that is from Telefonica de Espana, so to determine if this is or is not a spoofed packet we must rely on what we believe the intent of the sender of this packet. For instance, from the Impact section of CERT Advisory CA-2002-19 “Buffer Overflows in Multiple DNS Resolver Libraries”:

http://www.cert.org/advisories/CA-2002-19.html

II. Impact

An attacker who is able to send malicious DNS responses could remotely exploit these vulnerabilities to execute arbitrary code or cause a denial of service on vulnerable systems. Any code executed by the attacker would run with the privileges of the process that calls the vulnerable resolver function.

Note that an attacker could cause one of the victim's network services to make a DNS request to a DNS server under the attacker's control. This would permit the attacker to remotely exploit these vulnerabilities.

Figure 3 - From CERT Advisory CA-2002-19

If the sender is trying to exploit vulnerabilities to gain access to the remote sever then they are not likely to be using a spoofed source address. In the vendors help screen used to explain this event to the analyst, the vendor does not provide references to this or any other CERT advisory.

References

Microsoft TechNet, Windows NT Server Concepts and Planning Manual
Chapter 9 - Monitoring Events
http://www.microsoft.com/technet/winnt/Winntas/manuals/concept/xcp09.asp
Internet Software Consortium (ISC) Web site
Current release
http://www.isc.org/products/BIND
ISS X-Force
DNS hostname exceeding maximum length
http://www.iss.net/security_center/static/636.php

Figure 4 - References from the vendor's explanation of this event.

Port 53 DNS is almost perpetually in the Top 10 List of most attacker services. See the most recent data at the Internet Storm Center Ports Report for to:

http://isc.sans.org/port_details.html?port=53

In a comment from Marcus H. Sachs, of the SANS Institute:

```
SANS Top-20 Entry:
U1 BIND Domain Name System

http://isc.sans.org/top20.html#u1

<snip>

The ubiquity and critical nature of BIND has made it a frequent
```

```
target, especially in Denial of Service (DoS) attacks, which can
result in a complete loss of accessibility to the Internet for
services and hosts.

<snip>

-----
Submitted by: Marcus H. Sachs, SANS Institute
```

This event is likely an attempt to cause a Denial of Service by exploiting an old vulnerability in BIND. For this situation, the source is probably spoofed.

4. Description of attack.

Before an event is triggered in a commercial product such as ISS Real Secure, the packet is compared to an attack signature database the same as it would be in open source detection products such as Snort. For the commercial product from ISS, attack signature data is considered proprietary and the vendor does not release the specifics of these signatures. In this event we see that a packet was sent from 80.38.19.30 port 53 destined for our host, ccc.yyy.138.122 port 53. Referring to the description of the signature, that ISS calls DNS_Hostname_Overflow we get the explanation in Figure 5.

DNS hostname exceeding maximum length (DNS_Hostname_Overflow)

About this signature or vulnerability

RealSecure Server Sensor, BlackICE Server Protection, BlackICE PC Protection, RealSecure Sentry, RealSecure Guard, BlackICE Agent for Server, RealSecure Desktop Protector, RealSecure Network Sensor, RealSecure OS Sensor:

This signature detects a Windows event log message indicating that a domain name exceeding the maximum name length has been detected by the Microsoft DNS Server.

Systems affected

Any application: Any version, DNS: Any version, Windows 2000: Any version, BIND: 4.x, Windows NT: Any version

Type

Suspicious Activity

Vulnerability description

DNS responses for hostnames should not exceed a certain fixed length. A domain name exceeding the maximum length of 255 octets could indicate one of the following events:

a zone file error

incorrectly entered hostnames in nslookup queries

an attempt by an attacker to manipulate the DNS server

When Microsoft DNS Server encounters a resource record with a domain name exceeding the maximum length of 255 octets, the resource record is ignored by the DNS server.

Versions 4.x and earlier of BIND (Berkeley Internet Name Domain, a DNS server available for most versions of Unix) do not validate the maximum domain name length of 255 octets. Hostnames longer than this length can be returned to client programs performing DNS lookups. Client programs that do not check the length of the hostnames returned may overflow internal buffers when copying this hostname, allowing a remote attacker to gain root access or execute arbitrary commands on a targeted client computer.

Figure 5 - From IDS vendors help screen of this signature.

5. Attack mechanism.

The 'Vulnerability description' tells us what may have triggered the event. First off, it states that a DNS response should not exceed 255 octets, or bytes, and that exceeding this limit could be one of three things. A zone file error, an incorrectly entered request, or an attempted attack on a server.

Zone file transfers are done only using TCP and not UDP, but ISS does not let the analyst know if this packet is UDP or TCP. Looking back at the trace the field that contains the hostname in what would be the payload is 218 bytes long, but yet the contents of the field are empty.

Attribute Name : :length
Attribute Value : 218
Attribute Name : :dnsname
Attribute Value : <empty>

In reviewing other events of this type, captured previously on this system, we find that indeed there are mis-entered host names and others with 'garbage' characters. Here are snippets of two other events from the same ISS detection device.

```
Event Number : 3
Date/Time : 2003-11-05 12:58:55 CST
Tag Name : DNS_Hostname_Overflow

<snip>

Attribute Value Pairs for Event Number : 3
Attribute Name : :dnsname
Attribute Value : .5

<snip>


Event Number : 7
Date/Time : 2003-11-17 11:38:40 CST
Tag Name : DNS_Hostname_Overflow

<snip>

Attribute Value Pairs for Event Number : 7
Attribute Name : :dnsname
Attribute Value : vsÆ.‰¥Öš5

<snip>
```

Note the garbage characters in 'Event 7' and what looks like a mis-typed IP address in 'Event 3'. In both of these we can at least see what was happening, the second event being an error and the first one may be someone trying to throw a malicious character string at what they thought, or hoped, was a DNS server.

That packet may have been a feeble attempt at some DNS vulnerability, or it may be a stimulus packet looking for a response in a reconnaissance scan. In an attempt to clarify as to why, in our original detect, that the value for the 'dnsname' field is '<empty>', the log was sent to the support team from ISS. They did not provide any further insight and as RealSecure does not log the raw packets, by default, there is no other information to go on.

6. Correlations.

A Whois search of the records finds the source address belonging to the Telefonica de Espana. This ISP is most likely assigning IP addresses dynamically and therefore, making source address correlations is not very likely. If the ISP is truly using DHCP, the nslookup information on the current assignee of this IP address may not be the actual host that sent the packet in this trace. From the host name we speculate further that IP address provisioning is from a DHCP server's 'pooles' of addresses.

```
#nslookup 80.38.19.30
Server:  mydns
Address:  192.168.0.254

Name:    30.Red-80-38-19.pooles.rima-tde.net
Address:  80.38.19.30
```

For background information the following is returned from a Whois search:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:      80.36.0.0 - 80.39.255.255
netname:      RIMA
descr:        TELEFONICA DE ESPANA
descr:        Provider Local Registry
country:      ES
admin-c:      AFG2-RIPE
admin-c:      JB986-RIPE
tech-c:       FLT14-RIPE
tech-c:       FSB3-RIPE
status:       ASSIGNED PA
remarks:      ************************************************
remarks:      For ABUSE/SPAM/INTRUSION issues
remarks:      PLEASE CONTACT THROUGH LINK
remarks:      http://www.telefonicaonline.com/nemesys/
remarks:      or send mail to nemesys@telefonica.es
remarks:      any mail to adminis.ripe@telefonica.es will be ignored
remarks:      ************************************************
mnt-by:       MAINT-TdE
mnt-lower:    MAINT-TdE
mnt-routes:   MAINT-TdE
changed:      adminis.ripe@telefonica.es 20020530
changed:      adminis.ripe@telefonica.es 20030924
source:       RIPE

route:        80.39.0.0/16
descr:        RIMA (Red IP Multi Acceso)
origin:       AS3352
mnt-by:       MAINT-AS3352
changed:      adminis.ripe@telefonica.es 20011115
source:       RIPE

person:       Antonio Fuentes
address:      TELEFONICA DE ESPANA
address:      Emilio Vargas, 4
address:      28043-MADRID
address:      SPAIN
phone:        +34 91 5846497
fax-no:       +34 91 5842650
remarks:      ************************************************
remarks:      For ABUSE/SPAM/INTRUSION issues
remarks:      PLEASE CONTACT THROUGH LINK
```

```
remarks:        http://www.telefonicaonline.com/nemesys/
remarks:        or send mail to nemesys@telefonica.es
remarks:        any mail to adminis.ripe@telefonica.es will be ignored
remarks:        ***********************************************
```

```
<snipped for brevity>
```

There are no reports at the Internet Storm Center from this source address.

7. Evidence of active targeting.

There is only one packet from this source directed to the external Class C address range on this network, which has approximately 100 active addresses open to the Internet. The destination address in this trace is not currently assigned to a host and it is not a DNS server. With this, active targeting is not what has happened here.

8. Severity.

To calculate the severity of this detect the formula, prescribed by the GIAC GCIA instructions, is used. From http://www.giac.org/GCIA_assignment.php :

severity = (criticality + lethality) (system countermeasures + network countermeasures)
Each value should be ranked on a scale from 1 (lowest) to 5 (highest).

Criticality is a measure of how critical the targeted system is.
Lethality is a measure of how severe the damage to the targeted system would be if the attack succeeded.
System countermeasures is a measure of the strength of the defensive mechanisms in place on the host itself.
Network countermeasures is a measure of the strength of the defensive mechanisms in place on the network.

This trace's severity:

Criticality = 0

There is no active host at the destination address.

Lethality = 4

From the original capture of this trace the event was logged with no data in the dnsname field in the database, shown below, but is does appear to have been 218 bytes long.

Attribute Name : :length
Attribute Value : 218
Attribute Name : :dnsname

Attribute Value : <empty>

With the detection made by a commercial product vendor who considers their signature database to be proprietary, we cannot see the specifics that triggered it. With the content not being known it would be wise to think of the packets as possibly being crafted and therefore it's intent is malicious. A paranoid level of 4 properly covers the unknowns that this intrusion detection system has logged, or in this case, has not logged.

System countermeasures = 5
The destination was an unused IP address and no system weaknesses are possible

Network countermeasures = 5

The DNS servers for this e-commerce network are not hosted internally so the firewall rules drop all inbound packets for port 53, using both TCP and UDP protocols.

Plugging the above values into our formula:

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

severity = (0 + 4) - (5+ 5) = -6

This rating will not require further investigation and does not require escalation to a CIRT.

9. Defensive recommendations.

The firewalls are configured to drop all traffic destined to the Class C network range used by this company, so no additional defensive recommendations would be made.

10. Multiple-choice question.

From this detect's logged output, the event was triggered by:

A) A zone file error
B) A user who incorrectly entered hostnames in nslookup queries
C) An attempt by an attacker to manipulate the DNS server
D) There is not enough output to determine.

The log does not tell us if this is TCP or UDP. If it is a zone file transfer it would need to TCP, not UDP. So we cannot be certain of answer A).

Answer B) is not conclusive. From one of the other two examples of this detect there was a field which contained the text '.5', which could be a user of nslookup incorrectly entering an IP

address for a DNS lookup. Although our log reported the DNS name field as '<empty>' it also had a length of 218 bytes, which leaves us uncertain if this is a user who 'fat-fingered' their nslookup command by sending an empty request.

Although, this analysis leaned heavily toward this event being a malicious packet targeted at a DNS server, the information coming from only the log file did not provide conclusive evidence. Thus, answer C) is not our best answer to this question.

This best answer here is D). We do not have enough information from only the log output.. Recall that this IDS vendor considers it's signature algorithm to be confidential, so it is not possible to compare this detect with the rule to look for possible causes of the match to the signature which triggered this event.

Detect 3

```
12/10-00:31:13.297175  [**] [1:504:2] MISC source port 53 to <1024 [**]
[Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 193.135.0.83:53
-> 10.0.0.2:139

[**] MISC source port 53 to <1024 [**]
12/10-00:31:13.297175 0:E0:D0:13:4D:16 -> 0:C0:DF:E0:33:1A type:0x800
len:0x3C
193.135.0.83:53 -> 10.0.0.2:139 TCP TTL:48 TOS:0x0 ID:666 IpLen:20 DgmLen:40
******S* Seq: 0x29A  Ack: 0x0  Win: 0x80  TcpLen: 20

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

1. Source of Trace.

The source of this trace is a home network with a DSL connection.

2. Detect was generated by:

The detect was made with Snort Version 2.0.0 (Build 72). It is part of the default install of IPCop version 1.3, a Linux distribution specifically built as a firewall.

```
# uname -a
Linux ipcop 2.4.21 #1 do jul 31 17:06:58 CEST 2003 i586 unknown
```

The IPCop firewall uses iptables v1.2.7a, more information on this distribution can be found at here: http://www.ipcop.org

The Snort rule set is not modified from the standard installation of IPCop and the rule which triggered this alert is:

```
alert tcp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to
<1024"; flags:S; reference:arachnids,07; classtype:bad-unknown; sid:504;
rev:2;)
```

The ISP assigns the DSL router's external interface's address dynamically and the DSL router uses network address translation, or NAT, of the private addresses on its internal LAN interface. NAT provides some level of protection by hiding a host's actual address; all outbound traffic appears to come from one address, which is not the actual host address. If a packet arrives at the external address of the DSL router, only those that match the port and protocol of a NAT table entry will be passed to the internal address designated in that table entry. The Cisco 675 DSL router allows the user to add custom entries in the NAT table, so to increase the chances of capturing a wider range of detects and already having the protection of a firewall inside of the DSL router, the router's NAT table was modified to forward all traffic from the Internet to the firewall's interface.

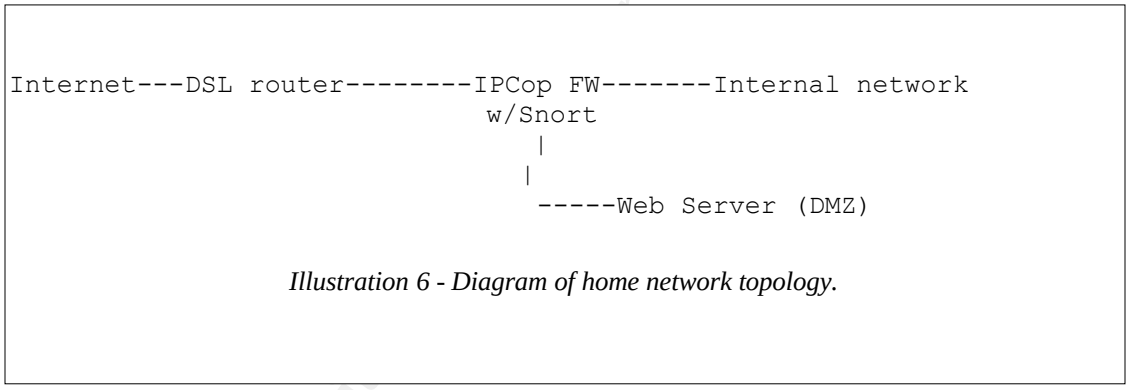


Illustration 6 - Diagram of home network topology.

The default installation of the IPCop firewall gives the user the option to enable intrusion detection if they wish. In enabling IDS, Snort is run in a basic network intrusion detection mode. All packets are not logged, only those that match a rule set as specified in the snort.conf file. In this mode the alerts logged in ASCII format and are placed in directory named for the source IP in the packet. For further analysis with tools such as tcpdump and Ethereal, the binary logs are not available. The home network address is passed to Snort at startup so the alerts are logged based upon this network address. This way the logs are put in directory names based on the remote address and not the home address.

3. Probability the source was spoofed.

This is a TCP SYN packet, which would mean that the sender might be trying to establish a connection to the destination, which would require a SYN/ACK reply. If the intent was to send only a SYN, such as in a half open SYN scan, the sender would still be expecting a response but would not respond when a SYN/ACK was returned. If either of these were true in this trace, then the source address would not have been spoofed.

According to the arachNIDS database at the Whitehats website, the packet which triggered the alert is considered to likely be spoofed. This information is from IDS7 "SOURCE PORT TRAFFIC-53-TCP"

> **Trusting The Source IP Address**
>
> Although this event was caused by a TCP packet, the packet is not thought to be a part of an existing TCP session. Therefor the source IP address could be easily forged.

http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids7&view=event

As will be discussed in the following sections, this detect has the appearance of being a stimulus packet and it's intent is very likely for reconnaissance purposes. As such, the user is expecting a response and the source is probably not spoofed.

4. Description of attack.

The signature of this attack is a packet sent from a source port 53(DNS) to a port less then 1024. Ports that are numbered less then 1024 are the reserved ports and are considered 'privileged', only to be used by process running with a high level of authority. In this packet the destination port is 139/tcp which is the NetBIOS datagram port, the “Windows File and Printer Sharing” port. This is the service often left on by unknowing and uninformed users. Almost all current firewalls use a default rule to drop inbound traffic except that which is specifically permitted, so they would block the NetBIOS ports by default and most often both inbound and outbound traffic on port 139 is blocked.

A note should be given to the Session ID - `ID:666` . This, by itself, is not an indicator of any type of malicious behavior except that it is in an already suspicious packet. As such, packet crafting is very likely.

5. Attack mechanism.

Using the source port 53 is often an attempt to bypass packet filters that allow all incoming traffic from DNS servers. Traffic from port 53 to a port less then 1023 is not normal and in the case of this detect, the destination port is 139, the Windows File and Printer Sharing port. Again, from the arachNIDS database at the Whitehats website, this information is from IDS7 "SOURCE PORT TRAFFIC-53-TCP"

> **Summary**
> This event indicates that an attacker is making a connection to a privileged port using the source port 53 (dns). This should not normally occur. Old or misconfigured packetfilters may allow the connection if they allow all dns traffic.
>
> http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids7&view=event

6. Correlations.

There is no CVE for this specific attack, but sourcing from port 53 to so-called 'privileged' ports is definitely an exploit of packet filter vulnerabilities. CVE does report numerous attacks to the destination port 53.

```
CVE-2001-001 - Buffer overflow in transaction signature (TSIG) handling code
in BIND 8 allows remote attackers to gain root privileges.
CVE-1999-0833 - Buffer overflow in BIND 8.2 via NXT records.
CVE-1999-0010 - Denial of Service vulnerability in BIND 8 Releases via
maliciously formatted DNS messages.
CVE-1999-0009 - Inverse query buffer overflow in BIND 4.9 and BIND 8
Releases.CAN-1999-0532 -A DNS server allows zone transfers.
```

With regards to the session ID of 666, there are no correlations that could be made that tie into the suspicious characteristic of the capture. The ID:666 is a known characteristic of the Stacheldraht denial of service attack, but this packet is not similar to that attack in other primary characteristics. From an old Stracheldaht analysis by David Dittrich <dittrich@cac.washington.edu>, University of Washington, Copyright 1999. All rights reserved.,December 31, 1999

> ```
> Once the agent has determined a list of potential handlers, it
> then starts at the beginning of the list of handlers and sends an
> ICMP_ECHOREPLY packet with an ID field containing the value 666
> and data field containing the string "skillz".
> ```

The only similarity is the value of the session ID, so nothing can actual be made of the 666 other

then to add to the suspicion of this detect.

For background information the following is returned from a Whois search:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:        193.134.0.0 - 193.135.255.255
netname:        CH-UNISOURCE-970528
descr:          ALLOCATED BLOCK
descr:          Provider Local Registry
descr:          Unisource Business Networks (Schweiz) AG
country:        CH
admin-c:        SIE3-RIPE
tech-c:         SIE3-RIPE
status:         ALLOCATED UNSPECIFIED
notify:         engineering@ip-plus.net
mnt-by:         RIPE-NCC-HM-MNT
mnt-lower:      CH-UNISOURCE-MNT
mnt-routes:     CH-UNISOURCE-MNT
changed:        hostmaster@ripe.net 19970528
changed:        hostmaster@ripe.net 19990421
changed:        hostmaster@ripe.net 20010208
changed:        hostmaster@ripe.net 20020419
changed:        hostmaster@ripe.net 20020423
changed:        hostmaster@ripe.net 20030507 # ch.unisource.andre
via https://lirportal.ripe.net
source:         RIPE

route:          193.134.0.0/15
descr:          Swisscom IP-Plus
descr:          UBN-CH-AGGR.5
origin:         AS3303
mnt-by:         CH-UNISOURCE-MNT
mnt-routes:     CH-UNISOURCE-MNT
mnt-routes:     AS6730-MNT
mnt-routes:     AS559-MNT
mnt-routes:     DTAG-RR
changed:        abuse@ip-plus.net 20030814
source:         RIPE

role:           Swisscom IP-Plus Engineering
address:        Genfergasse 14
address:        CH-3050 Bern
address:        Switzerland
fax-no:         +41 31 893 89 51
e-mail:         engineering@ip-plus.net
trouble:        abuse-email: abuse@ip-plus.net
admin-c:        RI192-RIPE
tech-c:         AC2332-RIPE
tech-c:         FB2326-RIPE
```

```
tech-c:         MM5218-RIPE
tech-c:         JAN27-RIPE
tech-c:         AF482-RIPE
nic-hdl:        SIE3-RIPE
remarks:        homepage: www.ip-plus.net
notify:         engineering@ip-plus.net
mnt-by:         CH-UNISOURCE-MNT
changed:        boogman@ip-plus.net 20030217
source:         RIPE
```

7. Evidence of active targeting.

Reviewing the total of five days of firewall logs revealed only one packet from this source. The firewall at the destination drops packets for port 139 and no hosts would respond to this stimulus. This is not active targeting.

```
#grep 193.135.0 ipcoplogs1205-1210.txt
00:31:13 INPUT IN=eth1 OUT= MAC=00:c0:df:e0:33:1a:00:e0:d0:13:4d:16:08:00
SRC=193.135.0.83 DST=10.0.0.2 LEN=40 TOS=0x00 PREC=0x00 TTL=48 ID=666
PROTO=TCP SPT=53 DPT=139 WINDOW=128 RES=0x00 SYN URGP=0
```

8. Severity.

To calculate the severity of this detect the formula, prescribed by the GIAC GCIA instructions, is used. From http://www.giac.org/GCIA_assignment.php :

severity = (criticality + lethality) (system countermeasures + network countermeasures)
Each value should be ranked on a scale from 1 (lowest) to 5 (highest).

Criticality is a measure of how critical the targeted system is.
Lethality is a measure of how severe the damage to the targeted system would be if the attack succeeded.
System countermeasures is a measure of the strength of the defensive mechanisms in place on the host itself.
Network countermeasures is a measure of the strength of the defensive mechanisms in place on the network.

This trace's severity:

Criticality = 0

There is no reachable host at port 139.

Lethality = 2

We have judged this to not being active targeting and most likely reconnaissance, a somewhat higher rating of 2 here is sufficient. In reality, it is probably a 1 or less.

System countermeasures = 5

Again, there is no reachable host at port 139.

Network countermeasures = 5

The firewall drops packets to and from port 139. All patches are up to date and the firewall is a stateful packet filter using iptables v1.2.7a. The only inbound traffic is to the web server ports 80 and 443. This traffic is forwarded to the firewall's DMZ interface, only.

Plugging the above values into our formula:

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

severity = (0 + 2) – (5 + 5) = -8

This rating will not require further investigation and does not require escalation to a CIRT.

9. Defensive recommendations.

The firewalls are configured to drop all traffic destined for port139, the Windows File and Printer Sharing port. Also recall that the Cisco DSL router's NAT tables where modified in order to collect a broader range of detects between the router and the firewall. So another recommendation for the perimeter defenses would be to use the capabilities of perimeter routers to allow only inbound traffic to the external firewall interface.

10. Multiple choice question.

This packet could be valid and may still have triggered an alert from the Snort rule if:

A) The destination port is an ephemeral port.
B) The destination port is 53.
C) This had been a UDP packet.
D) None of the above.

Answer B). If an external secondary DNS server is being hosted outside the network it would contact it's designated primary DNS server for zone file transfers. Zone transfers are done using TCP, rather then UDP and will be to port 53. The rule which triggered this matches on the packet being TCP, it's source being port 53, the destination less then 1023 and the SYN flag being set.

```
alert tcp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to
<1024"; flags:S; reference:arachnids,07; classtype:bad-unknown; sid:504;
rev:2;)
```

Ephemeral ports are those greater than 1023, so answer A) would not trigger this rule. Answer C) would not have matched the protocol of the rule and also would not trigger an alert.

Assignment Part 3 - Analyze This.

Executive Summary

The purpose of this analysis of GIAC University's intrusion detection logs, is to help the IT staff with an understanding of their intrusion detection information and to make recommendations based on the findings. These comments are to both provide an analysis of the event data gathered and also to help reduce the huge number of false alarms and other network noise. The data that is currently collected amounts to huge numbers of alerts and scans, and as such, it is virtually impossible to see the attacks and probes amongst this noise. Because of this large quantity of data, only the top events will be analyzed, and more importantly, the recommendations to reduce the noise in the logs can lead to a more manageable system. After implementing changes to reduce the logs to more accurate events, the University can then request another analysis and a much more in depth analysis can be done at that time.

The current intrusion detection installation of Snort contains many 'custom' rules and the alerts generated by this rule set appear to have many of the rule's parameters are 'global' in nature and should be refined. These parameters that are too general in nature, can lead to results such as the huge amount of data that is currently being collected. Another recommendation is that a current, and also a more standardized, set of rules be obtained from the Snort website and added to the current network sensors.

There is a huge community of IDS analysts that constantly monitor attacks in their systems. They openly discuss their findings in mail lists, news groups and on security websites. A down side of using many custom rules is that it becomes more difficult for the analyst to correlate events with security and intrusion detection sources outside of the University.

For the analysis of this network, the basic assumption was made that the intrusion detection system is within the network boundary and that there is some level of filtering at the perimeter. Recommendations regarding of the University's LAN/WAN infrastructure and the perimeter defenses will be made using this assumption. This will aid in the brevity of each section of the analysis and a summary will allow the executive reader to review only the recommendations, as they wish. The technical reader can follow as this report details the information used as the basis for the conclusions and recommendations.

Analyzed Logs.

The University's Central Computing department provided five consecutive days of logs, which where downloaded from http://www.incidents.org/logs/ . Despite being provided with this data, there was little other information regarding the infrastructure of the campus' network infrastructure. Three file types were used:

oos – out of spec packets.
alert – those packets which matched any of the Snort rule set used.
scans – detected network scanning activity

Filename:	Size:	Date:
oos_report_031217	773120	Sun Dec 21 05:02:40 2003
oos_report_031218	2657280	Mon Dec 22 05:01:55 2003
oos_report_031219	1064960	Tue Dec 23 05:01:49 2003
oos_report_031220	819200	Wed Dec 24 05:01:46 2003
oos_report_031221	32768	Thu Dec 25 05:01:12 2003
scans.031217.gz	29425807	Sun Dec 21 05:02:32 2003
scans.031218.gz	28868114	Mon Dec 22 05:01:50 2003
scans.031219.gz	25581614	Tue Dec 23 05:01:44 2003
scans.031220.gz	22096147	Wed Dec 24 05:01:43 2003
scans.031221.gz	24753500	Thu Dec 25 05:01:10 2003
alert.031217.gz	3640629	Sun Dec 21 05:01:56 2003
alert.031218.gz	3833481	Mon Dec 22 05:01:24 2003
alert.031219.gz	3320969	Tue Dec 23 05:01:21 2003
alert.031220.gz	2993101	Wed Dec 24 05:01:25 2003
alert.031221.gz	3086385	Thu Dec 25 05:00:52 2003

Table 1 - Five days of logs.

Replacing the first two octets of the real addresses with the text string, “MY.NET”, has sanitized the actual private network address. Snortsnarf (www.silicondefense.com/software/snortsnarf/) does not handle addresses sanitized in this way, so in order to use this analysis tool, the MY.NET in the alert and oos log files had been replaced with “10.0”. Later in this paper I will point out an issue with the scan files and the addresses of the private network from these files

Most Frequent Events.

First we can start by listing the most frequent alerts detected in the logs. The scans that where detected by the Snort pre-processor (spp_portscans) where seperated from the alerts logs. This

allows for cleaner reports and a better separation of alerts and scans discussed. The full list of all alerts can be found in Appendix B: SnortSnarf All Alerts.

Table 2 is the list of most frequent:

Signature (click for sig info)	# Alerts	# Sources	# Dests
Incomplete Packet Fragments Discarded	26282	83	59
10.0.30.3 activity	23990	107	1
10.0.30.4 activity	19654	287	1
EXPLOIT x86 NOOP	8501	333	130
connect to 515 from inside	4482	4	4
SMB Name Wildcard	4432	170	318
TFTP - Internal TCP connection to external tftp server	3783	6	6
High port 65535 udp - possible Red Worm - traffic	2507	114	119
NMAP TCP ping!	1741	177	51
ICMP SRC and DST outside network	1542	81	1392
High port 65535 tcp - possible Red Worm - traffic	1148	129	158

Table 2 - Top 10 alerts by quantity.

The following is a discussion of the Top Alerts: (by number of alerts)

1) Events Of Interest: Alert - Incomplete Packet Fragments Discarded

Severity: Noise **Reports: 26282**

Earliest such alert at **00:16:49**.334347 *on 12/17/2003*

Latest such alert at **22:55:13**.590371 *on 12/21/2003*

Incomplete Packet Fragments Discarded	83 sources	59 destinations
Priority: N/A	Classification: N/A	

12/17-02:19:15.830323 [**] Incomplete Packet Fragments Discarded [**] 10.0.21.67 -> 64.124.185.40
12/17-02:19:17.982067 [**] Incomplete Packet Fragments Discarded [**] 10.0.21.67 -> 64.124.185.40
12/17-02:19:18.384648 [**] Incomplete Packet Fragments Discarded [**] 10.0.21.67 -> 64.124.185.40

Text 1 - Sample Of Incomplete Packet Fragments Discarded

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
10.0.21.67	5556	5557	13	13
10.0.21.92	5232	5235	13	13
10.0.21.79	4859	4861	12	12
10.0.21.68	4689	4692	13	13
10.0.21.69	4162	4163	9	9
10.0.21.89	1332	1341	11	13
67.39.70.1	100	256	1	1

Table 3 - Top sources for EOI number1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
217.160.212.65	6523	6525	5	5
216.74.108.202	4506	4510	5	5
64.124.185.40	4355	4356	5	5
82.129.16.7	2647	2647	5	5
194.30.137.23	1369	1371	5	5
69.50.176.215	1060	1061	5	5
68.163.165.81	967	967	4	4
69.65.7.25	907	907	5	5
213.175.160.223	761	761	4	4
81.71.15.147	579	579	5	5

Table 4 - Top destinations for EOI number 1

```
12/17-02:19:15.830323 [**] Incomplete Packet Fragments Discarded [**] 10.0.21.67 -> 64.124.185.40
12/17-02:19:17.982067 [**] Incomplete Packet Fragments Discarded [**] 10.0.21.67 -> 64.124.185.40
12/17-02:19:18.384648 [**] Incomplete Packet Fragments Discarded [**] 10.0.21.67 -> 64.124.185.40
```

Text 2 - Sample Of Incomplete Packet Fragments Discarded

This alert is triggered by the Snort fragmentation preprocessor and is triggered because the fragment preprocessor will discard any fragmented packet that is not at least half full when the last fragment arrives. Not all the packets arrived and the stream could not be reassembled. This could be indicative of a fragmentation type attack and triggering the alert is described by Dragos Ruiu in this posting to the Snort User list:

http://www.geocrawler.com/archives/3/4890/2001/2/350/5151528/

```
FROM: Dragos Ruiu
DATE: 02/12/2001 13:52:30
SUBJECT: RE:  [snort-users] Incomplete Packet Fragments
Discarded?
This message is given by the defragmentation preprocessor when
packets bigger than 8k that are more than half empty when the
last
fragment is received are discarded.
This can be caused by:
- transmission errors
- broken stacks
- and fragmentation attacks
```

An early version of this preprocessor had problems and is explained in a posting by Martin Roesch (http://www.mcabee.org/lists/snort-users/Nov-01/msg00820.html)

```
That means that you're using the defrag preprocessor instead of
the
newer frag2 preprocessor and that you should switch to frag2.  :)
The defrag preprocessor had some fairly nasty failure modes and
has since been superceded by frag2, so I'd recommend using that
for now.
-Marty
```

Conclusions and recommendations:

Packet fragmentation is more common when there is a great many packets being transmitted, such as with file sharing programs. With known problems in early versions or the Snort defrag preprocessor, it would first be helpful to confirm if this preprocessor has been updated.

If it has not been, then the update should be done and this detects by this signature should be monitored. If the preprocessor has indeed be updated then the top sources should be checked for file sharing activities.

2) Events Of Interest: Alert - 10.0.30.3 activity

Severity: Noise **Reports: 23990**

Earliest such alert at **00:00:30**.563922 *on 12/17/2003*

Latest such alert at **23:20:08**.119743 *on 12/21/2003*

10.0.30.3 activity	107 sources	1 destinations
Priority: N/A	Classification: N/A	

Top Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
68.50.114.89	9462	10010	1	2
68.57.90.146	4178	4191	1	2
68.55.62.79	2119	2247	1	2
68.55.113.194	868	941	1	2
66.168.239.240	757	797	1	2
68.55.105.5	549	549	1	1
68.32.122.89	544	776	1	2
131.92.177.18	507	507	1	1

Table 5 - Top sources for EOI number2

```
12/19-16:55:10.601494 [**] 10.0.30.3 activity [**] 68.50.114.89:1168 -> 10.0.30.3:524
12/19-16:55:10.625523 [**] 10.0.30.3 activity [**] 68.50.114.89:1168 -> 10.0.30.3:524
12/19-16:55:10.630386 [**] 10.0.30.3 activity [**] 68.50.114.89:1168 -> 10.0.30.3:524
```

Text 3 - Sample Of 10.0.30.3 Activity

Destination	Port	# of Alerts
10.0.30.3	524	23662
10.0.30.3	2036	176
10.0.30.3	80	63
10.0.30.3	6129	18
10.0.30.3	4899	15

Table 6 - Top destination ports for EOI number2

Port 524 is NCP, the Novell Core Protocol. This is the communication port used by Novell and it can best be described as being similar to the NetBIOS ports used by Microsoft. The source

port should be an ephemeral port, greater then 1023. Novell will also use UDP on port 524 for time synchronization, along with UDP on port 123.

Checking into the port assignments from IANA, there was nothing listed for port 2036, but further research found the port related to services used by Novell software. Taken from support information at a Novell website we get an explanation of the probable service listening to port 2036.

http://developer.novell.com/research/appnotes/2003/june/02/a0306026.htm

```
"Using ZENworks for Servers 3 Server Management "

ZENworks for Servers 3 includes the capability to use the
RCONSOLEJ utility provided with NetWare to remotely manage a
server from the management console. This enables you to manage
NetWare servers based on your role in the organization and tasks
assigned to you.

You can use the ZENworks for Servers 3 remote management
capability to perform the following role-based remote management
tasks on a managed NetWare server:

...the TCP port number on which the agent will listen for
requests from RCONSOLEJ. (The default value is 2036.)
```

All of the connections to 10.0.30.3, port 2036, are from one external address, 68.50.114.89. This could be remote support by an employee or a contractor.

```
Search results for: 68.50.114.89
Comcast Cable Communications, Inc. JUMPSTART-1 (NET-68-32-0-0-1)
                                  68.32.0.0 - 68.63.255.255
Comcast Cable Communications, Inc. DC-4 (NET-68-50-0-0-1)
                                  68.50.0.0 - 68.50.255.255
```

Port 80, of course is the well known port for HTTP. Novell has many software packages that listen to port 80, including Apache.

Port 6129 is the default port used by Dameware remote administration software at http://www.dameware.com/ . Very recently, CERT has released a Vulnerability Note VU#909678, regarding a buffer overflow vulnerability in this software. http://www.kb.cert.org/vuls/id/909678

Port 4899 is known to be the default port used by Radmin from Famatech http://famatech.com/ . Radmin is a remote administration server used by help desks, etc. This software gives the user

the ability to remotely monitor, control and transfer files to and from his remote client via only password protected access.

Conclusions and recommendations:

This is a custom rule that was written to alert traffic destined to 10.0.30.3, all of which was from outside sources. It appears that this is a Novell server providing numerous services, many of which cannot be determined, in particular, the web applications running on port 80. Three of the top five alerts are quite possibly running various forms of remote access software, listening on ports 2036, 6129 and 4899.

To reduce the amount of noisy alerts, this rule should be modified to provide a more refined definition of alert parameters or additional rules could be written with this destination address but with more specific parameters. The creation of several rules would provide more granular results and alerts that are determined to be of little or no interest can be removed from the Snort rule sets. Further time should be spent on the remote access software, in particular, consolidation of these into one workable solution is recommended. If consolidation can be done, the firewall rule set can be modified to block the packets of the remote access software no longer used. Additionally, eliminating unnecessary software reduces exposure to un-patched vulnerabilities.

3) Events Of Interest: Alert - 10.0.30.4 activity

Severity: Noise **Reports: 19654**

Earliest such alert at **00:00:05**.876308 *on 12/17/2003*

Latest such alert at **23:46:40**.604260 *on 12/21/2003*

10.0.30.4 activity	287 sources	1 destinations
Priority: N/A	Classification: N/A	

```
12/21-22:52:11.230600 [**] 10.0.30.4 activity [**] 68.55.241.230:65149 -> 10.0.30.4:80
12/21-22:52:11.234462 [**] 10.0.30.4 activity [**] 68.55.241.230:65149 -> 10.0.30.4:80
12/21-22:52:11.255238 [**] 10.0.30.4 activity [**] 68.55.241.230:65149 -> 10.0.30.4:80
12/21-22:52:14.922760 [**] 10.0.30.4 activity [**] 68.55.241.230:65151 -> 10.0.30.4:51443
12/21-22:52:14.927467 [**] 10.0.30.4 activity [**] 68.55.241.230:65151 -> 10.0.30.4:51443
12/21-22:52:14.967112 [**] 10.0.30.4 activity [**] 68.55.241.230:65151 -> 10.0.30.4:51443
```

Text 4- Sample Of 10.0.30.4 Activity

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
68.55.241.230	4268	4268	1	1
68.55.62.244	2512	2512	1	1
67.20.160.15	587	587	1	1
68.50.114.89	548	10010	1	2
151.196.49.131	534	987	1	2
151.196.116.233	502	502	1	1

Table 7 - Top sources for EOI number3

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
10.0.30.4	19654	19654	287	287

Table 8 - Top destinations for EOI number3

Destination	Port	# of Alerts
10.0.30.4	80	8972
10.0.30.4	51443	8019
10.0.30.4	524	2447
10.0.30.4	2036	125
10.0.30.4	6129	22
10.0.30.4	4899	14

Table 9 - Top destination ports for EOI number3

IANA does not list an assignment of protocols, TCP nor UDP, to port 51443, but numerous Google sources show this port as being used for NetStorage.

```
NetStorage-Stateful TCP All  51443 ML2: NetStorage Stateful
```

NetStorage is a Novell Netware product along with iFolders. In simple terms, this is an Apache

web server that is typically configured to use HTTP over port 51080 and secure HTTPS over port 51443, that allows users to access resources from outside the company firewall. Referring back to Text 4, you can see that there is a series of connections from one source address to port 80. This is then followed by the same source connecting to port 51443. This is typical behavior of an HTTP GET that is redirected to a secure HTTPS connection.

Conclusions and recommendations:

This alert is very similar to the previous Top Five Alert, number 2. The discussion will be made here but much of it is a repeat of the Conclusions and Recommendations, from above. As before, this is a custom rule that was written to alert traffic destined to 10.0.30.3, all of which was from outside sources. It appears that this is a Novell server providing numerous services, many of which cannot be determined. In particular, the web applications running on port 80 and those which also are running a secure web connection on port 51443. Three of the top alerts are quite possibly running various forms of remote access software, listening on ports 2036, 6129 and 4899.

To reduce the amount of noisy alerts, this rule should be modified to provide a more refined definition of alert parameters or additional rules can be written with this destination address. The creation of several rules would provide more granular results and alerts that are determined to be of little or no interest can be removed from the Snort rule sets. Further time should be spent on the remote access software, in particular, consolidation of these into one workable solution is recommended. If consolidation can be done, then the firewall rules can be modified to block the packets of the remote access software no longer used. Additionally, eliminating unnecessary software reduces exposure to un-patched vulnerabilities.

4) Events Of Interest: Alert - EXPLOIT x86 NOOP

Severity: High **Reports: 8501**

Earliest such alert at **00:00:31**.758860 *on 12/17/2003*
Latest such alert at **23:47:20**.950235 *on 12/21/2003*

EXPLOIT x86 NOOP	333 sources	130 destinations
Priority: N/A	Classification: N/A	

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
66.149.34.140	4953	4964	24	26
212.202.57.13	786	786	5	5
211.108.90.6	481	481	1	1
212.202.55.36	371	371	5	5
158.109.232.129	319	319	20	20
131.118.254.130	124	137	1	1
61.248.245.128	91	91	1	1
65.203.33.194	75	75	3	3
62.234.24.4	52	52	30	30
211.213.173.27	50	50	1	1

Table 10 - Top sources for EOI number 4

The top two sources here are as follows:

```
Search results for: 66.149.34.140

OrgName:    EARTHLINK, INC
OrgID:      ERSD
Address:    1375 PEACHTREE ST, LEVEL A
City:       ATLANTA
StateProv:  GA
PostalCode: 30309
Country:    US
ReferralServer: rwhois://rwhois.admin.atl.earthlink.net:4321/

Search results for: 212.202.57.13

OrgName:    RIPE Network Coordination Centre
OrgID:      RIPE
Address:    Singel 258
Address:    1016 AB
City:       Amsterdam
StateProv:
PostalCode:
Country:    NL
ReferralServer: whois://whois.ripe.net
```

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
10.0.80.232	747	747	7	7
10.0.150.101	690	690	7	7
10.0.83.70	671	671	4	4
10.0.112.216	661	661	4	4
10.0.83.98	649	649	3	3
10.0.162.211	481	481	1	1
10.0.29.12	415	415	5	5
10.0.150.44	397	397	6	6
10.0.5.92	395	1498	5	7
10.0.29.8	351	351	5	5

Table 11 - Top destinations for EOI number4

Source IP	Destination IP	Destination Port	Total
66.149.34.140	10.0.112.226	80	4953
212.202.57.13	10.0.29.8	80	786
211.108.90.6	10.0.162.211	80	481
212.202.55.36	10.0.150.101	80	371
158.109.232.129	10.0.5.44	80	319
131.118.254.130	10.0.24.8	119	124
61.248.245.128	10.0.40.11	80	91
65.203.33.194	10.0.190.102	135	75
62.234.24.4	10.0.5.20	80	52
211.213.173.27	10.0.5.240	80	50

Table 12 - Top destination ports of EOI number 4

The NOOP signature detects on a series of numerous bytes that contain a 'no operation' code, sometimes referred to as a NOOP "sled". On the x86 architecture, this would be "0x90" hex code. A buffer overflow is vulnerability in a program where it does not check the input that it processes and an exploit takes its advantage from this by exceeding the allocated memory in the computer and overwriting into other memory space. This can often times crash the process and may even give the attacker access to the system command line. Another common ploy of a

buffer overflow attack is to pad the packet and then add a system command to the end of the packet. A no-operation is used to pad out a buffer for a buffer overflow attempt. The NOOP is often found while transferring large binary files, such as file sharing programs like KaZaA and Napster, etc.

From Table 11 we get the destination ports for the top alert generating sources. From this you can see that for the majority, the destination is port 80. Without full packet dumps of these alerts it cannot be certain what is going on with this alert. Viewing a full packet would let us know which, if any, are malicious attempts.

Note however that there are numerous alerts with destination ports of 135 and 119.

```
12/17-00:04:49.554916 [**] EXPLOIT x86 NOOP [**] 131.118.254.130:2336 -> 10.0.24.8:119
12/17-00:55:15.258464 [**] EXPLOIT x86 NOOP [**] 65.203.33.194:16310 ->
10.0.190.102:135
```

Port 119 is the usenet's NNTP service. A search of Domain Name Service (DNS) says 131.118.254.130 is news.ums.edu. Based on the name of this host it's pretty easy to assume this is a usenet server. Port 135 is the end-point mapper in the Windows operating system. When trying to connect to a service, you go through this mapper to discover where it is located. The process works the same as on the UNIX RPC portmapper. One difference is that a lot of services run on top of named pipes, which don't have a specific port. An ARIN search for the source IP returned this:

```
Search results for: ! NET-65-203-33-192-1

CustName:   Commonwealth Central Credit Union
Address:    1651 N 1st Street
City:       San Jose
StateProv:  CA
PostalCode: 95112-4507
Country:    US
RegDate:    2002-09-30
Updated:    2002-09-30
NetRange:   65.203.33.192 - 65.203.33.255
CIDR:       65.203.33.192/26
```

Conclusions and recommendations:

Searches of both Snort and the arachNID rules at White Hats turned up no message that exactly matched the message found in these alerts, so we may assume that this is a custom rule. The closest match is the arachNID rule, IDS181 "SHELLCODE-X86-NOPS". This rule does not specifically look for telltale signs of maliciousness such as found in other rules that trigger on

attempts to execute shell code. In the rule description for IDS 181 at http://www.whitehats.com/ the 'False Positive' section states that:

```
"Since all network traffic is watched, it is possible this sequence may occur
in any binary file transmission, and not be a part of an overflow attempt.
Confirm by looking at the packet trace generated by this alert."
```

The greatest majority of these are connection to the HTTP port 80. Without knowing the exact rule that triggered the alerts we cannot conclude if there were any malicious attacks. This should be investigated further by gathering complete packet dumps and examining them for exploits. After which, this rule should be modified to look for more explicit details of known exploits. Rather then attempt such a rule modification, it would be a better decision to remove this rule and add existing Snort rules that detect specific malicious activity. The use of the university's web servers for transfer of binary files should also confirmed. From searches of the log files, there are many external sources connecting to internal addresses on port 135. This access is something that must be stopped IMMEDIATELY at all external firewalls.

5) Events Of Interest: Alert - connect to 515 from inside

Severity: Noise **Reports: 4482**

Earliest such alert at **16:19:49**.683252 *on 12/18/2003*

Latest such alert at **20:48:12**.449473 *on 12/19/2003*

connect to 515 from inside	4 sources	4 destinations
Priority: N/A	Classification: N/A	

```
12/18-16:19:49.683252 [**] connect to 515 from inside [**] 10.0.162.41:721 -> 128.183.110.242:515
12/18-16:20:01.701164 [**] connect to 515 from inside [**] 10.0.162.41:721 -> 128.183.110.242:515
12/18-16:20:32.747434 [**] connect to 515 from inside [**] 10.0.162.41:721 -> 128.183.110.242:515
12/18-16:21:18.816947 [**] connect to 515 from inside [**] 10.0.162.41:721 -> 128.183.110.242:515
```

Text 5- Sample Of connect to 515 from inside

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
10.0.162.41	4469	4470	1	2
10.0.97.66	7	7	1	1
10.0.60.16	3	9	1	2
10.0.60.38	3	8	1	3

Table 13 - Top destinations for EOI number5

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
128.183.110.242	4469	4470	1	2
192.168.0.14	7	7	1	1
141.157.62.228	3	4	1	1
66.160.63.18	3	6	1	1

Table 14 - Top sources for EOI number5

Port 515 is the IANA assigned port for printer services and is typically open to a printer daemon listening on this port, for example, it could be the lpd daemon in Unix systems. Per the RFC1179, it states that connections to the printer daemon listening on port 515 must have a source port in the range of ports 721-731. The reason for this is the UNIX concept of privileged ports below port 1024, where these processes are to be run by an account equivalent to User ID 0, or root level. Under normal conditions, the ordinary user should not be able to originate traffic from ports 721-731 to port 515.

Conclusions and recommendations:

Further investigating of the source port in these alerts found that only host 10.0.162.41 had the source port specified by the RFC1179, port 721.

Source IP	Src Port	Destination IP	Dst Port	Count
10.0.97.66	3160	192.168.0.14	515	7
10.0.60.38	50602	141.157.62.228	515	3
10.0.60.16	40844	66.160.63.18	515	3

Table 15 - Sources not conforming with RFC1179

Several worms and trojans are in the wild for LPR vulnerabilities: IDS457 "LPRNG-REDHAT7-OVERFLOW-SECURITY.IS" http://www.whitehats.com/info/IDS457 CERT Incident Note IN-2001-01 Widespread Compromises via "ramen" Toolkit http://www.cert.org/incident_notes/IN-2001-01.html and also “Vulnerability Note VU#382365 - LPRng can pass user-supplied input as a format string parameter to syslog() calls” http://www.kb.cert.org/vuls/id/382365 and CVE-2000-0917 “Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands”.

These addresses, in Table 14, should be checked for any anomalies and trojans. There was no correlation found for the combination of these ephemeral source ports and port 515, but these connections do not comply with RFC 1179 and are therefore suspicious.

All of the alerts from 10.0.162.41 do follow the RFC 1179 standard in that the source port is 721 and the destination is to port 515. The purpose for connecting to an external print server is not known and is somewhat out of the ordinary. The policy of connecting to any external print server should be confirmed and the policy documentation updated.

Most Frequent Scan Events.

This analysis is of the scan log files collected over the five chosen days. The top two types of scans, SYN scans and UDP, are well known as being false alarms. Of the total number of scans, these two account for 99.98% of the total scans detected! Even knowing the high probability of the false alarms there are just too many to ignore them all. Looking at the destination ports in Table 20, we can get a better idea of what scans are of high interest and those that are merely noise.

Table 16 and Table 17 list the most active sources of scan sources and destinations. Note should be taken regarding the 130.85.0.0/16 address range. It appears that these are likely the internal private networks address and the scan log files provided had not been sanitized. The alert and OOS files had used the “MY.NET.” as the first two octets to hide the actual address. In order to use Snortsnarf on the alert and OOS file, the MY.NET. was replaced with 10.0. The network, which appeared to be the un-sanitized addresses, in the scan files was not modified to hide the true address. This will not affect the analysis of the University's network, but it is a point that needs to be known in order to avoid confusion.

TCP flags	Count
******S*	11673034
12****S*	6025
*2*A****	825
	782
***A*R*F	649
None	368
*******F	353
********	304
*2*A**S*	103
**U*P*S*	72

Table 16 - Most frequent TCP flag combinations

Scan type	Count
SYN scan (Externally-based)	11679065
UDP scan (Externally-based)	5729406
UNKNOWN scan (Externally-based)	1145
INVALIDACK scan (Externally-based)	910
NULL scan (Externally-based)	541
NOACK scan (Externally-based)	451
FIN scan (Externally-based)	356
VECNA scan (Externally-based)	88
NMAPID scan (Externally-based)	10
SPAU scan (Externally-based)	5
FULLXMAS scan (Externally-based)	4
XMAS scan (Externally-based)	3
21 scan (Externally-based)	2
SYNFIN scan (Externally-based)	2

Table 17 - Top scan type by name

Richard Baker, in his practical, listed types of scans and their TCP flags set. This list is taken from the SANS Institute Track 3 – Intrusion Detection In-Depth course.

```
UNKNOWN     --          Ref. spp_portscan.c source code
INVALIDACK  --          ACK set, not normal, no SPAU or FULLXMAS
NULL        --          None of SFRPAU
NOACK       --          A flag is missing
FIN         --          F flag
VECNA       --          One of the following: P, U, PU, FP, FU
NMAPID      --          SFPU flags
SPAU        --          SPAU flags
```

```
FULLXMAS     --          SFRPAU flags
XMAS         --          FPU flags
SYNFIN       --          SF flags
```

From the scan log files we get some insight into what scans where taking place, the source, the destination, and the scanned ports.

Source	# of scans
130.85.1.3	3332222
130.85.111.72	2676071
130.85.162.92	2369384
130.85.84.194	2368054
130.85.163.107	2363970
130.85.80.243	600486
130.85.1.4	518747
130.85.110.72	368463
130.85.153.37	264066
130.85.53.225	232693
130.85.185.13	207184
130.85.163.76	196132
130.85.70.225	154735
130.85.82.104	98162
130.85.70.207	81930
130.85.100.230	54569
130.85.112.159	52362
130.85.53.31	45129
207.172.214.7	41329
130.85.75.111	33018

Table 18 - Top scan sources

Destination	# of scans
192.26.92.30	60980
203.20.52.5	52783
69.6.61.11	49826
192.5.6.30	48179
204.29.185.132	44877
192.55.83.30	43625
69.6.25.84	43070
69.6.25.125	42861
69.6.61.10	38940
69.56.32.4	38424
128.194.254.5	36349
128.194.254.4	36214
165.230.209.227	33494
216.109.116.17	32105
69.20.36.152	30713
192.48.79.30	30333
69.20.36.154	29734
209.92.188.201	27219
131.118.254.33	26364
206.67.234.112	25721

Table 19 - Top scan destinations

Destination port	Count
135	10408395
53	3827953
80	283990
6129	230216
6257	184488
25	176061
4899	93040
6346	88730
21	72009
4000	57680
22321	53485
4662	39594
7674	33936
1257	33230
3389	28630
41170	22849
5900	20987
123	20031
113	19873
23	19199

Table 20 - Top Destination Ports

Table 21 and Table 22 differ from the data in Table 20, in that they are the breakout from the total of all scans for SYN scans and UDP scans, respectively.

Events Of Interest: SYN Scans

Severity: Noise **Reports: 11679065**

Info	Destination Port	Count
SYN scan (Externally-based)	135	10408269
SYN scan (Externally-based)	80	281930
SYN scan (Externally-based)	6129	230165
SYN scan (Externally-based)	25	175997
SYN scan (Externally-based)	4899	93015
SYN scan (Externally-based)	21	71771
SYN scan (Externally-based)	4000	57581

Table 21 - Top Destination ports for SYN scans

With this many SYN scans, there is no reasonable way to handle so many alerts with any sort of

detailed analysis. By far the vast majority of these detects is normal network activity. Besides the top 3 destination ports for NetBIOS, DNS and HTTP ports, there is a great deal of packets that are destined for ports known for remote access software packages. These are ports 6129 (Dameware), 4899 (Radmin) and 3389 (Microsoft Terminal Server). Other top destination ports that are likely being used for P2P file sharing or chat and messaging are ports 6257 (WinMX), 6346 (Gnutella) and 4000 (ICQ).

Conclusions and recommendations:

An in depth review of the perimeter filtering should be conducted; soon! Making certain that the proper filtering of both inbound and outbound activity is in place. Settings of the Snort scan pre-processor should be adjusted upward to reduce the number of false positives, as it is important that this scan noise be reduced drastically. Once this is done, then it will be possible to monitor for this pre-attack type of scan activity.

Events Of Interest: UDP Scans

Severity: Noise **Reports: 5729406**

Info	Destination Port	Count
UDP scan (Externally-based)	53	3827860
UDP scan (Externally-based)	6257	184485
UDP scan (Externally-based)	6346	62031
UDP scan (Externally-based)	22321	53483

Table 22 - Top Destination ports for UDP scans

There are far fewer UDP scans then SYN scans but a sound analysis is nearly impossible. The largest number of scans reported, over 3.8 million, are to destination port 53, DNS. DNS lookup requests are made with UDP packets so most, if not nearly all, of this is legitimate network traffic. The next two most scanned ports are port 6257 (WinMX) and port 6346 (Gnutella). Port 22321 is the known listener port for the Backdoor Q trojan.

Conclusions and recommendations:

An in depth review of the perimeter filtering should be conducted, soon. Follow the same recommendations that where described in the previous section for SYN scans as for these UDP scans. The destination host(s) of the Backdoor Q scans is all external computers and the source computers should be investigated to determine if they have been compromised. If these computers are compromised and if they are assets of the University, they must be taken off-line

and the necessary steps taken to remove the trojan. If they are compromised and they are not a University asset, they should be removed from the network until such time as they can be proved to no longer contain any trojans. These are the sources that are scanning for port 22321:

Source IP	Destination port	Count
130.85.53.31	22321	28755
130.85.97.19	22321	12185
130.85.97.212	22321	7618
130.85.97.77	22321	4925
130.85.60.38	22321	1
130.85.60.16	22321	1

Table 23 - Hosts scanning for Backdoor Q hosts

Events Of Interest: Other Scans

Severity: Noise **Reports: 3517**

UNKNOWN scans are triggered by the Snort scan pre-processor.

INVALIDACK scans are packets in which the ACK bit is set without being in combination with any other valid bits.

NULL scans are often used for reconnaissance. A host that is not listening on the destination port of the NULL scan packet will respond with a reset packet.

NOACK scans have flags other then the ACK bit set but which are expected to have the ACK bit to be a valid packet. These are scans, which may be correlated to other scans as pre-attack activity.

FIN scans are FIN only flags set in the packets. These were often found in activity of file sharing such as Gnutella.

VECNA is another scan packet around which is an invalid set of flags in which the ACK bit is not set. Like the FIN scans, this is often seen in file sharing programs KaZaA.

Conclusions and recommendations:

The sum total of these of the other scans, in Table 17, is much lower then the SYN and UDP scans. Most of these remaining scans are very common activity of Nmap and Queso scanning tools and are very indicative of reconnaissance scan activity. It would be just too resource intensive for follow-up on ALL of the scans, but they are good correlation information in case there is an incident later. This is mostly network noise and can be reduced by filtering at

firewalls and perimeter routers.

Events Of Interest: OOS

Out of specification packets, or OOS, are packets that do not conform to normal specifications. These are generally caused by several reasons:

- Packet corruption.
- Use of the ECN reserved bits.
- Crafted packets for scanning and/or fingerprinting.

The following are the out of specification packets that where received:

Flag	Count
12****S*	1462
********	66
12***R**	3
****P***	3
*2U**RSF	2
1**AP*SF	1

Table 24 - TCP flags in OOS packets

Table 25 and Table 26 show the OOS packet counts by both the source and destinations. As you can see, from Table 25, that no OOS packets were sourced from the internal network.

Source IP	Count
194.67.70.10	302
66.225.198.20	114
209.218.69.253	101
67.114.19.185	92
66.30.247.121	69
68.122.128.111	63
193.41.64.2	46
207.228.236.26	35
212.36.16.66	26
62.58.92.114	23

Table 25 - Top sources of OOS packets

Destination IP	Count
10.0.12.6	403
10.0.66.42	302
10.0.24.44	266
10.0.185.13	109
10.0.60.16	103
10.0.12.4	64
10.0.24.34	52
10.0.29.66	34
10.0.100.165	23
10.0.70.164	21

Table 26 - Top destinations of OOS packets

Events Of Interest: OOS – Packet Corruption

Severity: Low **Reports: Uncertain**

Packet corruption is fairly rare, but yet if you consider that the Internet is a somewhat unstable network, then packet corruption is inevitable.

Events Of Interest: OOS - ECN Bits Set

Severity: Noise **Reports: 1462 times**

The second possibility is the use of the reserved ECN bits. This is a fairly new implementation of the TCP header flag where these bits are set in the initial SYN packet and are used to perform a network congestion check. Looking at the 12****S* scans these are likely be legitimate SYN packets that are being transmitted and/or retransmitted by network equipment that is using the features of ECN as a congestion control mechanism. The use of the ECN bits requires an ECN enabled sender, receiver and router(s). The TCP sender sets both bits with in the SYN packet but any ECN enabled device can notify other ECN devices of increasing network congestion. Using this same discussion of the 12****S* packets, we can also say that the 12***R** packets are also legitimate and not out of specification. An RST without an ACK is often caused by a 'connection refused' message for a port not listening.

Top External Sources.

The most active source IPs is shown. The number of alerts from that IP as the source determines rank. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Source IP	# Signatures triggered	Destinations involved
rank #1	10010 alerts	68.50.114.89	2 signatures	10.0.30.3, 10.0.30.4
rank #2	4964 alerts	66.149.34.140	3 signatures	(26 destination IPs)
rank #3	4268 alerts	68.55.241.230	1 signatures	10.0.30.4
rank #4	4191 alerts	68.57.90.146	2 signatures	10.0.30.3, 10.0.30.4
rank #5	2512 alerts	68.55.62.244	1 signatures	10.0.30.4
rank #6	2247 alerts	68.55.62.79	2 signatures	10.0.30.3, 10.0.30.4
rank #7	1382 alerts	69.10.132.121	2 signatures	10.0.42.3
rank #8	1101 alerts	67.20.173.236	12 signatures	10.0.5.92
rank #9	987 alerts	151.196.49.131	2 signatures	10.0.30.3, 10.0.30.4
rank #10	941 alerts	68.55.113.194	2 signatures	10.0.30.3, 10.0.30.4
rank #20	915 alerts	10.0.163.76	1 signatures	(80 destination IPs)

Table 27 - Top external sources

Top External Talker With Suspicious Alerts

Alerts found using input module SnortFileInput, with sources:

../logs/alert_0312-all_nospp

Earliest: **22:18:39**.470989 *on 12/19/2003*
Latest: **22:43:50**.089472 *on 12/19/2003*

12 different signatures are present for *67.20.173.236* as a source

1 instances of Probable NMAP fingerprint attempt
1 instances of TCP SMTP Source Port traffic
1 instances of High port 65535 udp - possible Red Worm - traffic
1 instances of PHF attempt
1 instances of Possible wu-ftpd exploit - GIAC000623
1 instances of EXPLOIT NTPDX buffer overflow
1 instances of DDOS mstream client to handler
2 instances of Incomplete Packet Fragments Discarded
3 instances of Attempted Sun RPC high port access
4 instances of Null scan!
4 instances of TFTP - External TCP connection to internal tftp server
1081 instances of NMAP TCP ping!

There are 1 distinct destination IPs in the alerts of the type on this page:

10.0.5.92

ARIN information for this source address:

Search results for: 67.20.173.236

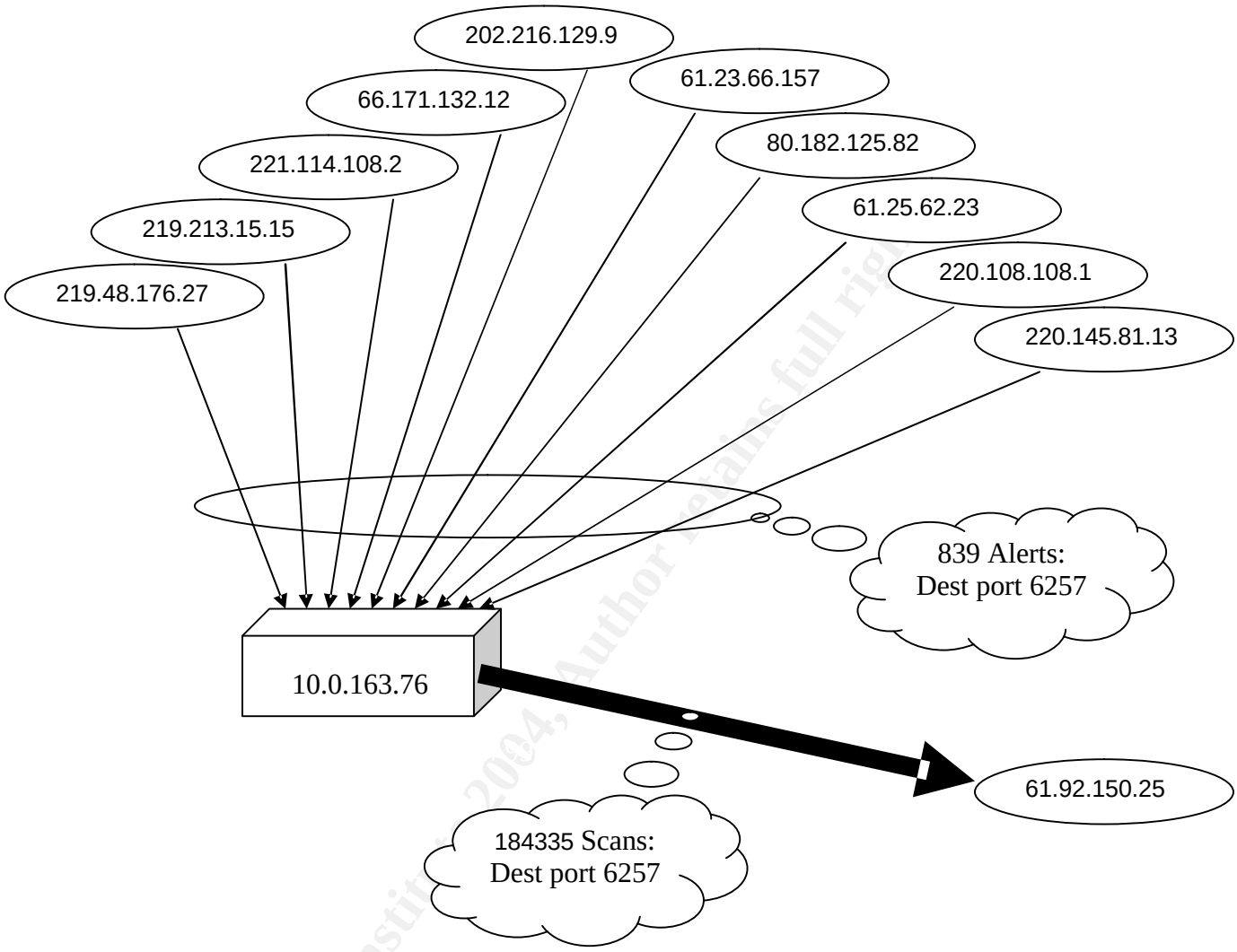
```
Adelphia Cable Communications ADELPHIA-CABLE-5 (NET-67-20-0-0-1)
                                  67.20.0.0 - 67.23.255.255
Adelphia 67201600-Z5 (NET-67-20-160-0-1)
                                  67.20.160.0 - 67.20.191.255

# ARIN WHOIS database, last updated 2004-01-24 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.

AbuseHandle: IPE-ARIN
AbuseName:   Internet Policy Enforcement
AbusePhone:  +1-866-473-2909
AbuseEmail:  abuse@adelphia.net
```

Conclusions and recommendations:

This source has been extremely active with attacks to 10.0.5.92 on the internal network. If this

user's IP address is dynamically assigned then a firewall rule to block this specific address will become useless, but for now this address should be blocked at the perimeter firewall. The logs should be monitored for a changing source of this activity. The user should be reported to the abuse contact in the above ARIN Whois.

Link Diagram:



This diagram provides a visual of the WinMX traffic, port 6257, between external sources and the internal computer most involved in file sharing. A review of the University's acceptable use policy should be reviewed with regards to fiel sharing. If the policy forbids this activity, then this computer should be taken off the network as soon as possible. The perimeter firewalls rules should be modified to prevent future file sharing activity.

Resources.

Beardsley, Tod. “Intrusion Detection and Analysis: Theory, Techniques, and Tools.” May 8, 2002. URL: http://www.giac.org/practical/Tod_Beardsley_GCIA.doc. (Jan. 4, 2003)

Bell, Mike. “GCIA Practical.” Jan. 2001. URL: http://www.giac.org/practical/Mike_Bell_GCIA.doc. (Jan. 4, 2003)

“defcon" <ion.storm@verizon.net>. “More XDCC/DDOS bots found on efnet #x-dcc!” May 7, 2002. URL: http://www.theorygroup.com/Archive/Unisog/2002/msg00618.html. (Jan. 4, 2003)

Ellis, Joe. “GCIA Practical Assignment, v3.0 Intrusion Detection In Depth.” May 14, 2002. URL: http://www.giac.org/practical/Joe_Ellis_GCIA.doc. (Jan. 4, 2003)

Garris, John. “Level II Intrusion Detection GCIA Practical Assignment – Version 2.7.” URL: http://www.giac.org/practical/John_Garris_GCIA.doc. (Jan. 4, 2003)

Gordon, Les. “Intrusion Analysis - The Director's Cut!.” Nov. 22, 2002. URL: http://www.giac.org/practical/GCIA/Les_Gordon_GCIA.doc. (Jan. 4, 2003)

Hoover, James. “GCIA Practical Version 3.0.” Dec. 23, 2000. URL: http://www.giac.org/practical/James_Hoover_GCIA.doc. (Jan. 4, 2003)

Internet Security Systems URL: http://www.iss.net

Jongen, Martijn. “Portnumbers.” 2002. URL: http://www.martijnjongen.com/portnumbers.htm. (Jan. 4, 2003)

Junginger, Jeremy. “Security Incidents: RE: UDP port 22321.” Incidents@insecure.org Sept. 9, 2002. URL: http://lists.insecure.org/lists/incidents/2002/Sep/0055.html.

Kuethe, Chris. “Chris Kuethe: GCIA Practical Assignment.” URL: http://www.giac.org/practical/chris_kuethe_gcia.html. (Jan. 4, 2003)

Lutton, R. “Cisco 675 Setup for QWEST.net - PPP mode.” URL: http://www.users.qwest.net/~rlutton/ADSL/NAT_Settings.html. (Jan. 4, 2003)

Martin, Daniel. “Re: Spoofed SMB name wildcard probes.” Incidents @ Security Focus.com mailing list. May 4, 2001. URL: http://cert.unistuttgart.de/archive/incidents/2001/05/msg00041.html. (Jan. 3, 2003)

Melhber, D. “SnortSnarf Alert Page.” March 20,2002. URL: http://www.nd.edu/~dmehlber/ids/html3/html/129/252/127/dest129.252.127.65.html. (Jan. 4, 2003)

Miller, Toby. “ECN and it’s impact on Intrusion Detection.” Global Incident Analysis Center - Special Notice -. 2000. URL: http://www.sans.org/y2k/ecn.htm. (Jan. 4, 2003)

Multiple Authors. “1214 Scans” Incidents.org Intrusions Index (by Thread). Feb. 4, 2002. URL: http://www.incidents.org/archives/intrusions/thrd118.html. (Jan. 4, 2003)

Multiple Authors. “SNORT FAQ” v 1.14. March 25, 2002. URL: http://www.snort.org/docs/4.17. (Jan. 3, 2003)

Online Discussion. “Back Door Q Access?.” Security Focus Online. May 4, 2001. URL: http://online.securityfocus.com/archive/75/182244/2002-11-04/2002-11-10/1. (Jan. 3, 2003)

Riddell, Trenton. “GCIA Certification Practical Assignment.” March 6, 2002. URL: http://www.giac.org/practical/Trenton_Riddell_GCIA.doc. (Jan. 3, 2003)

Ryan, John. "Security Logs and Checkpoint Firewall-1." Firewalls & Perimeter Protection. June 4, 2001. URL: http://www.sans.org/rr/firewall/logs.php. (Jan. 3, 2003)

Stearns, William. “Re: Scan Analysis” Intrusions@incidents.org May 31, 2002. URL: http://www.incidents.org/archives/intrusions/msg14734.html. (Jan. 4, 2003)

Stevens, W. Richard, “TCP/IP Illustrated, Volume 1, The Protocols” 1994, Addison-Wesley

Toren, Montgomery. “Intrusion Detection In Depth.” V 3.0. Feb. 11,2002. URL: http://www.giac.org/practical/Montgomery_Toren_GCIA.doc. (Jan. 4, 2003)

Wilkinson, Michael. “GCIA Practical for SANS Darling Harbour.” URL: http://www.giac.org/practical/michael_wilkinson_gcia.doc. (Jan. 4, 2003)

Williams, Al. “SANS GCIA Practical ver. 3.3.” URL: http://www.whitehats.ca/main/members/Herc_Man/Files/Al_Williams_GCIAPractical.pdf. (Aug. 2002)

Appendix A: Tools and Methods

- Microsoft Windows 2000
- Microsoft Word 2000
- SnortSnarf
- OpenOffice 1.1.0
- RedHat Linux version 8.x
- MySQL 3.23
- SQLyog 3.63
- Perl 5.8
- sed, awk, grep

After downloading the log files and unzipping them, I looked at taking several different approaches and which tools could best help at getting the logs in a format that is suitable for this analysis. These files are quite large and there where some lines that had been corrupted. Using the command, `grep -v ^12 <filename>`, turned up quite a few lines in the files that did not start with the correct date format. In this command, the results where all the lines that did not begin with '12', the number of the month (December) in the timestamp field.

The first steps where to remove these damaged lines from the files. The five files, of each type, where first concatenated into one file with the command, `cat alerts.0312* >> alerts.temp`. Then this temporary file was 'grepped' to remove those lines which where corrupt and the result was redirected to another file: `grep ^12 alerts.temp > alerts.0312`.

Many other GCIA practicals discuss using SnortSnarf as the analysis tool to use. SnortSnarf does not handle the use of 'MY.NET' that was used to sanitize the logs of the actual private network address. So, that text string was replaced with the '10.0. network address. This was easily done with the command: `sed s/'MY\.NET\./10\.0\./g alerts.0312 > alerts.0312.10`.

SnortSnarf was installed on an older Dell server with 4 Pentium Pro processors and 2 GB of memory. The program would handle the alert files, that where created as discussed in the previous paragraph. When SnortSnarf was run with the scan files, the program would fail. SnortSnarf would sometimes run for many hours before terminating with a dump error. After a great deal of time spent looking for a solution to this problem, numerous Google searches found

that many GCIA students had the very same difficulty with very large files. In order to approach this analysis with different tools, the results from the SnortSnarf where used on the alert files.

This left a problem with how to handle the large scan files. Again, researching many past GCIA practicals, it became apparent that a database would be the best approach. A current build of MySQL was installed on the Dell server, along with any required client and library programs.

The next issue was getting the log data into the database tables. This required that the files be parsed and convert to a 'comma delimited' format or 'csv' file. After concatenating and cleaning up the logs files needed to be converted to the csv format for importing into the MySQL database. Using the Perl code from Al Williams practical as a basis, the following four Perl scripts where customized as needed for this analysis.

alert2csv.pl

```
#!/usr/bin/perl
# Convert alert.DDDDDD files to csv
while(<>) {
next unless m/^\d/;
next if m/spp_portscan/;
chomp;
($date_time,$alert,$addrs) = split(/\s+\Q[**]\E\s+/);
($source, $dest) = ($addrs =~ m/(.*)\s+->\s+(.*)/);
($date,$time) = split(/-/,$date_time);
($source_ip, $source_port) = split(/:/,$source);
($dest_ip, $dest_port) = split(/:/,$dest);
print "$date,$time,$alert,$source_ip,$source_port,$dest_ip,$dest_port\n";
}
```

alertspp2csv.pl

```
#!/usr/bin/perl
# Convert alert.DDDDDD files to csv
while(<>) {
next unless m/^\d/;
chomp;
($date_time,$alert) = split(/\s+\Q[**]\E\s+/);
($date,$time) = split(/-/,$date_time);
($alert,$sourceip,$tail) = split(/:/,$alert);
($sourceip,$comment) = split(/\(/,$sourceip);
@srcip = split(/\s/,$sourceip);
$srcip = @srcip[(scalar @srcip)-1];
print "$date,$time,$alert,$srcip,$comment,$tail\n";
}
```

oos2csv.pl

```
#!/usr/bin/perl
# Convert oos.DDDDDD files to csv
while (<>)
{
#next unless m/^[A-Z]/;
chomp;
($month,$datetime,$brak1,$info,$brak2,$source,$dir,$dest,$proto,@flags) =
split;
($date,$time) = split(/-/,$datetime);
($src_ip,$src_port) = split(/:/,$source);
($dst_ip,$dst_port) = split(/:/,$dest);
$flags = (@flags[5] eq "DF") ?  @flags[6] : @flags[5];
#print "$flags\n";
print
"$month,$date,$time,$info,$src_ip,$src_port,$dst_ip,$dst_port,$proto,$flags\n
";
}
```

scans2csv.pl

```
#!/usr/bin/perl
# Convert scans.DDDDDD files to csv
while (<>)
{
next unless m/^[A-Z]/;
chomp;
($month,$day,$time,$source,$dir,$dest,$proto,$flags) = split;
#$month = $months{$month};
#$date = sprintf("%02d/%02d", $month, $day);
($src_ip,$src_port) = split(/:/,$source);
($dst_ip,$dst_port) = split(/:/,$dest);
print
"$month,$day,$time,$src_ip,$src_port,$dst_ip,$dst_port,$proto,$flags\n";
}
```

The csv formatted output from each of these scripts was then loaded into a MySQL database named ‘gcia’. This command is an example of how one file was loaded. The other three log files where loaded in with a similar command. Each type of logs where loaded into a separate table within the database.

```
LOAD DATA INFILE 'alerts.0312' INTO TABLE alerts > FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';
```

The use of MySQL and SQL queries provided a great deal of flexibility in viewing the data. It is not possible to list all the queries made against the data, but the following is a sample of a few of these:

```
select count(*) from alerts;
```

```
select count(distinct srcip) from alerts;
select count(*) from spp;
select count(distinct srcip) from spp;
select srcip, count(*) as count from alerts group by srcip order by count
desc limit 25;
select dstip, count(*) as count from alerts group by dstip order by count
desc limit 25;
select count(*) as count, dstip from alert group by dstip order by count desc
limit 25;
select src, count(*) as count from alerts where src like "10.0." group by src
order by count desc limit 25;
select count(srcip) as count from alerts where srcip like "10.0.";
select count(distinct attack) as count from alerts;
select count(distinct attack) as count from spp;
select srcip, dstip, count(distinct srcip) as count from alert group by dstip
order by count desc limit 25;
select srcip, dstip, count(distinct dstip) as count from alert group by srcip
order by count desc limit 25;
```

Appendix B: SnortSnarf All Alerts

100832 alerts found using input module SnortFileInput, with sources: ../logs/alert_0312-all_nospp

Earliest alert at **00:00:05**.876308 *on 12/17/2003*

Latest alert at **23:47:20**.950235 *on 12/21/2003*

Signature (click for sig info)	# Alerts	# Sources	# Dests
10.0.30.3 activity	23990	107	1
10.0.30.4 activity	19654	287	1
EXPLOIT x86 NOOP	8501	333	130
connect to 515 from inside	4482	4	4
SMB Name Wildcard	4432	170	318
TFTP - Internal TCP connection to external tftp server	3783	6	6
High port 65535 udp - possible Red Worm - traffic	2507	114	119
NMAP TCP ping!	1741	177	51
ICMP SRC and DST outside network	1542	81	1392
High port 65535 tcp - possible Red Worm - traffic	1148	129	158
Null scan!	711	66	42
External RPC call	542	1	252
TCP SRC and DST outside network	256	21	53
SUNRPC highport access!	234	20	21
Possible trojan server activity	204	41	42
[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	189	54	51
FTP passwd attempt	116	87	2
SMB C access	90	43	3
[UMBC NIDS] External MiMail alert	67	43	1
EXPLOIT x86 setuid 0	56	41	39
TCP SMTP Source Port traffic	41	4	3
RFB - Possible WinVNC - 010708-1	35	17	15
EXPLOIT x86 setgid 0	34	21	21
FTP DoS ftpd globbing	29	9	1
DDOS shaft client to handler [arachNIDS]	27	1	1
TFTP - External TCP connection to internal tftp server	15	3	4
EXPLOIT NTPDX buffer overflow	14	8	7
EXPLOIT x86 stealth noop	11	8	8
EXPLOIT x86 NOPS	10	2	2
[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	9	1	2
TFTP - Internal UDP connection to external tftp server	9	6	7
External FTP to HelpDesk 10.0.70.50	6	5	1
External FTP to HelpDesk 10.0.70.49	5	4	1
DDOS mstream client to handler [CVE] [arachNIDS]	5	3	3
IRC evil - running XDCC	5	1	2

Signature (click for sig info)	# Alerts	# Sources	# Dests
Attempted Sun RPC high port access	5	2	2
[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.	4	2	2
[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	3	3	2
Tiny Fragments - Possible Hostile Activity	3	2	3
[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot	3	3	2
SYN-FIN scan!	2	2	1
External FTP to HelpDesk 10.0.53.29	2	2	1
Bugbear@MM virus in SMTP	2	2	1
[UMBC NIDS IRC Alert] Possible drone command detected.	2	1	1
SMB Name Wildcard [**] 10.0.150.19812/18-09:15:09.605362 [**] Null scan!	1	1	1
10.0.30.3 activity [**] 165.247.111.23512/18-16:39:24.617169 [**] 10.0.30.3 activity	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.7912/17-02:30:53.812014 [**] Incomplete Packet Fragments Discarded	1	1	1
High port 65535 tcp - possible Red Worm - traffic [**] 66.57.196.18412/20-20:59:23.280870 [**] TFTP - Internal TCP connection to external tftp server	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.9212/17-08:02:27.246920 [**] Incomplete Packet Fragments Discarded	1	1	1
EXPLOIT x86 NOOP [**] 211.33.14.5312/17-09:38:41.038872 [**] Incomplete Packet Fragments Discarded	1	1	1
connect to 515 from inside [**] 10.0.162.4112/19-13:44:03.703496 [**] 10.0.30.4 activity	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.7912/17-09:40:56.412514 [**] Incomplete Packet Fragments Discarded	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.6712/17-03:12:10.173112 [**] Incomplete Packet Fragments Discarded	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.7912/17-07:56:45.685548 [**] Incomplete Packet Fragments Discarded	1	1	1
PHF attempt	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.6812/17-09:48:57.596883 [**] Incomplete Packet Fragments Discarded	1	1	1
10.0.30.3 activity [**] 68.57.90.14612/17-12:29:49.384564 [**] Incomplete Packet Fragments Discarded	1	1	1
Possible wu-ftpd exploit - GIAC000623	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.6912/17-09:37:28.403010 [**] Incomplete Packet Fragments Discarded	1	1	1
[UMBC NIDS] External MiMail alert [**] 24.102.139.8612/20-12:53:44.176991 [**] Incomplete Packet Fragments Discarded	1	1	1
TFTP - Internal TCP connection to external tftp server [**] 69.10.132.12112/20-14:27:27.416375 [**] 10.0.30.3 activity	1	1	1
Probable NMAP fingerprint attempt	1	1	1

Signature (click for sig info)	# Alerts	# Sources	# Dests
Traffic from port 53 to port 123	1	1	1
10.0.30.3 activity [**] 68.55.113.19412/20-10:48:36.678280 [**] SMB Name Wildcard	1	1	1
[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.6712/17-03:18:45.075265 [**] Incomplete Packet Fragments Discarded	1	1	1
Incomplete Packet Fragments Discarded [**] 10.0.21.6712/17-09:41:33.744640 [**] Incomplete Packet Fragments Discarded	1	1	1
TFTP - External UDP connection to internal tftp server	1	1	1