Global Information Assurance Certification Paper

Intrusion Detection in Depth
GCIA Practical Assignment
Version 3.3

Mike McCabe
November 3, 2002

0.0 Abstract Summary

This practical is presented in three parts. The first part is an adventure in Intrusion Detection Strategy and Installation. This depicts an actual project that was undertaken by Campus Labs to design and implement an Intrusion Detection System to alert on attacks and scans going across their perimeter.

The second part of this practical consists of three network detects. All of the detects were taken from the raw detects files on intrusions.org. The first detect was of the ISAPI .ida overflow attack that CodeRed uses. The second detect was of a DNS Named Version Attempt. And finally, the third detect was a directory traversal attack.

The third section of this practical is a detailed analysis of the GIAC Universities Snort logs for a period of time from August 1, 2002 to August 5, 2002. The analysis found a host infected by the Nimda virus. The other alerts included some CodeRed detects, some RPC scans, TFTP attacks and SMB Name Wildcard attempts.

1.0 Describe the State of Intrusion Detection – An Adventure in Intrusion Detection Strategy and Implementation

1.1 Introduction

This paper will be a story about our attempt to develop and install an Intrusion Detection System at the Campus Labs facility. We started with only firewall logs from a Checkpoint FW-1 firewall and ended up with a complete perimeter Intrusion Detection System using the Snort IDS system.

Our strategy included installations for a complete perimeter Intrusion Detection System along with host-based systems, network systems and an event correlation system. The host-based systems and network systems were selected to be both at the perimeter and at key critical points in the network.

1.2 The Strategy

We wanted to develop a complete strategy for monitoring our network including both the perimeter and critical hosts and networks. Included in this strategy was a system to perform event correlation between all of the intrusion detections systems and the firewall. As can be seen in the diagram in Figure #1.1 the strategy for placement of the intrusion detection systems consisted of key points on both the perimeter of the network and key internal systems. The intrusion detection appliances are placed to monitor the outside of the firewall, the DMZ network or as it's known at campus labs the collaborative network, the inside of the firewall and on key networks on the internal network.

The appliance placed on the outside of the firewall is to show all attacks that are directed at our network from the Internet. The rule set for this appliance will be set to try and eliminate false positives generated by normal traffic through the firewall. The appliance placed in the collaborative network is there to detect any attacks or scans that are directed to some web servers that are placed in the collaborative network. These web servers are all setup to allow only connections to them that use SSL and RSA SecurID and the traffic to them is light because these are for very distinct customers to use instead of the public.

The appliance that is placed on the inside of the firewall is there to detect any attacks or scans that actually make it past the firewall into the network. They will also detect any attacks or scans that are perpetrated by the internal network against another system or network on the Internet. The appliances on the inside of the network are placed on critical servers and critical networks to monitor inside traffic to both catch any malware that is loose inside the network and to also catch any internal attacks on the critical servers. The console is a system that will coordinate and correlate events across the IDS systems. It is

accessible via a web based interface from anywhere within the internal network but not accessible from the Internet.

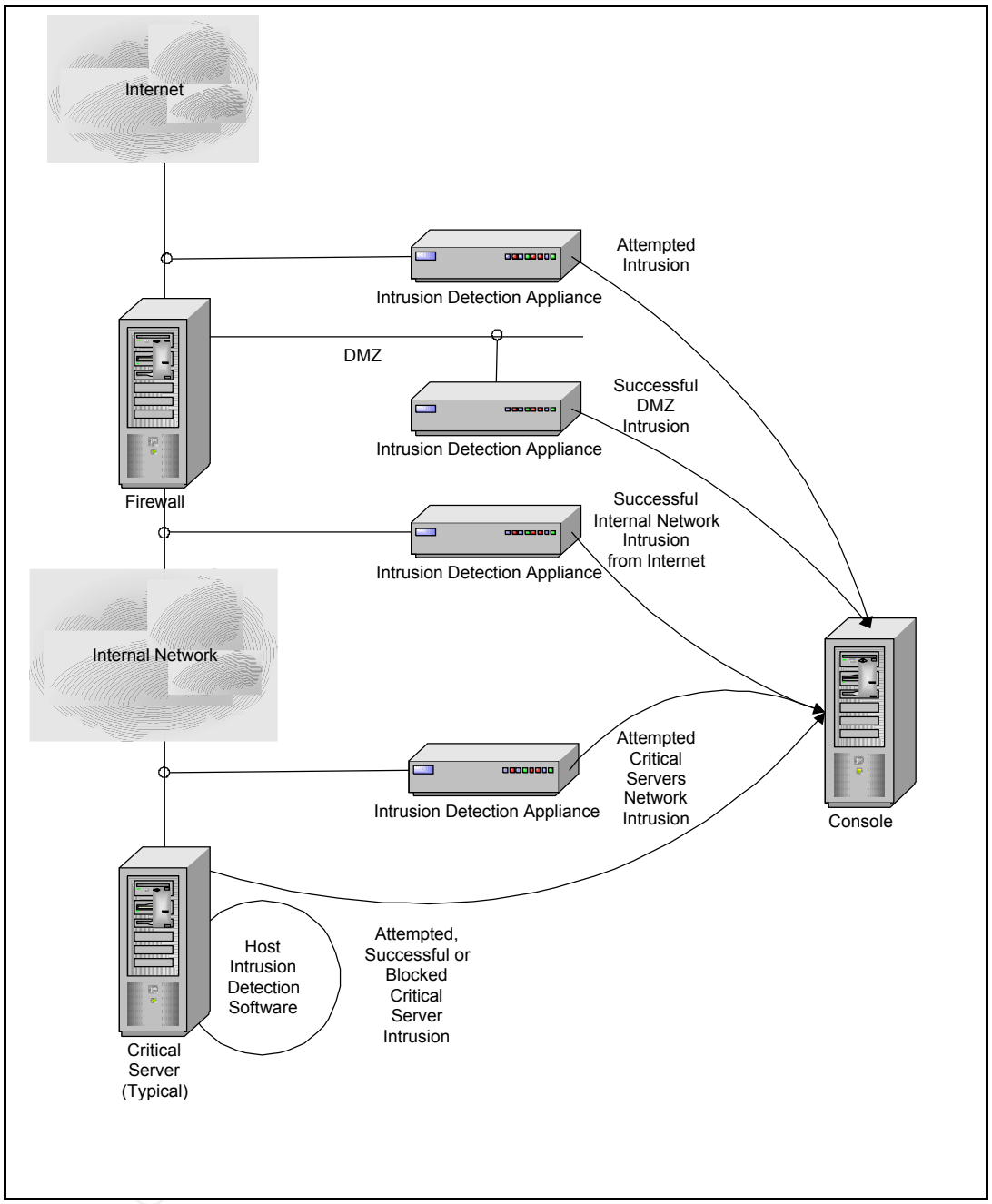

Figure 1.1

1.3 The Equipment Selection

The next step towards completion of the project was the selection of the equipment and vendors. This step took the longest time because many vendors were looked at and investigated. The features of various appliances and software based solutions were determined and evaluated based on signature lists, packet bandwidth capability, logging ability and interoperability with other systems.

The final decision made was to utilize the Cisco IDS set of appliances for Network Intrusion Detection and their host based system. The Campus Labs network is comprised of mostly Cisco network equipment. The Cisco IDS systems come in many forms and capabilities, everything from a blade that works in their upper end switches to 1U high appliances running proprietary hardware. The systems are all capable of being read by the event correlation software we decided upon which was Net Forensics.

The event correlation software accepts log files and/or snmp traps from the various appliances and the firewall and provides a comprehensive database from which multiple detects are turned into a single event that can then be viewed via a web based interface.

1.4 Reality Sets In

The basic price of doing just the perimeter based intrusion detection system, which was determined to be the minimum requirements for a starter solution, was approximately $100,000. Unfortunately this could not be budgeted for during 2002 or 2003 so some alternatives were needed. It was determined to look into using a server with multiple network interfaces and the free Snort IDS system. A server that was being retired was redirected to the test lab for developing a test IDS system.

1.5 The Test Lab

We began by investigating the Snort system on both Linux and Windows systems. Linux was chosen because it seemed from our research that the Linux system was much better supported by Snort and it was possible to run the combination of Snort, Mysql and Acid. The distribution of Linux selected was RedHat 7.3 which at the time was the latest and greatest version of the RedHat Linux system. The Mysql database was selected because of previous experience with this database. The version of the database was 11.16, which at the time was the most recent version. The Acid web interface was selected so that the alerts that were being generated by the Snort system could be easily viewed via the web interface.

The system was installed originally with only two network interfaces. One for communicating back to the real world and one for being set to promiscuous mode for watching of the outside interface of the firewall. Later two more network interfaces were added to be setup in promiscuous mode for monitoring of the collaborative network and inside interface of the firewall.

1.6 Tuning the System

The next step in the process of putting the system in place was to tune it to remove false positives. This process took approximately 2 months and actually still continues to this day. The main things that were found to produce false positives included the following areas:

- DNS Zone Transfers between primary and secondary name servers.
- DDOS alerts that use only the port number to detect on.
- ICMP administratively blocked alerts.
- Compaq Insight Directory alerts caused by central monitoring system.
- ICMP Level 3 retriever pings between hosts across the firewall.

The system will continue to be tuned and updated with new rules on a periodic basis.

1.7 Conclusion

The project was determined to be a rousing success. It has saved Campus Labs a great deal of money so far and has prompted our production support group to begin supporting the Linux operating system. The next step in the process will be to actually put the system into production. It is planned to also separate the database and reporting functions (i.e. Mysql and Acid) from the Snort sensor to allow for more sensors to be put in place across the network on the critical networks. We are also investigating some free host based intrusion detection systems to possibly allow us to continue with our strategy even during our budget problems.

1.8 References

Shipley, Greg. "Intrusion Detection, Take Two" URL: http://www.networkcomputing.com/1023/1023f1.html (November 3, 2002)

"Cisco Announces IDS Host Sensor for Mitigating Attacks Against Server Resources" URL: http://newsroom.cisco.com/dlls/prod_090401c.html (November 3, 2002)

"Q&A Cisco Intrusion Detection System" URL: http://www.cisco.com/en/US/products/hw/vpndevc/ps976/products_qanda_item09186a00800887c2.shtml (November 3, 2002)

Scott, Steven J. "Snort Installation Manual – Snort, MySql, Redhat 7.3" URL: http://www.snort.org/docs/snort-rh7-mysql-ACID-1-5.pdf (November 3, 2002)

Danyliw, Roman. "ACID: Installation and Configuration" URL: http://www.snort.org/external/?url=http://www.andrew.cmu.edu/~rdanyliw/snort/acid_config.html (November 3, 2002)

Green, Chris. "Snort User Manual" URL: http://www.snort.org/docs/writing_rules-1.9.0/ (November 2, 2002)

2.0 Network Detects

2.1 Detect #1 – WEB-IIS ISAPI .ida attempt

2.1.1 Snort Dump of Detect

```
[**] [1:1243:2] WEB-IIS ISAPI .ida attempt [**]
[Classification: Web Application Attack] [Priority: 1]
07/08-03:01:36.634488 12.238.113.32:3993 -> 46.5.180.133:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:1504
***AP*** Seq: 0xFE1C569E  Ack: 0x395F03D4  Win: 0x7D78  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS552]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0071]
```

2.1.2 TCPDUMP of Detect

```
03:01:36.634488 12.238.113.32.3993 > 46.5.180.133.80: P 4263270046:4263271510(1464) ack 962528212 win 32120 [tos
0x10]  (ttl 240, id 0, len 1504, bad cksum 0!)
0x0000   4510 05e0 0000 0000 f006 0000 0cee 7120        E.............q.
0x0010   2e05 b485 0f99 0050 fe1c 569e 395f 03d4        .......P..V.9_..
0x0020   5018 7d78 0000 0000 4745 5420 2f64 6566        P.}x....GET./def
0x0030   6175 6c74 2e69 6461 3f4e 4e4e 4e4e 4e4e        ault.ida?NNNNNNN
0x0040   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0050   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0060   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0070   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0080   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0090   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00a0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00b0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00c0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00d0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00e0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x00f0   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0100   4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e        NNNNNNNNNNNNNNNN
0x0110   4e4e 4e4e 4e4e 4e4e 4e00 0000 0000 0000        NNNNNNNNN.......
0x0120   0000 0000 0000 0000 c303 0000 0078 00fa        .............x..
0x0130   2025 7539 3039 3025 7536 3835 3825 7563        .%u9090%u6858%uc
0x0140   6264 3325 7537 3830 3125 7539 3039 3025        bd3%u7801%u9090%
0x0150   7536 3835 3825 7563 6264 3325 7537 3830        u6858%ucbd3%u780
0x0160   3125 7539 3039 3025 7539 3039 3025 7538        1%u9090%u9090%u8
0x0170   3139 3025 7530 3063 3325 7530 3030 3325        190%u00c3%u0003%
0x0180   7538 6230 3025 7535 3331 6225 7535 3366        u8b00%u531b%u53f
0x0190   6625 7530 3037 3825 7530 3030 3025 7530        f%u0078%u0000%u0
0x01a0   303d 6120 2048 5454 502f 312e 300d 0a43        0=a..HTTP/1.0..C
0x01b0   6f6e 7465 6e74 2d74 7970 653a 2074 6578        ontent-type:.tex
0x01c0   742f 786d 6c0a 484f 5354 3a77 7777 2e77        t/xml.HOST:www.w
0x01d0   6f72 6d2e 636f 6d0a 2041 6363 6570 743a        orm.com..Accept:
0x01e0   202a 2f2a 0a43 6f6e 7465 6e74 2d6c 656e        .*/*.Content-len
0x01f0   6774 683a 2033 3536 3920 0d0a 0d0a 558b        gth:.3569.....U.
0x0200   ec81 ec18 0200 0053 5657 8dbd e8fd ffff        .......SVW......
0x0210   b986 0000 00b8 cccc cccc f3ab c785 70fe        ..............p.
0x0220   ffff 0000 0000 e90a 0b00 008f 8568 feff        .............h..
0x0230   ff8d bdf0 feff ff64 a100 0000 0089 4708        .......d......G.
0x0240   6489 3d00 0000 00e9 6f0a 0000 8f85 60fe        d.=.....o.....`.
0x0250   ffff c785 f0fe ffff ffff ffff 8b85 68fe        ..............h.
0x0260   ffff 83e8 0789 85f4 feff ffc7 8558 feff        .............X..
0x0270   ff00 00e0 77e8 9b0a 0000 83bd 70fe ffff        ....w.......p...
0x0280   000f 85dd 0100 008b 8d58 feff ff81 c100        .........X......
0x0290   0001 0089 8d58 feff ff81 bd58 feff ff00        .....X.....X....
0x02a0   0000 7875 0ac7 8558 feff ff00 00f0 bf8b        ..xu...X........
```

```
0x02b0   9558 feff ff33 c066 8b02 3d4d 5a00 000f          .X...3.f..=MZ...
0x02c0   859a 0100 008b 8d58 feff ff8b 513c 8b85          .......X....Q<..
0x02d0   58fe ffff 33c9 668b 0c10 81f9 5045 0000          X...3.f.....PE..
0x02e0   0f85 7901 0000 8b95 58fe ffff 8b42 3c8b          ..y.....X....B<.
0x02f0   8d58 feff ff8b 5401 7803 9558 feff ff89  .X....T.x..X....
0x0300   9554 feff ff8b 8554 feff ff8b 480c 038d  .T.....T....H...
0x0310   58fe ffff 898d 4cfe ffff 8b95 4cfe ffff  X.....L.....L...
0x0320   813a 4b45 524e 0f85 3301 0000 8b85 4cfe          .:KERN..3.....L.
0x0330   ffff 8178 0445 4c33 320f 8520 0100 008b          ...x.EL32.......
0x0340   8d58 feff ff89 8d34 feff ff8b 9554 feff  .X.....4.....T..
0x0350   ff8b 8558 feff ff03 4220 8985 4cfe ffff  ...X....B...L...
0x0360   c785 48fe ffff 0000 0000 eb1e 8b8d 48fe          ..H...........H.
0x0370   ffff 83c1 0189 8d48 feff ff8b 954c feff  .......H.....L..
0x0380   ff83 c204 8995 4cfe ffff 8b85 54fe ffff  ......L.....T...
0x0390   8b8d 48fe ffff 3b48 180f 8dc0 0000 008b          ..H...;H........
0x03a0   954c feff ff8b 028b 8d58 feff ff81 3c01  .L.......X....<.
0x03b0   4765 7450 0f85 a000 0000 8b95 4cfe ffff          GetP........L...
0x03c0   8b02 8b8d 58fe ffff 817c 0104 726f 6341          ....X....|..rocA
0x03d0   0f85 8400 0000 8b95 48fe ffff 0395 48fe          ........H.....H.
0x03e0   ffff 0395 58fe ffff 8b85 54fe ffff 8b48  ....X.....T....H
0x03f0   2433 c066 8b04 0a89 854c feff ff8b 8d54          $3.f.....L.....T
0x0400   feff ff8b 5110 8b85 4cfe ffff 8d4c 10ff  ....Q...L....L..
0x0410   898d 4cfe ffff 8b95 4cfe ffff 0395 4cfe  ..L.....L.....L.
0x0420   ffff 0395 4cfe ffff 0395 4cfe ffff 0395  ....L.....L.....
0x0430   58fe ffff 8b85 54fe ffff 8b48 1c8b 140a  X.....T....H....
0x0440   8995 4cfe ffff 8b85 4cfe ffff 0385 58fe  ..L.....L.....X.
0x0450   ffff 8985 70fe ffff eb05 e90d ffff ffe9  ....p...........
0x0460   16fe ffff 8dbd f0fe ffff 8b47 0864 a300  ...........G.d..
0x0470   0000 0083 bd70 feff ff00 7505 e938 0800          .....p....u..8..
0x0480   00c7 854c feff ff01 0000 00eb 0f8b 8d4c          ...L...........L
0x0490   feff ff83 c101 898d 4cfe ffff 8b95 68fe  ........L.....h.
0x04a0   ffff 0fbe 0285 c00f 848d 0000 008b 8d68          ...............h
0x04b0   feff ff0f be11 83fa 0975 218b 8568 feff  .........u!..h..
0x04c0   ff83 c001 8bf4 50ff 9590 feff ff3b f490  ......P......;..
0x04d0   434b 434b 8985 34fe ffff eb2a 8bf4 8b8d          CKCK..4....*....
0x04e0   68fe ffff 518b 9534 feff ff52 ff95 70fe  h...Q..4...R..p.
0x04f0   ffff 3bf4 9043 4b43 4b8b 8d4c feff ff89  ..;..CKCK..L....
0x0500   848d 8cfe ffff eb0f 8b95 68fe ffff 83c2  ..........h.....
0x0510   0189 9568 feff ff8b 8568 feff ff0f be08  ...h.....h......
0x0520   85c9 7402 ebe2 8b95 68fe ffff 83c2 0189          ..t.....h.......
0x0530   9568 feff ffe9 53ff ffff 8b85 68fe ffff  .h....S.....h...
0x0540   83c0 0189 8568 feff ff8b 4d08 8b91 8400          .....h....M.....
0x0550   0000 8995 6cfe ffff c785 4cfe ffff 0400  ....l.....L.....
0x0560   0000 c685 d0fe ffff 688b 4508 8985 d1fe          ........h.E.....
0x0570   ffff c785 d5fe ffff 5b53 53ff c785 d9fe  ........[SS.....
0x0580   ffff 6378 9090 8b4d 088b 5110 8995 50fe          ..cx...M..Q...P.
0x0590   ffff 83bd 50fe ffff 0075 268b f46a 008d  ....P....u&..j..
0x05a0   854c feff ff50 8b8d 68fe ffff 518b 5508  .L...P..h...Q.U.
0x05b0   8b42 0850 ff95 6cfe ffff 3bf4 9043 4b43 .B.P..l...;..CKC
0x05c0   4b83 bd50 feff ff64 7d5c 8b8d 50fe ffff  K..P...d}\..P...
0x05d0   83c1 0189 8d50 feff ff8b 9550            .....P.....P
```

2.1.3 Snort Rule for Detect

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .ida attempt"; uricontent:".ida?"; nocase; dsize:>239; flags:A+; reference:arachnids,552; classtype:web-application-attack; reference:cve,CAN-2000-0071; sid:1243; rev:2;)

2.1.4 Source of Trace

The source of this trace was from the file http://www.incidents.org/logs/raw/2002.6.8 which was downloaded on August 23, 2002. The binary tcpdump file that was used was then further analyzed using tcpdump and snort.

2.1.5 Detect Generated By

This detect was generated by Snort v1.9-dev (Build 92) running on Redhat Linux 7.3.

The snort command used to generate the above snort alert was:

Snort –c /etc/snort/snort.conf –r 2002.6.8

The snort ruleset and snort.conf files were vanilla copies directly from snort.org. Once snort was run on the binary file the alert file located in the log directory was examined using an editor for events of interest. The next step was to generate and examine the packet to try and determine if it was really an attack attempt or a false positive. To do this tcpdump was run using the following command:

Tcpdump –xX –vv –n –r 2002.6.8 host 12.238.113.32 > 200268.txt

2.1.6 Probability the Source Address was spoofed

There is always some probability that the source address is spoofed. The biggest consideration of whether this is true or not is to ask ourselves does the attacker want a result back from this attack. The answer to the question of whether they want a response in this case is no since if the packet is examined closely one notices that it seems that a reference to a web site http://www.worm.com is present in the packet. In review of this web site it was found to be in a placeholder status at dotregister.com. This leads one to believe that whoever owns this web site periodically moves it around. A whois lookup on the source address shows that AT&T is the owner of the address block. The most likely answer to the question of whether the source address was spoofed is no it was not spoofed. It is most likely coming from a temporary address assigned by AT&T and any response desired is going back to worm.com.

To also determine if the source address may have been spoofed a check was done using tcpdump to look back at files 2002.6.7, 2002.6.6, 2002.6.5, 2002.6.4, 2002.6.3, 2002.6.2 and 2002.6.1. No evidence of a reconnaissance scan was found.

A quick check of http://www.arin.net looking for information about where this attack is coming from revealed that the attack is coming from an AT&T source. The whois information is shown below:

```
OrgName:    AT&T WorldNet Services
```

```
OrgID:      ATTW

NetRange:   12.0.0.0 - 12.255.255.255
CIDR:       12.0.0.0/8
NetName:    ATT
NetHandle:  NET-12-0-0-0-1
Parent:
NetType:    Direct Allocation
NameServer: DBRU.BR.NS.ELS-GMS.ATT.NET
NameServer: DMTU.MT.NS.ELS-GMS.ATT.NET
NameServer: CBRU.BR.NS.ELS-GMS.ATT.NET
NameServer: CMTU.MT.NS.ELS-GMS.ATT.NET
Comment:    For abuse issues contact abuse@att.net
RegDate:    1983-08-23
Updated:    2002-08-23

TechHandle: DK71-ARIN
TechName:   Kostick, Deirdre
TechPhone:  +1-919-319-8249
TechEmail:  help@ip.att.net

OrgAbuseHandle: ATTAB-ARIN
OrgAbuseName:   ATT Abuse
OrgAbusePhone:  +1-919-319-8130
OrgAbuseEmail:  abuse@att.net

OrgTechHandle: ICC-ARIN
OrgTechName:   IP Customer Care
OrgTechPhone:  +1-888-613-6330
OrgTechEmail:  qhoang@att.com

OrgTechHandle: ITS3-ARIN
OrgTechName:   IP Tier Support
OrgTechPhone:  +1-888-613-6330
OrgTechEmail:  NIPAtier3@ems.att.com

OrgTechHandle: IPSWI-ARIN
OrgTechName:   IP SWIP
OrgTechPhone:  +1-888-613-6330
OrgTechEmail:  swipid@nipaweb.vip.att.net
```

2.1.7 Description of the Attack

According to the dictionary at http://cve.mitre.org there are two current vulnerabilities associated with .ida files:

CVE-2001-0500
CAN-2002-0071

The description on CVE-2001-0500 lists the Code Red virus as using this vulnerability to attack systems. The concept that this could be a scan looking for machines vulnerable to the code red virus is plausible because of the reference to worm.com. All variants of CodeRed use this vulnerability to attempt spreading themselves to other hosts. The vulnerability is in the Index Server which has a problem with a buffer overflow programming bug. CodeRed attempts to exploit this vulnerability. If it is successful then it lodges itself in memory and starts to try and infect other systems. If this is the .C variant then it also attempts to gain web site access to the C: and D: drives, puts a copy of

a file called explorer.exe on the C: drive, changes some registry settings and tries to write a file called root.exe into the scripts directory of the web server so that root access can be gotten at any time.

2.1.8 Attack Mechanism

The attack mechanism for Code Red uses a vulnerability in the Index Server that comes with Microsoft Windows 2000 and especially the Internet Information Server (IIS). The actual vulnerability is a buffer overflow that exists in the Index server. It is not necessary to be running the Index Server for this vulnerability to be abused since all that is needed is the mapping for .ida and .idq files to be in place. The only other requirement is that the IIS server be accessible to the attacker. As with most buffer overflow vulnerabilities the attacker can use this to execute unwanted code on the server therefore making the system vulnerable to root kitting or having a Trojan put in place. In the C variant of CodeRed the virus actually puts a Trojan or backdoor in place on the system. The virus also checks to see if the system it infects has the Chinese language and if it does it doubles the number of attack threads it runs to 600 from the standard 300. It is also selective about what systems it scans for vulnerabilities. It probes nearby systems with a 50% probability, a 37.5% probability in the same Class A network and a 12.5% probability that it will be in the same Class B network.

This vulnerability is well documented and patches have been out for a good amount of time to protect against this virus.

2.1.9 Correlations

This same attack has been heavily documented by other GIAC writers including:

http://www.giac.org/practical/Christopher_Lee_GCIA.doc
http://www.giac.org/practical/David_Begg_GCIA.doc

The practical by Christopher Lee was helpful in the description of the attack and the attack mechanism discussions. It also confirms that the standard CodeRed attack is typically not targeted. The David Begg practical was helpful in his discussion of the severity calculation and in the attack mechanism area.

A quick check of http://dshield.com reveals that this IP address has been reported as being an attacker 101 times.

Also, the site at http://cve.mitre.org has the following identifiers for this vulnerability:

CVE-2001-0500
CAN-2002-0071

2.1.10 Evidence of Active Targeting

Since this is a pretty old vulnerability and only one packet was directed at the web server it is most likely just a scan looking for vulnerable IIS servers, possibly even a compromised host looking to spread the virus. This would mean that the hacker is not actively targeting our systems. In the descriptions of the vulnerabilities and on the web site at http://vil.nai.com/vil/content/v_99177.htm is a good description of the methods that CodeRed uses to find new hosts to infect. Basically it randomly picks an IP address based on a probability formula that has it pick hosts close to the source 50% of the time, IP addresses in the same Class A network 37.5% of the time and an IP address in the same Class B network 12.5% of the time. The remainder of the time it picks a random address from the entire IP address range.

2.1.11 Severity

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

Criticality : 4 - The system targeted is obviously a public web server and therefore of great interest to its owner.

Lethality : 5 - Codered and its variants can contain Trojans that can be left behind and will begin actively targeting other systems in the local vicinity.

System Countermeasures : 5 – Since a review of logs shows that the web server did not respond to this attack by being compromised. Nor did it start showing any signs of beginning to attack other hosts this system is either not running IIS or is sufficiently patched to guard against this attack.

Network Countermeasures : 1 – There is no evidence of any Network based controls that would stop this packet from reaching the intended host.

Severity = (4 + 5) – (5 + 1) = 3

2.1.12 Defensive Recommendation

The best defense against this attack is to make sure that all of your Windows NT/2000 systems are patched and up to date. To actively guard against this attack it is recommended that your Windows NT/2000 system be upgraded to the latest service pack available. If you are running Windows NT 4.0 server then there is a security rollup patch available at:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q299444

Note that Windows 2000 Professional should not be susceptible to this attack since they do not run the IIS server.

2.1.13 Multiple Choice Test Question

CodeRed exploits a vulnerability in which server?

A. Microsoft Windows Internet Information Server
B. Microsoft Windows Index Server
C. Apache Web Server
D. Microsoft Exchange Server

The correct answer is B.

2.1.14 Intrusions@incidents.org posting

The above attack was posted to the mailing list for the third time at intrusions@incidents.org on October 13, 2002. Only one response was ever received for this attack.

Chris Baker sent the following questions on October 13:

In the source of trace section:

```
I like it when detects read like a walkthrough. I would recommend
ordering the detect data by the timeframe in which they were
captured.
Take me through the detection. Here you indicate how you found this
detect, however the data at the top of your analysis starts with
tcpdump first.

What ruleset was used? Is this a vanilla snort.conf, or was it
modified? If modified, then what?
```

My Response:

```
Thanks for the tip... I have re-ordered the sections to better
reflect how the information was gathered.  I have also added some
commentary on the ruleset from snort.
```

In the probability the source address was spoofed section:

```
Was there any other data from the attacker which might further
support
that this was a code red infection. Were there any recon scans?
```

My Response:

```
I have checked as far back as 2002.6.1 and found no evidence of any
kind of recon scans.  I will update the verbiage to reflect this
check.
```

In the description of the attack section:

```
Ok, I'm starting to believe that this is code red, now I want to know
more about why code red is using the .ida exploit. What happens if
this succeeds?
```

My Response:

```
Ok, I have beefed up the verbiage and explained how the .ida exploit
```

is used and what happens if this succeeds...

In the Attack Mechanism section:
More information about what we misssed and how it is important to code
red's operation would be nice to see here.

My Response:
Not really sure what you mean by "what we missed" I have described a little bit about what codered does to the system when it infects.

In the correlations section:
You make references to other documents, however I would prefer to not read the full document to find the correlating information. How are these documents helpful to you. What insight did they give you about the attack?

My Response:
I have described in more detail the information that was gathered from these practicals.

In the evidence of active targeting section:
Given that the assumption regarding this being the second order effect of a code red infection, a description of the various scanning techniques would be valuable here. This would further support the argument that it is nontargeted.

My Response:
I have beefed up this section to describe the randomness of the codered attack mechanism for picking IP addresses to check.

2.2 Detect #2 – DNS named version attempt

2.2.1 TCPDUMP of Detect

20:04:16.804488 203.107.138.81.3666 > 46.5.131.155.53: [bad udp cksum f9f9!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 23301, len 58, bad cksum 3157!)
20:07:11.354488 203.107.138.81.3006 > 46.5.236.209.53: [bad udp cksum f9f9!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 26548, len 58, bad cksum bb71!)
21:36:25.264488 203.107.138.81.1950 > 46.5.52.228.53: [bad udp cksum faf7!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 5508, len 58, bad cksum c78e!)
22:08:04.704488 203.107.138.81.3113 > 46.5.13.16.53: [bad udp cksum f8f8!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 51255, len 58, bad cksum 3bb1!)
22:44:33.924488 203.107.138.81.2122 > 46.5.14.142.53: [bad udp cksum faf7!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 29337, len 58, bad cksum 90cf!)
00:40:44.314488 203.107.138.81.2772 > 46.5.46.54.53: [bad udp cksum f8f8!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 42594, len 58, bad cksum 3c60!)
01:24:08.654488 203.107.138.81.1610 > 46.5.25.205.53: [bad udp cksum faf7!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 30843, len 58, bad cksum 7fae!)
01:33:56.754488 203.107.138.81.2079 > 46.5.158.163.53: [bad udp cksum f9f9!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 41622, len 58, bad cksum cebd!)
02:14:53.064488 203.107.138.81.4642 > 46.5.213.6.53: [bad udp cksum f7fa!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 18724, len 58, bad cksum f0ce!)
02:47:57.584488 203.107.138.81.1085 > 46.5.45.177.53: [bad udp cksum faf7!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 53349, len 58, bad cksum 13e0!)
03:22:29.304488 203.107.138.81.4793 > 46.5.3.139.53: [bad udp cksum faf7!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 54953, len 58, bad cksum 37c2!)

2.2.2 Snort Dump of Detect

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/07-20:04:16.804488 203.107.138.81:3666 -> 46.5.131.155:53
UDP TTL:45 TOS:0x0 ID:23301 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/07-20:07:11.354488 203.107.138.81:3006 -> 46.5.236.209:53
UDP TTL:45 TOS:0x0 ID:26548 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/07-21:36:25.264488 203.107.138.81:1950 -> 46.5.52.228:53
UDP TTL:45 TOS:0x0 ID:5508 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/07-22:08:04.704488 203.107.138.81:3113 -> 46.5.13.16:53
UDP TTL:45 TOS:0x0 ID:51255 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/07-22:44:33.924488 203.107.138.81:2122 -> 46.5.14.142:53
UDP TTL:45 TOS:0x0 ID:29337 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/08-00:40:44.314488 203.107.138.81:2772 -> 46.5.46.54:53
UDP TTL:45 TOS:0x0 ID:42594 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/08-01:24:08.654488 203.107.138.81:1610 -> 46.5.25.205:53
UDP TTL:45 TOS:0x0 ID:30843 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/08-01:33:56.754488 203.107.138.81:2079 -> 46.5.158.163:53
UDP TTL:45 TOS:0x0 ID:41622 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/08-02:14:53.064488 203.107.138.81:4642 -> 46.5.213.6:53
UDP TTL:45 TOS:0x0 ID:18724 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

```
[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/08-02:47:57.584488 203.107.138.81:1085 -> 46.5.45.177:53
UDP TTL:45 TOS:0x0 ID:53349 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/08-03:22:29.304488 203.107.138.81:4793 -> 46.5.3.139:53
UDP TTL:45 TOS:0x0 ID:54953 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]
```

2.2.3 Snort Rule for Detect

```
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version attempt"; content:"|07|version"; offset:12;
content:"|04|bind"; nocase; offset: 12; reference:arachnids,278; classtype:attempted-recon; sid:257; rev:1;)
```

2.2.4 Source of Trace

The source of this trace was from the file http://www.incidents.org/logs/raw/2002.6.8 which was downloaded on August 23, 2002. The binary tcpdump file that was used was then further analyzed using tcpdump and snort.

2.2.5 Detect Generated By

This detect was generated by Snort v1.9-dev (Build 92) running on Redhat Linux 7.3.

The snort command used to generate the above snort alert was:

Snort –c /etc/snort/snort.conf –r 2002.6.8

Once snort was run on the binary file the alert file located in the log directory was examined using an editor for events of interest. The next step was to generate and examine the packet to try and determine if it was really an attack attempt or a false positive. To do this tcpdump was run using the following command:

Tcpdump –vv –n –r 2002.6.8 host 203.107.138.81 > 200268.txt

2.2.6 Probability the Source Address was spoofed

It is very unlikely that the source address has been spoofed because in this type of attack the attacker is collecting information about our domain name servers. The attacker is trying to find the version number so that they can exploit any known vulnerabilities that are associated with that version number of bind. This leads us to perform a whois lookup on the address to see who is testing for our name servers. Using http://www.apnic.net whois server produces the following output:

```
inetnum:        203.107.128.0 - 203.107.255.255
netname:        COMNET-TH
```

```
descr:          KSC Commercial Internet Co. Ltd.
descr:          2/4 Samaggi Insurance Tower 10th Fl.,
descr:          Viphavadee-Rangsit RD
descr:          Thungsonghong, Laksi
descr:          Bangkok 10210
country:        TH
admin-c:        JT183-AP
tech-c:         TO94-ORG
remarks:        service provider
remarks:        Delegate four /19 to /17
mnt-by:         APNIC-HM
mnt-lower:      KSC-ADMIN
changed:        hostmaster@apnic.net 20000301
changed:        hostmaster@apnic.net 20011016
changed:        hm-changed@apnic.net 20020830
status:         ALLOCATED PORTABLE
source:         APNIC

person:         Joost Th.A Doevelaar
nic-hdl:        JT183-AP
e-mail:         jdoevelaar@ksc.net
address:        KSC Commercial Internet Co.,Ltd.
address:        2/4 Samaggi Insurance Tower 10th Fl., Viphavadee-Rangsit Rd.,
address:        Thungsonghong, Laksi
address:        Bangkok 10210
phone:          +66-2-9797777 ext. 8900
fax-no:         +66-2-5885665
country:        TH
changed:        netadmin@ns.ksc.co.th 20020828
mnt-by:         KSC-ADMIN
source:         APNIC

person:         Technical Operation Center
address:        KSC Commercial Internet Co.,Ltd.
address:        Operation Department
address:        2/4 Samaggi Insurance Tower 10th Fl., Viphavadee-Rangsit Rd.,
address:        Thungsonghong, Laksi
address:        Bangkok 10210
country:        TH
phone:          +66-2-9797777 ext. 8428
e-mail:         netadmin@ns.ksc.co.th
nic-hdl:        TO94-ORG
mnt-by:         KSC-ADMIN
changed:        admin@ns.ksc.co.th 20011012
source:         APNIC
```


2.2.7 Description of the Attack

The site at http://cve.mitre.org lists the following CVE and CAN number for various vulnerabilities associated with the Named (BIND) server.

Name	Description
CVE-1999-0009	Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases.
CVE-1999-0010	Denial of Service vulnerability in BIND 8 Releases via maliciously formatted DNS messages.
CVE-1999-0011	Denial of Service vulnerabilities in BIND 4.9 and BIND 8 Releases via CNAME record and zone transfer.
CVE-1999-0024	DNS cache poisoning via BIND, by predictable query IDs.
CVE-1999-0184	When compiled with the -DALLOW_UPDATES option, bind allows dynamic updates to the DNS server, allowing for malicious modification of DNS records.
CVE-1999-0385	The LDAP bind function in Exchange 5.5 has a buffer overflow that allows a remote attacker to conduct a denial of service or execute commands.
CVE-1999-0833	Buffer overflow in BIND 8.2 via NXT records.
CVE-1999-0835	Denial of service in BIND named via malformed SIG records.
CVE-1999-0837	Denial of service in BIND by improperly closing TCP sessions via so_linger.

CVE-1999-0848	Denial of service in BIND named via consuming more than "fdmax" file descriptors.
CVE-1999-0849	Denial of service in BIND named via maxdname.
CVE-1999-0851	Denial of service in BIND named via naptr.
CVE-2000-0887	named in BIND 8.2 through 8.2.2-P6 allows remote attackers to cause a denial of service by making a compressed zone transfer (ZXFR) request and performing a name service query on an authoritative record that is not cached, aka the "zxfr bug."
CVE-2000-0888	named in BIND 8.2 through 8.2.2-P6 allows remote attackers to cause a denial of service by sending an SRV record to the server, aka the "srv bug."
CVE-2001-0010	Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges.
CVE-2001-0011	Buffer overflow in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges.
CVE-2001-0012	BIND 4 and BIND 8 allow remote attackers to access sensitive information such as environment variables.
CVE-2001-0013	Format string vulnerability in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges.
CVE-2001-0497	dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates.
CAN-1999-1499	named in ISC BIND 4.9 and 8.1 allows local users to destroy files via a symlink attack on (1) named_dump.db when root kills the process with a SIGINT, or (2) named.stats when SIGIOT is used.
CAN-2002-0400	ISC BIND 9 before 9.2.1 allows remote attackers to cause a denial of service (shutdown) via a malformed DNS packet that triggers an error condition that is not properly handled when the rdataset parameter to the dns_message_findtype() function in message.c is not NULL.
CAN-2002-0651	Buffer overflow in the DNS resolver code used in libc, glibc, and libbind, as derived from ISC BIND, allows remote malicious DNS servers to cause a denial of service and possibly execute arbitrary code via the stub resolvers.
CAN-2002-0684	Buffer overflow in DNS resolver functions that perform lookup of network names and addresses, as used in BIND 4.9.8 and ported to glibc 2.2.5 and earlier, allows remote malicious DNS servers to execute arbitrary code through a subroutine used by functions such as getnetbyname and getnetbyaddr.

Any of these could be used against a vulnerable bind server. The attacker is looking for a version number to see if they can exploit one of these vulnerabilities.

2.2.8 Attack Mechanism

This detect is more of a scan than an attack. However, if a bind server has not been properly configured to not give out version information then it will be susceptible to any number of different attacks from the Internet.

2.2.9 Correlations

A check of http://www.dshield.org shows that the IP address 203.107.138.81 has an attack record of 20 times.

A number of CVE numbers are listed for various vulnerabilities to bind. Checking for the version number of bind is a well-known reconnaissance method.

2.2.10 Evidence of Active Targeting

There is no evidence of active targeting by this host. The destination addresses appear to be randomly selected. There is some correlation between times of the packets as they generally appear at the hour and 30 minutes after the hour. These times however, are pretty well distributed so this is unlikely something being run on a time basis through CRON or the AT service on windows.

2.2.11 Severity

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

Criticality : 4 – A successful attack on your name server is a very serious problem. It can lead to further problems thru the use of cached entries.

Lethality : 3 – Since this is not an attack the lethality of this scan is deemed to be lower.

System Countermeasures : 5 – There is no evidence that any of the systems responded to this scan in any way.

Network Countermeasures : 1 – There is no blocking of these packets at the network level leading one to believe that they are simply allowed to propagate on the network.

Severity = (4 + 3) – (5 + 1) = 1

2.2.12 Defensive Recommendation

The best defense against a bind attack is to configure the named or bind server on your DNS hosts to not provide version numbers in any of their conversations. Another good technique is to always keep your version of bind up to date with the latest patches.

2.2.13 Multiple Choice Test Question

The following Tcpdump line would be evidence of what kind of attack?
01:24:08.654488 203.107.138.81.1610 > 46.5.25.205.53: [bad udp cksum faf7!] 4660 [b2&3=0x80] TXT CHAOS? version.bind. (30) (ttl 45, id 30843, len 58, bad cksum 7fae!)

A. A buffer overflow attacking bind
B. A web server exploit
C. A scan looking for exploitable bind servers
D. A zone transfer from a bind server

The correct answer is C.

2.3 Detect #3 – WEB-IIS ..\.. access

2.3.1 TCPDUMP of Detect

```
06:23:17.694488 210.85.160.200.2399 > 46.5.181.185.80: P [bad tcp cksum db5f!] 907527789:907527906(117) ack
2403715340 win 17520 (DF) (ttl 109, id 61385, len 157, bad cksum ccba!)
0x0000   4500 009d efc9 4000 6d06 ccba d255 a0c8        E.....@.m....U..
0x0010   2e05 b5b9 095f 0050 3617 c66d 8f45 c90c        ....._.P6..m.E..
0x0020   5018 4470 b965 0000 4745 5420 2f5f 7674        P.Dp.e..GET./_vt
0x0030   695f 6269 6e2f 2e2e 2535 632e 2e2f 2e2e        i_bin/..%5c../..
0x0040   2535 632e 2e2f 2e2e 2535 632e 2e2f 7769        %5c../..%5c../wi
0x0050   6e6e 742f 7379 7374 656d 3332 2f63 6d64        nnt/system32/cmd
0x0060   2e65 7865 3f2f 632b 6469 7220 632b 6469        .exe?/c+dir.c+di
0x0070   7220 4854 5450 2f31 2e30 0d0a 486f 7374        r.HTTP/1.0..Host
0x0080   3a20 7777 770d 0a43 6f6e 6e6e 6563 7469        :.www..Connnecti
0x0090   6f6e 3a20 636c 6f73 650d 0a0d 0a        on:.close....

06:23:29.644488 210.85.160.200.2399 > 46.5.181.185.80: P [bad tcp cksum db5f!] 0:117(117) ack 1 win 17520 (DF) (ttl
```

```
109, id 63642, len 157, bad cksum c3e9!)
0x0000    4500 009d f89a 4000 6d06 c3e9 d255 a0c8        E.....@.m....U..
0x0010    2e05 b5b9 095f 0050 3617 c66d 8f45 c90c        ....._.P6..m.E..
0x0020    5018 4470 b965 0000 4745 5420 2f5f 7674        P.Dp.e..GET./_vt
0x0030    695f 6269 6e2f 2e2e 2535 632e 2e2f 2e2e        i_bin/..%5c../..
0x0040    2535 632e 2e2f 2e2e 2535 632e 2e2f 7769        %5c../..%5c../wi
0x0050    6e6e 742f 7379 7374 656d 3332 2f63 6d64        nnt/system32/cmd
0x0060    2e65 7865 3f2f 632b 6469 7220 632b 6469        .exe?/c+dir.c+di
0x0070    7220 4854 5450 2f31 2e30 0d0a 486f 7374        r.HTTP/1.0..Host
0x0080    3a20 7777 770d 0a43 6f6e 6e6e 6563 7469        :.www..Connnecti
0x0090    6f6e 3a20 636c 6f73 650d 0a0d 0a       on:.close....
```

2.3.2 Snort Dump of Detect

```
[**] [1:974:3] WEB-IIS .... access [**]
[Classification: Web Application Attack] [Priority: 1]
07/17-06:23:17.694488 210.85.160.200:2399 -> 46.5.181.185:80
TCP TTL:109 TOS:0x0 ID:61385 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x3617C66D  Ack: 0x8F45C90C  Win: 0x4470  TcpLen: 20
[Xref => http://www.securityfocus.com/bid/2218]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0229]

[**] [1:974:3] WEB-IIS .... access [**]
[Classification: Web Application Attack] [Priority: 1]
07/17-06:23:29.644488 210.85.160.200:2399 -> 46.5.181.185:80
TCP TTL:109 TOS:0x0 ID:63642 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x3617C66D  Ack: 0x8F45C90C  Win: 0x4470  TcpLen: 20
[Xref => http://www.securityfocus.com/bid/2218]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0229]
```

2.3.3 Snort Rule for Detect

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ..\.. access";flags: A+; content:"|2e2e5c2e2e|";
reference:bugtraq,2218; reference:cve,CAN-1999-0229; classtype:web-application-attack; sid:974; rev:3;)
```

2.3.4 Source of Trace

The source of this trace was from the file http://www.incidents.org/logs/raw/2002.6.17 which was downloaded on September 17, 2002. The binary tcpdump file that was used was then further analyzed using tcpdump and snort.

2.3.5 Detect Generated By

This detect was generated by Snort v1.9-dev (Build 92) running on Redhat Linux 7.3.

The snort command used to generate the above snort alert was:

Snort –c /etc/snort/snort.conf –r 2002.6.17

Once snort was run on the binary file the alert file located in the log directory was examined using an editor for events of interest. The next step was to generate and examine the packet to try and determine if it was really an attack attempt or a false positive. To do this tcpdump was run using the following command:

Tcpdump –vv –n –r 2002.6.17 host 210.85.160.200 > 2002617.txt

The file was then examined and determined to be an attack due to the use of the windows backslash surrounded by parent directory periods. Also, it was deemed to likely be an attack since the ultimate goal of the url seems to be running \winnt\system32\cmd.exe.

2.3.6 Probability the Source Address was spoofed

The likelihood that the source address was spoofed in this case is little to none since the objective of the attack was to get a command prompt. In this case a little further investigation into the source address is warranted so a quick visit to http://www.apnic.net to determine who owns the address shows the following information:

```
inetnum:        210.85.0.0 - 210.85.255.255
netname:        ETWEBS-TW
descr:          ETWebs Taiwan Co. Ltd.
descr:          Taiwan Cable Modem Service Provider
descr:          Taiwan CATV operator
descr:          General Internet Service Provider
country:        TW
admin-c:        HE6-AP
tech-c:         HE6-AP
mnt-by:         APNIC-HM
mnt-lower:      MAINT-AP-ETWEBS
changed:        hostmaster@apnic.net 20010510
status:         ALLOCATED PORTABLE
source:         APNIC

person:         Hostmaster ETWebs
address:        1F, No.108, Zuikuang Rd., Neihu Dist., Taipei, Taiwan, R.O.C.
country:        TW
phone:          +886-2-87921111
fax-no:         +886-2-87920000
e-mail:         admin@ethome.net
nic-hdl:        HE6-AP
mnt-by:         MAINT-AP-ETWEBS
changed:        admin@ethome.net 20010511
source:         APNIC
```

Since this is a cable modem provider it most likely is from one of their customers. It would be recommended to send the ISP a note showing the attack attempt and ask them to investigate.

2.3.7 Description of the Attack

This attack is an attempt to gain administrator access to a system using the command processor in Windows NT/2000. It is possible that the tool used in this case includes the use of the Whisker anti-ids tool that was produced by Rain Forest Puppy. It is also possible that this is a scan from an infected Nimda host.

A check of http://cve.mitre.org reveals that this has an entry dating back to 2001.

Name	Description

CVE-2001-0333	Directory traversal vulnerability in IIS 5.0 and earlier allows remote attackers to execute arbitrary commands by encoding .. (dot dot) and "\" characters twice.

2.3.8 Attack Mechanism

The attack attempts to use the _vti_bin processor and use it to execute \winnt\system32\cmd.exe, which is the command line processor for Windows NT/2000. It attempts to use a method depicted in Rain Forest Puppy's Whisker to avoid detection by the IDS.

According to the CVE entry it targets Microsoft Windows Internet Information Server 5.0 (IIS 5.0) or earlier web servers. A check for other references from the CVE database reveals that this is a common CGI attack that is commonly used by Nimda and its variants to gain access to systems.

2.3.9 Correlations

A directory traversal attack is a well-known attack that is quite prevalent in the wilds of the Internet. The web site at http://cve.mitre.org has numerous entries on the exploits of directory transversal issues with numerous web servers. All of these exploits are against Windows based systems however. Looking closely at the cve.mitre.org site one finds that CVE-2001-0333 seems to be the most likely vulnerability in this case.

In searching through previous practicals this seems to be the first occurrence of this attack. This is not completely unexpected as it is probably based somewhat on the Nimda virus which is of recent origin.

2.3.10 Evidence of Active Targeting

There is no evidence of active targeting in this attack. It simply looks like a scan looking for a vulnerable web server.

2.3.11 Severity

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

Severity : 3 – This looks to be a scan that is in search of a vulnerable IIS server. It does not appear to be targeted at our main public web server which appears to be at 46.5.180.133.

Lethality : 5 – If this was successful then the ramifications would be total exposure of the system to whatever the hacker wanted to do.

System Countermeasures : 4 – The actual patch levels of this system are unknown. If the system is running a web server is even unknown. However, if the system is running a

web server there was no evidence that the system reacted to the attempt so it is possible that the system is patched against this attack.

Network Countermeasures : 2 – The Snort IDS system did pick up on the fact that the directory traversal was occurring but did not pick up on the cmd.exe attempt. Its possible that a better rule could be developed for Snort to detect this attack in a more precise fashion.

Severity = (3 + 5) – (4 + 2) = 2

2.3.12 Defensive Recommendation

The best defense against this attack is to make sure that all of your Windows Web servers are appropriately patched against directory transversal attacks. Since this attack is possibly Nimda based it is recommended that you follow the procedures for securing against this virus which can be found on http://www.nai.com or other similar sites from Trend Micro and others.

2.3.13 Multiple Choice Test Question

A directory traversal attack could be evidence of what virus?

A. El Kern
B. Nimda
C. Mellissa
D. Funlove

The correct answer is B.

2.4 References

"CodeRed Virus" Network Associates URL: http://vil.nai.com/vil/content/v_99142.htm (September 16, 2002)

"CodeRed Virus .C Variant" Network Associates URL: http://vil.nai.com/vil/content/v_99177.htm (October 15, 2002)
"Nimda Virus" Network Associates URL: http://vil.nai.com/vil/content/v_99209.htm (September 17, 2002)

Snort Users Manual – Snort Release: 1.9.x. URL: http://www.snort.org/docs/writing_rules-1.9.0 (September 17, 2002).

TCPDUMP Users Manual URL: http://www.tcpdump.org/tcpdump_man.html (September 15, 2002)

"Directory Traversal Exploit" Common Vulnerabilities and Exposures URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0333 (September 18, 2002)

"DNS Named Version Attempt" Whitehats.com URL: http://www.whitehats.com/info/ids278 (September 13, 2002)

"ISAPI .ida attempt" Whitehats.com URL: http://www.whitehats.com/info/ids552 (September 16, 2002)

3.0 Analyze This

3.1 Executive Summary

In general the University network system is in good shape. There are however a few things that need to be looked into and done to improve it. The first thing that showed up in the analysis was an overwhelming number of alerts and scans that were due to one machine being infected with the Nimda virus. Obviously this host needs to be looked into immediately and rebuilt and cleaned of the virus, but then something needs to be done to prevent the various viruses like Nimda and CodeRed from infecting hosts on the campus network.

The network seems to be well supported by an Intrusion Detection System that has been fairly well tuned to the Universities needs, including the inclusion of some custom or local rules developed for the University. It identifies some of the more prevalent viruses, attacks and scans efficiently. There was some trouble with the log files overwriting and/or skipping information when the Snort system got very busy dealing with the host that was infected with Nimda. It might be a good idea to change the logging of the Snort processor to log binary files instead of text logs. This would allow the Snort processor to log more efficiently.

A security awareness program should be developed to train and teach the various campus users of computer systems how to protect themselves. In addition to this a program to cheaply deploy anti-virus and personal firewall technology should be investigated and implemented campus wide.

A program to close down any TFTP services, RPC services and POP mail services should be developed and implemented across the campus. These services were attacked or scanned a good deal during the analysis. Mostly these services should be turned off across the campus except where there is a good reason not to.

3.2 List of Files Analyzed

The files analyzed for the GIAC University were a representative sample from 5 days worth of Snort logs including the Alert logs, the Portscan logs and the Out of Spec packet logs. A list of the files appears below:

Alerts	Scans	Out of Specs
Alert.020801.gz	Scans.020801.gz	Oos_Aug.1.2002.gz
Alert.020802.gz	Scans.020802.gz	Oos_Aug.2.2002.gz
Alert.020803.gz	Scans.020803.gz	Oos_Aug.3.2002.gz
Alert.020804.gz	Scans.020804.gz	Oos_Aug.4.2002.gz
Alert.020805.gz	Scans.020805.gz	Oos_Aug.5.2002.gz

These files were obtained from the following link:

http://www.incidents.org/logs

3.3 Detects based on Number of Occurrences

The following tables show all of the alerts that were detected by the standard Snort ruleset that produced more than 10,000 alerts. We will in the following sections describe each one.

EOIs	Alert Description
877528	NIMDA – Attempt to execute cmd from campus host
494119	Spp_http_decode: IIS Unicode attack detected
482402	IDS552/web-iss_IIS ISAPI Overflow ida INTERNAL nosize
123305	NIMDA – Attempt to execute root from campus host
106883	UDP SRC and DST outside network
53562	Spp_http_decode: CGI Null Byte attack detected
30083	SMB Name Wildcard
24220	TFTP – External UDP connection to internal tftp server
14578	External RPC call
11921	Watchlist 000220 IL-ISDNNET-990517

Figure 3.3.1 – Detects from Alert Logs

EOIs	Scan Description
3112455	UDP Scan (Externally-based)
997435	SYN Scan (Externally-based)

Figure 3.3.2 – Detects from Scan Logs

EOIs	Source Address	Destination Address
652	68.32.126.64	MY.NET.6.7
241	62.76.241.129	MY.NET.97.217
104	62.76.241.129	MY.NET.97.238

Figure 3.3.3 – Detects from OOS Logs

3.3.1 – NIMDA - Attempt to execute cmd from campus host – 877,528 alerts

This alert is from a rule that fires off if the Nimda virus attempts to gain access to a system via the cmd.exe interface vulnerability. A total of 877,519 of these alerts are coming from a single host at 172.31.100.208 also known as moses.cs.university.edu. The other 9 packets from this alert are noise from a variety of systems around the network. As the alert says all the alerts from this rule are being caused by internal hosts on the 172.31.*.* network. It is recommended that the host moses.cs.university.edu be looked at closely for evidence of the Nimda virus. Also see 3.3.2 and 3.3.4 for further evidence that this host is infected with Nimda.

3.3.2 – Spp_http_decode: IIS Unicode attack detected – 494,119 alerts

This alert is from a rule that detects the Microsoft Internet Information Service Unicode attack. Of the 494,119 alerts that were logged on this rule a total of 436,235 came from the system at 172.31.100.208 also known as Moses.cs.university.edu. The Unicode attack is another vulnerability that can be used by a virus to try and spread to new hosts. So as stated above the system moses.cs.university.edu should be examined for evidence of a virus. The remaining 57,884 alerts logged by this rule are spread throughout the network. The table shown in Figure 3.3.2.1 shows some of the more active hosts that are causing this alert.

IP Address	FQDN	Alerts
172.31.10.86	gis16.university.edu	175
172.31.15.212	pplant104pc-12.university.edu	1141
172.31.111.196	trc102apc-04.university.edu	965
172.31.116.84	bs412pc-01.biosci.university.edu	426
172.31.143.107	fa466pc-02.university.edu	394
172.31.152.19	lib037pc50.ucslab.university.edu	2884
172.31.153.143	libstkpc05.libpub.university.edu	1854
172.31.153.145	libstkpc07.libpub.university.edu	2826
172.31.153.168	libstkpc23.libpub.university.edu	2002
172.31.85.74	acserv106fmac-01.university.edu	6981
80.137.90.34	p50895A22.dip.t-dialin.net	6888

Figure 3.3.2.1

Note that these hosts are not the only other hosts producing this alert but just some of the more active ones. These hosts should be checked for a virus just like the host moses.cs.university.edu. It is noticed that only one of these addresses comes from outside the University. The host at 80.137.90.34 seems to be a dial-up line, so these packets are either an attempt to gain access or a reflection of a host that is infected. Looking at this outside host further shows that it spent most of its time attacking the internal host 172.31.105.204 also known as umbbal01srv01.university.edu. This host should be checked for the appropriate service pack and hotfix, patch releases to protect against this vulnerability.

3.3.3 – IDS552/web-iss_IIS ISAPI Overflow ida INTERNAL nosize – 482,402 alerts

By the reference to IDS552 in this alert title it is assumed that it is a reference to the IDS numbering scheme in place at whitehats.com. Looking at http://www.whitehats.com/info/ids552 shows us that the vulnerability in question is the famous one for a buffer overflow problem with the Index Server under Windows. This vulnerability is exploited by viruses such as CodeRed, CodeBlue, Nimda, etc… The main thing to notice about this set of alerts however is that all of the alerts are being generated

by one host at 172.31.84.234 also known as engr-84-234.pooled.university.edu. It is recommended that this host be looked at for the existence of a Trojan program or an infection of one of the above viruses.

3.3.4 – NIMDA – Attempt to execute root from campus host – 123,305 alerts

Another aspect of the Nimda virus is to attempt exploit a vulnerability in the IIS server that allows it to execute a file called root.exe. This alert set is obviously alarming that a campus host is attempting to gain access to other hosts by using this vulnerability. In review of the alert logs it was determined that all the traffic that is generating this alert is coming from our already infected with Nimda host of 172.31.100.208 which is also known as moses.cs.university.edu.

3.3.5 – UDP SRC and DST outside network – 106,883 alerts

Some interesting things are evident from an analysis of this alert. The first thing that was found was that the majority of the alerts were being caused by a host at 3.0.0.99 on port 137 talking to a host at 10.0.0.1 on port 137. A total of 51,358 of the 106,883 alerts were generated by this traffic. The most likely cause of this traffic looks to be communication between windows hosts on a poorly configured host at 3.0.0.99. This address should be tracked down on the network and the physical device identified and rectified.

Another interesting thing that was noticed during the course of the analysis of this set of alerts is that a good portion of the alerts not from 3.0.0.99 are going to a host at 233.2.171.1 which does not translate to an FQDN name but doesn't look to be a host on this network. In further investigation the 233 network is reserved for multicast. So this appears to be outside host multicast traffic at our sensor point. It might be a good idea to review the setup of our perimeter routers to verify that they are not setup to pass outside multicast traffic onto the network. This is most likely not local traffic because the source addresses for these alerts are all on the outside of our network space of 172.31.*.*.

3.3.6 – Spp_http_decode: CGI Null Byte attack detected – 53,562 alerts

This alert generated an interesting pattern of destination addresses. The majority of the alerts were directed at three outside addresses including:

IP Address	FQDN	Alerts
152.163.210.84	sand-int-v11.ptn.aol.com	6,138
216.241.219.28	Not Translatable	39,484
209.10.239.135	Not Translatable	3,631

The most interesting thing about this set of alerts is that they almost all originate from the inside of the network and attack hosts on the outside of the network. This could be a specific user on the University network abusing their access by trying to attack outside

hosts. With this in mind the internal IP addresses should be investigated and the users tracked down and reminded or reprimanded on the use of the acceptable use policies of the University. A better view of these connections can be seen in section 3.6 of this document under Link Graph.

This attack uses a method that was put forward by Rain Forest Puppy in his Whisker Anti-IDS techniques document, which can be found at:

http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html

The basic premise of this attack revolves around the fact that programs that have been written in the 'C' programming language use the Null byte character to terminate strings. It uses this fact to try and hide attempts to gain access to the system by imbedding null bytes into the URL request so that an IDS system would not see the attempt as an attack. Fortunately most IDS systems now detect this form of attack.

It is possible that these are false positives. Some sites send urlencoded binary data inside cookies that can cause this to detect.

3.3.7 – SMB Name Wildcard – 30,083 alerts

The SMB Name Wildcard scan is a well known scan which can either be legitimate traffic from Windows machines that are looking for shares via Windows Explorer. They can also be caused by using the command NBTSTAT –A <Target IP Address>. With the notable exception of some traffic coming from 192.168.0.0/24 all of this traffic originates on the outside of the network. A good portion of the traffic is destined for just a few IP addresses on the inside of the network as shown in Figure 3.3.7.1. The rest of the traffic seems to be randomly destined for other hosts on the Universities network.

IP Address	FQDN	Alerts
172.31.104.204	pooled-104-204.university.edu	2,737
172.31.110.224	ecs335pc02.cs.university.edu	1,396
172.31.117.137	ucommons-117-137.pooled.university.edu	2,762
172.31.163.107	physics105pc-01.university.edu	2,392
172.31.179.77	dinosaur.university.edu	761
172.31.198.204	rwd-204.university.edu	1,165

Figure 3.3.7.1

In researching this detect is most likely caused either by hackers on the Internet scanning our network using the NBTSTAT command, or because of a worm called network.vbs. The worm network.vbs was originally discovered back in 2000 and well documented by Abe Singer of the San Diego Supercomputer Center. Basically the network.vbs worm simply attempts to keep track of vulnerable machines. The author can then go back to this log file and get a list of vulnerable machines that the worm has found for later attacks. The system can then be taken over by loading a Trojan onto the system like BackOrifice

or WinTrin00 using the same vulnerability that allowed the network.vbs worm to infect the system.

3.3.8 – TFTP – External UDP connection to internal tftp server – 24,220 alerts

This detect shows numerous connections during the entire time span of the analysis where all packets are going to a host at 192.168.0.216. This is an internal only address that should not be routed to the outside network so we can assume that this traffic is probably all locally based at the University. In analyzing the detect it was determined that 4 systems were the main contributors to the traffic. These hosts along with the number of detects attributable to each can be found in Figure 3.3.8.1

IP Address	FQDN	Alerts
172.31.109.105	No Translation	6053
172.31.111.219	c00074.university.edu	6007
172.31.111.230	c00004.university.edu	6090
172.31.111.231	c00056.university.edu	6059

Figure 3.3.8.1

It must be noted that these systems appear to be linked in some manner since the detects generated all used a source port of 69 and a varying destination port. The destination port was somehow coordinated such that all four systems shown above were using the same destination port at approximately the same time. This could be the result of a mini denial of service attack on whatever host is at 192.168.0.216. Another possible reason for this traffic is an attempt on the part of Nimda to TFTP the virus from these hosts down to the host at 192.168.0.216. If this is the case then the University should determine the physical location of the system at 192.168.0.216 and check it for evidence of the Nimda virus.

3.3.9 – External RPC call – 14,578 alerts

This alert is from a custom rule in the Universities rule base. It obviously is written to alert whenever an external packet is destined for port 111 on an internal host. The alerts generated due to this rule show that there are 4 main external hosts trying to gain access to the remote procedure call service of numerous internal hosts. The list of hosts found in Figure 3.3.9.1 shows the hosts trying to attack over this mechanism.

IP Address	FQDN	Alerts
194.98.189.139	No Translation	8352
202.108.109.100	No Translation	774
203.239.159.2	Woonkyung.or.kr	918
61.182.50.241	No Translation	4519

Figure 3.3.9.1

There are many exploits and denial of service attacks that are based on the ability to attack an RPC server. A quick look at http://cve.mitre.org reveals the following 69 possible

vulnerabilities and exploits:

Name	Description
CVE-1999-0003	Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd)
CVE-1999-0008	Buffer overflow in NIS+, in Sun's rpc.nisd program
CVE-1999-0208	rpc.ypupdated (NIS) allows remote users to execute arbitrary commands.
CVE-1999-0212	Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.
CVE-1999-0228	Denial of service in RPCSS.EXE program (RPC Locator) in Windows NT.
CVE-1999-0320	SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.
CVE-1999-0353	rpc.pcnfsd in HP gives remote root access by changing the permissions on the main printer spool directory.
CVE-1999-0493	rpc.statd allows remote attackers to forward RPC calls to the local operating system via the SM_MON and SM_NOTIFY commands, which in turn could be used to remotely exploit other bugs such as in automountd.
CVE-1999-0687	The ToolTalk ttsession daemon uses weak RPC authentication, which allows a remote attacker to execute commands.
CVE-1999-0696	Buffer overflow in CDE Calendar Manager Service Daemon (rpc.cmsd)
CVE-1999-0817	Lynx WWW client allows a remote attacker to specify command-line parameters which Lynx uses when calling external programs to handle certain protocols, e.g. telnet.
CVE-1999-0900	Buffer overflow in rpc.yppasswdd allows a local user to gain privileges via MD5 hash generation.
CVE-1999-0969	The Windows NT RPC service allows remote attackers to conduct a denial of service using spoofed malformed RPC packets which generate an error message that is sent to the spoofed host, potentially setting up a loop, aka Snork.
CVE-1999-0974	Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad service.
CVE-1999-1127	Windows NT 4.0 does not properly shut down invalid named pipe RPC connections, which allows remote attackers to cause a denial of service (resource exhaustion) via a series of connections containing malformed data, aka the "Named Pipes Over RPC" vulnerability.
CVE-1999-1258	rpc.pwdauthd in SunOS 4.1.1 and earlier does not properly prevent remote access to the daemon, which allows remote attackers to obtain sensitive system information.
CVE-1999-1481	Squid 2.2.STABLE5 and below, when using external authentication, allows attackers to bypass access controls via a newline in the user/password pair.
CVE-2000-0277	Microsoft Excel 97 and 2000 does not warn the user when executing Excel Macro Language (XLM) macros in external text files, which could allow an attacker to execute a macro virus, aka the "XLM Text Macro" vulnerability.
CVE-2000-0289	IP masquerading in Linux 2.2.x allows remote attackers to route UDP packets through the internal interface by modifying the external source IP address and port number to match those of an established connection.
CVE-2000-0466	AIX cdmount allows local users to gain root privileges via shell metacharacters.
CVE-2000-0508	rpc.lockd in Red Hat Linux 6.1 and 6.2 allows remote attackers to cause a denial of service via a malformed request.
CVE-2000-0771	Microsoft Windows 2000 allows local users to cause a denial of service by corrupting the local security policy via malformed RPC traffic, aka the "Local Security Policy Corruption" vulnerability.
CVE-2000-0816	Linux tmpwatch --fuser option allows local users to execute arbitrary commands by creating files whose names contain shell metacharacters.
CVE-2000-0861	Mailman 1.1 allows list administrators to execute arbitrary commands via shell metacharacters in the %(listname) macro expansion.
CVE-2000-1027	Cisco Secure PIX Firewall 5.2(2) allows remote attackers to determine the real IP address of a target FTP server by flooding the server with PASV requests, which includes the real IP address in the response when passive mode is established.
CVE-2001-0331	Buffer overflow in Embedded Support Partner (ESP) daemon (rpc.espd) in IRIX 6.5.8 and earlier allows remote attackers to execute arbitrary commands.
CVE-2001-0662	RPC endpoint mapper in Windows NT 4.0 allows remote attackers to cause a denial of service (loss of RPC services) via a malformed request.
CVE-2001-0717	Format string vulnerability in ToolTalk database server rpc.ttdbserverd allows remote attackers to execute arbitrary commands via format string specifiers that are passed to the syslog function.
CVE-2001-0731	Apache 1.3.20 with Multiviews enabled allows remote attackers to view directory contents and bypass the index page via a URL containing the "M=D" query string.
CVE-2001-0779	Buffer overflow in rpc.yppasswdd (yppasswd server) in Solaris 2.6, 7 and 8 allows remote attackers to gain root access via a long username.
CAN-1999-0078	pcnfsd (aka rpc.pcnfsd) allows local users to change file permissions, or execute arbitrary commands through arguments in the RPC call.
CAN-1999-0195	Denial of service in RPC portmapper allows attackers to register or unregister RPC services or spoof RPC services using a spoofed source IP address such as 127.0.0.1.
CAN-1999-0528	A router or firewall forwards external packets that claim to come from inside the network that the router/firewall is in front of.
CAN-1999-0568	rpc.admind in Solaris is not running in a secure mode.
CAN-1999-0613	The rpc.sprayd service is running.

CAN-1999-0625	The rpc.rquotad service is running.
CAN-1999-0632	The RPC portmapper service is running.
CAN-1999-0795	The NIS+ rpc.nisd server allows remote attackers to execute certain RPC calls without authentication to obtain system information, disable logging, or modify caches.
CAN-1999-1225	rpc.mountd on Linux, Ultrix, and possibly other operating systems, allows remote attackers to determine the existence of a file on the server by attempting to mount that file, which generates different error messages depending on whether the file exists or not.
CAN-1999-1549	Lynx 2.x does not properly distinguish between internal and external HTML, which may allow a local attacker to read a "secure" hidden form value from a temporary file and craft a LYNXOPTIONS: URL that causes Lynx to modify the user's configuration file and execute commands.
CAN-1999-1558	Vulnerability in loginout in Digital OpenVMS 7.1 and earlier allows unauthorized access when external authentication is enabled.
CAN-2000-0114	Frontpage Server Extensions allows remote attackers to determine the name of the anonymous account via an RPC POST request to shtml.dll in the /_vti_bin/ virtual directory.
CAN-2000-0544	Windows NT and Windows 2000 hosts allow a remote attacker to cause a denial of service via malformed DCE/RPC SMBwriteX requests that contain an invalid data length.
CAN-2000-0800	String parsing error in rpc.kstatd in the linuxnfs or knfsd packages in SuSE and possibly other Linux systems allows remote attackers to gain root privileges.
CAN-2000-1009	dump in Red Hat Linux 6.2 trusts the pathname specified by the RSH environmental variable, which allows local users to obtain root privileges by modifying the RSH variable to point to a Trojan horse program.
CAN-2001-0509	Vulnerabilities in RPC servers in (1) Microsoft Exchange Server 2000 and earlier, (2) Microsoft SQL Server 2000 and earlier, (3) Windows NT 4.0, and (4) Windows 2000 allow remote attackers to cause a denial of service via malformed inputs.
CAN-2001-1124	rpcbind in HP-UX 11.00, 11.04 and 11.11 allows remote attackers to cause a denial of service (core dump) via a malformed RPC portmap requests, possibly related to a buffer overflow.
CAN-2001-1135	ZyXEL Prestige 642R and 642R-I routers do not filter the routers' Telnet and FTP ports on the external WAN interface from inside access, allowing someone on an internal computer to reconfigure the router, if the password is known.
CAN-2001-1272	wmtv 0.6.5 and earlier does not properly drop privileges, which allows local users to execute arbitrary commands via the -e (external command) option.
CAN-2001-1279	Buffer overflow in print-rx.c of tcpdump 3.x (probably 3.6x) allows remote attackers to cause a denial of service and possibly execute arbitrary code via AFS RPC packets with invalid lengths that trigger an integer signedness error, a different vulnerability than CVE-2000-1026.
CAN-2001-1293	Buffer overflow in web server of 3com HomeConnect Cable Modem External with USB (#3CR29223) allows remote attackers to cause a denial of service (crash) via a long HTTP request.
CAN-2002-0039	rpcbind in SGI IRIX 6.5 through 6.5.15f, and possibly earlier versions, allows remote attackers to cause a denial of service (crash) via malformed RPC packets with invalid lengths.
CAN-2002-0085	cachefsd in Solaris 2.6, 7, and 8 allows remote attackers to cause a denial of service (crash) via an invalid procedure call in an RPC request.
CAN-2002-0234	NetScreen ScreenOS before 2.6.1 does not support a maximum number of concurrent sessions for a system, which allows an attacker on the trusted network to cause a denial of service (resource exhaustion) via a port scan to an external network, which consumes all available connections.
CAN-2002-0357	Vulnerability in rpc.passwd in the nfs.sw.nis subsystem of SGI IRIX 6.5.15 and earlier allows local users to gain root privileges.
CAN-2002-0359	xfsmd for IRIX 6.5 through 6.5.16 uses weak authentication, which allows remote attackers to call dangerous RPC functions, including those that can mount or unmount xfs file systems, to gain root privileges.
CAN-2002-0391	Integer overflow in xdr_array function in RPC servers for operating systems that use libc, glibc, or other code based on SunRPC including dietlibc, allows remote attackers to execute arbitrary code by passing a large number of arguments to xdr_array through RPC services such as rpc.cmsd and dmispd.
CAN-2002-0567	Oracle 8i and 9i with PL/SQL package for External Procedures (EXTPROC) allows remote attackers to bypass authentication and execute arbitrary functions by using the TNS Listener to directly connect to the EXTPROC process.
CAN-2002-0573	Format string vulnerability in RPC wall daemon (rpc.rwalld) for Solaris 2.5.1 through 8 allows remote attackers to execute arbitrary code via format strings in a message that is not properly provided to the syslog function when the wall command cannot be executed.
CAN-2002-0586	Format string vulnerability in Ns_PdLog function for the external database driver proxy daemon library (libnspd.a) of AOLServer 3.0 through 3.4.2 allows remote attackers to execute arbitrary code via the Error or Notice parameters.
CAN-2002-0587	Buffer overflow in Ns_PdLog function for the external database driver proxy daemon library (libnspd.a) of AOLServer 3.0 through 3.4.2 allows remote attackers to cause a denial of service or execute arbitrary code via the Error or Notice parameters.
CAN-2002-0652	xfsmd for IRIX 6.5 through 6.5.16 allows remote attackers to execute arbitrary code via shell metacharacters that are not properly filtered from several calls to the popen() function, such as export_fs().
CAN-2002-0677	CDE ToolTalk database server (ttdbserver) allows remote attackers to overwrite arbitrary memory locations with a zero, and possibly gain privileges, via a file descriptor argument in an AUTH_UNIX procedure call, which is used as a table index by the _TT_ISCLOSE procedure.
CAN-2002-0678	CDE ToolTalk database server (ttdbserver) allows local users to overwrite arbitrary files via a symlink attack on the transaction log file used by the _TT_TRANSACTION RPC procedure.
CAN-2002-0679	Buffer overflow in Common Desktop Environment (CDE) ToolTalk RPC database server (rpc.ttdbserverd) allows remote attackers to execute arbitrary code via an argument to the _TT_CREATE_FILE procedure.
CAN-2002-0763	Vulnerability in administration server for HP VirtualVault 4.5 on HP-UX 11.04 allows remote web servers or privileged external processes to bypass access restrictions and establish connections to the server.
CAN-2002-0830	Network File System (NFS) in FreeBSD 4.6.1 RELEASE-p7 and earlier, and possibly other operating systems, allows remote attackers to cause a denial of service (hang) via an RPC message with a zero length payload, which causes NFS to reference a previous payload and enter an infinite loop.
CAN-2002-1140	The Sun Microsystems RPC library Services for Unix 3.0 Interix SD, as implemented on Microsoft Windows NT4, 2000, and XP, allows remote attackers to cause a denial of service (service hang) via malformed packet fragments, aka "Improper parameter size check leading to denial of service."

CAN-2002-1141	An input validation error in the Sun Microsystems RPC library Services for Unix 3.0 Interix SD, as implemented on Microsoft Windows NT4, 2000, and XP, allows remote attackers to cause a denial of service via malformed fragmented RPC client packets, aka "Denial of service by sending an invalid RPC request."

One thing the University should consider is blocking the use of RPC ports from all external hosts. If there are specific needs to allow external hosts to mount NFS partitions or to perform some other function that requires RPC then the firewall can be opened for those systems individually.

3.3.10 – Watchlist 000220 IL-ISDNNET-990517 – 11,921 alerts

This set of alerts is most likely caused by the source address of the packets being on a watchlist that the University has setup. The source address for these packets all come from the 212.179.*.* network. Two of these addresses involved are 212.179.101.118 and 212.179.27.6. These addresses translate to cablep-179-101-118.cablep.bezeqint.net and clnt-27006.bezeqint.net respectively. A quick look at www.ripe.net's whois database reveals the following information for the two hosts:

```
% This is the RIPE Whois secondary server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      212.179.100.0 - 212.179.124.255
netname:       L2TP-PROJECT
mnt-by:        INET-MGR
descr:         1st-pool-Dailup-L2TP-client
country:       IL
admin-c:       MR916-RIPE
tech-c:        ZV140-RIPE
status:        ASSIGNED PA
notify:        hostmaster@bezeqint.net
changed:       hostmaster@bezeqint.net 20020917
source:        RIPE
route:        212.179.64.0/18
descr:         ISDN Net Ltd.
origin:        AS8551
notify:        hostmaster@bezeqint.net
mnt-by:        AS8551-MNT
changed:       hostmaster@bezeqint.net 20020618
source:        RIPE
person:       Miri Roaky
address:       Bezeq Inernational
address:       hashacham 40
address:       Petach Tiqua
address:       Israel
phone:         +972-3-9203010
phone:         +972-3-9203005
e-mail:        hostmaster@bezeqint.net
nic-hdl:       MR916-RIPE
changed:       hostmaster@bezeqint.net 20020502
source:        RIPE
person:       Zehavit Vigder
address:       bezeq-international
address:       40 hashacham
address:       petach tikva 49170 Israel
phone:         +972 52 770145
fax-no:        +972 9 8940763
e-mail:        hostmaster@bezeqint.net
nic-hdl:       ZV140-RIPE
```

```
changed:      zehavitv@bezeqint.net 20000528
source:       RIPE

% This is the RIPE Whois secondary server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      212.179.27.4 - 212.179.27.7
netname:      ADI-ASSOCIATION
descr:        ADI-ASSOCIATION-SERIAL
country:      IL
admin-c:      NP469-RIPE
tech-c:       NP469-RIPE
status:       ASSIGNED PA
notify:       hostmaster@isdn.net.il
mnt-by:       RIPE-NCC-NONE-MNT
changed:      hostmaster@isdn.net.il 20000106
source:       RIPE
route:        212.179.0.0/18
descr:        ISDN Net Ltd.
origin:       AS8551
notify:       hostmaster@bezeqint.net
mnt-by:       AS8551-MNT
changed:      hostmaster@bezeqint.net 20020618
source:       RIPE
person:       Nati Pinko
address:      Bezeq International
address:      40 Hashacham St.
address:      Petach Tikvah  Israel
phone:        +972 3 9257761
e-mail:       hostmaster@isdn.net.il
nic-hdl:      NP469-RIPE
changed:      registrar@ns.il 19990902
source:       RIPE
```

Looking at this information reveals that the attack is coming from Israel. This is obviously a network that a local rule has been developed to trigger on for systems on this particular network. It's possible that Bezeq is an ISP that resells IP address space for the state of Israel. It is impossible to conjecture as to why the University wants to track this traffic but obviously it is important to know when this network is accessing University resources. A good majority of these alerts had a destination port of 1214. In doing some research on this, the port was found to have a possible link to the Kazaa file-sharing program. Therefore these alerts are most likely somebody in Israel trying to share files with local hosts on the University network.

3.3.11 – UDP scan (Externally-based) – 3,112,455 packets

In looking at the UDP Scans that were present in the scans files it appears that 4 hosts are producing the majority of the noise on the network. The table in Figure 3.3.11.1 shows the hosts and the number of events that each has caused.

IP Address	FQDN	Alerts
172.31.70.200	calamari.ucs.university.edu	2,437,158
172.31.70.207	ecs020pc09.university.edu	137,225

172.31.82.2	oit-82-02.pooled.university.edu	127,724
172.31.83.150	aciv-83-150.pooled.university.edu	90,039

Figure 3.3.11.1

As can be seen in the table the most frequent talker was the host at 172.31.70.200 also known as calamari.ucs.university.edu. This host should be checked for the existence of a virus or Trojan program as soon as possible. The packets from this host started on 8/2/2002 at approximately 9:32 am and continued throughout the scan time till 8/5/2002 at approximately 11:07 am. The port that it used was 4946 source with a destination port of 41170. A quick check of the online ports database reveals that the ChiliASP module uses the port 4946 for the Apache web server.

For the systems at 172.31.70.207 and 172.31.82.2 the ports used as a source port were pretty consistently 12203 and 12300. A quick check of the ports database online reveals that these ports have no special attachment.

The final system at 172.31.83.150 was found to be using a destination port of 6257. Another quick check of the ports database online reveals that this is an old port used by the WinMX file sharing program. This program allows for the sharing of files between systems sometimes even through a firewall. If this file sharing is against University policy then the IP address should be physically located and cleaned of this system.

3.3.12 – SYN scan (Externally-based) – 997,435 packets

There were two hosts that caused the majority of the Sync scan packets that were seen during the scanning period. The first one was 172.31.100.208, which a quick translation thru nslookup resolves to moses.cs.university.edu. This is the same host that we determined was probably infected with the Nimda virus earlier in our alert scan checks. An interesting destination port phenomenon found in the scans was that the majority of the scans are destined for port 80 but every once in a while a packet comes thru that is destined for either port 139 or 445 the Netbios ports. This reinforces the theory that this machine is infected with the Nimda virus because this pattern of packet scans fits the known infection paths that Nimda takes to infect and find new machines. It uses a web folder transversal error, which would explain the port 80 scans. It also tries to infect via using the old CodeRed Trojan ports which explains the Netbios scans.

The other host that is generating the majority of Sync scans is 172.31.84.234, which translates via nslookup to Engr-84-234.pooled.university.edu. It produced a total of 478,406 packets that were caught by the scan engine. All of these packets are destined to port 80 (http). This could indicate the existence of a couple of different viruses including CodeRed, CodeBlue and Nimda. If more detailed packet information was available then it might be possible to determine exactly which virus has infected this machine.

The remaining scans are being generated by outside hosts, but no significant single host shows up in the database. There is also no significant pattern for the destination host or

port numbers for these scans. Therefore it is believed that the remainder of these scans are most likely just normal port scanning levels constantly bombarding the University from the Internet.

3.3.13 – 68.32.126.64 – MY.NET.6.7

The system at 68.32.126.64 sent out of spec packets to MY.NET.6.7 a total of 652 times. The packets all were destined to port 110, which is the port for POP (Post Office Protocol). The out of specification reason is that the reserved bits are set for 0x21. In a normal packet these bits should not be set at all during the execution of a Sync packet. The chance that this is a mistake on the part of some TCP/IP stack is negligible. This is more likely either an attack against a vulnerable POP mail server or a reconnaissance scan by a tool such as Queso or NMAP. An example out of spec dump by Snort is shown below:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-00:03:02.100571 68.32.126.64:26052 -> MY.NET.6.7:110
TCP TTL:48 TOS:0x0 ID:432  DF
21S***** Seq: 0x1D18A45   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 51874805 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The attacking host is registered to Comcast Cable. Shown below is the Arin database entry for the offending host:

```
CustName:   Comcast Cable Communications, Inc.
Address:    3 Executive Campus Cherry Hill NJ 08002
Country:    US
RegDate:    2002-06-15
Updated:    2002-06-15

NetRange:   68.32.112.0 - 68.32.143.255
CIDR:       68.32.112.0/20, 68.32.128.0/20
NetName:    JUMPSTART-BALTIMOR-A3
NetHandle:  NET-68-32-112-0-1
Parent:     NET-68-32-0-0-1
NetType:    Reassigned
Comment:
RegDate:    2002-06-15
Updated:    2002-06-15
```

3.3.14 – 62.76.241.129 – MY.NET.97.217 & MY.NET.97.238

This detect was generated by the outside host at 62.76.241.129 and directed at two internal hosts at MY.NET.97.217 and MY.NET.97.238. The packets that were detected showed the reserved bits set to 0x21 and the sync flag set. They were destined for the Post Office Protocol port in both cases. This is most likely another scan for reconnaissance by a person using either Nmap or Queso port scanning software or something equivalent. Below are the packets detected by Snort and the Ripe database entry for the offending host.

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

```
08/01-00:25:36.030054 62.76.241.129:36622 -> MY.NET.97.217:113
TCP TTL:45 TOS:0x0 ID:39364  DF
21S***** Seq: 0x65BBACF8   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 59651388 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-09:12:43.536107 62.76.241.129:39304 -> MY.NET.97.238:113
TCP TTL:45 TOS:0x0 ID:46115  DF
21S***** Seq: 0x2BCEC2B2   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 62813841 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
inetnum:        62.76.240.0 - 62.76.243.255
netname:        UDMEDU-NET
descr:          Internet Center of Udmurt State University
descr:          ul. Universitetskaja, 1, k.6, Izhevsk, Russia
country:        RU
admin-c:        BGA4-RIPE
tech-c:         DMIR-RIPE
status:         ASSIGNED PA
notify:         dm@uni.udm.ru
mnt-by:         ROSNIIROS-MNT
changed:        ip-dbm@ripn.net 20000128
source:         RIPE
route:          62.76.240.0/22
descr:          UDMEDU-NET
origin:         AS13094
mnt-by:         UDSU-MNT
changed:        dm@uni.udm.ru 20010124
source:         RIPE
person:         Basil G. Ananin
address:        ul. Universitetskaja, 1, k.6, room 320
address:        Internet Center of UdSU
address:        Izhevsk, Russia
phone:          +7 3412788697
fax-no:         +7 3412788697
e-mail:         anan@uni.udm.ru
nic-hdl:        BGA4-RIPE
notify:         dm@uni.udm.ru
notify:         ip-reg@ripn.net
changed:        dm@uni.udm.ru 20000128
source:         RIPE
person:         Dmitry N. Mironov
address:        1, ul. Universitetskaja
address:        Izhevsk
address:        Russia
phone:          +7 3412 751758
fax-no:         +7 3412 788697
e-mail:         ddd@uni.udm.ru
nic-hdl:        DMIR-RIPE
notify:         ddd@uni.udm.ru
changed:        ddd@uni.udm.ru 19981020
source:         RIPE
```

3.4 Top Talkers

In the following Figure (3.4.1) is shown the top ten talkers found in the files that were analyzed. The basis for inclusion in this list is simply who had the most events of interest as denoted by the Snort IDS system. Some of the hosts show up in both the scans and

alerts databases. It is interesting to note that the two top talkers are both what appear to be servers. One seems to be in the Computer Science department and the other in a department that uses the domain name ucs.university.edu. Of particular note is that from a port scan standpoint all of the top talkers are within the University, which shows that there is not a concerted effort being waged from the outside trying to map the Universities network. The only top talker that is outside the University is from General Electric at address 3.0.0.99. This host was responsible for most of the UDP SRC and DST outside of network alerts.

EOIs	IP Address	FQDN
2439514	172.31.70.200	calamari.ucs.university.edu
1604128	172.31.100.208	Moses.cs.university.edu
959740	172.31.84.234	Engr-84-234.pooled.university.edu
137226	172.31.70.207	ecs020pc09.university.edu
127792	172.31.82.2	oit-82-02.pooled.university.edu
104553	172.31.165.24	slate.university.edu
90049	172.31.83.150	aciv-83-150.pooled.university.edu
51359	3.0.0.99	No Translation
49208	172.31.137.7	No translation
42744	172.31.70.133	kenny.ucs.university.edu

Figure 3.4.1 – Top Talkers

3.4.1 – 172.31.70.200 - calamari.ucs.university.edu

The host responsible for the most events of interest during the time period of the files that were analyzed was 172.31.70.200 also known as calamari.ucs.university.edu. This host produced over 2 million events most of which were UDP scans to outside addresses from port 4946 to port 41170. A quick check of the ports database at http://www.portsdb.org reveals that nothing known is attached to this destination port. However, the source port is linked to an ASP module for Apache web servers. One of the possible causes of this traffic could either be a scan of outside networks in an attempt to map the networks or it could be a very busy web server that is pushing out some kind of ASP page to many other sites. We feel that the first is the case because of the consistent destination port.

3.4.2 – 172.31.100.208 - Moses.cs.university.edu

This host was responsible for the second most traffic on the network. It shows up in both the alert database and the scans database as one of the top three talkers in both lists. From a scans database standpoint all of the traffic was on the 5th of August and was all in the form of a Syn Scan to various outside hosts using port 80 as a destination. From the alerts database it was noted that the host was generating alerts for Nimda using both the cmd and root exploits as well as an IIS Unicode exploit. As stated above this host should be immediately removed from the network and examined for evidence of the Nimda virus having compromised the system.

3.4.3 – 172.31.84.234 - Engr-84-234.pooled.university.edu

This host was responsible for just under 1 million events between the alerts database and the scans database. It was the #2 host for events on both databases. It produced almost 500,000 alerts for IDS552, which is an ISAPI overflow alert. It also produced almost 500,000 events as SYN scans. All of these events took place on August 4th. It is suspected that the host at 172.31.84.234 is infected by a virus that is in the process of scanning outside networks looking for other hosts that might be vulnerable to its attack.

3.4.4 – 172.31.70.207 - ecs020pc09.university.edu

This host was responsible for almost 150,000 UDP scans in the scans database. There was only one reference in the alerts database to this host and it's a reference to a destination port of 55850 that occurred on August 1st at 6:03 in the morning. A quick check of the online ports database reveals nothing for this port. In looking at the times that these events occurred there are definite gaps which makes it look like whatever is causing this is probably only active when the computer is on or being used. This could mean that the University has somebody who is actively scanning networks in their midst.

3.4.5 – 172.31.82.2 - oit-82-02.pooled.university.edu

This host was responsible for a UDP scan of the outside network between August 1st and August 3rd. It was also noted that this host was the subject of an SMB name wildcard scan during these times. There are no correlations between the hosts that the UDP scan was directed at and the hosts that scanned this machine with the SMB Name Wildcard scan.

3.4.6 – 172.31.165.24 - slate.university.edu

This host was responsible for UDP scanning the outside network using a port number from an old protocol (WinMX). The port number used for the UDP scanning was port 6257. This host was also responsible for some SYN scan activity using various destination and source ports. It was involved in these scans from August 1st at around 5:25 am till August 2nd at 11:45 am. In looking at the alert database it was also noted that this host generated 76 events on the use of a destination port number of 65535. The alert calls out a possible "Red Worm" traffic. This host should be checked for the presence of a Trojan program.

3.4.7 – 172.31.83.150 - aciv-83-150.pooled.university.edu

This host was responsible for UDP scanning the outside network using the same port number as the last host we described (WinMX port #6257). It was also involved in "Red Worm" traffic but on the receiving side of the traffic not on the generating side of the traffic. This host also received some SMB Name Wildcard traffic directed at it from

outside the network. It is reasonable to guess that this host has somehow been compromised the same way that slate.university.edu has been. It should be immediately checked for the presence of a virus or Trojan program.

3.4.8 - 3.0.0.99 – No Translation

This host was responsible for almost all of the UDP Source and Destination outside network alerts that were generated between August 1st and August 5th. All of the alerts from this host were destined to 10.0.0.1, which is an address that should not be going across the Internet. Therefore, it is believed that this host is either misconfigured or in some way the address on this system is not updating via DHCP or any other boot protocol the University might be using. It will be very hard to find this system but it should be harmless since the packets are not going to get very far since any response to the address 3.0.0.99 should be routed back to GE and not to the host in the University.

3.4.9 – 172.31.137.7 – No Translation

This host is a popular host for attacks. It received SMB Name Wildcard attacks; IIS Unicode attacks and ISAPI overflow attacks. It was also responsible for generating almost 50,000 UDP scans to the outside network. These UDP scans took two forms. First, quite a few of the scans originated from port 88, which is used by Kerberos on 172.31.137.7. The other scans seem to have been destined for port 53, which is the port, used by domain name servers. All of the Kerberos traffic is destined for hosts in the 204.183.84.0 network. A quick check of the Arin database shows the following company owns this address:

```
OrgName:    Ashby & Geddes
OrgID:      ASHBYG

NetRange:   204.183.84.0 - 204.183.84.255
CIDR:       204.183.84.0/24
NetName:    ASGE001-204-183-84
NetHandle:  NET-204-183-84-0-1
Parent:     NET-204-183-80-0-1
NetType:    Reassigned
Comment:
RegDate:    1998-09-30
Updated:    1998-09-30

TechHandle: AG89-ARIN
TechName:   Geddes, Ashby
TechPhone:  +1-302-654-1888
TechEmail:  dns@dca.net

# ARIN Whois database, last updated 2002-10-28 19:05
# Enter ? for additional hints on searching ARIN's Whois database.
```

It is possible that the University has a contract with this organization. This should be looked into to rule out the possibility that an attack is occurring. As far as the port 53 traffic the system at this address should be looked at for any kind of scanning activity as this traffic is very random for destination addresses but uses the same source port for a lot

of its packets. Since it uses the same port a lot for its source port it is possible that this is some malicious user on the University System.

3.4.10 – 172.31.70.133 - kenny.ucs.university.edu

This host saw one SMB Name Wildcard attack on August 2nd at 16:43 and then on August 3rd started to generate UDP scans on a periodic basis. It is possible that this host was infected with something and found to be vulnerable to the SMB Name Wildcard attack. This host should be looked at for a possible Trojan or root kit. The UDP scans that were generated look like the packets are contrived due to them all being in the low 7000 range. The host also seems to be spending a good portion of the time attacking the network at 216.254.108.0. This network belongs to RIO Motor Sports. Here is their Arin database entry:

```
CustName:   RIO MOTOR SPORTS, INC
Address:    25 Broadway New York NY 10004
Country:    US
RegDate:    2001-11-09
Updated:    2001-11-09

NetRange:   216.254.108.16 - 216.254.108.31
CIDR:       216.254.108.16/28
NetName:    SPEK-272665-0
NetHandle:  NET-216-254-108-16-1
Parent:     NET-216-254-0-0-1
NetType:    Reassigned
Comment:
RegDate:    2001-11-09
Updated:    2001-11-09

# ARIN Whois database, last updated 2002-10-29 19:05
# Enter ? for additional hints on searching ARIN's Whois database.
```

3.5 Selected External Sites of Interest

EOIs	IP Address	FQDN
51,359	3.0.0.99	No Translation
39,484	216.241.219.28	No Translation
32,117	63.250.213.12	Dal-qcwm213012.broadcast.com
8,375	194.98.189.139	No Translation
6,899	80.137.90.34	p50895A22.dip.t-dialin.net

3.5.1 - 3.0.0.99

This host was chosen because it's the most talkative of the external sites. It does not have a reverse lookup entry in GE's DNS server. This host was responsible for a good portion of the alert for UDP traffic where the Source and Destination are outside the network. It is interesting that this traffic is being picked up by the University scanner as since most of it goes to a host at 10.0.0.1 it should not be making it across the Internet at all. This could

be caused by a host on the local network that has somehow gotten the 3.0.0.99 address. Possibly from a General Electric Laptop that has been plugged in and did not receive a DHCP update for some reason.

```
OrgName:    General Electric Company
OrgID:      GENERA-9

NetRange:   3.0.0.0 - 3.255.255.255
CIDR:       3.0.0.0/8
NetName:    GE-INTERNET
NetHandle:  NET-3-0-0-0-1
Parent:
NetType:    Direct Assignment
NameServer: ns.ge.com
NameServer: ns1.ge.com
NameServer: ns2.ge.com
Comment:
RegDate:    1988-02-23
Updated:    2002-09-26

TechHandle: GET2-ORG-ARIN
TechName:   General Electric Company
TechPhone:  +1-518-612-6672
TechEmail:  genictech@ge.com
```

3.5.2 – 216.241.219.28

This host was chosen because it seems to coincide with a specific attack against it from a few hosts within the University. Each time this host appears in the destination address field of an alert it is being attacked from inside the University by using a CGI null byte vulnerability attempt. There were 4 specific times that this address was the object of attack each time a different internal IP address was responsible. This we believe coincides with an individual using different resources to perform a scan of this external host. The address is owned by The Cobalt Group, Inc which is provider of e-solutions to the automotive industry.

```
OrgName:    The Cobalt Group, Inc
OrgID:      THECOB

NetRange:   216.241.208.0 - 216.241.223.255
CIDR:       216.241.208.0/20
NetName:    COBALT-NET2
NetHandle:  NET-216-241-208-0-1
Parent:     NET-216-0-0-0-0
NetType:    Direct Assignment
Comment:
RegDate:    1999-11-16
Updated:    1999-11-16

TechHandle: MF401-ARIN
TechName:   Fitzgerald, Michael
TechPhone:  +1-800-909-8244
TechEmail:  mikef@cobaltgroup.com

# ARIN Whois database, last updated 2002-10-28 19:05
# Enter ? for additional hints on searching ARIN's Whois database.
```

3.5.3 - 63.250.213.12

This host was chosen because it is the second most talkative outside host. It like the one above (3.0.0.99) is also generating UDP SRC and DST outside network alerts. The interesting thing about this address is that it talks directly to a host on the 233 network which is reserved for multicast. This is probably a host on the University network that is trying to view a video across the Internet. It is a good guess that it doesn't work because it would be directing any return traffic back to the Yahoo Broadcast site.

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

NetRange:   63.250.192.0 - 63.250.223.255
CIDR:       63.250.192.0/19
NetName:    NETBLK2-YAHOOBS
NetHandle:  NET-63-250-192-0-1
Parent:     NET-63-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.YAHOO.COM
NameServer: NS2.YAHOO.COM
NameServer: NS3.YAHOO.COM
NameServer: NS4.YAHOO.COM
NameServer: NS5.YAHOO.COM
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    1999-11-24
Updated:    2002-03-27

TechHandle: NA258-ARIN
TechName:   Netblock Admin, Netblock
TechPhone:  +1-408-349-7183
TechEmail:  netblockadmin@yahoo-inc.com
```

3.5.4 - 194.98.189.139

This host was selected for its interesting traffic that it generated. It was involved in a sweep of the Universities network. The host on August 3rd did a Sync scan of the University network looking for Sunrpc ports that were open. From the information that was provided in the files it cannot be determined if the scan was successful and if any hosts were compromised. It might be a good idea to block all external access to RPC ports unless they are specifically needed for some application that the University is engaged in.

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:      194.98.189.128 - 194.98.189.143
netname:      INGENCYS-NET1
descr:        INGENCYS
country:      FR
admin-c:      DR5-RIPE
tech-c:       JB371-RIPE
status:       ASSIGNED PA
remarks:      abuse@fr.uu.net
mnt-by:       IWAY-NOC
changed:      frederic.martzel@mciworldcom.fr 20010924
```

```
source:       RIPE
route:        194.98.0.0/16
descr:        UUNET-BLOCK1
descr:        UUNET France Block 1
origin:       AS702
remarks:      ************************************
remarks:      For all spamming or hacking problems
remarks:      please send your requests directly to
remarks:      abuse@fr.uu.net
remarks:      ************************************
notify:       net-adm@mciworldcom.fr
mnt-by:       IWAY-NOC
changed:      net-adm@iway.fr 19981109
changed:      frederic.martzel@mciworldcom.fr 20011114
source:       RIPE
role:         technical contact
address:      UUNET FRANCE
address:      215, Avenue Georges Clemenceau
address:      F-92024 NANTERRE Cedex
phone:        +33 1 56 38 22 00
fax-no:       +33 1 56 38 22 01
e-mail:       net-adm@mciworldcom.fr
admin-c:      VP1616-RIPE
admin-c:      FM7174-RIPE
admin-c:      AW7486-RIPE
tech-c:       ZM321-RIPE
tech-c:       AH6610-RIPE
tech-c:       TC334-RIPE
nic-hdl:      JB371-RIPE
remarks:      ------------------------------------
remarks:      For all spamming or hacking problems
remarks:      please send your requests directly to
remarks:      abuse@fr.uu.net
remarks:      ------------------------------------
mnt-by:       IWAY-NOC
changed:      frederic.martzel@mciworldcom.fr 20010828
source:       RIPE
person:       Monsieur De Royer
address:      INGENCYS
address:      4, Rue de la Madeleine
address:      45140 ST JEAN DE LA RUELLE, France
phone:        +33 2 37 25 12 00
fax-no:       +33 2 37 25 12 00
nic-hdl:      DR5-RIPE
mnt-by:       IWAY-NOC
changed:      frederic.martzel@mciworldcom.fr 20010924
source:       RIPE
```

3.5.5 - 80.137.90.34

This host was selected because it was involved in a scan of the University network looking for machines that are vulnerable to an IIS Unicode Attack. All of the events of interest for this host were produced on August 5th between 10:01 am and 11:15 am. As can be seen from the RIPE whois output for this address the owner of the IP address is an ISP in Germany called Deutsche Telekom. They should be contacted at their abuse address and told to investigate this scan. Another interesting point about this scan is that it looks like they already have a decent map of the University network because the attack was not a completely random search for vulnerable machines it was directed at specific machines.

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
```

```
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
inetnum:        80.128.0.0 - 80.146.159.255
netname:        DTAG-DIAL16
descr:          Deutsche Telekom AG
country:        DE
admin-c:        DTIP-RIPE
tech-c:         ST5359-RIPE
status:         ASSIGNED PA
remarks:        ****************************************************************
remarks:        * ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK ATTACKS,      *
remarks:        * ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC.        *
remarks:        ****************************************************************
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20020108
source:         RIPE
route:          80.128.0.0/11
descr:          Deutsche Telekom AG, Internet service provider
origin:         AS3320
mnt-by:         DTAG-RR
changed:        bp@nic.dtag.de 20010807
source:         RIPE
person:         DTAG Global IP-Adressing
address:        Deutsche Telekom AG
address:        Bayreuther Strasse 1
address:        D-90409 Nuernberg
address:        Germany
phone:          +49 911 68909856
e-mail:         ripe.dtip@telekom.de
nic-hdl:        DTIP-RIPE
mnt-by:         DTAG-NIC
changed:        ripe.dtip@telekom.de 20020717
source:         RIPE
person:         Security Team
address:        Deutsche Telekom AG
address:        Technikniederlassung Schwaebisch Hall
address:        D-89070 Ulm
address:        Germany
phone:          +49 731 100 84055
fax-no:         +49 731 100 84150
e-mail:         abuse@t-ipnet.de
nic-hdl:        ST5359-RIPE
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20010321
source:         RIPE
```


3.6 Link Graph of Attack using CGI Null Bytes

The link graph shown in Figure 3.6.2 shows an interesting pattern of attack that hosts on the inside of the University performed against 3 outside hosts. The hosts are from AOL, The Cobalt Group and Ifilm Inc. All of these internal hosts used the CGI Null Byte vulnerability to attempt to gain access to the 3 outside sites. The table shown in Figure #3.6.1 shows the list of internal hosts that were involved in the scan of the outside hosts. It is noted that a good portion of these hosts seem to be pooled hosts which it is assumed means that they are temporary addresses and so it may be difficult to determine the exact

system that was using that host at the time in question.

IP Address	FQDN
172.31.109.83	ecs314pc-08.engr.university.edu
172.31.182.91	rac120pc-01.university.edu
172.31.88.155	lib-88-155.pooled.university.edu
172.31.87.52	chem-87-52.pooled.university.edu
172.31.81.37	erk-81-37.pooled.university.edu
172.31.178.219	ad412pc-02.university.edu
172.31.70.48	ecs020pc-carole.ucs.university.edu
172.31.84.189	engr-84-189.pooled.university.edu
172.31.85.78	acserv107gpc-01.university.edu
172.31.87.103	chem-87-103.pooled.university.edu

Figure #3.6.1

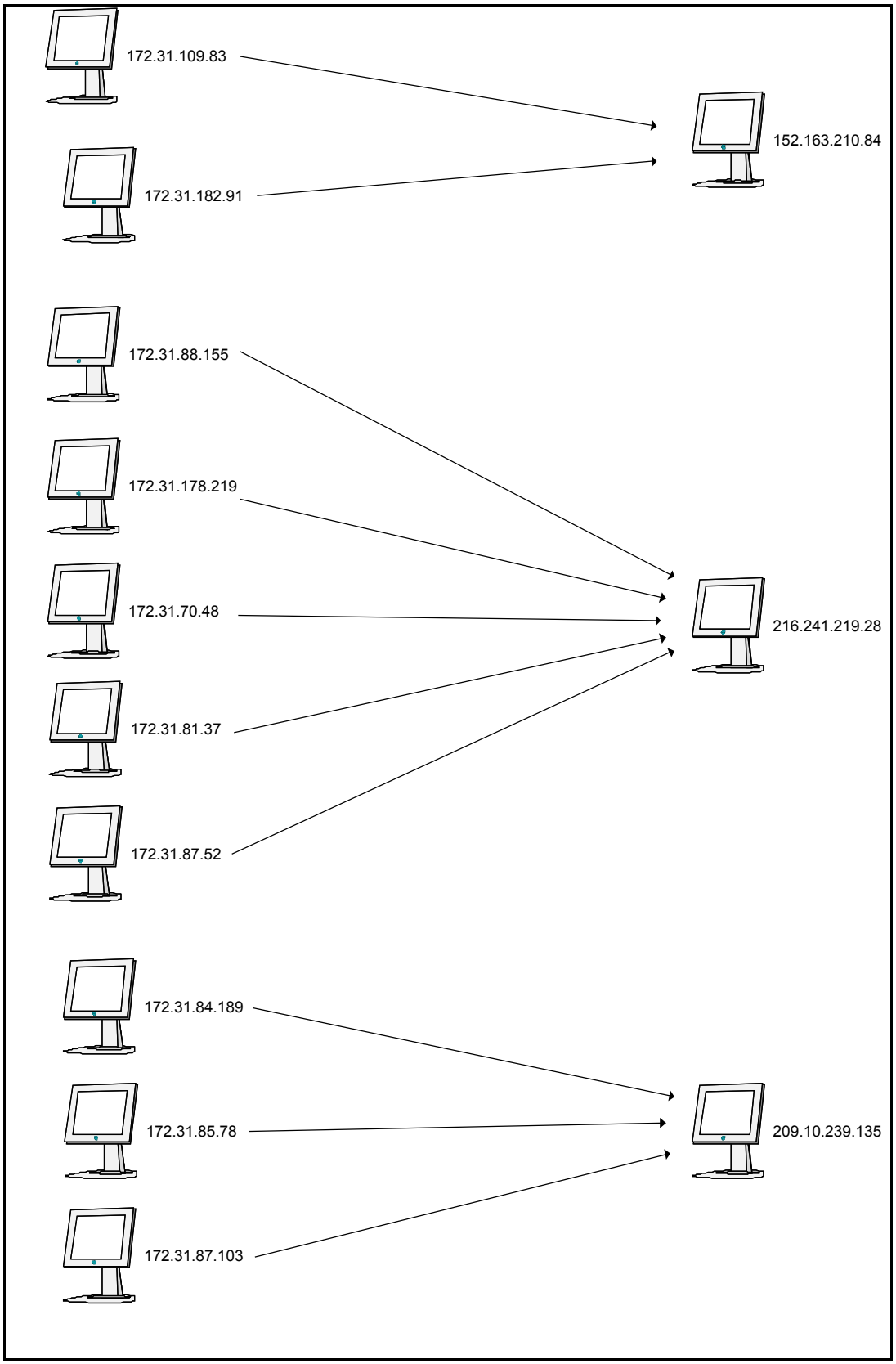

Figure #3.6.2

For reference the outside hosts have been checked using the Arin database and their

individual records are shown in the next few paragraphs.

The following is the Arin database entry for 152.163.210.84:

```
OrgName:    America Online
OrgID:      AOL

NetRange:   152.163.0.0 - 152.163.255.255
CIDR:       152.163.0.0/16
NetName:    AOL-BNET
NetHandle:  NET-152-163-0-0-1
Parent:     NET-152-0-0-0-0
NetType:    Direct Assignment
NameServer: DNS-01.NS.AOL.COM
NameServer: DNS-02.NS.AOL.COM
Comment:
RegDate:    1992-04-01
Updated:    1999-12-02

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

The following is the Arin database entry for 216.241.219.28:

```
OrgName:    The Cobalt Group, Inc
OrgID:      THECOB

NetRange:   216.241.208.0 - 216.241.223.255
CIDR:       216.241.208.0/20
NetName:    COBALT-NET2
NetHandle:  NET-216-241-208-0-1
Parent:     NET-216-0-0-0-0
NetType:    Direct Assignment
Comment:
RegDate:    1999-11-16
Updated:    1999-11-16

TechHandle: MF401-ARIN
TechName:   Fitzgerald, Michael
TechPhone:  +1-800-909-8244
TechEmail:  mikef@cobaltgroup.com
```

The following is the Arin database entry for 209.10.239.135:

```
CustName:   IFilm Corp
Address:    1024 North Orange Drive HOLLYWOOD CA 90038
Country:    US
RegDate:    2002-10-01
Updated:    2002-10-01

NetRange:   209.10.239.128 - 209.10.239.191
CIDR:       209.10.239.128/26
NetName:    IP007442-209-10-239
NetHandle:  NET-209-10-239-128-1
Parent:     NET-209-10-0-0-1
NetType:    Reassigned
Comment:
RegDate:    2002-10-01
Updated:    2002-10-01
```

3.7 Correlations

There are many correlations for the analysis put forth in my analysis. The following are just some of the articles, mail messages and white papers that support the various conclusions in the above anaylsis:

Spp_http_decode: IIS Unicode attack detected:

http://www.geocrawler.com/archives/3/4890/2001/8/0/6521002/

IDS552/web-iss_IIS ISAPI Overflow ida INTERNAL nosize:

http://listserv.unipr.it/pipermail/staffnet/2001/000191.html

Spp_http_decode: CGI Null Byte attack detected:

http://archives.neohapsis.com/archives/snort/2000-11/0244.html

SMB Name Wildcard:

http://archives.neohapsis.com/archives/snort/2000-01/0222.html

TFTP – External UDP connection to internal tftp server:

http://www.cert.org/body/advisories/CA200126_FA200126.html

Watchlist 000220 IL-ISDNNET-990517:

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

3.8 Defensive Recommendations

In general the Universities network is in pretty good shape. They have obviously dedicated some time into developing an effective IDS solution. They have developed some local specific rules to assist in the finding of attacks. There still seems to be some issue with CodeRed and Nimda viruses within the campus however and this needs to be looked into as soon as possible.

A recommendation that should be considered is to develop a comprehensive project to patch and update systems within the University to protect against the more common viruses like Nimda and CodeRed. It might be possible to develop and enterprise wide solution to this problem by working with the Universities virus protection vendor.

Another recommendation is for the University to develop a comprehensive security awareness program to educate students, faculty and administration to the various ways that computer systems can be compromised. This awareness program should include easy methods for people to patch and keep their systems up to date. Possibly this could be tied in with a discount for a anti-virus system or possibly even a personal firewall system. This would greatly help in protecting the University from being compromised by the multitude of types of systems that are in use at the University.

3.9 Analysis Process

These files were downloaded to both a Windows 2000 system and a Linux system with the intent that the files would be processed and analyzed in different ways that the two systems are better at. These files were first analyzed using some customized scripts that were obtained from Appendix A of the practical submitted by Tod Beardsley which is available at the following link:

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

The scripts allowed for first and foremost a conversion of the files into comma separated value based format, which allowed the information to be processed into Microsoft Access for further analysis. The second script summarized the information by counting up the number of incidents based on time, alert and source address.

The scripts gave us a general list of the most active alerts and scans so that the Access database that had been generated from the comma separated value files could be used to do analysis.

A query was developed that would sort the database by alert or scan and then by source address. Selecting a specific alert and applying a standard filter to the data then further reduced this information. This was then used to analyze both the source and destination information to develop an understanding of the alert.

Linux was used to investigate the Out of Specification data using the Grep and Wc commands in commands of the following form:

```
Grep "MY\.NET\.6\.7" oos.txt | wc –l
```

The out of specification files were also viewed using the VI editor to develop a full understanding of the information.

3.10 References

"Arin Network Whois Database" URL: http://ws.arin.net/cgi-bin/whois.pl (October 9, 2002)

"Ripe Whois Database" URL: http://www.ripe.net (November 2, 2002)

Puppy, Rainforest. "A look at Whisker's anti IDS tactics" URL: http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html (November 2, 2002)

"Netbios Name Query" URL: http://www.whitehats.com/info/ids177 (October 10, 2002)

"Ports Database" URL: http://www.portsdb.org (October 15, 2002)

Alexander, Bryce. "Port 137 Scan" Sans Intrusion Detection FAQ URL: http://www.sans.org/newlook/resources/IDFAQ/port_137.htm (October 1, 2002)

Singer, Abe. "Network.vbs Analysis" San Diego Supercomputer Center URL: http://security.sdsc.edu/publications/network.vbs.shtml (October 1, 2002)

"Nimda Worm Info" Farm 9 URL: http://farm9.com/content/0918worm (November 2, 2002)

Beardsley, Tod. "Intrusion Detection and Analysis: Theory, Techniques and Tools" URL: http://www.giac.org/practical/Tod_Beardsley_GCIA.doc (October 2, 2002)

Holstein, Michael. "SANS GCIA Practical Assignment" URL: http://www.giac.org/practical/Michael_Holstein_GCIA.doc (October 5, 2002)

Drew, Steven. "Intrusion Detection in Depth GCIA Practical Assignment version 3.1" URL: http://www.giac.org/practical/Steven_Drew_GCIA.doc (October 6, 2002)