Global Information Assurance Certification Paper

GIAC GCIA Practical Assignment

Version 4.1

University Security Assessment

Rudy Ristich

March 5th, 2005

© SANS Institute 2000 - 2005 Author retains full rights.

Table of Contents

Executive Summary

This report contains the analysis of network traffic for University XYZ from March 25th through March 27th 2004. Log files were obtained from the University in order to identify threats to the security and performance of the network. Included in the set of logs to be analyzed were signature alert logs, port scanning logs, and out of specification (OOS) logs. Signature alert logs identify traffic that has patterns similar to behavior that may be malicious or detrimental to the performance of the network. Port scanning logs record probes both to and from the local network and may identify information gathering attempts or host misconfigurations as well as assisting in profiling the University Network. OOS logs provide information on network traffic that may not match the specified networking protocols which can sometimes be an indication of malicious activity or a broken application.

The results of this analysis reveal that there are several existing and potential threats to the network and at least 67 compromised hosts on the network. An FTP server on the network has been verified to be dangerously unprotected. This machine is running an outdated version of server software that allows it to be easily crashed by a remote user. Vulnerabilities also exist in this software that would allow a clever hacker to gain full access to the system via a buffer overflow attack. Among the most prevalent threats are worms. Worms are malicious programs that replicate autonomously and often with out human intervention. The NIMDA worm has the ability to destroy or corrupt data on the infected machines as well as use the infected host as a tool for sending out SPAM and scanning for new hosts to spread to. These activities are known to adversely affect network performance. A variant of the Agobot worm is also present on the network. Agobot gives an attacker complete control of the infected machine through Internet Relay Chat, a popular service on the internet. A number of machines on the network are hooked into one of these control channels and is carrying out malicious scanning of remote hosts. A detailed technical report of each of these threats follows as well as an explanation of the analysis so that the university network staff may conduct their own analysis in the future. The technical content of the report assumes a working knowledge of the TCP/IP protocol as well as knowledge of fundamental networking concepts and applications.

I am making several security based recommendations. An upgrade of the intrusion detection devices is highly recommended. The signature set is outdated; if these are updated more often, more accurate information can be gathered for analysis. Logging more information in a more flexible format will help analysts assess threats to the network more quickly. If a firewall exists implementing a strict default deny policy will help reduce the spread of infectious mal ware and ward of any unauthorized activities. It is also very important that an internal policy be created regarding patching of systems against known vulnerabilities. Some of the worms on the system are over two

years old and have several known fixes available that are simple to install and some service network machines such as the FTP server mentioned earlier are at a great risk that can be eliminated with a simple upgrade. Remote users dialing in also pose a threat to the integrity of the network. Users with personal computers are often times behind on software updates and patches that would deter infections. When users with infected machines log on to the network they introduce the risk of new threats spreading across the network. It is advised that the network staff implement a solution that would prevent the spread of an infection from the dial up subnet to the remainder or the university network. If these recommendations are considered and implemented the university network will be more secure and will likely prove to be a greater resource to the institution.

Part II: Detailed Analysis

1. Files Analyzed:

This data for this analysis originates from a University that we will refer to as UXYZ. The temporal span of the data in question is March 25th 2004 to March 27th 2004. This analysis is the result of the correlation of three types of log files as well as research from trusted internet sources and the SANS GCIA practical repository.

Scan logs: The first type of files that were analyzed are scan log files. These logs report all scanning or "scan-like" activity from port scan alerts generated by the snort rule set. These represent possible information gathering attempts against the victim net work and are useful for correlation to IDS alerts. The list of specific scan files is as follows: scans.040325, scans.040326, and scans.040327.

Alert logs: These logs contain the alerts generated by the snort rule set. These are indications of traffic patterns that warrant a record. Scans are also logged in the alert file with supplemental information held in the scan log files described above. Each alert represents a possible security breach on the system and is worth careful investigation. The list of specific alert files is as follows: alert.040325, alert.040326, and alert.040327.

Out of Specification (OOS) logs: These logs contain packets interpreted by the snort engine as out of RFC specification. This can be an indication of malicious traffic or anomalous isolated behavior due to packet corruption or software not following protocol standards. The list of specific OOS files is as follows[1]: oos_report_040321, oos_report_040322, and oos_report_040323.

These files are all freely available through the SANS Internet Storm Center at: http://ics.sans.org/logs/

2. Relational Analysis:

Meaningful analysis of network traffic requires a working knowledge of the network architecture and the relationship between each node of the network; relationships that are both expected and anomalous. Forming a visual representation of a network in question is often an invaluable skill when performing intrusion analysis. The pairing of an intricate knowledge of standard network protocols with a keen insight of packet traces can help us form fairly accurate graphs of networks and suspect activity.

After inspecting alert and scan logs we can make an educated guess as to the

[1] It is important to note that although the files suggest a date of March 21st through the 23rd upon further inspection these files do in fact have time stamps that are corollary to the rest of the data set.

topology of the network. An extremely large percentage of traffic responding to DNS (port 53) traffic comes from MY.NET.1.3 and MY.NET.1.4. It is safe to assume that these are the primary DNS servers for the University's network and we can verify this by running a dig on the IP addresses and see that they do indeed resolve to NS records. The large majority of traffic toward common server ports including FTP, news, active directory, and web services seem to be from the entire MY.NET.24.0/24 network. We will label this as the main server network, there is not sufficient evidence to be certain that this is a DMZ but it is very likely. The only service points of interest that do not lie on the .24 subnet are the mail servers and relays, including IMAP, POP3, and exchange. The bulk of SMTP traffic originated from 25.67 through 25.71 with a fairly uniform load distribution. Correlating this information with results from dig confirms that these address map to MX records and their hostnames mx1in, mx1out, etc imply they are, in fact, mail relays. These services lie predominately across the MY.NET.25.0/24 network so we can safely assume this is a mail cluster. There seems to be plenty of traffic indicating that this particular network is very robust, encompassing most of the MY.NET.0.0/16 address space of the University. Included among names that resolve are department networks, dialup banks, land library networks; just to name a few. Figure 2.1 below is a graphical representation of the results discussed above.

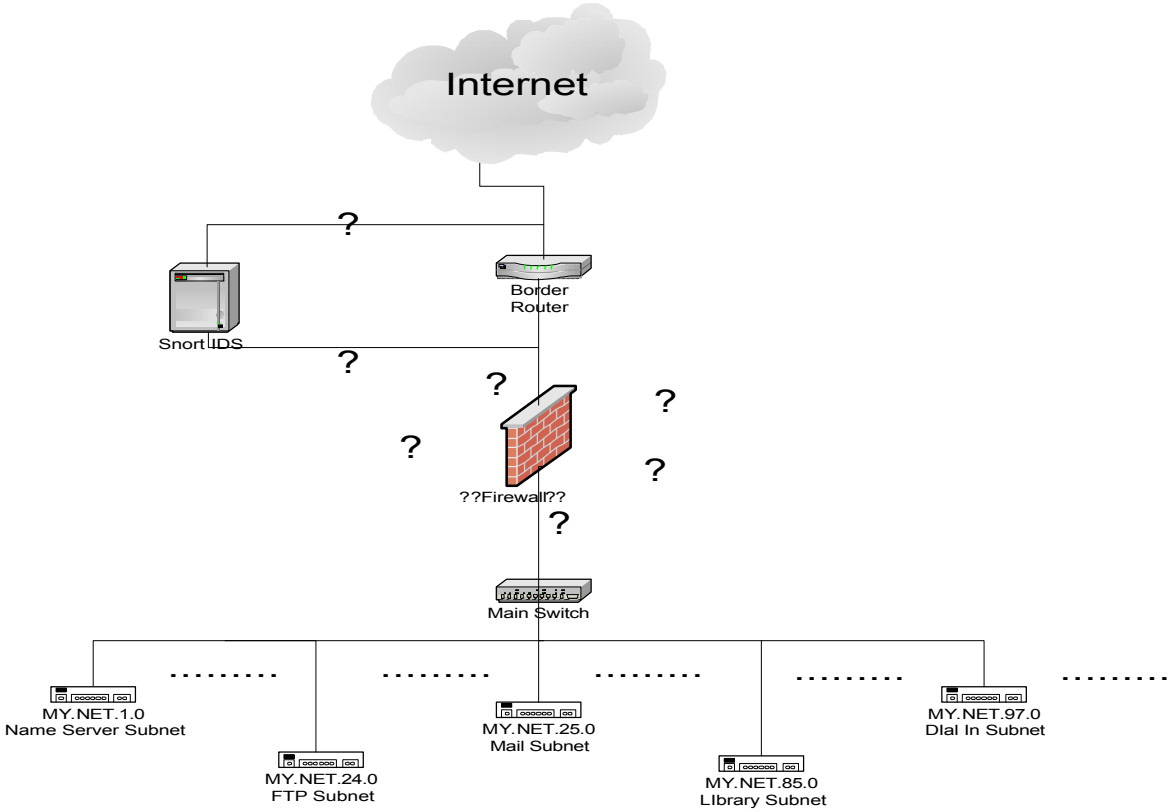

Figure 2.1 Suggested Network Topology. Question marks denote assumptions that cannot be confirmed through the data.

A link graph is included with each network detect in this report to help give a visual aid to the traffic being analyzed. It is important to note that even though

printed sources from the internet state there is a firewall on the actual University network, there is no information in the data analyzed that can prove the existence of a firewall. Therefore in the diagram some assumptions are made, denoted by question marks. More information on the methods and tools used to arrive at the analysis of the network topology is contained in Part III of this report.

3. Network Detects:

List of detects: Table 3.1 shows a list of network detect summaries from the alert logs from March 25th through March 27th 2004. There were 47 different signatures triggered for a total of 76875 alerts over 72 hours.

Triggers	Summary
22875	MY.NET.30.3 activity
15949	MY.NET.30.4 activity
13788	connect to 515 from outside
11387	High port 65535 tcp - possible Red Worm - traffic
4462	EXPLOIT x86 NOOP
3662	SMB Name Wildcard
1364	Null scan!
510	NMAP TCP ping!
347	FTP DoS ftpd globbing
286	Possible trojan server activity
285	[UXYZ NIDS IRC Alert] IRC user /kill detected, possible trojan.
268	[UXYZ NIDS] External MiMail alert
194	EXPLOIT x86 NOPS
173	[UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC
173	TFTP - External TCP connection to internal tftp server
146	Incomplete Packet Fragments Discarded
143	SUNRPC highport access!
95	Tiny Fragments - Possible Hostile Activity
89	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
88	FTP passwd attempt
73	SMB C access
72	ICMP SRC and DST outside network
74	High port 65535 udp - possible Red Worm - traffic
57	TCP SRC and DST outside network
55	NIMDA - Attempt to execute cmd from campus host
37	RFB - Possible WinVNC - 010708-1
34	TCP SMTP Source Port traffic
32	Attempted Sun RPC high port access
21	EXPLOIT x86 setuid 0
19	IRC evil - running XDCC
19	EXPLOIT x86 setgid 0
18	SYN-FIN scan!
18	DDOS shaft client to handler
15	[UXYZ NIDS IRC Alert] Possible drone command detected.
7	DDOS mstream client to handler
6	TFTP - Internal UDP connection to external tftp server
6	Probable NMAP fingerprint attempt
6	External RPC call
5	External FTP to HelpDesk MY.NET.53.29
3	SITE EXEC - Possible wu-ftpd exploit - GIAC000623
3	External FTP to HelpDesk MY.NET.70.50
3	EXPLOIT NTPDX buffer overflow
2	Traffic from port 53 to port 123
2	NIMDA - Attempt to execute root from campus host
2	EXPLOIT x86 stealth noop """"
1	External FTP to HelpDesk MY.NET.70.49
1	DOS Real Server template.html

Detects:

1. FTP DoS ftpd globbing

```
03/25-16:36:37.925561  [**] FTP DoS ftpd globbing [**] 24.126.121.235:62115 ->
MY.NET.153.81:21
03/25-16:25:15.404003  [**] FTP DoS ftpd globbing [**] 24.126.121.235:62115 ->
MY.NET.153.81:21
03/25-16:25:18.372807  [**] FTP DoS ftpd globbing [**] 24.108.229.185:33365 ->
MY.NET.153.81:21
03/25-16:47:26.575815  [**] FTP DoS ftpd globbing [**] 24.126.121.235:62115 ->
MY.NET.153.81:21
03/25-16:25:22.091119  [**] FTP DoS ftpd globbing [**] 24.126.121.235:62115 ->
MY.NET.153.81:21
03/25-16:47:30.626910  [**] FTP DoS ftpd globbing [**] 217.155.204.164:63491 ->
MY.NET.153.81:21
03/25-16:36:48.265511  [**] FTP DoS ftpd globbing [**] 24.108.229.185:33365 ->
MY.NET.153.81:21
03/25-16:36:51.546463  [**] FTP DoS ftpd globbing [**] 24.108.229.185:33365 ->
MY.NET.153.81:21

03/26-11:37:04.704310  [**] FTP DoS ftpd globbing [**] 4.43.36.23:4415 -> MY.NET.24.27:21
03/26-11:40:29.981262  [**] FTP DoS ftpd globbing [**] 168.215.141.208:4690 -> MY.NET.24.27:21
03/26-19:35:54.964765  [**] FTP DoS ftpd globbing [**] 24.106.112.249:3237 -> MY.NET.24.27:21
03/27-03:28:23.370648  [**] FTP DoS ftpd globbing [**] 208.251.137.90:1810 -> MY.NET.24.27:21
03/27-03:33:03.678960  [**] FTP DoS ftpd globbing [**] 208.251.137.90:1810 -> MY.NET.24.27:21
```

Description: FTP "globbing" is the process whereby wildcards are inserted into filenames in order to match more than one file.[2] In certain FTP daemons globbing by use of large strings such as:

/..//../*/../*/../*/../*/../*/../*/../*/../*/../*/../*

can cause a buffer overflow on a remote system. These overflows can result in command execution on the remote host running the FTP daemon, or in the case we will discuss here a Denial of Service where by the daemon crashes due to the string overflow.

In this detect the alert triggered 347 times over a three day period and involved 15 distinct external hosts and two internal hosts, presumably running an ftp server. Table 3.2 summarizes the information gathered about the internal and external hosts.

Table 3.2 "FTP DoS ftpd globbing" External and Intenal Talkers

Alerts	External Address	FQDN
114	24.108.229.185	S010600095b250a93.gv.shawcable.net
93	24.126.121.235	c-24-126-121-235.we.client2.attbi.com
84	217.155.204.164	dsl-217-155-204-164.zen.co.uk
13	80.126.240.251	ytsberg.xs4all.nl
10	217.13.22.40	217-13-22-40.dd.nextgentel.com
9	208.251.137.90	Does not resolve
7	168.215.141.204	168-215-141-204.gen.twtelecom.net
7	140.239.150.248	ip248.netriplex.com
4	80.167.220.190	x1-6-00-0e-5c-ed-13-6d.k283.webspeed.dk
2	204.210.29.72	dt042n48.san.rr.com
1	81.215.157.14	dsl81-215-40334.adsl.ttnet.net.tr
1	4.43.36.23	Does not resolve
1	24.147.170.175	h000a95698250.ne.client2.attbi.com
1	24.106.112.249	rrcs-24-106-112-249.central.biz.rr.com
1	168.215.141.208	168-215-141-208.gen.twtelecom.net

[2] http://www.iss.net/security_center/advice/Intrusions/2001350/default.htm

Alerts	Internal Address	FQDN
321	MY.NET.153.81	refweb17.libpub.uxyz.edu.
26	MY.NET.24.27	ragnarok.uxyz.edu

Research on Bugtraq reveals that a heap corruption causes a segmentation fault in wu-ftpd versions 2.6.1, 2.6.0, and 2.5.0 when using certain globbing characters.[3] Basically, anyone who can obtain access to a machine running these outdated versions of wu-ftpd has the ability to successfully carry out a denial of service attack on the offending system. This includes the use of an anonymous login if enabled. An illustration of such an attack is included in Figure 3.1 below:

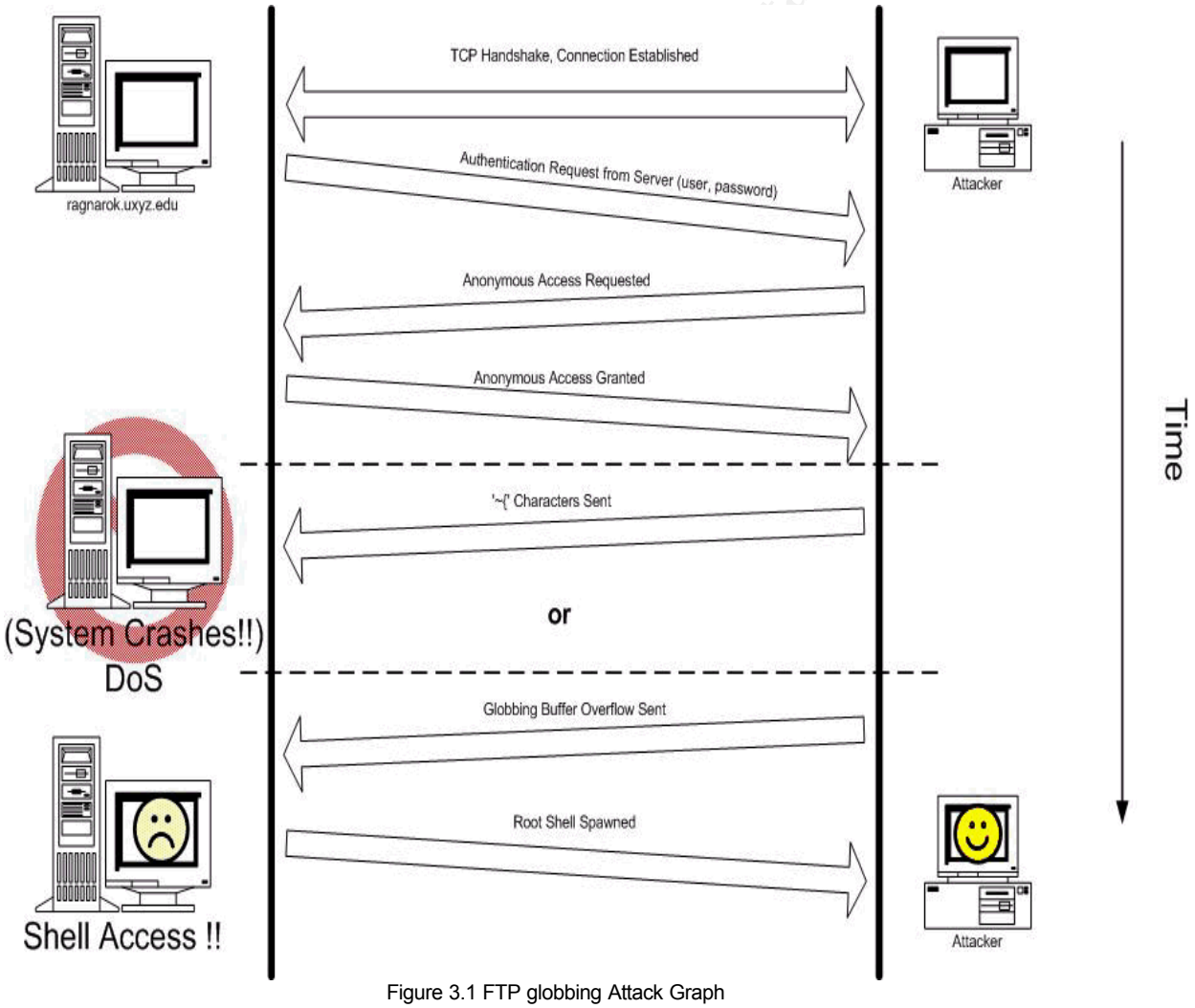

Figure 3.1 FTP globbing Attack Graph

Reasoning for selection:

[3] http://www.securityfocus.com/bid/3581/exploit/

This attack was selected because wu-ftd is of the most popular and easiest to configure ftp daemons available. Alerts on this signature do not necessarily imply that a host will be compromised because an intrusion detection system does not have the ability to detect whether or not a host is running a vulnerable piece of software, it will simply alert on a pattern in TCP traffic. It is however possible to do some investigation after an alert comes in to determine the probability that an attempt to compromise was carried out.

Attempting to connect to our top talking host in this alert (MY.NET.153.81) fails to yield any results:

```
[rudy@localhost alerts]$ telnet MY.NET.153.81 21
Trying MY.NET.153.81...
telnet: connect to address MY.NET.153.81: Connection refused
telnet: Unable to connect to remote host: Connection refused
[rudy@localhost alerts]$ ftp MY.NET.153.81
ftp: connect: Connection refused
```

Since we cannot determine what daemon this host was running it is hard to tell for certain if there is a danger of Denial of Service on this host. What we can do is observe the nature of the traffic in the logs to determine if the traffic looks ordinary and seems to simply be generating false positives. Inspection of the logs indicates that the traffic to this over our three day time span really only occurred for about 33 minutes:

```
03/25-16:13:58.562001  [**] FTP DoS ftpd globbing [**] 80.126.240.251:16027 ->
MY.NET.153.81:21
03/25-16:14:02.700631  [**] FTP DoS ftpd globbing [**] 24.108.229.185:33365 ->
MY.NET.153.81:21
03/25-16:14:03.051681  [**] FTP DoS ftpd globbing [**] 24.108.229.185:33365 ->
MY.NET.153.81:21
03/25-16:14:03.079163  [**] FTP DoS ftpd globbing [**] 24.108.229.185:33365 ->
MY.NET.153.81:21
~
~
03/25-16:47:14.813443  [**] FTP DoS ftpd globbing [**] 80.126.240.251:16027 ->
MY.NET.153.81:21
03/25-16:47:15.177355  [**] FTP DoS ftpd globbing [**] 217.155.204.164:63491 ->
MY.NET.153.81:21
03/25-16:47:26.575815  [**] FTP DoS ftpd globbing [**] 24.126.121.235:62115 ->
MY.NET.153.81:21
03/25-16:47:30.626910  [**] FTP DoS ftpd globbing [**] 217.155.204.164:63491 ->
MY.NET.153.81:21
```

More importantly, the source port for each host does not change over this time span. No strange new connection causes this logging to abruptly stop. Finally this traffic was in a volume large enough to generate 321 alerts, nearly 10 alerts per minute, implying an active file transfer session. Since globbing is common and efficient we can initially make the assumption that these are false positives. Simply assuming does not indicate an accurate analysis so it would be with in due diligence to correlate this alert data with system logs on the system to verify that no compromise or denial of service has taken place. Checking ability to access systems anonymously would also be helpful because these attacks do require some sort of access to implement.

Most interesting is our second host (MY.NET.24.27) which we are able to connect to and reveals something quite disturbing. Upon connecting to this host we can prove that it is running a vulnerable version of the very same wu-ftpd software that this alert was designed to identify. The following are the results of a banner grab ran on MY.NET.24.27:

```
[rudy@localhost alerts]$ telnet MY.NET.24.27 21
Trying MY.NET.24.27...
Connected to ragnarok[4].uxyz.edu (MY.NET.24.27).
Escape character is '^]'.
HELLO
220 ragnarok.uxyz.edu FTP server (Version wu-2.6.1(3) Thu Jun 28 19:17:44 EDT 2001) ready.
530 Please login with USER and PASS.
```

According to CVE-2001-0550, wu-ftpd 2.6.1 allows remote attackers to execute arbitrary commands via a "~{" argument to commands such as CWD, which is not properly handled by the glob function (ftpglob).[5]

Now, if we are able to obtain access, even anonymous access, to this system it is possible to execute arbitrary commands or subsequently carry out a denial of service attack as described on Bugtraq.[6] It is recommended that this host be isolated and analyzed for signs of compromise or denial of service being carried out.

To detect this dangerous situation with the Snort IDS we can write signatures similar the following that utilize flowbits. Snort versions 2.0 and great allow state tracking with the flow preprocessor. The following signature set can alert us if a user logs in to a potentially vulnerable FTP server anonymously:

```
alert tcp $HOME_NET 21 -> $EXTERNAL_NET any (content:"Version wu-2.6.1"; msg:"Vulnerable FTP server in use"; flags:S; flowbits:set,ftp.vuln; classtype:attempted-dos;)

alert tcp $HOME_NET 21 -> $EXTERNAL_NET any (content:" 230 Guest login ok, access restrictions apply"; msg:"User Logged On anonymously to Vulnerable FTP server"; flowbits:isset,ftp.vuln; flags:A;)
```

Detect Generation:

This detect was generated by a Snort intrusion detection system. The rule is however outdated but a search of the snort rule data base reveals that the current snort signatures regarding this attack now have SIDs 1377 and 1378, "FTP wu-ftp bad file completion attempt [" and "FTP wu-ftp bad file completion attempt {" respectively[7]:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP wu-ftp bad file completion attempt {"; flow:to_server,established; content:"~"; content:"{"; distance:1; reference:bugtraq,3581; reference:bugtraq,3707; reference:cve,2001-0550; reference:cve,2001-0886; classtype:misc-attack; sid:1378; rev:14;)
```

---

[4] In Norse mythology Ragnarok is the doom of the gods. After an ice age human kind will be destroyed and the gods will then do battle; ushering in the new golden age.

[5] http://cve.mitre.org/cgi-bin/cvename.cgi?name=2001-0550

[6] http://www.securityfocus.com/bid/3581/discussion/

[7] http://www.snort.org/snort-db/sid.html?id=1378 and http://www.snort.org/snort-db/sid.html?id=1377

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP wu-ftp bad file completion attempt [";
flow:to_server,established; content:"~"; content:"["; distance:1; reference:bugtraq,3581; reference:bugtraq,3707;
reference:cve,2001-0550; reference:cve,2001-0886; classtype:misc-attack; sid:1377; rev:14;)
```

Probability of a spoof:

Establishing a connection to an FTP server requires an established TCP session with a three way handshake and bidirectional interaction afterward. It is very unlikely that an attacker would be able to spoof this attack. It is possible that through some port magic that the packets could be bounced through a couple intermediate hosts making the end location harder to reveal but this would require the attacker to control multiple systems from which to redirect traffic. Spoofing is not considered feasible in this situation.

Attack Mechanism:

The software in question, wu-ftpd, uses its own code to handle globbing characters, most other programs use the underlying system libraries to accomplish this. According to CERT advisory 2001-33:[8]

"When the globbing code is called, it allocates memory on the heap to store a list of file names that match the expanded glob expression. The globbing code is designed to recognize invalid syntax and return an error condition to the calling function. However, when it encounters a specific string, the globbing code fails to properly return the error condition. Therefore, the calling function proceeds as if the glob syntax were correct and later frees unallocated memory that can contain user-supplied data. If intruders can place addresses and shell code in the right locations on the heap using FTP commands, they may be able to cause WU-FTPD to execute arbitrary code by later issuing a command that is mishandled by the globbing code."

If an attacker can successfully generate a string containing globing characters and some shell code that is large enough to create a buffer overflow, he or she then has the ability to access a shell with root privileges. In other cases bad processing of the characters can create a denial of service condition via a segmentation fault as described in a report on the heap corruption vulnerability by Securiteam.[9]

Correlation:

The wu-ftpd globbing vulnerabilities are widely documented. There are two Bugtraq IDs associated with this vulnerability, IDs 3581 and 3707, both referenced earlier in this discussion. There are two CERT advisories regarding these vulnerabilities: CA-2001-33 and CA-2001-07. There are also two Common Vulnerabilities and Exposure listings regarding globbing vulnerabilities,

[8] http://www.cert.org/advisories/CA-2001-33.html
[9] http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-11/0065.html

CVE-2001-0550 and CVE-2001-0886.[10] Several SANS GCIA students discussed this vulnerability in depth in their practical assignments including Wouter Claire[11], Michael Holstein[12], and Brian Granier[13]

Active Targeting:

There were no additional events to correlate out side of the globbing for these hosts. There were a few hits in the scan log but all were part of a scan of the entire address space. No probes for ftp service ports were done for either of these addresses. This does not rule out, however, the possibility of information gathering done via a scan out side the temporal domain of this analysis. This should not detract from the fact that this is a very vulnerable service running on a system that is possibly critical.

Severity:

Criticality: 4 - It is likely that this is part of a service cluster that would be available to the entire University Network. It is not entirely impossible that this is a personal machine but we should err on the side of caution and assume it contains sensitive data. Wu-ftpd runs mainly on Linux or Unix variants, there is a slight possibility that it is running under a Win32 port.

Lethality: 4 – Successful compromise can lead to root access on the machine, this is however through a complicated buffer overflow. Failed attempts with most likely result in a segmentation fault and thus a denial of service.

System Countermeasures: 0 – The software on this host is known to be vulnerable and the vulnerable version number is further more advertised in the banner. Anonymous access is granted to this host, leaving it completely open to attack.

Network Countermeasures: 0 – Access is not restricted to this server as we could grab a banner and log on to this host from any external network.

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

---

[10] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0886
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0550

[11] http://www.giac.org/practical/Wouter_Claire_GCIA.pdf

[12] http://www.giac.org/practical/Michael_Holstein_GCIA.doc

[13] http://www.giac.org/practical/GCIA/Brian_Granier_GCIA.pdf

Severity = (4+4) – (0-0) = 8

Detects:

2. NIMDA infected hosts.

```
03/27-17:02:26.181572  [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize [**]
MY.NET.97.242:3083 -> 130.189.128.52:80
03/27-17:02:26.421057  [**] NIMDA - Attempt to execute cmd from campus host [**]
MY.NET.97.242:3083 -> 130.189.128.52:80
```

Description: Correlation between several alerts and scan logs in the data set indicate that there are several hosts on the network that are infected with the NIMDA worm or a variant. NIMDA is a mass mailing worm that spreads via email. It is more sophisticated than the Code Red worm that plagued the internet in 2002. After a host is infected by NIMDA the worm scans hosts for web servers running Microsoft IIS 4.0 or 5.0 and propagates through directory traversal vulnerabilities. A directory traversal vulnerability is one where an attacker will use Unicode extensions to attempt to access or launch files in a directory relative to the root directory of a web server. An example would be: http://www.victim.com/../../c/winnt/system32/cmd.exe?/c+dir.
If by chance the victim machine is vulnerable to this attack a command shell could be acquired with system privileges.

This combination of IDS alerts triggered for two internal hosts and 56 external hosts. Since the NIMDA worm chooses its scanning targets somewhat randomly they are not as much of an interest as the internal hosts are. The two hosts that we will look at in this detect are MY.NET.97.242 (ppp2-242.dialup.uxyz.edu) and MY.NET.97.12 (ppp1-12.dialup.UXYZ.EDU).

According to CERT Advisory CA-2001-26 NIMDA has the following impact:

"Intruders can execute arbitrary commands within the Local System security context on machines running the unpatched versions of IIS. In the case where a client is compromised, the worm will be run with the same privileges as the user who triggered it. Hosts that have been compromised are also at high risk for being party to attacks on other Internet sites. The high scanning rate of the Nimda worm may also cause bandwidth denial-of-service conditions on networks with infected machines."[14]

Reasoning for selection: There are a number of hosts with various worms currently operating on the network in question. Worm infections usually indicate a deficiency in software updates and operating system patching. The fact that NIMDA is over two years old at the time of the detect makes it an interesting topic for discussion here. These infected worms that mainly scan networks to propagate cause an unnecessary volume of traffic that limits bandwidth to those wishing to use it for legitimate purposes.

Often time's links are written is such a way that they create false positives for these signatures, the following is a process to help distinguish bonafide NIMDA traffic from false positives. The first step would be to identify the sorts of traffic that are triggering the alerts. If we look at the external destination addresses that trigger these alerts we will see the following results displayed in table 3.3:

[14] http://www.cert.org/advisories/CA-2001-26.html

Table 3.3 External Destinations for NIMDA Traffic

IIS Overflow		NIMDA cmd Attempt	
Alerts	External Destination	Alerts	External Destination
11	69.90.32.141	6	130.223.102.249
2	67.123.169.93	5	130.223.8.221
2	130.227.229.33	5	130.223.242.28
2	130.223.121.225	5	130.223.167.130
2	130.194.13.177	4	130.223.249.208
2	130.158.95.113	4	130.223.131.46
2	130.158.40.102	2	67.123.169.93
2	130.158.102.214	2	130.94.93.199
1	220.60.79.122	2	130.36.90.190
1	220.31.62.33	2	130.227.229.33
1	207.44.250.46	2	130.227.165.214
1	130.94.93.199	2	130.223.119.29
1	130.94.75.60	2	130.194.13.177
1	130.94.216.231	2	130.160.197.138
1	130.94.159.103	2	130.158.95.113
1	130.89.1.16	2	130.158.74.142
1	130.75.157.43	2	130.158.54.35
1	130.36.90.190	2	130.158.40.102
1	130.239.204.208	2	130.158.129.153
1	130.236.33.89	2	130.158.102.214
1	130.234.78.31	1	64.112.195.140
1	130.219.32.114	1	220.60.79.122
1	130.209.105.110	1	220.31.62.33
1	130.207.87.13	1	212.45.11.104
1	130.205.150.39	1	207.44.250.46
1	130.189.128.52	1	130.94.75.60
1	130.161.1.194	1	130.94.159.103
1	130.160.197.138	1	130.89.1.16
1	130.158.89.241	1	130.75.157.43
1	130.158.74.142	1	130.239.204.208
1	130.158.70.231	1	130.236.33.89
1	130.158.54.35	1	130.234.78.31
1	130.158.45.212	1	130.223.121.225
1	130.158.41.239	1	130.219.32.114
1	130.158.29.127	1	130.209.105.110
1	130.158.233.94	1	130.207.87.13
1	130.158.194.72	1	130.207.82.4
1	130.158.171.211	1	130.205.150.39
1	130.158.141.252	1	130.189.128.52
1	130.132.80.52	1	130.161.1.194
		1	130.158.89.241
		1	130.158.70.231
		1	130.158.49.230
		1	130.158.45.212
		1	130.158.41.239

		1130.158.29.127
		1130.158.233.94
		1130.158.194.72
		1130.158.171.211
		1130.158.147.156
		1130.158.141.252
		1130.132.80.52

We see here that a large subset of the IIS over flow attempts also appear on the list for the NIMDA alerts. Given these entries targets would see traces such as this in their system logs:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET
/msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
```

Only one or two of these web requests are generating the alerts that we see, but these requests are highly out of the ordinary and not even necessary for normal http traffic.

Another clue that indicates that these two hosts are legitimately infected with NIMDA is the address space that they are scanning for exploits fits the modus operandi of the worm. According to the CERT advisory on NIMDA referenced earlier a NIMDA host will select a list “of potential target IP addresses follows these rough probabilities:

> * 50% of the time, an address with the same first two octets will be chosen
> * 25% of the time, an address with the same first octet will be chosen
> * 25% of the time, a random address will be chosen “

It is likely that the NIMDA infection here is operating under the second case choosing the first octet. We can clearly see from table 3.3 that the scanning is occurring for pseudo-random hosts on the 130.0.0.0/8 network with a few random addresses in between. One anomalous point on the chart to external host 69.90.32.141 11 times may be legitimate behavior from the user but it should not detract from the fact that this is likely a NIMDA infection.

Last but not least the temporal information about these alerts tells us that this is

most likely an automated scan. Alerts for host MY.NET.97.12 started with time stamp 03/27-21:02:22.320873 and ended with time stamp 03/27-21:06:20.810002, a period of about four minutes touching 21 external hosts. The MY.NET.97.242 host alerts ended at 03/27-17:23:59.599562 and started at 03/27-17:02:25.268850 for a total 68 external hosts in a matter of in 21 minutes. It is my experience that normal http clients do not make requests on 21 hosts in a matter of four minutes nor do they visit greater than three distinct sites per minute for 21 minutes straight. It is also unlikely that a dial up user will be running any sort of caching engine or anything that would generate this traffic aside from a worm.

Detect Generation:

This detect was generated by a Snort intrusion detection system. These signatures are not in the default snort rule base and are likely custom written by the university staff. Concerning the IIS Overflow signature there is a slight match available in the arachNIDS at Whitehats.com:

alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS552/web-iis_IIS ISAPI Overflow ida"; dsize: >239; flags: A+; uricontent: ".ida?"; classtype: system-or-info-attempt; reference: arachnids,552;)[15]

In order to modify the above signature to match the one used in our detects we would simply exchange the internal and external variables and adjust our flags, the signature in question would read:

alert TCP $INTERNAL any -> $EXTERNAL 80 (msg: "IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize"; dsize: >239; flags: S+; uricontent: ".ida?"; classtype: system-or-info-attempt; reference: arachnids,552;)

The best guess at the signature for the signature reading “NIMDA - Attempt to execute cmd from campus host” would be:

alert TCP $INTERNAL any -> $EXTERNAL 80 (msg: “NIMDA - Attempt to execute cmd from campus host” dsize: >239; flags: S+; uricontent:”cmd.exe” classtype: system-or-info-attempt;).

The actual signature in use may likely be more specific that the one proposed but the basic concept of an internal host to an external host on port 80 containing a GET command and cmd.exe should suffice.

Attack Mechanism:

NIMDA initially spreads either through blank emails with an attachment or is downloaded from a compromised Microsoft IIS server. Once a host is infected it will begin to scan a pseudo random address space as specified earlier for other vulnerable IIS servers and begin to propagate. Once a shell is obtained through an IIS vulnerability the worm is transferred via TFTP and infects the vulnerable system. Once installation and scanning is complete NIMDA will run a mass mailing routine about every 10 days from the infected system with its native SMTP server and an MX record it gets from a local DNS entry. More detailed

[15] http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids552&view=signatures

information is available through Symantec[16].

Probability of a spoof:

Spoofing is almost uncertain in this case. It is evident that the dial up hosts are infected with some sort of NIMDA variant and the sources are internal to external passing through an IDS system that is likely near the perimeter of the network. This is also a reconnaissance attempt by the worm so its need for the ability to establish a connection for a shell would rule out any spoofing. A connection of this type requires a three way handshake between two hosts which involves exchange of random sequence numbers. Spoofing is not considered feasible in this situation.

Correlation:

Information on NIMDA is widely available on the internet. It is the subject of CERT Advisory 2001-26 as discussed earlier and Symantec's web site has a good analysis of it. NIMDA takes advantage of the vulnerabilities addressed in the MS01-044 security bulletin[17], US-CERT vulnerability VU#111677[18], and CERT advisory CA-2001-12[19]. These detects were also discussed in SANS GIAC GCIA practical assignments by Donald Gregory[20], Glenn Larret[21], and Al Williams[22].

These two hosts account for a significant amount of scanning in the scan logs, although not enough to make the top talkers list. This could indicate that there may be multiple infections on a few of these machines. If logs other than the IDS ASCII logs were available for analysis it may likely show a more definitive trace of NIMDA traffic. Again, there is no concrete evidence in this data that a firewall exists but this information could be in the firewall logs.

Active Targeting:

It is unlikely that there is active targeting at play here. This worm is in high distribution on the internet and was just dying down at the time of these detects. The worm spreads randomly and can be contracted by any random user not savvy to the dangers of accepting unsolicited files from either a web site or a random email. This host does show up in the scan logs every day and was the subject of some outside vulnerability scans. Heavy scanning does occur during this time and does reveal that the host scans more hosts than show up due to

[16] http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html
[17] http://www.microsoft.com/technet/security/bulletin/MS01-044.mspx
[18] http://www.kb.cert.org/vuls/id/111677
[19] http://www.cert.org/advisories/CA-2001-12.html
[20] http://www.giac.org/practical/GCIA/Donald_Gregory_GCIA.pdf
[21] http://is.rice.edu/~glratt/practical/Glenn_Larratt_GCIA.html
[22] http://www.whitehats.ca/main/members/ Herc_Man/Files/Al_Williams_GCIAPractical.pdf

IDS alerts. Due to the fact that this host is on the dialup network it is likely that this is a different user each day and even a different user at different times of the day.

Severity:

Criticality: 4 - NIMDA predominantly spreads through Microsoft IIS vulnerabilities. The level of patching on the current system would determine how wide this worm would spread internally. A massive infection can negatively impact the performance of the entire network. Critical systems are most likely running Linux or Unix or something not-Windows. There is a likely hood of an IIS web server being vulnerable so that will increase the criticality slightly.

Lethality: 4 – NIMDA can damage documents and files on the infected host as well as eat up bandwidth degrading network performance.

System countermeasures: 3 – Most of the machines infected are dial up users so it is assumed that they are users with remote systems that are not up to date on patching.

Network countermeasures: 0 - There seems to be no block of the outbound scans that these NIMDA infections are doing. A peek into the scan logs shows the worms did get a few responses.

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

Severity = (4+4) – (3-0) = 5

Detects:

IRC Zombie Networks:

```
03/25-21:15:31.107170  [**] [UXYZ NIDS IRC Alert] IRC user /kill detected, possible trojan. [**]
139.165.206.128:6666 -> MY.NET.97.209:3065
03/25-20:59:44.499360  [**] [UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to
IRC [**] MY.NET.97.209:3041 -> 139.165.206.128:6666
03/25-20:59:45.214150  [**] [UXYZ NIDS IRC Alert] IRC user /kill detected, possible trojan. [**]
139.165.206.128:6666 -> MY.NET.97.209:3041
03/25-21:20:56.851066  [**] [UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to
IRC [**] MY.NET.97.209:4280 -> 139.165.206.128:6666
03/25-21:14:25.511709  [**] [UXYZ NIDS IRC Alert] Possible drone command detected. [**]
139.165.206.128:6666 -> MY.NET.97.209:3065
```

Description: This detect is the result of the correlation between several alerts and traces in the scan logs. The following three rules triggered for 47 different internal addresses and 31 external addresses:

```
[UXYZ NIDS IRC Alert] IRC user /kill detected, possible trojan.
[UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC
[UXYZ NIDS IRC Alert] Possible drone command detected
```

These alerts pertain to Internet Relay Chat traffic on the network. These signatures detect traffic that maybe associated with IRC trojan “bot nets” or “zombie networks”. These signatures are also prone to false positives so a more detailed look at the traffic will determine if this is legitimate IRC traffic or malicious worm traffic. A common trend among newer worms in to hook into zombie networks running on IRC channels to receive instructions after they have compromised their victims. These worms can have various purposes varying from mass mailing SPAM, to creating denial of service through traffic floods, to performing reconnaissance to spread to new vulnerable systems. The link graph in Figure 3.2 illustrates how these hosts would interact in a zombie network:

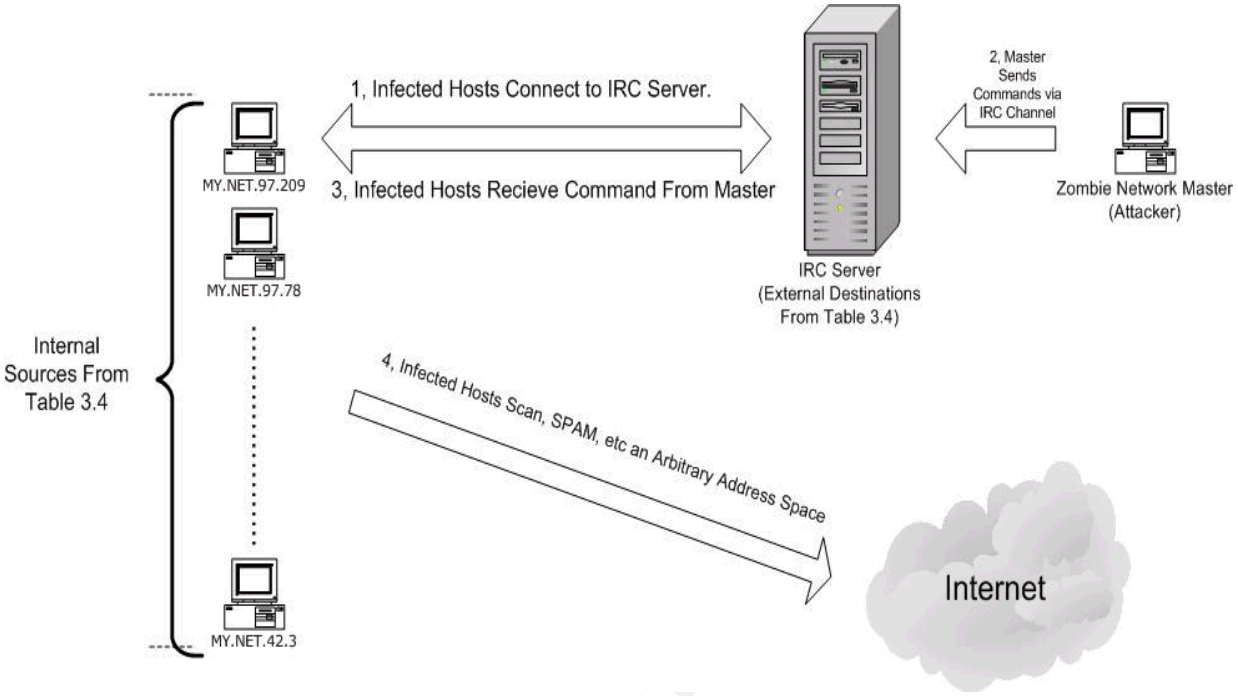

Figure 3.2 Zombie Network Link Graph

Table 3.4 shows a summary of the internal and external hosts in question for this detect.

Table 3.4 Internal And External Hosts for [UXYZ NIDS IRC Alert] Alerts

	Internal Sources		External Destinations
Alerts	Address	Alerts	Address
54	MY.NET.97.209	368	139.165.206.128
51	MY.NET.97.78	26	209.126.201.103
47	MY.NET.97.235	11	205.177.13.100
39	MY.NET.97.102	10	199.184.165.133
36	MY.NET.97.113	7	206.252.192.194
34	MY.NET.97.92	4	69.42.74.6
32	MY.NET.97.50	4	207.36.180.241
26	MY.NET.111.51	4	204.152.184.80
25	MY.NET.97.185	3	216.109.195.222
25	MY.NET.60.11	3	209.25.160.96
11	MY.NET.98.49	2	69.36.232.118
11	MY.NET.97.159	2	69.28.182.110
9	MY.NET.97.218	2	209.17.76.190
8	MY.NET.97.124	2	195.149.88.251
7	MY.NET.70.16	1	81.27.36.155
6	MY.NET.97.25	1	69.31.68.94
6	MY.NET.97.170	1	66.199.180.252
6	MY.NET.82.79	1	66.198.160.2
5	MY.NET.42.4	1	66.10.76.20
5	MY.NET.153.94	1	64.85.20.76
3	MY.NET.97.187	1	64.124.166.200
3	MY.NET.60.40	1	64.124.0.204
2	MY.NET.97.197	1	217.17.33.10

2	MY.NET.97.131	1	216.82.127.45
2	MY.NET.150.207	1	205.207.137.30
1	MY.NET.97.99	1	205.207.137.3
1	MY.NET.97.79	1	198.252.195.2
1	MY.NET.97.70	1	198.175.186.5
1	MY.NET.97.66	1	195.169.138.124
1	MY.NET.97.47	1	193.163.220.3
1	MY.NET.97.24	1	130.233.48.242
1	MY.NET.97.204		
1	MY.NET.97.19		
1	MY.NET.97.189		
1	MY.NET.97.179		
1	MY.NET.97.177		
1	MY.NET.97.174		
1	MY.NET.97.164		
1	MY.NET.97.163		
1	MY.NET.97.156		
1	MY.NET.97.111		
1	MY.NET.97.104		
1	MY.NET.97.103		
1	MY.NET.84.145		
1	MY.NET.60.16		
1	MY.NET.42.3		
1	MY.NET.1.3		

As you can tell a large majority (about 2/3) of the internal hosts in question lie on the MY.NET.97.0/24 network which is a PPP dial-in block. Since this data spans a three day time span its is quite possible that some of these addresses are being assigned to the same infected hosts that disconnect from the network and then re connect at a later time. There are also several machines on the list that belong to libraries or departments at the university so this detect should not be dismissed to the fact that there may be a large amount of careless students using the university network.

Reason for Selection: Agobot variants can allow attackers to use network resources for purposes that are often times illegal in some states or countries. The top talker on the list for this detect is among the top five total talkers for the entire network. (A discussion of top network talkers will come later). It is important to note that a host that is a top talker does not necessarily imply he is a top user of bandwidth.

It is evident that not only are these machines compromised but they are carrying out illicit activities under the control of the zombie network. This is evident by correlating the scan logs with the alert logs. Here is one such example:

From our alert logs:
03/25-21:03:31.414673 [**] [UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC [**] MY.NET.97.209:3065 -> 139.165.206.128:6666

We see a possible sdbot connection to an IRC server. And a few seconds later this scan occurs:

```
Mar 25 21:03:34 MY.NET.97.209:3122 -> 130.24.0.163:2745 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3124 -> 130.24.0.163:1025 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3126 -> 130.24.0.163:3127 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3127 -> 130.24.0.163:6129 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3129 -> 130.24.0.163:80 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3162 -> 130.223.17.130:2745 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3164 -> 130.223.17.130:1025 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3166 -> 130.223.17.130:3127 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3167 -> 130.223.17.130:6129 SYN ******S*
Mar 25 21:03:34 MY.NET.97.209:3169 -> 130.223.17.130:80 SYN ******S*
```

This scan touches 38456 distinct hosts and probes for ports 2745, 1025, 3127, 6129, and 80. These are in no way a random collection of ports each has the possibility of allowing the worm to propagate through their respective vulnerabilities. If we reference the port list at linklogger.com and other internet resources we find that these ports associate with Bagel backdoor port[23], Microsoft RPC[24], myDoom backdoor[25], Dameware[26], and IIS vulnerabilities respectively. This worm is likely looking to propagate via back doors left by other mal ware on the probed host.

The University staff has created these rules for their own intrusion detection system. This tells us that they are likely aware that these infections exist and are likely trying to identify the infected hosts.

Detect Generation: This detect was generated by a Snort intrusion detection system. These signatures are not in the default snort rule base and are likely custom written by the university staff. We can make our best guess at the signature in these rules through experience with snort rules.

[UXYZ NIDS IRC Alert] IRC user /kill detected, possible trojan.

alert tcp $EXTERNAL 6667 -> $INTERNAL any (msg: "[UXYZ NIDS IRC Alert] IRC user /kill detected, possible trojan."; content:"/kill"; established;)

Assuming that the trigger will alert where any internal host receives a "/kill" command from an IRC channel.

[UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC

alert $INTERNAL any -> $EXTERNAL 6667 (msg: "[UXYZ NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC"; flags:S+;)

It is difficult to tell what content will differentiate an sdbot floodnet from a normal IRC connection so this proposed signature alerts on all new IRC connections.

[UXYZ NIDS IRC Alert] Possible drone command detected

[23] http://www.linklogger.com/TCP2745.htm
[24] http://www.linklogger.com/TCP1025.htm
[25] http://www.linklogger.com/TCP3127.htm
[26] http://www.linklogger.com/TCP6129.htm

```
alert tcp $EXTERNAL 6667 -> $INTERNAL any (msg: "[UXYZ NIDS IRC Alert] Possible drone command
detected" content:<some known drone command>; established;)
```

We can complete this signature with <some known drone command>, it will alert when those commands are detected coming from an IRC server to the "infected host" on an established connection.

Attack Mechanism:

Worms utilizing IRC Trojans generally spread through vulnerabilities in software. There are several popular exploits. The infections in this case seem to be looking to spread through backdoors left open by previous infections such as myDoom, the Bagel virus, and NIMDA (or variants of these). It is very likely that these infections are actually a variant of the Agobot IRC trojan. Agobot is an extremely robust trojan that scans for the vulnerabilities we have discussed earlier. Its features allow it to be used for phishing username: password pairs, email addresses, and even sending SPAM. An excellent analysis of Phatbot (a peer to peer implementation of Agobot) is available from the LURHQ Corporation[27].

Probability of a spoof:

Spoofing is almost uncertain in this case. It is evident that the hosts are infected and are the machines that are actually carrying out the attack, mostly in the form of scanning. Since a TCP connection is required to log in to the IRC server it is also unlikely that the control channel connection is spoofed. A controlled connection to the bot net not only involves a three way handshake between two hosts but the victim must enter an IRC channel and receive more commands from the master. This implies an active session virtually eliminating the possibility of spoofing.

Correlation: There is much correlation that can be done regarding researching this activity. Since the scope of such IRC Trojans is so large there are many vulnerabilities that can be taken advantage of. Symantec has several analysis and documentation on worms that utilize IRC control channels to spread or server out their purpose.

McAfee has information about W32/Polybot.I!irc[28] irc trojan worm. F-secure has documentation regarding Agobot[29] variants. LURHQ Corporation, as discussed earlier, has documented the Phatbot IRC Trojan.[30]

The CVE has many CAN entries that are exploited by these worms and their variants five of the most popular would be: CVE-2003-0109, CVE-2003-0352,

[27] http://www.lurhq.com/phatbot.html
[28] http://vil.nai.com/vil/content/v_101100.htm
[29] http://www.f-secure.com/v-descs/agobot.shtml
[30] http://www.lurhq.com/phatbot.html

CVE-2003-0533, and CVE-2003-0717.

The five most popular Microsoft vulnerabilities exploited are discussed in the following bulletins: MS-03-001, MS03-007, MS03-026, and MS03-043.

This activity has also been discussed in GCIA practicals by Peter Storm[31], Mihai Cojocea[32], Wouter Claire[33], and Patrik Sternudd[34].

Active Targeting: It would be practical to call this activity active targeting. Although the worm is spread autonomously it would seemingly be so that the commands to spread it form one network to another via scanning is certainly possible. After the infected host is on the control channel it is fully under the control of the master and will actively carry out the commands as instructed. We can consider this a form of active targeting.

Severity:

Criticality: 4 - These types of worms predominantly spread through Microsoft vulnerabilities. The level of patching on the current system would determine how wide this worm would spread internally. A massive infection can affect the performance of the victim machines greatly. Critical systems are most likely running Linux or Unix or something not-Windows. It is likely that some hosts generating these alerts are Unix machines running IRC clients but all legitimate infections would be on a Win32 machine.

Lethality: 5 – Machines with these Trojans are vulnerable to information disclosure, give the zombie network master complete control of the system, and drastically decrease the performance of the system.

System countermeasures: 3 – Most of the machines infected are dial up users so it is assumed that they are users with remote systems that are not up to date on patching. Despite a few internal hosts that are compromised out side the dial up block this infection does not seem to be terribly wide spread.

Network countermeasures: 0 - There seems to be no block of communications between the IRC server and the host. It would be wise to block the ports for IRC servers at the firewall, if there is one. It is evident that this is not the case in this scenario. Outgoing scans do not appear to be blocked either.

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

---

[31] http:// www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf

[32] http://www.giac.org/practical/GCIA/Mihai_Cojocea_GCIA.pdf

[33] http://www.giac.org/practical/GCIA/Wouter_Claire.pdf

[34] http:// www.giac.org/practical/GCIA/Patrik_Sternudd_GCIA.pdf

Severity = (4+5) – (3-0) = 6

4. Network Statistics

Top Talkers: The following two tables show the top talkers on the University network from March 25th to March 27th 2004. It is important to note before continuing that the top talkers do not implicate that these hosts are the primary bandwidth hogs, simply that they cause the IDS to alert most frequently on their traffic. Top talkers are obtained simply by extracting the source addresses from each log file type and tallying up the occurrences. We will simply identify the hosts generating the most traffic on the University network and then discuss if the results are to be expected or not. Table 4.1 lists the internal hosts talking in each log file and Table 4.2 lists the external hosts talking in each log file. Details as to how these statistics were obtained are contained in Part III of this report.

Table 4.1 Internal Top Talkers

Scans Log files	Alerts Log Files	Out of Spec. Log files
Entries IP Address	Entries IP Address	Entries IP Address
294746 MY.NET.1.3	101056 MY.NET.1.3	806 MY.NET.6.7
86896 MY.NET.70.229	83137 MY.NET.1.4	550 MY.NET.12.6
70026 MY.NET.1.4	40527 MY.NET.190.92	195 MY.NET.24.44
28885 MY.NET.97.209	36521 MY.NET.84.235	60 MY.NET.34.11
17738 MY.NET.97.39	22872 MY.NET.30.3	57 MY.NET.12.4

Host MY.NET.1.3 shows up most frequently in both the scan and alert files. Examining the network topology (discussed above), we find this host is a DNS server. After inspecting the alerts associated with this address we find that they are mainly port scans. Correlating this information with the scan log data reveals that this is UDP traffic out bound to other DNS servers, most likely for name resolution lookups. Another name server (1.4) ranks 3rd on the scan list and second on the alert list for the same reasons. MY.NET.70.229 matched a few signatures beyond its scan logs but most were in regards to port 2612, which research shows is a Qpasa Agent used for network infrastructure monitoring. MY.NET.97.209 had a fair amount of scans but did not trigger any alert signatures aside from port scans. Further inspection of the scan logs shows us that this host is active on ports 2745, 1025, 3127, and 6129. These ports are associated with the bagel virus, RPC, the Agobot worm, and Dameware respectively. This host is at high risk and most likely infected with several variants of malware. RPC and Dameware are often used by malicious programs to control compromised machines[35]. The 97.209 host also had activity

on port 6666 which is a common IRC port. Using IRC channels to control armies of infected hosts is a current trend among malware and is most likely associated with the Agobot activity on this host. Common sense dictates that these ports should be blocked at firewalls (if they are present), and this host isolated and disinfected. MY.NET.97.39 registered a fair amount of port scans, mostly for port 6346 which is associated with gnuetella (a common file sharing peer to peer application).[36]

MY.NET.190.92 and MY.NET.84.235 made the top talker list regarding alerts but also because they hit on so many port scans. MY.NET.30.3 closed out the top five talkers amongst the alert files; this is simply because there is apparently a rule in the snort configuration that logs all traffic to this host. We suspect that host contains some sort of sensitive information, or is currently being audited.

OOS packets can indicate the use of covert channels or poorly written/misconfigured applications. All of these hosts are main servers (uxyz8,mxin,userpages,web1.cs, and mail respectively) for the network. They are negotiating on the ECN protocol which we will discuss later in this section.

Table 4.2 External Top Talkers

Scans Log files	Alerts Log Files	Out of Spec. Log files
Entries IP Address	Entries IP Address	Entries IP Address
27005 213.120.116.27	13332 68.32.127.158	787 68.54.84.49
25126 81.226.62.240	10491 80.181.112.186	105 66.225.198.20
23374 203.68.87.44	9317 67.31.152.200	59 67.114.19.186
13719 195.146.221.38	6166 68.55.174.94	54 35.8.2.252
10114 209.34.46.165	5233 68.54.84.49	46 68.115.197.90

Examining the log files of the top talkers regarding scans we find several similarities. Three of the five top talkers are scanning the internal address space for openings on port 20168. Port 20168 is most commonly associated with the Lovegate mass mailer worm[37]. This worm was introduced in the summer of 2003 and is still in high distribution of the internet according to Symantec's security response website. The sources are also located in various differing geographic locations such as Britain, Virginia, and the Dominican Republic. This activity does not appear to be a directed attack on the network but rather the worm carrying out its routine procedure to identify and spread to any possible vulnerable host.

[35] http://www.linklogger.com/TCP6129.htm
[36] http://lists.jammed.com/incidents/2001/07/0310.html
[37] http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.lovgate.j@mm.html

The remaining two hosts in the scan file both are scanning for Dameware on the UXYZ address space. As discussed earlier Dameware is used by several worms to remotely administer or control the hosts which are infected. This also is likely not a directed attack on the network but simply a reconnaissance attempt by a control client of the worm to execute commands on a remote host that may be infected, most likely for sending out mass emails. The sources identified here are from the Baltic Region and Texas, so they are not concentrated in any one geographic location. There are notes at the SANS Internet Storm Center that indicate that this is Dameware. There are buffer overflow vulnerabilities associated with older Dameware versions[38] so closely monitoring these hosts for signs of compromise is recommended.

The alert log top talkers also make up for some isolated traffic of little interest. 68.32.127.158 accounts for a majority of the alerts from the signature with the summary "connect to 515 from outside". This is a common port for print spooling according to Kurt Seifreid's port directory.[39] This activity is most likely attributed to an off campus user printing documents to a networked printer. This is further supported by the fact that a reverse DNS lookup shows that this is host is pcp0011023458pcs.arlngt01.va.comcast.net which is a Comcast address from Virginia and the source is printhost.uxyz.edu, obviously an authorized print server. It is likely that this is authorized activity considering the close proximity to the University from the source location. It may be beneficial to tweak this rule to not trigger for authorized printers or alternatively block port 515 traffic at the firewall (if there is a firewall) to all hosts that are not authorized printers.

The next host of interest is 80.181.112.186. This host is responsible for a majority of the alerts with the summary "High port 65535 tcp - possible Red Worm – traffic". Subsequently all the traffic that alerts for destination port 65535 in these logs originates from port 1122. This port is associated with Availant IT management software[40]. The source address has the name host186-112.pool80181.interbusiness.it when we run it through a reverse DNS lookup and the source address is in the dial up pool for the university. Although this activity does not appear to be malicious it may not be authorized.

The device at address 67.31.152.200 (dialup-67.31.152.200.Dial1.Denver1.Level3.net.) accounts for much of the MY.NET.30.4 alerts. This traffic is predominantly a port scan of the MY.NET.30.4 host. It is possible that it is running an NMAP scan attempt to fingerprint the operating system.

The host at 68.55.174.94(pcp0011464957pcs.chrchv01.md.comcast.net) seems to be running a client on port 1078 that is talking to the MY.NET.30.3 box on port 3019. This is triggering the MY.NET.30.3 alert but appears to be legitimate activity as 3019 is a Novell Network Resource management port and 1078 is the port that eManageCstp a network management client operates on. Consulting the network administrators would yield more information.

Host 68.54.84.49 (pcp0011109240pcs.elkrdg01.md.comcast.net) is a user

[38] http://isc.sans.org/port_details.php?port=6129

[39] http://www.seifried.org/security/ports/0/515.html

[40] http://www.availant.com/

who has an ISP local to the campus that is checking their email every minute. This is causing alerts in the form of portscans. This traffic does not appear to be hostile. The administrators may wish to track down the user and request that he/she increase the interval in which email is checked. This host also happens to be at the top of the Out Of Spec talkers list. This is due to the fact that it is using the reserve bits in the TCP packet. One such entry from the OOS logs looks as such:

```
03/25-00:06:04.037093 68.54.84.49:54989 -> MY.NET.6.7:110
TCP TTL:51 TOS:0x0 ID:61622 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x97634FCC  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 759611258 0 NOP WS: 0
```

The part of the protocol that appears "out of spec" in this case is the flag set (12****S*). After some research on the internet we come to RFC 3168[41]. RFC 3168 proposes a solution to queue drops at routers by using the most significant bits of the 14th byte to establish a connection that is congestion aware. The standard states that to establish an ECN connection the requestor sends a SYN packet with the most two significant bits in the 14th byte set. It is apparent that the email client here is attempting establish a connection that uses ECN, this is not hostile activity.

The remaining hosts in the OOS top talker list are also using this standard. 66.225.198.20 is a host from a local ISP sending email. It may be a commercial mailer but the occurrences a spread out enough to make it be legitimate, logs from the mail server would help clear that question up. 67.114.19.186 (adsl-67-114-19-186.dsl.pltn13.pacbell.net) is another local ISP connection; it is accessing the user pages web server at the university. Occurrences are always three or four requests all with significant time between so it would seem that this is someone affiliated with the university that has their home page set to this server. 35.8.2.252(mdlv2.h-net.msu.edu) is probably also sending email with a client using ECN, the fact that the activity is from another educational institution helps legitimize the traffic and is probably from an ex-student or associate professor. Finally 68.115.197.90 (static-cb-68-115-197-090.spa.sc.charter.com) seems to be carrying out some legitimate activity on the network including establishing some ssh connections, checking email, surfing the web, and using the ftp server all the packets just seem to use the ECN standard. Nothing from these talkers seems hostile; it may be possible to upgrade the snort IDS to identify this as a legitimate protocol.

Top Target Services: In order to determine the top target services the total hits for each port were determined. Afterward, an investigation was done to see which signatures these actually triggered. The list was then adjusted based on the signature's rate of false positives and chance of hostility Table 4.3 shows the top port targets from raw data:

Table 4.3 Top Target Services (Initial)

Alerts	Service	Port
16828	Novell	524

[41] ftp://ftp.isi.edu/in-notes/rfc3168.txt

13777	Print Spooler	515
10494	Availant-mgr	1122
8092	RPC	51443
5745	HTTP	80

It is now important to consider the signatures generating these counts and determine if there is actually hostile activity involved or if the traffic is generating a false positive. First, Novell traffic accounts for a large majority of the "MY.NET.30.4" alerts. This may very well be legitimate traffic but since there is a custom rule to alert for it we will keep it on the list. Port 515 triggers the "connection to 515 from outside" alert which just seems to be a recurrent print attempt and not malicious. Port 1122 is associated with the "High port 65535 tcp - possible Red Worm – traffic" alerts it appears to be legit traffic from a dial in user to a business management site but it should be investigated just in case so we will leave it on the list. RPC port 51443 triggers on various signatures and should be a concern due to vulnerabilities associated with RPC. A lot of HTTP is legit traffic but due to the fact that many worms are scanning for IIS vulnerabilities on this port we will leave it on the list. This will brings port 3091 on to the list which is responsible for "MY.NET.30.3" alerts which we will keep on the list for the same reasoning that we left the Novell port on the list. Table 4.4 shows our final decision based upon the above discussion:

Table 4.4 Top Target Services (Adjusted)

Alerts	**Service**	**Port**
16828	Novell	524
10494	Availant-mgr	1122
8092	RPC	51443
5745	HTTP	80
5499	Resource Management	3091

One final look at target services is through scans, this gives us a different perspective as these scans can reveal where current worms are looking for vulnerabilities. Table 4.5 Shows the top ports scanned.

Table 4.5 Top Target Services (Scans)

Alerts	**Service**	**Port**
223181	DameWare Backdoor	6129
149225	Trojan Backdoor	20168
99291	HTTP	80
66980	Trojan Backdoor	4899
63290	ICQ	4000
49528	VNC	5900

These are all pretty much ports of concern, three of these are known backdoor ports, HTTP can be moved lower on the list put should still be of concern due to IIS vulnerabilities and ICQ vulnerabilities[42] that can give a user root privileges. VNC can also give a user root privileges.

[42] www.lafferty.ca/software/icq/icq.html211.37.20.6

Three Most Suspicious External Hosts:

139.165.206.128 - cerm22.chim.ulg.ac.be
This host is the top talker regarding the IRC Trojan discussion earlier, it likely has the ability to control all the boxes infected with the Agobot variant discussed in section 3 detect 3. This machine is at a University in Belgium it would be wise to work with administrators there to disable this host. Whois information is available in the appendix. It is likely that this machine is running ircd on a Unix variant OS.

24.108.229.185 - S010600095b250a93.gv.shawcable.net
This is the top external talker that is accessing the ragnarok machine that is running the out dated and vulnerable wu-ftpd software. A query of the whois server indicated that the registration information was not retrievable.

67.31.152.200 - dialup-67.31.152.200.Dial1.Denver1.Level3.net.
This host accounts for much of the MY.NET.30.4 alerts, since it is not local to the university this activity of interest may wish to be investigated further to determine if it is in fact authorized. Whois information is available in the Appendix. It is hard to tell what operating system this machine runs without a fingerprint scan.

5. Correlation

An extremely large amount of research was done to find correlations to support and enhance the analysis. Much was taken from trusted internet resources, the SANS practical repository, the Internet Storm Center, and analysis by leading security companies. References and supporting information are contained through out this report and noted when appropriate. A list of references is available in the Appendix.

6. Compromised Hosts

There are approximately 67 compromised hosts operating on the network for these three detects. Each detect contains a list of internal hosts that are possibly infected or compromised. Hosts that should be investigated promptly would be ragnarok.uxyz.edu because it is running the out dated wu-ftpd software and can possibly be compromised or taken down relatively easily. Secondly any host listed to have an Agobot variant infection (detect 3) or NIMDA infection (detect 2) should be isolated and scanned for mal ware. It is important to mention that a large majority of the hosts with worms are on the MY.NET.97.0/24 network which is a dialup subnet. It may be hard to track down these users and some of the IPs noted may actually be the same user dialing in at different times. Correlating with access logs, if available, should speed up this process. Once the massive worm infections have been reduced it will become easier to identify compromised hosts on the network as there will not

be so much “noise” from the current infections.

7. Recommendations

Below is a list of recommendations to help harden and secure the network as well as prepare it for easier and quicker intrusion analysis.

- Upgrade the Snort IDS box. It is evident that system on this network is not running the most current technology available. This will reduce things such as false positives in the OOS logs, and possible eliminate bugs that cause inconsistency in the logs.
- Use the most current default snort rule set in addition to any custom rules that may be necessary for the university environment. It is clear from detect 1 that some of the rules are out of date, new rules are often more precise and lead to less false positives leading to a quicker analysis of the network traffic.
- Invest in the hardware that is necessary to log the data in pcap format and store whole sessions that trigger alerts if possible. This will allow the analyst to load the traffic and run it through different tools in order glean more insight out of the information given. Working with straight ASCII logs can often leave some questions unanswered that may be able to be answered using packet analysis tools.
- Tune the portscan preprocessor to not log on known authorized activities. Although it was very helpful in profiling the network logging traffic such as DNS and Web traffic from their authorized sources can be costly. For future analyses it maybe of benefit to have authorized activity excluded from scan logs whenever possible. An example configuration would be similar to this in the snort.conf file:

  ```
  Preprocessor portscan: MY.NET.0.0/16 8 5 /var/log/scan.log
  ```

  This configuration only logs scans including 8 or more hosts or ports in less than 5 seconds. It would also be useful to remove single packet alerts for scans such as NULL, or SYN/FIN scans by commenting them out of the scan-lib file.
- Provide firewall (if a firewall exists) and critical system logs for additional correlation. Very useful information is often contained in these logs and can help speed up analysis greatly.
- Implement a default deny policy on the firewall, if there is one. It is not known what the current policy would be but only allowing access to authorized services on authorized machines will greatly cut down on the spread of worms internally as well as successful reconnaissance of the network. This can be done rather painlessly with a fair amount if planning in my experience migrating university environments to a default deny policy.
- Use Nessus or a similar tool to scan for vulnerabilities on university maintained machines and repeat the process regularly. There is a similar

process in place at the University of Missouri.

- Draft and implement a policy for patching critical systems and other maintained machines. This will ensure that only the most recent exploits will jeopardize a system on the network.
- If possible use VLANs to segment the dialup network in case of outside worm infections from a user dialing into the network. A more extreme countermeasure would be to run a virus scan on each host as it logs on to the network to reduce the spread of infections from the outside. An excellent webcast concerning this topic is covered in a SANS webcast by Brent Deterding entitled Segmenting Networks: ACLs[43].

The best plan of action in order to implement these changes would be to follow these steps. First, is possible, locate and patch all boxes that are physically onsite. This will ensure that existing threats do not re-infect the same hosts. Second, if there is not a firewall install one before the perimeter router, otherwise, if there is a firewall migrate it to a default deny policy. The next most important step would be to upgrade the IDS signature set and create a process for frequent updates. This process should include regularly scheduled audits, upgrading, tuning, and patching. Finally, additional precautions such as virtually segmenting the network may be taken if the security posture is not satisfactory.

[43] https://www.sans.org/webcasts/show.php?webcastid=90512

Part III: Analysis Process

This section describes in detail the tools, techniques, and methodology used to arrive at the results presented in Part II of the report. These techniques and tools build on those used and presented by previous students as well as displaying the proper thought process that goes into undertaking a task such as analyzing a set of events of this magnitude.

Tools: The hardware used for this analysis includes a 1.2GHz AMD Athlon Thunderbird processor with 1.5GB of SDRAM and plenty of extra disk space to store logs and data segments. Additionally for extracting data that required much more processing power to complete the requested parses in a timely fashion the graduate student computing server at Purdue University was utilized. This computer is a 4X480MHz Processor with 4GB memory and 190G raid Ultra 450 disks. The home Athlon station was running Fedora Core 3 distribution of Linux and the computing server is running a SUN SPARC Ultra-4 5.8 Generic_117350-16 kernel.

Data was collected in plain text and manipulated using Excel on a Windows XP virtual machine using VMWare. The document was drafted in Word and the diagrams created in Visio, all Microsoft Products.

Techniques: The techniques and software tools used in analysis were kept as simple and primitive as possible. The data that was given to work with was completely in ASCII text. This makes it quick and easy to write command line scripts that will extract the needed data from the logs on the fly.

It is important to note that the data that was given in the log files is far from perfect. There were more than several instances where lines were overlapped or written over by another alert. This seems to be due to an error with the snort output plug-in or perhaps due to multiple instances attempting to log to the same file simultaneously.

The alert files contain much of the data that would reveal specific attacks that have a distinct signature, or are at least in someway distinguishable from normal network traffic. A short random sample of the alert log looks as such:

```
03/25-01:05:13.516667  [**] spp_portscan: portscan status from MY.NET.97.100: 2 connections
across 2 hosts: TCP(0), UDP(2) [**]
03/25-00:49:06.332909  [**] NMAP TCP ping! [**] 209.109.246.253:80 -> MY.NET.1.3:3968
03/25-00:49:06.427810  [**] NMAP TCP ping! [**] 216.29.45.253:80 -> MY.NET.1.3:3968
03/25-01:05:14.290127  [**] spp_portscan: portscan status from MY.NET.1.3: 29 connections across
29 hosts: TCP(0), UDP(29) [**]
03/25-01:05:14.290448  [**] spp_portscan: portscan status from MY.NET.1.4: 4 connections across 4
hosts: TCP(0), UDP(4) [**]
```

A large majority of the alert logs are from the portscan pre-processor. All of this data is contained in more detail in the scan logs. This means that to get a more detailed list of alerts with out the clutter the following shell command was used

to obtain a list of alerts with out portscan loggings:

```
zcat alert. 04032*.gz | awk '($3 !~ /spp_portscan:/){print;}' > all_alerts
```

This simple awk script prints each line that does not contain spp_portscan in the third field. The fact that the dating and delimiters from the log are uniform for the first two fields allows this to be done very quickly.

From our all_alerts file we can create a comprehensive list of alerts for which to begin pulling the needles from the haystack. The following command gives us an ordered list of alerts for our time frame:

```
cat all_alerts | cut –d']' –f2 | cut –d'[' –f1 | sort | uniq –c | sort –rn
```

This essentially gets us the summaries between the '[**]' delimiters and consolidates the duplicates (uniq –c) and then orders them again from greatest occurrence to least occurrence. This gives us a starting point for identifying deviant activity and the springing point for the rest of the analysis process.

Scan logs are invaluable for identifying individual infections, distinguishing normal activity from misconfigurations, and also for profiling a network based on traffic patterns. It was fortunate in this case that the logging level for the scan pre-processor was very sensitive. This gave plenty of information to dissect in order to get a mental picture of the network traffic. A typical scan log looks as such:

```
Mar 25 00:10:44 MY.NET.112.152:3538 -> 193.253.217.190:4661 SYN ******S*
Mar 25 00:10:44 MY.NET.112.152:3540 -> 82.82.89.15:4662 SYN ******S*
Mar 25 00:10:44 MY.NET.112.152:3542 -> 168.243.216.48:23 SYN ******S*
Mar 25 00:10:44 MY.NET.112.152:4672 -> 82.50.39.28:4672 UDP
Mar 25 00:10:44 MY.NET.112.152:4672 -> 195.202.51.137:4672 UDP
Mar 25 00:10:44 MY.NET.112.152:4672 -> 62.235.105.121:4672 UDP
Mar 25 00:10:44 MY.NET.112.152:4672 -> 82.130.146.70:4672 UDP
Mar 25 00:10:44 MY.NET.112.152:4672 -> 83.31.155.141:4672 UDP
```

These logs simply give us a timestamp, and source address and port pair, and a destination address and port pair. We can quickly obtain a list of internal top talkers with the following:

```
zcat scans.04032*.gz | awk '{for(x=2;x<=NF;x=x+1){if($x ~ /130\.85\.[0-9]+\.[0-9]+:[0-
9]+/){print $x}}}' | cut –d':' –f1 | sort | uniq –c | sort –rn
```

For top internal services:

```
zcat scans.04032*.gz | awk '{for(x=2;x<=NF;x=x+1){if($x ~ /130\.85\.[0-9]+\.[0-9]+:[0-
9]+/){print $x}}}' | cut –d':' –f2 | sort | uniq –c | sort –rn
```

For top external talkers:

```
zcat scans.04032*.gz | awk '{for(x=2;x<=NF;x=x+1){if($x ~ /[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+:[0-
9]+/){print $x}}}' | cut –d':' –f1 | sort | uniq –c | sort –rn
```

For top external services:

```
zcat scans.04032*.gz | awk '{for(x=2;x<=NF;x=x+1){if($x ~ /[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+:[0-
9]+/){print $x}}}' | cut –d':' –f1 | sort | uniq –c | sort –rn
```

Similar commands can be used to create a data set pertaining to the number of hosts talking over a specific port. For instance if we wanted a list of external hosts with traffic to port 445 we could write a quick command line such as:

```
zcat scans.04032*.gz | awk '{zcat scans.04032*.gz | awk '{for(x=2;x<=NF;x=x+1){if($x ~ /[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+:445/){print $x}}}' | cut –d':' –f1 | sort | uniq –c | sort –rn
```

As one can probably tell, this series of commands can be used as a quick template for gathering all sorts of network statistics. Once we know the top talkers we can do further correlation with DNS lookups or banner grabs to identify legitimate hosts. Best of all the data files contained import quickly into Excel spread sheets where we can get quick visual graphs of statistics to identify anomalous or malicious behavior more quickly.

Out of Specification logs can tell inform us of possible network misconfigurations or serve as additional clues to malicious activity such as bad traffic generated by a packet crafting trojan. Standard OOS logs look as such:

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

03/26-03:03:32.115219 68.54.84.49:56579 -> MY.NET.6.7:110
TCP TTL:51 TOS:0x0 ID:10770 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x737FFE36  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 769315876 0 NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/26-03:04:37.446229 68.54.84.49:56580 -> MY.NET.6.7:110
TCP TTL:51 TOS:0x0 ID:55987 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x7744371A  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 769322409 0 NOP WS: 0

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

Detailed information is given for each packet that does not match specification of the given protocol. Initially a list of top talkers was made by first creating a file with simply the headers containing the timestamp and the addresses in question. This was done with the following shell command:

```
zcat oos.04*.gz | awk '($1 ~ /03\/[0-9]+-.*/){print;}' > head
```

From the head file we created a list of top talkers in similar fashion to the alert and scan log files. If we needed to correlate some strange behavior to the oos logs that could be done quickly by using the search feature in the vi editor.

These methods of analysis are by no means new, they are simply variations on techniques used by students such as Raffael Marty[44],Peter Van Oosterom[45], and Maarten Van Horenbeeck[46] just to name a few.

---

[44] http://security.raffy.ch/projects/Raffael_Marty_GCIA.pdf
[45] http://www.giac.org/practical/GCIA/Peter_Van_Oosterom.pdf
[46] http://www.giac.org/practical/GCIA/Maarten_Vanhorenbeeck_GCIA.pdf

Methodology: Most important in analysis is the method by which one arrives to the conclusion of the cause of traffic. Intrusion detection is more of an art form rather than a routine of running a few shell scripts to parse some logs or querying a database of statistics on network traffic.

Starting with a firm foundation in TCP/IP and general networking concepts makes it significantly easier to identify behavior that is strange or possibly malicious. For instance, if a generation of a list of top talkers reveal a high number of occurrences of a source port that is not a commonly known service or is an ephemeral port it should be investigated further.

Much of intrusion analysis involves correlating with research and past events with all the data that is available to make an informed and logical decision concerning a set of data. Resources that have been invaluable during this analysis have been Google[47], the SANS Internet Storm Center[48], Kurt Siefreid's security site[49], arachNIDS[50], and of course the GIAC practical repositories[51]. Just to name a few. After much research and investigating all the possibilities concerning certain detects it is possible to create a picture of what may or may not be happening on the network in question.

An important practice to use is that of investigating the mechanism that is creating a certain event. For instance all of the data we had to work with was generated by a snort intrusion detection system so when we are investigating an alert it is just as important to investigate the cause. Some signatures are not as precise and will generate false positives often. This is good information to know when trying to differentiate a false alarm from an immanent threat.

Most everything in intrusion analysis involves correlation from multiple sources. For instance just because a signature triggers on port WXYZ and the alert states that there is a possible infection of Worm W32.A there is no clear indication that this is an actual infection. But if we correlate with scan logs and see that immediately after the connection on port WXYZ there is a sudden burst of scanning for ports with known vulnerabilities there is a higher likely hood of an infection being present.

Determining the severity of an event is another skill that is a result of information correlation. A perfect example of this was the detect of FTP globbing vulnerabilities in this report. There was an alert that has the likely hood of triggering on legitimate traffic. After some research on the internet we were able to determine that the vulnerability affected only a few versions of a certain ftp daemon. The internal hosts that were identified as targets were connected to in

[47] http://www.google.com
[48] http://isc.sans.org
[49] http://www.seifred.org
[50] http://www.whitehats.com/cgi/arachNIDS/
[51] http://www.giac.org/practical/

order to gather more information. The information that was gathered led us to determine that one host was not vulnerable but another was very vulnerable and was in fact quite a severe situation.

The method in use for intrusion detection is the most important step in the process of evaluating threats on a computer network. It is the opinion of the author that the best analysis is the combination of comprehensive research, vast correlation, and due diligence.

Appendix

References:

"The Addition of Explicit Congestion Notification (ECN) to IP"
URL: ftp://ftp.isi.edu/in-notes/rfc3168.txt (15 Jan 05).

CERT "CERT Advisory CA-2001-33 Multiple Vulnerabilities in WU-FTPD" 15 Feb 2002
URL: http://www.cert.org/advisories/CA-2001-33.html (16 Dec 04).

CERT "CERT Advisory CA-2001-26 Nimda Worm" 25 Sept 2001
URL: http://www.cert.org/advisories/CA-2001-26.html (23 Jan 05).

CERT. "CERT Advisory CA-2001-12 Superfluous Decoding Vulnerability in IIS"
URL: http://www.cert.org/advisories/CA-2001-12.html (13 Dec 05).

Claire, Wouter, "GIAC Certified Intrusion Analyst (GCIA) Practical Assignment" 5 Oct 2004
URL: http://www.giac.org/practical/GCIA/Wouter_Claire.pdf (13 Dec 04)

Cojocea, Mihai. "GIAC GCIA Practical Assignment" 22 Dec 2003
URL: http://www.giac.org/practical/GCIA/Mihai_Cojocea_GCIA.pdf (13 Dec 04)

CVE. "CVE-2001-0886"
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0886 (15 Dec 04)

CVE. "CVE-2001-0550"
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0550 (15 Dec 04)

F-Secure. "F-Secure Virus Descriptions : Agobot" 21 Oct 2004
URL: http://www.f-secure.com/v-descs/agobot.shtml (25 Jan 05).

Granier, Brian. "Intrusion Analysis and LaBrea Sentry" 21 Oct 2002.
URL: http://www.giac.org/practical/GCIA/Brian_Granier_GCIA.pdf (23 Jan 05).

Gregory, Donald. "SANS Intrusion Detection In Depth"
URL: http://www.giac.org/practical/GCIA/Donald_Gregory_GCIA.pdf (25 Jan 05).

Holstein, Michael. "GIAC Certified Intrusion Analyst (GCIA) Practical Assignment"
URL: http://www.giac.org/practical/Michael_Holstein_GCIA.doc (24 Jan 05).

Internet Security Systems "FTP Demi Glob"
URL: http://www.iss.net/security_center/advice/Intrusions/2001350/default.htm (16 Dec 04).

Jammed.com. "incidents 2001/07: TCP port 6346 " 30 Jul 2001.
URL: http://lists.jammed.com/incidents/2001/07/0310.html (12 Dec 04).

Lafferty, Rich. "ICQ Vulnerabilities"
URL: http://www.lafferty.ca/software/icq/icq.html (07 Jan 05).

Larratt, Glenn. "Intrusion Detection in Depth"
URL: http://is.rice.edu/~glratt/practical/Glenn_Larratt_GCIA.html (12 Dec 04).

Link Logger. "TCP Port 6129". 9 Feb 2004.
URL: http://www.linklogger.com/TCP6129.htm (12 Dec 04).

Link Logger. "TCP Port 2745". 9 Feb 2004.
URL: http://www.linklogger.com/TCP2745.htm (12 Dec 04).

Link Logger. "TCP Port 1025". 9 Feb 2004.
URL: http://www.linklogger.com/TCP1025.htm (12 Dec 04).

Link Logger. "TCP Port 3127". 9 Feb 2004.
URL: http://www.linklogger.com/TCP3127.htm (12 Dec 04).

LURHQ. "Phatbot Trojan Analysis – LURHQ" 15 March 2004.
URL: http://www.lurhq.com/phatbot.html (24 Jan 05).

Marty, Raffael. "The Big Barnyard".

URL: http://security.raffy.ch/projects/Raffael_Marty_GCIA.pdf (19 Dec 04).

McAfee. "W32/Polybot.I!irc" 14 March 03.

URL: http://vil.nai.com/vil/content/v_101100.htm (07 Jan 05).

Microsoft. "Microsoft Security Bulletin MS01-044" 13 July 2004

URL: http://www.microsoft.com/technet/security/bulletin/MS01-044.mspx (24 Jan 05).

SANS. "SANS Institute Free Webcast: Segmenting Networks: ACLs" 01 July 2004.

URL: https://www.sans.org/webcasts/show.php?webcastid=90512 (02 Feb 05).

SANS. "SANS - Internet Storm Center (Port 6129 Graph)"

URL: http://isc.sans.org/port_details.php?port=6129 (07 Jan 05).

Securiteam. "Securiteam: [UNIX] Wu-Ftpd File Globbing Heap Corruption Vulns" 30 Nov 2001.

URL: http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-11/0065.html (16 Dec 04).

SecurityFocus "Wu-Ftpd File Globbing Heap Corruption Vulnerability" 27 Nov 2001

URL: http://www.securityfocus.com/bid/3581/discussion/ (16 Dec 04).

SecurityFocus "Wu-Ftpd File Globbing Heap Corruption Vulnerability" 7 Feb 2001

URL: http://www.securityfocus.com/bid/3581/exploit/ (16 Dec 04).

Seifried, Kurt. "Port 515 TCP, UDP" 14 Jan 2003.

URL: http://www.seifried.org/security/ports/0/515.html (12 Dec 04).

Snort.org "Snort Signature Database"

URL: http://www.snort.org/snort-db/sid.html?id=1378 and http://www.snort.org/snort-db/sid.html?id=1377

Sternudd, Patrik. "Snort Overdrive". 25 Jul 2004

URL: http:// www.giac.org/practical/GCIA/Patrik_Sternudd_GCIA.pdf (24 Jan 05)

Storm, Pete. "GIAC Certified Intrusion Analyst (GCIA) Practical Assignment" 15 Oct 2004

URL: http://www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf " (12 Dec 04).

Symantec. W32.nimda.a@mm Security Response 30 Jul 2004.

URL: http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html (15 Jan 05).

Symantec. W32.hllw.lovgate.j@mm Security Response. 29 July 2004.

URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.lovgate.j@mm.html

US-CERT. "Vulnerability Note VU#111677" 18 Sept 2001

URL: http://www.kb.cert.org/vuls/id/111677 (13 Dec 05).

Van Oosterom, Peter. "GIAC Certified Intrusion Analyst (GCIA) Practical Assignment".

URL: http://www.giac.org/practical/GCIA/Peter_Van_Oosterom.pdf (12 Dec 04).

Vanhorenbeeck, Maarten. "GIAC Certified Intrusion Analyst (GCIA) Practical Assignment"

URL: http://www.giac.org/practical/GCIA/Maarten_Vanhorenbeeck_GCIA.pdf (07 Jan 05)

Williams, Al. "SANS GCIA Practical" Aug 2002.

URL: http://www.whitehats.ca/main/members/ Herc_Man/Files/Al_Williams_GCIAPractical.pdf (12 Dec 04).

White Hats. "IDS552 "IIS ISAPI OVERFLOW IDA"

URL: http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids552&view=signatures (23 Jan 05).

Appendix

Whois Information:

1. 139.165.206.128 - cerm22.chim.ulg.ac.be

```
[rudy@localhost ~]$ whois 139.165.206.128
[Querying whois.arin.net]
[Redirected to whois.ripe.net]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Whois query server #2.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html

inetnum:      139.165.0.0 - 139.165.255.255
netname:      UOFLIEGE-BE
descr:        Universite de Liege (ULg)
country:      BE
admin-c:      SU25-RIPE
tech-c:       SU25-RIPE
status:       ASSIGNED PI
remarks:      --------------------------------
remarks:      In case of abuse, please contact:
remarks:      abuse@ulg.ac.be
remarks:      --------------------------------
mnt-by:       BELNET-MNT
changed:      pirard@vm1.ulg.ac.be 19910327
changed:      piet@cwi.nl 19910404
changed:      Stephan.Biesbroeck@belnet.be 19930915
changed:      Marc.Roger@belnet.be 19980721
changed:      ad.hm@belnet.be 20031024
changed:      er-transfer@ripe.net 20040303
changed:      ad.hm@belnet.b 20040315
source:       RIPE

route:        139.165.0.0/16
descr:        UOFLIEGE-BE
origin:       AS2611
mnt-by:       BELNET-MNT
changed:      stephan@belnet.be 19950831
changed:      Eric.Luyten@belnet.be 19960419
changed:      Marc.Roger@belnet.be 19980721
source:       RIPE

role:         SEGI ULG
address:      Service General d'Informatique
address:      Universite de Liege
address:      B26 Sart Tilman
address:      B-4000 Liege
address:      Belgium
phone:        +32 4 3664904
fax-no:       +32 4 3662920
e-mail:       ripe@segi.ulg.ac.be
trouble:      call
admin-c:      FB7-RIPE
admin-c:      DK1178-RIPE
tech-c:       MF2348-RIPE
nic-hdl:      SU25-RIPE
mnt-by:       BELNET-MNT
changed:      ad.hm@belnet.be 20031024
source:       RIPE
```

Appendix

Whois Information:

2. 67.31.152.200 - dialup-67.31.152.200.Dial1.Denver1.Level3.net

OrgName: Level 3 Communications, Inc.
OrgID: LVLT
Address: 1025 Eldorado Blvd.
City: Broomfield
StateProv: CO
PostalCode: 80021
Country: US

NetRange: 67.24.0.0 - 67.31.255.255
CIDR: 67.24.0.0/13
NetName: LC-ORG-ARIN-BLK3
NetHandle: NET-67-24-0-0-1
Parent: NET-67-0-0-0-0
NetType: Direct Allocation
NameServer: NS1.LEVEL3.NET
NameServer: NS2.LEVEL3.NET
Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate: 2001-11-07
Updated: 2002-08-08

TechHandle: LC-ORG-ARIN
TechName: level Communications
TechPhone: +1-877-453-8353
TechEmail: ipaddressing@level3.com

OrgAbuseHandle: APL8-ARIN
OrgAbuseName: Abuse POC LVLT
OrgAbusePhone: +1-877-453-8353
OrgAbuseEmail: abuse@level3.com

OrgTechHandle: TPL1-ARIN
OrgTechName: Tech POC LVLT
OrgTechPhone: +1-877-453-8353
OrgTechEmail: ipaddressing@level3.com

OrgTechHandle: ARINC4-ARIN
OrgTechName: ARIN Contact
OrgTechPhone: +1-800-436-8489
OrgTechEmail: arin-contact@genuity.com