Global Information Assurance Certification Paper

GIAC Certified Intrusion Analyst (GCIA)
Practical Assignment
Version 3.4

Dana Webber
May 9 2004

Describe the State of Intrusion Detection: The Swen Enigma

Abstract

Email viruses are a major problem now. There is a lot of information available but it is mainly from companies that are selling anti-virus scanners. Here we look at how IDS and IPS can be configured for the Swen virus.

What is Swen?

Swen is a prime example of modern malicious software. According to F-Secure, it can spread by windows networking, email, Kazaa and IRC [1]. There are two email versions that will execute as soon as the email is viewed. You may not even need to open the attached file. And “best of all” everything works. This has made it very “popular” and millions of PC's got it. At its peak one out of 87 of the emails that MessageLabs filtered had it. [2]

Anybody who is technically competent enough to read this already knows that they need to have patches and virus definitions up to date. Therefore we will assume that a PC with Swen has a “non professional” administrator and probably is in a home, small business, or at an educational institution. However the professional IT community still has exposure. Virus email is a costly flood, the affected user may telephone support or a "friend" and sometimes the patches do not work. [3] I will argue that Swen probably has some "undocumented features" and that undetected fraud may be occurring.

What the user sees.

The test system had Windows 2000 SP4 with FAT32 and a non Internet IP of 192.168.43.254. It was connected to a dual homed PC running RH 8 Linux, Snort 4.05, Ethereal 0.9.16 and Sendmail. It was setup to accept any relay request, but to not actually send any email. [4]

Winstal LE was installed and a “snapshot” taken. Then all the Winstal LE files were moved to another system. The Swen "installation" went smoothly. There were several information screens that appeared to be from Microsoft. Afterwards the PC seemed to work as before. Then the system was booted in safe mode, the Winstal files returned and the “after snapshot” was done.

Next, Mcafee virus 4.5 scan was installed with no error messages. When IE was used to obtain the update from Mcafee.com, several pages said that this version was no longer supported. However, the default update method used FTP and showed no warnings.

After a reboot, every time any icon was opened the virus scanner warned about the Swen virus in the file qzzs.exe and then the application would not open. The IE desktop

icon was an exception. The virus file could not be deleted. When the system was rebooted in safe mode the same things happened.

1.3 What the user does not see

Fig 1 The Swen userdata key.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\YJTY]
"CacheBox Outfit"="yes"
"Counter Visited"="yes"
"Email Address"="swen_1@<host_name_removed>"
"Install Item"="ltvoyu"
"Installed"="... by Begbie"
"Server"="192.168.43.2"
"Unfile"="athzk.kee"
"VicName"="some name"
```

The registry key in Fig 1 shows some of the local system data Swen recorded. The line "Counter Visited"="yes" is worth a close look. When Swen was installed it did a DNS lookup for ww2.fce.vutbr.cz and then tried to make a TCP connection to <IP>:80 but the firewall prevented this. Swen had ignored the IE proxy setting. When the port was redirected with NAT to the proxy server, Swen sent the string seen in fig 2. It did not add the other lines that a browser would. At first, this “fast” programming style seems not to be consistent with the overall quality but the author could be testing for a system that has a direct connection to the Internet. The transparent proxy added extra lines to the HTTP transaction and Swen may have detected this.

Fig 2 The Counter visit

```
DNS query :  19:18:09.873764 192.168.43.254.1056 > 192.168.43.2.53:  [udp sum ok] 7+
A? ww2.fce.vutbr.cz. [|domain] (ttl 128, id 230, len 62)

http request :
GET http://ww2.fce.vutbr.cz/bin/counter.gif/link=bacillus&width=6&set=cnt006 HTTP/1.0\r\n
```

After the emails were sent, Swen tried to access numerous hosts at port 119 (NNTP, newsgroups). The firewall prevented any connection and after a few minutes Swen stopped. Swen appeared to detect when the Linux firewall was rebooted, it immediately tried to access the newsgroups again. The firewall was rebooted several more times and the Swen PC was rebooted as well. However no more NNTP attempts were detected.

1.5 Names and IP's

Up to the spring of 2003, the return path in most virus emails I received was correct. If not, then searching the mail directory for the IP's in the virus email header usually found a match. Swen was a big "advance" because these methods did not work anymore.

My observations are not consistent. When Swen was installed it sent emails to every entry in the Outlook address book and to addresses in received emails. This indicates that some of the Swen emails I have received should be from systems that have received email from me. Even if they got my address from a mailing list, then I should have other emails from some of them. I spent several days writing scripts to cross reference IP's and names between my legitimate email and the Swen email. I have kept every email over the last two years and can find no connection from any of them and the Swen emails. This is significant because when I could contact the owner of a PC that sent me a virus, they usually did not know they had a problem.

ISP mail servers.

Normally if there is a problem with a mail server, the admin should be notified. However every admin knows about email viruses and has decided to respond or ignore. A Google search on one host sending me Swen email, found that it was a commercial ISP. A legitimate user wrote, "212.123.84.81 is in a black list ... Now 212.123.84.81 is iron-c-1.tiscali.it and Tiscali is one of Italy's biggest ISPs, the one I use during the day " [5]. Their error logs must be huge! It would be a waste of time to report another Swen email to them.

There is a good reason why many ISP's choose to "ignore" this problem. The alternative could be costly. It takes more work to setup mail servers that filter email. To keep up with the rate viruses are evolving requires daily maintenance. A big problem, especially in the US, is that they can be sued if they miss an important update. The relevant law here includes "industry standards" and "due diligence". It means that you need to provide the same level of security as everybody else does. This can get very complicated and employ many lawyers.

1.4 What is the real purpose of Swen?

The author is very knowledgeable and has obviously spent much time on this. He may be part of the professional “white hat” community. This reminds me of discussions with other IT professionals about how poorly virus code is written and how “to do it better”.

Most Kazaa users know they are not “totally legal” and ISP`s want nothing to do with it. We know that Swen has code to deal with Kazaa. Who would notice or even look for a few unusual port 1214 datagrams? A PC with direct access to the Internet and a valid email account that also runs Kazaa would be ideal to hide a RAT in.

How can we be sure that the author subscribes to the same morals standards as Kevin Mitnick?[6] He has a way to get several thousand Quickbooks data files. Identity theft would be just the start. When the theft was discovered, the police may find clues that had been planted to setup another innocent person in another country. In many places this would be an automatic conviction and a long sentence. The convicts' only real crime would be to not have kept the virus definitions up to date.

The international aspect is a big problem. All the Swen email I have received in the last few months was from non Canadian IP's. This may mean that CSIS, not the RCMP, should handle it.

Swen detects and terminates if it is run in a debugger [7] and it sets a registry key if it has direct HTTP access to the Internet. It is likely that it does more tests. It was observed sending mail and doing a lot of NNTP activity also the string Kazaa occurs five times the Swen executable. Swen may be waiting for a coded response before doing anything more. The author could “research” the system before answering. If he is aware of the public Snort logs at the University of Minnesota he may choose not respond to any systems there.

It is quite possible that the author is attempting to fool the virus researchers so observations of how Swen behaves “in the wild” are necessary.

1.5 Email Virus Rules for Snort and Snort-Inline.

The file virus.rules included with Snort contains the comment "We don't care about virus rules anymore". They apparently believe that email viruses are better handled by the mail server. However that happens after the email has been received. Snort and Snort-inline can react before the SMTP connection is over.

Here is the existing Snort alert for ".exe" files.

Fig 5 The entry for ".exe" in Snort virus.rules

```
alert tcp $SMTP_SERVERS any -> $EXTERNAL_NET 25 (
msg:"VIRUS OUTBOUND .exe file attachment"; flow:to_server,established;
content:"Content-Disposition|3a|"; content:"filename=|22|"; distance:0;within:30;
content:".exe|22|"; distance:0; within:30; nocase;
classtype:suspicious-filename-detect; sid:2160; rev:1;
)
```

This is an example of the section of an email that will trigger this alert.

```
------=_NextPart_000_0016_01C3FB0D.6EE69E20\r\n
Content-Type: application/x-msdownload;\r\n
        name="2nd_file.exe"\r\n
Content-Transfer-Encoding: base64\r\n
Content-Disposition: attachment;\r\n
        filename="2nd_file.exe"\r\n
\r\n
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

However if the TCP packet boundary cuts the word filename then Snort-inline will not alert. Snort can reassemble the two packets but Snort-inline cannot, everything has to be in the same packet.

```
------=_NextPart_000_0016_01C3FB0D.6EE69E20\r\n
Content-Type: application/x-msdownload;\r\n
        name="2nd_file.exe"\r\n
Content-Transfer-Encoding: base64\r\n
Content-Disposition: attachment;\r\n
        fil
ename="2nd_file.exe"\r\n
\r\n
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Swen will not trigger an alert because it does not have the string "filename"

```
--sxbxsqxn\r\n
Content-Type: application/x-msdownload; name="Install99.exe"\r\n
Content-Transfer-Encoding: base64\r\n
Content-Disposition: attachment\r\n
\r\n
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Proposed general email virus rules.
Swen would alert on "Content-Type: " and "name=". They can be on the same text line.

```
alert tcp any any -> any 25 (
msg:".exe file attachment 3 "; flow:to_server,established;
content:"|0a|Content-Type|3a| "; nocase;
content:" name=|22|";  nocase; distance:0; within:60;
content: ".exe|22|"; distance:0; within:30; nocase; classtype:suspicious-filename-
detect;
)
```

Or, name can be on the next line and be preceded with a tab.

```
alert tcp any any -> any 25 (
msg:".exe file attachment 2 "; flow:to_server,established;
content:"|0a|Content-Type|3a| "; nocase;
content:"|0a||09|name=|22|";  nocase; distance:0; within:60;
content: ".exe|22|"; distance: 0; within:50; nocase; classtype:suspicious-filename-
detect4170;
)
```

Also, it is possible that they are in different packets.

```
grep ".exe"  proposed.virus.rules
alert tcp any any -> any 25 (
msg:".exe file attachment 1 "; flow:to_server,established;
content:"|0a||09|name=|22|";  nocase; distance:0; within:60;
content: ".exe|22|"; distance:0; within:30; nocase; classtype:suspicious-filename-
detect;
)
```

A search of my virus mailbox indicated that other email viruses would alert as well.

```
grep -RiA1 "^Content-Disposition: " ~/Mail/virus/cur|grep -i "filename="|sed
's/^.*[Ff]ilename=.*\(\....\)"\?\W*$/\1/'|sort|uniq -c|sort -nr| grep -i exe
41 .exe
```

```
grep -RiA1 "^Content-Type: " /1/user1/Mail/virus/cur|grep "name=.*\....\""|sed
's/^.*\Wname="\?.*\(\....\)\"\W*$/\1/'|sort|uniq -c|sort -nr| grep -i exe
395 .exe
```

Lets try the new rules on tcpdump_w_jan_17_2004_a.libcap.
Default snort rules found nothing, but the proposed rules alerted 852 times

```
grep virus snort.conf
include $RULE_PATH/virus.rules
```

```
grep -c "^\[\*" snort/logs/alert
0
```

```
grep virus snort.conf
include $RULE_PATH/proposed.virus.rules
```

```
grep  "^\[\*" logs/alert | sort| uniq -c
      1 [**] [1:0:0] .bat file attachment 3  [**]
    851 [**] [1:0:0] .exe file attachment 3  [**]
```

There were 420 attachments named bsggjjmd.exe and 424 named crmoropt.exe/

```
grep -iR "[^ ]\.ex$\|[^ ]\.exe" snort/logs/2* | cut -d\  -f2-|sort|uniq -c|sort -nr|cut -c1-8,56-
   424   q388674.exe"..Co
   420   ame="bsggjjmd.ex
     6   7.exe"
     1   e="crmoropt.exe"
```

First. Swen makes a HTTP query to ww2.fce.vutbr.cz. IE proxy settings are ignored.

```
#Snort Rule Swen Install
alert tcp any any -> any 80  ( msg: "Swen Virus Just Installed"; uricontent:
"http\://ww2.fce.vutbr.cz/bin/counter.gif/link=bacillus&width=6&set=cnt006";
nocase; classtype: trojan-activity ; )
```

Consider an alert on the first line of the Swen executable. There are 787 matches in my virus directory

```
grep -R ^TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA"  ~/Mail_virus_cur* | grep -c ""
787
```

However some other viruses have four less characters on the first line..

```
grep -R "^TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA"  ~/Mail_virus_cur* | grep -c ""
792
```

The proposed alert uses the shorter line so it will alert on other viruses as well.

```
#Snort Rule Swen in email
alert tcp any any -> any 25 (  msg: "Swen Virus in email";  content:
"|0d||0a||0d||0a|TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA"; nocase; classtype:suspicious-filename-detect; )
```

Here is a breakdown of exe files in my virus directory that had the shorter line.

```
grep -RB6 \
"^TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA$"  ~/Mail_virus_cur* | grep -i name

/home/user1/Mail_virus_cur/1048159349.3785.lBwi:2,S-     name=setup.exe
/home/user1/Mail_virus_cur/1048465815.30423.f1wA:2,S-    name=README.scr
/home/user1/Mail_virus_cur/1048555332.18719.5aYF:2,S-    name=PM,49, .scr
/home/user1/Mail_virus_cur/1048619262.23809.wtUU:2,S-    name=align.pif
/home/user1/Mail_virus_cur/1048818753.10370.LZ87:2,S-    name=href.scr
```

Lets try the new rules on the capture file from detect three.

Default snort virus rules found no dangerous attachments	ls -sh tcpdump_w_jan_17_2004_a.libcap 56M tcpdump_w_jan_17_2004_a.libcap
All default Snort virus rules and all proposed rules except Swen rules	1 [**] [1:0:0] .bat file attachment 3 [**] 851 [**] [1:0:0] .exe file attachment 3 [**]
All default Snort virus rules and all proposed virus rules	6 [**] [1:0:0] .exe file attachment 3 [**] 858 [**] [1:0:0] Swen Virus in email [**]

In a recent GCIH practical, Stephan Reid used the following alerts. [8] The first alert alerts on a different line in the attachment. The second alert catches the version that shows garbage or will "automatically install" if viewed with an old unpatched IE.

```
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.Swen mm -
SMB";content:"|59 59 85 C0 74 09 6A 01 58 83 4D FC FF EB 15 FF 85
E0 FE FF FF EB C7 6A 01 58 C3 8B 65 E8 83 4D|"; classtype:misc-activity;rev:1;)
```

```
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.Swen mm -
MIME";content:"QABohKNAAGShAAAAAFBkiSUAAAAAgewUAQAAU1ZXiWXoM/+Jff
yJvdz +//+LdQhW6NORAABZhcB0"; classtype:misc-activity;rev:1;
```

1.5 Snort_Inline

If an internal computer has a virus then action should be taken. If feasible, then automatically close the Ethernet port. Other measures include blocking the IP at the firewall. In an environment like a University that has computer systems that are run by volunteers or students, it would be prudent to filter outgoing SMTP traffic. It would be embarrassing to get a mail server blacklisted and Snort_Inline could reduce this risk.

Snort-Inline[9] can drop a packet that triggers an alert. The text cannot be split across two packets. To it is better to have more then one alert.

CounterSnipe [10] says Snort_Inline can reduce the load on the mail server. The testing done here does not agree but it does indicate that it could be another layer of defense.

For the external test, Snort-inline was setup to reject any SMTP packet that contained the first base-64 encoded line of either version of the Swen executable. This meant that it dropped the packet and sent a reject packet to the other system. It could not be setup to send a reject to the target host so the port on our mail server had to time out. This left the socket open longer. The external mail servers just kept trying again and again until they timed out. This usually took five days.

Internal test: The first packet Swen sent was a query for ww2.fce.vutbr.cz to the default DNS. It was dropped and the initial http: query did not happen. Next Swen tried to send email to Outlook's default SMTP server but Snort_Inline always dropped a packet from the email data without sending a reset back. Ten hours later Swen was still trying to send the first email and had not tried to access any NNTP servers or send any other emails. If Snort had sent a warning to IT, Then the infected computer could have been fixed before the virus had a chance to spread.

The internal web servers also could be protected this way. This brings up a new concept; the IPS is not at the Internet connection, it is inside the intranet. The first issue here is can it handle all the traffic. CounterSnipe claims 1 Gig throughput for a Xeon based system but that sounds a little optimistic. Checkpoint has a solution they call Interspect. They point out that there are many more protocols to deal with. It divides the intranet into zones using VLan. What if someone writes a virus that can jump VLans?

1.6 How Swen could be “improved”

It let Mcaffee Antivirus 4.5 be installed and the latest definition files downloaded. When the system was scanned the virus type the virus filenames were identified but could not be deleted. This means that the user would know they needed to seek help. If the scanner had been prevented from detecting the virus, the user would have assumed everything was OK. Most servers do not send back warnings now so Swen could stay hidden for years.

Since Swen was released there have been several "advances." The "niftiest" one is to encode the file so you need a password to open it. The password may be sent as a image that a person could easily read but a computer could not. Other viruses leave a trojan in place and there are reports that lists of these systems are for sale on the internet. [11]

W2k SP4 showed a warning that the file did not have a Microsoft digital signature. That is theoretically imposable to forge but the virus writers have consistently outsmarted everybody.

References

[1] F-Secure Corporation. F-Secure Computer Virus Information Pages: Gibe.B.
URL: http://www.europe.f-secure.com/v-descs/gibe_b.shtml (17 March 2004).

[2] MessageLabs.
MessageLabs Threats And Analysis: Virus Info - Virus Details - W32/Swen.A-mm.
URL: http://messagelabs.com/viruseye/info/default.asp?
%20%20%20frompage=top+ten&fromURL=%2Fviruseye%2Fthreats%2F&virusname=
W32%2FSwen%2EA%2Dmm (17 March 2004).

[3] Barnum, Guy. Bugtraq: Outlook security updates not stopping Swen. 24 Sep 2003.
URL: http://seclists.org/lists/bugtraq/2003/Sep/0403.html (17 March 2004).

[4] Spencer, Brad. Fighting Relay Spam the Honeypot Way. 13 Jun 2002. URL:
http://fightrelayspam.homestead.com/files/antispam06132002.htm (17 March 2004).

[5] Ayers, Danny. Raw: Italian Technology Infrastructure. 19 December 2003
URL: http://dannyayers.com/archives/002118.html (17 March 2004).

[6] Salkever, Alex. Should You Trust a Reformed Hacker?
BW Online | August 8, 2000 | Should You Trust a Reformed Hacker?
8 AUGUST 2000. URL:
www.businessweek.com/careers/content/aug2000/ca2000088_738.htm (17 Mar 2004).

[7] FRISK Software International. F-Prot Antivirus for Windows, Linux, BSD, Exchange,
AIX, Solaris and DOS | F-Prot AVES - anti-spam and anti-virus e-mail filtering service.
URL: http://www.f-prot.com/virusinfo/descriptions/swena.html (17 March 2004).

[8] Reid, Stephan. Swen: The Worm with Social Engineering Aspirations 16 Dec 2003.
http://www.giac.org/practical/GCIH/Reid_Stephan_GCIH.pdf (4 May 2004).

[9] SourceForge.net: Project Info - snort_inline.
URL: http://sourceforge.net/projects/snort-inline/ (17 March 2004).

[10] Countersnipe Technologies. Case Study W32/Swen.A.
23 Sep 2003. URL: http://countersnipe.com
/downloads/case_studies/APD_Perimeter_Virus_case_study_030923.pdf (17 March
2004).

[11] Leyden, John. The illicit trade in compromised PCs 30 April 2004.
http://www.theregister.co.uk/2004/04/30/spam_biz/ (4 May 2004).

GIAC GCIA Version 3.4 Practical Detect: Possible VMware penetration test.

Source of Trace: http://www.incidents.org/logs/2003.12.15.tgz

Detect was generated by:

Detect File Overview	
Archive URL	http://www.incidents.org/logs/2003.12.15.tgz
files in archive	2003.12.15,.1 2003.12.15.2, ..., 2003.12.15.14
File Format	libpcap, Snaplen = 96 bytes
Start time	11/18/2003 13:57:23.130647
End time	11/18/2003 15:15:57.147884
Duration	1 hour and 18 minutes.
Packets captured	449147
total Alerts	31791
distinct alerts	66
Source MAC's	30
Destination MAC's	39
Alerts to 0:50:56:40:0:6d	28633

MAC address obfuscation is normally not done and IP address obfuscation was not necessary because all the detected IP's were reserved for private networks

None of the IP's are legal on the Internet. They were all in reserved ranges.		
Command: whois $ip @whois.arin.net;		
NetRange:	10.0.0.0 - 10.255.255.255	This block is reserved for special purposes.
NetRange:	192.168.0.0 - 192.168.255.255	This block is reserved for special purposes.
NetRange:	172.16.0.0 - 172.31.255.255	This block is reserved for special purposes.
NetRange:	224.0.0.0 - 239.255.255.255	This block is reserved for special purposes.

It appears that every packet was captured in libpcap format with snaplen 96. It was assumed that all traffic at the target's LAN connection was captured. This may not be correct. Most of the systems were not analyzed. It is assumed they were monitoring the test. Eight consecutive Cisco MAC's were detected but none were associated with any IP's. They were transmitting data link level spanning tree data. This could indicate a large router that was not allowed to transmit IP traffic. It may have been used for the data capture.

Probability the source address was spoofed: 50%

The places I have worked at would buy a Smartbits [1] and connect it with one Ethernet cable to the unit to be tested and then "run a test". The Smartbits would generate the traffic and capture the data by itself. Other companies make similar devices.

Program that was used to generate the alert file.:

```
#!/bin/sh
#A problem with Snort-2.0.5 is that if a packet triggers two alerts then the second
#alert takes precedence. This produced ICMP alerts with "(Undefined Code!)".
#The fix:move alerts with "(Undefined Code!)" in file icmp-info.rules to the top.

  cd /1/backup/giac/gcia_assignment
  COUNTER='1';
  while [ "$COUNTER" != 15 ]; do
    RAWPATH="incidents.org.logs.raw.2003.12.15/2003.12.15.$COUNTER";
    echo $RAWPATH
    ls -l $RAWPATH
    SNORTPATH="snort/snort-2.0.5-bin";
    LOGPATH="p2_detect_1/logs";
    CONFPATH="snort/snort.conf";
    COMMAND="$SNORTPATH  -X -d -e -c $CONFPATH -r $RAWPATH -l
$LOGPATH"
    echo "$COMMAND";
   `$COMMAND`
    COUNTER=$[$COUNTER + 1];
  done;

#command: snort -X -d -e -c $SNORT_CONF -r $RAW_FILE -l $LOG_FILE
#-X  means dump the raw packet data starting at the link layer
#-d  means dump the Application Layer
#-e  means display the second layer header info
#$RAW_FILE is varied from 2003.12.15.1 to 2003.12.15.14
```

2.1.3. Description of attack:

One out of every 27 packets triggered an alert. Most MAC's had only one or two IP's associated with them. Usually a LAN has one or more routers and each router has a MAC associated with many IP's. There were 18 distinct source IP's and 1542 distinct destination IP's detected for MAC 0:50:56:40:0:6d . However, ninety percent of the alerts have 0:50:56:40:0:6d as the destination MAC and the three top alert source MAC's were registered to Apple Computer, Intel and Dell. Furthermore the MAC 0:50:56:xx.xx.xx is registered to VMware INC which does not make routers. VMware enables multiple virtual machines on one physical system. Each virtual machine needs a distinct IP. This may be the reason that this MAC had 18 IP's. Normally most of the attacks all have the same source MAC because they come from the Internet and go through a router/gateway/firewall. A University campus link would likely have attacks in both directions. The only reasonable explanation here is a penetration test of a VMware system and that all the attacks are part of the same test. Such a test set-up should be isolated therefore an Internet gateway is not required.

Manufacturers of equipment on the LAN.

Ando makes expensive test equipment. [2]
Ambit makes expensive test equipment.[3]
Abocom is a OEM manufacturer [4].
VMware, the MAC was probably programmed in specifically for testing.[5]
Compaq, makes a variety of PC products.
Gateway, makes a variety of PC products.
Dell, makes a variety of PC products.
3Com, makes NIC's and Ethernet switches
Intel makes a variety of PC products.
Connectix made some video conferencing equipment, Microsoft bought them.[6]
Sony makes products in every category.[7]
DOD MAC's are sometimes used by companies that do have not registered a OUI.[8]

Commands used to generate OUI Registrant [9], MAC, and IP list

Command used to obtain the list of source MAC addresses.

```
 j=1;while [ "$j" != 15 ];do /usr/sbin/tcpdump -ennr
../incidents.org.logs.raw.2003.12.15/2003.12.15.$j \
 |cut -d\  -f 2|grep ':'|sort|uniq;j=$[$j+1];done|sort|uniq
```

Command used to obtain the list of destination MAC addresses.

```
  j=1;while [ "$j" != 15 ];do /usr/sbin/tcpdump -ennr
../incidents.org.logs.raw.2003.12.15/2003.12.15.$j |\
cut -d\  -f 3|grep ':'|sort|uniq;j=$[$j+1];done|sort|uniq
```

Commandsused to obtain the list of source MAC and IP pairs.

```
j=1;while [ "$j" != 15 ];do /usr/sbin/tcpdump -nne ip - \
incidents.org.logs.raw.2003.12.15/2003.12.15.$j >> tcpdump.eth.5.txt;j=$[$j+1];done
cut -d\  -f2,6 tcpdump.eth.5.txt|sed  's/^\(.* .*\..*\..*\..*\)\..*$/\1/' |sort|uniq
```

Commands used to obtain the list of destination MAC and IP pairs.

```
j=1;while [ "$j" != 15 ];do /usr/sbin/tcpdump -nne ip -r\
../incidents.org.logs.raw.2003.12.15/2003.12.15.$j >>
tcpdump.eth.5.txt;j=$[$j+1];done;
cut -d\  -f3,8 tcpdump.eth.5.txt|sed  's/^\(.* .*\..*\..*\..*\)\..*$/\1/' |sort|uniq
```

OUI Registrant [8], MAC, and IP list	
Point Multimedia Systems 0:0:39:f2:67:88	 10.10.10.117
Ando Electric Corporation 0:0:e2:92:ee:f 0:0:e2:94:b0:2a	 10.10.10.222 10.10.10.226
3COM CORPORATION 0:1:2:79:91:ed 0:1:3:88:29:92 0:4:76:45:61:39	 10.10.10.112 10.10.10.234 10.10.10.195
Compaq Computer Co 0:2:a5:b6:e2:e3	 10.10.10.186
Intel Corporation 0:a0:c9:ba:6d:85 0:3:47:8c:89:c2	 10.10.10.196 10.10.10.165 192.168.117.1 192.168.213.1
Connectix 0:3:ff:df:95:84	 10.10.10.228
VMWare, Inc. 0:50:56:40:0:64 0:c:29:14:1e:63 0:c:29:39:6e:67 0:c:29:9e:ef:53 0:50:56:40:0:6d	 10.10.10.2 10.10.10.142 10.10.10.160 10.10.10.224 10.10.10.1, 10.30.30.2, 172.20.11, 172.20.11.2, 172.20.11.3, 172.20.11.52, 172.20.11.80, 172.20.201.1 172.20.201.135, 172.20.201.198, 172.20.201.2, 192.168.17.129, 192.168.17.135, 192.168.17.2, 192.168.17.65, 192.168.17.66, 192.168.17.68, 192.168.22.207
Dell Computer Corp. 0:6:5b:d8:bf:ed 0:6:5b:e6:f8:43 0:8:74:5:b7:f8 0:8:74:7:31:ee	 10.10.10.122 10.10.10.231 10.10.10.147 10.10.10.111, 172.16.8.189
Dell ESG PCBA Test 0:b:db:17:f4:c9 0:b:db:9b:46:fe 0:b:db:df:53:8d	 10.10.10.194, 169.254.135.50, 172.16.9.13, 192.168.222.1 192.168.84.1 10.10.10.164 10.10.10.123
IBM Corporation 0:9:6b:2:e9:3d	 10.10.10.212, 172.16.8.229
Apple Computer, Inc. 0:a:95:7c:24:0 0:a:95:d9:95:84	 10.10.10.113 10.10.10.232

OUI Registrant [8], MAC, and IP list
AMBIT MICROSYSTEMS CORP 0:d0:59:c6:5e:14 10.10.10.141 10.10.10.144 238.122.10.140
Cisco Systems 0:d:bc:17:4:ce N/A 0:d:bc:17:4:cf N/A 0:d:bc:17:4:d0 N/A 0:d:bc:17:4:d2 N/A 0:d:bc:17:4:d4 N/A 0:d:bc:17:4:d5 N/A 0:d:bc:17:4:d6 N/A 0:d:bc:17:4:d8 N/A
AboCom Systems, Inc 0:e0:98:a1:7f:da 10.10.10.174
GATEWAY 2000 0:e0:b8:3d:20:40 10.10.10.214
SONY CORPORATION LTD. 8:0:46:79:f7:7c 10.10.10.230
DoD Internet Multicast (RFC-1112) 1:0:5e:0:0:16 224.0.0.22 DoD Internet Multicast (RFC-1112) 1:0:5e:0:0:2 224.0.0.2 DoD Internet Multicast (RFC-1112) 1:0:5e:0:0:5 224.0.0.5 DoD Internet Multicast (RFC-1112) 1:0:5e:0:0:6 224.0.0.6 DoD Internet Multicast (RFC-1112) 1:0:5e:37:96:d0 229.55.150.208 DoD Internet Multicast (RFC-1112) 1:0:5e:7a:a:8c 238.122.10.140 DoD Internet Multicast (RFC-1112) 1:0:5e:7f:ff:fa 239.255.255.250 DoD Internet Multicast (RFC-1112) 1:0:5e:7f:ff:fd 239.255.255.253 DoD Internet Multicast (RFC-1112)
Cisco Inter-Switch Link 1:0:c:0:0:0
Cisco Discovery Protocol 1:0:c:cc:cc:cc
Spanning tree Multicast 1:80:c2:0:0:0

There were many suspicious alerts. They are sorted in order of occurrences.
command: grep "\[**\] .* \[**\]" alert | sort | uniq -c | sort -nr

```
18176 [**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**]
 5041 [**] [1:465:1] ICMP ISS Pinger [**]
 2144 [**] [1:620:3] SCAN Proxy (8080) attempt [**]
 1326 [**] [1:402:4] ICMP Destination Unreachable (Port Unreachable) [**]
  181 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length
 1020 [**] [1:474:1] ICMP superscan echo [**]
  690 [**] [1:399:4] ICMP Destination Unreachable (Host Unreachable) [**]
  418 [**] [1:401:4] ICMP Destination Unreachable (Network Unreachable) [**]
  394 [**] [1:615:4] SCAN SOCKS Proxy attempt [**]
  287 [**] [1:1417:2] SNMP request udp [**]
  146 [**] [1:1420:2] SNMP trap tcp [**]
  140 [**] [1:1418:2] SNMP request tcp [**]
  116 [**] [1:1421:2] SNMP AgentX/tcp request [**]
  105 [**] [1:618:4] SCAN Squid Proxy attempt [**]
   70 [**] [1:1443:2] TFTP GET passwd [**]
   60 [**] [1:1411:3] SNMP public access udp [**]
   59 [**] [1:467:1] ICMP Nemesis v1.1 Echo [**]
   46 [**] [1:1419:2] SNMP trap udp [**]
   45 [**] [1:408:4] ICMP Echo Reply [**]
   29 [**] [1:553:4] POLICY FTP anonymous login attempt [**]
   28 [**] [1:365:5] ICMP PING (Undefined Code!) [**]
   27 [**] [1:628:2] SCAN nmap TCP [**]
   24 [**] [111:10:1] (spp_stream4) STEALTH ACTIVITY (XMAS scan) detection [**]
   21 [**] [111:12:1] (spp_stream4) NMAP FINGERPRINT (stateful) detection [**]
   18 [**] [1:449:4] ICMP Time-To-Live Exceeded in Transit [**]
   14 [**] [1:524:6] BAD-TRAFFIC tcp port 0 traffic [**]
   14 [**] [1:361:7] FTP site exec [**]
   12 [**] [1:2049:1] MS-SQL ping attempt [**]
   10 [**] [1:491:6] INFO FTP Bad login [**]
   10 [**] [1:1413:2] SNMP private access udp [**]
    8 [**] [1:1777:2] FTP EXPLOIT STAT * dos attempt [**]
    6 [**] [1:566:3] POLICY PCAnywhere server response [**]
    6 [**] [1:528:4] BAD-TRAFFIC loopback traffic [**]
    6 [**] [1:1957:3] RPC sadmind UDP PING [**]
    6 [**] [1:1893:1] SNMP missing community string attempt [**]
    6 [**] [1:1867:1] MISC xdmcp info query [**]
    6 [**] [1:1504:5] MISC AFS access [**]
    6 [**] [105:1:1] spp_bo: Back Orifice Traffic detected (key: 31337) [**]
    4 [**] [1:453:4] ICMP Timestamp Request [**]
    4 [**] [1:451:4] ICMP Timestamp Reply [**]
    4 [**] [1:388:4] ICMP Address Mask Request [**]
    4 [**] [1:382:4] ICMP PING Windows [**]
    4 [**] [1:336:5] FTP CWD ~root attempt [**]
```

There were many suspicious alerts. They are sorted in order of occurrences.
command: grep "\[**\] .* \[**\]" alert | sort | uniq -c | sort -nr

```
   4 [**] [1:237:1] DDOS Trin00:MastertoDaemon(defaultpassdetected!) [**]
   4 [**] [1:1992:1] FTP LIST directory traversal attempt [**]
   4 [**] [1:1449:3] POLICY FTP anonymous (ftp) login attempt [**]
   3 [**] [1:604:5] RSERVICES rsh froot [**]
   3 [**] [1:501:2] MISC source route lssre [**]
   3 [**] [1:1432:4] P2P GNUTella GET [**]
   2 [**] [1:718:6] TELNET login incorrect [**]
   2 [**] [1:659:4] SMTP expn decode [**]
   2 [**] [1:356:5] FTP passwd retrieval attempt [**]
   2 [**] [1:332:5] FINGER 0 query [**]
   2 [**] [1:330:6] FINGER redirection attempt [**]
   2 [**] [1:327:5] FINGER remote command pipe execution attempt [**]
   2 [**] [1:326:5] FINGER remote command ; execution attempt [**]
   2 [**] [1:323:4] FINGER root query [**]
   2 [**] [1:1728:2] FTP CWD ~<CR><NEWLINE> attempt [**]
   1 [**] [1:489:5] INFO FTP No Password [**]
   1 [**] [1:404:4] ICMP Destination Unreachable (Protocol Unreachable) [**]
   1 [**] [1:335:4] FTP .rhosts [**]
   1 [**] [1:255:8] DNS zone transfer TCP [**]
   1 [**] [1:245:1] DDOS mstream handler ping to agent [**]
   1 [**] [1:239:1] DDOS shaft handler to agent [**]
   1 [**] [1:236:3] DDOS Stacheldraht client check gag [**]
   1 [**] [1:1928:3] FTP shadow retrieval attempt [**]
   1 [**] [1:1919:3] FTP CWD overflow attempt [**]
   1 [**] [1:1444:2] TFTP Get [**]
```

Sorting by source IP shows the top two alert sources. Otherwise it is confusing list.
command: egrep "IpLen" alert|cut -d \ -f 1|cut -d ":" -f 1|sort|uniq -c|sort -nr

```
18202 10.10.10.113 Apple Computer, Inc
 9211 10.10.10.165 Intel Corporation
  912 10.10.10.164 Dell ESG PCBA Test
  711 172.20.201.2 VMWare, Inc
  669 10.10.10.231 Dell Computer Corp
  650 10.10.10.224 VMWare, Inc.
  573 10.30.30.2   VMWare, Inc
  433 10.10.10.234 3COM CORPORATION
  425 10.10.10.1   VMWare, Inc
  390 10.10.10.141 AMBIT MICROSYSTEMS CORP
  378 10.10.10.2   VMWare, Inc.
  236 10.10.10.195 3COM CORPORATION
  230 172.20.201.198  VMWare, Inc.
  198 172.20.201.135  VMWare, Inc.
  196 172.20.201.1 VMWare, Inc.
```

Sorting by source IP shows the top two alert sources. Otherwise it is confusing list.
command: egrep "IpLen" alert|cut -d \ -f 1|cut -d ":" -f 1|sort|uniq -c|sort -nr

```
 140 10.10.10.212 IBM Corporation
 132 10.10.10.112 3COM CORPORATION
 129 192.168.17.2 VMWare, Inc.
  80 10.10.10.194 Dell ESG PCBA Test
  57 172.20.11.3  VMWare, Inc
  46 10.10.10.186 Compaq Computer Corporation
  41 10.10.10.174 AboCom Systems, Inc
  34 10.10.10.196 Intel Corporation
  23 10.10.10.228 Connectix
  20 10.10.10.232 Apple Computer, Inc.
  19 172.20.11.2  VMWare, Inc.
  15 0.0.0.0      <Used For Bootp>
  12 10.10.10.226 Ando Electric Corporation
  11 10.10.10.230 SONY CORPORATION LTD.
   9 10.10.10.222 Ando Electric Corporatio
   7 10.10.10.160 VMWare, Inc.
   7 10.10.10.142 VMWare, Inc.
   7 10.10.10.122 Ando Electric Corporation
   6 169.254.135.50 Dell ESG PCBA Test
   4 192.168.17.68  VMWare, Inc.
   3 238.122.10.140 AMBIT MICROSYSTEMS CORP
   2 172.20.11.80  VMWare, Inc.
   2 172.20.11.52  VMWare, Inc.
   2 10.10.10.214 GATEWAY 2000
   1 172.20.11.1  VMWare, Inc.
   1 10.10.10.147 Dell Computer Corp.
   1 10.10.10.144 AMBIT MICROSYSTEMS CORP
   1 10.10.10.111 Dell Computer Corp.
```

Sort by destination IP: VMWare was the target. Here are the top 42 destination.
command: grep "IpLen" alert|cut -d \ -f 3|cut -d ":" -f 1|sort|uniq -c|sort -nr|head -n 42

```
6650 192.168.17.129  VMWare, Inc
6546 192.168.17.68   VMWare, Inc
5321 192.168.17.135  VMWare, Inc
1375 10.10.10.165    Intel Corporation
 994 172.20.201.2    VMWare, Inc
 876 172.20.201.198  VMWare, Inc
 795 172.20.201.135  VMWare, Inc
 780 172.20.201.1    VMWare, Inc
 581 10.10.10.224    VMWare, Inc
 471 172.20.201.3    VMWare, Inc
 407 172.20.11.2     VMWare, Inc
 351 10.10.10.164    Dell ESG PCBA Tes
 293 192.168.17.66   VMWare, Inc
```

Sort by destination IP: VMWare was the target. Here are the top 42 destination.
command: grep "IpLen" alert|cut -d \ -f 3|cut -d ":" -f 1|sort|uniq -c|sort -nr|head -n 42

```
238 192.168.17.67   VMWare, Inc
235 149.134.30.62   VMWare, Inc
189 149.134.52.149  VMWare, Inc
176 172.22.201.1    VMWare, Inc
156 172.20.11.80    VMWare, Inc.
122 172.22.201.2    VMWare, Inc
120 192.168.22.207  VMWare, Inc.
120 192.168.17.1    Intel Corporation
113 172.20.11.3     VMWare, Inc.
111 10.10.10.255    BROADCAST ADDRESS
105 10.10.10.212    IBM Corporation
 91 10.10.10.195    3COM CORPORATION
 89 10.10.10.226    Ando Electric Corporation
 80 10.10.10.231    Dell Computer Corp.
 64 172.22.201.3    VMWare, Inc
 56 255.255.255.255 BROADCAST ADDRESS
 51 172.20.11.52    VMWare, Inc
 39 10.10.10.142    VMWare, Inc
 37 10.10.10.222    Ando Electric Corporation
 36 172.11.11.80    VMWare, Inc
 36 10.10.10.112    3COM CORPORATION
 34 10.10.10.2      VMWare, Inc
 24 229.55.150.208  DoD Internet Multicast (RFC-1112)
 24 172.20.201.0    INVALID
 24 10.10.10.234    3COM CORPORATION
 23 192.168.17.65   VMWare, Inc.
 22 10.10.10.141    AMBIT MICROSYSTEMS CORP
 21 172.10.11.80    VMWare, Inc
 20 10.10.10.122    Dell Computer Corp
```

Sorting by destination MAC shows that the attacks targeted one particular system.
command: grep " -> " alert|grep -v IpLen |cut -d \ -f 4|sort|uniq -c|sort -nr

```
28633 0:50:56:40:0:6D   VMWare, Inc
 1375 0:3:47:8C:89:C2   Intel Corporation
  581 0:C:29:9E:EF:53   VMWare, Inc
  351 0:B:DB:9B:46:FE   Dell ESG PCBA Test
  173 FF:FF:FF:FF:FF:FF Broadcast address
  105 0:9:6B:2:E9:3D    IBM Corporation
   91 0:4:76:45:61:39   3COM CORPORATION
   89 0:0:E2:94:B0:2A   Ando Electric Corporation
   80 0:6:5B:E6:F8:43   Dell Computer Corp
   39 0:C:29:14:1E:63   VMWare, Inc
   37 0:0:E2:92:EE:F    Ando Electric Corporation
   36 0:1:2:79:91:ED    3COM CORPORATION
```

Sorting by destination MAC shows that the attacks targeted one particular system. command: grep " -> " alert|grep -v IpLen |cut -d \ -f 4|sort|uniq -c|sort -nr
34 0:50:56:40:0:64 VMWare, Inc 24 1:0:5E:37:96:D0 DoD Internet Multicast (RFC-1112) 24 0:1:3:88:29:92 3COM CORPORATION 22 0:D0:59:C6:5E:14 AMBIT MICROSYSTEMS CORP 20 0:6:5B:D8:BF:ED Dell Computer Corp 11 0:E0:B8:3D:20:40 GATEWAY 2000 11 0:8:74:7:31:EE Dell Computer Corp 10 0:C:29:39:6E:67 VMWare, Inc 9 0:2:A5:B6:E2:E3 Compaq Computer Corporation 8 0:E0:98:A1:7F:DA AboCom Systems, Inc 7 0:B:DB:17:F4:C9 Dell ESG PCBA Test 5 8:0:46:79:F7:7C SONY CORPORATION LTD. 4 0:3:FF:DF:95:84 Connectix 3 1:0:5E:7F:FF:FA DoD Internet Multicast (RFC-1112) 3 0:A0:C9:BA:6D:85 Intel Corporation 2 0:B:DB:DF:53:8D Dell ESG PCBA Test 2 0:A:95:7C:24:0 Apple Computer, Inc. 1 0:8:74:5:B7:F8 Dell Computer Corp 1 0:0:39:F2:67:88 Point Multimedia Systems

The target was heavily scanned.

IP addresses with the destination MAC 0:50:56:40:0:6d	
cut -d\ -f3,8 tcpdump.eth.5.txt|sed 's/^\(.* .*\..*\..*\..*\)\..*$/\1/' |grep \ '0:50:56:40:0:6d'|uniq|sort|uniq > 0-50-56-40-0-6d.ip.txt cut -d\ -f 2 0-50-56-40-0-6d.ip.txt | sed 's/\://' | sort | uniq | grep -c "" 1542	
cut -d\ -f 2 0-50-56-40-0-6d.ip.txt | sed 's/\://' | sort | uniq | cut -d\. -f1,2,3 | sort -n | uniq -c |sort -nr| sed 's/$/.0\/24/'	
256 172.20.11.0/24 255 172.22.201.0/24 255 172.20.201.0/24 255 172.20.12.0/24 254 192.168.22.0/24 254 192.168.17.0/24 1 198.41.0.0/24 1 198.123.30.0/24 1 172.27.1.0/24 1 172.20.102.0/24	1 172.11.11.0/24 1 172.10.11.0/24 1 149.134.52.0/24 1 149.134.30.0/24 1 134.248.127.0/24 1 127.0.0.0/24 1 12.162.170.0/24 1 10.3.200.0/24 1 102.168.17.0/24

The alerts from the target MAC were all error responses to bad traffic, except for "Short UDP packet" which was caused by snaplen 96 (the capture length)

```
grep -B2 " 0:50:56:40:0:6D -> " alert | grep "\[\*\*\] " | sort|uniq -c | sort -nr
   1326 [**] [1:402:4] ICMP Destination Unreachable (Port Unreachable) [**]
    690 [**] [1:399:4] ICMP Destination Unreachable (Host Unreachable) [**]
    418 [**] [1:401:4] ICMP Destination Unreachable (Network Unreachable) [**]
     34 [**] [1:408:4] ICMP Echo Reply [**]
     33 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length
     18 [**] [1:449:4] ICMP Time-To-Live Exceeded in Transit [**]
     10 [**] [1:491:6] INFO FTP Bad login [**]
      7 [**] [1:524:6] BAD-TRAFFIC tcp port 0 traffic [**]
      4 [**] [1:451:4] ICMP Timestamp Reply [**]
      4 [**] [1:382:4] ICMP PING Windows [**]
      2 [**] [1:718:6] TELNET login incorrect [**]
      1 [**] [1:404:4] ICMP Destination Unreachable (Protocol Unreachable) [**]
```

The "tcp port 0 traffic" alerts were resets in response to syn packets.

```
egrep  " 172\.20\.11\.2\.0 > 10\.10\.10\.141\.|10\.10\.10\.141\..* > 172.20.11.2.0:"
tcpdump.eth.5.txt | tail -n 2
14:09:22.899069 0:d0:59:c6:5e:14 0:50:56:40:0:6d 0800 60: 10.10.10.141.62917 >
172.20.11.2.0: S 3868:3868(0) win 512
14:09:22.939627 0:50:56:40:0:6d 0:d0:59:c6:5e:14 0800 60: 172.20.11.2.0 >
10.10.10.141.62917: R 0:0(0) ack 3869 win 0 (DF)
```

A lot of malicious traffic was sent to the target MAC
command: grep -B2 " -> 0:50:56:40:0:6D " alert | grep "\[**\] " | sort | uniq -c | sort -nr

```
  18176 [**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**]
   5041 [**] [1:465:1] ICMP ISS Pinger [**]
   2144 [**] [1:620:3] SCAN Proxy (8080) attempt [**]
   1020 [**] [1:474:1] ICMP superscan echo [**]
    545 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length
    394 [**] [1:615:4] SCAN SOCKS Proxy attempt [**]
    287 [**] [1:1417:2] SNMP request udp [**]
    146 [**] [1:1420:2] SNMP trap tcp [**]
    140 [**] [1:1418:2] SNMP request tcp [**]
    116 [**] [1:1421:2] SNMP AgentX/tcp request [**]
    105 [**] [1:618:4] SCAN Squid Proxy attempt [**]
     70 [**] [1:1443:2] TFTP GET passwd [**]
     60 [**] [1:1411:3] SNMP public access udp [**]
     59 [**] [1:467:1] ICMP Nemesis v1.1 Echo [**]
     46 [**] [1:1419:2] SNMP trap udp [**]
     29 [**] [1:553:4] POLICY FTP anonymous login attempt [**]
     27 [**] [1:628:2] SCAN nmap TCP [**]
     24 [**] [1:365:5] ICMP PING (Undefined Code!) [**]
```

A lot of malicious traffic was sent to the target MAC
command: grep -B2 " -> 0:50:56:40:0:6D " alert | grep "\[**\] " | sort | uniq -c | sort -nr

```
     24 [**] [111:10:1] (spp_stream4) STEALTH ACTIVITY (XMAS scan) detection [**]
     21 [**] [111:12:1] (spp_stream4) NMAP FINGERPRINT (stateful) detection [**]
     14 [**] [1:361:7] FTP site exec [**]
     12 [**] [1:2049:1] MS-SQL ping attempt [**]
     10 [**] [1:1413:2] SNMP private access udp [**]
      8 [**] [1:1777:2] FTP EXPLOIT STAT * dos attempt [**]
      7 [**] [1:524:6] BAD-TRAFFIC tcp port 0 traffic [**]
      7 [**] [1:408:4] ICMP Echo Reply [**]
      6 [**] [1:566:3] POLICY PCAnywhere server response [**]
      6 [**] [1:528:4] BAD-TRAFFIC loopback traffic [**]
      6 [**] [1:1957:3] RPC sadmind UDP PING [**]
      6 [**] [1:1893:1] SNMP missing community string attempt [**]
      6 [**] [1:1867:1] MISC xdmcp info query [**]
      6 [**] [1:1504:5] MISC AFS access [**]
      6 [**] [105:1:1] spp_bo: Back Orifice Traffic detected (key: 31337) [**]
      4 [**] [1:453:4] ICMP Timestamp Request [**]
      4 [**] [1:388:4] ICMP Address Mask Request [**]
      4 [**] [1:336:5] FTP CWD ~root attempt [**]
      4 [**] [1:237:1] DDOS Trin00:MastertoDaemon(defaultpassdetected!) [**]
      4 [**] [1:1992:1] FTP LIST directory traversal attempt [**]
      4 [**] [1:1449:3] POLICY FTP anonymous (ftp) login attempt [**]
      3 [**] [1:604:5] RSERVICES rsh froot [**]
      3 [**] [1:501:2] MISC source route lssre [**]
      3 [**] [1:1432:4] P2P GNUTella GET [**]
      2 [**] [1:659:4] SMTP expn decode [**]
      2 [**] [1:356:5] FTP passwd retrieval attempt [**]
      2 [**] [1:332:5] FINGER 0 query [**]
      2 [**] [1:330:6] FINGER redirection attempt [**]
      2 [**] [1:327:5] FINGER remote command pipe execution attempt [**]
      2 [**] [1:326:5] FINGER remote command ; execution attempt [**]
      2 [**] [1:323:4] FINGER root query [**]
      2 [**] [1:1728:2] FTP CWD ~<CR><NEWLINE> attempt [**]
      1 [**] [1:489:5] INFO FTP No Password [**]
      1 [**] [1:335:4] FTP .rhosts [**]
      1 [**] [1:255:8] DNS zone transfer TCP [**]
      1 [**] [1:245:1] DDOS mstream handler ping to agent [**]
      1 [**] [1:239:1] DDOS shaft handler to agent [**]
      1 [**] [1:236:3] DDOS Stacheldraht client check gag [**]
      1 [**] [1:1928:3] FTP shadow retrieval attempt [**]
      1 [**] [1:1919:3] FTP CWD overflow attempt [**]
      1 [**] [1:1444:2] TFTP Get [**]
      1 [**] [111:13:1] (spp_stream4) STEALTH ACTIVITY (SYN FIN scan) detection [**]
```

There are a few alerts that do not involve the target MAC, but they can be ignored. It is possible that the other systems may have been communicating with each other.
grep ':..\?:..\?:..\?:..\?:..\?|^\[**\] ' alert|grep -vB1 '^\[| 0:50:56:40:0:6D'|grep '^\['|sort|uniq -c 603 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [4 [**] [1:365:5] ICMP PING (Undefined Code!) [**] 4 [**] [1:408:4] ICMP Echo Reply [**]

One issue with spoofing is if the target could be used as a dummy for an idlescan[10] The table below indicates that the target may maintain a different IP sequence for each foreign IP and therefore could not be used. However a more through analysis is required for a real answer..

The target may maintain a different IP sequence for each foreign IP.
egrep " 192\.168\.17\.68\..* > " tcpdump.eth.5.txt| grep " ack "| grep -v 'ack [210] win '| cut -d\ -f6-12| tail -n 32|head -n 12
192.168.17.68.80 > 10.10.10.165.4729: R 0:0(0) ack 1180636066 192.168.17.68.80 > 10.10.10.165.4732: R 0:0(0) ack 1181123598 192.168.17.68.80 > 10.10.10.234.1069: R 0:0(0) ack 1730051307 192.168.17.68.80 > 10.10.10.165.4735: R 0:0(0) ack 1181576792 192.168.17.68.80 > 10.10.10.165.4737: R 0:0(0) ack 1182006667 192.168.17.68.80 > 10.10.10.165.4739: R 0:0(0) ack 1182438820 192.168.17.68.80 > 10.10.10.165.4742: R 0:0(0) ack 1182878632 192.168.17.68.80 > 10.10.10.165.4744: R 0:0(0) ack 1183275579 192.168.17.68.80 > 10.10.10.165.4750: R 0:0(0) ack 1184026471 192.168.17.68.80 > 10.10.10.234.1070: R 0:0(0) ack 1732013017 192.168.17.68.80 > 10.10.10.165.4753: R 0:0(0) ack 1184605667 192.168.17.68.80 > 10.10.10.165.4755: R 0:0(0) ack 1185072914

10.10.10.113 was the source for 18202 alerts, Mainly a NULL Scan.of the target.
grep -B3 '^10.10.10.113.* -' logs/alert|grep -iB2 '0:50:56:40:0:6d'|grep '\[**\]'|sort|uniq -c
4 [**] [111:10:1] (spp_stream4) STEALTH ACTIVITY (XMAS scan) detection [**] 8 [**] [111:12:1] (spp_stream4) NMAP FINGERPRINT (stateful) detection [**] 18162 [**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**] 8 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**] 4 [**] [1:628:2] SCAN nmap TCP [**]

Destination IP's
grep '0:50:56:40:0:6d' tcpdump.eth.5.txt|grep "10\.10\.10\.113\..* > "|cut -d\> -f2|cut -d\. -f1,2,3,4|sort|uniq -c|sort -nr
Count IP 6626 192.168.17.129 6321 192.168.17.68 5291 192.168.17.135

The number of distinct destination ports scanned for each IP
grep '0:50:56:40:0:6d' tcpdump.eth.5.txt|grep "10\.10\.10\.113\..* > "|cut -d\> -f2|cut -d\. -f1,2,3,4,5|sort|uniq|cut -d\. -f1,2,3,4|uniq -c
Count IP 1670 192.168.17.129 1657 192.168.17.135 1691 192.168.17.68

The Alert Breakdown
grep -B3 '^10.10.10.113.* -> 192.168.17.129' logs/alert|grep -iB2 '0:50:56:40:0:6d'|grep '\[**\]'|sort|uniq -c
1 [**] [111:10:1] (spp_stream4) STEALTH ACTIVITY (XMAS scan) detection [**] 2 [**] [111:12:1] (spp_stream4) NMAP FINGERPRINT (stateful) detection [**] 6607 [**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**] 2 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**] 1 [**] [1:628:2] SCAN nmap TCP [**]
grep -B3 '^10.10.10.113.* -> 192.168.17.68' logs/alert|grep -iB2 '0:50:56:40:0:6d'|grep '\[**\]'|sort|uniq -c
3 [**] [111:10:1] (spp_stream4) STEALTH ACTIVITY (XMAS scan) detection [**] 6 [**] [111:12:1] (spp_stream4) NMAP FINGERPRINT (stateful) detection [**] 6264 [**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**] 6 [**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**] 3 [**] [1:628:2] SCAN nmap TCP [**]
grep -B3 '^10.10.10.113.* -> 192.168.17.135' logs/alert|grep -iB2 '0:50:56:40:0:6d'|grep '\[**\]'|sort|uniq -c
5291 [**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**]

Syn's were sent to ports 1 and 20 of 192.168.17.129 and 192.168.17.68. No Syn's were sent to 192.168.17.135
command grep '0:50:56:40:0:6d' tcpdump.eth.5.txt|grep " S "|grep "10\.10\.10\.113\..* > "|cut -d\ -f8,9|sort|uniq -c
Count IP 6 192.168.17.129.1: S 1 192.168.17.129.20: S 18 192.168.17.68.1: S 3 192.168.17.68.20: S

It looks like every open port will respond with a reset to a null scan.	
grep '0:50:56:40:0:6d .* > 10\.10\.10\.113\.' tcpdump.eth.5.txt|cut -d\ -f6,9|egrep -v "68\.20 |129\.20 "|sort|uniq -c|sort -nr	
1 192.168.17.68.80 R 1 192.168.17.68.53 R 1 192.168.17.68.443 R 1 192.168.17.68.25 R 1 192.168.17.68.23 R 1 192.168.17.68.22 R 1 192.168.17.68.21 R 1 192.168.17.135.53 R	1 192.168.17.135.22 R 1 192.168.17.135.20 R 1 192.168.17.129.80 R 1 192.168.17.129.53 R 1 192.168.17.129.443 R 1 192.168.17.129.25 R 1 192.168.17.129.23 R 1 192.168.17.129.21 R

192.168.17.135 answering to a null scan from10.10.10.113.
grep "10\.10\.10\.113\." tcpdump.eth.5.txt|grep "192\.168\.17\.135\."|egrep "135\.53[:]|135\.20[:]|135\.22[:]"|sort|uniq
14:14:55.665504 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: 10.10.10.113.59194 > 192.168.17.135.22: . win 4096 14:14:55.671788 0:50:56:40:0:6d 0:a:95:7c:24:0 0800 60: 192.168.17.135.22 > 10.10.10.113.59194: R 0:0(0) ack 0 win 0 (DF) 14:15:51.827656 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: 10.10.10.113.59194 > 192.168.17.135.53: . win 2048 14:15:51.854980 0:50:56:40:0:6d 0:a:95:7c:24:0 0800 60: 192.168.17.135.53 > 10.10.10.113.59194: R 0:0(0) ack 0 win 0 (DF) 14:16:10.128439 0:a:95:7c:24:0 0:50:56:40:0:6d 0800 60: 10.10.10.113.59194 > 192.168.17.135.20: . win 1024 14:16:10.132772 0:50:56:40:0:6d 0:a:95:7c:24:0 0800 60: 192.168.17.135.20 > 10.10.10.113.59194: R 0:0(0) ack 0 win 0 (DF)

Three of the hosted VM's . could be Linux systems. The TTL discrepancy could be caused by a separate VM on the target that that is configured as a firewall.
cmd grep -A2 ' 0:50:56:40:0:6D -> ' logs/alert|grep ' TCP .* IpLen:'|cut -d\ -f1,4,5|sed 's/:.. T/ T/' |sort|uniq -c|sort -k2
7 172.20.11.2:0 TCP TTL:62 4 172.20.201.135 TCP TTL:62 8 172.20.201.198 TCP TTL:62

Attack Mechanism

Here is a brief description of the idle scan that was invented by Antirez[8] The goal of the idle scan is that no traffic travels between the attacker and the target. The attacker sends a non-spoofed packet to a third system (the dummy) that uses sequential IP ID's. The attacker then reads the IP ID from the reply. Next the attacker sends a spoofed packet where the source field contains the IP of the real target. The dummy system responds with an error message to the target. If the target has the same port open it

should respond otherwise it should not respond. A response would increment the IP ID counter on the dummy. Then the attacker sends a non-spoofed packet to the dummy machine and reads the IP ID. From the difference of the IP ID's he can tell of the target responded or not. If the third system is not busy then many ports can be queried before the attacker has to use a non-spoofed packet. Fig 2.1.2 indicates that the target may maintain an independent IP sequence for each foreign IP. Therefore it could not be used as a dummy for an idle scan. However a more through analysis is required for a definitive answer

A null scan is a TCP packet with no flags. When it is sent to an Open port a reset may be sent back but would not be entered into the hosts logs. A closed port should not respond. This way an attacker can scan for open ports without detection. Some older firewalls would not drop a null TCP packet. Traffic to and from port 0 is not allowed in the RFC's. Therefore different OS's respond differently. It can be used for OS typing.

Correlations:

Several other people have posted detects from the same source file to intrusions@incidents.org. They all came to different results because they did not consider a pen test. I could not find a CVE entry for a Null Scan [11]

Evidence of active targeting: The attacks were mainly to three specific IP's

Severity:
Severity = (criticality + lethality)-(system countermeasures + network countermeasures) = (5 + 5) - (3 + 0) = 7
Criticality: 5 A Vmware server may hold several virtual servers. If compromised then all may be compromised
Lethality 5 If an attack of this size happened on a production LAN, I would turn everything off. Normal Business would stop and the business continuity plan would be invoked.
System countermeasures 3 The target was not broken into, but it did responded to the null scan and port 0 traffic.
Network countermeasures 0 Nothing was blocked from the VMware server.

Defensive recommendation:

It should be obvious that none of this traffic be allowed to get to the Internet or a production LAN.

There are several critical unknowns with this analysis. VMware hosts other OS's but they are not known. There may have been a separate firewall on the target and/or the hosted systems had some sort of firewall. The target configuration was unknown. The target may be dual homed.

The target seemed to do OK but there is room for improvement.

Drop all traffic to port 0. Traffic to and from port 0 can be used for OS typing. This would be useful to an attacker and probably not much use to IT.

Drop all null TCP packets. The target answered to a null scan. There is no legal reason for this traffic.

The error messages from the target would be useful to an attacker or a legitimate SA. It would increase security to disable the error responses, and the LAN would still work, but it would be much harder to diagnose and repair problems. One option would be to set up one of the virtual machines as a firewall/router and log the errors. Then the SA would be able to see the messages but an attacker would not.

Multiple choice test question:

Which IP range includes IP's that are legal on the Internet?
A) 10.0.0.0 - 10.255.255.255.
B) 192.168.0.0 - 192.168.255.255.
C) 172.16.0.0 - 172.31.255.255 .
D) 224.0.0.0 - 239.255.255.255
E) none

Answer E: All subnets are reserved by ICANN. See http://whois.arin.net

References

[1] Spirent Communications. Security & Web Infrastucture.
URL: http://www.spirentcom.com/analysis/index.cfm?WS=65 (2 May 2004)

[2] Ando Electric Co., Ltd. URL: http://www.ando.com (20 March 2004).

[3] Cahners Publishing Company.
Electronic News: CADENCE TO ACQUIRE AMBIT?(Cadence Design Systems considers acquisition of Ambit Design Systems) (Company Business and Marketing). 24 August 1998.
URL: http://www.findarticles.com/cf_dls/m0EKF/n2233_v44/21054771/p1/article.jhtml (20 March 2004).

[4] AboCom Systems , Inc. URL: http://www.abocom.com/ (20 March 2004).

[5] VMware, Inc. URL: http://www.vmware.com/support/ (20 March 2004).

[6] Heim, Kristi. Mercury News Microsoft buys assets of Connectix. 20 Feb. 2003.
URL: http://www.siliconvalley.com/mld/siliconvalley/5222197.htm (20 March 2004).

[7] Sony Corporation of America. URL: http://sony.com/ (20 March 2004).

[8] Cavebear. Ethernet Codes: Multicast (including Broadcast) Addresses. 09 March 1999.
URL: http://www.cavebear.com/CaveBear/Ethernet/multicast.html (20 March 2004).

[9] IEEE OUI and Company_id Assignments
URL: http://standards.ieee.org/regauth/oui/index.shtml (20 March 2004).

[10] Vaskovich, Fyodor. Idle Scanning and related IPID games.13 November 2003.
URL: http://www.insecure.org/nmap/idlescan.html (20 March 2004).

[11] URL: http://cve.mitre.org (20 March 2004).

GIAC GCIA Version 3.4 Practical Detect 2

Source of Trace: http://www.incidents.org/logs/oos/oos_report_031228 [1]

These files were also needed as well
alert.031211.gz alert.031214.gz alert.031217.gz alert.031220.gz alert.031223.gz alert.040107.gz alert.040110.gz alert.040118.gz alert.040121.gz alert.040124.gz alert.031212.gz alert.031215.gz alert.031218.gz alert.031221.gz alert.040105.gz alert.040108.gz alert.040116.gz alert.040119.gz alert.040122.gz alert.031213.gz alert.031216.gz alert.031219.gz alert.031222.gz alert.040106.gz alert.040109.gz alert.040117.gz alert.040120.gz alert.040123.gz
oos_report_030711 oos_report_031010 oos_report_031216 oos_report_031221 oos_report_031226 oos_report_031231 oos_report_040105 oos_report_040110 oos_report_040115 oos_report_040120 oos_report_030719 oos_report_031024 oos_report_031217 oos_report_031222 oos_report_031227 oos_report_040101 oos_report_040106 oos_report_040111 oos_report_040116 oos_report_040121 oos_report_030723 oos_report_031102 oos_report_031218 oos_report_031223 oos_report_031228 oos_report_040102 oos_report_040107 oos_report_040112 oos_report_040117 oos_report_040122 oos_report_030823 oos_report_031211 oos_report_031219 oos_report_031224 oos_report_031229 oos_report_040103 oos_report_040108 oos_report_040113 oos_report_040118 oos_report_040123 oos_report_030905 oos_report_031215 oos_report_031220 oos_report_031225 oos_report_031230 oos_report_040104 oos_report_040109 oos_report_040114 oos_report_040119 oos_report_040124
scans.031213.gz scans.031216.gz scans.031219.gz scans.031223.gz scans.031226.gz scans.031229.gz scans.040107.gz scans.031211.gz scans.031214.gz scans.031217.gz scans.031221.gz scans.031224.gz scans.031227.gz scans.040105.gz scans.040108.gz scans.031212.gz scans.031215.gz scans.031218.gz scans.031222.gz scans.031225.gz scans.031228.gz scans.040106.gz scans.040109.gz

Detect was generated by:

There is no MAC data in any of the files so the hardware is unknown. The IP's in many of the files belong to a large university in the US. For security reasons they have all been changed to MY.NET.xxx.xxx. The assumption is that a NIDS running Snort[2] is sniffing the link to the Internet.

Here is an interesting detect.

Null TCP packet with source port 8
grep -A 2 "68.122.128.111:8 -> " oos_report_031228
01/01-04:40:30.475644 68.122.128.111:8 -> MY.NET.12.4:110 TCP TTL:80 TOS:0x0 ID:4660 IpLen:20 DgmLen:40 ******** Seq: 0x6AFD001 Ack: 0x589824C1 Win: 0x800 TcpLen: 20

This is from a Snort Out Of Spec file. It contains captures of IP packets that claim to be TCP but have illegal settings. The packet shown above had no flags set. It has source port 8 which is unassigned and is not associated with malware according to SANS [3]

Probability the source address was spoofed: This packet was crafted with a program such as nmap[4] or Hping2[5]. It could be spoofed. However later it will be argued that it probably is not spoofed.

Description of attack:

The attack started on December 20 2003 and lasted to at least January 28 2004

```
cmd grep -a "68\.122\.128\.111" oos* | sort | head -n 1
oos_report_031216:12/20-00:22:43.689805 68.122.128.111:57092 ->
MY.NET.12.4:110
cmd grep -a "68\.122\.128\.111" oos* | sort | tail -n 1
oos_report_040124:01/28-04:47:39.537950 68.122.128.111:61966 ->
MY.NET.12.4:110
```

There were over 500 oos packets detected.

```
grep -a "68\.122\.128\.111" oos* | grep -c " -> MY.NET\.12\.4:110"
516
```

Some scans were not Out of Spec

```
ls scans.*.gz | sed 's/^.*$/gzip -cd & | grep "68\.122\.128\.111"/' |sh | grep -c
"MY\.NET\.12\.4:110"
584
```

The mail server was the only destination.

```
  grep -a "68\.122\.128\.111" oos* | grep -cv " -> MY.NET\.12\.4:110"
  0
  ls scans.*.gz | sed 's/^.*$/gzip -cd & | grep "68\.122\.128\.111"/' |sh |grep -cv
  "MY\.NET\.12\.4:110"
  0
```

The OOS files contained no duplicate source ports,

```
grep -a "68\.122\.128\.111" oos* | \
sed 's/^.* 68.122.128.111:\(.*\) -> .*$/\1/' |sort|uniq -c|sort -nr|head -n 1
     1 9998
```

The scan files have 216 duplicate source ports.and one triplicate

```
grep -a "68\.122\.128\.111:" scans/* | sed 's/^.* 68.122.128.111:\(.*\) -> .*$/\1/'
| sort|uniq -c|sort -nr|sed 's/^\W*\(.\)\W.*$/\1/'|uniq -c
     1 3
   216 2
   149 1
```

There were 217 Syn scans and 365 Null scans

```
grep -a "68\.122\.128\.111" scans/* | grep -c 'SYN'
217
grep -a "68\.122\.128\.111" scans/* | grep -c 'NULL'
365
```

There are no duplicate Syn scans or duplicate Null scans

```
grep -a "68\.122\.128\.111:" scans/* | grep  "SYN" | sed 's/^.* 68.122.128.111:\(.*\) ->
.*$/\1/' | sort|uniq -c|sort -nr|head -n 1
      1 9994
grep -a "68\.122\.128\.111:" scans/* | grep  "NULL" | sed 's/^.* 68.122.128.111:\(.*\) -
> .*$/\1/' | sort|uniq -c|sort -nr|head -n 1
      1 9994
```

Here are two other scans. They are have the ECN bit set. The first one caused the triplicate port entry. These entries are not separated. That would cause grep to place a "-" between them.

```
grep -a "68\.122\.128\.111:" scans/* | grep -vC2  "SYN\|NULL"
scans/031218.txt:Dec 18 08:30:37 68.122.128.111:29188 -> MY.NET.12.4:110 SYN ******S*
scans/031218.txt:Dec 18 08:30:37 68.122.128.111:29188 -> MY.NET.12.4:110 NULL ********
scans/031218.txt:Dec 18 08:31:06 68.122.128.111:29188 -> MY.NET.12.4:110 UNKNOWN
*2*A**** RESERVEDBITS
scans/031218.txt:Dec 18 09:15:09 68.122.128.111:29700 -> MY.NET.12.4:110 NULL ********
scans/031218.txt:Dec 18 09:37:03 68.122.128.111:29956 -> MY.NET.12.4:110 SYN ******S*
scans/031218.txt:Dec 18 09:37:03 68.122.128.111:29956 -> MY.NET.12.4:110 NULL ********
scans/031218.txt:Dec 18 11:26:49 68.122.128.111:31236 -> MY.NET.12.4:110 NULL ********
scans/031218.txt:Dec 18 11:49:46 68.122.128.111:31492 -> MY.NET.12.4:110 UNKNOWN
*2*A**** RESERVEDBITS
scans/031218.txt:Dec 18 13:39:31 68.122.128.111:32772 -> MY.NET.12.4:110 NULL ********
scans/031218.txt:Dec 18 15:29:24 68.122.128.111:34052 -> MY.NET.12.4:110 SYN ******S*
```

All the null scans are the same size, have a TTL of 80 or 81.and have ID:4660

```
grep -aA1 "68\.122\.128\.111" oos* | grep -c "TCP TTL:8[10] TOS:0x0 ID:4660 IpLen:20 DgmLen:40"
515
```

```
grep -aA1 "68\.122\.128\.111" oos* | grep -c "TCP TTL:81 TOS:0x0 ID:4660 IpLen:20 DgmLen:40"
12
```

There was one bad entry. The end was Dg instead of DgmLen:40"

```
grep -aA1 "68\.122\.128\.111" oos* | grep " TTL:"|grep -v "TCP TTL:8[10] TOS:0x0
ID:4660 IpLen:20 DgmLen:40"
oos_report_040102-TCP TTL:80 TOS:0x0 ID:4660 IpLen:20 Dg
```

The attack consists of a series of sequences. In any one sequence, the source port is incremented by a multiple of 256. In the next sequence the port is one higher.

Perl program to analyze the sequences in detect 2.2

```
#!/usr/bin/perl -w
$option = shift;
$cmd =  "grep -ia \"68\\.122\\.128\\.111:\" $option/* |sort|cut -d: -f5 |cut -d\\  -f1\n";
print "$cmd";
@ar1 = `$cmd`;
print "the total number of detects is $#ar1 \n";
for $_ (@ar1) { $_ = trim ( $_) ; }
$lastVal = $ar1[0];
$duplicatePorts = 0;
for( $indx = 1; $indx < $#ar1 ; $indx++ ) {
 if ( $lastVal == $ar1[$indx] ) {  $duplicatePorts++; }
 if ( $lastVal > $ar1[$indx] ) {  print "$indx:  $lastVal > $ar1[$indx]\n"; }
 if (( $lastVal % 256 ) != ( $ar1[$indx] % 256 )) { print  "$indx:  $lastVal % 256 = "
. ( $lastVal % 256 ) ." !=  $ar1[$indx] % 256 = "  . ( $ar1[$indx] % 256 ) ."\n"; }
  $lastVal = $ar1[$indx];
  }
 print "Number of duplicate ports: $duplicatePorts\n";
sub trim {    my $a = shift;    $a =~  s/^\s*(.*?)\s*$/$1/;    return $a; }
```

Here is the scan analysis

```
./test_sequence_1.pl scans
grep -ia "68\.122\.128\.111:" scans/* |sort|cut -d: -f5 |cut -d\  -f1
the total number of detects is 583
133:  56580 > 8453
133:  56580 % 256 = 4 !=  8453 % 256 = 5
311:  58373 > 7177
311:  58373 % 256 = 5 !=  7177 % 256 = 9
505:  65033 > 10
505:  65033 % 256 = 9 !=  10 % 256 = 10
Number of duplicate ports: 218
```

Here is the Out of Spec Analysis

```
./test_sequence_1.pl oos
grep -ia "68\.122\.128\.111:" oos/* |sort|cut -d: -f5 |cut -d\  -f1
the total number of detects is 515
-13:  60164 > 8197
13:  60164 % 256 = 4 !=  8197 % 256 = 5
65:  61701 > 9990
65:  61701 % 256 = 5 !=  9990 % 256 = 6
111:  62982 > 12039
111:  62982 % 256 = 6 !=  12039 % 256 = 7
163:  65287 > 8
163:  65287 % 256 = 7 !=  8 % 256 = 8
207:  65288 > 265
207:  65288 % 256 = 8 !=  265 % 256 = 9
254:  52489 > 778
254:  52489 % 256 = 9 !=  778 % 256 = 10
308:  54538 > 2827
308:  54538 % 256 = 10 !=  2827 % 256 = 11
359:  56331 > 4364
359:  56331 % 256 = 11 !=  4364 % 256 = 12
411:  58124 > 6413
411:  58124 % 256 = 12 !=  6413 % 256 = 13
464:  59917 > 8206
464:  59917 % 256 = 13 !=  8206 % 256 = 14
Number of duplicate ports: 0
```

There are no scans involving 68.122.128.111 that do not involve MY.NET.12.4:110

```
j=10;while [ "$j" != 23 ];do gzip -cd scans.0312$j.gz | grep -a "68\.122\.128\.111:" |
grep -acv "MY\.NET\.12\.4:110";j=$[$j+1];done | grep -c ""
0
```

```
j=5;while [ "$j" != 10 ];do gzip -cd scans.04010$j.gz | grep -a "68\.122\.128\.111:" |
grep -acv "MY\.NET\.12\.4:110";j=$[$j+1];done| grep -c ""
0
```

Possibly they are getting DHCP leases from an IP range belonging to SBC in the US.

The OOS detects with ID:4660, sorted by number of occurrences.

```
grep -aiB1 " ID:4660 IpLen:" oos* |grep " -> "|cut -d: -f3|cut -d\  -f2|sort|uniq -c|sort -nr
    516 68.122.128.111  Pac Bell Internet Services
    364 67.119.233.217  SBC
    244 67.119.234.194  SBC
     62 67.119.232.52   SBC
     61 67.119.236.220  SBC
 ++     48 67.119.236.146  SBC
     13 151.196.169.97
     12 67.119.238.218  SBC
```

The OOS detects with ID:4660, sorted by number of occurrences.

```
     12 151.196.126.92
      7 61.59.124.234
      4 61.48.18.176
      3 195.111.1.93
      2 216.190.151.114
      1 210.49.212.4
```

None of the 67.119.x.x IP's are in the Dsheild [6] database of attackers.

According to ARIN All 67.119.0.0/16 and 68.122.0.0/16 IP's are in North America. [7]

```
whois 68.122.128.111 @whois.arin.net
Pac Bell Internet Services PBI-NET-10 (NET-68-120-0-0-1)
                          68.120.0.0 - 68.127.255.255
whois 67.119.233.217 @whois.arin.net
rback3.sndg02 PPPoX SBC067119232000020627 (NET-67-119-232-0-1)
                          67.119.232.0 - 67.119.235.255
whois 67.119.236.220 @whois.arin.net
PPPoX Pool - Rback3 SNDG02 SBC067119236000030309 (NET-67-119-236-0-1)
                          67.119.236.0 - 67.119.237.255
whois 67.119.238.218 @whois.arin.net
PPPoX Pool - Rback3 SNDG02 SBC067119238000030505 (NET-67-119-238-0-1)
                          67.119.238.0 - 67.119.239.255
```

All the oos packets with source 67.119.x.x have destination port 110

```
grep -a " 67\.119\." oos* | sed -e 's/^.* 67.119./67.119./' -e 's/:.* -> /: -> /'|sort|uniq -c
 62 67.119.232.52: -> MY.NET.12.4:110
182 67.119.233.217: -> MY.NET.12.4:110
 41 67.119.233.217: -> MY.NET.25.21:110
 41 67.119.233.217: -> MY.NET.25.22:110
 58 67.119.233.217: -> MY.NET.25.23:110
 42 67.119.233.217: -> MY.NET.25.24:110
 244 67.119.234.194: -> MY.NET.12.4:110
 48 67.119.236.146: -> MY.NET.12.4:110
 61 67.119.236.220: -> MY.NET.12.4:110
 12 67.119.238.218: -> MY.NET.12.4:110
```

This is the port for POP. That is how many PC's retrieve their email.

They all have ID:4660

```
grep -aA1 " 67\.119\." oos* | grep " ID:" | grep -c "TCP TTL:[78]. TOS:0x0 ID:4660
lpLen:20 DgmLen:40"
791
grep -aA1 " 67\.119\." oos* | grep " ID:" | grep -vc "TCP TTL:[78]. TOS:0x0 ID:4660
lpLen:20 DgmLen:40"
0
```

The TTL ranges from 77 to 83. This is just another sign that the packets are crafted.

```
grep -aA1 " 67\.119\." oos* |grep -v "^--"|sed -e 'N'  -e 's/\n//'  -e 'p' -e 'd'|sed -e 's/^.*
67.119.\(.*\):.*\( ->.*\)oos.*\(TTL:...\).*$/67.119.\1 \3/'|sort|uniq -c
```

```
    62 67.119.232.52 TTL:80
    14 67.119.233.217 TTL:77
    14 67.119.233.217 TTL:78
     9 67.119.233.217 TTL:79
    45 67.119.233.217 TTL:80
   155 67.119.233.217 TTL:81
   123 67.119.233.217 TTL:82
     4 67.119.233.217 TTL:83
   244 67.119.234.194 TTL:80
    48 67.119.236.146 TTL:82
    61 67.119.236.220 TTL:82
    12 67.119.238.218 TTL:80
```

Some IP's were active at different times but others overlap. Below they are sorted by filename and filetime, then sequential duplicate rows are counted

```
grep -aiB1 " ID:4660 IpLen:" oos* |grep " -> " |sort| cut -d: -f3|cut -d\  -f2|uniq -c| grep
"\W67\|\W68"
```

```
128 67.119.233.217
236 67.119.233.217
  5 67.119.236.220
 56 67.119.236.220
  4 67.119.236.146
 44 67.119.236.146
 62 67.119.232.52
 56 67.119.234.194
  5 67.119.234.194
 56 67.119.234.194
  5 67.119.234.194
  4 67.119.238.218
 56 67.119.234.194
  5 67.119.234.194
  4 67.119.238.218
 56 67.119.234.194
  5 67.119.234.194
  4 67.119.238.218
311 68.122.128.111
205 68.122.128.111
```

The alert files held nothing of interest.

```
ls alert.*.gz | sed 's/^.*$/gzip -cd & | grep "68\.122\.128\.111"/' | sh >
grep_alert.68.122.128.111.txt
```

```
grep -c "spp_portscan\| Null scan" grep_alert.68.122.128.111.txt
3315
```

```
grep -vc "spp_portscan\| Null scan" grep_alert.68.122.128.111.txt
0
```

```
ls alert.*.gz | sed 's/^.*$/gzip -cd & | grep "\\[\\*\\*\\] 67\\.119\\."/' | sh
12/19-15:59:35.152788  [**] Incomplete Packet Fragments Discarded [**]
67.119.153.47:0 -> MY.NET.100.165:0
01/17-08:27:00.148959  [**] Incomplete Packet Fragments Discarded [**]
67.119.172.223:0 -> MY.NET.97.82:0
01/17-11:21:07.784365  [**] Incomplete Packet Fragments Discarded [**]
67.119.172.223:0 -> MY.NET.97.82:0
01/17-12:21:45.344019  [**] Incomplete Packet Fragments Discarded [**]
67.119.172.223:0 -> MY.NET.97.82:0
```

There are a few other source IP's with a similar TTL and ID:4660

```
grep -aB1 " TTL:[78]. .* ID:4660 " oos* |grep -v "^--"|sed -e 'N'  -e 's/\n//'  -e 'p' -e 'd'|cut -d\  -f2-|sed -e 's/\(:..\?.\?.\?.\? ->.*\)oos.*\(TTL:..\).*$/ \2/' | sort|uniq -c| grep -v "\W67\.119\.\|\W68\.122\.128\.111"
```

12 151.196.126.92 TTL:88	2 216.190.151.114 TTL:80
13 151.196.169.97 TTL:89	4 61.48.18.176 TTL:76
1 210.49.212.4 TTL:80	7 61.59.124.234 TTL:85

Attack Mechanism

All could be spoofed There is no sign of a TCP connection. If they came from the same system then some are spoofed. It is unlikely that many different systems would send such similar packets. The question is "are they all spoofed". The IP's are all US and none are in the Dsheild attacker list. The destination is the mail server of US University.

```
cmd dig -x MY.NET.12.4 | grep -A1 "ANSWER SECTION" |grep "\."
4.12.MY.NET.in-addr.arpa. 86400 IN      PTR    mail.<some big university in the US>
```

The attacker is asking for trouble if all were not spoofed, but, there was not enough traffic for a DOS attack

```
grep -a "68\.122\.128\.111" scans/* | grep -c "
584
grep -a "68\.122\.128\.111" oos/* | grep -c "
516
```

Could this be a slow scan?

Every TCP packet has a different source port. That could mean that 68.122.128.111 is the target of an idle scan [8] and MY.NET.12.4 is being used as the "dummy".

The default mail server of a big university should be very busy, so we should see many non-spoofed packets.

```
grep -a "MY.NET.12.4" scans/* |sort|cut -c34-|cut -d: -f1|sort|uniq -c|sort -nr|head -n 5
      584 68.122.128.111
      304 68.122.140.118
       75 MY.NET.12.4
       33 69.139.78.59
        6 210.72.240.227
```

The first detect of 68.122.128.111 occurs after the last detect for 68.122.140.118. This rules out an idle scan.

```
cmd grep  "68\.122\.140\.118" scans/* | sort | tail -n 1
scans/031217.txt:Dec 17 10:12:37 68.122.140.118:43536 -> MY.NET.12.4:110 SYN
******S*
cmd grep  "68\.122\.128\.111" scans/* | sort | head -n 1
scans/031217.txt:Dec 17 14:14:35 68.122.128.111:16388 -> MY.NET.12.4:110 NULL
********
```

If the mail server uses sequential IP ID's and it responds to each null and syn scan then the attacker could be monitoring the number of IP packets sent and received. From this he could deduce the amount of email sent and received.

The goal of a null scan is that it is not recorded in the host's logs, however it stands out in the NIDS logs because there is no honest reason to use it on the Internet. Syn packets do not get recorded in the oos or alert files so they would be more easily hidden in all the spam and normal traffic. So why use any null scan packets at all?

Lets consider a trojan. According to SANS, the ProMail trojan is specific to port 110[9]. However a google[10] search for promail and "null scan" turn op nothing. It could be another trojan or that someone is trying to get the owner of 68.122.128.111 in trouble. They could be doing a "test" to see if and what the response is. The real attacker may read GCIA practicals. (If it's you, then you should be ashamed of yourself!)

This attack may not be malicious at all. This could be an undergrad computer science major who does not realize the consequences. It the mail server gats hacked, then the NIDS logs will be analyzed. If they see an American IP, the police could come visiting

If the systems use a shared cable and somebody has can sniff it. He could receive all the replies for the spoofed traffic, but that still does not explain the null scan.

My guesstimate is 30% that all the address's were spoofed. That leaves 70% that not all are spoofed, the attacker is in North America and is traceable.

Correlations: I did not find any strong correlations. ID:4660 was noticed by S. Gamble for a different attack. He also said "There is no CVE number for a Null Scan." [11]

Evidence of active targeting: Most of the traffic had the same destination IP and port.

Defensive recommendation:
It is probably feasible to find the attackers. They should be told that they went "over the line" and may have broken the law. I assume they made an innocent mistake here.
The mail server should not use predictable IP ID's. Most OS's have a patch, however it may not be installed because it indirectly reduces the risk to the system that needs it.
Null scan packets should be stopped at the firewall

Severity:
Severity=(criticality + lethality)(system countermeasures + network countermeasures) = (5 + 4) - (1 + 3) = 5
Criticality: 5 The default mail server for an organization is a necessity.
Lethality 4 They are probably not malicious but they are "over the line" and this attack could interrupt somebody's email.
system countermeasures 1 The mail server may have sequential ID's The mail server may be responding to all the Syn packets and null scan packets.
Network countermeasures 3 The NIDS captured the anomalous traffic. It is assumed that the null packets were stopped at the firewall. The attacking system should be "locket out" at the firewall until further investigation.

Multiple choice test question:
Which TCP packet is legal?

A ********
B ******S*
C ******SF
D *****RS*
E They all legal

Answer: B
A is a null scan , C is a SYN-FIN scan ,D Reset means an error message.

2.2.11 References

[1] SANS Institute. URL: http://www.incidents.org/logs (20 March 2004).

[2] URL: http://www.snort.org (20 March 2004).

[3] SANS Institute.
URL: http://isc.sans.org/port_details.html?port=8 (20 March 2004).

[4] URL: http://www.insecure.org (20 March 2004).

[5] URL: http://www.hping.org. (20 March 2004).

[6] URL: http://dshield.org/warning_explanation.php (20 March 2004).

[7] American Registry for Internet Numbers. ARIN WHOIS Database Search.
URL: http://ww2.arin.net/whois/ (20 March 2004).

[8] Vaskovich, Fyodor. Idle Scanning and related IPID games.13 November 2003.
URL: http://www.insecure.org/nmap/idlescan.html (20 March 2004).

[9] SANS Institute. Internet Storm Center Port 110.
URL: http://isc.sans.org/port_details.html?port=110 (20 March 2004).

[10] http://www.google.ca (20 March 2004).

[11] Gamble, Steven. GCIA Practical Assignment v3.3. 12 May 2003
URL: http://www.giac.org/practical/GCIA/Steven_Gamble_GCIA.pdf (20 March 2004).

GIAC GCIA Version 3.4 Practical Detect 3

Source of Trace: The target had Redhat 8.0, kernel 2.4.20-28.8,sendmail-8.12.8-9.80,Snort-2.0.5 and apache.2.0.4. It functioned as a mailserver, web server, firewall and IDS. The Internet connection has a wireless downlink and telephone uplink. Three days before the attack, RedHat had declared "end of life" for RH 8.0.

Detect was generated by:

Sendmail was configured not to send any email unless root ran "sendmail -q". [1]
Then it was set to accept any relay request and left for several months. When the spam relay attack started, a tcpdump capture was started. Tcpdump -s 0 -w was run from 01/17/2004 12:34:16 to 01/18/2004 12:47:10.

Probability the source address was spoofed: The IP cannot be spoofed for SMTP.

Program used to analyze capture files. Snort[2] was run with all rules uncommented.

```
#!/bin/sh
#In Snort-2.0.5 if two alerts are triggered then the second takes precedence. The result is
#alerts with "(Undefined Code!)". fix: move problem alerts in icmp-info.rules to the top.
cd /1/backup/giac/gcia_assignment
RAWPATH="p2_detect_3/tcpdump_w_jan_17_2004_a.libcap";
echo $RAWPATH
ls -l $RAWPATH
SNORTPATH="snort/snort-2.0.5-bin";
LOGPATH="p2_detect_3/logs";
CONFPATH="snort/snort.conf";
COMMAND="$SNORTPATH  -X -d -e -c $CONFPATH -r $RAWPATH -l    $LOGPATH"
echo "$COMMAND";
 `$COMMAND`
#-c specify location of the Snort configuration file
#-h sets the "HOME_NET" variable used in snort.conf to identify the monitored networks
#-d display application-layer data in the alerts
#-e display link-layer data in the alerts
#-k  sets checksum checking off altogether because we know they are all wrong
#-r   read data from a file rather than from a network interface
#-l    write the logs to this directory
#-q  don't display all that banner, initialisation and summary info.
#-X  display hex output
```

Description of attack

From Jan 17 2004 11:02:14 EST to Jan 18 06:54:57 the target received 6134 relay attempts for a spam email from 56 different hosts. Also received were 12 relay attempts from seven hosts, for an email with a 0 or 17 byte message with "ameill" in the header. I assume these were probes from the spammer.

There were a total of 12 probe emails.

```
egrep -ci 'from:? drone|for <ameill' ../mqueue/drone/q*|grep -c ''
12
```

Six of the probe messages were 0 bytes and the other six were 17 bytes.

```
ls -g ../mqueue/drone/d*|sed -e 's/^.*smmsp *//' -e 's/ .*//'|sort|uniq -c|sort -nr
      6 17
      6 0
```

There were seven hosts. One sent three probes, two sent two and the rest sent one.

```
egrep -i 'Received: from:? ' ../mqueue/drone/q*|cut -d\[  -f2-|cut -d\]  -f1|sort|uniq -c
```

3 218.70.10.177	1 219.153.151.210	2 219.153.153.92
2 218.70.137.24	1 219.153.153.179	1 219.153.154.205
1 218.70.9.201		

17 byte messages were from www.xyz34.uk.co.sg. The 0 b one's were from ameill-1'

```
egrep -i 'Received: from:? ' ../mqueue/drone/* | sed 's/..\/mqueue\/drone\/q//' > 1.tmp
ls -l ../mqueue/drone/d* | sed -e 's/^.*smmsp *//' -e 's/Jan .*\.\./\.\./'\
 -e 's/^\(..\? \)\(.*\)$/\2  message size = \1 bytes/'|sed 's/..\/mqueue\/drone\/d//' | sort
fi0DNsc3E016673:H??Received: from www.xyz34.uk.co.sg ([219.153.156.126])
fi0DNsc3E016673  message size = 17  bytes
fi0E0ub3E030563:H??Received: from www.xyz34.uk.co.sg ([218.70.9.201])
fi0E0ub3E030563  message size = 17  bytes
fi0EBon3E003921:H??Received: from ameill-1 ([219.153.154.205])
fi0EBon3E003921  message size = 0  bytes
fi0EN493E007547:H??Received: from www.xyz34.uk.co.sg ([218.70.10.177])
fi0EN493E007547  message size = 17  bytes
fi0F5RX3E009850:H??Received: from ameill-1 ([218.70.10.177])
fi0F5RX3E009850  message size = 0  bytes
fi0FDNH3E012178:H??Received: from ameill-1 ([218.70.10.177])
fi0FDNH3E012178  message size = 0  bytes
fi0FNWC3E016230:H??Received: from www.xyz34.uk.co.sg ([219.153.153.92])
fi0FNWC3E016230  message size = 17  bytes
fi0GBKf3E019882:H??Received: from ameill-1 ([219.153.153.92])
fi0GBKf3E019882  message size = 0  bytes
fi0H2qY3E027771:H??Received: from www.xyz34.uk.co.sg ([219.153.153.179])
fi0H2qY3E027771  message size = 17  bytes
fi0HG4HQk002845:H??Received: from ameill-1 ([218.70.137.24])
fi0HG4HQk002845  message size = 0  bytes
fi0I0CMQk005636:H??Received: from www.xyz34.uk.co.sg ([218.70.137.24])
fi0I0CMQk005636  message size = 17  bytes
fi0I82eQk008067:H??Received: from ameill-1 ([219.153.151.210])
fi0I82eQk008067  message size = 0  bytes
```

The log records having to do with the probes. Taken on Jan 21 19:58:59 EST 2004

32 log entries contained the phrase "ameil"
egrep 'ameil' /var/log/maillog*|grep -c 'ameil' 32
These referred to 24 different probe emails.
grep ameil /var/log/maillog*| cut -d\ -f6|sort|uniq|grep -c '' 24
These referred to 62 maillog entries.
grep ameil /var/log/maillog*| cut -d\ -f6|sort|uniq|sed 's/^.*$/grep & \/var\/log\/maillog*'/|sh|sort|uniq | grep -c '' 62

There were two more relevant maillog entries that are not from or to ameil

grep ameil /var/log/maillog*| cut -d\ -f6|sort|uniq|sed 's/^.*$/grep & \/var\/log\/maillog*'/ |sh > egrep_ameil_maillog.c.txt
grep -i '' from'' egrep_ameil_maillog.c.txt|sed 's/^.*from[=]//'|cut -d\ -f1|sort|uniq|\ sed -e 's/^.\(.*\).$/grep \1 \/var\/log\/maillog* /' -e 's/>//'|sh|sort|uniq | grep -c '''' 64

At least 13 probe emails arrived after promiscuous relaying was revoked.

egrep 'arg1=<ameill.*reject=550' /var/log/maillog*|grep -c '' 13

There were 12 distinct IP addresses. They are all from Asia.

grep '\[..\?.\?\...\?.\?\...\?.\?\...\?.\?\]' egrep_ameil_maillog.e.txt \ | sed 's/^.*\[\(..\?.\?\...\?.\?\...\?.\?\...\?.\?\)\].*$/\1 \2\3/' |sort|uniq	
211.158.77.186	219.153.151.210
218.70.10.177	219.153.153.1
218.70.137.24	219.153.153.179
218.70.9.201	219.153.153.92
219.153.150.64	219.153.154.68
219.153.151.109	219.153.156.126

Message text of one of the 17 byte messages

N/S|N/S|N/S|N/S

Partial list of probe source ports.

sed -e 's/^.*\[/grep /' -e 's/\].*$/ 2.tmp/' 1.tmp|sh|sort|uniq 218.70.137.24.3606 218.70.137.24.3786 218.70.137.24.4324 219.153.151.210.3881 219.153.151.210.4557

Header file for a probe message.

```
V6
T1074038082
K1074038093
N0
P30125
F8bs
$_[219.153.156.126]
$rESMTP
$swww.xyz34.uk.co.sg
${daemon_flags}
${if_addr}$TARGET_IP
S<bss@fre.sg.co.nz>
rRFC822; ameill1@pufan.com
RPFD:<ameill1@pufan.com>
H?P?Return-Path: <?g>
H??Received: from www.xyz34.uk.co.sg ([219.153.156.126])
        by <hostname removed> (8.12.8/8.12.8) with ESMTP id i0DNsc3E016673
        for <ameill1@pufan.com>; Tue, 13 Jan 2004 18:54:42 -0500
H?M?Message-Id:
<200401132354.i0DNsc3E016673@TARGET_HOSTNAME>
H??From: bss@fre.sg.co.nz
H??To: ameill1@pufan.com
H??Subject: dbk`cfj`jf`chi?b`b`b`b?_c
H??Date: Thu, 15 Jan 2004 07:52:31 +0800
```

Unfortunately most of the probe emails were not captured by tcpdump

```
grep -R 'Received: from' ../mqueue/drone|sed -e 's/.*Received: from .*\[//' \
-e 's/\].*//' |sort -r|cut -d\.  -f1,2,3,4 |uniq|sed -e 's/^/grep /' -e 's/$/
tcpdump_w_jan_17_2004_a.pf0.b.txt/' |sh
219.153.151.210: UNKNOWN [16384:113:1420:1:-1:1:1:48].
218.70.137.24: UNKNOWN [16384:114:1420:1:-1:1:1:48].
```

The spam relay attack:

The spam relay attack started at Sat, 17 Jan 2004 11:02:09 -0500. Promiscuous relaying was revoked at Sun, 18 Jan 2004 07:00 -0500

```
grep -RiA1 "by TARGET_HOSTNAME"  ../mqueue/spam_q | grep -v "by
TARGET_HOSTNAME "|sed 's/^\W*for //'|grep -v '^--'|cut -d\  -f3-|sort|head -n 1
Sat, 17 Jan 2004 11:02:09 -0500
grep -RiA1 "by TARGET_HOSTNAME"  ../mqueue/spam_q | grep -v "by
TARGET_HOSTNAME "|sed 's/^\W*for //'|grep -v '^--'|cut -d\  -f3-|sort|tail -n 1
Sun, 18 Jan 2004 06:59:52 -0500
```

6134 Spam emails were received for relay before promiscuous relaying was revoked.

```
grep -lR "" ../mqueue/spam_d | grep -c ""
6134
```

Letters l and o were randomly replaced with the numbers 0 and 1. Sometimes a dollar sign was inserted. Note that 5853 + 281 = 6134

```
egrep -R " [1l]mpr[o0]vement "  ../mqueue/spam_d |egrep -v " improvement "|cut -d\: -f1|uniq|grep -c ""
5853
egrep -R " But y\\\$?[o0]u d\\\$?[o0]n't want t\\\$?[o0]"  ../mqueue/spam_d|grep -v " But you don't want to "|cut -d\: -f1|uniq|grep -c "
281
```

5853 spams like this were received

```
<html><body bgcolor=#FFFFFF text=#000000><p><b><font color=#FF0000>
duh_gig:<br>Supreme Formu1a HGH--------Take 20 Years off Y0ur Aqe</font></b><br>
<br></p><p><b>Lo0K AT THESE AMAZING TEST RESUITS!</b><br></p>
<p>£®£®£®Body Fat L0ss --------- 82% 1mprovement <br>
  £®£®¡£Wrink1e Reducti0n ---------- 61% lmprovement <br>
  £®¡£¡£Energy Level ----------- 84% 1mpr0vement <br>
  £®¡££®Musc1e Strength ------------ 88% improvement <br>
  ¡£¡£¡£Sexua1 Potency _________ 75% lmpr0vement <br>
  ¡£¡££®Emoti0na1 Stability _________ 67% 1mprovement <br>
  ¡££®£®Mem0ry __________ 62% improvement
  <p><a href=http://border.duh></a></p>
  <a href=http://popggg.com/on/>P1ease Vls1t 0ur Web Site: Click Here</a><br>
  <p><font color=#ffffff>acquiesce.duh</font></p>
</p>
</body></html>
```

The spam messages had 758 distinct sizes that ranged from 758 bytes to 1003 bytes.

```
ls -g ../mqueue/spam_d|sed -e 's/^.*smmsp *//' -e 's/ .*//'|sort|uniq -c|sort -nr|grep -c ''
758
ls -g ../mqueue/spam_d|sed -e 's/^.*smmsp *//' -e 's/ .*//'|sort|uniq -c|sort -nr|head -n 1
     47 783
ls -g ../mqueue/spam_d|sed -e 's/^.*smmsp *//' -e 's/ .*//'|sort|uniq -c|sort -nr|tail -n 1
      1 1003
```

Fig 2.3.3. 281 spams like this were received

```
<html><body bgcolor=#FFFFFF text=#000000>
<p><font color=#000099><b><font color=#FF0000><b>anderso : </b></font>Have you
been l0nging for a Designer Handbag,
  Wa1let, etc? But y0u don't want t0 spend the Big Bucks? We1l, then here
  is the p1ace to be - A place where y0ur dreams come true! Enj0y the Superior
  qua1ity and m0st affordable price and add a sty1e t0 your life!
  </b></font></p>
<p><font color=#FF0000><b>anderso:</b></font><br>
  <font color=#ff0000><b><a href=http://hot.4hoster.com/?11159901>P1ease Visit
  0ur Vebsite see more:Cllck-Here</a><font color=#000099>
<p>anderso<a href=http://WWW.anderso>.</a></p><p><a
href=http://WWW.anderso>.</a></p><p><a href=http://WWW.anderso>.</a></p><p><a
href=http://WWW.anderso>.</a></p><p><a href=http://WWW.anderso>.</a></p><p><a
href=http://WWW.anderso>.</a></p><p><a
href=http://WWW.anderso>.</a></p><p>anderso<a href=http://WWW.anderso>.</a></p>
<font color=#ff0000>
<p><a
href=http://hot.4hoster.com/lifehosteronline/remove>De1ete_____Delete</a></p>
</font>
</body></html>
```

Hostnames that sent the spam.

```
cmd  grep -R 'Received: from' ../mqueue/spam_q|sed 's/.*Received:/Received:/'|sort|uniq -c|sort -nr
    878 Received: from sauces ([61.171.77.135])
    561 Received: from scalars ([218.16.115.241])
    553 Received: from covariant ([61.51.176.203])
    447 Received: from bonds ([61.51.145.72])
    311 Received: from postmen ([218.18.214.43])
    288 Received: from popping ([211.161.44.133])
    155 Received: from plop (dsl-200-95-82-57.prod-infinitum.com.mx [200.95.82.57])
    155 Received: from brazenly ([211.158.119.74])
    149 Received: from housewife ([211.158.66.245])
    133 Received: from atlantica (dsl-200-95-14-101.prod-infinitum.com.mx [200.95.14.101])
  130 Received: from explorers ([218.2.176.23])
    128 Received: from technicality ([61.145.234.108])
      128 Received: from bach ([61.51.176.203])
    109 Received: from postorder ([211.158.71.77])
    107 Received: from practicably ([218.70.9.254])
    104 Received: from tallow (dsl-200-95-75-4.prod-infinitum.com.mx [200.95.75.4])
     93 Received: from evaporative (MG023212.user.veloxzone.com.br [200.165.23.212])
     88 Received: from bodyguard (RJ165128077.user.veloxzone.com.br [200.165.128.77] (may be forged))
     81 Received: from expirations (dup-200-95-124-48.prod-infinitum.com.mx [200.95.124.48])
     80 Received: from addison (ES222216001.user.veloxzone.com.br [200.222.216.1] (may be forged))
     77 Received: from thanking ([211.158.71.77])
     74 Received: from critics ([218.2.209.15])
     72 Received: from powersets ([211.158.71.77])
     70 Received: from berlioz ([218.58.27.156])
     68 Received: from screen ([211.161.189.231])
     64 Received: from scattergun (MG007076.user.veloxzone.com.br [200.165.7.76])
     60 Received: from plugboard ([218.109.170.251])
```

Hostnames that sent the spam.
59 Received: from agnew (dup-200-95-124-48.prod-infinitum.com.mx [200.95.124.48]) 57 Received: from sealer ([61.171.254.93]) 52 Received: from allah (dsl-200-95-7-214.prod-infinitum.com.mx [200.95.7.214]) 50 Received: from menhaden ([218.70.9.164]) 47 Received: from porter (dsl-200-95-57-231.prod-infinitum.com.mx [200.95.57.231]) 38 Received: from braggart ([211.161.44.133]) 34 Received: from populated (dsl-200-95-3-17.prod-infinitum.com.mx [200.95.3.17]) 32 Received: from expensively (dup-200-95-122-198.prod-infinitum.com.mx [200.95.122.198]) 30 Received: from metamathematical (dsl-200-95-75-246.prod-infinitum.com.mx [200.95.75.246]) 26 Received: from belmont ([218.16.136.128]) 25 Received: from telegraphing ([211.158.71.77]) 25 Received: from polloi ([61.49.221.14]) 24 Received: from memoryless (MG023212.user.veloxzone.com.br [200.165.23.212]) 24 Received: from hydrometer (175172.telemar.net.br [200.165.175.172] (may be forged)) 24 Received: from excepted (dsl-200-95-75-246.prod-infinitum.com.mx [200.95.75.246]) 24 Received: from addis (ES216117169.user.veloxzone.com.br [200.216.117.169] (may be forged)) 23 Received: from exquisitely ([218.58.27.156]) 22 Received: from seam ([61.171.235.43]) 22 Received: from microinstruction (dsl-200-95-77-32.prod-infinitum.com.mx [200.95.77.32]) 22 Received: from expenditures ([219.153.151.75]) 22 Received: from adherents ([211.158.66.245]) 20 Received: from hunted ([218.2.198.198]) 17 Received: from secedes (dsl-200-95-74-148.prod-infinitum.com.mx [200.95.74.148]) 16 Received: from tasting ([61.49.205.178]) 16 Received: from boasts ([211.161.189.231]) 15 Received: from menfolk (dup-200-95-127-184.prod-infinitum.com.mx [200.95.127.184]) 15 Received: from breakfasted (dup-200-95-122-198.prod-infinitum.com.mx [200.95.122.198]) 14 Received: from scrotum (dup-200-95-125-128.prod-infinitum.com.mx [200.95.125.128]) 14 Received: from mettlesome ([218.70.8.160]) 14 Received: from counties ([211.158.71.77]) 14 Received: from acquaints (dsl-200-95-96-239.prod-infinitum.com.mx [200.95.96.239]) 13 Received: from mate ([61.171.216.111]) 12 Received: from explainable ([218.13.27.44]) 11 Received: from teething (dsl-200-95-35-111.prod-infinitum.com.mx [200.95.35.111]) 9 Received: from poisonousness ([61.51.145.72]) 9 Received: from evaporated ([218.2.209.15]) 9 Received: from adjudicates ([218.58.40.188]) 8 Received: from meekness (dup-200-95-125-128.prod-infinitum.com.mx [200.95.125.128]) 8 Received: from hopscotch ([218.70.9.154]) 7 Received: from teacher ([218.16.81.149]) 7 Received: from brave (host-200.95.237.93-cust.telemedia.net.mx [200.95.237.93] (may be forged)) 7 Received: from achievable ([218.58.40.188]) 6 Received: from plodding ([61.171.216.111]) 6 Received: from existent ([61.51.145.72]) 5 Received: from tetrachloride (BA222221005.user.veloxzone.com.br [200.222.221.5] (may be forged)) 5 Received: from btl (dsl-200-95-7-18.prod-infinitum.com.mx [200.95.7.18]) 4 Received: from illustration ([218.70.11.245]) 4 Received: from exclusive (MG023212.user.veloxzone.com.br [200.165.23.212]) 2 Received: from polyhedron (RJ165128077.user.veloxzone.com.br [200.165.128.77] (may be forged)) 1 Received: from merry ([218.58.156.11])

The hosts that sent spam had 56 distinct IP's

```
grep -R 'Received: from' ../mqueue/spam_q|sed 's/.*Received:/Received:/'|sed -e
"s/\(^\)\(.*\)\[\(.*\)\]/\3/"\
  -e 's/(may be forged)//' -e 's/)//'|sort|uniq -c|sort -nr|grep -c "
56
```

```
grep -R 'Received: from' ../mqueue/spam_q|sed 's/.*Received:/Received:/'|sed -e
"s/\(^\)\(.*\)\[\(.*\)\]/\3/"\
 -e 's/(may be forged)//' -e 's/)//'|sort|uniq -c|sort -nr
```

878 61.171.77.135	84 211.161.189.231	22 200.95.125.128
681 61.51.176.203	83 218.2.209.15	20 218.2.198.198
561 218.16.115.241	80 200.222.216.1	19 61.171.216.111
462 61.51.145.72	64 200.165.7.76	17 200.95.74.148
326 211.161.44.133	60 218.109.170.251	16 61.49.205.178
311 218.18.214.43	57 61.171.254.93	16 218.58.40.188
297 211.158.71.77	54 200.95.75.246	15 200.95.127.184
171 211.158.66.245	52 200.95.7.214	14 218.70.8.160
155 211.158.119.74	50 218.70.9.164	14 200.95.96.239
155 200.95.82.57	47 200.95.57.231	12 218.13.27.44
140 200.95.124.48	47 200.95.122.198	11 200.95.35.111
133 200.95.14.101	34 200.95.3.17	8 218.70.9.154
130 218.2.176.23	26 218.16.136.128	7 218.16.81.149
128 61.145.234.108	25 61.49.221.14	7 200.95.237.93
121 200.165.23.212	24 200.216.117.169	5 200.95.7.18
107 218.70.9.254	24 200.165.175.172	5 200.222.221.5
104 200.95.75.4	22 61.171.235.43	4 218.70.11.245
93 218.58.27.156	22 219.153.151.75	1 218.58.156.11
90 200.165.128.77	22 200.95.77.32	

Here are IP addresses that were associated with multiple hostnames.

```
grep -R 'Received: from' ../mqueue/spam_q|sed 's/.*Received:/Received:/'|sort|uniq|sed \
-e 's/^.*\[//' -e 's/\].*$//'|sort |uniq -c|sort -nr|grep -v '^\W*1\W'
```

5 211.158.71.77	2 218.58.27.156	2 200.95.125.128
3 61.51.145.72	2 218.2.209.15	2 200.95.124.48
3 200.165.23.212	2 211.161.44.133	2 200.95.122.198
2 61.51.176.203	2 211.161.189.231	2 200.165.128.77
2 61.171.216.111	2 211.158.66.245	
2 218.58.40.188	2 200.95.75.246	

The IP that had five hostnames also used five ports for five different sessions.

```
tcpdump -Sttttvvvnn -r tcpdump_w_jan_17_2004_a.libcap|grep
'211\.158\.71\.77\..*> $TARGET_IP\.25:'|cut -d\  -f3|uniq|sort|uniq
211.158.71.77.1490
211.158.71.77.1910
211.158.71.77.2035
211.158.71.77.2831
211.158.71.77.4915
```

84 sessions did not terminate properly.

```
tcpdump -Sttttvvvnn -r tcpdump_w_jan_17_2004_a.libcap|grep -c ' > MY.HOST.IP\.25: S'
221
tcpdump -Sttttvvvnn -r tcpdump_w_jan_17_2004_a.libcap|grep -c ' > MY.HOST.IP\.25: F '
137
```

The target was in Canada but none of the attackers IP's belong to North America [3]

```
grep -R 'Received: from' ../mqueue/spam_q|sed 's/.*Received:/Received:/'|sed -e
"s/\(^\)\(.*\)\[\(.*\)\]/\3/"\
 -e 's/(may be forged)//' -e 's/)//'|sort|cut -d\. -f1|uniq -c|sort -nr
   2288 61
   1503 218
   1287 200
   1033 211
     22 219
```

```
cmd whois 61.0.0.0 @whois.arin.net | egrep "NetRange:|NetName:"
NetRange:   61.0.0.0 - 61.255.255.255
NetName:    APNIC3
cmd whois 218.0.0.0 @whois.arin.net | egrep "NetRange:|NetName:"
NetRange:   218.0.0.0 - 218.255.255.255
NetName:    APNIC4
cmd whois 200.0.0.0 @whois.arin.net | egrep "NetRange:|NetName:"
NetRange:   200.0.0.0 - 200.255.255.255
NetName:    LACNIC-200
cmd whois 211.0.0.0 @whois.arin.net | egrep "NetRange:|NetName:"
NetRange:   210.0.0.0 - 211.255.255.255
NetName:    APNIC-CIDR-BLK2
cmd whois 219.0.0.0 @whois.arin.net | egrep "NetRange:|NetName:"
NetRange:   219.0.0.0 - 219.255.255.255
NetName:    APNIC5
```

200.95.75.246.1794 and 211.161.44.133.1794 are the only hosts that used the same port. This is probably just a coincidence.

The other hosts used different source ports, making it hard to write a NIDS rule.

```
grep -R 'Received: from' ../mqueue/spam_q|sed
's/.*Received:/Received:/'|sort|uniq -c|sort -nr > 1.tmp
```

```
tcpdump -Sttttvvvnn -r tcpdump_w_jan_17_2004_a.libcap|grep ' >
$TARGET_IP\.25:'|cut -d\  -f3|uniq|sort|uniq  > 2.tmp
```

```
sed -e 's/^.*\[/grep /' -e 's/\].*$/ 2.tmp/' 1.tmp|sh|sort|uniq|cut -d\. -f5|sort|uniq -c|sort
-n|tail -n2
      1 58312
      2 1794
```

The full list of Source ports for the spam sessions has no obvious pattern

cmd sed -e 's/^.*\[/grep /' -e 's/\].*$/ 2.tmp/' 1.tmp|sh|sort|uniq		
200.165.128.77.3121	200.95.75.246.1713	218.2.209.15.4123
200.165.128.77.3585	200.95.75.246.1794	218.58.156.11.3751
200.165.175.172.2541	200.95.75.4.3185	218.58.27.156.4419
200.165.23.212.3858	200.95.75.4.4977	218.58.27.156.4730
200.165.23.212.3993	200.95.77.32.1722	218.58.40.188.2138
200.165.23.212.4242	200.95.82.57.3598	218.58.40.188.2809
200.165.7.76.1181	200.95.96.239.4475	218.70.11.245.4844
200.216.117.169.3744	211.158.119.74.2861	218.70.8.160.3360
200.222.216.1.2798	211.158.66.245.2704	218.70.9.154.3549
200.222.221.5.3306	211.158.66.245.2942	218.70.9.164.4106
200.95.122.198.1691	211.158.71.77.1490	218.70.9.254.4375
200.95.122.198.3425	211.158.71.77.1910	219.153.151.75.4228
200.95.124.48.2533	211.158.71.77.2035	61.145.234.108.3307
200.95.124.48.2904	211.158.71.77.2831	61.171.216.111.1881
200.95.124.48.4030	211.158.71.77.4915	61.171.216.111.3634
200.95.125.128.57046	211.161.189.231.1969	61.171.235.43.4509
200.95.125.128.58312	211.161.189.231.2239	61.171.254.93.33503
200.95.127.184.3484	211.161.44.133.1794	61.171.77.135.4496
200.95.14.101.2714	211.161.44.133.3748	61.49.205.178.3179
200.95.237.93.1137	218.109.170.251.2868	61.49.221.14.4331
200.95.3.17.3363	218.13.27.44.1311	61.49.221.14.4561
200.95.35.111.4877	218.16.115.241.2753	61.51.145.72.3450
200.95.57.231.4214	218.16.136.128.3660	61.51.145.72.3701
200.95.7.18.3992	218.16.81.149.2152	61.51.145.72.4108
200.95.7.214.1892	218.18.214.43.1959	61.51.145.72.4113
200.95.74.148.3900	218.2.176.23.4189	61.51.176.203.3792
200.95.75.246.1032	218.2.198.198.4213	61.51.176.203.4339
200.95.75.246.1454	218.2.209.15.3994	

It looks like the drone hosts are all MS Windows

```
p0f -s tcpdump_w_jan_17_2004_a.libcap  | sort | uniq -c | sort -n > pf0.c.txt
grep -R 'Received: from' ../mqueue/spam_q|sed -e 's/.*Received: from .*\[//' -e 's/\].*//' |sort -r|\
cut -d\.  -f1,2,3,4 |uniq|sed -e 's/^/grep /' -e 's/$/ pf0.c.txt/' |sh| grep grep -c ' Windows '
68
```

Possibly the trojan modified some TCP settings

```
grep -R 'Received: from' ../mqueue/spam_q|sed -e 's/.*Received: from .*\[//' -e 's/\].*//'\ |sort -r|cut -d\. -f1,2,3,4 |uniq|sed -e 's/^/grep /' -e 's/$/ pf0.c.txt/' |sh|grep  -v 'Windows'

61.49.205.178:3179 - UNKNOWN [16384:49:1:52:M1452,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
61.171.254.93:33503 - Linux 2.4 (late, uncommon) (up: 7 hrs) -> MY.NET.84.167:25 (distance 16, link: sometimes DSL(3
218.58.27.156:4419 - UNKNOWN [16384:47:1:52:M1440,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: IPv6/IPIP)
218.58.27.156:4730 - UNKNOWN [16384:47:1:52:M1440,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: IPv6/IPIP)
211.158.71.77:1490 - UNKNOWN [8192:47:0:48:M1452,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
211.158.71.77:1910 - UNKNOWN [8192:47:0:48:M1452,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
211.158.71.77:2035 - UNKNOWN [8192:47:0:48:M1452,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
211.158.71.77:2831 - UNKNOWN [8192:47:0:48:M1452,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
211.158.71.77:4915 - UNKNOWN [8192:47:0:48:M1452,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
211.158.66.245:2704 - UNKNOWN [S10:47:1:48:M1452,N,N,S:.:?:?] (NAT!) -> MY.NET.84.167:25 (link: pppoe (DSL))
211.158.66.245:2942 - UNKNOWN [S10:47:1:48:M1452,N,N,S:.:?:?] (NAT!) -> MY.NET.84.167:25 (link: pppoe (DSL))
200.95.75.246:1032 - UNKNOWN [32767:49:1:52:M1452,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
200.95.75.246:1454 - UNKNOWN [32767:49:1:52:M1452,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
200.95.75.246:1713 - UNKNOWN [32767:49:1:52:M1452,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
200.95.75.246:1794 - UNKNOWN [32767:49:1:52:M1452,N,W0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
200.95.7.214:1892 - UNKNOWN [S20:112:1:44:M1452:.:?:?] -> MY.NET.84.167:25 (link: pppoe (DSL))
200.222.221.5:3306 - UNKNOWN [992:17:1:64:M1414,N,W5,N,N,T0,N,N,S:.:?:?] -> MY.NET.84.167:25 (link: sometimes
```

The Spam URL's

The spam emails list 6078 different URL's

```
grep -iR "http:" ../mqueue/spam_d|cut -d\  -f3-|sed -e 's/^.*http:\/\///'|cut -d\>  -f1|sort|uniq -c|sort -nr|grep -c ""
6078
```

Nine hostnames often occurred.

```
egrep -iR "http:" ../mqueue/spam_d|cut -d\  -f3-|sed -e 's/^.*http:\/\///'|cut -d\>  -f1|\
cut -d\/ -f1|sort|grep -v '^$'|uniq -c|sort -nr|head
```

777 www.erd55x.com	726 fanddc.com	704 www.77yy4.com
749 creatorr.com	712 www.easy5544.com	280 hot.4hoster.com
739 purchze3.com	707 popggg.com	2 WWW.svivian
739 practizze.com		

Only four URL's could be resolved to IP addresses.

```
egrep -iR "http:" ../mqueue/spam_d|cut -d\  -f3-|sed -e 's/^.*http:\/\///'|cut -d\>  -f1|\
 cut -d\/ -f1|sort|grep -v '^$'|uniq -c|sort -nr|head|cut -d\/ -f1|sed 's/^/dig /'|sh|grep -A1
"ANSWER SECTION"|egrep -v "^--|;; ANSWER"
www.erd55x.com.         2233    IN      A       61.234.218.191
www.easy5544.com.       2349    IN      A       61.234.218.192
www.77yy4.com.          2349    IN      A       61.234.218.191
hot.4hoster.com.        2352    IN      A       211.162.148.227
```

All four resolvable hostnames are in China.[4]

```
whois  61.234.218.191 @WHOIS.APNIC.NET | grep  country | uniq
country:      CN
whois  61.234.218.192 @WHOIS.APNIC.NET | grep  country | uniq
country:      CN
whois  211.162.148.227 @WHOIS.APNIC.NET | grep  country | uniq
country:      CN
```

The Whois information about the four resolvable hostnames refers to China[5]

Domain Name:77yy4.com China tel: 86 311 5053513 guoru2n4s@yahoo.com.cn Primary DNS: ns0.nictxt.com Secondary DNS: ns1.nictxt.com	Domain Name: EASY5544.COM China tel: 86 311 5053513 guoru2n4s@yahoo.com.cn Primary DNS: ns0.nictxt.com Secondary DNS: ns1.nictxt.com
Domain Name:4hoster.com China tel: 86 755 8299369 aniu_3518318@hotmail.com Primary DNS: ns1.gslztx.com 211.162.148.227 Secondary DNS: ns2.gslztx.com	Domain Name:erd55x.com China tel: 86 431 7973627 mueer34@yahoo.com.cn Primary DNS: ns0.nictxt.com Secondary DNS: ns1.nictxt.com

Four attackers had snort alerts

```
sed 's/^.*$/grep "&" logs\/alert/'  5.tmp |sh|cut -d\  -f1-3|sed -e 's/:.* ->/ ->/'\
 -e 's/:...\?.\?.\?.\?$//' -e 's/$TARGET_IP//' -e 's/ -> //' |sort|uniq
```

200.95.14.101	211.161.44.133	61.171.77.135
211.158.119.74	219.153.151.210	

The alert for 200.95.14.101

```
grep -B3 '200.95.14.101' logs/alert | grep '^\[\*'
[**] [1:1432:4] P2P GNUTella GET [**]
```

The alert for 211.158.119.74

```
[**] [1:407:4] ICMP Destination Unreachable (Undefined Code!) [**]
[Classification: Misc activity] [Priority: 3]
01/17-23:07:29.052317 0:20:CD:2:C3:D8 -> 0:48:54:65:FD:91 type:0x800 len:0x46
211.158.29.49 -> $TARGET_IP ICMP TTL:238 TOS:0x0 ID:33618 IpLen:20
DgmLen:56
Type:3  Code:13  DESTINATION UNREACHABLE: ADMINISTRATIVELY
PROHIBITED,
PACKET FILTERED
** ORIGINAL DATAGRAM DUMP:
$TARGET_IP:25 -> 211.158.119.74:2861 TCP TTL:37 TOS:0x0 ID:62499 IpLen:20
DgmLen:93 DF
Seq: 0x627CE71B  Ack: 0x6F353561
** END OF DUMP
```

The alert for 211.161.44.133

```
grep -B3 '211.161.44.133' logs/alert | grep '^\[\*'
[**] [1:1432:4] P2P GNUTella GET [**]
```

878 emails came from 61.171.77.135 here are the alerts. Maybe it got overloaded.

```
grep -B3 '61.171.77.135' logs/alert | grep '^\[\*' |sort|uniq -c
      1 [**] [1:1432:4] P2P GNUTella GET [**]
      9 [**] [1:402:4] ICMP Destination Unreachable (Port Unreachable) [**]
```

The alert for 219.153.151.210

```
grep -B3 '219.153.151.210' logs/alert | grep '^\[\*'
41 [**] [1:567:9] POLICY SMTP relaying denied [**]
```

61 relay attempts were made after relaying was revoked

```
cmd grep  ' reject=550' /var/log/maillog* |grep -c ''
61
```

The maillog shows that 51 hosts tried to relay email after it was revoked.

```
grep  ' reject=550' /var/log/maillog* |cut -d\  -f9-12|cut -d: -f2-|sort | uniq -c|grep -c ''
51
```

```
grep  ' reject=550' /var/log/maillog* |cut -d\  -f9-12|cut -d: -f2-|sort | uniq -c
      1  2@aol.com>, relay=[61.149.167.144], reject=550
      1 relay=200-171-100-222.dsl.telesp.net.br [200.171.100.222], reject=550 5.7.1
      5 relay=[211.158.77.186], reject=550 5.7.1 <ameill@xinhuanet.com>...
      2 relay=[211.158.77.186], reject=550 5.7.1 <pufanab@yahoo.com.cn>...
      1 relay=[211.161.45.142], reject=550 5.7.1 <newz@petfinder.demon.ru>...
      1 relay=[211.162.42.142], reject=550 5.7.1 <smuel587@yahoo.com>...
      1 relay=[218.13.192.30], reject=550 5.7.1 <tdavis8145@thedorm.com>...
      1 relay=[218.16.82.134], reject=550 5.7.1 <bbirds4238@yahoo.com>...
      1 relay=[218.18.154.40], reject=550 5.7.1 <petec1978@yahoo.com>...
      1 relay=[218.2.176.23], reject=550 5.7.1 <mrbee@goplay.com>...
      1 relay=[218.58.37.41], reject=550 5.7.1 <kenaero@flash.net>...
      1 relay=[218.58.97.91], reject=550 5.7.1 <drtritsch@beachlink.com>...
      1 relay=[218.59.194.82], reject=550 5.7.1 <frogsprkle@aol.com>...
      1 relay=[218.70.10.138], reject=550 5.7.1 <rix@hiwaay.net>...
      1 relay=[218.70.10.61], reject=550 5.7.1 <globetvl@general.net>...
      1 relay=[218.70.11.144], reject=550 5.7.1 <pufanab@yahoo.com.cn>...
      1 relay=[218.70.11.182], reject=550 5.7.1 <rizden@netdoor.com>...
      1 relay=[218.70.11.24], reject=550 5.7.1 <furnsdfriy@geocities.com>...
      1 relay=[218.70.136.83], reject=550 5.7.1 <cretney@dzine.co.za>...
      1 relay=[218.70.8.107], reject=550 5.7.1 <gmcleod@idt.net>...
      1 relay=[218.70.8.234], reject=550 5.7.1 <reception@pascalls.co.uk>...
      1 relay=[218.70.8.85], reject=550 5.7.1 <patree@kornet.net>...
      1 relay=[218.72.46.209], reject=550 5.7.1 <athor63044@aol.com>...
      1 relay=[218.88.85.177], reject=550 5.7.1 <kmack@usa.net>...
      2 relay=[219.153.150.64], reject=550 5.7.1 <ameill@xinhuanet.com>...
      1 relay=[219.153.151.109], reject=550 5.7.1 <ameill@xinhuanet.com>...
      1 relay=[219.153.151.210], reject=550 5.7.1 <ameill1@pufan.com>...
      2 relay=[219.153.151.210], reject=550 5.7.1 <ameill2@19.cn>...
```

```
1 relay=[219.153.151.210], reject=550 5.7.1 <ameill@xinhuanet.com>...
1 relay=[219.153.153.57], reject=550 5.7.1 <adwp@ig.com.br>...
1 relay=[219.153.154.68], reject=550 5.7.1 <ameill1@pufan.com>...
2 relay=[219.153.154.68], reject=550 5.7.1 <ameill@xinhuanet.com>...
2 relay=[219.153.154.68], reject=550 5.7.1 <pufan1@tom.com>...
1 relay=[219.153.156.39], reject=550 5.7.1 <patmar@adelphia.net>...
1 relay=[219.153.156.65], reject=550 5.7.1 <rase@mail.com>...
1 relay=[61.144.102.227], reject=550 5.7.1 <roymat@mesoscale.meteo.mcgill.ca>...
1 relay=[61.144.129.214], reject=550 5.7.1 <wcace44859@msn.com>...
1 relay=[61.145.88.75], reject=550 5.7.1 <gayboy@telstra.easymail.com.au>...
1 relay=[61.149.201.189], reject=550 5.7.1 <sales@emotioncommunications.com>...
1 relay=[61.171.27.106], reject=550 5.7.1 <eaglesoul@attglobal.net>...
1 relay=[61.49.179.169], reject=550 5.7.1 <cristi1374@aol.com>...
1 relay=[61.51.109.209], reject=550 5.7.1 <drexas@usa.net>...
1 relay=[61.51.109.209], reject=550 5.7.1 <lo439894@yahoo.com>...
1 relay=[61.51.153.109], reject=550 5.7.1 <djamama@att.net>...
1 relay=[61.51.163.198], reject=550 5.7.1 <amyrobben@flashmail.com>...
1 relay=[61.51.163.198], reject=550 5.7.1 <jclark6647@msn.com>...
1 relay=BA199063049.user.veloxzone.com.br [200.199.63.49] (may be
1 relay=ES152027.user.veloxzone.com.br [200.149.152.27] (may be
2 relay=ppp173-26.pppoe.mtu-net.ru [81.195.173.26], reject=550 5.7.1
1 relay=RJ165159092.user.veloxzone.com.br [200.165.159.92] (may be
1 relay=RJ195225.user.veloxzone.com.br [200.165.195.225], reject=550 5.7.1
```

Attack Mechanism

This attack used several layers of defence. The target was in Canada, the attackers in Mexico or China so it is not feasible to examine the systems. Another layer of defence was that over 100 distinct IP's were detected and it is almost certain that they were all trojaned drones. The port number is very important in analyzing an attack, but a random source port was used. It looks like the drones all had MS Windows but I was not able to find out anything about the trojan. The systems that were controlling the drones may be in another country such as Russia, or even Canada.

The emails had several URL's that included registered hostnames. It is likely that the registration information is false, except for the email addresses. They were yahoo mail or hotmail so it could be hard to trace the real user.

To avoid spam filters, there were many random differences in the emails. Zero was used for O, one was used for the letter "l".

Correlations:
Some of the web sites are mentioned on news groups such as news.admin.net-abuse.sightings. [6]
John Bokma has reported that guoru2n4s@yahoo.com.cn is associated with other domain names that were used with spam.[7] A Google for mueer34@yahoo.com.cn and aniu_3518318@hotmail.com found only that both were associated with spam.[8]
There is a candidate CVE, CAN-1999-0512, for "A mail server is explicitly configured to allow SMTP mail relay, which allows abuse by spammers"[9]

Evidence of active targeting: 6000 relay requests were made to the target in 24 hours

Severity:
severity=(Criticality+Lethality)-(System Countermeasures+Network Countermeasures) = (5 + 5) - (3 + 2) = 5
criticality 5 Mail server is a critical target
lethality 5 relaying 6000 spams would cause trouble elsewhere and get the target blacklisted
network countermeasures 2 traffic is allowed to target
system countermeasures 3 target did not relay anything, but normal email could not be sent. (4)

Defensive recommendation:

This attack shows some expertise. There were over 100 different IP addresses. None of the drone IP's belong on the same continent as the target so police action is not feasible. The host that was controlling them was not discovered The URL's in the spam points to web servers in China. There is nothing legally that can be done about them.

The trojan used to control the drones was not identified however it is probably in all the latest AV updates. Many computer owners do not realize how important this is.

The target system had only one mail queue. Normal email could not sent without spam being sent as well. That is why promiscuous relaying was disabled after 6000 spams. Two mail queues would have allowed normal usage while the attack was in progress and allowed it to continue. Having more data collected would increase the chance of finding out who was behind it.

If the analysis procedure was automated then the results could have been reported during the attack.

This scam involved quite a lot of work and none of the Spam was relayed so it was not a success. If many other mail servers were set-up to accept mail for relay but not send it, then more Spam would not get into your inbox. This would make it less profitable for the spamers.

It is likely that the whois entries are false and the instigator will remain unknown. There is no reason for him not to do it again. As Mike T said "The only way to stop spam from chinanet.net is to go to China and smash their servers."[10]

There are several political issues here. Tracking down and stopping scams like this would require international cooperation. This scam obviously was created by one or more people who are very technically knowledgeable. However the salary they could

legally get in China is probably very low. If they have no legal way to earn a decent living, then they will probably keep scamming. There was a lot of deception in this attack so it is very possible that it did not originate in China.

Multiple choice test question:

```
Return-Path: <sprindle_1@charter.net>
Received: from mxsf05.cluster1.charter.net (mxsf05.cluster1.charter.net
[209.225.28.205])
        by TARGET.HOSTNAME (8.12.8/8.12.8) with ESMTP id hBNIem0d002282
        for <YOU@TARGET.HOSTNAME>; Tue, 23 Dec 2003 13:40:49 -0500
Received: from dxpz (cpe-24-176-85-084.hky.nc.charter.com [24.176.85.84])
        by mxsf05.cluster1.charter.net (8.12.10/8.12.8) with SMTP id hBNIHlnm040431;
        Tue, 23 Dec 2003 13:17:47 -0500 (EST)
        (envelope-from sprindle_1@charter.net)
Date: Tue, 23 Dec 2003 13:17:47 -0500 (EST)
Message-Id: <200312231817.hBNIHlnm040431@mxsf05.cluster1.charter.net>
FROM: "Mail System" <emailform@puremail.net>
TO: "email recipient" <client@emaildomain.com>
SUBJECT: undeliverable message returned to mailer
Mime-Version: 1.0
Content-Type: multipart/alternative;
  boundary="rqxrltqjc"
Status: RO
X-Status: O
```

Which important piece of information in the spam header above cannot be forged?

A mxsf05.cluster1.charter.net
B [209.225.28.205]
C [24.176.85.84]
D cpe-24-176-85-084.hky.nc.charter.com
E They all can be forged

Answer: B The host IP that made the connection to your server cannot be spoofed.

References

[1] Spencer, Brad. Fighting Relay Spam the Honeypot Way. 13 Jun 2002. URL: http://fightrelayspam.homestead.com/files/antispam06132002.htm (17 Mar 2004).

[2] URL: http://www.snort.org (20 March 2004).

[3] American Registry for Internet Numbers ARIN WHOIS Database Search. URL: http://ww2.arin.net/whois/ (20 March 2004).

[4] Asia Pacific Network Information Centre. URL: http://www.apnic.net/

[5] whois.crsnic.net (20 January 2004).

[6] news.admin.net-abuse.sightings. Google Search: news.admin.net-abuse.sightings. http://groups.google.ca/groups?hl=en&lr=&ie=UTF-8&group=news.admin.net-abuse.sightings (22 March 2004).

[7] bokma john. Zheng Zhou. 12 Feb 2004. http://www.johnbokma.com/spam/zhengzhou.html (20 March 2004).

[8] http://www.google.ca (20 March 2004).

[9] CAN-1999-0512 (under review). 27 April 2002 04. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0512 (20 March 2004).

[10] DJ Mike T, Online Scams. http://www.geocities.com/danceawaysb/spam/scams.html (20 March 2004).

Analyze This:

Source of Trace : downloaded from http://www.incidents.org/logs.

Log files analyzed.

alert.040316	oos_report_040312	scans.040316.gz
alert.040317	oos_report_040313	scans.040317.gz
alert.040318	oos_report_040314	scans.040318.gz
alert.040319	oos_report_040315	scans.040319.gz
alert.040320	oos_report_040316	scans.040320.gzq

Executive Summary

There were 53822 alerts and 8228204 scans. The size created a problem when analyzing the data, but even more logging is needed.

There appears to be some kind of spam attack from Texas but some critical information was not logged. A good project for your forensic students would be to look into filling charges under the new CAN-SPAM law. [1] They should start by looking into a case where a company recently sued the University of Texas because they blocked unsolicited emails. [2] This company may be connected to this spam attack. [3]

There is a lot of P2P going on. Pete Storm pointed out that this could leave the organization "open to legal action if copyrighted material is being offered" and he recommended that they "control it" [4]. However if the controls were not enforced properly, this would create another liability. Also there is the $10 monthly fee per student that the RIAA wants. [5]

Anonymous FTP is OK, but an alert is still needed. The reason is that there are several FTP alerts and if they are not for Anonymous FTP then they are serious. Several alerts for trojan server activity look like they are actually part of a FTP session. It would nice to confirm this.

The TCP:69 activity could caused by some student project or it could be a trojan. Start by contacting the owner of MY.NET.84.203.

There is a misconfiguration with the primary name server, MY.NET.1.3. It looks like a minor issue but needs to be looked into.

The Information Technology Security: Policies & Guidelines [6] needs to be clarified. When it says, "Intentionally corrupt, misuse" does that include portscanning? An example of a well-written policy is the GIAC "POLICY ON ACTIVE SCANNING AND ANALYSIS." [7] Some students were experimenting with spoofing on the Internet. This should be done on an isolated LAN or the police may get involved. A clear policy on this could prevent a lot of trouble.

I do not recommend reducing the number of alerts, as Pete Storm recommends. That would increase the chance of missing something important. A better way to analyze all the data needs to be found.

The hosts that need attention are, MY.NET.1.3, MY.NET.84.203 MY.NET.190.92, MY.NET.190.93, MY.NET.190.95, MY.NET.190.97, MY.NET.190.102 MY.NET.75.13 MY.NET.97.74 MY.NET.27.232 and MY.NET.12.6. The details are described later.

Defensive recommendations

High risk Ports such as TFTP, Windows networking and printing, should be blocked.

Add alerts for port 99,Metagram and port 6129, Dameware Mini Remote Control.

Consider the IPS discussed in "Describe the State of Intrusion Detection". It could "tie up" an internal PC sending malicious email until it gets fixed.

"Dynamic IDS" systems have a bad reputation, but one could have alerted when 10% of the DNS load went to four new IP address. Then all communication to that subnet could have been captured and analyzed.

Top 20 Internal Scan Sources

IP	Total Scans	Dest Hosts	Dest Port	Destination Service	Count	Dest Hosts	Source Port	Source Service	Count	Dest Hosts
MY.NET.1.3	2888928	101604	53	domain	2878168	101541				
			123	ntp	8428	41				
							32783	cachefsd RPC	2877959	101539
							123	ntp	10760	
MY.NET.190.92	216915	460750	135	epmap	1084593	414057				
			445	microsoft-ds	1080950	413077				
			5000	commplex-mai	1502	1151				
MY.NET.66.17	1003743	463389	445	microsoft-ds	501862	337549				
			135	epmap	500531	338074				
MY.NET.1.4	297114	30894	53	domain	293774	30868				
			123	ntp	2874	22				
							32788	client src	293741	30868
							123	ntp	3340	
MY.NET.153.174	231851	231847	135	epmap	231833	231833				
MY.NET.110.72	185251	9185					8767	TeamSpeak	134156	43
							12203	Medal Of H	37365	52
							12300	unknown	13515	9126
MY.NET.153.30	93229	20683								
MY.NET.82.15	87728	4103	27900	EA-Battle	1948	5				
							8888	ddi-udp-1	78349	261
							8889	ddi-udp-2	7605	3928
MY.NET.34.14	87009	1390	25	smtp	85707	572				
MY.NET.98.24	60751	27756	22321	wnn6_Tw	59039	27332				
							22321	Wnn6-Tw	60707	27744
MY.NET.18.27	43799	33368	445	microsoft-ds	43794	33363				
MY.NET.80.224	31804	31802	135	epmap	31796	31796				
MY.NET.111.34	31514	19002	4672	rfa	15693	10003				
			4673	eMule	5875	3624				
			4662	edonkey	1496	939				

IP	Total Scans	Dest Hosts	Dest Port	Destination Service	Count	Dest Hosts	Source Port	Source Service	Count	Dest Hosts
							4672	rfa	28568	17899
MY.NET.70.207	22539	3811					12203	Medal Of H	15985	
							12300		6502	
MY.NET.81.77	19572	5614	6346	gnutella-svc	5585	1259				
							24621	unknown	4152	2442
MY.NET.25.71	17758	3197	25	smtp	13437	828				
			113	auth	4321	2395				
MY.NET.97.24	16746	12708	80	http	16745	12707				
MY.NET.25.67	15647	2147	25	smtp	13007	628				
			113	auth	2640	1534				
MY.NET.69.209	15579	8415	8402	abarsd	15573	8413				
MY.NET.153.76	13366	5217					2797	esp-encap	12481	4594
MY.NET.97.105	12624	5879	6346	gnutella-svc	7065	3452				
			6348	Comobi	2230	1050				
MY.NET.84.203	12476	2467	4662	edonkey	6173	1627				
							4672	rfa	3445	1355
			4672	rfa	2829	1100				
MY.NET.25.69	12228	2878	25	smtp	8607	793				
			113	auth	3621	2105				
MY.NET.25.73	8004	1594	25	smtp	6286	509				
			113	auth	1718	1094				
MY.NET.25.70	7802	1814	25	smtp	5690	503				
			113	auth	2112	1324				
MY.NET.81.59	6978	4897	445	microsoft-ds	4864	3162				
			135	epmap	2111	1734				
MY.NET.153.80	6673	3164	6881	Bittorrent	1949	854				
MY.NET.98.68	6312	3473	41170	P2P Blubster	6183	3444				
MY.NET.153.76	13366	5217					1431	rgtp	6183	3444
MY.NET.42.2	6045	1782	6346	gnutella-svc	1492	313				
MY.NET.97.226	5801	3812	22321	wnn6_Tw	3759	3102				
MY.NET.153.76	13366	5217					22321	wnn6_Tw	3915	3224
							6545	unknown	934	517
							6112	dtspcd	873	52
MY.NET.84.235	5780	1490	4662	edonkey	3745	960				
MY.NET.25.68	4193	846	25	smtp	3417	328				
MY.NET.112.186	4155	2747					5082	Sip Phone	3329	2324
MY.NET.97.178	4114	3494	80	http	4114	3494				
MY.NET.97.239	3957	565	6112	dtspcd	3136	473				
							6112	dtspcd	3956	564
MY.NET.25.66	3794	842	25	smtp	2990	277				
MY.NET.42.1	3387	599	6346	gnutella-svc	1339	101				
MY.NET.42.4	3342	1028	6346	gnutella-svc	891	219				
MY.NET.97.82	2861	1941	6346	gnutella-svc	2614	1795				
MY.NET.82.86	2304	1711	6346	gnutella-svc	1241	936				
MY.NET.97.231	1869	1008	41170	P2P Blubster	1849	1003				
							1431	rgtp	1849	1003
MY.NET.153.91	1803	1340	4662	edonkey	919	694				
MY.NET.153.76	13366	5217					4672	rfa	166	166
			4672	rfa	136	136				
MY.NET.84.216	1735	839	6257	WinMX	1568	761				
							6257	WinMX	1692	822

List of Alerts

Alert	Total	Src Ext	Sest Int	Src Int	Dest Ext	In	Out	Int	Ext
MY.NET.30.4 activity	29864	279	1			29864			
MY.NET.30.3 activity	10934	171	1			10934			
SMB Name Wildcard	3493			172	416		3493		
EXPLOIT x86 NOOP	3243	315	327			3243			
OOS	2896	465	65	5	7	2854	7	35	
Null scan!	1701	93	61			1701			
High port 65535 tcp - possible Red Worm	1305	62	34	29	80	482	823		
NMAP TCP ping!	694	178	62			694			
High port 65535 udp - possible Red Worm	525	68	18	9	26	352	173		
Possible trojan server activity	418	35	19	20	37	197	221		
Incomplete Packet Fragments Discarded	250	84	64	1	1	249	1		
External RPC call	153	1	119			153			
<UMBC NIDS IRC Alert> IRC user /kill detected	147	48	48			147			
IRC evil - running XDCC	144			6	6		144		
TCP SRC and DST outside network	133	33			52				133
SUNRPC highport access!	123	20	25			123			
TFTP-Internal TCP connection to external svr	102	1	1	1	1	46	56		
SMB C access	99	26	5			99			
ICMP SRC and DST outside network	81	21			40				81
<UMBC NIDS> External MiMail alert	58	17	1			58			
<UMBC NIDS IRC Alert> Possible Incoming XDCC Send Request	39	6	5			39			
FTP passwd attempt	39	38	1			39			
EXPLOIT x86 setuid 0	32	30	23			32			
<UMBC NIDS IRC Alert> Possible sdbot floodnet detected attempting to IRC	31			8	1		31		
TCP SMTP Source Port traffic	24	3	2			24			
EXPLOIT x86 setgid 0	22	22	20			22			
TFTP - Internal UDP connection to external svr	16	3	2	1	2	9	7		
TFTP - External TCP connection to internal svr	13	2	4	4	2	7	6		
RFB - Possible WinVNC - 010708-1	12	3	3	3	3	5	7		
Tiny Fragments - Possible Hostile Activity	12	4	4			12			
EXPLOIT NTPDX buffer overflow	11	4	5			11			
<UMBC NIDS IRC Alert> XDCC client detected attempting to IRC	10			4	6		10		
SYN-FIN scan!	10	6	6			10			
connect to 515 from inside	10			2	2		10		
Probable NMAP fingerprint attempt	7	5	5			7			
External FTP to HelpDesk MY.NET.53.29	6	3	1			6			
EXPLOIT x86 NOPS	6	2	5			6			
EXPLOIT x86 stealth noop	6	6	6			6			
FTP DoS ftpd globbing	4	2	1			4			
External FTP to HelpDesk MY.NET.70.49	3	3	1			3			
External FTP to HelpDesk MY.NET.70.50	3	3	1			3			
<UMBC NIDS IRC Alert> User joining XDCC channel detected. Possible XDCC bot	2	2	2			2			
DDOS shaft client to handler	1	1	1			1			
Attempted Sun RPC high port access	1	1	1			1			
NIMDA - Attempt to execute cmd from campus host	1			1	1		1		
TFTP - External UDP connection to internal srv	1	1	1			1			

Top 20 External Scan Sources

IP	Total	Total	Dest Scans	Service Hosts	count Port		
61.129.45.60	34062	15451	80	http	28936	Hosts	Hosts that responded With
			99	metagram	4513		"SMB Name Wildcard"
64.174.25.65	27282	15511	4000	terabase	27282	15449	MY.NET.150.198 MY.NET.150.44
217.219.124.3	22019	14335	20168	virus/trojan	22019	142	
207.6.223.108	21540	13875	4899	radmin-port	21540	15511	MY.NET.150.198 MY.NET.150.44
212.244.160.35	21076	13877	6129	dameware	21076	14335	MY.NET.150.198 MY.NET.150.44
193.225.21.225	19612	13439	6129	dameware	19612	13875	MY.NET.150.198 MY.NET.150.44
195.197.143.215	18140	12869	6129	dameware	18140	13877	MY.NET.150.198 MY.NET.150.44 .109.86
213.165.186.246	18027	12648	6129	dameware	18027	13439	MY.NET.150.198 MY.NET.150.44
24.74.156.88	16583	10810	3410	networklenss	9790	12869	
			4000	terabase	6793	12648	MY.NET.150.198 MY.NET.150.44 .109.86
81.218.51.117	13821	10690	20168	virus/trojan	13821	8078	MY.NET.150.44
143.229.22.29	12356	10119	80	http	12356	5774	
62.179.203.99	12339	10028	1257	shockwave2	12339	10690	
221.147.75.247	12165	9675	6129	dameware	12165	10119	MY.NET.150.44
211.217.116.48	11240	9087	6129	dameware	11240	10028	
217.227.66.107	11112	9477	80	http	11112	9675	MY.NET.150.198 MY.NET.150.44 .109.86
192.117.165.62	10702	8361	4899	radmin-port	10702	9087	MY.NET.150.44 MY.NET.109.86
66.210.242.18	10560	8482	21	ftp	10560	9477	
213.9.173.98	10308	8324	80	http	10308	8361	MY.NET.150.44
193.6.41.157	10306	8172	6129	dameware	10306	8482	MY.NET.150.198 MY.NET.150.44
202.178.129.22	9955	5637	443	https	9955	8324	MY.NET.109.86
						8172	MY.NET.150.198 MY.NET.150.44

Top 10 incoming scans grouped by destination IP and destination port

Destination	Destination Port	Service	Count	Sources	
MY.NET.12.6	25	smtp	2730	460	possible student mailserver
MY.NET.6.7	110	pop3	2224	1	possible Student mailserver
MY.NET.81.77	24621	unknown	1226	306	port 24621 can be Gnutella
MY.NET.24.47	21	ftp	1052	15	Probable FTP server
MY.NET.24.44	80	http	373	120	Probable WWW serve
MY.NET.153.76	0	private	364	1	
MY.NET.70.164	4662	edonkey	261	9	P2P
MY.NET.153.80	0	private	245	2	
MY.NET.12.4	110	pop3	158	3	possible student mail server

Proposed Alert 69.6.57.0/24

Total	Source External	Dest Internal	Source Internal	Dest External	Incoming	Outgoing
31133	1	1	12	19	1	311076

There should be an alert for any traffic to or from 69.6.57.0/24. One in ten scans from the primary name server, MY.NET.1.3, had a destination in 69.6.57.0/24 .

Top scan destinations for nameserver MY.NET.1.3

Source	Destination	Scans		Source	Destination	Scans
MY.NET.1.3	69.6.57.9	74659		MY.NET.1.3	69.6.57.7	73952
MY.NET.1.3	69.6.57.8	74569		MY.NET.1.3	Other (101541 hosts)	2580792
MY.NET.1.3	69.6.57.10	74196				

IP's from 69.6.57.x/24 that have been reported as attackers to Dshield.org

Results of search at http://www.dshield.org	Earliest date
(69.6.57.7) appears as an attacker 394 times in the DShield database.	2004-03-30.
(69.6.57.9) appears as an attacker 374 times in the DShield database.	3/30/2004

The mail servers should be checked to make sure they are protected against spam from 69.6.57.0/24. 69.6.57.4 was the spam mail server.

List of all scans to 69.6.57.0/24

Source	Destination	Dest Port	Count
MY.NET.1.3	69.6.57.9	53	74659
MY.NET.1.3	69.6.57.8	53	74569
MY.NET.1.3	69.6.57.10	53	74196
MY.NET.1.3	69.6.57.7	53	73952
MY.NET.1.4	69.6.57.8	53	994
MY.NET.1.4	69.6.57.7	53	965
MY.NET.1.4	69.6.57.9	53	948
MY.NET.1.4	69.6.57.10	53	936
MY.NET.25.66	69.6.57.4	25	356
MY.NET.25.67	69.6.57.4	25	3527

Source	Destination	Dest Port	Count
MY.NET.25.68	69.6.57.4	25	321
MY.NET.25.69	69.6.57.4	25	1910
MY.NET.25.70	69.6.57.4	25	828
MY.NET.25.70	69.6.57.72	113	1
MY.NET.25.71	69.6.57.219	113	1
MY.NET.25.71	69.6.57.4	25	1602
MY.NET.25.73	69.6.57.4	25	1271
MY.NET.34.14	69.6.57.160	113	1
MY.NET.34.5	69.6.57.4	25	39

The alerts from MY.NET.25.x were all "false alarms". 65535 is a standard ephemeral port. The alerts from MY.NET.75.13 could be due to an unrelated virus.

Source	Src port	Destination	Dest Port	Alert
MY.NET.25.66	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.25.67	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.25.68	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.25.69	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.25.70	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.25.71	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.25.73	65535	69.6.57.4	25	High port 65535 tcp - possible Red Worm - traffic
MY.NET.75.13	137	69.6.57.125	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.127	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.141	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.185	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.190	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.193	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.220	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.232	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.237	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.249	137	SMB Name Wildcard
MY.NET.75.13	137	69.6.57.25	137	SMB Name Wildcard

It looks like some spam could have got through the internal mail servers.

Internal mailservers that have been reported as attackers to Dshield.org [8]

Results of search at http://www.dshield.org	Earliest date
MY.NET.25.66) appears as an attacker 48 times in the DShield database.	2004-03-30.
MY.NET.25.67) appears as an attacker 39 times in the DShield database.	2004-03-30.
MY.NET.25.68) appears as an attacker 34 times in the DShield database.	2004-03-30.
MY.NET 25.69) appears as an attacker 67 times in the DShield database	2004-03-30.
MY.NET.25.70) appears as an attacker 93 times in the DShield database.	2004-03-30.
MY.NET.25.71) appears as an attacker 96 times in the DShield database.	2004-03-30.
MY.NET.25.73) appears as an attacker 66 times in the DShield database.	2004-03-30.

Registration information

69.6.57.4 is associated with , 7349.slipperywhendrunkmarketing.com [9]
3219.flyingcoffeebeanmarketing.com and 5855.fatalfantasymail.com [10]

whois hittheinboxharder.com , whois slipperywhendrunkmarketing.com, whois alwaysclickingonemails.com and whois alwaysclickingonemails.com. all returned this

```
status:       hold,invalid-address
domain:       alwaysclickingonemails.com
status:       hold,invalid-address
organization: Brilliant Marketing, Inc.
owner:        Matthew Scholl
email:        brilliantmarketing2000@yahoo.com
title:        Abuse Manager
address:      PO BOX 2207
city:         Austin
state:        Texas
postal-code:  78768-220
country:      US
admin-c:      brilliantmarketing2000@yahoo.com#0
tech-c:       brilliantmarketing2000@yahoo.com#0
billing-c:    brilliantmarketing2000@yahoo.com#0
nserver:      a.ns.alwaysclickingonemails.com 69.6.57.7
nserver:      b.ns.alwaysclickingonemails.com 69.6.57.8
nserver:      c.ns.alwaysclickingonemails.com 69.6.57.9
nserver:      d.ns.alwaysclickingonemails.com 69.6.57.10
```

Since they all have status "hold,invalid-address", dig <hostname> returns no IP. However dig <hostname> @a.ns.alwaysclickingonemails.com returns 69.6.57.5. We see below that four numbers for the sub domain returns 69.6.57.4.

flyingcoffeebeanmarketing.com. 900 IN A 69.6.57.5
slipperywhendrunkmarketing.com.. 900 IN A 69.6.57.5
alwaysclickingonemails.com. 900 IN A 69.6.57.5
fatalfantasymail.com. 900 IN A 69.6.57.5
3219.flyingcoffeebeanmarketing.com. 900 IN A 69.6.57.4
7349.slipperywhendrunkmarketing.com. 900 IN A 69.6.57.4
4321.alwaysclickingonemails.com. 900 IN A 69.6.57.4
5855.fatalfantasymail.com. 900 IN A 69.6.57.4

Below is the Texas Corporation Search Results for Brilliant Marketing [11]

BRILLIANT MARKETING INC
5919 GREENVILLE AVE # 140
DALLAS, TX 75206-1906

Status: IN GOOD STANDING NOT FOR DISSOLUTION OR WITHDRAWAL through April 7, 2005

Registered Agent: AMY DUNCAN 5919 GREENVILLE #140 DALLAS, TX 75206
Registered Agent Resignation Date:

State of Incorporation: TX
-File Number: 0800289371
Charter/COA Date: January 8, 2004
Charter/COA Type: Charter
Taxpayer Number: 32014134897

DIRECTOR
MARK SCARDINO
23404 W. LYONS AVE., #223
NEWHALL , CA 91321

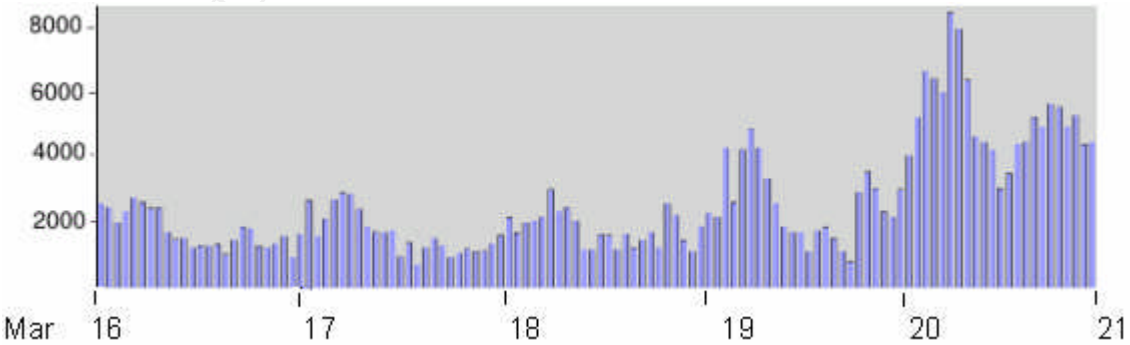
The director information has recently changed. On march 26 2004 David Ramalho posted [12]

Officers and Directors: SETH POULOS (Director)
1808B FAIRLAWN LANE
AUSTIN , TX 78704

Mail Boxes Etc. Store #3699
5919 GREENVILLE,
DALLAS, 75206 Texas, United States
NEAR SMU - GREENVILLE @ SOUTHWESTERN

Alerts+Scans

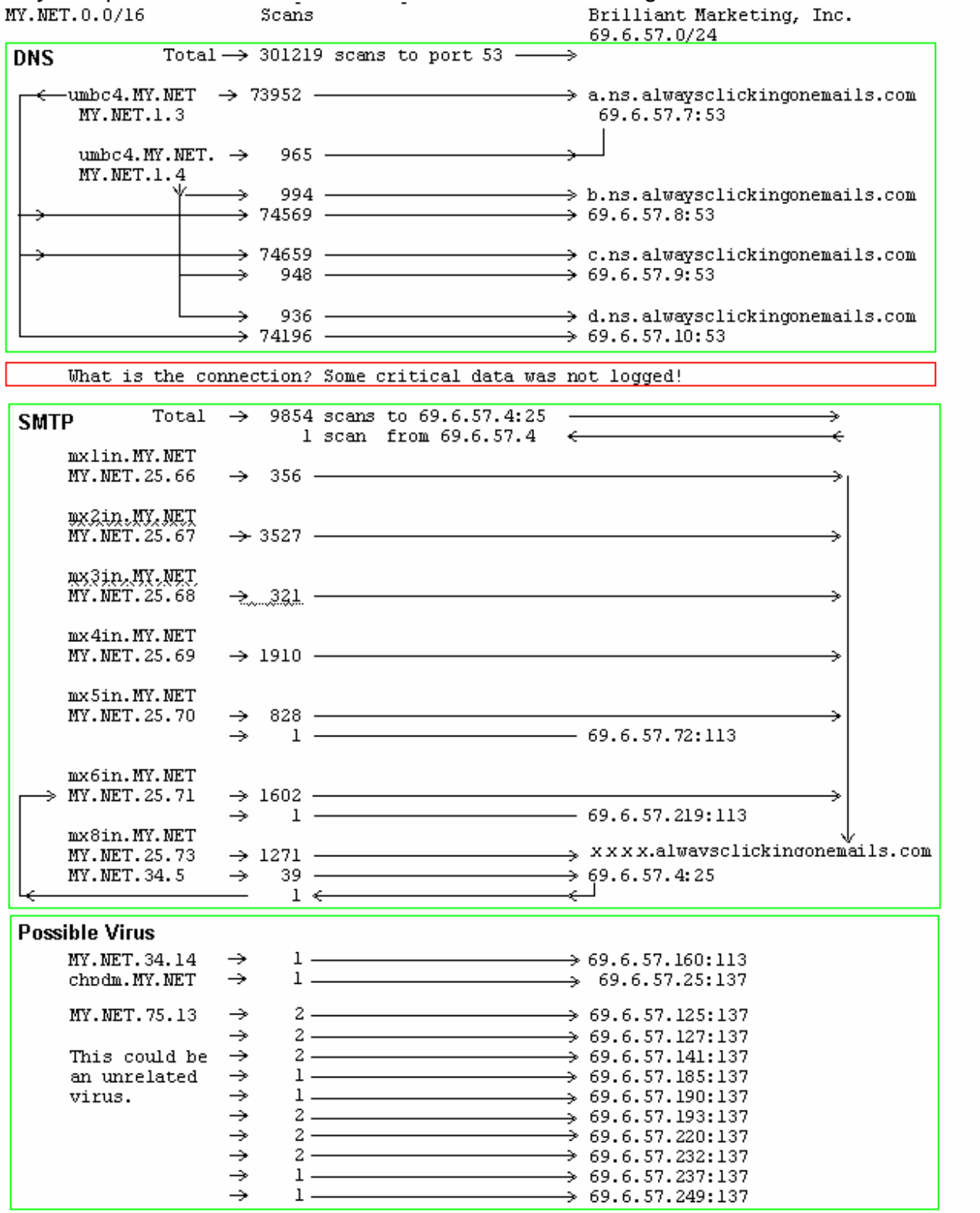
Hourly Traffic to 69.6.57.0/24



Traffic was usually highest around 4 AM, when nobody will notice! As people get up and use their systems, it decreases. What happened the evening of March 20?

Link Diagram for a possible spam attack from 69.6.57.0/24.

No communication between the internal mailservers and the name servers was logged. Only one packet was from 69.6.57.0/24.The mail server logs need to be checked.



Windows Networking

Alert	Total	External Sources	Internal Dest	Internal Sources	External Destinations	Incoming	Outgoing
SMB Name Wildcard	3493			172	416		3493
SMB C access	99	26	5			99	

Windows Networking uses should always be blocked at the firewall. The systems listed below sent SMB messages to each other. The internal ones need to be checked.

```
03/20-15:51:56.011801  [**] EXPLOIT x86 NOOP [**] 200.100.143.220:2805 -> MY.NET.190.102:135
03/20-16:16:23.790173  [**] SMB Name Wildcard [**] MY.NET.190.102:137 -> 200.100.143.220:137
03/20-16:16:25.934303  [**] SMB Name Wildcard [**] MY.NET.190.102:137 -> 200.100.143.220:137

03/20-15:51:45.254354  [**] EXPLOIT x86 NOOP [**] 200.100.143.220:2796 -> MY.NET.190.93:135
03/20-16:16:28.251325  [**] SMB Name Wildcard [**] MY.NET.190.93:137 -> 200.100.143.220:137
03/20-16:16:29.803975  [**] SMB Name Wildcard [**] MY.NET.190.93:137 -> 200.100.143.220:137

03/19-07:18:36.128036  [**] SMB Name Wildcard [**] MY.NET.190.93:137 -> 62.219.158.207:137
03/19-07:18:37.637391  [**] EXPLOIT x86 NOOP [**] 62.219.158.207:3413 -> MY.NET.190.93:135

03/19-23:02:33.975839  [**] EXPLOIT x86 NOOP [**] 65.94.92.205:1589 -> MY.NET.190.93:135
03/19-23:11:27.914367  [**] SMB Name Wildcard [**] MY.NET.190.93:137 -> 65.94.92.205:137

03/19-23:02:38.014194  [**] EXPLOIT x86 NOOP [**] 65.94.92.205:1751 -> MY.NET.190.95:135
03/19-23:11:27.372794  [**] SMB Name Wildcard [**] MY.NET.190.95:137 -> 65.94.92.205:137

03/16-05:03:01.845255  [**] SMB Name Wildcard [**] MY.NET.190.97:137 -> 68.94.252.180:137
03/16-05:03:02.188437  [**] EXPLOIT x86 NOOP [**] 68.94.252.180:3393 -> MY.NET.190.97:135
03/16-05:03:02.319761  [**] SMB Name Wildcard [**] MY.NET.190.97:137 -> 68.94.252.180:137
03/16-05:03:02.796590  [**] SMB Name Wildcard [**] MY.NET.190.97:137 -> 68.94.252.180:137
```

Each internal IP had SMB C access alerts with multiple external sources. This indicates that there may be communication among the attackers.

Destination	Port	Alert	External Sources
MY.NET.190.102	139	SMB C access	10
MY.NET.190.92	139	SMB C access	9
MY.NET.190.93	139	SMB C access	15
MY.NET.190.95	139	SMB C access	18
MY.NET.190.97	139	SMB C access	14

SMB Name Wildcard alerts on port 137. [13] It looks like the ISP is blocking this port but that should not be depended upon. This alert caught 50 internal hosts sending to port 137 and an external destination. Here is the top ten.

Source	Total	Dest	03/16	03/17	03/18	03/19	03/20	Comments
MY.NET.75.13	557	154	82	76	80	128	191	Probable Virus
MY.NET.150.198	364	137	56	63	45	85	115	Responds to scans
MY.NET.150.44	203	68	44	34	23	39	63	Responds to scans
MY.NET.29.30	55	1	9	7	13	6	20	199.239.137.216 NY Times Digital
MY.NET.11.4	53	0	1	52	0	0	0	Reserved dest, could be game.
MY.NET.190.92	39	25	13	4	3	2	17	No 2 Scan source, Probable Virus
MY.NET.109.86	26	15	7	0	5	14	0	Responds to scans
MY.NET.112.152	24	4	2	5	2	5	10	eDonkey, Also had Red Worm alert.
MY.NET.152.17	24	1	0	0	0	24	0	Dest 63.163.24.78 DELMARVA BANK
MY.NET.153.85	24	1	0	0	24	0	0	Dest 216.145.5.196, no reverse DNS

The data for MY.NET.75.13 looked random, but when it was sorted by destination, 78 out of 154 destinations went to one of five class C subnets. It could be a virus scan.

Source	Count	Subnet
MY.NET.75.13	23	65.60.17.0/24
MY.NET.75.13	22	65.60.18.0/24
MY.NET.75.13	12	66.239.205.0/24
MY.NET.75.13	11	69.6.57.0/24
MY.NET.75.13	10	63.218.84.0/24

Several internal hosts scanned port 135, but none for more then two consecutive days. This could be because IT had notified the system owners. Two of them restarted, they could have been cleaned and then have been compromised again.

Date	MY.NET.190.92 Count	MY.NET.66.17 Count	MY.NET.153.174 Count	MY.NET.80.224 Count	MY.NET.81.59 Count
16-Mar	16576	530269	18203	31804	
17-Mar		473475			
18-Mar			24988		
19-Mar	138411		188660		6978
20-Mar	1864986				

No Windows autoconfig IP's were detected, but 169.254.45.176 could be game server.

Date	Destestination	Port	Alert	Count	Internal Source's
03/16	169.254.45.176	137	SMB Name Wildcard	120	74
03/17	169.254.45.176	137	SMB Name Wildcard	147	87
03/18	169.254.45.176	137	SMB Name Wildcard	124	76
03/19	169.254.45.176	137	SMB Name Wildcard	183	97
03/20	169.254.45.176	137	SMB Name Wildcard	122	50

MY.NET.30.3 activity and MY.NET.30.4 activity

Alert	Total	External Sources	Internal Dest	Incoming
MY.NET.30.4 activity	29864	279	1	29864
MY.NET.30.3 activity	10934	171	1	10934

These are Novell servers for the faculty and Staff.[14] The alerts log all connection attempts from the Internet. There were 24592 alerts for destination port 51443 on MY.NET.30.4. This port is sometimes used for Novell's secure iFolder. [15]

Destination Port	Service	MY.NET.30.3	MY.NET.30.4
51443	unknown	0	24592
524	ncp	9861	3927
80	http	288	1238
99	metagram	48	46
6129	dameware	25	25
4899	radmin-port	14	14

There were 187406 scans from 22 external sources to 15710 internal destinations for port 6129. There were 4590 scans from 4 external sources to 144 internal destinations for port 99. An alert for port 99 and port 6129 needs to be added ASAP. The scans to +6129 could be lookedfor "Dameware Mini Remote Control"[16]

TFTP - Internal TCP connection to external tftp server

This is an odd alert. TCP 69 is a "well known" port, but nothing is known to use it. So anything that triggered this alert is suspicious, and when it is the primary nameserver, MY.NET.1.3:69, then it needs to be looked into. Below we see that three packets were sent with source MY.NET.1.3:69. One possibility is that it was spoofed. This would not be hard to do but it would require an internal PC and an external PC. It is more likely that MY.NET.1.3 is responding with a reset packet. However port 69 should not do this because it should be closed. My guess is that there is a firewall on this system that is rejecting packets instead of dropping them. This should be corrected ASAP.

```
03/19-16:26:46.341667  [**] TFTP - External TCP connection to internal tftp server [**] 165.127.89.114:17139 ->
MY.NET.1.3:69
03/19-16:26:46.341947  [**] TFTP - External TCP connection to internal tftp server [**] MY.NET.1.3:69 ->
165.127.89.114:17139
03/19-16:26:46.844832  [**] TFTP - External TCP connection to internal tftp server [**] 165.127.89.114:17139 ->
MY.NET.1.3:69
03/19-16:26:46.844993  [**] TFTP - External TCP connection to internal tftp server [**] MY.NET.1.3:69 ->
165.127.89.114:17139
03/20-19:01:50.226639  [**] TFTP - External TCP connection to internal tftp server [**] 213.184.233.169:57928 ->
MY.NET.1.3:69
03/20-19:01:50.226833  [**] TFTP - External TCP connection to internal tftp server [**] MY.NET.1.3:69 ->
213.184.233.169:57928
```

MY.NET.24.15, MY.NET.24.44 and MY.NET.6.7 also responded to scans to port 69. They should be checked as well. MY.NET.84.203 initiated five TCP exchanges with 213.22.228.37:69 in Portugal. The external hosts above also did some scanning.

This could be some students that need to be contacted to make sure nothing malicious is being done and to be told that portscanning is not professional behavior. If the owner of MY.NET.84.203 does not know who is doing this, then it could be malicious a trojan.

Source	Destination	Count	Ports scanned
165.127.89.114	MY.NET.1.3	513	482
165.127.89.114	MY.NET.153.149	549	505
165.127.89.114	MY.NET.24.15	541	496
165.127.89.114	MY.NET.24.44	532	492
165.127.89.114	MY.NET.30.3	332	308
165.127.89.114	MY.NET.6.7	277	258
165.127.89.114	MY.NET.69.217	628	583
165.127.89.114	MY.NET.97.35	692	650
213.184.233.169	MY.NET.1.3	883	868

Some Malicious Scans

Source	Count	Dest Port	Alert	Target
61.129.45.60	2184	80	EXPLOIT x86 NOOP	138 hosts scanned
61.129.45.60	120	80	MY.NET.30.3 activity	MY.NET.30.3
61.129.45.60	48	99	MY.NET.30.3 activity	MY.NET.30.3
61.129.45.60	117	80	MY.NET.30.4 activity	MY.NET.30.4
61.129.45.60	46	99	MY.NET.30.4 activity	MY.NET.30.4
130.39.190.86	153	111	External RPC call	119 hosts scanned

There is no recorded response to 130.39.190.86. Below is an interesting exchange with 61.129.45.60. MY.NET.150.44 responds to any scan with an "SMB Name Wildcard". This might be two virus-infected hosts "fighting".

```
03/20-06:21:59.549308  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:21:59.857108  [**] SMB Name Wildcard [**] MY.NET.150.44:137 -> 61.129.45.60:137
03/20-06:22:01.346512  [**] SMB Name Wildcard [**] MY.NET.150.44:137 -> 61.129.45.60:137
03/20-06:22:02.846539  [**] SMB Name Wildcard [**] MY.NET.150.44:137 -> 61.129.45.60:137
03/20-06:22:03.487237  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:22:17.987510  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:22:32.127396  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:2636 -> MY.NET.150.44:80
03/20-06:22:32.350267  [**] SMB Name Wildcard [**] MY.NET.150.44:137 -> 61.129.45.60:137
03/20-06:22:35.347102  [**] SMB Name Wildcard [**] MY.NET.150.44:137 -> 61.129.45.60:137
03/20-06:22:36.112848  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:22:37.458186  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:3274 -> MY.NET.150.44:80
03/20-06:22:42.787726  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:3831 -> MY.NET.150.44:80
03/20-06:22:48.117119  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:4401 -> MY.NET.150.44:80
03/20-06:22:50.613091  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:22:53.444482  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:1158 -> MY.NET.150.44:80
03/20-06:22:58.775134  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:1714 -> MY.NET.150.44:80
03/20-06:23:04.102596  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:2307 -> MY.NET.150.44:80
03/20-06:23:09.433843  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:2858 -> MY.NET.150.44:80
03/20-06:23:14.753137  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:3505 -> MY.NET.150.44:80
03/20-06:23:15.992279  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:23:20.092390  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:4076 -> MY.NET.150.44:80
03/20-06:23:25.420557  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:4748 -> MY.NET.150.44:80
03/20-06:23:30.748570  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:1414 -> MY.NET.150.44:80
03/20-06:23:36.077064  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:1954 -> MY.NET.150.44:80
03/20-06:23:41.395734  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:2538 -> MY.NET.150.44:80
03/20-06:23:46.734154  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:3076 -> MY.NET.150.44:80
03/20-06:23:50.426547  [**] SMB Name Wildcard [**] MY.NET.150.44:1065 -> 61.129.45.60:137
03/20-06:23:52.063575  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:3723 -> MY.NET.150.44:80
03/20-06:23:57.385441  [**] EXPLOIT x86 NOOP [**] 61.129.45.60:4297 -> MY.NET.150.44:80
```

TFTP - Internal UDP connection to external tftp server

This port, UDP 69, should be blocked by the firewall. A closer look showed that this was really an attempt to access the AFS (port 4672) from port 69. This is obviously a crafted packet. Seven exchanges started with 217.147.34.242:69 -> MY.NET.111.34:4672. There was only one packed sent back, so it could be a reset. It would be prudent to put a firewall on all AFS servers to drop traffic that is not needed.

High port 65535 udp - possible Red Worm - traffic

Total	External Src	Internal Dest	Internal Src	External Dest	Incoming
68	18	9	26	352	173

This is not the red worm. That uses TCP. Below is an interesting exchange. Each time slot is minutes apart. It may be a P2P that is designed to be hard to analysis.

Start time	End	UDP Datagrams	Source	Destination
03/18-03:12:22	03/18-03:18:52	120	69.140.137.209:65535	MY.NET.6.62:65535
03/18-03:31:30	03/18-03:32:00	36	MY.NET.6.62:65535	69.140.137.209:65535
03/19-13:20:08	03/19-13:20:38	3	MY.NET.6.62:65535	129.2.109.131:65535

Possible trojan server activity

27374 is a valid ephemeral port, but 4883,7602,9277,24621,43333 are not registered for any service. Port 3938 is registered for anet-b OMF data b.

On Sat Apr 24 2004 SORBS DNSbl reported that MY.NET.27.232 had open proxy and smtp relay ports. [17] If this is true then the person responsible should be reprimanded, unless it is an approved honeypot.

FTP passwd attempt

Total 39 alerts from 38 hosts. The destination each time was MY.NET.24.47.
This could be a busy anonymous FTP server. Some people, including the author, frequently do not spell anonymous correctly the first time. Anonymous FTP is OK for public files, but password protected FTP can be broken into. SFTP should be used.

IP_src	P_src	Srvc	IP_dest	Dst_P	Srvc	No.	Comments
MY.NET.24.47	4883	N/A	67.114.251.118	27374	virus	80	1052 scans to MY.NET.24.47:21 and 158 scans to other ports
67.114.251.118	27374	virus	MY.NET.24.47	4883	N/A	46	FTP server? could trigger alert.
66.90.79.46	27374	virus	MY.NET.27.232	7602	N/A	5	Nothing else from or to 66.90.79.46.
MY.NET.27.232	7602	N/A	66.90.79.46	27374	virus	5	Started with dest 27374, server port.
66.90.79.46	27374	virus	MY.NET.97.74	9277	N/A	5	MY.NET.27.232, .97.74 need checking.
MY.NET.97.74	9277	N/A	66.90.79.46	27374	virus	5	
MY.NET.84.235	27374	virus	217.229.193.215	43333	N/A	2	.84.235 did 3745 scans to 4662/edonkey
MY.NET.42.11	3938	N/A	69.13.85.50	27374	virus	1	.42.11 did 52 scans to 6881/Bittorrent

TCP SRC and DST outside network

Source	No.	Domain	Comment
172.x.x.x	43	AOL	Possible modem connection to AOL
134.192.145.215	1	U of Maryland at Baltimore	Possible Laptop
146.94.38.115	1	Wilkes University	Possible Laptop
10.10.11.68	1	BlackHole	Should use NAT to access Internet.
192.168.x.x	7	BlackHole	Should use NAT to access Internet

<UMBC NIDS> External MiMail alert and TCP SMTP Source Port traffic

MY.NET.12.6 needs to be checked. It could be a student run email server. It appears as an attacker 166 times in the DShield database. All with source port 25, the earliest was 2004-03-22 A Google.com search for MY.NET.12.6 found about 40 spam complaints from 24 Jul 2003 to 23 Apr 2004.[18
]

Srm Port	Dest	Dest Port	Count	Alert
25	MY.NET.12.6	25	58	<UMBC NIDS> External MiMail alert
25	MY.NET.12.6	25	23	TCP SMTP Source Port traffic
25	MY.NET.27.167	1016	1	TCP SMTP Source Port traffic

Connect to 515 from inside

Source	Alert	Count	Destination	Dest Port	Service
MY.NET.60.16	connect to 515 from inside	1	128.244.225.45	515	printer
MY.NET.97.192	connect to 515 from inside	9	128.183.16.169	515	printer

Port 515 is registered for printers. It is very insecure and should be blocked.

NIMDA - Attempt to execute cmd from campus host

Source	Source Port	Destination	Destination Port
MY.NET.98.101	1073	64.70.33.122	80

This could be Windows Update. It has been reported that wxpsp2.windowsupdate.microsoft.com can resolve to 64.70.33.122 [19]..

Out Of Spec scans.

There were 2899 OOS packets. 2624 of them had the flags set to 12****S*. flag 1 is for Congestion Window Reduced and flag 2 is for Explicit Congestion Notification - Echo. These flags were recently added to TCP and still considered illegal by the snort setup used here. A Google search for ECN found many postings by people who were experimenting with the Linux kernel.

1051 OOS packets had source 68.54.84.49 and destination MY.NET.6.7:110 and they all had the flags set to 12****S*. No two packets came in the same minute. A sort by hour, showed they were usually around 40 packets an hour. This looks like a Linux or BSD system that has new/modified kernel and is querying the mailserver with the POP protocol every 90 seconds.

803 OOS packets had destination MY.NET.12.6:25. 801 of them had the flags set to 12****S*. The table below shows six subnets that have to many distinct sources. The ack flag is not set, so they have to be spoofed. They did not come fast enough for DOS attack. The destination has many other alerts. It is probably a student run mailserver and the "attacker" is likely a student who does not realize he is risking arrest. The University IT policy is thoroughly unclear on this and so are the laws. It would be in

everybody's interest to have the policy clearly prohibit scanning and especially idle scanning except in a designated test LAN.

OOS packets with MY.NET.12.6

Destination	Packets	Source	Distinct source IP's
MY.NET.12.6:25	91	66.225.198.20	1
MY.NET.12.6:25	75	67.72.78.212	1
MY.NET.12.6:25	64	35.8.2.252	1
MY.NET.12.6:25	45	66.180.237.99	1
MY.NET.12.6:25	25	35.8.2.251	1
MY.NET.12.6:25	218	66.232.231.x/8	141
MY.NET.12.6:25	69	66.249.111.x/8	31
MY.NET.12.6:25	57	66.232.233.x/8	48
MY.NET.12.6:25	23	38.118.189.x/8	19
MY.NET.12.6:25	22	216.95.201.x/8	9
MY.NET.12.6:25	18	66.249.100.x/8	9
61.135.147.27:3737	1	MY.NET.12.6:25	1
64.42.130.159:4645	1	MY.NET.12.6:25	1
219.153.1.170:43141	1	MY.NET.12.6:25	1
208.55.43.103:10658	1	MY.NET.12.6:25	1

In detect 2 we looked at null packets being sent to MY.NET.12.4. Here we have 50 null scans from 68.122.128.1 to MY.NET.12.4". So it is still going on. The owner or MY.NET.12.4 should be contacted. If he does not know who is doing this, then further investigation would be prudent.

For three hours starting at 00:32 on March 16m 68.6.102.188 sent 50 OOS packets to MY.NET.42.2. This included 12 null scans and 23 SynFin scans, most with other flags. In total there were 33 different flag settings. There were only 36 scans recorded during this time. There were 14 alerts. It is interesting that they came in pairs with identical time stamps. This looks like an error, but a search if the alert logs shows only 79 identical pairs of alerts. That means that the source must have been sending two identical packets at the same time, and that snort analyzed them together. This IP belongs to Cox Communications in Atlanta . This looks like some curious student, but that needs to be confirmed. Again, the University IT policy is thoroughly unclear and should be revised to clearly prohibit this. This is a legitimate thing to study, but not on the Internet where they may be mistaken as "hackers." A secure LAN needs to be setup so students can experiment and not risk being arrested!

Selected External Source Addresses and Registration	
Host: 66.90.79.46 Net range: 66.90.64.0 - 66.90.127.255 Name: FDCservers.net Country: US Address: 141 w jackson blvd #1135 Chicago, IL 60604 Contact: kral, petr 312 913-9200 sales@fdcservers.net	Host: 213.184.233.169 Net range: 213.184.233.0 - 213.184.233.255 Name: Belarus ISP Company Country: BY Address: Sovetskaya, 97, korp. 4 Gomel Contact: Boris Borsukov +375 232-57-87-11 boris@server.by
Host: 165.127.89.114 Net range: 165.127.0.0 - 165.127.255.255 Name: State of Colorado General Government Computer Country: US Address: 690 Kipling St. LakeWood, CO 80215 Contact: Applebach, Ron 1-303-239-4313 Postmaster@state.co.us	Net range: 172.128.0.0 - 172.211.255.255 43 IP's in this range were detected internally. Name: America Online Country: US Address: 22000 AOL Way Dulles VA, 20166 Contact: +1-703-265-4670 domains@aol.net
Host: 213.22.228.37 Net range: 213.22.228.0 - 213.22.229.255 Name: TVCABO-Portugal Cable Modem Network Country: PT Address: Avenida 5 de Outubro, 208 Edifício Santa Maria 9 andar 1069-203 Lisboa Contact: + 351 217824760 + 351 217914800 ABUSE@TVCABO.PT	Host: 69.6.57.4, 69.6.57.7, 69.6.57.8, 69.6.57.9, 69.6.57.10 Net range: 69.6.57.0 - 69.6.57.255 Name: Brilliant Marketing, Inc Country: US Address: 5919 GREENVILLE AVE # 140 DALLAS, TX 75206-1906 Contact: Matthew Scholl brilliantmarketing2000@yahoo.com

Analysis process

I did not like any of the open source programs I tried. ACID has a pretty display, but I found it difficult to correlate events. Others, such as SnortSnarf could not handle all the data. I looked at the scripts Les Gordon used, [20] they were written by Hee So. [21] I decided to write a better analysis script. It took over a month and was a worthwhile learning experience. It is named analyze_this.pl and uses Perl and Postgresql. When the program has finished, the data is in indexed tables, and ready to query with SQL.

There were several changes to the data, The /16 subnet was changed to MY.NET.
All "[" and "]" characters in the alert names were changed to "<" or ">" because of a bug.

My system has 1.2G of ram in three pc133 slots. It took under three hours to process the data Les Gordon used. When I used the data for this practical it had not finished after 24 hours. It could handle the alerts, but there was to much scan data. I had to create a separate database for the scans. This made the SQL queries more complicated. A PC with 2 G of ram may be able to run with everything in one database. This would be easier to query.

Analyze_this.pl will be released with the GPL copyright. If anybody wants to improve it, then they are very welcome to. [22]
Some areas that need work are:
1 Performance tuning to reduce thrashing.
2 Implement UDP, TCP, ICMP protocols
3 Split the scan data across two or more databases.
4 Fix the problem with "[" and "]" characters

References

[1] Spam Laws: United States: Federal Laws: CAN-SPAM Act of 2003.
URL: http://www.spamlaws.com/federal/108s877.html (28 April 2004).

[2] Hatchit, Ann. UT wins victory in anti-spam case. 26 Mar 26 2004. URL:
http://austin.bizjournals.com/austin/stories/2004/03/22/daily37.html (6 May 2004).

[3] Carlton, Rick. [SpamCop-List] Re: UT wins victory in anti-spam case 26 Mar 2004.
URL: http://news.spamcop.net/pipermail/spamcop-list/2004-March/077671.html
(6 May 2004).

[4] Storm, Peter. GIAC Certified Intrusion Analyst (GCIA) Practical Assignment Version 3.3. 15 Nov 2003. URL: http://www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf
(9 Feb 2004).

[5] Vance, Ashlee. RIAA tax could add millions to education fees 28th April 2004.
http://www.theregister.co.uk/2004/04/28/riaa_sues_moreschools/ (28 April 2004).

[6] UMBC IT-01: Policy For Responsible Computing 6 September 1996. URL:
http://www.umbc.edu/oit/sans/security/policy/2-UMBC/IT-01-final.html (28 April 2004).

[7] POLICY ON ACTIVE SCANNING AND ANALYSIS 24 Sep 2003.
URL: http://www.giac.org/GCIA_assign34.php (28 April 2004).

[8] URL: http://www.dshield.org (28 April 2004).

[9] Cami <camis(-at-)mweb.co.za>.[AMaViS-user] new trend. (13 Apr 2004).
http://archive.netbsd.se/?list=amavis-user&a=2004-04&mid=104739 (28 April 2004).

[10] Kriwitsky, Dan. [cobalt-users] Mailserver keeps crashing 26 Mar 2004. http://list.cobalt.com/pipermail/cobalt-users/2004-March/101544.html (28 April 2004).

[11] State of Texas - Comptroller of Public Accounts. Franchise Tax Certification of Account Status. http://ecpa.cpa.state.tx.us/coa/servlet/cpa.app.coa.CoaGetTp?Pg=tpid&Search_Nm=BRILLIANT%20MARKETING%20&Button=search&Search_ID=32014134897 (10 April 2004).

[12] Ramalho, David. JotzBlog. 26 March 2004. URL: http://bloglike.com/jeff/archives/000129.html (10 April 2004).

[13] Ruiu, Dragos. SNORT FAQ 25 march 2002. URL: http://www.snort.org/docs/faq.html#4.15 (10 April 2004).

[14] Office of Information Technology. Novell for Windows NT/2000/XP Installation. http://www.umbc.edu/oit/sans/desktopsupport/installation/novell/windows/2000_xp/ (10 April 2004).

[15] Novell. NetWare Server Issues in iFolder 2.1. URL: http://www.novell.com/documentation/lg/ifolder21/index.html?page=/documentation/lg/ifolder21/readme/data/ahf1v06.html (2 May 2004).

[16] cwagner17 Networking: pc anywhere alike. URL: http://beta.experts-exchange.com/Networking/Q_20676922.html (29 mar 2005)

[17]Subject: SORBS DNS Blocklist Stats [Sat Apr 24 12:00:02 2004]. news-only@sorbs.net. URL: http://groups.google.ca/groups?q=%22130.85.27.232%22&hl=en&lr=&ie=UTF-8&oe=ISO-8859-1&selm=c6chuu%24548%241%40bunyip.cc.uq.edu.au&rnum=1 (28 April 2004).

[18] Google Groups. URL: http://www.groups.google.com (28 April 2004).

[19] joe00100. Forums » OS and Software » Microsoft help » wxpsp2.windowsupdate? Wednesday April 14th, @03:14PM. URL: http://www.dslreports.com/forum/remark,9970314~mode=flat (28 April 2004).

[20] Gordon, Les. Intrusion Analysis - The Director's Cut! GIAC Certified Intrusion Analyst (GCIA). Practical Assignment. Nov. 22 2002. URL: URL: http://www.giac.org/practical/GCIA/Les_Gordon_GCIA.doc (2 May 2004).

[21] So, Hee. "GCIA Intrusion Detection In Depth. GCIA Practical. Feb. 16 2002. URL: http://www.giac.org/practical/Hee_So_GCIA.doc (2 May 2004)

[22] Analyze_this. http://dunrobin.dyn.dhs.org/pub/analyze_this.html (9 May 2004).