Global Information Assurance Certification Paper

SANS GIAC Level Two –

Intrusion Detection In Depth

GCIA Practical Assignment
Practical Assignment Version 3.0
Jack D. Green, MCSE, GSEC

Assignment 1 – Describe the State of Intrusion Detection

Honeypots, Description and Illustration

The tools available to intrusion analyst include network and host based intrusion detection, system and event logs and packet sniffers. Honeypots represent another relatively recent tool for the analyst's kit.

The purpose of this paper is to describe honeypot functions by illustrating examples of those functions using the Specter 6.0 honeypot.[1]

Three characteristics of a honey pot:

A honeypot is a computer attached to a network *whose value is in being attacked or compromised*[2]. This is a machine specifically designed to be set in harm's way, gather information about who is visiting it and send out alerts to the parties interested in those visits.

These machines typically have *no production functions* other than being attacked and gathering data on the attackers.

They *offer some level of interaction with the attackers*. Please refer to Figure 1 during this discussion.

> 1) They may be configured to emulate different operating systems and versions. . The leftmost column, *Operating System*, show a wide array of selections as well as *characteristics* of the operating system. Most are self explanatory. Of interest is aggressive. Under this mode the unit will gather information, then announce itself by sending the custom warning (bottom right hand corner).
>
> 2) They offer simulations of interesting services. These services may return behavior characteristic enough of a production computer to keep an attacker occupied. For example, refer to Detect #2 above to see the interaction between the scanftp program and Specter.
>
> 3) Specter offers Counter-intelligence, the *Intelligence* Column, to scan the attacker and return information about the attacker's platform.
>
> 4) They can provide traps (port monitoring devices) which provide timely alerts of intrusion attempts. An example of an alert, emailed to me, is shown in Detect #5. Additionally they provide logging services of attempts.

[1] A special thanks to the Specter people for offering an evaluation copy of their software. Further information on Specter is available in the reference section.
[2] http://www.enteract.com/~lspitz/honeypot.html

The purposes of a honeypot:

A honeypot can *provide detection*. When the bad guys get to the place on the net where your honeypot is located, you know that perimeter has been compromised. The honeypot acts as a sensor, emulating a vulnerable host. It will be among the first to detect an intruder's attack and alert you.

A honeypot can *provide prevention.* If you're perimeter has been compromised and you've been alerted to the intrusion, some might call this a dubious claim. The other side of that argument is that while the intruder is playing with your honeypot, they are not playing with your production machines. Additionally, you are placed on notice that your next perimeter (defense in depth) may be at risk.

Once your network is compromised, a honeypot can aid in your *reaction to attack.* The honeypot is expendable. You may pull it off-line to get information about the attackers without compromising your organization's business functions. You can't necessarily pull your web server off-line to do a forensic study on it. Since the honey pot has no production value, you can. Also, it may provide more information about the attacker than your production hosts. Recall that the Specter honeypot provides counter-intelligence about the attacker. Some useful features include:

- Trace Route back to the attacker
- Whois lookup
- Finger information (if available)
- SMTP banner

The advantages of a honeypot:

A honeypot is *versatile*. Referring to the leftmost column of Figure 1, one host may be set to emulate a wide number of operating systems. One may set traps based upon current concerns. For example, a generic trap has been set to watch for KaZaA activity on the network.

A honeypot provides *high-fidelity information.* Referring to Detect #2 (below), you can see that I logged a great deal of information without having to run the ftp service. Specter provided an emulated response to the issued commands, allowing us to view the script in its entirety.

A honeypot aids in *flexible data gathering.* A honey pot can email out to an intrusion response email group, log to a central syslog server and make an entry in an incident database. Each of these options may be turned on or off based upon the organizations needs.

A honeypot can act as a *deterrent.* When the honeypot announces itself to intruders, it serves notice to attackers that the network is protected by people who are security-aware! An argument can also be made that if every organization used one, there would be *hundreds of thousands* of hosts out there distracting the bad guys with non-exploitable computers.

The disadvantages of a honeypot:

A honeypot might become *a compromised host.* Specter is an excellent product. Mine was running on the Windows NT 4.0 Operating. This is an operating system that has been tested *in vivo* for years. Yet, flaws remain. Add to that scenario a misconfigured or poorly patched system and you have a vulnerable host sitting in a vulnerable area of your network. Once compromised, the system can be used as a platform to launch attacks.

A honeypot can *consume the resources of your staff*. Specter is easy to set up. Other honeypots may not be as easy. The more interaction with the attackers, the more difficult it can be to set up and configure properly.

A honeypot can *consume the resources of your network bandwidth*. If you allow unlimited connections you have set the occasion for a loss of service. The honeypot or Operating system that you use ought to be able to limit connections over a period of time. Too few connections tips off the bad guys that something is amiss, too many connection and you're wasting precious bandwidth.

There remain *questions of legality*. You've caught the bad guy. Did you entrap him/her? A system has been deployed *whose sole purpose was to be compromised*! Perhaps, the system was compromised and sensitive information is now on display to the world. Even though you didn't put the information there, you set up a system *whose sole purpose…* Are you liable for providing a platform that others use to display credit card information?

A honeypot offers only *one data point.* If the attackers miss your honeypot but attack the production servers first, the honeypot did no good.

Conclusion:

A honeypot is an effective tool for the intrusion analyst. As can be seen from this paper, they can provide additional information along with IDS's and log monitors.

They can provide a great deal of information about attackers and when properly configured are as safe to run as any other host.

Resources:

Honeypot sources:

- BackOfficer Friendly - http://www.nfr.com/products/bof/
- Deception Toolkit - http://www.all.net/dtk/download.html
- Mantrap - http://www.recourse.com/
- Specter - http://www.specter.com/

References

Even, Loras R. "What is a Honeypot? Honey Pot Systems Explained" 12 July 2000 URL: http://www.sans.org./newlook/resources/IDFAQ/honeypot3.htm

Raikow, David. "Sweet Temptation" 24 Sept 2001. URL: http://www.eweek.com/article/0,3658,s=723&a=15414,00.asp

Spitzner, Lance. "Honeypots *Definitions and Value of Honeypots* ." 08 March 2002 URL: http://www.enteract.com/~lspitz/honeypot.html

Sink, Michael. "The Use of Honeypots and Packet Sniffers for Intrusion Detection" 15 April 2001. URL: http://rr.sans.org/intrusion/honey_pack.php

Spitzner, Lance. Know Your Enemy: I, II and III. 2000. URL: http://project.honeynet.org/papers/

Thompson, Clive. "How do corporations stop hackers? They don't. They simply lure them to a "honeypot." Nov 2001 URL: http://www.globetechnology.com/robmag/robmagnov_01.html

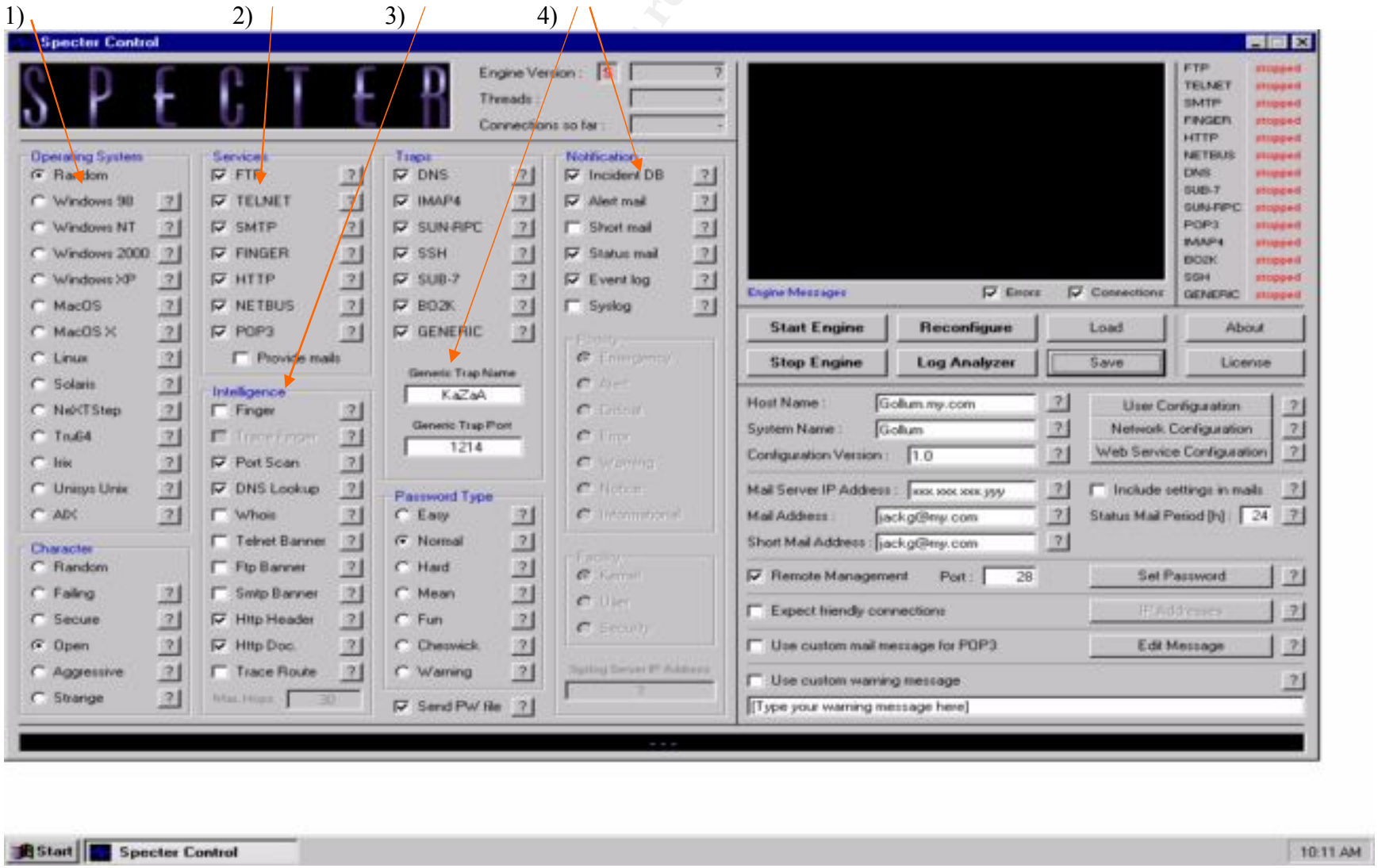


Figure 1 – Specter Control Console

Assignment 2 – Network Detects

Detect #1

Source of the Detect:

This intrusion attempt was taken from an employer's network

Detect Generated By:

The snort detect was generated by Snort version 1.7 using the snort.org ruleset available at http://www.snort.org. The specific rule is identified below:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS CodeRed
v2 root.exe access"; flags: A+; uricontent:"scripts/root.exe?"; nocase;
classtype:web-application-attack; sid:1256; rev:2;)
```

Probability the Source Address was Spoofed:

The attacking host is a victim of sadmind/IIS worm. It was seeking unpatched Microsoft IIS hosts. Using internet explorer http://209.100.126.144 revealed an infected site actively trying to infect systems. Upon attachment, the host offers a download, apparently of the virus. The site is defaced in the standard coarse way. The virus is offered on a web page and a process shown in Figure 2 is initiated.

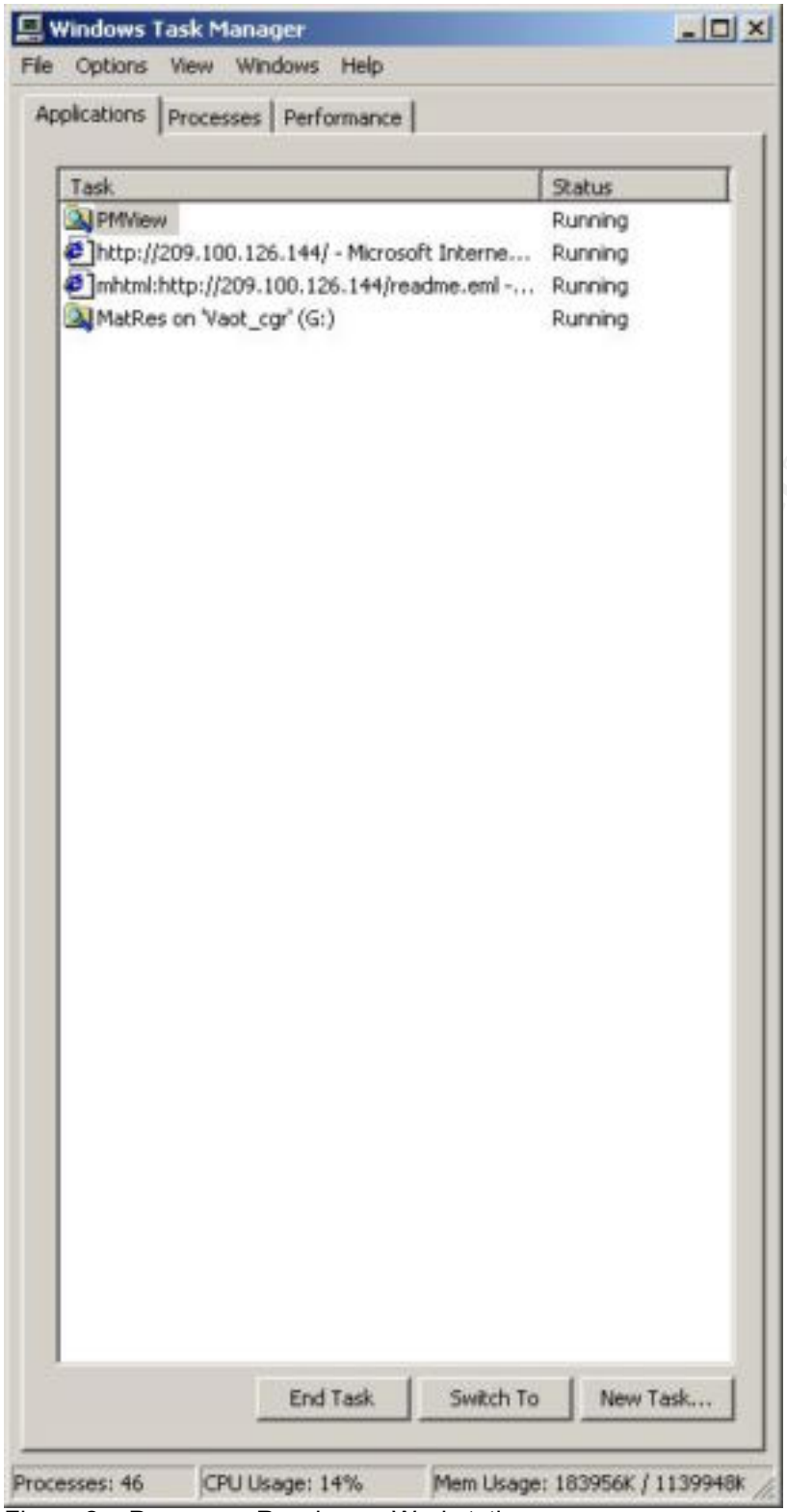


Figure 2 – Processes Running on Workstation

It is highly unlikely that this address is spoofed.

Description of the attack:

This was a challenging attack to identify. While snort is identifying this as Code Red v2, this is attack is the result of a sadmind/IIS infected host. Also the variation on the defacement as the original sadmind/IIS substituted *USA Government* for *Chinese Government* added to the puzzle.

The attack is using exploit CVE-2001-0333. The directory traversal vulnerability,

Directory traversal vulnerability in IIS 5.0 and earlier allows remote attackers to execute arbitrary commands by encoding .. (dot dot) and "\" characters twice.[3]

Scott Wunsch mirrored an example of the defaced (but harmless) page at :

http://www.wunsch.org/mirrors/codered/

A *parpaphrased* example is shown below

Attack Mechanism:

An unpatched Solaris host (7, 2.6, 2.5.1,2.5, 2.4, and 2.3 (SunOS(tm) 5.7, 5.6, 5.5.1, 5.5, 5.4 and 5.3) is vulnerable[4] to a buffer overflow on Solstice sadmind that allows a compromised machine to run arbitrary code. Once infected the worm launches attacks on a random set of Class B addresses looking for other Solaris as well as IIS machines. Once a Solaris system is infected it seeks to infect 2000 IIS servers. Upon completion of this goal, Sadmind/IIS defaces its host's Index.html *home page*.

Apparently the compromised Solaris machine then attacked with *W32.Nimda.A@MM* as these characteristic alerts appeared:

[3] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0333
[4] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0977

- GET /scripts/root.exe?/c+dir
- GET /MSADC/root.exe?/c+dir
- GET /c/winnt/system32/cmd.exe?/c+dir

The alert is shown below:

```
[**] WEB-IIS CodeRed v2 root.exe access [**]

02/21-04:04:49.199418 209.100.126.144:3563 -> 192.168.1.2:80

TCP TTL:114 TOS:0x0 ID:18083 IpLen:20 DgmLen:112 DF

***AP*** Seq: 0xE5FB1A8A  Ack: 0xB28C08  Win: 0x2238  TcpLen: 20

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

Examining the dump of the attack:
```

tcpdump -x -r tcpdump19 'host 209.100.126.144'

```
04:04:49.129418 eth0 P 209.100.126.144.3563 > mywebserver.http: S
3858438793:3858438793(0) win 8192 <mss 1460> (DF)

Syn

04:04:49.129418 eth0 P mywebserver.http > 209.100.126.144.3563: S
11701255:11701255(0) ack 3858438794 win 8760 <mss 1460> (DF)

Syn Ack

04:04:49.189418 eth0 P 209.100.126.144.3563 > mywebserver.http: .
1:1(0) ack 1 win 8760 (DF)

Ack

04:04:49.199418 eth0 P 209.100.126.144.3563 > mywebserver.http: P
1:73(72) ack 1 win 8760 (DF)

GET /scripts/root.exe?/c+dir HTTP/1.0 Host: Connnection nlose

 4500 0070 46a3 4000 7206 b045 d164 7e90
 c0a8 0102 0deb 0050 e5fb 1a8a 00b2 8c08
 5018 2238 035e 0000 4745 5420 2f73 6372
 6970 7473 2f72 6f6f 742e 6578 653f 2f63
 2b64 6972 2048 5454 502f 312e 300d 0a48
 6f73 743a 2077 7777 0d0a 436f 6e6e 6e65
 6374 696f 6e3a 2063 6c6f 7365 0d0a 0d0a

04:04:49.319418 eth0 P 209.100.126.144.3581 > mywebserver.http: P
1:71(70) ack 1 win 8760 (DF)

GET /MSADC/root.exe?/c+dir HTTP…

 4500 006e 7fa3 4000 7206 7747 d164 7e90
```

```
 c0a8 0102 0dfd 0050 e5fb 1b43 00b2 8c15
 5018 2238 ac7f 0000 4745 5420 2f4d 5341
 4443 2f72 6f6f 742e 6578 653f 2f63 2b64
 6972 2048 5454 502f 312e 300d 0a48 6f73
 743a 2077 7777 0d0a 436f 6e6e 6e65 6374
 696f 6e3a 2063 6c6f 7365 0d0a 0d0a
```

```
[**] WEB-IIS cmd.exe access [**]
02/21-04:04:49.499418 209.100.126.144:3609 -> 192.168.1.2:80
TCP TTL:114 TOS:0x0 ID:59555 IpLen:20 DgmLen:120 DF

***AP*** Seq: 0xE5FD1070  Ack: 0xB28C2E  Win: 0x2238  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

(TCPDump of cmd.exe omitted)

Nimda is attacking port 80 on mywebserver. A great deal has been written on Nimda. An excellent resource is found at http://www.incidents.org/react/nimda.pdf . In summary, Nimda presents a four-pronged approach to propogation:

- HTTP scanning for IIS vulnerabilities
    - Unicode directory traversal
    - IIS/PWS
    - Backdoors from Code Red/Sadmind
- E-MAIL (via MAPI and user intervention)
    - Copies of itself sent to email addresses from various x86 clients
    - Javascript propogation
- Internet Explorer HTTP iframe and javascript autoexec
    - readme.eml
    - readme.exe
- Open Windows File sharing
    - Network aware copying to shares on other computers

Correlation

Figure 3 , provided by incidents.org[5], shows that in the period between 9/18/01 and 9/25/01, the number of recorded scans on port 80 increased by as much as five times the normal scanning rate. There were more than 86,000 unique IP addresses reported scanning port 80.

[5] http://www.incidents.org/react/nimda.pdf

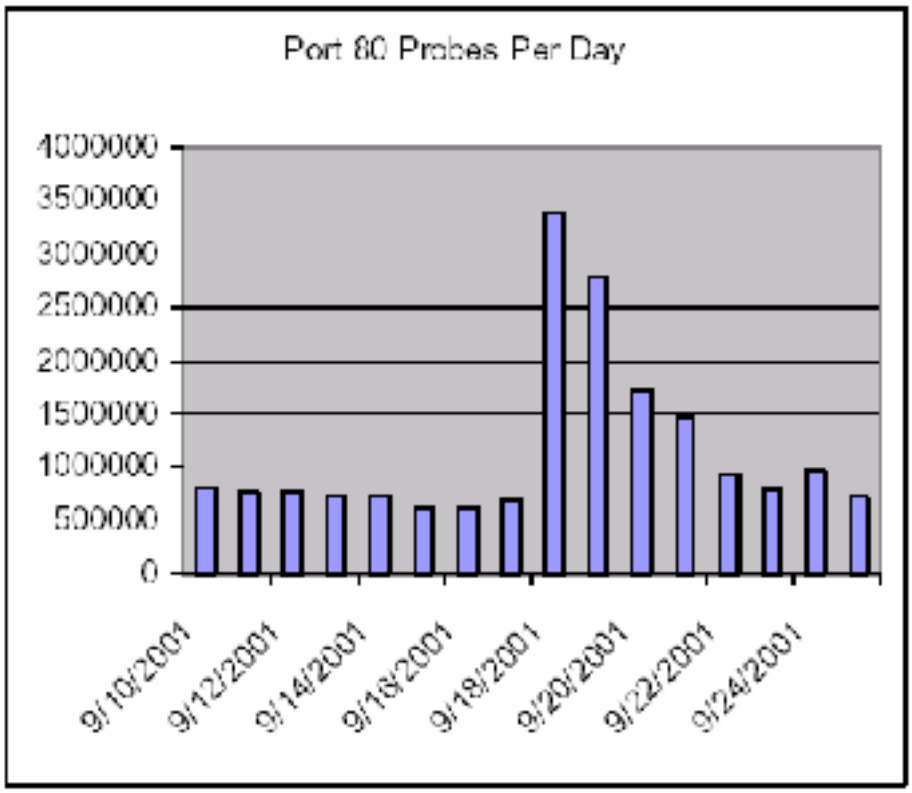


Figure 3 – Port 80 Probes per Day

Since then countermeasures have been installed. However, as can be seen form this anecdote, the virus remains active.

The author reported this victim host to Genuity.com and it was removed from the net.

Evidence of Active Targeting

The targeting by both Sadmind and Nimda are random. Sadmind scans random Class B addresses. Nimda can propogate itself through many different methods preferring to target its neighbors. It attack IP neighbors with the same first octet with a 25% probability, with the same first two octets 50% probability[6]

Severity

Both Sadmind and Nimda provides attackers with administrator authority over compromised systems. Once in control any code may be executed from the infected systems. Further, because of the breadth of attacks, it is exteremely difficult to clean.

[6] http://www.incidents.org/react/nimda.pdf, page 5

Criticality. This unit houses password protected web pages and files for sharing among business partners. Criticality = 3.

Lethality – The attack, if successful would've provided administrator access throughout the network. Lethality = 5.

System Countermeasures – All patches were current and Anti-virus was current. System Countermeasures = 5

Network Countermeasures – Permissive firewall allows port 80 requests in. Network Countermeasures = 2

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

(5 + 3) – (5 + 2) = +1

Defensive Recommendations

* Microsoft has posted IIS updates (well before the advent of this worm) at:

ttp://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms01-044.asp

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms01-020.asp

* Solaris offered its patch well before sadmind also at:

http://sunsolve.sun.com/pub-cgi/retrieve.pl?doctype=coll&doc=secbull/191&type=0&nav=sec.sba

* All of the anti-virus have offered updates to prevent this worm:

Each workstation attached to the LAN must be current.

* Additionally clean-up tools are offered at these sites:

NAI:
http://download.nai.com/products/mcafee-avert/NimdaScn.zip

Symantec

http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.removal.tool.html

A word about clean-up. The general body of literature seems to view clean-up in a dubious light. While tools are available those who recommend an O/S rebuild state that the worm is so intrusive that it is unclear that all traces can be removed.

Multiple Choice Question

Which of the following is NOT characteristic of Nimda?

a) Email propogation through MAPI
b) IIS Exploit
c) Buffer overflow of Sadmind
d) Copy to network shares

Detect #2

Source of the Detect:

This intrusion attempt was taken from an employer's network outside the firewall.

Detect Generated By:

This detect was generated by Specter 6.0. Specter is a honeypot program that is discussed later in this paper. It was configured to emulate an FTP server (among other things) running a failing NT system.

Probability the Source Address was Spoofed:

The attacking host is seeking anonymous writable ftp servers. It is unlikely that the Source address is spoofed.

Description of the attack:

The scan logs in as anonymous and proceeds to try to change directories to many commonly named directories. Once Specter returned a *200 CWD successful* (there are no such directories, of course), the script attempts to create a directory and, if successful, log the IP.

Note the fidelity of the logs coming from Specter:

```
Client connecting: 217.82.8.88
Client tries anonymous Login
--->331 Anonymous access allowed, send identity (e-mail name) as
password.
Client sent PASS 'Ngpuser@home.com'
--->230 User anonymous logged in.
Client wants to change current directory to
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015701p': command not understood.
Client wants to change current directory to incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015703p': command not understood.
Client wants to change current directory to _vti_pvt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015704p': command not understood.
Client wants to change current directory to upload/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015705p': command not understood.
Client wants to change current directory to temp/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015706p': command not understood.

Client wants to change current directory to tmp/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015707p': command not understood.
Client wants to change current directory to pub/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015708p': command not understood.
Client wants to change current directory to pub/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015709p': command not understood.
Client wants to change current directory to _vti_txt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015710p': command not understood.
Client wants to change current directory to _vti_log/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015711p': command not understood.
Client wants to change current directory to wwwroot/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015712p': command not understood.
Client wants to change current directory to anonymous/
--->200 CWD command successful.
```

```
Command not understood
--->502 'MKD 020217015713p': command not understood.
Client wants to change current directory to public/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015714p': command not understood.
Client wants to change current directory to public/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015715p': command not understood.
Client wants to change current directory to outgoing/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015716p': command not understood.
Client wants to change current directory to anonymous/_vti_pvt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015717p': command not understood.
Client wants to change current directory to anonymous/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015718p': command not understood.
Client wants to change current directory to mailroot/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015719p': command not understood.
Client wants to change current directory to ftproot/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015720p': command not understood.
Client wants to change current directory to anonymous/pub/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015721p': command not understood.
Client wants to change current directory to anonymous/public/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015722p': command not understood.
Client wants to change current directory to _vti_cnf/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015723p': command not understood.
Client wants to change current directory to anonymous/_vti_cnf/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015724p': command not understood.
Client wants to change current directory to images/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015725p': command not understood.
Client wants to change current directory to _private/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015726p': command not understood.
Client wants to change current directory to cgi-bin/
--->200 CWD command successful.
Command not understood
```

```
--->502 'MKD 020217015728p': command not understood.
Client wants to change current directory to usr/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015729p': command not understood.
Client wants to change current directory to usr/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015730p': command not understood.
Client wants to change current directory to home/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015731p': command not understood.
Client wants to change current directory to outgoing/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015732p': command not understood.
Client wants to change current directory to _kurdt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015733p': command not understood.
Client wants to change current directory to ~tmp/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015734p': command not understood.
Client wants to change current directory to anonymous/_vti_pvt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015735p': command not understood.
Client wants to change current directory to anonymous/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015736p': command not understood.
Client wants to change current directory to mailroot/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015737p': command not understood.
Client wants to change current directory to ftproot/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015738p': command not understood.
Client wants to change current directory to anonymous/pub/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015739p': command not understood.
Client wants to change current directory to anonymous/public/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015740p': command not understood.
Client wants to change current directory to anonymous/_vti_cnf/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015741p': command not understood.
Client wants to change current directory to images/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015742p': command not understood.
```

```
Client wants to change current directory to cgi-bin/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015743p': command not understood.
Client wants to change current directory to admin/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015745p': command not understood.
Client wants to change current directory to administrator/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015746p': command not understood.
Client wants to change current directory to inbox/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015747p': command not understood.
Client wants to change current directory to up/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015748p': command not understood.
Client wants to change current directory to dropbox/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015749p': command not understood.
Client wants to change current directory to
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015750p': command not understood.
Client wants to change current directory to winnt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015751p': command not understood.
Client wants to change current directory to macos/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015752p': command not understood.
Client wants to change current directory to unix/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015754p': command not understood.
Client wants to change current directory to mark/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015755p': command not understood.
Client wants to change current directory to master/
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015756p': command not understood.
Client wants to change current directory to
--->200 CWD command successful.
Command not understood
--->502 'MKD 020217015757p': command not understood.
Connection timed out
Closing connection with 217.82.8.88
```

Attack Mechanism

A host from Deutsche Telekom is providing this stimulus. The target services is FTP.

```
inetnum:        217.80.0.0 - 217.89.31.255
netname:        DTAG-DIAL14
descr:          Deutsche Telekom AG
country:        DE
admin-c:        DTIP-RIPE
tech-c:         ST5359-RIPE
status:         ASSIGNED PA
remarks:
*******************************************************
remarks:        * ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK
ATTACKS, *
remarks:        * ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC.
*
remarks:
*******************************************************
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20020108
source:         RIPE
```

The attack mechanism may be Grims Ping[7]. Grim's Ping offers subnet scanning and logging/printing scan results.

Once public writable directories are found the hosts may be used to store files of the attacker's choice or a a base from which to launch *Bounce Attacks*.

Name	CVE-1999-0017
Description	FTP servers can allow an attacker to connect to arbitrary ports on machines other than the FTP client, aka FTP bounce.

In summary a bounce attack allows the attacker to obscure his/her own IP by using the ftp server's services. S/he my then transfer files to other hosts using the data port (TCP20) or may scan other hosts while evading the victims IDS.

Our particular attack is a reconnaissance searching for suitable hosts.

[7] http://grimsping.cjb.net/tutorial.htm

Correlation

There were two immediately apparent correlations offered on www.incidents.org. Both are reported by Laurie Zirkle. An excerpt below shows the similarity in the script execution.

- *Date*: Fri, 16 Nov 2001 11:51:34 -0500
- *From*: Laurie Zirkle <lat@xxxxxxxxxx>
- *Subject*: November 15, 2001 probes (part 1)

```
        inetnum:      80.13.82.0 - 80.13.82.255
        netname:      IP2000-ADSL-BAS
        descr:        BSTOU104 Toulouse Bloc1
        country:      FR

Dec  7 23:00:16 hostsa ftpd[20566]: refused connect from AOrleans-201-
1-3-70.abo.wanadoo.fr
Dec  7 23:00:16 hostz ftpd[22467]: refused connect from AOrleans-201-1-
3-70.abo.wanadoo.fr
Dec  7 23:00:16 hostt ftpd[19928]: refused connect from AOrleans-201-1-
3-70.abo.wanadoo.fr
Dec  7 23:01:09 hostca in.ftpd[28908]: refused connect from AOrleans-
201-1-3-70.abo.wanadoo.fr
Dec  7 23:01:09 hostca in.ftpd[28909]: refused connect from AOrleans-
201-1-3-70.abo.wanadoo.fr
Dec  7 23:01:09 hostca in.ftpd[28910]: refused connect from AOrleans-
201-1-3-70.abo.wanadoo.fr
Dec  7 23:01:12 hostca in.ftpd[28911]: refused connect from AOrleans-
201-1-3-70.abo.wanadoo.fr
Dec 07 23:00:16 hostl proftpd[18815] hostl (AOrleans-201-1-3-
70.abo.wanadoo.fr[80.13.82.70]): FTP session opened.
Dec 07 23:00:17 hostl proftpd[18815] hostl (AOrleans-201-1-3-
70.abo.wanadoo.fr[80.13.82.70]): ANON anonymous: Login successful.
Dec 07 23:00:17 hostl proftpd[18815] hostl (AOrleans-201-1-3-
70.abo.wanadoo.fr[80.13.82.70]): FTP session closed.
Dec 08 03:41:53 hostl proftpd[21112] hostl (AOrleans-201-1-3-
70.abo.wanadoo.fr[80.13.82.70]): FTP session opened.
Dec 08 03:41:57 hostl proftpd[21112] hostl (AOrleans-201-1-3-
70.abo.wanadoo.fr[80.13.82.70]): ANON anonymous: Login successful.
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [07/Dec/2001:23:00:17 -
0500] "PASS anonymous" 230 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:41:57 -
0500] "CWD /pub/" 250 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:41:57 -
0500] "PASS Ngpuser@home.com" 230 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:41:58 -
0500] "CWD /pub/incoming/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:41:58 -
0500] "CWD /public/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:41:58 -
0500] "MKD 011208094152p" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:02 -
0500] "CWD /incoming/" 250 -
```

```
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:03 -
0500] "CWD /" 250 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:03 -
0500] "CWD /_vti_pvt/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:03 -
0500] "MKD 011208094157p" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:04 -
0500] "CWD /_vti_pvt/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:04 -
0500] "CWD /_vti_txt/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:04 -
0500] "CWD /upload/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:04 -
0500] "MKD 011208094158p" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:05 -
0500] "CWD /_vti_log/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:05 -
0500] "CWD /anonymous/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:05 -
0500] "CWD /wwwroot/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:06 -
0500] "CWD /outgoing/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:06 -
0500] "CWD /public/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:06 -
0500] "CWD /temp/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:08 -
0500] "CWD /anonymous/_vti_pvt/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:08 -
0500] "CWD /anonymous/incoming/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:08 -
0500] "CWD /tmp/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:09 -
0500] "CWD /anonymous/pub/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:09 -
0500] "CWD /ftproot/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:09 -
0500] "CWD /mailroot/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:10 -
0500] "CWD /_private/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:10 -
0500] "CWD /_vti_cnf/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:10 -
0500] "CWD /anonymous/public/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:10 -
0500] "CWD /images/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:11 -
0500] "CWD /cgi-bin/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:11 -
0500] "CWD /cgibin/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:12 -
0500] "CWD /usr/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:12 -
0500] "CWD /usr/incoming/" 550 -
AOrleans-201-1-3-70.abo.wanadoo.fr UNKNOWN ftp [08/Dec/2001:03:42:13 -
0500] "CWD /home/" 550 -
```

```
Dec 08 03:42:13 hostl proftpd[21112] hostl (AOrleans-201-1-3-
70.abo.wanadoo.fr[80.13.82.70]): FTP session closed.

=-=-=-=-=-=-=-=-=-=-=-=
Dec  7 23:05:36 hostmau portsentry[210]: attackalert: Connect from
host: 63.119.202.150/63.119.202.150 to TCP port: 80


--
Laurie
```

Evidence of Active Targeting

This was a random scan against IP's in the organizations block with publicly available IP addresses.

Severity

Criticality – Given that the scanner was targeting all hosts on the subnet including the email server. However, no ftp services are being offered on this subnet Particularly on the honeypot. Criticality = 1.

Lethality – This is a scan. Upon identification of a potentially susceptible client, the attacker must still find ftp services that permit bounce attacks or file storage. Lethality = 1.

System Countermeasures – The scan was on a honeypot. It has none of the services it advertises. It runs on a systems with the latest patches. System Countermeasures = 5

Network Countermeasures – A permissive firewall allowed the scan to proceed on the honeypot as designed. No other scanned hosts were running ftp. Network Countermeasures = 5

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

(1 + 1) – (5 + 5) = -8

Defensive Recommendations

Defensive recommendations include:

1) ftp service is not necessary, it is not run
2) configure router to drop FTP control requests (TCP21)
3) configure ftp server to allow passive-mode client data

Multiple Choice Question:

When running an ftp service, best practices includes:
(Choose 2)

1) only run anonymous service
2) block incoming TCP port 21 requests for security
3) configure ftp server to allow passive mode clients only when possible
4) install latest patches from the ftp vendor

Detect #3

Source of Detect:

This scan came from an employer's network. The network's ISP is responsible for providing NAT services.

Detect was generated by:

Snort 1.7 is running on a RedHat system that sits outside the firewall. The network's ISP is responsible for providing NAT services. The snort box is also running tcpdump.

The alert, a Syn-Fin scan, was logged by the stream4 preprocessor rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN
FIN";flags:SF; reference:arachnids,198; classtype:attempted-recon;
sid:624; rev:1;)
```

Probabilitiy that the address was spoofed

The purpose of this scan is reconnaissance of a subnet. The scan tried each host once (that I caught). The probability of spoofing is low.

Description of the attack

The attack sent TCP Packets directed at ftp (port 21) with the syn and fin flags set. It swept through 15 live hosts on the subnet within 2 seconds at 4:24 p.m. looking for live hosts.

Counter-reconnaisance

```
IP resolves to neocyber21.net which is in the Asia-Pacific net.

www.apnic.net resolves the IP block as assigned to Korea
```

```
inetnum            210.108.0.0 - 210.115.255.255
netname            KRNIC-KR
descr              KRNIC
descr              Korea Network Information Center
country            KR
admin-c            HM127-AP, inverse
tech-c             HM127-AP, inverse
remarks            ******************************************
remarks            KRNIC is the National Internet Registry
remarks            in Korea under APNIC. If you would like to
remarks            find assignment information in detail
remarks            please refer to the KRNIC Whois DB
remarks            http://whois.nic.or.kr/english/index.html
remarks            ******************************************
mnt-by             APNIC-HM, inverse
mnt-lower          MNT-KRNIC-AP, inverse
changed            hostmaster@apnic.net 19970430
changed            hostmaster@apnic.net 20010606
source             APNIC


person             Host Master, inverse
address            Korea Network Information Center
address            Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-
ku, Seoul, 137-070, Republic of Korea
country            KR
phone              +82-2-2186-4500
fax-no             +82-2-2186-4496
e-mail             hostmaster@nic.or.kr, inverse
nic-hdl            HM127-AP, inverse
mnt-by             MNT-KRNIC-AP, inverse
changed            hostmaster@nic.or.kr 20010514
source             APNIC
```

```
[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.659418 210.114.174.238:21 -> 192.168.1.2:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=


[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.669418 210.114.174.238:21 -> 192.168.1.3:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

```
[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.679418 210.114.174.238:21 -> 192.168.1.4:21
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.689418 210.114.174.238:21 -> 192.168.1.5:21
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.699418 210.114.174.238:21 -> 192.168.1.6:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.739418 210.114.174.238:21 ->
192.168.1.124:21
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.759418 210.114.174.238:21 ->
192.168.1.101:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
```

```
02/20-16:24:11.779418 210.114.174.238:21 ->
192.168.1.111:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.789418 210.114.174.238:21 ->
192.168.1.109:21
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.819418 210.114.174.238:21 ->
192.168.1.119:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.849418 210.114.174.238:21 ->
192.168.1.106:21
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.859418 210.114.174.238:21 ->
192.168.1.105:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.889418 210.114.174.238:21 ->
192.168.1.113:21
```

```
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:11.899418 210.114.174.238:21 ->
192.168.1.102:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:12.159418 210.114.174.238:21 ->
192.168.1.117:21
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection
[**]
02/20-16:24:12.189418 210.114.174.238:21 ->
192.168.1.112:21
TCP TTL:23 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x30AB5218  Ack: 0x187A3935  Win: 0x404
TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Correlating tcpdump

[root@roac log]# tcpdump -x -r tcpdump19 'host 210.114.174.238'

```
16:24:11.659418 eth0 P 210.114.174.238.ftp > mywebserver.ftp: SF
816534040:816534040(0) win 1028

16:24:11.659418 eth0 P mywebserver.ftp > 210.114.174.238.ftp: R 0:0(0)
ack 816534042 win 0

16:24:11.669418 eth0 P 210.114.174.238.ftp > mailsrv.ftp: SF
816534040:816534040(0) win 1028

16:24:11.669418 eth0 P mailsrv.ftp > 210.114.174.238.ftp: R 0:0(0) ack
816534042 win 0

16:24:11.679418 eth0 P 210.114.174.238.ftp > citrix.ftp: SF
816534040:816534040(0) win 1028
```

```
16:24:11.679418 eth0 P citrix.ftp > 210.114.174.238.ftp: R 0:0(0) ack
816534042 win 0

16:24:11.689418 eth0 P 210.114.174.238.ftp > 192.168.1.5.ftp: SF
816534040:816534040(0) win 1028


16:24:11.689418 eth0 P 192.168.1.5.ftp > 210.114.174.238.ftp: R 0:0(0)
ack 816534042 win 0

16:24:11.699418 eth0 < 210.114.174.238.ftp > roac.ftp: SF
816534040:816534040(0) win 1028

16:24:11.699418 eth0 > roac.ftp > 210.114.174.238.ftp: R 0:0(0) ack
816534042 win 0 (DF)


16:24:11.739418 eth0 P 210.114.174.238.ftp > 192.168.1.124.ftp: SF
816534040:816534040(0) win 1028

16:24:11.739418 eth0 P 192.168.1.124.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0


16:24:11.759418 eth0 P 210.114.174.238.ftp > 192.168.1.101.ftp: SF
816534040:816534040(0) win 1028

16:24:11.759418 eth0 P 192.168.1.101.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0

16:24:11.779418 eth0 P 210.114.174.238.ftp > 192.168.1.111.ftp: SF
816534040:816534040(0) win 1028

16:24:11.779418 eth0 P 192.168.1.111.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0

16:24:11.789418 eth0 P 210.114.174.238.ftp > 192.168.1.109.ftp: SF
816534040:816534040(0) win 1028

16:24:11.789418 eth0 P 192.168.1.109.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0

16:24:11.819418 eth0 P 210.114.174.238.ftp > 192.168.1.119.ftp: SF
816534040:816534040(0) win 1028

16:24:11.819418 eth0 P 192.168.1.119.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0

16:24:11.849418 eth0 P 210.114.174.238.ftp > 192.168.1.106.ftp: SF
816534040:816534040(0) win 1028


16:24:11.849418 eth0 P 192.168.1.106.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0
```

```
16:24:11.859418 eth0 P 210.114.174.238.ftp > 192.168.1.105.ftp: SF
816534040:816534040(0) win 1028


16:24:11.859418 eth0 P 192.168.1.105.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0


16:24:11.889418 eth0 P 210.114.174.238.ftp > 192.168.1.113.ftp: SF
816534040:816534040(0) win 1028

16:24:11.889418 eth0 P 192.168.1.113.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0


16:24:11.899418 eth0 P 210.114.174.238.ftp > 192.168.1.102.ftp: SF
816534040:816534040(0) win 1028

16:24:11.899418 eth0 P 192.168.1.102.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0

16:24:12.159418 eth0 P 210.114.174.238.ftp > 192.168.1.117.ftp: SF
816534040:816534040(0) win 1028
 4500 0028 9a02 0000 1806 c54f d272 aeee


16:24:12.159418 eth0 P 192.168.1.117.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0

16:24:12.189418 eth0 P 210.114.174.238.ftp > 192.168.1.112.ftp: SF
816534040:816534040(0) win 1028

16:24:12.189418 eth0 P 192.168.1.112.ftp > 210.114.174.238.ftp: R
0:0(0) ack 816534042 win 0
```

Attack Mechanism

The scan was likely generated by *Synscan 1.6*[8] which may be identified by a window size of 0x404 and an ID of 39426. Note also that the Sequence and Acknowledgement numbers remain identical throughout these transactions.

A Syn/Fin scan may be able to avoid the detection of an IDS

Evidence of Active Targeting

As pointed out earlier, this is a subnet scan. The scanner is marching through all available IP addresses within the public subnet. Please refer to my defensive recommendation for further clarification

[8] http://www.psychoid.lam3rz.de/synscan.html

Correlations

There were no correlations available for this particular IP address. It is likely that other scans were not reported. Given that the tool that generated this is freely available there were many reports of scans using Synscan, including:

Mou-Liang Kung)

Dear Stephen, I saw all kinds of SYN/FIN Scans this month from East or West (but not US). Has SYN/FIN become an international vocabulary? Notice that id numbers are ALL IDENTICAL!

July 1, 2000, SYN/FIN Scan for DNS from KRNIC-KR-23, Korea Network Information Center:

```
07:46:50.476744 211.50.42.3.53 > 255.255.255.255.53:
  SF  204309722:204309722(0) win 1028 (ttl 26, id 39426)
07:46:51.674731 211.50.42.3.53 > MyNet.189.53:
  SF 1746714120:1746714120(0) win 1028 (ttl 26, id 39426)
07:46:53.015248 211.50.42.3.53 > 255.255.255.255.53:
  SF 360140609:360140609(0) win 1028 (ttl 26, id 39426)
```

July 10, 2000, SYN/FIN Scan for DNS from SCARAMEA, an ISP in Amsterdam, Netherlands

```
21:59:07.187661 193.173.174.119.53 > 255.255.255.255.53:
  SF 1937116546:1937116546(0) win 1028 (ttl 28, id 39426)
21:59:08.416647 193.173.174.119.53 > MyNet.189.53:
  SF 544517778:544517778(0) win 1028 (ttl 28, id 39426)
21:59:09.747158 193.173.174.119.53 > 255.255.255.255.53:
  SF 1324192513:1324192513(0) win 1028 (ttl 28, id 39426)
```

July 16, 2000, SYN/FIn Scan for Portmap from a host on The Communications Authority of Thailand, an International Telecommunications Service Provider

```
16:45:37.850907 202.47.250.70.111 > 255.255.255.255.111:
  SF 1288942995:1288942995(0) win 1028 (ttl 28, id 39426)
16:45:39.069900 202.47.250.70.111 > MyNet.189.111:
  SF 992428653:992428653(0) win 1028 (ttl 28, id 39426)
16:45:40.390414 202.47.250.70.111 > 255.255.255.255.111:
  SF 1444642537:1444642537(0) win 1028 (ttl 28, id 39426)
```

July 18, 2000, SYN/FIN Scan for FTP from MUSIKPROJEKT-DK, dk.uu.net in Denmark

```
05:38:20.705950 195.24.7.228.21 > 255.255.255.255.21:
  SF  931517680:931517680(0) win 1028 (ttl 28, id 39426)
05:38:21.921944 195.24.7.228.21 > MyNet.189.21:
  SF 315207050:315207050(0) win 1028 (ttl 28, id 39426)
05:38:23.242450 195.24.7.228.21 > 255.255.255.255.21:
  SF 1095310691:1095310691(0) win 1028 (ttl 28, id 39426)
```

This is the month of January SYN-FIN scan detected on my cable modem. The ports targetted were (TCP): 21, 53, 109, 111, 1578, 27374

```
[**] SCAN-SYN FIN [**]
01/01-03:13:45.850825 192.168.4.1:111 -> 192.168.30.1:111
TCP TTL:26 TOS:0x0 ID:39426
**SF**** Seq: 0x7256C6F8 Ack: 0x5E4B7209 Win: 0x404
[**] SCAN-SYN FIN [**]
01/04-10:11:11.165753 207.105.159.130:21 -> 192.168.30.1:21
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x29125EBB Ack: 0x7D70D534 Win: 0x404
[**] SCAN-SYN FIN [**]
01/05-14:20:06.153749 209.112.47.7:27374 -> 192.168.30.1:27374
TCP TTL:36 TOS:0x0 ID:39426
**SF**** Seq: 0x4DE0B257 Ack: 0x293A7863 Win: 0x404
[**] IDS198 - SCAN-SYN FIN [**]
01/13-00:36:26.188265 207.21.74.78:109 -> 192.168.30.1:109
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x2AF2476F Ack: 0x4F460C6 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/14-08:32:02.156295 24.176.79.249:53 -> 192.168.30.1:53
TCP TTL:31 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0xC9B87E3 Ack: 0x2FEEB7A Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/14-17:39:35.504418 24.176.79.249:21 -> 192.168.30.1:21
TCP TTL:31 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x39E613E7 Ack: 0x67713768 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/19-08:32:59.745976 203.197.78.161:21 -> 192.168.30.1:21
TCP TTL:30 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x72C5248F Ack: 0x53B679D1 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/20-22:04:18.982408 216.233.82.222:53 -> 192.168.30.1:53
TCP TTL:29 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x1EB9AAE0 Ack: 0x225BA5D5 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/21-07:09:58.115945 210.179.12.76:109 -> 192.168.30.1:109
TCP TTL:24 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x59920A68 Ack: 0x238C17E9 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/25-09:41:31.562435 24.20.193.34:53 -> 192.168.30.1:53
TCP TTL:29 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x3F471914 Ack: 0x710F183 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/27-15:30:22.068157 24.9.81.251:1578 -> 192.168.30.1:1578
TCP TTL:34 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x738F19BC Ack: 0x4BD47DD0 Win: 0x404 TcpLen: 20
[**] IDS198 - SCAN-SYN FIN [**]
01/28-03:11:09.078172 216.12.241.2:111 -> 192.168.30.1:111
TCP TTL:29 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x3555E960 Ack: 0x55B057B9 Win: 0x404 TcpLen: 20
```

Severity

Criticality – Given that the scanner was targeting all hosts on the subnet including the email server. Without the scanning of all hosts scanning of an E-mail server would set the criticality index at 4. Add all host and it peaks at 5. Criticality = 5.

Lethality – This scan did not do any damage, but it likely resulted in the attacker mapping the network. . Lethality = 3.

System Countermeasures – The Operating System on the email server remains current on all patches. However, the client hosts are older operating systems with spotty patches. Given the number and variety of hosts being scanned, I'll take a weighted measure. Eight hosts had current operatng systems and patches (8*5). Seven hosts are basic desktop workstations with olders OS'es.(7*3). System Countermeasures = ((8*5 + (7*3) /15 = int(4..06) = 4

Network Countermeasures – A permissive firewall and incorrectly configured NAT allowed the attack to proceed. Network Countermeasures = 2

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

(5 + 3) – (4 + 2) = +2

Defensive Recommendations

As I mentioned previously, the ISP is (for the time being) providing network address translation to this subnet. While reconfiguring routers, they inadvertently provided static mapping to all 62 public IP addresses. Under the original configuration, only a few hosts had publicly available addresses.

Defensive recommendations include:

1) restrict static mapping to only necessary hosts
2) configure router to drop TCP packets with syn/fin set

Multiple Choice Question:

From a live host, the Syn-Fin scan expects to receive

1)a reset from the attacked host
2) a reset/ack from the attacked host
3) a syn/ack from the attacked host

4) a fin from the attacked host

Detect #4

Source of Detect:

This scan came from an employer's network. I investigated this alert because I was somewhat familiar with the subnet from which it came. If it was indeed an attack, the owner would want to know.

Detect was generated by:

Snort 1.7 is running on a RedHat system that sits outside the firewall. The network's ISP is responsible for providing NAT services. The snort box is also running tcpdump.

The alert, a large ICMP (over 800 bytes), was logged by the rule:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large ICMP
Packet"; dsize: >800; reference:arachnids,246; classtype:bad-unknown;
sid:499; rev:1;)
```

Probability that the address was spoofed

The purpose of this scan is MTU discovery from an AIX mail server. It was not spoofed.

Description of the Attack:

As mentioned earlier, I was concerned about nefarious activity coming from a nearby network. Notice that TTL is 248 on these packets. Further, these packets were directed at the mail server on the DMZ. A review of large ICMP packet alerts revealed nothing immediately threatening.

Name	Description
CVE-2000-0041	Macintosh systems generate large ICMP datagrams in response to malformed datagrams, allowing them to be used as amplifiers in a flood attack.
CVE-2001-0057	Cisco 600 routers running CBOS 2.4.1 and earlier allow remote attackers to cause a denial of service via a large ICMP echo (ping) packet.

CVE-2001-0754	Cisco CBOS 2.3.8 and earlier allows remote attackers to cause a denial of service via a series of large ICMP ECHO REPLY (ping) packets, which cause it to enter ROMMON mode and stop forwarding packets.
CVE-2001-0861	Cisco 12000 with IOS 12.0 and line cards based on Engine 2 and earlier allows remote attackers to cause a denial of service (CPU consumption) by flooding the router with traffic that generates a large number of ICMP Unreachable replies.
CAN-2001-0592	** CANDIDATE (under review) ** Watchguard Firebox II prior to 4.6 allows a remote attacker to create a denial of service in the kernel via a large stream (>10,000) of malformed ICMP or TCP packets. [9]

I returned to my TCPDump logs and found the correlation with SMTP packets. Since there was a total of 20 identical sets across several days, the exhibit is abbreviated. for clarity

Set 1-Snort Alert

```
[**] MISC Large ICMP Packet [**]
02/20-10:33:54.909418 159.105.23.130 -> 192.168.1.3
ICMP TTL:248 TOS:0x0 ID:11560 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Set 1 -TCPDump log

```
10:33:54.909418 eth0 P 159.105.23.130 > mailsrv: icmp: echo request
(DF)
10:33:54.909418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: S
2438921473:2438921473(0) win 16384 <mss 512>
10:33:54.949418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: .
2438921474:2438921474(0) ack 12355218 win 16384
10:33:55.009418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: P 0:22(22)
ack 94 win 16384 (DF)
10:33:55.039418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: . 22:22(0)
ack 223 win 16384 (DF)
10:33:55.049418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: P 22:70(48)
ack 223 win 16384 (DF)
10:33:55.079418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: P 70:96(26)
ack 297 win 16384 (DF)
10:33:55.109418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: P 96:102(6)
ack 334 win 16384 (DF)
10:33:55.169418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: .
102:614(512) ack 370 win 16384 (DF)
10:33:55.169418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: .
614:1126(512) ack 370 win 16384 (DF)
```

[9] http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=large+icmp

```
10:33:55.179418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: P
1126:1307(181) ack 370 win 16384 (DF)
10:33:55.319418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: P
1307:1313(6) ack 378 win 16384 (DF)
10:33:55.349418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: F
1313:1313(0) ack 402 win 16360 (DF)
10:33:55.379418 eth0 P 159.105.23.130.51859 > mailsrv.smtp: .
1314:1314(0) ack 403 win 16359 (DF)
```

Set 2 -Snort Alert

```
[**] MISC Large ICMP Packet [**]
02/20-10:36:22.719418 159.105.23.130 -> 192.168.1.3
ICMP TTL:248 TOS:0x0 ID:29173 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Set 2 -TCPDump log

```
10:36:22.719418 eth0 P 159.105.23.130 > mailsrv: icmp: echo request
(DF)
10:36:22.719418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: S
51279805:51279805(0) win 16384 <mss 512>
10:36:22.779418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: .
51279806:51279806(0) ack 12355293 win 16384
10:36:22.839418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: P 0:22(22)
ack 94 win 16384 (DF)
10:36:22.879418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: P 22:72(50)
ack 223 win 16384 (DF)
10:36:22.909418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: P 72:97(25)
ack 299 win 16384 (DF)
10:36:22.949418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: P 97:103(6)
ack 335 win 16384 (DF)
10:36:23.009418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: .
103:615(512) ack 371 win 16384 (DF)
10:36:23.019418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: .
615:1127(512) ack 371 win 16384 (DF)
10:36:23.019418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: P
1127:1353(226) ack 371 win 16384 (DF)
10:36:23.119418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: .
1353:1353(0) ack 379 win 16376 (DF)
10:36:23.159418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: P
1353:1359(6) ack 379 win 16384 (DF)
10:36:23.179418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: F
1359:1359(0) ack 403 win 16360 (DF)
10:36:23.229418 eth0 P 159.105.23.130.52100 > mailsrv.smtp: .
1360:1360(0) ack 404 win 16359 (DF)
```

As can be seen from the exhibit, an ICMP echo request with an Datagram Length of 1500. Immediately preceded the normal transfer of SMTP traffic to the networks mail server. Upon further research, I discovered that AIX will try to set packet size before transferring data. The scan of the week from http://project.honeynet.org/scans/arch/scan4.txt .

Attack Mechanism

This is path MTU discovery, not an attack.

Evidence of Active Targeting

It certainly was active targeting, however the behavior, however, annoying, is by design.

Correlations

There are a number of anecdotal reports of this mechanism. Since it is not an attack, I suggest reading

http://project.honeynet.org/scans/arch/scan4.txt

or

http://lists.insecure.org/incidents/2001/Jul/0275.html

for further information.

Severity

Criticality – Not Applicable. Criticality = 0.

Lethality – The structure of the scan might be replicated to disguise a true scan. Many routers and firewalls are being configured to drop ICMP echo requests anyway. Lethality = 2.

System Countermeasures – The Operating System on the email server remains current on all patches.. System Countermeasures = 5

Network Countermeasures – A permissive firewall allowed the discovery to proceed. Network Countermeasures = 2

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

(0 + 2) – (5 + 2) = -5

Defensive Recommendations

The DMZ might be configured to drop ICMP packets. This organization prefers to respond to ICMP echo requests.

Multiple Choice Question:

The large ICMP Packet probe is used :

5) to find live hosts on a given subnet
6) to create a DOS condition on a firewall
7) to find a path's maximum transmission unit
8) to communicate with trinoo zombies without IDS detection

Detect #5

Source of Detect:

This scan came from an employer's network.

Detect was generated by:

Specter 6.0 mailed an alert to me. Specter starts by explaining its current configuration as set by the administrator. Most options are self-explanatory.

```
************************************************************
  System name : Gollum
  Config file version : 1.0
  Maximum connections : 5
  Connection throttle : on
  Connections/min. : 10
  Flood blocking : off
  Send status mail : no
  Send mails : yes
  Send short mails : no
  Log to files : yes
  Log to event log : no
  Log to syslog : no
  Do finger probe : no
  Do port scan : yes
  Whois lookup : no
  Log telnet banner : no
  Log ftp banner : no
  Log smtp banner : no
  Log http document : yes
  Log http header : yes
  Custom warning msg. : no
  Custom POP3 msg. : no
  Provide POP3 msg. : no
```

```
  Use web graphics : no
  Use custom web doc. : no
  Expect friendly con. : no
  Remote management : yes
  Remote mgmt. port : 28
  Trace route : no
  Do reverse lookup : yes
  Send password files : yes
  Password type : normal
  Activated services : FTP TELNET SMTP POP3 NETBUS FINGER HTTP
  Activated traps : DNS SUN-RPC SUBSEVEN SSH IMAP BO2K
  Mail Server : xxx.xxx.xxx.xxx
  Mail Address : email@my.com
  Short Mail Address : email@my.com

  Role OS : Linux
  Role Character : Strange System
  Role Hostname : Gollum.client.org
  Crowd Level : Multiple users
  User Names : Default
************************************************************

Port scan information :
Found 2 active port(s) on host 62.248.238.25 at Thu Feb 21 09:01:16
2002

Active ports:
 22  ssh
 513  who

************************************************************

HTTP server header :

Could not get http server header.

HTTP server document :
HTML document was not logged.

************************************************************

SSH TRAP connection
Host  : 62.248.238.25  (ua25d41hel.dial.kolumbus.fi)
Time  : Thu Feb 21 09:00:03 2002
```

Probability that the address was spoofed

This is a scan for ssh hosts. The IP is not likely to be spoofed

Description of the Attack:

This is an ssh server scan run against a subnet. The attacker is using scanssh to harvest ssh addresses and versions.

Scanssh scans the Internet for SSH server versions …
(it) scans the given addresses and networks for running SSH servers. It will query their version number and displays the results in a list.[10]

Secure log on ssh Server

The attack logged to secure log.

```
Feb 21 09:09:03 roac sshd[4123]: Did not receive identification string
from 62.248.238.25.
```

[root@roac log]# tcpdump -r tcpdump19 'host 62.248.238.25'

The tcpdump log has been abbreviated for clarity. As mentioned earlier, it swept the subnet of publicly available IP's. The attacker (62.248.238.25) is sending a syn to each IP address, the host replies with a reset.

```
09:08:55.689418 eth0 P 62.248.238.25.ssh > server.ssh: S
211681249:211681249(0) win 6275
09:08:55.689418 eth0 P server.ssh > 62.248.238.25.ssh: R 0:0(0) ack
211681250 win 0

09:08:55.689418 eth0 P 62.248.238.25.ssh > mailsrv.ssh: S
171477926:171477926(0) win 6275
09:08:55.689418 eth0 P mailsrv.ssh > 62.248.238.25.ssh: R 0:0(0) ack
171477927 win 0

09:08:55.689418 eth0 P 62.248.238.25.ssh > citrix.ssh: S
1947986516:1947986516(0) win 6275
09:08:55.689418 eth0 P citrix.ssh > 62.248.238.25.ssh: R 0:0(0) ack
1947986517 win 0

09:08:55.699418 eth0 P 62.248.238.25.ssh > Logsrv.ssh: S
2151049373:2151049373(0) win 6275
09:08:55.699418 eth0 P Logsrv.ssh > 62.248.238.25.ssh: R 0:0(0) ack
2151049374 win 0
```

Attack Mechanism

This snippet of the log shows a successful harvesting of an ssh server.

[10] man page, scanssh v.1-0

First the scanning script (BadGuy) notices a syn syn/ack connection.

```
09:08:55.699418 eth0 < 62.248.238.25.ssh > roac.ssh: S
1869507506:1869507506(0) win 6275
09:08:55.699418 eth0 > roac.ssh > 62.248.238.25.ssh: S
1216889638:1216889638(0) ack 1869507507 win 5840 <mss 1460> (DF)
```

Now that the Bad Guy has found the ssh server it opens a connection.

```
09:08:56.359418 eth0 < 62.248.238.25.1210 > roac.ssh: S
1217625634:1217625634(0) win 32120 <mss 1380,sackOK,timestamp 560436052
0,nop,nop,nop,nop> (DF)
09:08:56.359418 eth0 > roac.ssh > 62.248.238.25.1210: S
1217267333:1217267333(0) ack 1217625635 win 5792 <mss
1460,sackOK,timestamp 34404533 560436052> (DF)
09:08:56.509418 eth0 < 62.248.238.25.1210 > roac.ssh: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 560436068 34404533> (DF)
```

Lastly Roac (ssh server) foolishly pushes text “SSH-1. 99-OpenSSH_2.5.2 p2.:” to bad guy for harvesting then politely closes the connection.

```
09:08:56.619418 eth0 > roac.ssh > 62.248.238.25.1210: P 1:26(25) ack 1
win 5792 <nop,nop,timestamp 34404559 560436068> (DF)
09:08:56.769418 eth0 < 62.248.238.25.1210 > roac.ssh: . 1:1(0) ack 26
win 32120 <nop,nop,timestamp 560436094 34404559> (DF)
09:09:03.789418 eth0 < 62.248.238.25.1210 > roac.ssh: F 1:1(0) ack 26
win 32120 <nop,nop,timestamp 560436795 34404559> (DF)
09:09:03.789418 eth0 > roac.ssh > 62.248.238.25.1210: F 26:26(0) ack 2
win 5792 <nop,nop,timestamp 34405276 560436795> (DF)
09:09:03.929418 eth0 < 62.248.238.25.1210 > roac.ssh: . 2:2(0) ack 27
win 32120 <nop,nop,timestamp 560436810 34405276> (DF)
```

Correlations

Secure shell remains among the top 10 ports scanned. Figure 4 shows the incidents.org report at the time of writing.

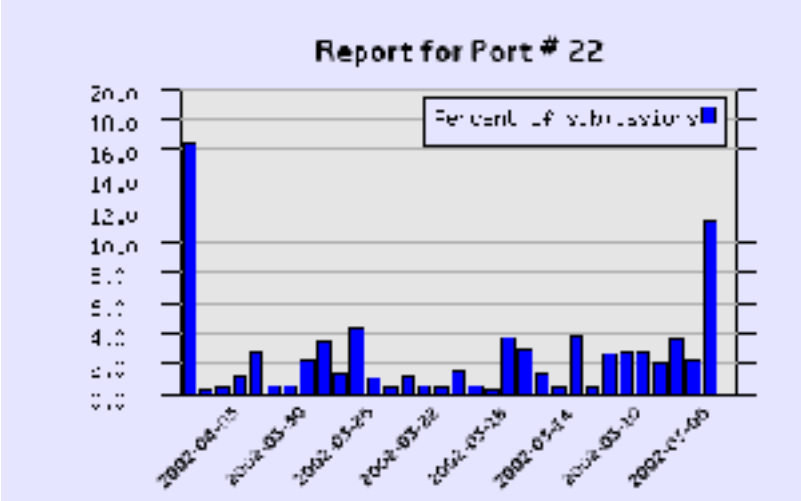


Figure 4 – Activity on port 22

Furthermore, there are numerous anecdotal reports of scanssh leaving its calling card. Examples include:

http://lists.insecure.org/incidents/2001/Dec/0246.html
http://cert.uni-stuttgart.de/archive/incidents/2001/12/msg00245.html
http://www1.dshield.org/pipermail/list/2001-December/002310.html

Evidence of Active Targeting

This is a scan that blasted through the subnet. No evidence of active targeting.

Severity

Criticality – The only unit susceptible to the SSH scan was the sniffer. It may be used as a lauch point for attacks on the DMZ machines or the internet, hence it receives its rating. Criticality = 2

Lethality – The scan attack succeeded in finding one host. The scanning agent could return with an attack to compromise that host, however we are focusing on the scan attack. Lethality = 2

System Countermeasures – The Operating System and the sshd version were current on all patches. The ssh is version 1 and is susceptible to attack. System Countermeasures = 3

Network Countermeasures – A permissive firewall allowed the discovery to proceed. Network Countermeasures = 2

(Criticality + Lethality) –

(System Countermeasures + Network Countermeasures) = Severity

(2 + 2) – (3 + 2) = -1

Defensive Recommendations

The only ssh server has since been removed from the network. Additional countermeasures that may be implemented include:

- Silently dropping syn packets arriving on TCP22 via the router or firewall
- Setting an alert on the NIDS watching for outgoing SSHD version identification.
- Silently dropping *all* unnecessary syn packets

Multiple Choice Question:

The purpose of the Scanssh attack is (Choose 2)

1) to find live hosts on a given subnet
2) to find live ssh servers on a given subnet
3) to launch a series of attacks once a host is identified
4) to log the version of sshd being run

Assignment 3– "Analyze This" Scenario

University Security Audit

The University provided Snort Intrusion Detection System logs for the third week January and requested an audit of network activity..

Executive Summary

This Audit provides analyses of alerts, scans, and out of specification (OOS) packet data provided by the University for the period of Jan 21is through Jan 25th, 2002. During this timeframe, there were 115,329 Snort Intrusion Detection System (IDS) alerts, 1,246,797 scans, and 27 OOS packets detected.
Alerts whose frequency were greater than 1000 in count were analyzed and specific defensive recommendations were offered.

Data Included in Analysis

An analysis of the alert, scan and out of spec files was conducted for the week of January 21-25, 2002. At the time of writing, these data were within 60 days of the analysis period as required by the assignment. The data used in this analysis are shown in Table 1.

Alerts	OOS	Scans
Alert.020121	oos_Jan.21.2002	Scans.020121
Alert.020122	oos_Jan.22.2002	Scans.020122
Alert.020123	oos_Jan.23.2002	Scans.020123
Alert.020124	oos_Jan.24.2002	Scans.020124
Alert.020125	oos_Jan.25.2002	Scans.020125

Table 0 – Files used for Audit

List of Detects

Summary of Alerts	
Signature	**Frequency**
connect to 515 from inside	31425
SNMP public access	25657
spp_http_decode: IIS Unicode attack detected	18668
MISC Large UDP Packet	16804
INFO MSN IM Chat data	5189
spp_http_decode: CGI Null Byte attack detected	3716
High port 65535 udp - possible Red Worm - traffic	3622
ICMP Router Selection	1701
ICMP Echo Request CyberKit 2.2 Windows	1486
ICMP Fragment Reassembly Time Exceeded	1116
Null scan!	1021
Watchlist 000220 IL-ISDNNET-990517	802
ICMP Echo Request BSDtype	728
SMB Name Wildcard	700
ICMP Echo Request L3retriever Ping	432
FTP DoS ftpd globbing	375
ICMP Echo Request Windows	266
WEB-IIS view source via translate header	194
ICMP Destination Unreachable (Communication Administratively Prohibited)	164
NMAP TCP ping!	121
ICMP Destination Unreachable (Host Unreachable)	112
WEB-MISC Attempt to execute cmd	106
SCAN Proxy attempt	91
ICMP Echo Request Nmap or HPING2	80
Incomplete Packet Fragments Discarded	75
INFO FTP anonymous FTP	69
EXPLOIT NTPDX buffer overflow	68
Possible trojan server activity	58

Summary of Alerts	
Signature	**Frequency**
INFO Inbound GNUTella Connect request	57
INFO Possible IRC Access	56
ICMP Destination Unreachable (Protocol Unreachable)	52
MISC traceroute	47
WEB-CGI scriptalias access	42
SCAN Synscan Portscan ID 19104	34
WEB-IIS _vti_inf access	23
WEB-FRONTPAGE _vti_rpc access	22
INFO Inbound GNUTella Connect accept	21
ICMP traceroute	13
Port 55850 tcp - Possible myserver activity - ref. 010313-1	12
SCAN FIN	11
TCP SRC and DST outside network	11
SUNRPC highport access!	9
Attempted Sun RPC high port access	6
WEB-MISC compaq nsight directory traversal	6
Queso fingerprint	6
INFO Outbound GNUTella Connect accept	5
EXPLOIT x86 setuid 0	5
High port 65535 tcp - possible Red Worm - traffic	4
SYN-FIN scan!	4
TFTP - External UDP connection to internal tftp server	4
WEB-MISC 403 Forbidden	4
EXPLOIT x86 NOOP	3
INFO - Possible Squid Scan	3
EXPLOIT x86 setgid 0	3
ICMP SRC and DST outside network	3
ICMP Source Quench	3
Back Orifice	2
ICMP Echo Request Cisco Type.x	2
Probable NMAP fingerprint attempt	2
Port 55850 udp - Possible myserver activity - ref. 010313-1	1
TFTP - Internal UDP connection to external tftp server	1
MISC Large ICMP Packet	1
WEB-MISC http directory traversal	1
MISC source port 53 to <1024	1
RFB - Possible WinVNC - 010708-1	1
INFO Napster Client Data	1
RPC udp traffic contains bin sh	1

Connect 515 from inside

Connect 515 is a frequently occurring scan. Chart 1 from Incidents.org shows a consistent reporting level.

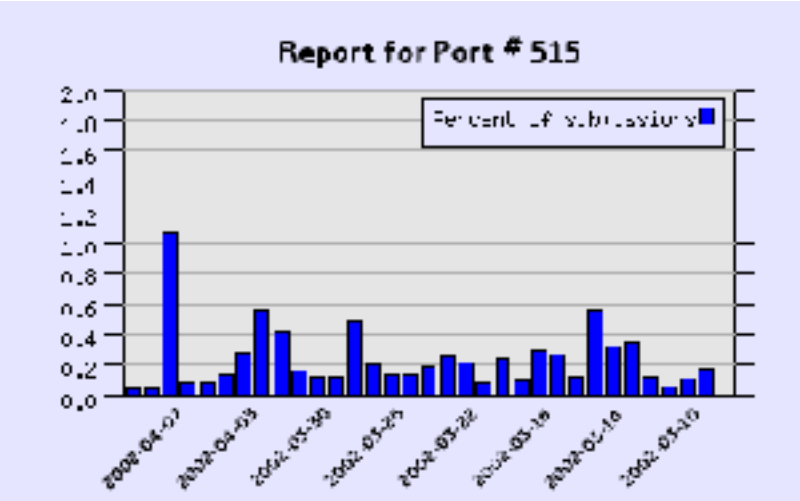


Chart 1 – Port 515 scans

Port 515 is the lpr port. The attackers could be trying to find a buffer overflow in LPRng or LPDng. LPR has had a number of vulnerabilities identified including format string overflows that allow the user to run as lp (or worse).
While this alert occurred 31,425 times and accounted for almost 21% of the alerts, it may well be harmless. Table 2 shows that the top ten alerters are all internal. All 474 alerters were internal hosts, probably using the print services.

Top ten Connect 515 Alerter	
Count	**Description**
2285	MY.NET.153.118:1188 -> MY.NET.150.198:515
1633	MY.NET.153.113:3184 -> MY.NET.150.198:515
938	MY.NET.153.114:1825 -> MY.NET.150.198:515
915	MY.NET.153.114:1823 -> MY.NET.150.198:515
580	MY.NET.153.112:1788 -> MY.NET.150.198:515
439	MY.NET.153.111:2933 -> MY.NET.150.198:515
436	MY.NET.153.112:2599 -> MY.NET.150.198:515
420	MY.NET.153.160:1683 -> MY.NET.150.198:515
296	MY.NET.88.148:1151 -> MY.NET.150.198:515
292	MY.NET.88.148:1147 -> MY.NET.150.198:515

Table 2 – Top Ten Alerters

Countermeasure:

Enable ingress filtering to port 515, if not already in place. Ensure that all hosts have current patches, if possible.

Correlations:

David Hed
http://www.giac.org/practical/David_Ded_GCIA.zip
Scott Shinberg
http://www.giac.org/practical/Scott_Shinberg_GCIA.doc
Lorraine Weaver
http://www.giac.org/practical/Lorraine_Weaver_GCIA.zip

SNMP public access

Simple Network Management Protocol allows data gathering and trapping to occur on network devices such as router and switches. There are any number of opportunities for misconfigurations including no or known passwords. Additionally, a student from Finland announced a vulnerability that would crash SNMP devices with one packet. The tools was released before manufacturers had a chance to react.
The traffic, as can be seen in Table 3 shows that all SNMP alerts are internal. There is not enough information to determine whether or not this is traffic originating from network appliances or consoles. In any case, they should not be using the public string.
Table 3 show the top ten alerters for Public Access.

SNMP Public Access	
Count	**Description**
7128	MY.NET.88.240:1026 -> MY.NET.150.195:161
2256	MY.NET.150.41:1027 -> MY.NET.152.109:161
1716	MY.NET.70.177:1070 -> MY.NET.5.96:161
1683	MY.NET.70.177:1070 -> MY.NET.5.128:161
1669	MY.NET.70.177:1070 -> MY.NET.5.127:161
1602	MY.NET.70.177:1070 -> MY.NET.5.37:161
1350	MY.NET.70.177:1070 -> MY.NET.5.249:161
1298	MY.NET.150.198:1025 -> MY.NET.151.114:161
929	MY.NET.70.177:1070 -> MY.NET.5.141:161
797	MY.NET.153.220:1245 -> MY.NET.152.109:161

Table 3 - Top Ten Alerters for Public Access

Countermeasures:

The best one is to NOT run SNMP. If you must check the manufacturers for fixes, ensure that you have changed the config's on your devices from *public*, and ensure that your devices are not internet accessible.

Correlations:

David Hed
http://www.giac.org/practical/David_Ded_GCIA.zip

Scott Shinberg
http://www.giac.org/practical/Scott_Shinberg_GCIA.doc
Chris Baker
http://www.giac.org/practical/Chris_Baker_GCIA.zip

spp_http_decode: IIS Unicode attack detected

The Unicode attack is a vulnerability in Internet Information Server 4.0 and 5.0. Its CVE is CVE-2000-0884. The vulnerability allows directories to be traversed and viewed and commends to be run in the context of IISUR_machinename. It takes advantage of the (..) directory traversal commands and may be considered dangerous.
Table 4 shows the top five internal Unicode scanners.

Host	Count	Victim Address
My.Net.152.14	1531	211.115.231.202
My.Net.153.141	1411	211.115.213.202
My.Net.153.114	181	211.32.117.31
My.Net.153.110	140	211.32.117.26
My.Net.153.151	125	216.33.148.250

Table 4 – Top five internal attackers

The top five victims are registered at Arin.net and the Asia Pacific net as belonging to:

211.115.231.202 211.115.213.202	
inetnum	211.104.0.0 - 211.119.255.255
netname	KRNIC-KR
descr	KRNIC
descr	Korea Network Information Center
country	KR
admin-c	HM127-AP, inverse
tech-c	HM127-AP, inverse

remarks	**
remarks	KRNIC is the National Internet Registry
remarks	in Korea under APNIC. If you would like to
remarks	find assignment information in detail
remarks	please refer to the KRNIC Whois DB
remarks	http://whois.nic.or.kr/english/index.html
remarks	**
mnt-by	APNIC-HM, inverse
mnt-lower	MNT-KRNIC-AP, inverse
changed	hostmaster@apnic.net 20000414
changed	hostmaster@apnic.net 20010606
source	APNIC

211.32.117.31
211.32.117.26

inetnum	211.32.0.0 - 211.39.255.255
netname	KRNIC-KR
descr	KRNIC
descr	Korea Network Information Center
country	KR
admin-c	HM127-AP, inverse
tech-c	HM127-AP, inverse
remarks	**
remarks	KRNIC is the National Internet Registry
remarks	in Korea under APNIC. If you would like to
remarks	find assignment information in detail

remarks	please refer to the KRNIC Whois DB
remarks	http://whois.nic.or.kr/english/index.html
remarks	**
mnt-by	APNIC-HM, inverse
mnt-lower	MNT-KRNIC-AP, inverse
changed	hostmaster@apnic.net 19990827
changed	hostmaster@apnic.net 20010606
source	APNIC

216.33.148.250

Exodus Communications Inc. (NETBLK-ECI-7)

1605 Wyatt Dr. Santa Clara, CA

95054US

US

Netname: ECI-7

Netblock: 216.32.0.0 - 216.35.255.255

Maintainer: ECI

Coordinator:

Center, Network Control (NOC44-ARIN) ipaddressadmin@exodus.net

(888) 239-6387 (FAX) (888) 239-6387

Domain System inverse mapping provided by:

DNS01.EXODUS.NET	209.1.222.244
DNS02.EXODUS.NET	209.1.222.245

DNS03.EXODUS.NET	209.1.222.246
DNS04.EXODUS.NET	209.1.222.247

Countermeasures:

The patches have long since been available at:

Microsoft IIS 4.0:
http://www.microsoft.com/ntserver/nts/downloads/critical/q269862
Microsoft IIS 5.0:
http://www.microsoft.com/windows2000/downloads/critical/q269862

Correlations:

Gregory LaJon
http://www.giac.org/practical/Gregory_LaJon_GCIA.doc

MISC Large UDP Packet

These could be signs of gaming, not too surprising for a university. The alert is triggered when a UDP datagram exceeds 400 bytes.

Top five Gamers	
Count	**Description**
3878	63.210.47.81:44230 -> MY.NET.153.45:1221
1539	211.172.232.21:2106 -> MY.NET.153.144:1992
821	63.210.47.81:0 -> MY.NET.153.45:0
763	216.106.166.212:20352 -> MY.NET.153.45:1742
709	211.202.0.47:1663 -> MY.NET.153.171:4442

Evaluating the activity of our top reporter received 3878 packets across the web. It is interesting to note that this is not two-way traffic. The 3878 packets were distributed across 44 minutes of session time making it likely that it is not an automated scan/attack.

Correlations:

Gregory LaJon
http://www.giac.org/practical/Gregory_LaJon_GCIA.doc

INFO MSN IM Chat data

This is an information message on non-suspicious *Microsoft Internet Messenger* traffic. It is triggered by port 1863 traffic

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 1863 (msg:"INFO MSN IM Chat
data";flags: A+; content:"|746578742F706C61696E|"; depth:100;
classtype:not-suspicious; sid:540; rev:1;)
```

This traffic, while frequent, is happening at keystroke level. Table 5 shows a sample of two-way communication between our parties. These data are typical of the traffic seen from the reporting period.

Info MSN Chat	
Time	**Description**
01/21-10:40:45.084434	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:40:50.880446	MY.NET.150.165:1361 -> 64.4.12.177:1863
01/21-10:40:53.826723	MY.NET.150.165:1361 -> 64.4.12.177:1863
01/21-10:40:59.109783	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:41:21.043209	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:41:37.051860	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:41:52.698555	MY.NET.150.165:1361 -> 64.4.12.177:1863
01/21-10:42:00.945587	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:42:02.946363	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:42:06.415019	64.4.12.177:1863 -> MY.NET.150.165:1361
01/21-10:42:24.465985	MY.NET.150.165:1361 -> 64.4.12.177:1863

Table 5 – Traffic from MSN Chat

Whether or not the traffic should be allowed is an issue for the University's InfoSec policy group. Doubtless blocking chats would cause an uproar and allegations of University collaboration with the phone company.

Countermeasures:

An ACL on Cisco firewalls (I hold a CCNA) would look like this
access-list 101 deny tcp any any eq 1863

Correlations:

Mike Poor
http://www.giac.org/practical/Mike_Poor_GCIA.doc

spp_http_decode: CGI Null Byte attack detected

This is a recently observed attack, cited by rain forest puppy. This attack masks system commands behind the null byte %00, a packet that CGI scripts don't typically watch for. Among the threats is an upload bomb that can fill available disk space. Of course, if the bad guys can upload, there is a potential for warez

storage, etc. This is a potentially dangerous alert and University Machines are the attackers. The top two hosts listed below own the majority of alerts.

Attacker	Count	Victim
My.Net.150.121	2982	216.241.219.14:80
My.Net.153.194	310	209.143.193.79:80

Countermeasures:

According to the many authors at
http://www.linuxsecurity.com/resource_files/intrusion_detection/snort-FAQ-1.8.txt
Having the packet dumps is the only way to tell for sure if you have a real attack on your hands, but this is true for any content-based alert.
I've not seen this attack but content filtering and good programming practices seem to be in order. According to RFP, free perl scripts may be the first thing to check.
It would be valuable to identify these two hosts. Perhaps the users can be identified and discouraged from illicit activity. It would be nice to think that this is research against external test machines.

Correlations:

http://www.wiretrip.net/rfp/p/doc.asp/i8/d37.htm
http://www.wiretrip.net/rfp/p/doc.asp/i2/d6.htm

High port 65535 udp – possible Red Worm - traffic

Also known as the *Adore* worm, red worm looks for Linux hosts exhibiting vulnerable rpc-statd, BIND, wu-ftp and LPRng daemons. Red worm places a Trojan version of ps on the victim and then sends email with system files attached to several email addresses. It sets a back door listener and a rootshell. Then it removes itself and reboots your system. This attack should be considered dangerous.

Countermeasures:

Infected hosts may be identified and removed by a routine written at the Dartmouth ISTS
http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm
To prevent infection, Access Control lists may be placed on your routers.
http://www.sans.org/y2k/lion_protection.htm

Correlations:

Michael Reiter
http://www.sans.org/y2k/practical/Michael_Reiter_GCIH.zip

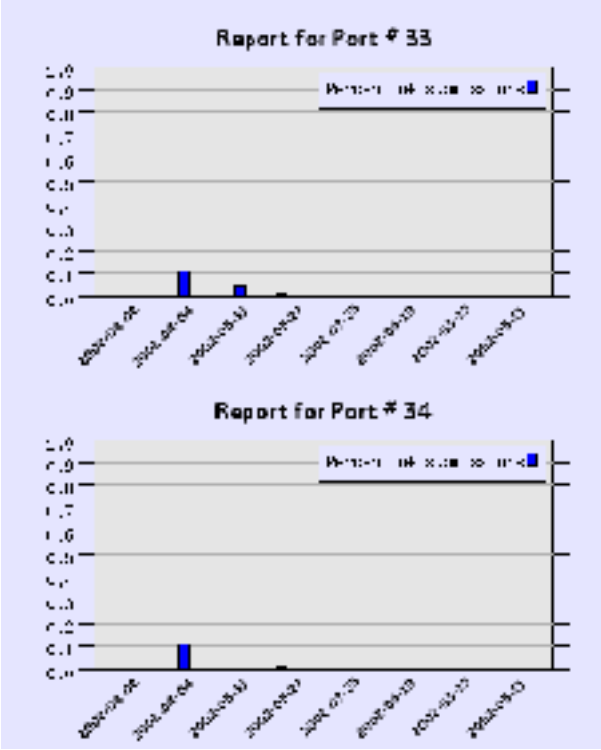
ICMP Echo Request CyberKit 2.2

CyberKit is a reconnaissance tool. From a Windows GUI, the user can perform DNS lookups, traceroute, whois , and etc.
The host listed below did most of his/her scanning in two minutes attacking port 33 Display Support Protocol and port 34 unassigned.

ICMP Echo Request	
Count	**Description**
733	MY.NET.150.49 -> 204.71.200.33
729	MY.NET.150.49 -> 204.71.200.34

Countermeasures:

A stateful inspection firewall will block fast scans. There are no correlations in reviewed papers. This incident occurred three times from two different hosts and may be considered low priority. Further, the CID database reports no major activity against these two ports, as shown below:



Correlations:

None

ICMP fragment reassembly time exceeded

If the router processing a datagram finds the time to live field set to zero it must discard the datagram. The router may set type 11 to code = 1 and notify that the reassembly time was exceeded. This can be due to fragments being "lost". The low frequency may indicate a network problem. The alert may also be generated when fragmenting to identify a firewall.

ICMP Fragmentation Attacks	
Count	**Description**
334	MY.NET.153.159 -> 211.234.110.20
119	MY.NET.153.171 -> 211.174.63.106
88	MY.NET.88.137 -> 210.158.194.98
84	MY.NET.153.45 -> 208.172.128.163
70	MY.NET.153.197 -> 211.234.110.20

Countermeasures:

These attacks are internally based. Egress filtering should be applied to border routers.

Correlations:

None!
MountAraratBlossom describes

Null Scan!

A null scan is a packet that contains a TCP packet payload with none of the control bits are set. Also the sequence number is set to zero. This is a crafted packet whose purpose is to either avoid IDS detection (or router ACL) from a syn scan alert. Also, they may be used to fingerprint a system/'s operating system.

Countermeasures:

Most firewalls will allow for some sort of detection. For instance the PIX series detect is 3015.

Correlations:

David Singer
http://www.sans.org/y2k/practical/David_Singer_GCIA.doc

Lorraine Weaver
http://www.giac.org/practical/Lorraine_Weaver_GCIA.zip

Out of Spec Analysis

Out of Spec (OOS) packets are packets which do not conform to the guidelines in the applicable RFC. Operating system designers do not anticipate these permutations., As a result, each operating system may respond in unique (or disastrous) ways to the unexpected These packets are often crafted to aid in identifying a target host or to simply crash the attacked host.
This analysis period had light OOS traffic. There were 27 OOS packets recorded by the snort system during the analysis period. The top four are listed below. Among the other entries, each of the six records were unique.

IP Address	Frequency
24.180.218.241	11
24.234.240.101	5
24.73.8.140	3
129.81.155.31	2

A common thread

Chart 1 shows that all external hosts were linking to 2 University hosts. In all cases there was at least one packet to port 1214.

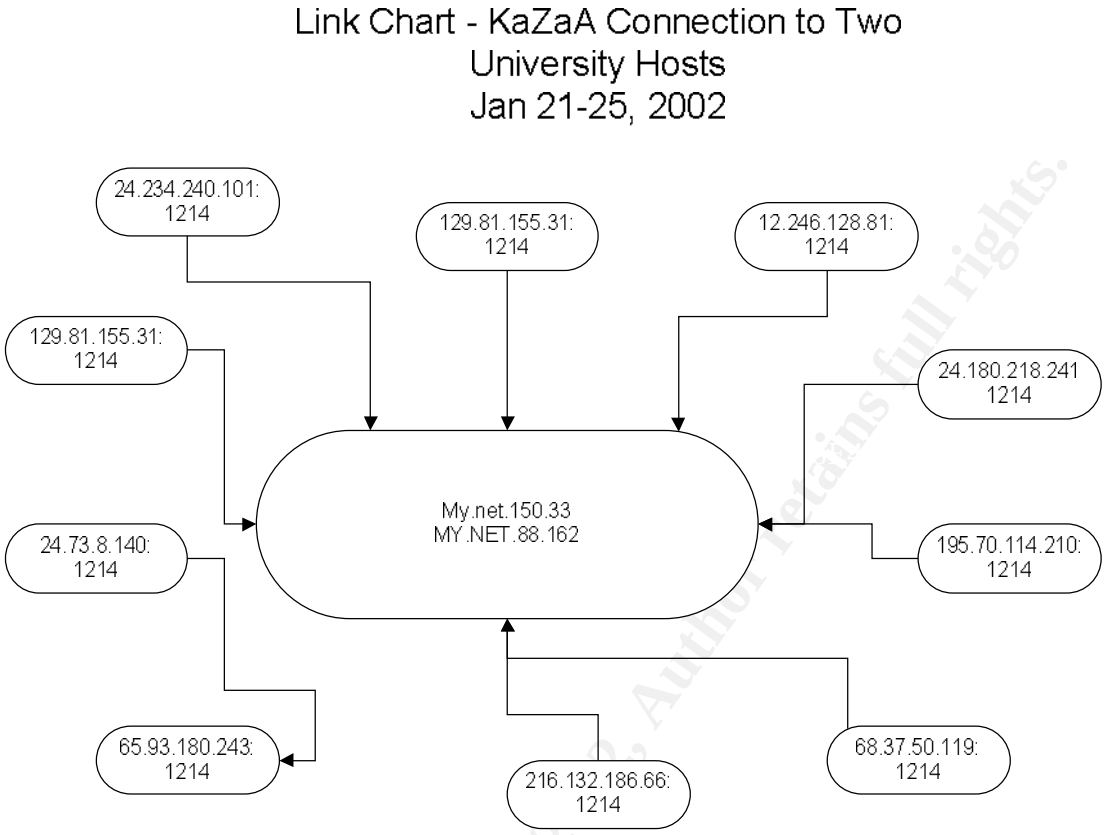


Chart 1 – Links to hosts from out of spec alerts

Both hosts received multiple traffic on the KaZaA port (1214). KaZaA is a popular peer-to-peer file sharing client. It has been in the top 10 scan ports for weeks. Figure 5 shows the most current traffic

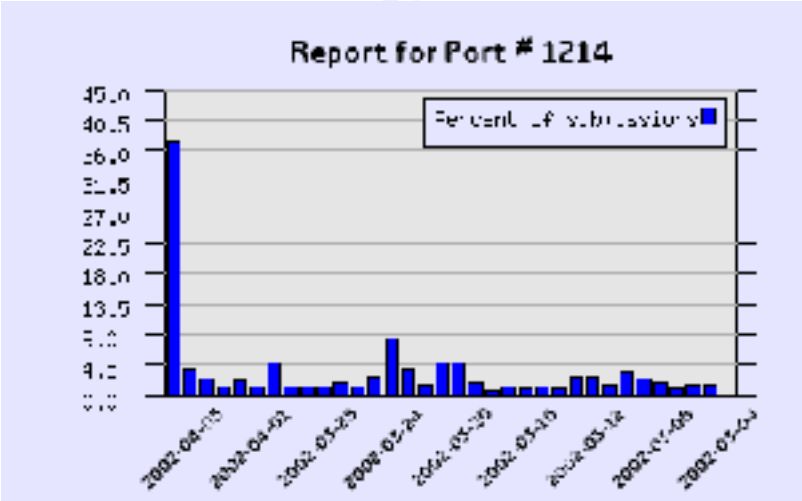


Figure 5 – CID scan reports for KaZaA port

During the period in which this paper was written KaZaA has been in the top ten, at either position six or seven.

It is highly likely that these two hosts are KaZaA peers. The scans that we see likely have one or two purposes:

1) Avoid network detection are likely packets crafted to avoid detection of the firewalls.

2) They are sending packets crafted to crash the KaZaA systems.

Evaluating the packets from the most frequent scan we see that the top assailant hails from the @Home Network.

```
@Home Network (NETBLK-HOME-2BLK)HOME-2BLK         24.176.0.0 -
24.183.255.255
@Home Network (NETBLK-PHLAPA1-PA-3) PHLAPA1-PA-3 24.180.208.0 -
24.180.223.255
```

The top Assailants packets are shown in Table 6

Top Assailant				
SrcIP	**SrcPort**	**DestIP**	**DestPort**	**opts**
24.180.218.241:	0	MY.NET.88.162:	2259	*1SF**A*
24.180.218.241:	0	MY.NET.88.162:	2173	*1SF**A*
24.180.218.241:	2342	MY.NET.88.162:	1214	21**R*AU
24.180.218.241:	1796	MY.NET.88.162:	1214	21**RP**
24.180.218.241:	13	MY.NET.88.162:	1661	21**RP*U
24.180.218.241:	1796	MY.NET.88.162:	1214	21**RPAU
24.180.218.241:	166	MY.NET.88.162:	1796	21**RPAU
24.180.218.241:	2342	MY.NET.88.162:	1214	21*FR**U
24.180.218.241:	1522	MY.NET.88.162:	1214	21S*****
24.180.218.241:	1449	MY.NET.88.162:	1214	21SFR***
24.180.218.241:	2173	MY.NET.88.162:	1214	21SFRP*U

Table 6 Packet detail of top OOS assailant

The attacker crafted a series of 11 OOS packets targeting host my.net.88.162. The preponderance of destination ports were 1214, popularly used by KaZaA. The reconnaissance occurred over several hours on Jan 23rd at sporadic times indicating that crafted packets were sent.

Packets similar to the exhibit below were reported in the handler's diary at http://www.sans.org/y2k/022100.htm . David Singer reported similar packets sets and suggested a possible insertion attack.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-11:04:37.124400 24.180.218.241:1449 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:26186  DF
21SFR*** Seq: 0xEF0852   Ack: 0xE04BA843   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-11:21:37.108327 24.180.218.241:1522 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:5049  DF
21S***** Seq: 0x860EA65   Ack: 0x26B814   Win: 0x8010
00 00 01 01 05 0A B8 14 63 76 B8 14              ........cv..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-11:36:54.343888 24.180.218.241:13 -> MY.NET.88.162:1661
TCP TTL:112 TOS:0x0 ID:55507  DF
21**RP*U Seq: 0x4BE086C   Ack: 0xCDEDC59A   Win: 0x5010
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-11:45:07.298404 24.180.218.241:166 -> MY.NET.88.162:1796
TCP TTL:112 TOS:0x0 ID:48718  DF
21**RPAU Seq: 0x4BE0876   Ack: 0xD75D005   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-11:45:14.004829 24.180.218.241:1796 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:31824  DF
21**RPAU Seq: 0xA60876   Ack: 0xD75D006   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL EOL NOP NOP NOP SackOK EOL EOL
EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-11:45:20.899317 24.180.218.241:1796 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:17746  DF
21**RP** Seq: 0x8760D75   Ack: 0xA6D008   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK NOP NOP TS: 0 0 EOL EOL
EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-13:03:57.619978 24.180.218.241:2173 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:51381  DF
21SFRP*U Seq: 0xCB08BB   Ack: 0x91611CE2   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-13:22:13.297592 24.180.218.241:0 -> MY.NET.88.162:2173
TCP TTL:112 TOS:0x0 ID:21326  DF
*1SF**A* Seq: 0x4BE08BB   Ack: 0x91611D1B   Win: 0x5010
00 00 08 7D 04 BE 08 BB 91 61 1D 1B 0B 93 50 10  ...}.....a....P.
22 38 EE F9 00 00 00 00 00 00                    "8........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-13:31:06.386445 24.180.218.241:0 -> MY.NET.88.162:2259
TCP TTL:112 TOS:0x0 ID:18581  DF
21*F*PAU Seq: 0x4BE08D5   Ack: 0xE89A3AB0   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-14:00:40.024927 24.180.218.241:2342 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:54316  DF
21**R*AU Seq: 0xEF08EB   Ack: 0xAB705306   Win: 0x5010
3D F4 50 10 22 08 6B F5 00 00 00 00 00 00        =.P.".k.......
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/23-14:06:49.148192 24.180.218.241:2342 -> MY.NET.88.162:1214
TCP TTL:112 TOS:0x0 ID:36676  DF
21*FR**U Seq: 0x8EBAB70   Ack: 0xD5306   Win: 0x5010
00 00 00 00 00 00                                 ......

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

Counter Measures
If there were a security/usage policy against KaZaA there would be router ACLs in place to deny port 1214. A Cisco example is shown below:

access-list 101 deny tcp any any eq 1214

There are only two of these hosts. Perhaps an undergraduate assistant can be sent ot chase them down an apply the latest version of upgrade (1.6 at the time of writing).
These packets represent a miniscule percentage of the total alerts received. The University may choose to ignore this incident.

Scan Analysis

Snort logged 1, 246,797 scans during the evaluation period. These scans originated from 837 addresses. Of those, 406 were internal addresses scanning external addresses. Conversely, there were 429 external scans against university addresses.

The 10 most scanned University addresses are listed in Table 6. As can be seen, Host MY.Net.1.3 was by far the most attacked system.

Top Ten University Addresses Scanned	
Scanned Address	**Count**
MY.NET.1.3	34906
MY.NET.1.4	24879
MY.NET.88.163	23837
MY.NET.6.45	22019
MY.NET.153.194	21476
MY.NET.60.43	19654
MY.NET.152.10	19229
MY.NET.153.210	18368
MY.NET.152.19	14438

Top Ten University Addresses Scanned	
Scanned Address	**Count**
MY.NET.153.173	14077

Table 6 – Top Ten external Scans

Table 7 shows the 10 most frequent scanners. The host MY.NET.60.43 shows the highest scanning activity. That count is very nearly equal to the sum of the other 9 scanner's activities.

Top ten internal talkers

Internal Scans Against External Addresses	
Source Address	**Count**
MY.NET.60.43	352739
MY.NET.6.49	79371
MY.NET.6.45	73707
MY.NET.6.48	45883
MY.NET.6.52	41698
MY.NET.6.50	34740
MY.NET.6.60	31839
MY.NET.153.171	28542
MY.NET.6.53	25589
MY.NET.151.17	10128

Table 7 – Top Ten Internal Scans

Table 8 shows the top 10 scanners against *all* University addresses.

Top ten external talkers

External Scans Against Internal Addresses	
Source Address	**Count**
66.38.185.141	23798
205.188.228.33	12173

External Scans Against Internal Addresses	
Source Address	**Count**
205.188.228.65	8326
205.188.228.17	7303
205.188.228.1	6617
216.106.172.148	5099
216.106.173.149	4918
216.106.172.149	4800
216.106.173.147	3612
216.106.173.148	3308

Table8 – Top Ten Internal Scans

The top two scanners originate from a net block registered in Ontario and a block registered to America On-line.

Address	Net Block Owner
66.38.185.141	GT Group Telecom Services Corp. (NETBLK-GROUPTELECOM-BLK-3) 20 BAY STREET SUITE 700 TORONTO, ON M5J 2N8 CA Netname: GROUPTELECOM-BLK-3 Netblock: 66.38.128.0 - 66.38.255.255 Maintainer: GTGR Coordinator: GT Group Telecom Services Corp. (ZG40-ARIN) hostmaster@gt.ca 416-848-2000 Domain System inverse mapping provided by: NS1.CLGRAB.GROUPTELECOM.NET 139.142.2.3 NS2.TOROON.GROUPTELECOM.NET 209.135.99.3 ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE Record last updated on 27-Jun-2001.

	Database last updated on 6-Apr-2002 19:57:34 EDT
205.188.228.33	America Online, Inc (NETBLK-AOL-DTC) 22080 Pacific Blvd Sterling, VA 20166 US Netname: AOL-DTC Netblock: 205.188.0.0 - 205.188.255.255 Coordinator: America Online, Inc. (AOL-NOC-ARIN) domains@AOL.NET 703-265-4670 Domain System inverse mapping provided by: DNS-01.NS.AOL.COM 152.163.159.232 DNS-02.NS.AOL.COM 205.188.157.232 Record last updated on 27-Apr-1998. Database last updated on 6-Apr-2002 19:57:34 EDT.

Hosts Possibly Compromised:

Based upon network activity the following hosts should be evaluated for NIMDA/CodeRed infections:
My.Net.152.14
My.Net.153.141
My.Net.153.114
My.Net.153.110
My.Net.153.151

These hosts may be having problems with KaZaA
24.180.218.241
24.234.240.101

This host should be evaluated for the Red Worm Trojan
MY.NET.150.198

Evaluation Methods

Given the size of the logs to be analyzed, the author chose to do the analyses with Access queries and reports. Snortsnarf literally ran for days and was used to confirm certain statistics.

To present data to Access in a comma-delimited format, a VB program was written. BASIC is still the quickest at simple I./O but that's another paper. Appendixes A and B show the data scrubbing routines for preparing the logs for import.

References – Assignment 3

Brenton, Chris. “Protection Against The Lion Worm”. 26 March 2001 URL: http://www.sans.org/y2k/lion_protection.htm

Counterpane Internet Security, Inc. “Multiple SNMP Vulnerabilities”. 12 February 2002. URL: http://www.counterpane.com/alert-snmp.html

Glaser, Thomas. “**TCP/IP Stack Fingerprinting Principles”. 25 Oct 2000. URL:**http://www.sans.org/newlook/resources/IDFAQ/TCP_fingerprinting.htm

MountAraratBlossom. “Firewall Penetration Testing”. 20 Nov 2000 URL: http://www.wittys.com/files/mab/fwpentesting.html
.
Stutzman, Jeff. “Handler’s Diary” 21 Feb 2000. URL: http://www.sans.org/y2k/022100.htm

Appendix A

Prepare Data for Import into a Database

```
' Read the scan file into a comma delimited file for the db of your choice
Dim Str, src, srcPort, dst, dstPort, Typ As String

Open "d:\scans" For Input As #1    ' Open file for input.
Open "d:\out.txt" For Output As #2   ' Open file for output.

Do While Not EOF(1)   ' Loop until end of file.
  Input #1, Str   ' Read a line

If Mid$(Str, 2, 2) <> "**" Then
 If Mid$(Str, 2, 2) <> "no" Then
   dt = Left$(Str, 15)
   src = Mid$(Str, 17, (InStr(22, Str, ":") - 17))
   srcPort = Mid$(Str, (InStr(22, Str, ":") + 1), (InStr(22, Str, " ") - InStr(22, Str, ":")
- 1))
   dst = Mid$(Str, (InStr(30, Str, ">") + 1), ((InStr(34, Str, ":") - 1) - InStr(30, Str,
">")))
   dstPort = Mid$(Str, (InStr(44, Str, ":") + 1), (InStr(44, Str, " ") - InStr(44, Str, ":")
- 1))
   Starttype = (InStr(44, Str, ":") + 1) + ((InStr(44, Str, " ") - InStr(44, Str, ":") - 1))
   Endtype = Len(Str)

   Typ = Right$(Str, Endtype - Starttype)

   Print #2, dt;        ' Print Date to file.
   Print #2, ",";
   Print #2, src;       ' Print Source Address to file.
   Print #2, ",";
   Print #2, srcPort;   ' Print Source Port to file.
   Print #2, ",";
   Print #2, dst;       ' Print Dest address to file.
   Print #2, ",";
   Print #2, dstPort;   ' Print Dest Port to file.
   Print #2, ",";
   Print #2, Typ;       ' Print Typw to file.
   Print #2, vbCrLf;    ' Print CR and LF to file.
    Kount = Kount + 1

   txtNumber.Text = Kount
 End If
End If

Loop
```

```
Close #1   ' Close file.
Close #2   ' Close file.
```

Appendix B

```
' Read the alert file into a comma delimited file for the db of your choice

Dim Str, dt, Alert, Desc As String
Dim StartDesc, EndDesc As Integer

Open "d:\al.txt" For Input As #1    ' Open file for input.
Open "d:\alertout.txt" For Output As #2   ' Open file for output.

Do While Not EOF(1)   ' Loop until end of file.

Line Input #1, Str   ' Read a line


If Mid$(Str, 2, 2) <> "**" Then
 If Left$(Str, 1) <> Chr$(9) Then


   dt = Left$(Str, 21)
   On Error Resume Next
   Alert = Mid$(Str, (InStr(22, Str, "]") + 2), (InStr(32, Str, "[") - InStr(22, Str, "]") -
3))
   StartDesc = (InStr(44, Str, "]") + 1)
   EndDesc = (Len(Str))
   Desc = Right$(Str, EndDesc - StartDesc)


   Print #2, dt;        ' Print Date to file.
   Print #2, ",";
   Print #2, Alert;       ' Print Source Address to file.
   Print #2, ",";
   Print #2, Desc;  ' Print Source Port to file.
   Print #2, vbCrLf;    ' Print CR and LF to file.
   Kount = Kount + 1

   txtNumber.Text = Kount

 End If
End If

Loop

Close #1   ' Close file.
Close #2   ' Close file.
```