Becky Bogle
GIAC Certification Practical
SANS Security 2001, New Orleans

Table of Contents:

Detects Analyzed

Detect #1

```
Server used for this query: [ whois.ripe.net ]
        inetnum:      62.153.97.0 - 62.153.97.127
        netname:      BIGBROTHER-GERMANY-1
        descr:        Endemol Entertainment GmbH
        descr:        Cologne
        descr:        temporary until 20000630
        country:      DE

Feb  5 02:11:14 hostm snort[10550]: IDS10 - RPC - portmap-request-
rstatd: 62.153.97.75:874 -> z.y.w.98:111
Feb  5 02:11:14 hostm snort[10550]: IDS362 - MISC - Shellcode X86 NOPS-
UDP: 62.153.97.75:875 -> z.y.w.98:32772
```

Source of Trace: http://www.sans.org/y2k/020601-1000.htm

Detect was generated by: Snort Intrusion Detection System was used with the –S option to send alerts to the Syslog. The fields are:

Date Time Host Process[ProcessID]: Alert: SourceIP:SourcePort -> DestIP:DestPort

Probability the Source Address was Spoofed:

The source address was probably not spoofed. Although the attacker does not need a response from the buffer overflow attempt in the second alert, a response is needed from the pre-attack probe in the first alert.

Description of Attack:

These alerts suggest that a buffer overflow exploit was attempted on the rpc.statd service. The CVE entry is *CVE-1999-0018: Buffer Overflow in statd allows root privileges.*

Attack Mechanism:

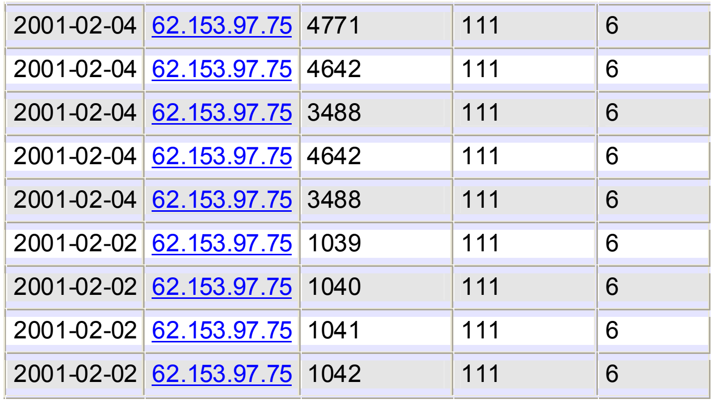
First, a query is sent to port 111, the expected location of portmap, which keeps tract of the port locations of various RPC services. The attacker is looking for port information for the rstatd service. This service can provide detailed information about the host, and older versions of statd are vulnerable to buffer overflow attacks. Next, a buffer overflow exploit is attempted on port 32772, the expected location of the rstatd service. The alarm for this exploit triggers when a string of the character 0x90 is detected. This string can indicate a buffer overflow as many of these exploits use a series of 0x90 to pad their chances of a successful exploit.

Correlations:

```
From: http://www.sans.org/y2k/020901-0930.htm
Jan 27 21:18:03 myhost tcplogd: "Syn probe"
62.153.97.75[62.153.97.75]:[1779]->myhost[192.168.30.1]:ftp
```

```
From: http://www.dshield.org (Dshield provides a searchable database of
logs that have been submitted by firewall users)
```

Date	**Source**	**Source Port**	**Target Port**	**Protocol**
2001-02-14	62.153.97.75	0	111	6
2001-02-11	62.153.97.75	49003	46995	6
2001-02-10	62.153.97.75	2879	111	6
2001-02-10	62.153.97.75	2880	111	6
2001-02-10	62.153.97.75	2881	111	6
2001-02-08	62.153.97.75	1972	111	6
2001-02-06	62.153.97.75	3230	111	6
2001-02-05	62.153.97.75	0	111	0
2001-02-04	62.153.97.75	3860	111	6
2001-02-04	62.153.97.75	3861	111	6
2001-02-04	62.153.97.75	3862	111	6
2001-02-04	62.153.97.75	3863	111	6
2001-02-04	62.153.97.75	4771	111	6



2001-02-04	62.153.97.75	4771	111	6
2001-02-04	62.153.97.75	4642	111	6
2001-02-04	62.153.97.75	3488	111	6
2001-02-04	62.153.97.75	4642	111	6
2001-02-04	62.153.97.75	3488	111	6
2001-02-02	62.153.97.75	1039	111	6
2001-02-02	62.153.97.75	1040	111	6
2001-02-02	62.153.97.75	1041	111	6
2001-02-02	62.153.97.75	1042	111	6

Evidence of Active Targeting:

There is some evidence of active targeting here, as the attacker is not just scanning, but is attempting to exploit a vulnerability in a specific service (rstatd), presumably to gain root access on this machine (z.y.w.98). There are no other machines targeted in this particular trace, although our correlations show that this attacker has been busy looking for vulnerabilities on other machines as well.

Severity:

Criticality: 3

Not sure what kind of machine this is

Lethality: 5

If successful, this attack can allow an attacker to gain root privileges.

System Countermeasures: 3

Not sure if this machine is running statd

Network Countermeasures: 2

This attack was detected, but it does not appear to be intercepted. We are only given Snort logs.

Severity = (Criticality+Lethality) – (System Countermeasures + Network Countermeasures)
= (5+3)-(3+2) = 3

Defensive Recommendation:

If NFS is not being used in your environment, then there is no need for the statd program to be running and it should be disabled. If the statd program is necessary on a system that is vulnerable, upgrade the operating system or apply the latest vendor patches. Also, block unneeded ports at your firewall.

Multiple Choice Test Question:

What Snort option is used in the following trace:

```
Feb  5 02:11:14 hostm snort[10550]: IDS10 - RPC - portmap-request-rstatd:
62.153.97.75:874 -> z.y.w.98:111
```

a) -S to send alerts to the syslog
b) -A for fast alerts
c) -O to change the order of rules precedence
d) -M to enable logging to a Windows host

The answer is *a) -S to send alerts to the syslog*.

Detect #2

```
Server used for this query: [ whois.ripe.net ]
        inetnum:     195.77.136.0 - 195.77.136.255
        netname:     ABCTELEMATIC
        descr:       ABC telematica
        country:     ES

Jan  4 22:14:49 hostmf /kernel: Connection attempt to TCP a.b.f.167:21
from
  195.77.136.108:21
Jan  4 22:17:26 hosth inetd[50480]: refused connection from
195.77.136.108,
  service ftpd (tcp)
Jan  4 22:18:33 hostda in.ftpd[1079]: refused connect from
195.77.136.108
Jan  4 22:18:34 hostda in.ftpd[1080]: refused connect from
195.77.136.108
Jan  4 22:18:34 hostda in.ftpd[1081]: refused connect from
195.77.136.108
Jan  4 22:18:34 hostdo in.ftpd[1032]: refused connect from
195.77.136.108
Jan 04 22:18:34 hostl proftpd[6140] hostl
(195.77.136.108[195.77.136.108]):
  connected - local  : a.b.c.63:21
Jan 04 22:18:34 hostl proftpd[6140] hostl
(195.77.136.108[195.77.136.108]):
  connected - remote : 195.77.136.108:4611
Jan 04 22:18:34 hostl proftpd[6140] hostl
(195.77.136.108[195.77.136.108]):
  FTP session closed.
Jan 04 22:18:36 hostl proftpd[6141] hostl
(195.77.136.108[195.77.136.108]):
  connected - local  : a.b.c.159:21
Jan 04 22:18:36 hostl proftpd[6141] hostl
(195.77.136.108[195.77.136.108]):
  connected - remote : 195.77.136.108:4618
Jan 04 22:18:36 hostl proftpd[6141] hostl
(195.77.136.108[195.77.136.108]):
  FTP session closed.
Jan  4 22:18:36 hostci in.ftpd[28250]: refused connect from
195.77.136.108
Jan  4 22:18:37 hostki in.ftpd[23977]: refused connect from
195.77.136.108
```

```
Jan  4 22:18:38 hostka in.ftpd[586]: refused connect from
195.77.136.108

Jan  4 22:17:25 195.77.136.108:21 -> a.b.c.32:21 SYNFIN ******SF
Jan  4 22:17:25 195.77.136.108:21 -> a.b.c.33:21 SYNFIN ******SF
Jan  4 22:17:26 195.77.136.108:21 -> a.b.c.62:21 SYNFIN ******SF
Jan  4 22:17:26 195.77.136.108:4610 -> a.b.c.62:21 SYN ******S*
Jan  4 22:17:26 195.77.136.108:21 -> a.b.c.67:21 SYNFIN ******SF
Jan  4 22:17:26 195.77.136.108:21 -> a.b.c.71:21 SYNFIN ******SF
Jan  4 22:17:26 195.77.136.108:21 -> a.b.c.80:21 SYNFIN ******SF
Jan  4 22:17:27 195.77.136.108:21 -> a.b.c.101:21 SYNFIN ******SF
Jan  4 22:17:27 195.77.136.108:21 -> a.b.c.114:21 SYNFIN ******SF
Jan  4 22:17:27 195.77.136.108:21 -> a.b.c.121:21 SYNFIN ******SF
Jan  4 22:17:29 195.77.136.108:21 -> a.b.c.207:21 SYNFIN ******SF
Jan  4 22:17:29 195.77.136.108:21 -> a.b.c.211:21 SYNFIN ******SF
```

Source of Trace:

http://www.sans.org/y2k/011101.htm

Detect Was Generated By:

Snort Portscan Logs (using the Portscan preprocessor):

Date Time SourceIP:SourcePort -> DestIP:DestPort ScanType Flags

Syslog (recording FTP session activity):

Date Time Host Process[ProcessID]: Message

Probability Source Address Was Spoofed:

It is very unlikely that this source address was spoofed. The attacker needs to get information back from the destinations in order for this attack to be successful.

Description of Attack:

This is a reconnaissance scan looking for ftp servers. The ftp banner is 'grabbed' from responsive hosts in order to fingerprint the machine. The date of this trace and the pattern that is demonstrated here are hints that this could be the beginning portion of the Ramen Worm.

CERT Incident Note 2001-01: Widespread Compromises via "ramen" Toolkit
http://www.cert.org/incident_notes/IN-2001-01.html

Attack Mechanism:

The attacking IP is randomly scanning hosts on the network using reflexive source and destination ports – both are port 21 (FTP). The syn and fin flags are set during this scan.

When a responsive system is found, the attacker opens an FTP connection with the system and then immediately closes it, presumably to grab the ftp banner in order to fingerprint the machine. This pattern matches the beginning segment of the Ramen Worm (see Max Vision's trace at http://www.whitehats.com/print/library/worms/ramen/ramenattack.txt).
The Ramen Worm is known to begin with a random generation of target hosts, then a Syn/Fin scan with source and destination port equal to 21. FTP banners are grabbed from those hosts that are responsive in order to determine if the machine is a Redhat 6.2 or 7.0 server. This fingerprinting keys off of the datestamp. An attack is then launched against those machines with publicly available exploits of three known vulnerabilities. On Red Hat 6.2 systems, the worm exploits vulnerabilities in wu-ftpd and rpc.statd. On version 7.0, it attacks LPRng. In this particular trace the attacker was able to connect with 2 ftp servers. We are not given any indication that there was further activity from this source ip. This may be an indication that the machines were not Redhat 6.2 or 7.0 boxes, so the worm does not pursue them any further.

Correlations:

The attacking IP is present in the dshield.org database that contains firewall logs of subscribers. The event date in the database entries is January 1, 3 days before the date of the above trace, and the same source and destination ports are targeted. Since the Ramen Worm infects targets and then begins to scan other machines from this target, it's possible that 195.77.136.108 was infected with the worm and proceeded to scan other targets in order to infect them as well.

From http://www.dshield.org `(Dshield provides a searchable database of logs that have been submitted by firewall users)`

Date	Source	Source Port	Target Port	Protocol
2001-01-01	195.77.136.108	21	21	6
2001-01-01	195.77.136.108	21	21	6

Evidence of Active Targeting:
There is little evidence of active targeting in this trace. The attacker is scanning random hosts on the network for FTP, and appears to be only collecting reconnaissance information at this point. We don't know if this is part of a wider probe or not, although our correlation information suggests that this network was not the only one scanned. This will become active targeting if the attacker returns and attempts to compromise certain hosts based on the reconnaissance information that was collected.

Severity:

Criticality	3	Only FTP servers are targeted at this point.
Lethality	2	At this point the attacker is collecting reconnaissance.
System Countermeasures	3	We don't know what version of FTP is running on the hosts.

Network Countermeasures	2	Snort is able to detect these connection attempts. The attacker is given FTP access on some hosts.

Severity = (3+2)-(3+2) = 0

Defensive Recommendation:
Disable anonymous ftp access if possible. Verify appropriate access rights for all users. Ensure that you are not running a vulnerable version of FTP such as the wu-ftp daemon that is shipped with many distributions of Linux and other UNIX operating systems. Ensure that ftp banners are not giving away sensitive information. Ensure that effective traffic boundaries exist between machines that provide external services such as FTP, and machines that provide internal services.

Multiple Choice Test Question:

“Banner Grabbing” is a method used to:

a) retrieve password files from machines
b) fingerprint a machine
c) get warez files from an ftp server
d) get RPC portmapper information

The answer is b) fingerprint a machine.

Detect #3

```
        Server used for this query: [ whois.arin.net ]
        Wam Net Enterprises Inc. (NETBLK-UU-208-237-112)
        123 NW 13th St Boca Raton, FL 33432 US
        Netname: UU-208-237-112
        Netblock: 208.237.112.0 - 208.237.127.255
        Maintainer: WAM
```

```
Dec  2 09:26:37 hosth snort[9931]: connect to 515 from outside:
  208.237.124.172:2158 -> a.b.c.32:515
Dec  2 09:26:38 hosth snort[9931]: connect to 515 from outside:
  208.237.124.172:2188 -> a.b.c.62:515
Dec  2 09:26:38 hosth snort[9931]: connect to 515 from outside:
  208.237.124.172:2197 -> a.b.c.71:515
Dec  2 09:26:38 hosth snort[9931]: connect to 515 from outside:
  208.237.124.172:2206 -> a.b.c.80:515
Dec  2 09:26:38 hosth snort[9931]: connect to 515 from outside:
  208.237.124.172:2338 -> a.b.c.212:515
Dec  2 09:26:42 hosth snort[9931]: connect to 515 from outside:
  208.237.124.172:2912 -> a.b.f.21:515
```

SOURCE OF TRACE:

http://www.sans.org/y2k/120600-1400.htm

DETECT WAS GENERATED BY:

Snort using the –S option to send alerts to the syslog

PROBABILITY THE SOURCE ADDRESS WAS SPOOFED:

The source address was probably not spoofed. The source IP needs to get information back from the destination in order for this attack to be successful.

DESCRIPTION OF ATTACK:

This trace demonstrates a pre-attack probe looking for systems that can be compromised using a vulnerability in the Unix LPR service, which runs on port 515 (printer spooler).

References for the Unix LPR Service Vulnerability:

Bugtraq ID 1711: Multiple Vendor lpr Format String Vulnerability

CERT Advisory CA-2000-22 Input Validation Problems in LPRng

CVE 2000-0917: Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands.

SANS Alert: Increased probes to TCP port 515, Posted: 14:00 November 20, 2000
http://www.sans.org/newlook/alerts/port515.htm

ATTACK MECHANISM:

This trace demonstrates a pre-attack probe looking for systems that can be compromised using a vulnerability in the Unix LPR service, which runs on port 515 (printer spooler). LPR is a utility that queues print jobs and sends them to a destination. LPR version 3.6.24 and prior contain a format string vulnerability that allows attackers to execute arbitrary commands. If the attacker has access to the printer port (port 515), format string parameters could be passed that overwrite arbitrary addresses in the printing service's address space. Such overwriting can cause segmentation violations leading to denial of printing services or to the execution of arbitrary code.

CORRELATIONS:

From http://www.dshield.org (Dshield provides a searchable database of logs that have been submitted by firewall users)

Date	Source	Source Port	Target Port	Protocol
2000-12-03	208.237.124.172	1057	27374	6
2000-12-03	208.237.124.172	1057	27374	6

EVIDENCE OF ACTIVE TARGETING:

A specific service, the printer service, is targeted in this trace. However, the attacker does not seem to be targeting a specific machine. From the information provided, it appears that the attacker is scanning random machines on the network for port 515. We cannot tell whether or not this is part of a wider reconnaissance scan, although this is possible.

SEVERITY:

Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
Lethality = 2 This is a reconnaissance scan
Critcality = 3 We do not know how important these machines are
System countermeasures = 3 We are not sure whether these machines are running vulnerable versions of LPRng
Network Countermeasures = 2 Snort is the only network countermeasure evident in this trace
(2+3)-(3+2)= 5

DEFENSIVE RECOMMENDATIONS:

Upgrade to a non-vulnerable version of LPRng. If possible, block access to port 515 using a firewall or packet-filtering device. Note that simply blocking port 515 at a network perimeter would allow a remote user inside the perimeter to exploit this vulnerability.

MULTIPLE CHOICE TEST QUESTION:

Which of the following would be a good reason to block access to port 515 at your firewall:

a) The SubSeven Trojan is known to run on this port
b) This port is a common RPC port with known vulnerabilities.
c) You are running a vulnerable version of LPRng
d) You are running a vulnerable version of BIND

The answer is *c) You are running a vulnerable version of LPRng.*

Detect #4

Server used for this query: [whois.ripe.net]
inetnum: 194.102.199.0 - 194.102.199.255
netname: RECEP-NET
descr: SC RECEP SRL
descr: b-dul. Decebal Bl.P Parter

descr: 2700 DEVA, Hunedoara
country: RO

```
Oct 27 19:25:35 hosth /kernel: Connection attempt to UDP
  a.b.c.62:53 from 194.102.199.38:2456
Oct 27 19:23:35 hosth snort[225]: IDS277 - NAMED Iquery Probe:
  194.102.199.38:2244 -> a.b.c.32:53
Oct 27 19:24:51 hosth snort[225]: IDS277 - NAMED Iquery Probe:
  194.102.199.38:2379 -> a.b.c.51:53
Oct 27 19:25:35 hosth snort[225]: IDS277 - NAMED Iquery Probe:
  194.102.199.38:2456 -> a.b.c.62:53
Oct 27 19:26:11 hosth snort[225]: IDS277 - NAMED Iquery Probe:
  194.102.199.38:2522 -> a.b.c.71:53
Oct 27 19:26:47 hosth snort[225]: IDS277 - NAMED Iquery Probe:
  194.102.199.38:2588 -> a.b.c.80:53
```

Source of Trace:
http://www.sans.org/y2k/110200-1230.htm

Detect was generated by:
Snort Intrusion Detection System was used with the –S option to send alerts to the Syslog. The fields are:
Date Time Host Process[ProcessID]: Alert: SourceIP:SourcePort -> DestIP:DestPort

```
Snort rule for this alert:
alert udp !$HOME_NET any -> $HOME_NET 53 (msg:"IDS277 - NAMED Iquery
Probe"; content: "|0980 0000 0001 0000 0000|"; offset: "2"; depth:
"16";)
```

Probability the source address was spoofed:

The source address in this trace was probably not spoofed as the attacker needs to get a response in order for this pre-attack probe to be successful.

Description of attack:

This is a pre-attack probe attempting to determine if the targets support inverse query requests.

CERT (CA-1998-05):
http://www.cert.org/advisories/CA-1998-05.html

Bugtraq (Bugtraq ID: 134)
http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fvid%3D134

Arachnids (IDSKey: IDS277)
http://www.whitehats.com/info/IDS277

CVE (CVE-1999-0009)

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0009

Attack mechanism:
Certain versions of Bind (BIND 4.9 releases prior to BIND 4.9.7 and BIND 8 releases prior to 8.1.2) are vulnerable to buffer overflows such that a maliciously formed inverse query can crash the server or allow an attacker to gain root privileges. This probe is attempting to determine if the target is a vulnerable nameserver, in which case a buffer overflow attack may soon follow.

Correlations:
Eric Hacker analyzed an Inverse Query probe in his GIAC practical:
http://www.sans.org/y2k/practical/Eric_Hacker.html

Traces of this probe have been seen several times on the GIAC postings. Examples can be found in the following logs:
http://www.sans.org/y2k/022201.htm
http://www.sans.org/y2k/030501.htm
http://www.sans.org/y2k/030101-1600.htm

Evidence of Active Targeting:
There does not seem to be active targeting at this point. This appears to be a reconnaissance scan of hosts on the network. It is not clear if reconnaissance information was previously collected to isolate these hosts, which might indicate active targeting. If the attacker determines that a machine is a nameserver running a vulnerable version of BIND, he could then proceed to actively target this machine by exploiting vulnerabilities associated with this version.

Severity:

Criticality	3	We do not know the criticality of these systems.
Lethality	2	This is reconnaissance, not an attack.
System Countermeasures	3	We do not know if the machines are running Bind, and if so, what version.
Network Countermeasures	2	Snort detects the packets, but does not block them.

Severity = (3+2)-(3+2) = 0

Defensive Recommendation
If BIND is running on the machine, disable inverse queries and upgrade to the latest version of BIND.

Multiple Choice Question

```
Feb 28 21:02:53 hosty snort[80143]: IDS277 - NAMED Iquery Probe:
  203.226.49.1:1339 -> z.y.w.34:53
```

The alert above is most probably probing for what kind of vulnerability?

a) Bind Version
b) Wu-FTP
c) LPRng
d) Rpc.statd

The answer is a) Bind Version

Detect #5

On 12th Mar 2001 at 12:34 (UTC) detected many attempts to abuse various cgi scripts on several web servers attached to our network. This scan appears to have originated from 212.64.47.189. snort logs, times UTC +1300.

```
Mar 13 01:34:48 takahe snort[31580]: IDS128 - CVE-1999-0067 - CGI phf
  attempt: 212.64.47.189:1919 -> 130.216.35.105:80
Mar 13 01:34:54 takahe snort[31580]: IDS218 - CVE-1999-0070 - TEST-CGI
  probe: 212.64.47.189:1921 -> 130.216.35.105:80
Mar 13 01:34:58 takahe snort[31580]: IDS235 - CVE-1999-0148 - CGI-
HANDLER
  probe!: 212.64.47.189:1923 -> 130.216.35.105:80
Mar 13 01:35:00 takahe snort[31580]: WEB-CGI-Webgais CGI access
attempt:
  212.64.47.189:1924 -> 130.216.35.105:80
Mar 13 01:35:02 takahe snort[31580]: CVE-1999-0196 - WEB-CGI-
Websendmail
  CGI access attempt: 212.64.47.189:1925 -> 130.216.35.105:80
Mar 13 01:35:04 takahe snort[31580]: CVE-1999-0039 - WEB-CGI-Webdist
CGI
  access attempt: 212.64.47.189:1926 -> 130.216.35.105:80
Mar 13 01:35:05 takahe snort[31580]: CVE-1999-0262 - WEB-CGI-Faxsurvey
probe:
  212.64.47.189:1927 -> 130.216.35.105:80
Mar 13 01:35:09 takahe snort[31580]: CVE-1999-0264 - WEB-CGI-Htmlscript
CGI
  access attempt: 212.64.47.189:1928 -> 130.216.35.105:80
Mar 13 01:35:11 takahe snort[31580]: WEB-CGI-Cgichk Pf display access
  attempt: 212.64.47.189:1929 -> 130.216.35.105:80
Mar 13 01:35:13 takahe snort[31580]: IDS219 - WEB-CGI-Perl access
attempt:
  212.64.47.189:1930 -> 130.216.35.105:80
Mar 13 01:35:14 takahe snort[31580]: CVE-1999-0953 - WEB-MISC -
wwwboard.pl
  attempt: 212.64.47.189:1931 -> 130.216.35.105:80
Mar 13 01:35:16 takahe snort[31580]: IDS224 - CVE-1999-0045 - NPH CGI
access
  attempt: 212.64.47.189:1932 -> 130.216.35.105:80
Mar 13 01:35:19 takahe snort[31580]: CVE-1999-0146 - WEB-CGI-Campas CGI
  access attempt: 212.64.47.189:1934 -> 130.216.35.105:80
Mar 13 01:35:21 takahe snort[31580]: CVE-1999-0147 - WEB-CGI-Aglimpse
CGI
  access attempt: 212.64.47.189:1935 -> 130.216.35.105:80
```

```
Source: 212.64.47.189
Incident type: cgi abuse
re-distribute: yes
timezone: GMT + 1300
reply: no
Date: DXD
```

Source of Trace
http://www.sans.org/y2k/031601-1030.htm

Detect was generated by:
Snort Intrusion Detection System was used with the –S option to send alerts to the Syslog. The fields are:
Date Time Host Process[ProcessID]: Alert: SourceIP:SourcePort -> DestIP:DestPort

Probability the Source Address was spoofed:
The source address was probably not spoofed. The attacker needs a response back in order for some of these compromises to be successful.

Description of Attack:
This is an attempt to scan and exploit vulnerable CGI applications on a web server. Many popular CGI programs have known vulnerabilities that an attacker can use to deface a web page or gain remote access to the web server.

Attack Mechanism:
The trace shows 14 different Snort alarms targeting the web server 130.216.35.105. A brief description of each follows:

IDS128 - CVE-1999-0067 - CGI phf attempt: The attacker may be attempting to exploit a vulnerable CGI script called 'phf'. The attacker may be able to run arbitrary commands on the webserver if this script is present.
IDS218 - CVE-1999-0070 - TEST-CGI probe: The attacker is probing for the program 'test-cgi'. This program has a bug that may allow the attacker to obtain a list of all files on the webserver.
IDS235 - CVE-1999-0148 - CGI-HANDLER: The attacker is attempting to exploit the cgi-bin program called 'handler'. This program has a vulnerability that can allow an attacker to execute arbitrary commands on the webserver.
WEB-CGI-Webgais CGI access attempt: The attacker is attempting to exploit the Webgais program. This program contains a vulnerabilities which may allow an attacker to execute arbitrary commands on the webserver. (CVE: CVE-1999-0176)
CVE-1999-0196 - WEB-CGI-Websendmail: The attacker may be attempting to exploit the 'websendmail' program in the Webgais package. This program contains a vulnerability that may allow a remote user to access arbitrary files on the webserver.
CVE-1999-0039 - WEB-CGI-Webdist CGI: The attacker may be attempting to exploit the cgi program called 'webdist'. This program contains a vulnerability that may allow an attacker to execute arbitrary commands on the webserver.

CVE-1999-0262 - WEB-CGI-Faxsurvey probe: The attacker is probing for the 'faxsurvey' cgi script on Linux. This program contains a vulnerability that would allow the attacker to excute arbitrary commands on the webserver.
CVE-1999-0264 - WEB-CGI-Htmlscript CGI access attempt: The attacker is attempting to exploit the CGI program 'htmlscript'. This program has a vulnerability that may allow the attacker to have remote read access to files on the webserver.
WEB-CGI-Cgichk Pf display access attempt: The attacker is attempting to exploit the cgi program called 'pfdisplay' for SGI's Performer API Search Tool. This program has a vulnerability that would allow the attacker to have read access to files on the webserver.
IDS219 - WEB-CGI-Perl access attempt: The attacker is attempting to execute 'perl.exe'. If the perl interpreter is available to web clients, it can be used to execute arbitrary commands on the web server.(CVE: CAN 1999-0509)
CVE-1999-0953 - WEB-MISC - wwwboard.pl attempt: The attacker may be attempting to exploit the 'WWWBoard' program. This program stores encrypted passwords in a password file called 'passwd.txt' that can be accessed by remote attackers.
IDS224 - CVE-1999-0045 - NPH CGI access attempt: The attacker is attempting to exploit the CGI program 'NPH-test-cgi'. This program has a vulnerability that can allow an attacker to obtain a list of all files on the web server.
CVE-1999-0146 - WEB-CGI-Campas CGI access attempt: The attacker is attempting to exploit the CGI program called 'campas' that is included with some NCSA webservers. This program contains a vulnerability that may allow attackers to read arbitrary files on the web server.
CVE-1999-0147 - WEB-CGI-Aglimpse CGI access attempt: The attacker is attempting to exploit the cgi program called 'aglimpse' that is part of the Glimpse package. This program may allow the attacker to remotely execute arbitrary commands on the webserver.

Correlations:
Todd Garrison analyzed a CGI Attack trace in his practical:
http://www.sans.org/y2k/practical/Todd_Garrison.html

Evidence of Active Targeting:
There is evidence of active targeting here. The attacker is targeting specific machines (webservers) on the network, and is attempting to compromise these machines using attacks that were intended to exploit webserver vulnerabilities.

Severity:

Criticality	4	This attack is targeting the webservers.
Lethality	5	This attack could allow root access to the webserver.
System Countermeasures	3	We don't know if these vulnerable programs exist and are accessible on the webserver.
Network Countermeasures	3	The only countermeasure evident here is Snort, which will detect this attack, but will not intercept it.

Severity = (4+5)-(3+3) = 3

Defensive Recommendations:
Remove unnecessary cgi programs or disable them by removing execute permissions from the program. If this is not possible, apply appropriate vendor patches or revise the program code itself to remove the vulnerability.

Multiple Choice Test Question:

```
Mar 13 01:34:48 takahe snort[31580]: IDS128 - CVE-1999-0067 - CGI phf
  attempt: 212.64.47.189:1919 -> 130.216.35.105:80
```

The trace above indicates that the attacker is attempting to:

a) Exploit a vulnerable nameserver
b) Exploit a vulnerable webserver
c) Probe for vulnerable RPC services
d) Probe for vulnerable FTP servers.

The answer is b) Exploit a vulnerable webserver.

Evaluate an Attack

Netcat 1.1, Version 2.08.98, for Win 95/98/NT/2000 was downloaded from http://www.atstake.com/research/tools/index.html. The original version was written by *hobbit* hobbit@avian.org. The tool used in this evaluation is a port of the original version to Windows written by Weld Pond weld@atstake.com.

The basic features of netcat as described in http://www.l0pht.com/~weld/netcat/readment.txt include

```
* Outbound or inbound connections, TCP or UDP, to or from any ports
* Full DNS forward/reverse checking, with appropriate warnings
* Ability to use any local source port
* Ability to use any locally-configured network source address
* Built-in port-scanning capabilities, with randomizer
* Can read command line arguments from standard input
* Slow-send mode, one line every N seconds
* Hex dump of transmitted and received data
* Ability to let another program service established
  connections
* Telnet-options responder
```

Potential uses of netcat (from http://www.sans.org/infosecFAQ/audit/netcat.htm):

- Script backends
- Scanning ports and inventorying services
- Backup handlers

- File transfers
- Server testing and simulation
- Firewall testing
- Proxy gatewaying
- Network performance testing
- Address spoofing tests
- Protecting X servers
- 1001 other uses you`ll likely come up with

Usage Information:

```
      connect to somewhere:   nc [-options] hostname port[s]
[ports] ..
      listen for inbound:     nc -l -p port [options] [hostname]
[port]
      options:
              -d              detach from console, stealth mode

              -e prog         inbound program to exec
      [dangerous!!]
              -g gateway      source-routing hop point[s], up to
      8
              -G num          source-routing pointer: 4, 8, 12,
      ...
              -h              this cruft
              -i secs         delay interval for lines sent,
      ports scanned
              -l              listen mode, for inbound connects
              -L              listen harder, re-listen on socket
      close
              -n              numeric-only IP addresses, no DNS
              -o file         hex dump of traffic
              -p port         local port number
              -r              randomize local and remote ports
              -s addr         local source address
              -t              answer TELNET negotiation
              -u              UDP mode
              -v              verbose [use twice to be more
      verbose]
              -w secs         timeout for connects and final net
      reads
              -z              zero-I/O mode [used for scanning]
      port numbers can be individual or ranges: m-n [inclusive]
```

Portscanning is a reconnaissance technique that netcat can be used for. The following trace demonstrates a scan of ports 20-30 on a target machine. When a listening port is encountered, the target machine responds with a syn-ack. At the end of the scan, netcat provides the information it collected about the services that are running on listening ports.

The tool was downloaded to a Windows 98 box, along with Windump (http://netgroup-serv.polito.it/windump) to capture packets. The target box was SunOS5.8, with Snoop used to capture packets. The Sun box was set up to capture packets using the command 'snoop –P –v –o packets *sourcehost'*. The Windows box was set up to capture packets using the command ‘windump –I 5 –w packets host *targethost*’. Finally netcat was initiated from the Windows box using the command ‘nc –r *targethost* 20-30’. This command tells netcat to scan the targethost for ports 20-30. The –r option randomizes the order that the ports are scanned. This is evident from the traces:

Windump Trace from source box:

```
22:30:07.105077 sourcehost.22090 > targethost.26: [|tcp] (DF)
22:30:10.289153 sourcehost.22090 > targethost.26: [|tcp] (DF)
22:30:16.543792 sourcehost.22090 > targethost.26: [|tcp] (DF)
22:30:29.043318 sourcehost.22090 > targethost.26: [|tcp] (DF)
22:30:54.389492 sourcehost.12846 > targethost.27: [|tcp] (DF)
22:30:57.472225 sourcehost.12846 > targethost.27: [|tcp] (DF)
22:31:03.722016 sourcehost.12846 > targethost.27: [|tcp] (DF)
22:31:16.221496 sourcehost.12846 > targethost.27: [|tcp] (DF)
22:31:41.555013 sourcehost.31996 > targethost.24: [|tcp] (DF)
22:31:44.665435 sourcehost.31996 > targethost.24: [|tcp] (DF)
22:31:50.915172 sourcehost.31996 > targethost.24: [|tcp] (DF)
22:32:03.419676 sourcehost.31996 > targethost.24: [|tcp] (DF)
22:32:28.732872 sourcehost.20858 > targethost.23: [|tcp] (DF)
22:32:31.843576 sourcehost.20858 > targethost.23: [|tcp] (DF)
22:32:38.093347 sourcehost.20858 > targethost.23: [|tcp] (DF)
22:32:50.712830 sourcehost.20858 > targethost.23: [|tcp] (DF)
22:33:15.922953 sourcehost.21522 > targethost.29: [|tcp] (DF)
22:33:19.041764 sourcehost.21522 > targethost.29: [|tcp] (DF)
22:33:25.291531 sourcehost.21522 > targethost.29: [|tcp] (DF)
22:33:37.794680 sourcehost.21522 > targethost.29: [|tcp] (DF)
22:34:03.048132 sourcehost.17352 > targethost.20: [|tcp] (DF)
22:34:06.230050 sourcehost.17352 > targethost.20: [|tcp] (DF)
22:34:12.484723 sourcehost.17352 > targethost.20: [|tcp] (DF)
22:34:24.984225 sourcehost.17352 > targethost.20: [|tcp] (DF)
22:34:50.198961 sourcehost.9254 > targethost.30: [|tcp] (DF)
22:34:53.313129 sourcehost.9254 > targethost.30: [|tcp] (DF)
22:34:59.562903 sourcehost.9254 > targethost.30: [|tcp] (DF)
22:35:12.062396 sourcehost.9254 > targethost.30: [|tcp] (DF)
22:35:37.391203 sourcehost.22345 > targethost.25: [|tcp] (DF)
22:35:40.506330 sourcehost.22345 > targethost.25: [|tcp] (DF)
22:35:46.756066 sourcehost.22345 > targethost.25: [|tcp] (DF)
22:35:59.260557 sourcehost.22345 > targethost.25: [|tcp] (DF)
22:36:24.499963 sourcehost.31152 > targethost.22: [|tcp] (DF)
**target host is listening on port 22 (SSH)**
22:36:24.653072 targethost.22 > sourcehost.31152: S
3520055984:3520055984(0) ack 1144622 win 25452 <nop,nop,sackOK,mss
1460> (DF)
22:36:24.653227 sourcehost.31152 > targethost.22: [|tcp] (DF)
22:36:24.813469 targethost.22 > sourcehost.31152: P 1:26(25) ack 1 win
25452 (DF)
22:36:24.959574 sourcehost.31152 > targethost.22: [|tcp] (DF)
22:46:24.821747 targethost.22 > sourcehost.31152: F 26:26(0) ack 1 win
25452 (DF)
22:46:24.821914 arp reply targethost (0:0:0:11:22:33) is-at
0:0:0:11:22:33
22:46:24.821954 sourcehost.31152 > targethost.22: [|tcp] (DF)
22:46:24.824663 sourcehost.31152 > targethost.22: [|tcp] (DF)
```

```
22:46:24.974866 targethost.22 > sourcehost.31152: . ack 2 win 25452
(DF)
**SSH connection on target host has timed out**
22:46:25.401261 sourcehost.10766 > targethost.21: [|tcp] (DF)
22:46:28.491288 sourcehost.10766 > targethost.21: [|tcp] (DF)
22:46:34.741076 sourcehost.10766 > targethost.21: [|tcp] (DF)
22:46:47.230585 sourcehost.10766 > targethost.21: [|tcp] (DF)
22:47:12.453212 sourcehost.12068 > targethost.28: [|tcp] (DF)
22:47:15.564498 sourcehost.12068 > targethost.28: [|tcp] (DF)
22:47:21.934241 sourcehost.12068 > targethost.28: [|tcp] (DF)
22:47:34.433785 sourcehost.12068 > targethost.28: [|tcp] (DF)
```

Snoop Trace from target box:

```
1     0.00000 sourcehost -> targethost TCP D=22 S=31152 Syn Seq=1144621
Len=0 Win=5840 Options=,mss 1414,nop,nop,sackOK>
2     0.15437 sourcehost -> targethost TCP D=22 S=31152
Ack=3520055985 Seq=1144622 Len=0 Win=5840
3     0.30507 sourcehost -> targethost TCP D=22 S=31152
Ack=3520056010 Seq=1144622 Len=0 Win=5815
4   599.84214 sourcehost -> targethost TCP D=22 S=31152
Ack=3520056011 Seq=1144622 Len=0 Win=5815
5     0.00161 sourcehost -> targethost TCP D=22 S=31152 Fin
Ack=3520056011 Seq=1144622 Len=0 Win=5815
```

The only port that was accessible on the Sun box was port 22 (SSH). When the scan finished, netcat provided information about the service it found on port 22:

```
C:\netcat>nc -r targethost 20-30
SSH-1.99-OpenSSH_2.5.2p2
```

Besides portscanning, there are many other things that netcat can be used for. One was described in an article on the various uses of netcat: http://www.happyhacker.org/hhlist/windigest2_14.shtml. By using a command such as the following: 'nc -l -p1234 -d -e cmd.exe –L', you can use netcat to listen on a specific port and execute a file when a connection is made to the port. This particular command tells netcat to listen on port 1234 and execute cmd.exe upon connect, essentially providing a remote command prompt. Netcat can also be used to find vulnerable cgi scripts on a web server. An example of this is described on http://www.insecure.org/sploits/test-cgi.html. By issuing the following command, 'echo "GET /cgi-bin/test-cgi?*" | nc www.website.com 80', netcat will connect to port 80 on website.com and get information about the script test-cgi.

After using this tool, I believe that while netcat is a sufficient tool for portscanning, there are probably other tools such as nmap or Portscan by 7th Sphere that are better suited for this purpose. The strength of the netcat tool is in its flexibility and the variety of things for which it can be used.

Correlations:

A paper by Tom Armstrong on the various uses of netcat is available from the SANS website at:
http://www.sans.org/infosecFAQ/audit/netcat.htm

ANALYZE THIS!

Introduction

This analysis will examine GIAC Enterprise's Snort intrusion detection system data covering 11/24/2000 to 1/18/2001. The data is divided into three types of files: Scan Files, Alert Files, and OOS Files. A chart that maps each filename to the date it covers is provided below. From the chart it is evident that there are gaps in the data. Also, duplicate scan files were provided for 12/21/00 and 1/1/01. To ensure accuracy, these files (SnortS13.txt, SnortS14.txt, and SnortS39.txt) will not be included in the analysis.

	SCAN FILES		ALERT FILES		OOS FILES	
DATE	FILE	SIZE	FILE	SIZE	FILE	SIZE
11/24/2000			SnortA6	58042		
11/25/2000						
11/26/2000			SnortA9	217176		
11/27/2000						
11/28/2000			SnortA3	430830	ooscheck	730424
11/29/2000			SnortA2	101121		
11/30/2000						
12/1/2000			SnortA4	1096443		
12/2/2000			SnortA7	173816		
12/3/2000			SnortA8	149263		
12/4/2000			SnortA10	369874		
12/5/2000	SnortS2	3485763	SnortA5	317904		
12/6/2000	SnortS30	2576924	SnortA31	409671		
12/7/2000	SnortS28	337950	SnortA29	190374		
12/8/2000	SnortS23	1176670	SnortA26	389010		
12/9/2000	SnortS21	660923	SnortA27	532562	oosche22	1006974
12/10/2000	SnortS25	321151	SnortA24	441627	oosche20	1662280
12/11/2000						
12/12/2000	SnortS19	68602	SnortA17	1142324	oosche18	288590
12/13/2000	SnortS16	140748	SnortA14	214695	oosche15	606393
12/14/2000						
12/15/2000			SnortA13	1155162	oosche12	610378
12/16/2000	SnortS3	3924655	SnortA11	803165		

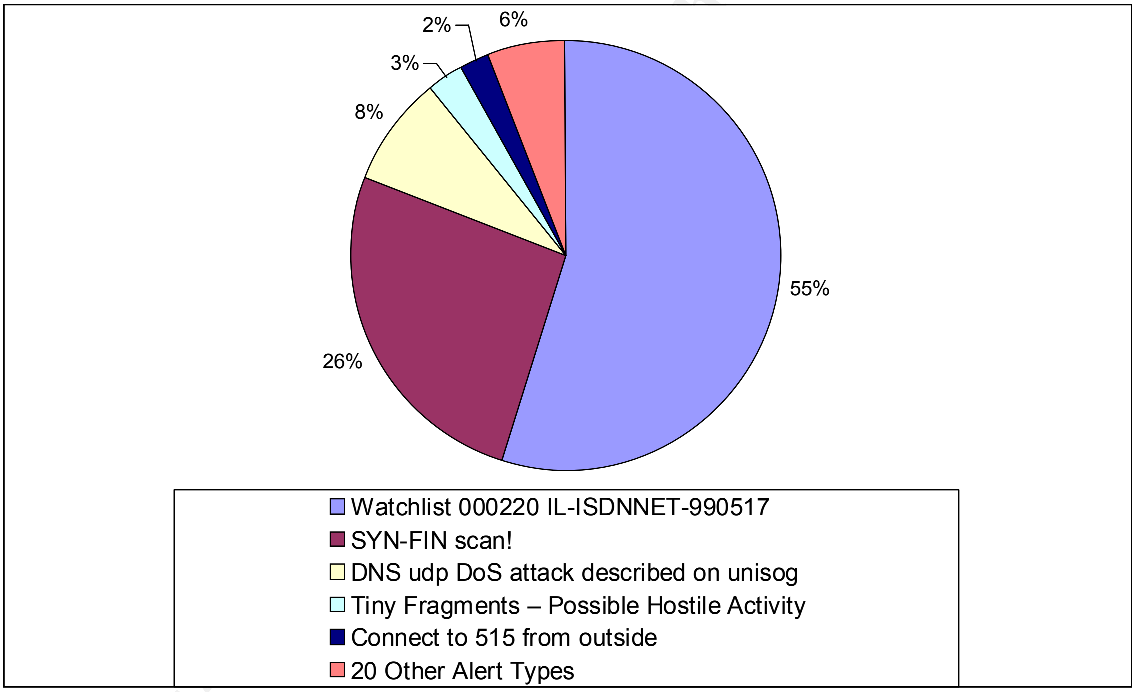
12/17/2000	SnortS10	3258756	SnortA12	623069		
12/18/2000						
12/19/2000					oosche2	69162
12/20/2000	SnortS17	5100583	SnortA20	1958861	oosche19	150276
12/21/2000	SnortS11,S13,S14	2942143	SnortA15	994509		
12/22/2000			SnortA46	2966484		
12/23/2000			SnortA41	2930203		
12/24/2000	SnortS43	1568309	SnortA44	725597		
12/25/2000	SnortS34	4929593				
12/26/2000			SnortA36	1907973		
12/27/2000	SnortS26	4551336				
12/28/2000	SnortS33	5057727	SnortA37	1215720	oosche27	593955
12/29/2000	SnortS24	1426217	SnortA25	715479		
12/30/2000	SnortS20	2851078	SnortA21	972125		
12/31/2000	SnortS22	2670948	SnortA23	1151488		
1/1/2001	SnortS29,S32	3830339	SnortA35	2018592		
1/2/2001	SnortS12	2959427	SnortA16	890655		
1/3/2001	SnortS18	5214757	SnortA19	860110		
1/4/2001			SnortA51	5060003	oosche39	63448
1/5/2001			SnortA50	1582404	oosche49	3451091
1/6/2001			SnortA47	2506111		
1/7/2001			SnortA45	3004708		
1/8/2001	SnortS42	3442612	SnortA43	1051262	oosche44	190881
1/9/2001	SnortS39	3252304	SnortA40	1029962	oosche41	172099
1/10/2001			SnortA38	2998689	oosche37	4674454
1/11/2001	SnortS35	2953590	SnortA34	4242168	oosche36	431405
1/12/2001	SnortS27	3083949	SnortA30	1681810	oosche28	513851
1/13/2001	SnortS31	1897901	SnortA32	1426926	oosche33	408229
1/14/2001					oosche38	411187
1/15/2001	SnortS15	4411948	SnortA18	1763836	oosche3	271127
1/16/2001			SnortA52	1529890	oosche51	458724
1/17/2001					oosche40	161074
1/18/2001			SnortA48	1501107	oosche50	494250

Alert Files

The following charts provide a summary of the 194,039 alerts found in the Alert Files:

Signature	# Alerts	# Sources	# Destinations
Watchlist 000220 IL-ISDNNET-990517	105918	46	100
SYN-FIN scan!	51192	37	27067
DNS udp DoS attack described on unisog	16146	8	6
Tiny Fragments – Possible Hostile Activity	5340	27	13
Connect to 515 from outside	4238	10	2877
Watchlist 000222 NET-NCFC	2401	31	19
WinGate 1080 Attempt	2239	474	572

Attempted Sun RPC high port access	2053	16	23
Null scan!	826	527	173
Queso fingerprint	710	52	72
SNMP public access	591	3	5
NMAP TCP Ping!	558	47	27
Russia Dynamo – SANS Flash 28-jul-00	546	2	2
SMB Name Wildcard	515	91	168
SUNRPC highport access!	204	25	19
Connect to 515 from inside	159	11	11
Broadcast Ping to subnet 70	154	24	1
TCP SMTP Source Port Traffic	100	5	88
Back Orifice	77	10	71
External RPC Call	59	15	25
Probable NMAP fingerprint attempt	8	5	6
Site exec – Possible wu-ftpd exploit – GIAC000623	2	2	2
SITE EXEC – Possible wu-ftpd exploit –GIAC000623	1	1	1
Happy 99 Virus	1	1	1
STATDX UDP Attack	1	1	1



Analysis of Alerts

Watchlist 000220 IL-ISDNNET-990517

This alarm triggers on activity coming from an ISP in Israel called Bezeq International. The ip ranges assigned to Bezeq are 212.25.121.0-212.25.121.255 and 212.179.68.120-212.179.68.127. A fair amount of these alarms seem to be caused by Napster and Gnutella traffic, indicated by target ports 6346, 6688, and 6699. There were 46 sources and 100 destinations for this alert.

SYN-FIN scan!

This alarm triggers on packets being sent with only the syn and fin flags set. The majority of these scans targeted ports 53 (DNS), 21 (FTP), and 109 (POP2). There were 37 sources and 27067 destinations for this scan.

DNS udp DoS attack described on unisog

This alarm appears to be triggering on any traffic from 209.67.50.X. While there were a total of 6 different source addresses for this alarm, all but 14 of the 16146 alerts came from 209.67.50.203. This traffic is related to a DDOS against register.com in early January 2001. Many networks noticed a flood of DNS requests to their nameservers from a spoofed IP, and the responses were being sent to register.com. This incident is referenced on the SANS website at http://www.sans.org/y2k/010901-1300.htm. There were 8 sources and 6 destinations for this alert. The activity has since ceased on this network, as the alerts were only triggered on 01/06/01.

Tiny Fragments – Possible Hostile Activity

This alarm triggers on tiny fragments, which can indicate a firewall penetration technique or a DoS attack There were 27 sources and 13 destinations for this attack. One of the sources was an internal host, MY.NET.219.122. On 11/29 at 20:31, this host initiated a connection to port 515 (printer spooler) on 128.2.166.68 (registered to Carnegie Mellon University). About 3 hours later, the same host, MY.NET.219.122, was the source of 7 'Tiny Fragments' alarms targeting 208.162.62.208 (registered to Covington Electric/Alaweb). This activity seems suspicious, and the host should be analyzed further to ensure that it has not been compromised.

Connect to 515 from outside

This alarm indicates that an external host is attempting to access port 515, the printer spooler port. Increased probes to this port were described on the SANS website at http://www.sans.org/newlook/alerts/port515.htm. The Unix LPR service runs on port 515, and this service contains vulnerabilities that could lead to root compromise from both local and remote systems. It appears that the majority of these alerts were caused by reconnaissance scans looking for LPR vulnerabilities. In the cases where the alert was not caused by a scan, ensure that the source IP is authorized to connect with the print service. Ensure that you are not running a vulnerable version of LPR to prevent this kind of compromise. There were 10 sources and 2877 destinations for this alert.

Watchlist 000222 NET-NCFC

This alert is triggering on the ip range 159.226.X.X, belonging to The Computer Network Center Chinese Academy of Sciences. There were 31 sources and 19 destinations for this alert. It seems that a large amount of these alerts are targeted to port 25 (SMTP) on MY.NET. Other ports that were common in these alerts are 113 (IDENT) and 443 (SSL).

WinGate 1080 Attempt

This alarm triggers on an attempt to access a Wingate proxy server on port 1080. This proxy can be used in order to surf anonymously on the web. There were 474 sources and

572 destinations for this alert. Some of these destinations had over 30 different source ips attempting to access port 1080. It's possible that these destinations may have been published as a publicly available proxy server. Scan your network for unauthorized proxy servers to prevent this kind of activity. Crist Clark point out in his practical (http://www.sans.org/y2k/practical/Crist_Clark_GCIA.html) that IRC servers can also cause this alarm to trigger.

Attempted Sun RPC high port access

This alert triggers on activity to port 32771, sometimes used as an alternate port for portmapper. This port is often targeted by attacks exploiting SUN RPC vulnerabilities. There were 16 sources and 23 destinations for this alert. Of the 16 sources, 13 were from the ip range 205.188.153.97-111 which is registered to America Online. Almost all activity from this IP range uses the port pair 4000 -> 32771, suggesting ICQ traffic.

```
Possible Snort Rules:
alert tcp any any -> $HOME_NET 32771 (msg: "Attempted Sun RPC high port
access";)
alert udp any any -> $HOME_NET 32771 (msg: "Attempted Sun RPC high port
access";)
```

Null scan!

This alert indicates tcp packets with no flags set. This techinique is commonly used for reconnaissance, such as OS fingerprinting. There were 527 sources and 173 destinations for this alert.

```
Possible Snort Rule:
alert tcp any any -> $HOME_NET any (msg:"NULL Scan"; flags: 0;)
```

Queso fingerprint

Queso is an operating system detection tool that is commonly used for reconnaissance. This alarm detects TCP packets with the S12 flags set, which is an indication that the Queso tool may be in use. A fair amount of this traffic involves port 6346, which may indicate Gnutella traffic. There were 52 sources and 173 destinations for this alert.

```
Possible Snort Rule:
alert tcp any any -> $HOME_NET any (msg:"Possible Queso Fingerprint
attempt"; flags: S12;)
```

SNMP public access

This alert triggers when a source tries to make an SNMP request using the password public. There were 3 sources and 5 destinations for this alert. Ensure that the community string 'public' is changed to avoid unauthorized activity.

```
Possible Snort Rule:
alert udp any any -> $HOME_NET 161 (msg: "SNMP public access";
content:"public";)
```

NMAP TCP Ping!

This alert triggers when a TCP packet has the acknowledgement field set to zero and the ACK flag set, characteristic of an NMAP TCP Ping.
This type of activity, which uses the NMAP port scanning tool (http://www.insecure.org), is often used for reconnaissance to determine if a network host is active.
There were 47 sources and 27 destinations for this alert.

```
Possible Snort Rule:
alert tcp any any -> $HOME_NET any (flags: A; ack: 0; msg:"NMAP TCP
ping!";)
```

Russia Dynamo – SANS Flash 28-jul-00

SANS recommended that traffic to or from the Russian IP range 194.87.6.X be blocked in a report on July 28, 2000 (http://www.sans.org/y2k/072818.htm). This was due to unusual activity consisting of internet wide port scanning for proxy servers, with the information being sent back to a Russian IP address. The flash advisory is referenced here: http://archives.neohapsis.com/archives/sans/2000/0068.html . This alert had 2 sources and 2 destinations, consisting of an internal host (123.456.205.138) talking with an external host (194.87.6.38). The port pair for this consisted of 2478 on the external host and 6699 on the internal host, suggesting napster traffic, as port 6699 is commonly used for Napster. All of this activity took place on 12/8.

SMB Name Wildcard

This alert is often caused by benign activity such as Windows systems trying to obtain the netbios name of other boxes it communicates with. A deliberate scan for port 137 might indicate someone trying to get reconnaissance information from the target hosts such as any netbios names known to the host. Scans for port 137 are analyzed on the SANS website at http://www.sans.org/newlook/resources/IDFAQ/port_137.htm. A probe of port 137 by itself is not evidence of an attack, but a simultaneous connection to port 139 could indicate that someone is trying to connect to your pc and access shared resources (http://www.dshield.org/ports/port137.html) This alert has 91 sources and 168 destinations. Pretty much all of the traffic is coming from external hosts using source and destination port 137 and appears to be harmless.

```
Possible Snort Rule:
alert udp any any -> $HOME_NET 137 (msg:"SMB Name Wildcard";
content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";)
```

SUNRPC highport access!

This alert appears to trigger on access to port 32771, sometimes used as an alternate port for portmapper (port 111). This port can provide information about the port locations of the various RPC services. If an RPC service is found to be listening at a particular port, it may be exploited using known vulnerabilities. There were 25 sources and 19 destinations for this alert. It is recommended that you monitor activity on this port closely as RPC services are subject to known vulnerabilities. Also, disable any unnecessary RPC services, and update services with applicable patches and version upgrades.

Connect to 515 from inside

This alarm indicates that an internal host is attempting to connect to port 515, the printer spooler port. Increased probes to this port were described on the SANS website at http://www.sans.org/newlook/alerts/port515.htm. The Unix LPR service runs on port 515, and this service contains vulnerabilities that could lead to root compromise from both local and remote systems. Examining the destinations for this alert, at least two need to be examined more closely. Destination 212.187.65.135 is registered to Nijmegen Cablemodems in the Netherlands. MY.NET.70.38 connected with port 515 on this host 3 times on 01/04. It seems suspicious that a box on my.net would need to connect to a print service in the Netherlands. Secondly, destination 148.243.214.7 is registered to Coordinacion Nacional de Progresa in Mexico. Host MY.NET.163.17 connected with port 515 on this external host on 12/20 at 21:58. Further analysis shows that on 12/15 (five days earlier) host 141.211.176.99 (registered to University of Michigan) scanned over 2200 boxes for port 515 on MY.NET including MY.NET.163.17. The fact that 141.211.176.99 is a university IP address suggests that it may be a compromised box. There is a possibility that MY.NET.163.17 has also been compromised, and this host needs to be examined more closely.

Broadcast Ping to subnet 70

This alarm triggers on a ping sent to the broadcast address MY.NET.70.255. This alarm has 24 sources and 1 destination. Some of these sources may be spoofed and using this subnet as a broadcast amplification site to initiate a DDOS attack. For example, 213.154.131.131 (registered to PCNET - ATM-ADSL Network in Bucharest) targeted MY.NET.123.70.255 with 52 broadcast pings on 12/01 from 19:11 to 20:39. To prevent this site from being used in this manner, ensure that the router is configured to prevent packets from being forwarded to broadcast addresses.

TCP SMTP Source Port Traffic

This alert triggers on traffic where the source port is 25 (SMTP – Simple Mail Transport Protocol). There were 5 sources and 88 destinations for this alert.

Back Orifice

This alert is triggered on traffic targeting port 31337, the default port for the Back Orifice Trojan. This activity appears to have been caused by potential attackers scanning for boxes that have been compromised with Back Orifice. There does not seem to be any evidence of an actual compromise at this time. This alert has 10 sources and 71 destinations.

```
Possible Snort Rule:
alert udp $EXTERNAL_NET any -> $HOME_NET 31337 (msg:"BACKDOOR
BackOrifice access"; content: "|ce63 d1d2 16e7 13cf 39a5 a586|";
reference:arachnids,399;)
```

External RPC Call

These alerts triggered on activity from external hosts targeting the portmapper servive on port 111, which is the contact point to determine what ports RPC services are running on. There are a number of vulnerabilities associated with RPC services. MY.NET.6.15 was the top destination for this alert. This host was probed for port 111 by 206.210.80.6 on 1/6 at 5:04. An hour and a half after the initial probe, the source IP launched a STATDX UDP attack against MY.NET.6.15. This host should be examined more closely to determine if it was compromised. Ensure that this host is not running vulnerable versions of rpc.statd. This alert had 15 sources and 25 destinations.

Probable NMAP fingerprint attempt

NMAP is a portscanning tool that can also be used for remote OS identification by TCP/IP fingerprinting (http://www.insecure.org). This alert triggers on tcp packets with the illegal flag setting SFPU, which may indicate that the NMAP tool is being used. This alert is an indication that someone is probably performing reconnaissance on your network and may follow up with an attack if vulnerabilities are found. This alert had 5 sources and 6 destinations.

```
Possible Snort Rule:
alert tcp any any -> $HOME_NET any (msg:"Possible NMAP Fingerprint
attempt"; flags: SFPU;)
```

Site exec – Possible wu-ftpd exploit – GIAC000623

Wu-ftpd is an ftp server that is vulnerable to a remote attack in the site exec implementation. This alert probably triggers when a long site exec command is issued, and it should be monitored closely because of the possibility of root compromise on the system. There were 2 sources and 2 destinations for this alert. Each of these sources was only responsible for one alert total, so the wu-ftp alert was not followed by any other transactions that might suggest a compromise. Ensure that these two hosts are not running vulnerable wu-ftpd services.

SITE EXEC – Possible wu-ftpd exploit –GIAC000623

It is unclear what the difference is between this alert and the one above. This alert had 1 source and 1 destination, with only one alert total between them. There were no transactions following the alert that might suggest a compromise.

Happy 99 Virus

The Happy 99 virus was sent to the host MY.NET.6.47 from 63.216.198.158 on 12/22. If the internal host opened the .exe attachment that was in the e-mail, this host is now infected. Ensure that the host has ant-virus software installed and running.

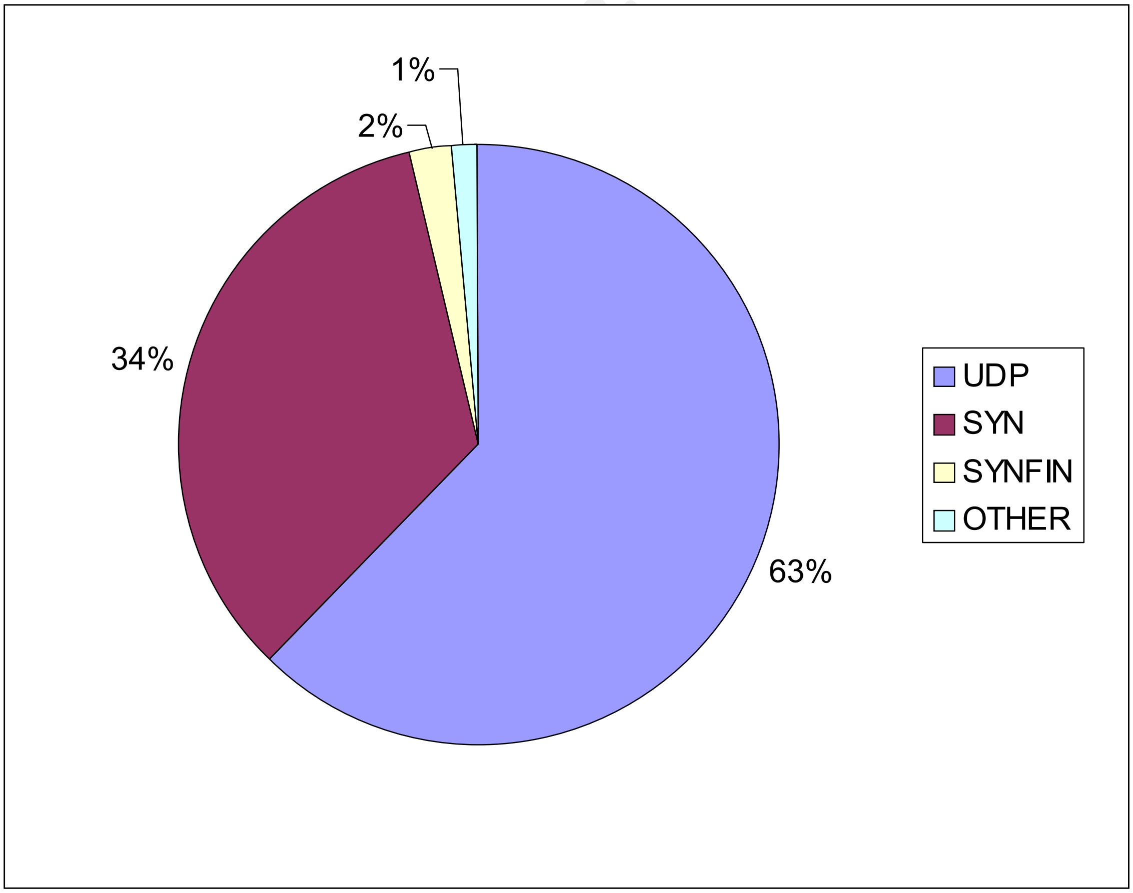
STATDX UDP Attack

This alert triggers on an attack using the stadx exploit which targets vulnerable statd services. There was one instance of this alert targeting MY.NET.6.15 on 01/06. This host was probed for port 111 by 206.210.80.6 on 1/6 at 5:04. An hour and a half after the initial probe, the source IP launched a STATDX UDP attack against MY.NET.6.15. This host should be examined more closely to determine if it was compromised. Ensure that this host is not running vulnerable versions of rpc.statd.

Scan Files

The following charts provide a summary of the 1,166,399 alerts found in the Scan Files:

SCAN	**COUNT**
UDP	724527
SYN	398422
SYNFIN	26034
NOACK	5212
INVALIDACK	3704
UNKNOWN	2498
FIN	2226
NULL	1651
VECNA	1209
FULLXMAS	355
XMAS	232
SPAU	186
NMAPID	143



Analysis of Scans

** MY.NET was replaced with 123.456 in the analysis of the scan files.

UDP SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.213.186	50245	203.164.58.41	6459
123.456.100.230	41730	207.46.204.86	6077
123.456.217.94	33715	123.456.98.133	5543
123.456.98.200	32402	216.15.60.112	5348
123.456.218.130	18420	194.251.249.182	3003
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
28800	98721	6112	90086
6112	94787	28800	89526
53	40592	7778	59846
9753	32153	27015	46323
0	22335	53	43808

SYN SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
24.180.134.156	31901	123.456.223.86	48279
123.456.253.24	30567	123.456.201.78	24781
212.187.94.162	29528	123.456.98.182	9272
24.4.196.167	29528	123.456.203.98	7142
212.64.74.169	22545	123.456.202.94	6669
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
10101	6308	21	144143
53	4770	25	48374
21	3651	27374	21852
2666	1224	2000	12145
20	921	5232	10877

SYNFIN SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
133.1.36.184	14941	142.104.195.55	39
147.8.182.157	4096	123.456.253.112	14
194.204.224.131	3052	212.187.40.220	14
200.194.102.99	1790	123.456.253.114	14
63.204.152.253	1242	63.204.84.150	11
Source Port	**Count**	**Top 5 Destination Port**	**Count**
21	17371	21	17396
109	7148	109	7148
53	1259	53	1259
2340	96	2340	55
0	15	4104	27

NOACK SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	2230	207.172.3.55	326
123.456.217.150	1176	24.16.33.38	248
123.456.219.126	636	142.104.195.55	199
123.456.217.182	443	142.103.53.239	153
123.456.217.126	154	204.210.50.13	122
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	3054	2340	1124
0	339	119	273
18245	202	21536	202
1	167	0	64
36	61	8874	53

INVALIDACK SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	1534	207.172.3.55	240
123.456.217.150	874	142.104.195.55	183
123.456.219.126	400	24.1.112.81	110
123.456.217.182	350	24.130.58.80	101
123.456.217.126	134	142.103.53.239	88
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	2076	2340	937
0	254	119	186
1	165	21536	79
18245	79	0	59
5635	57	6699	37

UNKNOWN SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	1065	207.172.3.55	131
123.456.217.150	551	142.104.195.55	128
123.456.219.126	220	24.130.58.80	68
123.456.217.182	149	142.103.53.239	65
123.456.217.126	84	123.456.6.39	65
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	1335	2340	587
0	136	119	114
12336	115	12336	53
1	110	21536	43
18245	43	62643	32

FIN SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	420	123.456.208.78	1438
209.157.133.43	347	207.172.3.55	64
209.44.81.175	325	142.104.195.55	47
123.456.217.150	193	24.130.58.80	26
129.120.59.15	118	193.253.232.220	25
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	602	113	1438
1163	348	2340	66
1997	325	119	59
1176	118	259	18
1813	118	1710	11

NULL SCAN

Top Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	479	24.180.132.123	134
123.456.217.150	239	123.456.60.11	96
123.456.186.16	93	123.456.60.8	68
123.456.217.182	87	123.456.6.39	57
123.456.219.126	84	123.456.6.44	57
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	796	0	173
0	160	6144	124
23	134	21576	92
65531	125	119	58
16725	92	2340	49

VECNA SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	533	142.103.53.239	136
123.456.217.150	279	142.104.195.55	95
123.456.217.182	127	207.172.3.55	70
123.456.219.126	84	123.456.253.114	62
123.456.217.126	31	193.253.209.94	30
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	698	2340	275
0	74	2875	68
18245	67	21536	67
1	37	119	65
5635	16	2606	28

FULLXMAS SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	133	24.1.112.81	32
123.456.217.150	79	142.104.195.55	27
123.456.217.182	60	204.210.50.13	23
123.456.219.126	41	207.172.3.55	23
123.456.217.126	16	24.3.170.90	13
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	216	2340	90
0	25	119	25
134	11	1498	11
1	9	1421	8
2203	5	3477	8

XMAS SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	112	123.456.253.114	31
123.456.219.126	22	207.172.3.55	14
123.456.217.150	19	64.228.45.111	11
123.456.217.182	15	64.228.45.95	10
123.456.217.126	7	64.230.26.53	9
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	104	2340	56
18245	35	21536	35
0	15	119	11
1	10	1743	10
202	5	1612	7

SPAU SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	73	24.66.236.32	21
123.456.217.150	51	24.130.58.80	14
123.456.217.182	25	207.172.3.55	13
123.456.219.126	22	204.210.50.13	9
130.234.187.194	2	213.105.104.88	8
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	107	2340	51
0	19	1623	11
134	5	119	8
1	4	1787	6
38	4	4666	6

NMAPID SCAN

Top 5 Source IP	Count	Top 5 Destination IP	Count
123.456.217.158	87	207.172.3.55	17
123.456.217.150	26	65.8.217.164	12
123.456.219.126	12	64.228.37.104	10
123.456.217.182	3	212.83.152.34	9
209.252.32.2	2	213.105.104.88	7
Top 5 Source Port	**Count**	**Top 5 Destination Port**	**Count**
2340	74	2340	36
1	11	119	11
0	10	1201	8
5635	3	3812	6
84	3	3657	5

ANALYSIS PROCESS

To perform analysis on the alert files, I used Snort Snarf. To perform analysis on the scan files, I decided to write my own Perl scripts. I wanted to do further analysis on the output I got for each scan type, including examination of the scans originating from MY.NET and the different port pair combinations that were frequently seen, but I ran out of time. This was my first attempt at using Perl. Here are the scripts I used:

To get the total counts for each type of scan:

```
#!/usr/bin/perl
use strict;
use warnings;

my($scanfile);
my(%scanhash);
%scanhash=();
opendir(SCANDIR,"c:\\giac\\scans");
while($scanfile=readdir SCANDIR)
        {
        print "$scanfile starting.\n";
        open(INFILE,"c:\\giac\\scans\\$scanfile");
        while(<INFILE>)
                {
                if (m/^Dec|^Nov|^Jan/)
                        {
                        chomp($_);
                        s/MY.NET/123.456/;
                        s/  / /;
                        my($line)="$_";
open(OUTFILE,">>c:\\giac\\scans\\newscans\\allscans.txt");
                        print OUTFILE "$line\n";

my($month,$day,$time,$source,$flow,$dest,$scan,$misc)=split(/ /,$_);
                        if (exists $scanhash{$scan})
                                {
                                $scanhash{$scan}++;
                                }
                        else {$scanhash{$scan}=1};
                        }
                }
        print "$scanfile done!\n";
        }
foreach(reverse sort {$scanhash{$a}<=>$scanhash{$b}} keys %scanhash)
        {
        print "$_ $scanhash{$_}";
        }
```

To divide the scans into individual files by scan type:

```
#!/usr/bin/perl
use strict;
use warnings;

open(INFILE,"c:\\giac\\scans\\newscans\\allscans.txt");
while(<INFILE>)
                {
                if (m/^Dec|^Nov|^Jan/)
                        {
                        chomp($_);

my($month,$day,$time,$source,$flow,$dest,$scan,$misc)=split(/ /,$_);

open(OUTFILE,">>c:\\giac\\scans\\newscans\\$scan.txt");
                        print OUTFILE "$_\n";
                        }
                }
```

To sort the source ips, dest ips, source ports, and dest ports in an individual scan file by number of occurrences:

```
#!/usr/bin/perl
use strict;
use warnings;

my(%s_ip);
%s_ip=();
my(%d_ip);
%d_ip=();
my(%s_port);
%s_port=();
my(%d_port);
```

```
%d_port=();
open(INFILE,"c:\\giac\\scans\\newscans\\nmapid.txt");
        while(<INFILE>)
                        {

my($month,$day,$time,$source,$flow,$dest,$scan,$misc)=split(/ /,$_);
                        my($source_ip,$source_port)=split(/:/,$source);
                        my($dest_ip,$dest_port)=split(/:/,$dest);
                        if (exists $s_ip{$source_ip})
                                {
                                $s_ip{$source_ip}++;
                                }
                        else {$s_ip{$source_ip}=1};
                        if (exists $d_ip{$dest_ip})
                                {
                                $d_ip{$dest_ip}++;
                                }
                        else {$d_ip{$dest_ip}=1};
                        if (exists $s_port{$source_port})
                                {
                                $s_port{$source_port}++;
                                }
                        else {$s_port{$source_port}=1};
                        if (exists $d_port{$dest_port})
                                {
                                $d_port{$dest_port}++;
                                }
                        else {$d_port{$dest_port}=1};
                        }

open(OUTFILE,">>c:\\giac\\scans\\newscans\\source_ip.txt");
                        print OUTFILE "\n\nSource IP by Volume\n\n";
                        foreach(reverse sort {$s_ip{$a}<=>$s_ip{$b}}
keys %s_ip)
                                {
                                print OUTFILE "$_ $s_ip{$_}\n"
                                }

open(OUTFILE,">>c:\\giac\\scans\\newscans\\dest_ip.txt");
                        print OUTFILE "\n\nDest IP by Volume\n\n";
                        foreach(reverse sort {$d_ip{$a}<=>$d_ip{$b}}
keys %d_ip)
                                {
                                print OUTFILE "$_ $d_ip{$_}\n"
                                }

open(OUTFILE,">>c:\\giac\\scans\\newscans\\source_port.txt");
                        print OUTFILE "\n\nSource Port by Volume\n\n";
                        foreach(reverse sort
{$s_port{$a}<=>$s_port{$b}} keys %s_port)
                                {
                                print OUTFILE "$_ $s_port{$_}\n"
                                }

open(OUTFILE,">>c:\\giac\\scans\\newscans\\dest_port.txt");
                        print OUTFILE "\n\nDest Port by Volume\n\n";
                        foreach(reverse sort
{$d_port{$a}<=>$d_port{$b}} keys %d_port)
                                {
                                print OUTFILE "$_ $d_port{$_}\n"
                                }
```