GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

Yujia Huang

GIAC Intrusion Detection
Practical Assignment #1

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #1

Source of Trace

Snort running outside our firewall.

```
Frame 13949 (86 on wire, 86 captured)
Ethernet II
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiate d Services Field: 0x00 (DSCP 0x00: Default; ECN:
0x00)
    Total Length: 72
    Identification: 0x3a73
    Flags: 0x00
    Fragment offset: 0
    Time to live: 113
    Protocol: UDP (0x11)
    Header checksum: 0xfdd1 (correct)
    Source: 210.217.4.243 (21 0.217.4.243)
    Destination: out.net.x.99
User Datagram Protocol
    Source port: 1357 (1357)
    Destination port: 161 (161)
    Length: 52
    Checksum: 0xa5d3
Simple Network Management Protocol
    Version: 1
    Community: public
    PDU type: GET
    Request Id: 0x2
    Error Status: NO ERROR
    Error Index: 0
    Object identifier 1: 1.3.6.1.4.1.11.2.4.3.10.6.0
    Value: NULL

(Same packets to several different destination ip addresses to our
network detected on 15/04/2001).
```

Detect was generated by:

Snort1.7. The listing above was dumped with Ethereal from snort log file.

Probability the source address was spoofed

Unlikely. Otherwise, they will not they the response.

Description of attack

SNMP public access.

Host 210.217.4.243 was sending S NMP get requests to our many of our hosts. The OID in the requests were 1.3.6.1.4.1.11.2.4.3.10.6.0. It uses 'public' as the community string.

Attack mechanism

© SANS Institute 2000 - 2002, Author retains full rights.

This is a stimulus. 'Public' is often the default community string on many device. The OID (1.3 .6.1.4.1.11.2.4.3.10.6.0) used in the get request is related to HP printer JetDirect software. (to be find out the exact meaning of this oid!!!).

There are several vulnerabilities exits in HP Jetdirect software. E.g., http://security - archive.merton.ox.ac.uk/bugtraq -199911/0281.html reported a buffer overflow in HP JetDirect module.

This is likely to be a reconnaissance. When it found a HP printer, it will try to launch further attack - e.g buffer overflow to get further access.

Correlations:
A search on http://www.apnic.net/ return the following:

```
inetnum              210.216.0.0   - 210.219.255.255
netname              KRNIC -KR-20
descr                National Computerization   Agency
descr                Korea Network Information Centercountry
KR
admin-c              WK1 -AP, inversetech -c              SL40 -
AP, inverse
remarks              KRNIC Allocation Block
remarks              Authoritative Information regard  ing
assignments and
remarks              allocations made from within this block
can also be
remarks              queried at whois.nic.or.kr
mnt-by               MNT -KRNIC-AP, inverse
mnt-lower            MNT -KRNIC-AP, inverse
changed              hostmast er@apnic.net 19990324source
APNIC
```

From http://www.office.ac/abuse/abuse.html

```
Feb 28 20:53:45 UDP: dgram to snmp from  210.217.4.243 :1033 (44
data bytes)
```

From http://www.sans.org/y2k/070100 -1300.htm

```
My Intrusion Detection System logs show attempts to connect to
computers in our organization from one of your computers or a
customers computer. This is likely not within the bounds of
company policy or your reasonable use policy. Please take
steps to insure that this activity is not repeated. My logs
follow:

Site: site-loc - Date: Jun23  - EDT: 20:00.

207.90.225.139 > a.b.c.5
20:23:57.429669 207.90.225.139.1072 > a.b.c.5.161:

|30|2a|02|01| 04|06|a0|1dGetRequest(29)|02|01|02|01|02|01|30|12
  |30|10|06|0c. 1.3.6.1.4.1.11.2.4.3.10.6.0 |05|00
  (ttl 18, id 39039)
```

The oid is the same as the one in the attack we see.

From http://certworks.net/ids/data/snfout.snort_portscan.log/210/217/4/src210.217.4.243.html

```
Jan 4 06:11:21  210.217.4.243 :1545-> 210.174.164.133: 161 UDP
Jan 4 06:11:21  210.217.4.243 :1545-> 210.174.164.132: 161 UDP
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Jan 4 06:11:21 210. 217.4.243:1545 -> 210.174.164.131:161 UDP
Jan 4 06:11:21 210.217.4.243:1545 -> 210.174.164.130:161 UDP
Jan 17 03:06:09 210.217.4.243:4551 -> 210.174.164.135:161 UDP
Jan 17 03:06:09 210.217.4.243:4551 -> 210.174.164.134:161 UDP
```

So we are not the only targe ts.

http://www.securiteam.com/securitynews/HP_Jetdirect_vulnerable_to_Invalid_FTP_Command_DoS.html

HP Jetdirect supports incoming FTP connections . The FTP server which is built-in into the product contains a security vulnerability that allows a remote attacker to cause a Denial of Service attack against the printer.

For those who wish to disable FTP through SNMP, there is an SNMP OID: .1.3.6.1.4. 1.11.2.4.3.5.19.0

From www.fastwave.gr.jp/diarysrv/goma/200102b.html

```
¡ü µÒ
210.217.4.243 ¤«¤é snmp
snmp ¤Ï½é¤á¤Æ¥«¥â¡£¤È¤³¤í¤Ç¡¢whois.nic.or.kr
¤Î½Ð ÎÏ¤¬ÆÉ¤á¤Í¤Ã¥¹¡£¥¦¥§¥Ö¸«¤Ë¹Ô¤Ã¤Æ¤â Æ±¤¸¤À¤Ã¤¿¤ê¤¹¤ë¤Î¤Ç¡¢¤¬
¤Ã¤«¤ê¡£
```

This is titled " It's a test". In Ja paness?, maybe explaining it is not an attack?

Evidence of active targeting

Unlikely. It seems to be just scanning the network (for hp printer?).

Severity

Criticality	3	while the targets are only printers, They may be used to jump to other system. Also maybe able to steal information sent to the printer.
Lethality	2	By itself it is only confidentiality attack.
System countermeasures	3	Quite likely no one remember to patch a printer.
Network Countermeasures	5	Validated restrictive firewall under close monitoring., very limited in/out.
Severity	**-3**	

Defensive recommendation:

Patch the jetdirect software on all printers if possible.

Multiple choice test question

a) This is normal SNMP traffic
b) SNMP get request will not cause security problem, and can be ignored.
c) This is a reconnaissance attempt from malicious host on the Internet.

Answer c)

Detect #2

Source of Trace

GIAC practical *Analyse this* data files.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
20001128 12:02:21 Queso fingerprint [**] 206.65.191.129:43747 -> MY.NET.219.114:568
20001128 12:02:21 Queso fingerprint [**] 206.65.191.129:43748 -> MY.NET.219.114:969
20001128 12:02:25 Queso fingerprint [**] 206.65.191.129:43981 -> MY.NET.219.114:147
20001128 12:02:28 Queso fingerprint [**] 206.65.191.129:44141 -> MY.NET.219.114:775
20001128 12:02:28 Queso fingerprint [**] 206.65.191.129:44154 -> MY.NET.219.114:8
20001128 12:02:33 Queso fingerprint [**] 206.65.191.129:44365 -> MY.NET.219.114:1512
20001128 12:02:33 Queso fingerprint [**] 206.65.191.129:44366 -> MY.NET.219.114:222
20001128 12:02:33 Queso fingerprint [**] 206.65.191.129:44367 -> MY.NET.219.114:1485
20001128 12:02:33 Queso fingerprint [**] 206.65.191.129:44368 -> MY.NET.219.114:370
20001128 12:02:37 Queso fingerprint [**] 206.65.191.129:44538 -> MY.NET.219.114:625
20001128 12:02:37 Queso fingerprint [**] 206.65.191.129:44539 -> MY.NET.219.114:1421

20001128 12:44:22 Queso fingerprint [**] 206.65.191.129:41021 -> MY.NET.219.114:338
20001128 12:44:22 Queso fingerprint [**] 206.65.191.129:41022 -> MY.NET.219.114:877
20001128 12:44:22 Queso fingerprint [**] 206.65.191.129:41023 -> MY.NET.219.114:805
20001128 12:44:22 Queso fingerprint [**] 206.65.191.129:41024 -> MY.NET.219.114:943
20001128 12:44:25 Queso fingerprint [**] 206.65.191.129:41054 -> MY.NET.219.114:502
20001128 12:44:25 Queso fingerprint [**] 206.65.191.129:41055 -> MY.NET.219.114:241
20001128 12:44:26 Queso fingerprint [**] 206.65.191.129:41062 -> MY.NET.219.114:92
20001128 12:44:26 Queso fingerprint [**] 206.65.191.129:41064 -> MY.NET.219.114:61
20001128 12:44:26 Queso fingerprint [**] 206.65.191.129:41066 -> MY.NET.219.114:241
20001128 12:44:26 Queso fingerprint [**] 206.65.191.129:41069 -> MY.NET.219.114:797
20001128 12:44:26 Queso fingerprint [**] 206.65.191.129:41071 -> MY.NET.219.114:969
20001128 12:44:28 Queso fingerprint [**] 206.65.191.129:41092 -> MY.NET.219.114:415
20001128 12:44:31 Queso fingerprint [**] 206.65.191.129:41113 -> MY.NET.219.114:818
```

(a total of 150 different ports to the same host in 43 minutes).

Detect was generated by:
Snort - unspecified version. The date field was reformatted to make sorting by date easier.

Probability the source address was spoofed
Unlikely.

Description of attack
Seeing 183 connection attempts with ECN -Echo and CWR bit set from 206.65.191.129 to MY.NET.219.114 on 150 different ports in 43 minutes.

Attack mechanism

Different operating system or different version of the same operating system behaves differently when the reserve bits are set in the tcp header. Some tools (e.g. Queso) use this to detect the target's operating system, i e., but setting reserve bit 1 and 2 to 1, and observe the response.

Since RFC 2481, reserve bit 1 and 2 are used as ECN Echo and CWR. So now it is possible to receive legitimate packets with reserve bit 1 and 2 set. E.g, a ECN capable host will set ECN -Echo, CWR and SYN flag to 1 when it initiate a TCP connection. If the destination is also ECN capable, the destination will reply with ECN-Echo, SYN and ACK flag set to 1.
Ref: http://www.sans.org/y2k/ecn.htm

Theory 1:
This is a stimulus. 206.65.191.129 was trying to fingerprint MY.NET.219.114.
We can not explain though why it needs to try so many ports?

Theory 2.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

206.65.191.129 is trying to do a TCP ping to MY.NET.219.114 to determine the network performanc e from MY.NET.219.114 to the Internet. The ECN Echo and CWR are set to see if MY.NET.219.114 supports ECN.

Nslookup return 206.65.191.129 as monitor.dslreports.com. Browsing the www.dslreports.com shows that they o ffer line monitoring service, ie. the performance from your ISP to the Internet.
From http://pittsburgh.dslreports.com/faq/monitorfaq/5.+Firewalls+and+NAT#329: -Line Monitoring FAQ - section 5. Firewalls and NAT

> (#329) Where is east coast and west coast monitor? (back)
> The two machines that (currently) do the monitoring are
> ny-monitor.dslreports.com and
> sjc-monitor.dslreports.com.
> The current IPs are **206.65.191.129** and 216.200.176.6
> These two hosts should be added to your fireall if ICMP ping is being blocked.
> If you must enter IP address and not DNS name, then please do an NSLOOKUP or PING to find their current IP address.

From this, we think the likely explanati on of this behaviour is that **206.65.191.129** is just collecting performance information (by looking at the round trip time) by sending connection with ECN -Echo and CWR bit set. This is likely to be requested by someone in the org.

Correlations:
http://pittsburgh.dslreports.com/faq/monitorfaq

From http://www.sans.org/y2k/122200.htm:

```
On Sun 17 Dec 2000 at 21:49 (UTC) we detected a scan of a ho  st
system on our network. This incident appears to have
originated from 206.65.191.129. Either some third party has
compromised 206.65.191.129 and is now using it to attack
others sites or a legitimate users of 206.65.191.129 are
engaging in practices that  are not condoned under most company
or ISP acceptable use policies. Sample logs, times are UTC +
1300, GPS synchronised:

18 Dec 00 10:49:30      tcp   206.65.191.129.57098  o>
130.216.221.75.1358  s
18 Dec 00 10:49:30      tcp   206.65.191.129.57099  o>
130.216.221.75.149   s
18 Dec 00 10:49:31      tcp  206.65.191.129.57113  o>
130.216.221.75.5997  s
18 Dec 00 10:49:31      tcp  206.65.191.129.57114  o>
130.216.221.75.983   s
18 Dec 00 10:49:31      tcp  206.65.191.129.57115  o>
130.216.221.75.170    s
18 Dec 00 10:49:31      tcp  206.65.191.129.57116  o>
130.216.221.75.2232  s
18 Dec 00 10:49:31      tcp  206.65.191.129.57117  o>
130.216.221.75.1488  s
18 Dec 00 10:49:31      tcp  206.65.191.129.57118  o>
130.216.221.75.165   s
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Source: 206.65.191. 129
Ports:
Incident type: Host_scan
re-distribute: yes
timezone: UTC + 1300
reply: no
Time: Sun 17 Dec 2000 at 21:49 (UTC)
```

This is the same behaviour as what we seen here.

Evidence of active targeting
Yes, through probably not malicious.

Severity

Criticality Lethality	2 1	MY.NET.219.114 is likely to be some desktop system. By itself it is only confidentiality attack.
System countermeasures	2	unknown
Network Countermeasures	2.5	With IDS, likely to have a permissive firewall.
Severity	**-1.5**	

Defensive recommendation:
Tighten firwall rules.
User education.

Multiple choice test question
a) Queso scan
b) Nmap scan
c) 3DNS load balancing
d) collecting network performance information.

Answer d).

Detect #3

Source of Trace
GIAC practical Analyse this data fil e.

```
20001229 00:10:46 MY.NET.60.38:1525 -> 209.26.238.82:10078 UDP
20001229 00:10:46 MY.NET.60.38:1525 -> 209.26.238.82:10633 UDP
20001229 00:10:46 MY.NET.60.38:1525 -> 209.26.238.82:1101 UDP
20001229 00:10:46 MY.NET.60.38:1525 -> 209.26.238.82:11125 UDP
20001229 00:10:46 MY.NET.60.38:1525 -> 209.26.238.82:11196 UDP
20001229 00:10:46 MY.NET.60.38:1525 -> 209.26.238.82:11336 UDP
……
```

(A total of 575 different port in 1 minute)

```
20010108 07:44:35 MY.NET.60.38:20 -> 38.203.242.9:21042 SYN **S*****
20010108 07:44:35 MY.NET.60.38:20 -> 38.203.242.9:21043 SYN **S*****
20010108 07:44:35 MY.NET.60.38:20 -> 38.203.242.9:21044 SYN **S*****
20010108 07:44:36 MY.NET.60.38:20 -> 38.203.242.9:21045 SYN **S*****
20010108 07:44:36 MY.NET.60.38:20 -> 38.203.242.9:21047 SYN **S*****
20010108 07:44:36 MY.NET.60.38:20 -> 38.203.242.9:21049 SYN **S*****
……
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

(A total of 81 different port in 20 seconds)

```
20010109 14:30:30 MY.NET.60.38:20 -> 136.160.174.71:2896 SYN **S*****
20010109 14:30:31 MY.NET.60.38:20 -> 136.160.174.71:2897 SYN **S*****
20010109 14:30:31 MY.NET.60.38:20 -> 136.160.174.71:2899 SYN **S*****
20010109 14:30:31 MY.NET.60.38:20 -> 136.160.174.71:2901 SYN **S*****
20010109 14:30:32 MY.NET.60.38:20 -> 136.160.174.71:2902 SYN **S*****
……
```

(A total of 87 different ports in 9 minutes)

Detect was generated by:
Snort - unspecified version. The date field was reformatted to make sorting by date easier.

Probability the source address was spoofed
Could be but unlikely. Note this is a case that internal host is scanning Internet hosts, so it is possible it can spoof another host's ip address on the same network segment and still be able to see the replies.
Since MY.NET.60.38 is likely a compromised host, the real attacker does not need to spoof MY.NET.60 .38's ip address.

Description of attack
TCP port scan. Reconnaissance, Trying to find out open tcp ports on target.

Attack mechanism
If the port on the target is listen, it will reply with flag syn/ack set. Otherwise, the target will reply with flag rst set. Base on the reply, the attacker can find out the active port on the target and plan further attacks.
Noted it also try to make the connection from port 20. This is the port use by active FTP data channel. Sites that are not using stateful filtering ro uter or firewall may allow anything from port 20 through to support active ftp.

Correlations:
This host was also doing the following:

```
20010108 21:26:34 connect to 515 from inside [**] MY.NET.60.38:513 -> 128.8.3.106:515
20010108 21:26:39 connect to 515 from inside [**] MY.NET.60.38:513 -> 128.8.3.106:515
20010108 21:29:04 connect to 515 from inside [**] MY.NET.60.38:513 -> 128.8.3.106:515
```

There are lots of attack/connection attempt to this host. It is likely that this host was compromised and some backd oor was installed. The above activities could be the result of this host executing instructions from its master somewhere in the Internet..

Evidence of active targeting
Likely. It only scan a small number of hosts, but large number of port on each hosts.

Severity
(This is accessing the severity of the MY.NET.60.38 having been compromised).

Criticality	2	not sure what kind host MY.NET.60.38 is
Lethality	5	attacker have backdoor installed and have total control of the machine (able to send from privilege port 20 if it is a Unix box)

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

System countermeasures	1	Unknown - assume the worst
Network Countermeasures	2.5	With IDS, likely to have a permissive firewall.
Severity	**3.5**	

Defensive recommendation:
(for us)
Check for backdoor, install virus protection software. Update os/patches. Tighten firewall rules. Actively monitor IDS alarm.

Multiple choice test question
a) ftp control channel connection
b) active ftp data channel connection
c) passive ftp data channel connection
d) syn scan from my.net.60.38

Answer: d

Detect #4

Source of Trace
GIAC practical *Analyse this* data file.

```
20010118 14:27:56 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.0:37558
20010118 14:28:04 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.0:40997
20010118 14:28:22 connect to 515 from inside [**] MY.NET.70.38:3426 -> MY.NET.0.1:515
20010118 14:28:24 connect to 515 from inside [**] MY.NET.70.38:3430 -> MY.NET.0.1:515
20010118 14:28:24 connect to 515 from inside [**] MY.NET.70.38:3432 -> MY.NET.0.1:515
20010118 14:28:38 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.1:41638
20010118 14:29:03 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.2:41014
20010118 14:29:12 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.2:32979
20010118 14:29:27 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.3:38829
20010118 14:29:59 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.3:33778
20010118 14:30:41 connect to 515 from inside [**] MY.NET.70.38:3466 -> MY.NET.0.4:515
20010118 14:30:51 connect to 515 from inside [**] MY.NET.70.38:3471 -> MY.NET.0.4:515
20010118 14:32:10 connect to 515 from inside [**] MY.NET.70.38:3477 -> MY.NET.0.5:515
20010118 14:32:34 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.5:37656
20010118 14:32:48 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.5:38614
20010118 14:33:23 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.5:38209
20010118 14:33:40 connect to 515 from inside [**] MY.NET.70.38:3503 -> MY.NET.0.6:515
20010118 14:34:13 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.6:33821
20010118 14:34:23 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.6:33821
20010118 14:34:53 connect to 515 from inside [**] MY.NET.70.38:3518 -> MY.NET.0.7:515
20010118 14:36:21 connect to 515 from inside [**] MY.NET.70.38:3533 -> MY.NET.0.8:515
20010118 14:36:35 NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.8:36703
```

……
(A total of 257 attempts to 128 different on MY.NET.0.x to port 515. A total of 137 NMAP ping to 89 different hosts on MY.NET.0.x. Last 9 hours).

Detect was generated by:
Snort - unspecified version. The date field was reformatted to make sorting by date easier.

Probability the source address was spoofed
No. the LPR attack will not work otherwise. Also there is not incentive for the real attacker to do so.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Description of attack
Attack against the LPR service.

CVE: CAN-2000-0917

Attack mechanism
This is a stimulus, exploits against a known vulnerability in some version of the LPR service.
Bugtraq (http://www.securityfocus.com/vdb/bottom.html?section=discussion&vid=1711) has the following discussion regarding this exploit :

> "lpr is a utility which queues print jobs and submits them to a destination. lpr contains a function called checkremote() which returns a pointer to a null terminated character string. This string is passed to syslog() as its primary argument, the format string. As a result, if this string is constructed so that malicious format specifiers can be included, syslog can crash or be exploited to execute arbitrary code. It has been reported that intentional user input into this string is not possible without root access and thus It is considered unlikely that this vulnerability is exploitable."

Correlations:
http://www.whitehats.com/info/IDS457
http://www.whitehats.com/info/IDS456
http://www.sans.org/newlook/alerts/port515.htm
http://www.securityfocus.com/vdb/bottom.html?vid=1711

Evidence of active targeting
Unlikely. It went through most of the hosts on the same subnet. Noted it was also doing nmap tcp ping around the same time on most of the hosts.

Severity

Criticality	2	not sure what kind host MY.NET.70.38 is
Lethality	5	attacker have backdoor installed
System countermeasures	1	Unknown - assume the worst
Network Countermeasures	2.5	With IDS, likely to have a permissive firewall.
Severity	**3.5**	

Defensive recommendation:
Check MY.NET.70.38 for backdoor. Investigate how it get there.
Remove LPR service if it is not needed or patch it. Refer to http://www.securityfocus.com/vdb/bottom.html?section=solution&vid=1711 for detail.
Actively monitor firewall and IDS alarm.

Multiple choice test question
a) MY.NET.70.38 is trying to locate a printer on the local network.
b) MY.NET.70.38 is scanning MY.NET.0.0.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

c) MY.NET.70.38 is exploiting the LPR exploit on MY.NET.0.0,

Detect #5

Source of Trace

GIAC practical *Analyse this* data file.

```
20001210 01:00:52 MY.NET.217.182:1757 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0 ID:6022
DF 21SFRP*U Seq: 0x477   Ack: 0x96899DEF   Win: 0x5010 TCP Options => Opt 32 (32):
2020 2000 3233 3435 3637 3839 0000 0000 0000 0000 0000 0000 0000 0000 0000 EOL EOL
EOL EOL

20001210 01:49:49 MY.NET.217.182:36 -> 207.172.3.46:1757 TCP TTL:126 TOS:0x0 ID:15629
DF **SF***U Seq: 0x770477   Ack: 0xC857B3AC   Win: 0x5010 TCP Options => EOL EOL EOL
EOL EOL EOL SackOK NOP WS: 0 EOL Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt
10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt
10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10 Opt 10

20001210 02:58:52 MY.NET.217.182:0 -> 207.172.3.46:2810 TCP TTL:126 TOS:0x0 ID:19110
DF 2*SF**** Seq: 0x7704E9   Ack: 0x51A17420   Win: 0x5010 51 A1 74 20 23 43 50 10 22
38 65 57 00 00 00 00  Q.t #CP."8eW.... 00 00
..

20001210 03:28:00 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:29286  DF 21SFR*** Seq: 0x104E9   Ack: 0x847187DA   Win: 0x5010 84 71 87 DA 2C C7
50 10 22 38 15 49 20 20 20 20  .q..,.P."8.I     20 00
.

20001210 03:39:20 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:56987  DF 2*SFRP** Seq: 0x4E9A3BF   Ack: 0x90A0   Win: 0x5010 TCP Options => Opt
32 (32): 2020 2000 0402 0C6D 11DC 0027 0000 0000 0000 0000 0000 0000 0000 0000 0000

20001210 03:49:46 MY.NET.217.182:172 -> 207.172.3.46:2810 TCP TTL:126 TOS:0x0
ID:55728  DF 2*SFR*A* Seq: 0x7704E9   Ack: 0xBF5F9875   Win: 0x5010 TCP Options =>
EOL EOL EOL EOL  00 00                                                  ..

20001210 03:53:07 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:30229  DF *1SF*PA* Seq: 0x4E9C9AA   Ack: 0x229B5C   Win: 0x5010

……
```

(A total of 2564 log entries regardi ng oos or packet with strange flag set)

Detect was generated by:

Snort - unspecified version. The date field was reformatted to make sorting by date easier.

The sa, ss, oos entries have been merged together and sort by date/time. The oos entries are join ed into one (long) line, while very unpleason for the eyes, sorting them by date with entries from all log files is a log easier.

Probability the source address was spoofed

Unlikely.

Description of attack

This is not an attack. The large amount of oos pa ckets was caused by faulty hardware.

Attack mechanism

```
207.172.3.46 = reader4.news.rcn.net
```

Theory 1:

These packets were crafted, as reported by snort.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

It is strange why it need to send so many similar one. Also, the flag is kind of random, all sort of flag imaginable. While a hacker may try this, but …

Theory 2:
All these oos packets are the result of a hardware fault on some network device that triggers lots of snort alarm.

```
Lets just pull out the data for one connection and have  a close
look.

20001210 03:28:00 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:29286  DF 21SFR*** Seq: 0x104E9   Ack: 0x847187DA   Win: 0x5010 84 71 87 DA 2C C7
50 10 22 38 15 49 20 20 20 20  .q..,.P."8.I     20 00
.
20001210 03:39:20 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:56987  DF 2*SFRP** Seq: 0x4E9A3BF   Ack: 0x90A0   Win: 0x5010 TCP Options => Opt
32 (32): 2020 2000 0402 0C6D 11DC 0027 0000 0000 0000 0000 0000 0000 0000 0000
20001210 03:53:07 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:30229  DF *1SF*PA* Seq: 0x4E9C9AA   Ack: 0x229B5C   Win: 0x5010
20001210 04:03:10 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:54824  DF *1SFRP** Seq: 0xD004E9   Ack: 0xE56CA33B   Win: 0x5010
20001210 04:18:25 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:30476  DF 21SFRPA* Seq: 0x4EA1839   Ack: 0xCB1A1   Win: 0x5010 20 00
.
20001210 04:28:12 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:48734  DF *1SFRP** Seq: 0x5404EA   Ack: 0x3343BBD3   Win: 0x5010 20 00
.
20001210 04:33:37 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:17182  DF *1SFRP** Seq: 0x4EA407A   Ack: 0x11C16A   Win: 0x5010
20001210 04:40:51 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:18734  DF *1SFRPAU Seq: 0x4EA5367   Ack: 0xC969   Win: 0x5010
```

Noted the sequence number increas e kind of logically (the one in **bold**), ie, by a small amount over time, typical NNTP client profile, the other look like random value - assume they were corrupted.

```
20001210 02:58:52 MY.NET.217.182:0 -> 207.172.3.46:2810 TCP TTL:126 TOS:0x0 ID:19110
DF 2*SF**** Seq: 0x7704E9   Ack: 0x51A17420   Win: 0x5010 51 A1 74 20 23 43 50 10 22
38 65 57 00 00 00 00  Q.t #CP."8eW.... 00 00
..
20001210 03:28:00 MY.NET.217.182:2810 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:29286  DF 21SFR*** Seq: 0x104E9   Ack: 0x847187DA   Win: 0x5010 84 71 87 DA 2C C7
50 10 22 38 15 49 20 20 20 20  .q..,.P."8.I     20 00
.
```

Noted the destination port number (2819) is the same as the source port of the second line.

```
also this:
20001210 06:06:43 MY.NET.217.182:10 -> 207.172.3.46:2819 TCP TTL:126 TOS:0x0 ID:21221
DF 21SFR**U Seq: 0x7705A6   Ack: 0xB3DF35E3   Win: 0x5010 00 77 05 A6 B3 DF 35 E3 15
E7 50 10 22 38 4D EC  .w....5...P."8M. 00 00 00 00 00 00
......
20001210 06:09:04 MY.NET.217.182:2819 -> 207.172.3.46:119 TCP TTL:126 TOS:0x0
ID:13134  DF **SF**** Seq: 0x5A6BB3E   Ack: 0x39072061   Win: 0x5010 0B 03 00 77 05
A6 BB 3E 39 07 20 61 00 03 50 10  ...w...>9. a..P. 22 38 38 EF 20 20 20 20 20 00
"88.     .
```

Kind of suggest there may be some memory corruption.

Also noted that the communication seems to be going ok despite the 21SFR*** flag. Maybe the faulty hardware is the sensor host or the switch it connected to? Otherwise, The connection would be closed by 207.172.3.46.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Also Most of the oos packets are related to port 119 (NNTP). If we assume that the hardware fault happen randomly, this would make sense because of the large NNTP traffic volume.

Also notice oos are between inside and outside, not a single one between internal hosts.

All these lead us to believe the more likely reason for most if not all of the oos packets are cause by hardware, possibly on a network segment near the sensor.

Correlations:
See the attack type table in the "analysis this' assignment.

Evidence of active targeting
No.

Severity
This is a False positive.

Criticality	2	not sure what kind host MY.NET.217.182 is
Lethality	1	attacker have backdoor installe d
System countermeasures	1	Unknown - assume the worst
Network Countermeasures	2.5	With IDS, likely to have a permissive firewall.
Severity	**-0.5**	

Defensive recommendation:
Fix the faulty hardware.

Multiple choice test question
a) Hping
b) Nmap tcp scan
c) Cause by faulty hardware.

Answer c)

© SANS Institute 2000 - 2002, Author retains full rights.

Yujia Huang

GIAC Intrusion Detection
Practical Assignment #2

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

IDWG Introduction

IDS is widely used along with firewalls as an integral part of the security infrastructure for many organisations. IDS typically composes of sensors, analysers and managers. The sensors collect events of interests and pass the event information to analysers. The analysers process the event information, summarise them and sent them the manager to take further actions, ie. block traffic from attacking sites. The manager gathers information from analysers and compose an overall picture of the system health. It also provides an interface to control and monitor all IDS component.

Quite often sensors, analysers and managers are in different hosts. E.g, to monitor traffic on unprotected network, it is best to just have the sensors there and leave the analyser and manager inside the firewall. With the increased use of switched network, it is also necessary to have at least one sensor on each network segment and/or use host based IDS to provide sufficient coverage.

While components from the same vendor may have no problem communicate with each other using proprietary protocol, trying to integrate components from different vendor is much harder. Yet some org may find that they have a need to use components from different IDS vendors. The reason could be:

- Use the best of its breed components.
- Have better coverage. Some products may only run on some platforms. If you have many different platforms in your environment, you may have to use different brands of products.
- Defence in depth - If you run different brands of products, one may pick up attacks the other products missed, and vice vasa.
- Intrusion detection information can be exchanged with other organisation easily.

Without a standard message format and communication mechanism between different IDS components. Integration of different IDS is a very difficult, time consuming and costly task.

To address these problems, The IDWG has proposed a standard infrastructure to facilitate the smooth communication between IDS components and different IDS products. They are currently in draft status and will be ratified soon. The IDWG has produced the following drafts

- Intrusion Detection Message Exchange Requirements [1] - defines what the infrastructure need to address.
- IDMEF Data Model and XML Representation [2] - define the structure of the message that flow between IDS components and how to use XML to represent them.
- IAP Transport [3] - define how intrusion alert data are transported between intrusion detection elements across IP networks.
- Others. The complete list of IDWG documents can be found on [5]

This rest of this document will give a overview of the data structure and the communication mechanism for sending IDS alert message.

It is assumed the audients have basic understand of IDS and object oriented design concepts.

Intrusion Detection Message Exchange Format

The Intrusion Detection Message Exchange Format (idmef) defines '*data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and the management systems which may need to interact with them*' [2]

The data model is defined using object-oriented terminology, ie. inheritance and aggregation, etc. In this model, the root class is the `idmef Message` class, which has two sub classes, `Alert` and `Heartbeat`, each represents alert message and heartbeat message.

Let take a closer look at the most used `Alert` class. Each alert message (an instance of the `Alert` class) is composed from the following sub-components

- `Analyser`,
- `CreateTime`,
- `DetectTime`,
- `AnalyzerTime`,
- `Source`, `Target`,
- `Classification`,
- `AdditionalData`.

Each of the above sub-components can have sub-components too. e.g, `Source` is compose of `Node`, `User`, `Process`, `Service`.

The same process continue until each sub-component is of basic data type.

The idmef has defined the fundamental classes `Alert`, `HeartBeat` and all the sub-component class (`Analyser`, `CreateTime`, `DetectTime`, etc).

Naturally, in the object oriented world class can be extended. The extended class must have the all the attribute and behaviour of its base class, but it can also has additional attributes and behaviours. E.g., if ISS need to include some information that does not fit in any of the existing class. they can extended the appropriate class(say the `Alert` class) to include those information. Suppose a ISS component sent such an alert to a non-

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

ISS manager, while the non-ISS manager may not be able to extract the additional ISS specific information, it can still decode the alert and extract the essential information properly, because this message is still derived from the Alert class.
The draft has chosen XML as the message encoding standard, this is because XML is very suitable to represent the data structure discussed above, it is easily extendable, and has very wide industry support.
The standardisation of the alert message data model and format benefit the end users, the IDS software vendors and system integrators.
To the software vendors, they can concentrate on what they do best, be it the sensors, analyzers or the managers. E.g., they no longer need to worry how to interface with different databases, as long as there is a gateway for each database that accept idmef message.
To end users and system integrators, it is a lot easier to assemble an IDS for their client by choosing the suitable components from different vendors. This reduces system complexity and assembly time significantly.

Intrusion Alert Protocol

While `idmef` defines how message are structured and presented, The Intrusion Alert Protocol (IAP) define how connection between IDS components are established, and how message are exchanged.
From a user's perspective, the following are of interests.

Authentication

IAP runs on top of TLS. It use the mutual authentication feature in TLS to ensure the two parties involved in message exchange will authenticate each other's identity before they communicate with each other. (TLS is a slightly modified version of SSL3.0)
This is very important. In a IDS system, some component may situate in a unprotected or hostile environment. If an attacker can send spoofed alert to the manager, that could have serious consequence, e.g., if the manager react the alert by blocking connection to the alleged 'attacker', the real attacker has just accomplished a successful DOS attack. Likewise, if someone can spoof the manager, they can disable the sensor easily.
To support this, all parties engage in exchanging Ids messages must have their own digital certificates first.

Confidentiality

Since IAP runs on top of TLS, IAP traffic is encrypted.
This is important because we don't want the attacker know what we know about them.

Connection Establishment

Quite often, IAP traffics need to pass through firewalls. IAP is designed to be firewall friendly, ie, IAP traffic can use proxy and can be initiated from either the sender or receiver.
For most organisations, it is preferable to have outgoing traffic through their firewall to minimise their exposure. If they have a sensor outside their firewall, it is better to let the manager to initiate a connection to the sensor. On the other hand, if an organization has outsource the management of their IDS to a service provider, then it make sense for the sensor on the inside to connect to the manager on the service provider's network. In both cases, the connection is established from inside to outside.

Status

The IDWG documents are still in draft status and we have not seen any major vendors releasing product that support it yet. But all the major IDS players are in IDWG and we expect many of them will support the IDWG standard when they are ratified.
An IDMEF XML plugin is available for snort [4]. Hopefully the other will follow.
From the author's experience, this is a very exciting development. While working on the GIAC practical assignment, I need to analyse over 200M of log entries from snort log files. A significant amount of time was spent on writing some perl script to break each log entries into fields before they can be imported into a database for further analysis. This is because different type of log file has different format. Event for the same type of log files, different types of event also have different format.

Reference

[1] Intrusion Detection Message Exchange Requirements <draft-ietf-idwg-requirements-05.txt>
http://www.silicondefense.com/idwg/draft-ietf-idwg-requirements-05.txt
[2] IDMEF Data Model and XML Representation http://www.silicondefense.com/idwg/draft-ietf-idwg-idmef-xml-03.txt
[3] IAP Transport Draft
http://www.silicondefense.com/idwg/draft-ietf-idwg-iap-05.txt
[4] A Snort plugin that creates IDMEF XML message alerts http://www.silicondefense.com/idwg/snort-idmef/
[5] All IDWG related information:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

http://www.silicondefense.com/idwg/

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Yujia Huang

GIAC Intrusion Detection
Practical Assignment #3

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Summary

There were large amounts of attacks from the Internet. Detail of the type of attacks and where they are from can be found in the Finding section.
We had found some internal hosts that are most likely have been compromised. Those hosts and a brief analysis can be found in the Sign of successful attack section. Please note that is not a complete list of all the compromised hosts. A detail analysis is out of the scope of this document.
There are some signs that some of the network devices may have some hardware fault. The hardware fault is responsible for some of the oos packet detected by snort.

Analysis process

All analysis is based on the snort log files provided by the GIAC Enterprises. This section discussed how the data are analysed.

Tools used

Perl was used to reformat the raw data and convert them into CSV files.
Perl, **grep**, **sort** was used to search text data file.
Oracle database and **Sql*Plus** were used to summarised the data.
Microsoft Access /Excel was used to access the oracle database and generate reports.
Microsoft Excel was used to generate charts.

The Process

The raw data is about 2 00 MB. There are three type of log files, sa, ss and oos, each has their own date and time format. The oos log files also consist of more than one line per log event. A Perl script is created for each type of file to convert the logging time to a standard format. For oos log file, all lines for the same event are joined into a single line. After the reformatting, correlating events from different types of log file can be done easily using **grep** and **sort**.

Each log entry is then broken into the following fiel ds:

Field	Field description
ldate	timestamp from the log entry,
src	source ip,
dst	destination ip,
sport	source port,
dport	destination port,
type	type of activity,
flag	ip flag,
dsc	Any further description of the activity.

The data is converted to CVS files and then loaded into Oracle database tables **ss**, **sa**, **oos**.

They are then summarised into table **sdtf** with the following fields:

Field	Field description
src	source ip,
dst	destination ip,
type	type of activity,
flag	ip flag,
cnt	The number of records with the above fields.

Table **sdtf** are summarised again to generate the following tables:

Table **outtype**: Keep a list of all the attack types originated from outside.
Table **intype**: Keep a list of all the attack types originated from inside.
They all share the same structure below:

Field	Field description
type	Type of attacks
dsrc	Distinct source ip
ddst	Distinct destination ip
total	Total log entries of this type

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

mindate	The earliest date this type of attack was detected
maxdate	The last date this type of attack was dectected.

Table **AttackerHost** : Information about the attacker.

src	Attacker's ip address
type	Type of attacks
ddstcnt	Number of distinct destination ip
total	Total log entries of t his type
mindate	The earliest date this type of attack was detected
maxdate	The last date this type of attack was detected.

Table **TargetHost** : Information about the target of attacks.

Field	Field description
dst	Target's ip address
type	Type of attac ks
dsrccnt	Number of distinct destination ip
total	Total log entries of this type
mindate	The earliest date this type of attack was detected
maxdate	The last date this type of attack was dectected.

Table **FishyLocalHost** : Information about hosts in my. net that 'attack' others, ie, the source address is in the form of 'my.net.xxx.xxx'.

src	Attacker's ip address
type	Type of attacks
ddstcnt	Number of distinct destination ip
total	Total log entries of this type
mindate	The earliest date this type of at tack was detected
maxdate	The last date this type of attack was detected.

To improve signal -to-noise ratio, some records are filtered out when they are deeming just *annoyance* .

The above tables are then sorted and visually inspected, when suspicious act ivities are noticed, all relevant log entries (from the text file) are extracted and sorted by time, and manually inspected and analysed. This normally also involves lots of ad -hoc database queries on relevant events, nslookup, search on the Internet, etc. www.google.com seems to be most helpful in many cases.

A lot of ad -hoc queries were also run to pick up abnormal patterns, e.g., a query to pick up hosts that connect to more than 30 ports on a single destination. Hos ts that fit these criteria are likely to be doing some scan.

Limitation

The following limitation has limit our ability to proper analysis the data properly.

Incomplete/insecure data

Many gaps were found in the log file. More analysis on this can be found i n the Finding session.
It is also unclear how well the log files are protected. Without the assurance of high fidelity log data, the analysis may not be correct because some critical data may have been intentionally or unintentionally altered or deleted.

No packets capture file

The log file produced by snort is often too brief. The unavoidability of packets capture files has prevented us to draw a conclusive conclusion in many cases. (Snort can log the trouble packets but this is not available to us from GI AC Enterprises)

No information on network structure, application profile, etc.

The lacks of these informations often leave us with too many possibilities, and unable to tell which one is the likely case.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Finding

Gaps in data

We were told some of the log files were unable because power or disk problem. To find out how many data were missing, we run a query to count number of log entries per hours. After plotting the hourly count data to a chart, we observed that when a particular type of log file is available for a particular date, the log file normally covers the whole 24 hours period. In other words, we do not notice log file stop in the middle of the day. This can not be explained using the 'no space' theory. If it was a space problem, we would expect logging stop randomly within the 24 hours period. May be we never have space problem? If so, then what is the real problem?

May worth to examine the admin procedure. Were the log file not archived for some reason or the log file were deleted by attacker?

Very open network

It looks like that a very permissive firewall is used between the GIAC Enterprises and the Internet.

Gaming activity

There are about 500 hosts in MY.NET sending udp traffic out to the Internet. Many of them are going to the well know port of Internet games. It is most likely Internet gaming is allowed within the GIAC Enterprises.
The gaming traffic generates a lot of noise, and disguises the malicious traffic.
On pattern we have observed is that a large number of internal hosts are sending small amount (<10) of mostly UDP packets to many external hosts.
In most case, there is no sign of return traffic. We can think of two possible explanations:

1. The incoming route to the Internet and outgoing route to the Internet are different, and the sensors were in a position that they can only see the outgoing traffic.
2. The internal host were probing for game servers, and they were unable to get them because they are unavailable or the traffic were blocked by firewalls.

Types of Attack

Below are the type of attacks detected by snort. Generally, more detail information can be found by doing a search on the attack type on http://www.whitehats.com .
Attacks originate from outside and intside are listed in two tables, because even for the same type of attack, they could have different implication depending on where they are originated.
It should be noted that there are many false positive in the snort log. E.g, many of the gaming traffic were reported by snort as scan.
Some time snort may also classify an attack to the wrong category, e.g, a SYN scan that scan a large range of port may be reported by snort as `connect to 515 from outside` when the port number happens to be 515.
The following table give an overall feel of the intensity of the attack. They also give some lead on what need to be investigated further, ie. the *lethal* attacks.

Attacks Originated from outside:

The targets of the *lethal* attacks should be checked to make sure they are ok. Unfortunately, this is not an easy task because the number of hosts involved. Only limited analysis has been done following the lead in this table.

TYPE	Src-dst count	DISTINCT SOURCE	DISTINCT destination	TOTAL Events logged	COMMENT
SYN	144665	392	31367	343840	
SYN-FIN scan!	51150	37	27067	51192	
SF**	47627	28	25952	47678	
SYNFIN	25856	42	19811	25871	
connect to 515 from	2916	10	2877	4238	Need to check if targets are ok
WinGate 1080 Attempt	1181	474	572	2239	Need to check if targets are ok
Null scan!	554	527	173	826	
NOACK	447	349	145	598	
NULL	394	373	114	638	
UDP	353	76	209	36821	
INVALIDACK	348	281	151	439	
UNKNOWN	271	185	143	427	
SMB Name Wildcard	176	90	167	437	
Watchlist 000220	132	46	100	105918	
VECNA	119	104	52	161	

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

NMAP TCP ping!	113	46	26	296	
Queso fingerprint	111	52	72	710	
TCP SMTP Source Port	88	5	88	100	
Back Orifice	74	10	71	77	Need to check if targets are ok
FIN	60	55	33	1499	
Watchlist 000222	56	31	19	2401	
Tiny Fragments - Pos	48	26	12	5333	Need to check if targets are ok
XMAS	44	44	17	67	
External RPC call	42	15	25	59	Need to check if targets are ok
Attempted Sun RPC hi	34	16	23	2053	
SUNRPC highport acce	27	25	19	204	Need to check if targets are ok
Broadcast Ping	24	24	1	154	
FULLXMAS	16	14	14	30	
NMAPID	13	11	11	15	
DNS udp DoS attack d	12	8	6	16146	
Probable NMAP finger	6	5	6	8	
Russia Dynamo - SANS	1	1	1	104	??
STATDX UDP attack	1	1	1	1	The target (MY.NET.6.15) was check. No sign of compromise found.
Happy 99 Virus	1	1	1	1	Need to check if targets are ok
SNMP public access	4	2	4	173	
site exec - possible wu-ftpd exploit - GIAC000623	3	3	3	3	Target are MY.NET.130.98, MY.NET.97.162 and MY.NET.156.127.
OOS	563	505	409	1017??	I must have delete some digits form the total!

Below are some checks we have done based o n the above tables.

MY.NET.97.162

This is the target of a 'site exec' attack so detail data is examined. It show it send many udp packets out to 128.11.20.140 on 20010101 around 23:00, ten days after the 'site exec' attack. A lookup on 128.11.20.140 return baldur.won.net. www.won.net seems to be hosting many games. So hopefully, this is ok if someone was 'working/gaming' around those hours.

Type of Attacks Originate from inside:

If internal hosts were involved in attacks, they are likely have been compromised. A detail analysis for each type is given after this table.

TYPE	Src-dst count	DISTINCT SOURCE	DISTINCT destination	TOTAL Events logged	COMMENT
UDP	188442	437	112855	783577	See discussion below
SYN	23434	259	19924	104710	
NMAP TCP ping!	130	1	130	262	
connect to 515 from inside	98	10	98	159	See discussion below
SNMP public access	18	18	3	418	See discussion below
SMB Name Wildcard	4	3	4	78	See discussion below
Russia Dynamo - SANS Flash 28-jul-00	1	1	1	442	
Tiny Fragments	1	1	1	7	
INVALIDACK	1252	20	289	3366	

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

NOACK	1705	15	321	4734	
UNKNOWN	698	18	201	2125	
VECNA	357	6	164	1068	
NULL	306	12	209	1078	
FIN	225	7	165	741	
XMAS	62	5	46	179	
FULLXMAS	126	6	82	332	
SPAU	59	5	41	180	
SYNFIN	55	7	39	166	
NMAPID	52	6	38	130	
OOS	3471	589	3109	12522	

UDP

There are 437 internal hosts sending 783577 udp packets to 112855 different hosts on the outside! After sampling a couple of them, we found most of them are going to gaming port. We believe most of them are gaming traffic.
Obliviously, just because the destination port is used by some popular games does not mean they are necessarily gaming traffic. They could well be malicious traffic *borrowing* the gaming port to avoid attention.
There are also a lot of traffic goes to *random* ports that we just can tell what they are.
The difficulty here is that there are just too much information to be processed and at the same time not enough information to draw a conclusion in most case. We just don't have enough time.
On thing we can not quite explain is that the inside host on average only send 7 packets to each hosts. Is this a normal behaviour? We think this should be further investigated.
The GIAC Enterprises may want to consider their security policy on gaming or enforce the existing policy.
We also found hosts that clearly show that they are doing scanning.

SYN

There are 259 internal hosts making 104710 connections to 19924 external hosts.
Similar to the UDP traffic analysis above, we think many of them are gaming traffic.
Could some of them are malicious traffic? Yes. It is just the number of hosts and connection involved prevent us to do further analysis.
We do find some host that are scanning the Internet.

OOS and INVALIDACK, SPAU etc.

There are a large number of oos packets and packet with funny flag set originated from internal hosts.
In general, there is no clear attack pattern, e.g., trying the same *attack* on different target, port, etc. The number of *targets* is normally small.
A close look at some of them indicates they are more likely caused by some faulty hardware. Please refer to detect assignment #5 for a detailed analysis.
Under normal circumstance, these type of packets originate from internal host will be a sign of hosts being compromised and trying to attacked others. Because of the faulty hardware, it is very difficult for us to tell because if there are real oos amount the (faulty) hardware generated one, the S/N is so low that we just can not easily pick them up.
The faulty hardware should be located and fix ASAP.

Connect to 515 from inside

Port 515 is normal used by the LP service. There are some buffer overflow exploits on the LP service and is often used by hacker to gain access to systems. [Ref: http://www.whitehats.com/info/IDS457]
It is highly unlike that there is any legitimate reason to print to the Internet. So any connection attempts to port 515 on the Internet should be investigated.
Sometimes this is a false alarm. E.g, we notice a SYN scan was reported as 'connect to 515' because it just happened to be scanning port 515, but that is something we need to investigate anyway.

SNMP public access

SNMP could reveal a lot of information about a host. Most often, only read access is allowed. *Security* is controlled by a community string, 'public' is the most used one.
There are 18 internal hosts using the 'public' community string to communicate with 3 internal hosts. While they are likely to be legitimate, it is better not to use the 'public' as the community string.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

SMB Name Wildcard

There are 3 internal hosts doing SMB Name Wildcard query to 4 internal hosts. While there are legitimate reason to use this, it may still worth to have a check.

NMAP TCP Ping

Host MY.NET.70.38 tried this on 130 internal hosts! Definitely fishy. Details on detect #4.

Russia Dynamo - SANS Flash 28-jul-00

There is only one instance. Most likely false positive.
The traffic was between MY.NET.205.138:6699 and host on a Russia network that were found malicious earlier. Hopefully false positive.
Some background on http://www.sans.GIAC Enterprises/y2k/073100-1030.htm

Tiny Fragments

This is one of the techniques used by hacker to by pass some filtering routers or firewalls.
This only happen once between MY.NET.219.122 and 208.162.62.208 on 20001129. MY.NET.219.122 also *happened* to connect to 128.2.166.68 on port 515 3 hours prior to the sending of Tiny Fragments. We feel this worth further investigation.

Source of Attack

In total, there are 4981 external hosts trying to scan/reconnaissance/attack hosts on my.net.
The top list of those hosts can be found in Appendix A. The list is order by (number_of_attack type_tried + number_of_acket/500). While this does not mean the host listed in the top are the most dangerous attackers, it does filter out the casual attackers from the more devoted one. Some simple scans, e.g syn scan, are removed form the list.
Please note that this method may miss the very stealth attack, ie. those do their attack from many different hosts, but keep the activity on each of those hosts very low.

Target of Attack

The top targets are list in Appendix B.
The hosts are ordered using weighting of sum(cnt) + total/500, which means hosts that were attacked by many different hosts, or different type of attack, or being repeatedly attacked will have a high weighting.
Some *trivial* attack types have been filtered out to improve S/N (signal to noise) ratio.
The purpose of this listing is to show the scale of the attacks, and the *popular* targets. If there are any infrastructure hosts are in this list, closer attention should be given to check the detail log. That may be a sign of active targeting.
This list is just trying to give some *clue* on what to look for. While the weighting formula is a bit too simple (e.g., it would be better to give each type a weighting rather than treating each type equally), we still believe the best way is to manually browse through this list and do further investigation if required.

Signs of successful attack

One of the most useful way to tell wether any host in MY.NET was compromised is to see if any host in MY.NET behave abnormally, e.g., doing scan to other hosts.

The following table list internal hosts that *attack* others, ie., they appear as the source of a scan/reconnaissance/attack. These hosts deserve some close attention.
One problem is that there appear to be many gaming activities going on. Many of them were reported as UDP scan or SYN. Without the packet capture file, it is hard to determine whether they are gaming traffic or not.
Below are some investigation we have done to some of the hosts. Please note that is very incomplete.

MY.NET.163.17:
On 20001220, this host scan 803 port on 148.243.214.7. This is unlikely to be a normal behaviour and should be further investigated. Prior to this, there were scans to MY.NET.163.17 and in particular, an attempt to connect to port 515 (LP) from 141.211.176.99.

MY.NET.100.230
This host sent 50208 UDP packets to 7070 distinct external hosts on port 53. Most were from port 32780. If this is a DNS server, it may be normal, but it is unlikely to be a DNS server. The reason being:

1. The from port were fix most of the time, mostly from 32780, some from 32781 and 32787. This is not the usual behaviour of DNS software (at least not bind).

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

2. The traffic were distributed very unevenly. The daily count (of packet sent) ranging from 14 to 10654.

This host also connected to 1115 distinct hosts on port 25. We don't think this is normal sendmail traffic because

1. Very high connection rate (>200 per minute),
2. There is a very clear pattern that it go through each host in a subnet.

Base on the above, we think this host was compromised.

MY.NET.98.238

This host does nothing but scan, lot of them, 5779 different host to be precise. It systematically scanned the 172.147 network on prot 27374.

This host was the target of 'Sun RPC hight port accesses before it started to scan 172.147. There is not enough evidence to say this is how it got compromised, but it is clear that it has been compromised.

MY.NET.253.24

This host systematically scan external hosts on port 25. While some of the traffic detected by snort as SYN scan could be legitimate mail traffic, if this host is a mail server. Most of the snort alert showed clearly they are not e-mail traffic, ie, there is a clear pattern of this host scanning subnet by subnet, and the pattern show up around midnight.

We believe this host is compromised.

MY.NET.210.70

Snort reported this host is doing a large number of udp scan.

Looking at the detail log, shows that most of the traffic went to destination port 27015-27017 and 28800, the port number used by game 'half life' and 'starsiege'.

Snort reported this host is doing a large number of udp scan. A simple database query return the following result:

Date	Hour	Number of destination hosts
1216	16	1368
1216	23	1419
1217	00	1299
1217	02	188
1217	03	971
1220	01	133
1220	02	6

If someone were burning midnight oil in the office that night, it is most likely that it is just gaming traffic.

On thing we can not quite explained is that why it need to communicate with over 3000 hosts on the Internet, and only exchange less than 6 packets with each of them in most case. There were only about 40 hosts receive more than 6 packets, the max number of packet sent to a single host is 44.

Another thing is that this hosts never got anything back, at least that was what the snort log said. Maybe the incoming route and the out going route were different for some reason, or all those incoming replies were blocked by the firewall, or even that the outgoing packet were blocked by the firewall? If that is the case, this is a very patient gamer.

MY.NET.217.182

This host sending out large number of OOS packets, if we assume this was caused by faulty hardware, then we may assume that this host was sending a large number of packets out - if we also assume the faulty hardware corrupted x percent of the packet passing through.

Most of those packets are from source port 2340. According to http://www.isi.edu/in-notes/iana/assignments/port-numbers, this port is for WRS Registry, what ever that is.

There were 113 external hosts communicated with this host on port 2340, there is no clear pattern on the destination port.

There were only 13 log entries related to destination port 2340 on this host, that all have abnormal flag. This suggest maybe traffic to port 2340 on this host is expect and snort were configured did not report connection attempt to it.

Hopefully this can be clarified easily by a phone call the owner of this host and find out if there is some legitimated server running on port 3420.

MY.NET.60.38

Compromised. See Detect #3 for detail.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.70.38
Compromised See Detect #4 for detail. **MY.NET.98.151**

Recommendation

Tighten firewall rules.
Check for back door and virus regularly.
Consider security policy on Gaming.
Make IDS aware of firewall rules to reduce false positive.
Standard log file format, Maybe XML.
Keep packet capture log files.
Fix faulty hardware.
Check network structure and make sure have com plete IDS coverage.

Appendix

Note:
DDST = Distinct destination
CNT : total number of source -destnation pair.
TOTAL: total number of events recorded.

Table - Source of Attack

SRC	TYPE	DDST	CNT	TOTAL
195.56.182.206	SYN-FIN scan!	9878	9878	9878
	oos	14919	14919	14919
194.234.48.26	SYN-FIN scan!	8565	8565	8565
	oos	12473	12473	12473
211.34.40.1	SYN-FIN scan!	17604	17604	17604
147.8.182.157	SYN-FIN scan!	4096	4096	4096
	oos	5907	5907	5907
194.204.224.131	SYN-FIN scan!	3052	3052	3052
	UDP	3	3	3
	oos	3596	3596	3596
139.130.61.206	SYN-FIN scan!	1914	1914	1951
	oos	2566	2566	2614
200.194.102.99	SYN-FIN scan!	1790	1790	1790
	oos	2088	2088	2088
194.197.170.7	SYN-FIN scan!	1580	1580	1580
	oos	1973	1973	1973
63.204.152.253	SYN-FIN scan!	1242	1242	1242
	UDP	4	4	4
	oos	2133	2133	2133
141.211.176.99	connect to 515 from	2195	2195	2236
193.253.202.9	SYN-FIN scan!	706	706	706
	oos	1030	1030	1030
132.68.37.141	SYN-FIN scan!	630	630	630
	oos	893	893	893
209.217.166.69	connect to 515 from	710	710	713
212.179.79.2	Watchlist 000220 IL-	41	41	48786
24.113.198.51	FIN	1	2	2
	FULLXMAS	1	2	2
	INVALIDACK	1	20	27
	NMAPID	1	2	2
	NULL	1	1	2
	Probable NMAP finger	2	2	4
	oos	2	68	148
	VECNA	1	3	5
	UNKNOWN	1	4	5
	SYN-FIN scan!	1	1	1
	SPAU	1	2	2
	Null scan!	2	2	8
	NOACK	1	21	26
63.11.25.117	External RPC call	1	1	7
	FIN	4	4	4
	TCP SMTP Source Port	84	84	84
	SYN-FIN scan!	4	4	4
	VECNA	6	6	6
12.77.204.44	WinGate 1080 Attempt	83	83	91
212.179.27.111	Watchlist 000220 IL-	2	2	39015

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

207.114.4.46	WinGate 1080 Attempt	74	74	91
209.212.128.47	WinGate 1080 Attempt	71	71	111
209.94.199.202	Back Orifice	32	32	32
	UDP	32	32	32
63.229.92.11	SYN-FIN scan!	25	25	25
	oos	31	31	31
64.161.240.254	SYN-FIN scan!	41	41	44
	TCP SMTP Source Port	1	1	1
205.188.146.23	UNKNOWN	37	37	38
209.67.50.203	DNS udp DoS attack	d	3	3 16132
198.63.2.192	WinGate 1080 Attempt	35	35	52
198.139.244.22	WinGate 1080 Attempt	33	33	41
205.136.57.121	WinGate 1080 Attempt	32	32	44
199.173.178.2	WinGate 1080 Attempt	31	31	55
212.72.75.236	WinGate 1080 Attempt	27	27	65
204.117.70.5	WinGate 1080 Attempt	26	26	67
212.73.162.30	WinGate 1080 Attempt	26	26	40
24.3.0.36	UDP	17	17	8252
152.163.206.134	UDP	9	9	8514
63.78.39.192	Queso fingerprint	12	12	144
	oos	12	12	60
130.239.129.109	FULLXMAS	1	1	1
	INVALIDACK	1	1	2
	NULL	1	1	2
	UNKNOWN	1	3	3
	Probable NMAP finger	1	1	1
	Null scan!	1	1	3
	NOACK	1	10	14
	oos	1	4	9
132.163.4.103	UDP	22	22	23
132.163.4.102	UDP	20	20	21
62.136.71.93	Back Orifice	20	20	20
208.194.160.1	WinGate 1080 Attempt	19	19	22
132.163.4.101	UDP	19	19	21
64.242.77.150	Queso fingerprint	7	7	7
	UNKNOWN	4	7	38
	oos	3	3	4
194.75.152.237	WinGate 1080 Attempt	17	17	19
128.8.10.71	UDP	14	14	1332
192.168.105.1	UDP	15	15	355
192.168.105.3	UDP	15	15	86
212.179.8.164	Watchlist 000220 IL-	14	14	566
194.87.13.86	WinGate 1080 Attempt	15	15	51
10.10.10.1	UDP	15	15	38
169.254.92.200	UDP	15	15	24
24.3.0.37	UDP	4	4	5219
209.94.199.143	Back Orifice	14	14	14
128.8.10.68	UDP	13	13	441
212.179.95.5	Watchlist 000220 IL-	4	4	4563
169.254.75.93	UDP	13	13	21
148.228.125.215	External RPC call	13	13	13
63.253.110.142	INVALIDACK	1	2	3
	VECNA	1	1	1
	oos	1	1	1
	Null scan!	3	3	19
	NOACK	2	2	2
	NULL	3	3	19
192.216.128.28	WinGate 1080 Attempt	12	12	20
24.112.150.20	NOACK	1	2	2
	oos	1	7	8
	UNKNOWN	1	1	1
	Null scan!	1	1	16
24.141.240.197	WinGate 1080 Attempt	11	11	11
216.99.200.242	Attempted Sun RPC hi	1	1	4
	SUNRPC highport acce	1	1	6
	UDP	1	1	2619
	WinGate 1080 Attempt	1	1	1
	Back Orifice	1	1	3
63.119.91.2	NMAP TCP ping!	10	10	41
207.126.106.118	WinGate 1080 Attempt	10	10	21
130.234.187.194	FULLXMAS	1	2	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	INVALIDACK	1	2	2
	NOACK	1	2	3
	SPAU	1	1	2
	VECNA	1	1	1
	UNKNOWN	1	2	6
63.216.170.100	WinGate 1080 Attempt	10	10	11
24.1.91.62	FULLXMAS	1	1	1
	Null scan!	1	1	1
	oos	1	4	4
	Queso fingerprint	1	1	1
	NOACK	1	2	2
	NULL	1	1	1
212.179.45.241	Watchlist 000220 IL-	8	8	1002
207.172.3.10	UDP	4	4	2866
10.109.13.244	UDP	9	9	27
64.80.118.241	Queso fingerprint	6	6	10
	oos	3	3	7
202.187.24.3	NMAP TCP ping!	9	9	16
216.234.161.197	WinGate 1080 Attempt	9	9	13
206.205.246.10	Queso fingerprint	1	1	1
	UNKNOWN	5	7	9
	oos	1	1	1
63.254.40.8	FIN	1	1	1
	INVALIDACK	2	5	5
	UNKNOWN	1	1	1
	NOACK	1	2	2
141.30.228.36	Queso fingerprint	4	4	30
	oos	4	4	40
194.133.58.129	NMAP TCP ping!	8	8	46
204.192.85.123	Queso fingerprint	4	4	19
	oos	2	2	4
	UDP	2	2	18
141.30.228.175	Queso fingerprint	4	4	10
	oos	4	4	14
212.43.196.5	WinGate 1080 Attempt	8	8	23
216.104.228.102	NMAP TCP ping!	8	8	19
24.180.167.70	INVALIDACK	1	1	1
	NOACK	1	1	1
	Null scan!	1	1	4
	UNKNOWN	2	2	2
	oos	1	3	5
128.183.103.20	Queso fingerprint	4	4	4
	oos	4	4	6
194.84.208.118	WinGate 1080 Attempt	8	8	10
195.116.66.14	External RPC call	8	8	8
63.254.40.12	FIN	1	1	1
	INVALIDACK	1	2	2
	NOACK	1	2	2
	VECNA	1	2	2
	UNKNOWN	1	1	1
24.180.132.70	INVALIDACK	1	2	2
	oos	1	1	1
	VECNA	1	2	2
	NOACK	1	1	1
	UNKNOWN	1	2	2
151.23.31.22	UDP	7	7	363
151.23.31.23	UDP	7	7	308
208.11.27.144	UDP	7	7	99
141.30.228.115	Queso fingerprint	4	4	15
	oos	3	3	13
130.54.113.11	SMB Name Wildcard	7	7	16
216.179.0.32	WinGate 1080 Attempt	7	7	10
212.186.226.182	Null scan!	1	1	2
	oos	1	6	6
63.254.34.82	NOACK	1	2	3
	NULL	1	1	1
	XMAS	1	1	1
	VECNA	2	2	2
	oos	1	1	1
24.188.94.86	Null scan!	1	1	1
	UNKNOWN	1	1	1
	oos	1	4	4

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

24.200.166.157	NOACK	1	2	2
	oos	1	2	2
	INVALIDACK	1	3	3
24.200.53.22	UNKNOWN	1	2	2
	NOACK	1	1	1
	INVALIDACK	1	4	4
24.188.94.86	VECNA	1	1	1
159.226.121.37	Watchlist 000222 NET	5	5	900
212.179.77.20	Watchlist 000220 IL-	2	2	2353
216.119.15.88	connect to 515 from	4	4	1273
63.248.55.245	UDP	4	4	1195
151.23.31.21	UDP	6	6	137
24.180.134.156	UDP	1	1	1601
	WinGate 1080 Attempt	2	2	3
141.30.228.43	Queso fingerprint	3	3	41
	oos	3	3	31
141.30.228.199	Queso fingerprint	3	3	23
	oos	3	3	35
216.152.64.142	WinGate 1080 Attempt	6	6	30
204.42.254.5	Queso fingerprint	3	3	11
	oos	3	3	18
63.253.110.153	INVALIDACK	2	2	3
	VECNA	1	1	1
	NOACK	1	1	1
	NULL	1	1	11
	Null scan!	1	1	5
64.64.226.2	NMAP TCP ping!	6	6	19
159.182.21.254	NOACK	6	6	18
63.253.110.161	NOACK	2	2	2
	Null scan!	2	2	5
	NULL	2	2	5
63.252.94.85	NOACK	3	3	4
	oos	1	1	4
	NULL	1	1	1
	Null scan!	1	1	1
168.122.220.88	FULLXMAS	1	1	1
	Null scan!	1	1	2
	VECNA	1	1	1
	INVALIDACK	1	2	3
	NULL	1	1	2
141.39.198.102	INVALIDACK	1	1	1
	oos	1	2	2
	NULL	1	1	2
	SYN-FIN scan!	1	1	1
	Null scan!	1	1	2
63.253.89.201	NULL	3	3	4
	Null scan!	3	3	4
209.255.180.144	INVALIDACK	1	1	1
213.51.45.44	INVALIDACK	2	2	4
	Null scan!	1	1	1
	oos	1	1	1
	UNKNOWN	1	1	1
	NULL	1	1	1
209.255.180.144	VECNA	2	2	2
	Null scan!	1	1	1
	NOACK	2	2	4
63.252.98.155	NOACK	1	3	3
	UNKNOWN	2	2	2
	VECNA	1	1	2
198.88.88.99	WinGate 1080 Attempt	6	6	7
10.0.6.196	NOACK	1	1	1
	NULL	1	1	1
	XMAS	1	1	1
	oos	2	2	2
205.188.234.88	UNKNOWN	6	6	6
193.253.200.163	oos	1	2	2
	XMAS	1	1	1
	NOACK	1	2	2
	INVALIDACK	1	1	1
10.0.6.196	Null scan!	1	1	1
24.201.230.239	FULLXMAS	1	1	1
	oos	1	3	3

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	UNKNOWN	1	1	1
	NOACK	1	1	1
63.253.113.23	FIN	1	1	1
	NOACK	1	2	2
	NULL	1	1	1
	UNKNOWN	1	1	1
	VECNA	1	1	1
63.253.115.164	INVALIDACK	1	1	1
	NULL	1	1	1
	UNKNOWN	1	1	1
	Null scan!	1	1	1
64.196.112.33	oos	1	2	2
	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
	NMAPID	1	1	1
63.253.115.164	NOACK	1	1	1
	FIN	1	1	1
209.255.87.149	NOACK	1	2	2
	VECNA	1	1	1
	oos	2	2	2
	XMAS	1	1	1
159.226.228.1	Watchlist 000222 NET	5	5	85
205.188.153.100	Attempted Sun RPC hi	4	4	570
159.226.45.3	Watchlist 000222 NET	5	5	69
198.92.138.226	WinGate 1080 Attempt	5	5	37
134.2.214.47	Queso fingerprint	5	5	21
216.152.64.213	WinGate 1080 Attempt	5	5	20
64.84.40.12	WinGate 1080 Attempt	5	5	18
63.253.106.62	FIN	1	1	3
	NOACK	2	2	6
	XMAS	1	1	3
	VECNA	1	1	6
4.41.126.148	WinGate 1080 Attempt	5	5	16
130.203.237.41	SMB Name Wildcard	5	5	14
130.207.201.28	SMB Name Wildcard	5	5	13
63.253.98.169	NOACK	1	1	1
	NULL	2	2	6
	Null scan!	2	2	6
63.253.110.154	INVALIDACK	1	2	6
	NULL	2	2	4
	Null scan!	1	1	1
130.60.57.66	SMB Name Wildcard	5	5	11
130.160.179.100	SMB Name Wildcard	5	5	10
208.180.106.106	WinGate 1080 Attempt	5	5	10
130.118.152.211	SMB Name Wildcard	5	5	9
130.91.206.91	SMB Name Wildcard	5	5	9
130.237.217.108	SMB Name Wildcard	5	5	9
24.8.137.12	NOACK	1	1	3
	NULL	1	1	1
63.253.104.165	NULL	2	2	4
	INVALIDACK	1	1	2
24.8.137.12	oos	1	2	4
	Null scan!	1	1	1
63.253.104.165	Null scan!	2	2	3
209.19.221.72	UNKNOWN	2	2	6
	oos	1	3	3
216.104.228.134	NMAP TCP ping!	5	5	8
24.64.156.201	FIN	1	1	1
	Null scan!	1	1	2
	NULL	1	1	2
	oos	1	2	3
24.200.184.215	FIN	1	1	3
	INVALIDACK	1	1	1
24.201.230.144	INVALIDACK	1	1	1
	Null scan!	1	1	2
63.253.112.234	Null scan!	1	1	2
	NULL	1	1	2
	NOACK	1	1	1
	INVALIDACK	1	1	1
24.69.232.101	oos	1	3	4
	INVALIDACK	1	2	3

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

24.201.230.144	oos	1	2	2
	NULL	1	1	2
24.200.184.215	NOACK	1	1	1
	oos	1	1	1
	UNKNOWN	1	1	1
63.253.112.234	XMAS	1	1	1
63.254.34.32	INVALIDACK	1	1	2
	NOACK	1	1	1
65.32.103.75	UNKNOWN	1	1	3
	NULL	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	2	2
64.196.112.129	oos	1	1	1
	XMAS	1	1	1
	VECNA	1	1	3
	NOACK	1	2	2
63.254.34.32	Null scan!	1	1	1
	NULL	2	2	3
209.212.128.41	WinGate 1080 Attempt	5	5	7
24.169.52.151	NOACK	1	1	1
	Null scan!	1	1	2
	VECNA	1	1	1
	NULL	1	1	2
	NMAPID	1	1	1
24.15.110.94	Queso fingerprint	2	2	3
	oos	3	3	4
130.207.158.225	SMB Name Wildcard	5	5	7
130.64.122.117	SMB Name Wildcard	5	5	7
129.2.246.25	FIN	1	1	1
	NOACK	1	1	1
	INVALIDACK	2	2	2
	oos	1	1	2
63.252.94.131	INVALIDACK	1	1	1
	XMAS	1	1	1
63.253.104.175	NOACK	1	1	1
	NULL	1	1	1
	VECNA	1	1	1
	Null scan!	2	2	3
63.252.94.131	VECNA	1	1	2
	NOACK	1	2	2
24.183.97.25	INVALIDACK	1	2	2
63.252.96.33	INVALIDACK	1	1	1
63.252.92.45	XMAS	1	1	1
	VECNA	1	1	1
	UNKNOWN	1	1	1
	NOACK	1	2	2
63.252.118.166	NOACK	1	1	1
	INVALIDACK	1	3	3
	FIN	1	1	1
24.183.97.25	NOACK	1	1	1
63.253.112.126	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
63.253.112.104	oos	1	1	1
	VECNA	1	1	1
	UNKNOWN	1	1	1
	NOACK	1	2	2
63.252.96.33	oos	1	3	3
	UNKNOWN	1	1	1
63.253.112.158	oos	1	1	1
	XMAS	1	1	1
	VECNA	1	1	1
	NOACK	1	2	2
63.253.112.126	oos	1	1	1
	VECNA	1	1	1
24.68.66.177	FIN	1	1	1
	oos	1	2	2
4.41.142.66	oos	1	2	2
	INVALIDACK	1	3	3
24.68.66.177	INVALIDACK	1	2	2
24.183.97.25	oos	1	2	2
63.253.112.47	SYN-FIN scan!	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	oos	3	4	4
209.156.49.155	INVALIDACK	1	1	1
	NOACK	1	2	2
	NULL	1	1	1
	Null scan!	1	1	1
209.255.208.61	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	2	2
209.255.180.151	oos	1	1	1
	VECNA	1	1	1
	UNKNOWN	1	1	1
	NOACK	1	2	2
24.132.169.189	WinGate 1080 Attempt	5	5	5
209.19.221.98	Null scan!	2	2	2
	UNKNOWN	2	2	2
	NULL	1	1	1
144.118.195.243	FIN	1	1	1
	Null scan!	1	1	1
	oos	1	1	1
161.58.8.77	WinGate 1080 Attempt	5	5	5
144.118.195.243	SPAU	1	1	1
	INVALIDACK	1	1	1
205.188.153.108	Attempted Sun RPC hi	4	4	493
212.179.56.5	Watchlist 000220 IL-	3	3	926
205.188.153.106	Attempted Sun RPC hi	4	4	398
212.179.42.102	Watchlist 000220 IL-	2	2	1387
212.179.27.6	Watchlist 000220 IL-	4	4	202
205.188.153.104	Attempted Sun RPC hi	4	4	154
159.226.114.1	Watchlist 000222 NET	4	4	98
144.51.17.1	UDP	4	4	97
205.188.153.105	Attempted Sun RPC hi	4	4	63
216.152.64.211	WinGate 1080 Attempt	4	4	27
212.179.37.92	Watchlist 000220 IL-	4	4	25
216.95.146.101	WinGate 1080 Attempt	4	4	23
141.30.228.178	Queso fingerprint	2	2	8
	oos	2	2	15
63.253.98.172	NULL	2	2	11
	Null scan!	2	2	11
62.4.171.156	WinGate 1080 Attempt	4	4	21
24.180.168.160	NOACK	1	1	1
	XMAS	1	1	6
	Null scan!	1	1	6
	NULL	1	1	4
212.179.44.105	Watchlist 000220 IL-	1	1	1517
216.78.155.50	WinGate 1080 Attempt	4	4	13
62.142.208.64	WinGate 1080 Attempt	4	4	13
63.253.85.90	FIN	1	1	1
	NULL	1	1	7
	NOACK	1	2	5
64.196.112.22	NULL	1	1	1
	Null scan!	1	1	1
	UNKNOWN	2	2	10
62.29.16.82	INVALIDACK	1	1	2
	oos	2	2	7
	NOACK	1	1	2
130.153.158.82	SMB Name Wildcard	4	4	11
130.54.36.42	SMB Name Wildcard	4	4	10
206.210.80.6	External RPC call	3	3	8
	STATDX UDP attack	1	1	1
130.49.220.28	SMB Name Wildcard	4	4	8
65.1.215.157	NULL	1	1	3
	Null scan!	1	1	3
	oos	1	1	1
	UNKNOWN	1	1	1
209.252.32.2	INVALIDACK	1	1	2
212.17.114.164	UDP	4	4	8
209.252.32.2	NMAPID	2	2	4
	NOACK	1	1	2
209.156.50.147	NOACK	1	2	3
	VECNA	1	1	2
216.94.134.20	WinGate 1080 Attempt	4	4	7

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

209.156.50.147	oos	1	1	2
63.255.40.130	NOACK	2	2	3
	oos	2	2	4
12.21.190.9	NMAP TCP ping!	4	4	7
195.130.141.88	Null scan!	1	1	1
	NULL	1	1	2
	NOACK	1	2	4
130.113.246.175	SMB Name Wildcard	4	4	7
199.197.135.21	NMAP TCP ping!	4	4	6
206.205.246.2	Queso fingerprint	2	2	2
	Probable NMAP finger	1	1	1
	NMAP TCP ping!	1	1	3
63.254.40.53	FIN	1	1	3
64.230.168.176	WinGate 1080 Attempt	4	4	6
63.254.40.53	NOACK	2	2	2
	oos	1	1	1
24.71.80.73	NOACK	1	1	3
63.253.110.141	Null scan!	1	1	1
	NULL	1	1	2
	NOACK	1	1	1
	INVALIDACK	1	1	2
24.71.80.73	NULL	1	1	1
63.252.96.12	NULL	2	2	4
	Null scan!	2	2	2
63.253.105.252	Null scan!	2	2	4
	NULL	2	2	2
24.71.80.73	VECNA	1	1	1
	Null scan!	1	1	1
209.255.181.157	NOACK	1	2	2
209.255.27.64	XMAS	1	1	2
	VECNA	1	1	2
209.255.181.157	VECNA	1	1	2
	XMAS	1	1	2
209.255.27.64	NOACK	1	1	1
	INVALIDACK	1	1	1
24.13.223.139	oos	1	4	5
38.144.87.21	UNKNOWN	2	2	2
63.253.106.1	Null scan!	1	1	1
	NULL	2	2	3
	NOACK	1	1	1
61.159.96.161	oos	1	1	1
	XMAS	1	1	2
38.144.87.21	oos	2	2	3
61.159.96.161	VECNA	1	1	1
	NOACK	1	1	1
63.253.113.87	SYN-FIN scan!	1	1	1
63.254.34.61	Null scan!	1	1	1
	NULL	2	2	2
	NOACK	1	1	2
63.253.140.177	oos	1	3	4
	SYN-FIN scan!	1	1	1
63.253.113.87	oos	1	3	4
63.254.213.167	NOACK	2	2	2
64.196.88.37	NOACK	1	2	2
	XMAS	1	1	1
	VECNA	1	1	1
63.254.213.167	oos	2	2	2
63.254.40.64	FIN	1	1	1
	INVALIDACK	2	2	2
	VECNA	1	1	1
24.200.91.109	INVALIDACK	1	1	1
24.22.254.40	NMAPID	1	1	1
24.23.37.248	INVALIDACK	1	2	2
	oos	1	1	1
24.28.13.63	NOACK	1	1	1
63.252.39.130	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
63.252.141.237	oos	1	1	1
	VECNA	1	1	1
	UNKNOWN	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	NOACK	1	1	1
24.94.34.232	UNKNOWN	1	1	1
63.252.96.136	oos	1	1	1
	XMAS	1	1	1
	NOACK	1	2	2
63.252.96.103	oos	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
63.252.94.152	XMAS	1	1	1
	VECNA	1	1	1
	NOACK	1	2	2
63.252.94.123	oos	1	1	1
	VECNA	1	1	1
	NOACK	1	2	2
63.252.92.209	VECNA	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
63.253.112.161	XMAS	1	1	1
	VECNA	1	1	1
	Null scan!	1	1	1
	NOACK	1	1	1
63.253.106.21	Null scan!	2	2	2
	NULL	2	2	2
63.253.106.18	oos	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
63.253.104.216	VECNA	1	1	1
	UNKNOWN	1	1	1
	NOACK	1	2	2
63.253.104.187	Null scan!	2	2	2
	NULL	2	2	2
24.94.34.232	Null scan!	1	1	1
	NULL	1	2	2
24.8.33.214	oos	1	2	2
	NOACK	1	1	1
	INVALIDACK	1	1	1
24.28.13.63	VECNA	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
24.23.37.248	UNKNOWN	1	1	1
24.22.254.40	oos	1	2	2
	FULLXMAS	1	1	1
24.200.91.109	NOACK	1	2	2
	Null scan!	1	1	1
209.168.7.24	INVALIDACK	1	1	1
	oos	1	1	1
212.182.16.101	oos	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
209.255.214.63	oos	1	1	1
	UNKNOWN	1	1	1
	SYN-FIN scan!	1	1	1
	INVALIDACK	1	1	1
209.255.180.162	UNKNOWN	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
209.168.7.24	NULL	1	1	1
216.51.102.134	oos	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
209.85.227.188	UNKNOWN	4	4	4
209.255.79.92	Null scan!	2	2	2
	NULL	2	2	2
209.168.7.24	Null scan!	1	1	1
131.173.58.195	FIN	1	1	1
	oos	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

195.159.0.151	UNKNOWN	1	1	1
195.6.101.49	INVALIDACK	1	1	1
202.188.179.33	oos	1	1	1
	NOACK	1	2	2
	INVALIDACK	1	1	1
195.6.101.49	oos	1	1	1
	NOACK	1	2	2
195.159.0.151	WinGate 1080 Attempt	3	3	3
131.173.58.195	NOACK	2	2	2
61.134.9.133	Tiny Fragments - Pos	3	3	458
192.232.16.86	UDP	3	3	428
210.12.160.130	Tiny Fragments - Pos	3	3	385
202.108.43.152	Tiny Fragments - Pos	3	3	326
202.108.43.151	Tiny Fragments - Pos	3	3	286
65.4.87.43	Tiny Fragments - Pos	2	2	733
212.179.38.135	Watchlist 000220 IL-	1	1	1221
128.46.156.231	SNMP public access	3	3	161
64.12.104.134	UDP	2	2	646
212.179.58.4	Watchlist 000220 IL-	3	3	60
141.30.228.161	Queso fingerprint	2	2	49
192.102.197.234	NMAP TCP ping!	3	3	55
141.30.228.161	oos	1	1	6
212.179.58.12	Watchlist 000220 IL-	1	1	1054
159.226.159.50	Watchlist 000222 NET	3	3	43
212.179.63.10	Watchlist 000220 IL-	3	3	41
159.226.47.217	Watchlist 000222 NET	3	3	24
141.30.228.182	Queso fingerprint	1	1	9
	oos	2	2	12
194.47.144.22	WinGate 1080 Attempt	3	3	20
63.252.96.36	INVALIDACK	1	1	1
	NULL	1	1	8
	Null scan!	1	1	8
213.61.112.10	WinGate 1080 Attempt	3	3	16
63.253.110.147	NULL	2	2	8
	Null scan!	1	1	7
24.155.12.96	INVALIDACK	1	1	3
	NOACK	1	2	9
209.19.221.76	NULL	1	1	3
212.179.21.74	Watchlist 000220 IL-	3	3	10
209.19.221.76	UNKNOWN	1	1	6
	Null scan!	1	1	1
63.252.92.108	NULL	1	1	4
	Null scan!	2	2	6
24.234.44.163	WinGate 1080 Attempt	3	3	9
63.252.92.165	Null scan!	1	1	4
	oos	1	1	1
	NULL	1	1	4
63.253.112.42	NULL	1	1	4
	Null scan!	2	2	5
209.249.181.211	WinGate 1080 Attempt	3	3	9
212.208.74.129	NMAP TCP ping!	3	3	8
63.253.102.4	NOACK	1	2	6
	UNKNOWN	1	1	2
130.39.126.168	SMB Name Wildcard	3	3	8
200.51.51.201	WinGate 1080 Attempt	3	3	8
130.251.107.6	SMB Name Wildcard	3	3	7
141.30.228.221	oos	1	1	2
208.47.228.120	WinGate 1080 Attempt	3	3	7
130.251.67.20	SMB Name Wildcard	3	3	7
141.30.228.221	Queso fingerprint	2	2	5
24.2.14.32	NULL	1	2	6
	Null scan!	1	1	1
63.253.110.146	Null scan!	1	1	3
	NULL	2	2	4
63.253.106.28	XMAS	1	1	3
	VECNA	1	1	3
	Null scan!	1	1	1
63.252.119.35	Null scan!	1	1	1
	NULL	1	1	3
	INVALIDACK	1	1	3
63.253.98.168	NULL	1	1	3
63.254.34.25	NULL	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	NOACK	2	2	6
63.253.98.168	VECNA	1	1	1
	Null scan!	1	1	3
209.19.221.105	NULL	1	1	2
	Null scan!	1	1	2
	UNKNOWN	1	1	3
24.115.76.63	WinGate 1080 Attempt	3	3	7
216.10.12.30	XMAS	1	1	2
	SUNRPC highport acce	2	2	5
212.46.76.30	WinGate 1080 Attempt	3	3	7
212.69.230.242	Queso fingerprint	1	1	2
	oos	2	2	4
63.254.34.56	NOACK	1	1	2
	NULL	1	1	2
	Null scan!	1	1	2
64.196.112.38	VECNA	1	1	2
	XMAS	1	1	2
	UNKNOWN	1	1	2
62.46.74.14	NOACK	1	2	4
	UNKNOWN	1	1	2
63.252.94.71	NULL	1	1	1
	NOACK	1	2	5
63.253.110.148	Null scan!	1	1	2
	NULL	1	1	2
	INVALIDACK	1	1	2
130.127.216.206	SMB Name Wildcard	3	3	6
159.226.66.130	Watchlist 000222 NET	3	3	6
144.118.212.111	VECNA	1	2	5
	NOACK	1	1	1
130.127.96.52	SMB Name Wildcard	3	3	5
208.185.235.100	External RPC call	3	3	5
130.225.149.103	SMB Name Wildcard	3	3	5
199.36.49.2	NMAP TCP ping!	3	3	5
24.3.161.193	Queso fingerprint	2	2	4
	oos	1	1	1
24.3.52.236	NOACK	1	1	1
24.43.244.147	NOACK	1	1	1
	FIN	1	1	2
24.3.52.236	Null scan!	1	1	3
	INVALIDACK	1	1	1
24.43.244.147	oos	1	1	2
63.252.92.239	SYN-FIN scan!	1	1	2
63.210.46.242	Tiny Fragments - Pos	3	3	5
62.31.29.232	oos	1	1	1
	Null scan!	1	1	2
	NULL	1	1	2
62.31.29.117	oos	1	1	1
	Null scan!	1	1	2
	NULL	1	1	2
62.31.28.162	Null scan!	1	1	2
63.253.104.199	Null scan!	1	1	2
	NULL	1	1	2
	NOACK	1	1	1
63.253.104.180	SPAU	1	1	1
	INVALIDACK	1	2	4
63.252.94.122	oos	1	1	3
	NOACK	1	1	1
	INVALIDACK	1	1	1
63.252.92.239	oos	1	2	3
63.253.105.254	Null scan!	1	1	2
	NULL	1	1	2
	FIN	1	1	1
62.31.28.162	NULL	1	1	2
	INVALIDACK	1	1	1
62.29.17.96	NOACK	1	2	4
	INVALIDACK	1	1	1
62.29.16.49	VECNA	1	1	1
	NOACK	1	1	3
	INVALIDACK	1	1	1
63.254.34.37	NOACK	1	1	2
	Null scan!	1	1	2
64.59.147.243	INVALIDACK	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	Null scan!	1	1	2
	NULL	1	1	2
63.254.34.37	NULL	1	1	1
209.19.221.110	Null scan!	1	1	1
24.10.163.21	FIN	1	1	1
	oos	1	1	2
	NOACK	1	1	2
216.232.33.71	Null scan!	1	1	1
209.19.221.110	UNKNOWN	2	2	4
216.152.64.151	WinGate 1080 Attempt	3	3	5
216.232.33.71	NULL	1	1	2
	INVALIDACK	1	1	2
209.180.140.17	WinGate 1080 Attempt	3	3	4
213.76.100.7	NULL	1	1	1
	Null scan!	1	1	1
	INVALIDACK	1	1	2
212.25.67.88	WinGate 1080 Attempt	3	3	4
63.253.115.152	NULL	1	1	1
	Null scan!	2	2	3
63.254.34.125	Null scan!	1	1	1
63.254.34.150	INVALIDACK	1	1	1
64.196.23.118	UNKNOWN	1	1	1
	SYN-FIN scan!	1	1	1
	NOACK	1	1	2
64.196.112.8	UNKNOWN	1	1	2
	NOACK	1	1	1
	INVALIDACK	1	1	1
63.254.34.150	oos	1	1	2
	NULL	1	1	1
63.254.34.125	UNKNOWN	1	1	2
	NULL	1	1	1
24.184.192.56	NOACK	1	1	2
63.253.106.25	NULL	1	1	1
63.253.104.169	VECNA	1	1	2
	UNKNOWN	1	1	1
	INVALIDACK	1	1	1
63.252.92.35	XMAS	1	1	1
	VECNA	1	1	2
	NOACK	1	1	1
63.252.118.204	VECNA	1	1	1
	NOACK	1	2	3
63.253.106.25	oos	1	1	1
	Null scan!	1	1	2
24.184.192.56	Null scan!	1	1	1
24.7.191.55	WinGate 1080 Attempt	3	3	4
24.184.192.56	NULL	1	1	1
129.2.192.79	oos	1	3	4
2.2.2.2	NMAP TCP ping!	3	3	4
204.254.126.101	oos	1	1	1
	UNKNOWN	1	1	2
	NOACK	1	1	1
204.155.48.3	NMAP TCP ping!	3	3	4
203.106.185.97	oos	2	2	3
	NOACK	1	1	1
202.84.134.141	External RPC call	3	3	4
169.229.55.14	oos	1	1	1
	UNKNOWN	1	1	2
129.242.219.27	Queso fingerprint	3	3	4
130.60.152.222	SMB Name Wildcard	3	3	4
169.229.55.14	INVALIDACK	1	1	1
130.206.43.173	SMB Name Wildcard	3	3	4
128.175.109.212	NULL	1	1	1
	UNKNOWN	1	1	1
195.159.99.65	UNKNOWN	3	3	3
195.113.26.140	oos	1	1	1
	SPAU	1	1	1
	INVALIDACK	1	1	1
194.47.253.201	VECNA	1	2	2
	UNKNOWN	1	1	1
194.30.132.18	NMAP TCP ping!	3	3	3
194.183.71.4	INVALIDACK	3	3	3
193.136.145.10	INVALIDACK	1	3	3

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

193.10.104.219	oos	1	1	1
	UNKNOWN	1	1	1
	INVALIDACK	1	1	1
167.206.68.184	oos	1	2	2
	NOACK	1	1	1
165.166.177.10	WinGate 1080 Attempt	3	3	3
145.98.39.190	VECNA	1	1	1
	NOACK	1	2	2
141.30.228.134	oos	2	2	2
	Queso fingerprint	1	1	1
141.213.219.132	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
136.160.130.100	WinGate 1080 Attempt	3	3	3
129.242.219.40	UNKNOWN	3	3	3
128.175.109.212	Null scan!	1	1	1
208.61.45.79	WinGate 1080 Attempt	3	3	3
207.86.108.130	Queso fingerprint	3	3	3
207.162.100.63	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
205.159.243.120	UDP	3	3	3
202.77.194.202	Null scan!	2	2	2
	NULL	1	1	1
198.182.76.100	WinGate 1080 Attempt	3	3	3
193.216.190.18	VECNA	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
129.188.136.100	UDP	3	3	3
24.200.94.202	INVALIDACK	1	1	1
	NULL	1	1	1
24.201.16.170	INVALIDACK	1	1	1
	oos	1	1	1
24.201.254.188	UNKNOWN	1	1	1
24.68.86.167	NOACK	1	1	1
	INVALIDACK	1	1	1
24.67.27.92	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
24.42.150.116	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
24.226.210.118	NULL	1	1	1
62.161.76.55	UNKNOWN	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
62.137.166.234	oos	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
62.137.149.87	oos	1	1	1
	NOACK	1	2	2
24.68.86.167	UNKNOWN	1	1	1
63.252.135.53	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
63.252.119.100	oos	1	2	2
	SYN-FIN scan!	1	1	1
63.252.118.230	oos	2	3	3
63.252.118.186	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
24.226.210.118	INVALIDACK	1	1	1
24.222.110.58	UNKNOWN	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
24.217.79.34	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
24.201.254.188	oos	1	1	1
63.253.112.106	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

63.252.96.125	oos	1	1	1
	NOACK	1	2	2
63.252.95.69	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
63.252.95.226	Null scan!	1	1	1
	NULL	1	1	1
	FIN	1	1	1
63.252.94.215	Null scan!	2	2	2
	NULL	1	1	1
63.252.92.193	Null scan!	2	2	2
	NULL	1	1	1
24.23.140.205	oos	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
24.226.210.118	Null scan!	1	1	1
24.201.254.188	NOACK	1	1	1
24.201.16.170	NOACK	1	1	1
24.200.94.202	Null scan!	1	1	1
63.253.112.90	Null scan!	1	1	1
63.253.113.119	NOACK	1	1	1
	oos	1	1	1
	UNKNOWN	1	1	1
63.254.40.61	UNKNOWN	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
63.254.40.4	VECNA	1	2	2
	INVALIDACK	1	1	1
63.254.40.14	NOACK	1	1	1
	INVALIDACK	2	2	2
63.253.136.29	XMAS	1	1	1
	VECNA	1	1	1
	UNKNOWN	1	1	1
63.253.115.154	UNKNOWN	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
63.253.115.105	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
63.253.142.73	Null scan!	1	1	1
	NULL	1	1	1
	FIN	1	1	1
63.253.140.43	UNKNOWN	1	1	1
	INVALIDACK	1	2	2
63.253.114.8	oos	1	1	1
	VECNA	1	1	1
	NOACK	1	1	1
63.253.112.90	oos	1	2	2
63.254.34.21	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
63.253.143.107	VECNA	1	1	1
	SYN-FIN scan!	1	1	1
	INVALIDACK	1	1	1
63.253.142.89	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
209.156.49.130	INVALIDACK	1	2	2
	NOACK	1	1	1
209.156.50.12	NOACK	1	1	1
	oos	1	1	1
	UNKNOWN	1	1	1
24.179.118.54	NOACK	1	1	1
	INVALIDACK	1	2	2
24.165.165.97	Null scan!	1	1	1
	NULL	1	1	1
	NOACK	1	1	1
24.13.101.55	oos	1	2	2
	VECNA	1	1	1
24.114.193.8	Null scan!	1	1	1
	NULL	1	1	1
213.76.100.14	oos	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	Null scan!	1	1	1
	NULL	1	1	1
209.255.88.70	VECNA	1	1	1
	UNKNOWN	1	1	1
	NOACK	1	1	1
209.255.52.9	UNKNOWN	1	1	1
	INVALIDACK	1	2	2
209.255.214.227	VECNA	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
209.255.213.73	UNKNOWN	1	1	1
	INVALIDACK	1	2	2
209.255.181.90	oos	1	1	1
	UNKNOWN	1	1	1
	NOACK	1	1	1
24.114.193.8	FIN	1	1	1
24.113.9.236	oos	1	1	1
	VECNA	1	1	1
	INVALIDACK	1	1	1
24.112.199.246	Null scan!	1	1	1
	NULL	1	1	1
	INVALIDACK	1	1	1
217.80.175.18	UNKNOWN	1	2	2
	NULL	1	1	1
212.181.196.52	oos	1	1	1
	XMAS	1	1	1
	NOACK	1	1	1
209.255.180.201	VECNA	1	1	1
	Null scan!	1	1	1
	NULL	1	1	1
209.255.180.133	VECNA	1	1	1
	NOACK	1	1	1
	INVALIDACK	1	1	1
207.217.77.82	UDP	2	2	463
202.101.43.222	Tiny Fragments - Pos	2	2	460
212.179.7.173	Watchlist 000220 IL-	2	2	453
61.140.75.3	Tiny Fragments - Pos	2	2	415
202.96.96.3	Tiny Fragments - Pos	2	2	391
61.140.75.5	Tiny Fragments - Pos	2	2	285
209.244.0.3	UDP	2	2	245
202.101.43.220	Tiny Fragments - Pos	2	2	243
61.140.75.4	Tiny Fragments - Pos	2	2	236
61.155.13.3	Tiny Fragments - Pos	2	2	199
202.101.43.223	Tiny Fragments - Pos	2	2	171
207.172.3.11	UDP	2	2	161
159.226.115.1	Watchlist 000222 NET	2	2	158
205.188.153.101	Attempted Sun RPC hi	2	2	150
213.221.174.134	UDP	2	2	123
194.177.103.35	UDP	2	2	117
159.226.39.4	Watchlist 000222 NET	2	2	97
205.188.153.102	Attempted Sun RPC hi	2	2	73
213.228.136.250	UDP	2	2	64
205.188.153.111	Attempted Sun RPC hi	2	2	59
212.224.28.210	UDP	2	2	58
213.140.4.75	UDP	2	2	53
212.179.16.107	Watchlist 000220 IL-	2	2	44
63.124.243.34	oos	2	2	44
212.224.24.226	UDP	2	2	41
212.179.7.161	Watchlist 000220 IL-	2	2	37
212.179.7.36	Watchlist 000220 IL-	2	2	35
61.134.9.134	Tiny Fragments - Pos	2	2	32
159.226.91.20	Watchlist 000222 NET	1	1	528
202.205.5.10	Tiny Fragments - Pos	1	1	521
63.253.98.171	NULL	1	1	10
	Null scan!	1	1	10
128.46.156.117	Queso fingerprint	1	1	8
	oos	1	1	8
159.226.61.62	Watchlist 000222 NET	2	2	15
63.253.110.157	NULL	1	1	6
	Null scan!	1	1	9
209.145.62.160	WinGate 1080 Attempt	2	2	14
209.156.50.45	NULL	1	1	6

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	Null scan!	1	1	6
63.253.140.94	NULL	1	1	6
	Null scan!	1	1	6
141.30.228.58	Queso fingerprint	1	1	5
	oos	1	1	7
203.134.52.163	WinGate 1080 Attempt	2	2	12
159.226.47.195	Watchlist 000222 NET	2	2	11
212.30.95.177	VECNA	1	2	11
62.31.29.134	NULL	1	1	5
	Null scan!	1	1	5
63.229.136.128	WinGate 1080 Attempt	2	2	8
63.252.119.23	NULL	1	1	4
	Null scan!	1	1	4
63.253.143.46	NULL	1	1	4
	Null scan!	1	1	4
63.253.7.223	Null scan!	1	1	2
63.254.72.89	NOACK	2	2	8
63.253.7.223	NULL	1	1	6
209.247.204.244	Tiny Fragments - Pos	2	2	8
24.156.144.237	WinGate 1080 Attempt	2	2	8
212.217.124.157	Back Orifice	1	1	1
	WinGate 1080 Attempt	1	1	7
128.61.58.60	INVALIDACK	1	1	4
	oos	1	1	4
130.226.14.50	SMB Name Wildcard	2	2	8
130.39.216.104	SMB Name Wildcard	2	2	8
130.251.101.96	SMB Name Wildcard	2	2	7
66.44.40.54	WinGate 1080 Attempt	2	2	7
61.139.110.69	SUNRPC highport acce	1	1	1
	WinGate 1080 Attempt	1	1	6
63.252.94.128	NOACK	2	2	7
63.253.110.155	Null scan!	1	1	4
	NULL	1	1	3
63.252.94.96	oos	1	2	7
63.252.119.77	NULL	1	1	3
63.252.93.128	NULL	1	1	3
	Null scan!	1	1	3
63.253.106.23	NULL	1	1	4
63.253.110.156	NULL	1	1	4
	Null scan!	1	1	2
63.253.106.38	NULL	1	1	4
	Null scan!	1	1	2
63.253.106.23	Null scan!	1	1	2
63.252.119.77	Null scan!	1	1	3
63.253.140.56	NULL	1	1	3
63.253.98.170	NULL	1	1	3
63.255.12.157	Null scan!	1	1	2
66.26.38.170	WinGate 1080 Attempt	2	2	6
63.255.12.157	NULL	1	1	4
63.253.98.170	Null scan!	1	1	3
63.253.140.56	Null scan!	1	1	3
63.253.164.236	oos	2	2	6
209.19.221.73	UNKNOWN	2	2	6
24.11.102.209	NOACK	1	1	3
	INVALIDACK	1	1	3
209.255.181.161	oos	1	1	5
	Null scan!	1	1	1
212.64.115.51	WinGate 1080 Attempt	2	2	6
24.1.57.202	NOACK	1	2	6
140.247.155.66	Queso fingerprint	1	1	3
	oos	1	1	3
156.17.208.4	FIN	1	1	3
	INVALIDACK	1	1	3
207.106.84.176	WinGate 1080 Attempt	2	2	6
207.96.19.36	oos	2	2	6
199.197.130.21	NMAP TCP ping!	2	2	6
130.127.120.116	SMB Name Wildcard	2	2	5
207.137.11.119	WinGate 1080 Attempt	2	2	5
130.212.20.72	External RPC call	2	2	5
208.185.24.251	WinGate 1080 Attempt	2	2	5
209.19.221.74	UNKNOWN	2	2	5
24.130.149.232	Queso fingerprint	1	1	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	oos	1	1	3
209.19.221.92	UNKNOWN	2	2	5
24.189.31.228	SMB Name Wildcard	1	1	4
24.200.70.242	INVALIDACK	2	2	5
24.9.145.183	WinGate 1080 Attempt	2	2	5
24.189.31.228	SUNRPC highport acce	1	1	1
63.252.92.114	NOACK	1	2	5
62.4.171.118	WinGate 1080 Attempt	2	2	5
32.101.117.171	WinGate 1080 Attempt	2	2	4
62.59.1.31	Null scan!	1	1	2
	NULL	1	1	2

Table -Targets of Attack

DST	TYPE	DSRC	CNT	TOTAL
MY.NET.60.11	INVALIDACK	2	2	2
	NOACK	1	1	1
	NULL	69	69	103
	Null scan!	96	96	131
	UNKNOWN	1	1	1
	WinGate 1080 Attempt	35	35	96
	oos	4	4	6
	XMAS	1	1	1
	Watchlist 000220 IL-	1	1	1054
	Tiny Fragments - Pos	2	2	168
	SUNRPC highport acce	1	1	2
	Queso fingerprint	3	3	4
MY.NET.253.114	INVALIDACK	10	10	11
	NMAP TCP ping!	5	5	10
	NMAPID	1	1	1
	NOACK	47	66	77
	UNKNOWN	7	7	7
	Watchlist 000222 NET	1	1	1
	oos	37	38	43
	XMAS	26	26	38
	VECNA	45	45	70
	SMB Name Wildcard	1	1	1
	Null scan!	5	5	5
	NULL	3	3	4
MY.NET.253.112	FIN	8	8	12
	INVALIDACK	31	35	49
	NMAPID	1	1	2
	NULL	27	27	35
	UNKNOWN	8	8	8
	SPAU	1	1	1
	Null scan!	6	6	13
	NOACK	33	36	57
	VECNA	10	10	11
	oos	25	36	45
	XMAS	3	3	6
	Watchlist 000220 IL-	1	1	262
MY.NET.60.8	Back Orifice	1	1	2
	FIN	1	1	1
	INVALIDACK	4	4	4
	Null scan!	67	67	94
	NULL	46	46	69
	NOACK	1	1	1
	Queso fingerprint	1	1	10
	oos	5	7	9
	WinGate 1080 Attempt	43	43	134
MY.NET.5.29	FIN	9	9	11
	INVALIDACK	22	29	34
	NOACK	12	15	16
	NULL	16	16	20
	NMAPID	2	2	3
	Null scan!	23	23	25
	VECNA	3	5	5
	Watchlist 000222 NET	1	1	275
	oos	23	29	29

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	Watchlist 000220 IL-	1	1	1125
	UNKNOWN	10	10	12
MY.NET.6.39	INVALIDACK	1	1	1
	NOACK	28	28	32
	NULL	20	20	58
	VECNA	1	1	1
	UNKNOWN	37	37	81
	Null scan!	25	25	62
	oos	1	1	1
MY.NET.60.38	NOACK	2	2	2
	NULL	28	28	34
	Null scan!	36	36	50
	XMAS	1	1	6
	oos	4	4	5
	WinGate 1080 Attempt	34	34	95
	Queso fingerprint	2	2	6
MY.NET.60.16	Back Orifice	1	1	1
	WinGate 1080 Attempt	23	23	47
	oos	1	1	1
	UNKNOWN	1	1	1
	INVALIDACK	1	1	1
	Null scan!	37	37	86
	Queso fingerprint	2	2	3
	UDP	1	1	1
	NULL	26	26	62
MY.NET.6.44	INVALIDACK	1	2	3
	Null scan!	21	21	61
	oos	3	3	3
	connect to 515 from	1	1	1
	VECNA	2	2	2
	UNKNOWN	27	27	54
	NULL	19	19	61
	NOACK	17	17	18
MY.NET.253.125	INVALIDACK	1	1	1
	NMAP TCP ping!	16	16	27
	NULL	1	1	1
	NOACK	28	29	49
	Null scan!	3	3	3
	oos	35	35	52
	connect to 515 from	1	1	1
	VECNA	1	1	1
	UNKNOWN	2	2	2
MY.NET.201.222	NULL	1	1	1
	Null scan!	2	2	2
	Probable NMAP finger	1	1	1
	Watchlist 000220 IL-	1	1	37604
	oos	3	3	3
MY.NET.100.165	FIN	1	1	1
	INVALIDACK	26	27	35
	NOACK	11	12	15
	NMAP TCP ping!	13	13	27
	NULL	2	2	2
	SMB Name Wildcard	1	1	7
	VECNA	1	1	1
	oos	20	20	48
	connect to 515 from	1	1	1
	UNKNOWN	1	1	1
	Null scan!	3	3	3
MY.NET.217.146	FIN	1	2	2
	FULLXMAS	1	2	2
	INVALIDACK	1	20	27
	NULL	1	1	2
	NOACK	1	21	26
	NMAPID	1	2	2
	Null scan!	1	1	3
	SPAU	1	2	2
	VECNA	1	3	5
	oos	5	17	24
	WinGate 1080 Attempt	5	5	22
	UNKNOWN	2	5	6
	Probable NMAP finger	1	1	1
MY.NET.105.120	NOACK	1	2	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	Null scan!	2	2	21
	Probable NMAP finger	1	1	3
	connect to 515 from	1	1	1
	oos	3	63	138
	WinGate 1080 Attempt	4	4	4
	UNKNOWN	1	1	1
MY.NET.6.7	INVALIDACK	3	3	3
	NULL	3	3	3
	oos	14	14	22
	XMAS	1	1	1
	Watchlist 000222 NET	4	4	540
	VECNA	3	3	3
	UNKNOWN	5	5	5
	Null scan!	3	3	3
	NOACK	17	19	30
	NMAP TCP ping!	3	3	4
MY.NET.202.94	Attempted Sun RPC hi	1	1	4
	Back Orifice	1	1	3
	FIN	1	1	1
	INVALIDACK	1	3	3
	SUNRPC highport acce	1	1	6
	UNKNOWN	2	2	3
	WinGate 1080 Attempt	11	11	54
	oos	5	6	6
	XMAS	1	1	1
	Watchlist 000220 IL-	4	4	5181
	UDP	1	1	2619
	Null scan!	2	2	2
	NULL	2	2	2
	NOACK	2	4	6
MY.NET.5.10	INVALIDACK	18	23	28
	NOACK	5	5	5
	NULL	2	2	3
	oos	5	5	5
	UNKNOWN	14	14	16
	Null scan!	5	5	5
MY.NET.220.126	Watchlist 000220 IL-	1	1	25182
	oos	1	1	1
MY.NET.60.14	INVALIDACK	1	1	1
	NMAP TCP ping!	16	16	23
	Watchlist 000220 IL-	1	1	1
	VECNA	1	1	1
	NOACK	11	12	17
	oos	17	17	18
MY.NET.15.178	WinGate 1080 Attempt	37	37	75
	oos	4	4	4
MY.NET.1.8	DNS udp DoS attack d	4	4	6
	NMAP TCP ping!	8	8	63
	Tiny Fragments - Pos	17	17	3148
	oos	1	1	1
	connect to 515 from	1	1	1
	UDP	1	1	1
MY.NET.201.130	FIN	1	1	1
	NOACK	1	1	1
	Null scan!	3	3	4
	Queso fingerprint	10	10	127
	NULL	2	2	2
	SMB Name Wildcard	1	1	2
	connect to 515 from	1	1	1
	oos	10	10	33
	Watchlist 000220 IL-	1	1	1912
MY.NET.201.78	INVALIDACK	4	4	6
	NMAPID	1	1	1
	Null scan!	6	6	9
	NULL	4	5	7
	NOACK	5	5	6
	Queso fingerprint	1	1	6
	UNKNOWN	1	1	1
	WinGate 1080 Attempt	1	1	2
	oos	3	3	3
	VECNA	2	2	2
	UDP	1	1	1601

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.202.46	FIN	1	1	1
	INVALIDACK	3	3	3
	NOACK	2	3	3
	Null scan!	1	1	2
	WinGate 1080 Attempt	1	1	2
	UNKNOWN	2	2	2
	Queso fingerprint	4	4	20
	NULL	1	1	2
	XMAS	1	1	1
	oos	11	14	44
MY.NET.201.66	Null scan!	1	1	1
	Watchlist 000220 IL-	1	1	2
	Queso fingerprint	7	7	28
	oos	17	18	52
MY.NET.1.4	DNS udp DoS attack d	1	1	5390
	NMAP TCP ping!	5	5	14
	UDP	6	6	7
	oos	4	4	4
MY.NET.70.255	Broadcast Ping	24	24	154
	oos	2	2	2
MY.NET.1.3	DNS udp DoS attack d	1	1	5411
	NMAP TCP ping!	9	9	38
	oos	3	3	3
	UDP	2	2	2
MY.NET.1.5	DNS udp DoS attack d	1	1	5331
	UDP	6	6	21
	oos	3	3	3
	NMAP TCP ping!	4	4	18
MY.NET.1.10	DNS udp DoS attack d	3	3	6
	NMAP TCP ping!	2	2	6
	oos	3	3	3
	Tiny Fragments - Pos	14	14	1264
MY.NET.209.78	FULLXMAS	1	1	1
	INVALIDACK	1	1	2
	NOACK	1	10	14
	Probable NMAP finger	1	1	1
	Null scan!	1	1	3
	NULL	1	1	2
	UNKNOWN	1	3	3
	oos	3	6	11
MY.NET.253.41	FIN	1	1	1
	NOACK	2	2	4
	NMAP TCP ping!	3	3	3
	Queso fingerprint	2	2	4
	oos	5	5	9
	Watchlist 000222 NET	8	8	278
	Watchlist 000220 IL-	2	2	8
MY.NET.201.62	FIN	1	1	1
	Null scan!	2	2	2
	Queso fingerprint	7	7	51
	NULL	2	2	2
	oos	10	10	68
MY.NET.98.133	Null scan!	1	1	1
	UDP	2	2	5543
	WinGate 1080 Attempt	4	4	4
	oos	4	4	4
MY.NET.180.26	FIN	1	1	1
	INVALIDACK	2	3	3
	NULL	4	4	11
	WinGate 1080 Attempt	1	1	1
	Queso fingerprint	1	1	2
	Null scan!	4	4	11
	connect to 515 from	1	1	1
	oos	6	7	9
MY.NET.225.234	Attempted Sun RPC hi	1	1	3
	Watchlist 000220 IL-	1	1	9309
	oos	1	1	1
MY.NET.213.158	Attempted Sun RPC hi	9	9	556
	Queso fingerprint	1	1	1
	SUNRPC highport acce	7	7	104
	oos	2	2	2
MY.NET.253.105	INVALIDACK	3	3	8

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	NMAP TCP ping!	1	1	2
	Null scan!	4	4	4
	NULL	2	2	3
	oos	10	10	11
MY.NET.253.42	FIN	1	1	1
	NMAP TCP ping!	1	1	1
	NOACK	1	1	3
	TCP SMTP Source Port	1	1	11
	Watchlist 000222 NET	6	6	112
	oos	5	5	8
	Watchlist 000220 IL-	1	1	32
	Queso fingerprint	3	3	4
MY.NET.219.246	INVALIDACK	1	1	1
	WinGate 1080 Attempt	10	10	11
	oos	3	3	3
	Null scan!	3	3	3
	NULL	2	2	2
MY.NET.253.43	NMAP TCP ping!	2	2	3
	NOACK	1	1	6
	Queso fingerprint	2	2	7
	Watchlist 000222 NET	6	6	103
	oos	4	4	14
	Watchlist 000220 IL-	3	3	56
MY.NET.6.15	External RPC call	9	9	26
	FIN	1	1	1
	SMB Name Wildcard	2	2	67
	STATDX UDP attack	1	1	1
	oos	2	2	2
	connect to 515 from	2	2	2
	VECNA	1	1	1
MY.NET.100.230	NMAP TCP ping!	3	3	11
	Queso fingerprint	1	1	1
	Watchlist 000222 NET	6	6	789
	Tiny Fragments - Pos	3	3	5
	oos	3	3	3
MY.NET.98.114	INVALIDACK	4	4	5
	NULL	3	3	3
	Null scan!	3	3	3
	WinGate 1080 Attempt	1	1	3
	oos	1	1	1
	Watchlist 000220 IL-	1	1	1221
	UDP	2	2	44
MY.NET.217.150	Back Orifice	1	1	1
	Null scan!	2	2	3
	SUNRPC highport acce	1	1	1
	UNKNOWN	1	1	1
	oos	2	7	7
	WinGate 1080 Attempt	1	1	1
	UDP	3	3	111
	SMB Name Wildcard	1	1	4
MY.NET.217.158	FULLXMAS	1	1	1
	INVALIDACK	1	1	1
	NULL	1	1	1
	Queso fingerprint	1	1	1
	oos	4	7	7
	WinGate 1080 Attempt	1	1	1
	UNKNOWN	1	1	2
	Null scan!	1	1	1
	NOACK	2	3	3
MY.NET.98.185	Queso fingerprint	1	1	1
	UDP	1	1	2278
	WinGate 1080 Attempt	2	2	8
	oos	4	4	5
	UNKNOWN	2	4	6
MY.NET.201.46	FIN	1	1	1
	Null scan!	1	1	1
	WinGate 1080 Attempt	9	9	25
	oos	2	2	2
	UDP	3	3	27
MY.NET.223.86	INVALIDACK	1	2	2
	NOACK	1	1	1
	UNKNOWN	1	2	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	connect to 515 from	1	1	1
	WinGate 1080 Attempt	4	4	13
	VECNA	1	2	2
	oos	4	4	4
MY.NET.217.242	INVALIDACK	1	1	1
	Null scan!	10	10	10
	WinGate 1080 Attempt	1	1	1
	oos	1	1	1
	Queso fingerprint	1	1	6
	NULL	2	2	4
MY.NET.181.144	NMAP TCP ping!	5	5	5
	oos	9	9	10
	Null scan!	1	1	1
	NOACK	1	1	2
MY.NET.208.78	FIN	12	12	1438
	oos	1	1	1
MY.NET.101.192	SMB Name Wildcard	1	1	58
	SNMP public access	14	14	316
MY.NET.219.18	INVALIDACK	3	3	3
	NOACK	1	1	1
	UDP	5	5	52
	WinGate 1080 Attempt	1	1	1
	oos	2	2	2
	Watchlist 000220 IL-	3	3	14
MY.NET.217.138	INVALIDACK	1	1	1
	NULL	1	1	2
	Null scan!	1	1	2
	oos	4	5	5
	WinGate 1080 Attempt	1	1	3
	Watchlist 000220 IL-	3	3	1438
MY.NET.201.142	INVALIDACK	2	3	3
	Null scan!	1	1	1
	Watchlist 000220 IL-	4	4	51
	NULL	1	1	1
	oos	5	5	5
MY.NET.201.74	INVALIDACK	2	2	2
	oos	2	2	2
	Queso fingerprint	1	1	3
	NOACK	3	3	4
	NULL	3	3	4
	Null scan!	3	3	4
MY.NET.217.94	UDP	11	11	934
	oos	1	1	1
MY.NET.98.174	SUNRPC highport acce	1	1	1
	UDP	1	1	2261
	oos	2	2	2
	WinGate 1080 Attempt	5	5	10
MY.NET.229.114	Watchlist 000220 IL-	1	1	5080
	oos	2	2	2
MY.NET.99.51	NULL	4	4	4
	SUNRPC highport acce	2	2	42
	oos	1	1	1
	Null scan!	5	5	5
	SMB Name Wildcard	1	1	2
MY.NET.98.152	WinGate 1080 Attempt	9	9	14
	oos	4	4	4
MY.NET.60.17	Back Orifice	1	1	1
	UNKNOWN	1	1	1
	UDP	1	1	1
	NOACK	1	1	1
	VECNA	1	1	1
	WinGate 1080 Attempt	5	5	6
	oos	1	1	2
	connect to 515 from	1	1	1
	Watchlist 000220 IL-	1	1	2
MY.NET.179.77	INVALIDACK	2	2	2
	NMAP TCP ping!	1	1	1
	NOACK	5	6	8
	XMAS	1	1	1
	oos	2	2	2
	UNKNOWN	1	1	1
MY.NET.201.30	FULLXMAS	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	NMAPID	1	1	1
	NOACK	1	2	2
	oos	4	5	5
	Null scan!	2	2	2
	NULL	2	2	4
MY.NET.228.214	Watchlist 000220 IL-	1	1	4445
	oos	3	3	3
MY.NET.6.35	NMAP TCP ping!	3	3	3
	NOACK	1	1	2
	Watchlist 000222 NET	3	3	55
	oos	4	4	4
	Watchlist 000220 IL-	1	1	2
MY.NET.223.226	NOACK	1	1	3
	Queso fingerprint	2	2	38
	WinGate 1080 Attempt	1	1	1
	XMAS	1	1	3
	oos	6	7	20
MY.NET.98.177	INVALIDACK	2	2	2
	UDP	1	1	12
	UNKNOWN	1	1	8
	oos	2	2	2
	WinGate 1080 Attempt	5	5	6
	NULL	1	1	1
MY.NET.98.123	Tiny Fragments - Pos	1	1	2
	Watchlist 000220 IL-	1	1	44
	oos	2	2	2
	WinGate 1080 Attempt	2	2	2
	UDP	2	2	1775
MY.NET.6.47	Happy 99 Virus	1	1	1
	oos	4	4	4
	NMAP TCP ping!	2	2	2
	SUNRPC highport acce	1	1	1
	Watchlist 000220 IL-	1	1	1
	Watchlist 000222 NET	1	1	15
	NOACK	1	1	2
MY.NET.224.62	Attempted Sun RPC hi	1	1	11
	INVALIDACK	1	4	4
	WinGate 1080 Attempt	1	1	2
	UNKNOWN	1	2	2
	NOACK	1	1	1
	oos	2	2	2
MY.NET.53.184	NOACK	1	1	1
	Queso fingerprint	4	4	6
	oos	5	5	11
	XMAS	1	1	1
MY.NET.100.130	External RPC call	5	5	5
	oos	4	4	4
	SMB Name Wildcard	2	2	10
MY.NET.98.132	UNKNOWN	1	2	2
	WinGate 1080 Attempt	6	6	11
	oos	3	3	5
MY.NET.98.193	NOACK	1	1	3
	oos	1	2	4
	WinGate 1080 Attempt	4	4	6
	UNKNOWN	1	2	2
	NULL	1	1	1
	Null scan!	1	1	1
MY.NET.1.9	DNS udp DoS attack d	2	2	2
	NMAP TCP ping!	2	2	5
	Tiny Fragments - Pos	5	5	8
	oos	1	1	1
	UDP	1	1	1
MY.NET.219.214	FULLXMAS	1	2	2
	SPAU	1	1	2
	VECNA	1	1	1
	WinGate 1080 Attempt	1	1	1
	UNKNOWN	1	2	6
	NOACK	1	2	3
	INVALIDACK	1	2	2
MY.NET.150.83	NULL	1	1	1
	oos	2	2	2
	UNKNOWN	8	8	13

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.201.210	FIN	1	1	2
	oos	3	3	4
	INVALIDACK	4	4	7
	NOACK	1	1	1
	WinGate 1080 Attempt	2	2	2
MY.NET.201.102	NOACK	1	2	4
	NULL	2	2	4
	oos	3	3	3
	Null scan!	4	4	4
MY.NET.202.186	INVALIDACK	2	2	2
	Null scan!	1	1	2
	NULL	1	1	2
	NOACK	1	1	1
	WinGate 1080 Attempt	1	1	1
	oos	4	5	5
MY.NET.15.127	External RPC call	6	6	6
	WinGate 1080 Attempt	1	1	1
	oos	4	4	4
MY.NET.98.188	INVALIDACK	1	1	3
	WinGate 1080 Attempt	5	5	12
	oos	1	1	1
	UDP	1	1	944
	NOACK	1	1	3
MY.NET.217.142	UDP	9	9	210
	oos	1	1	1
MY.NET.98.156	Null scan!	1	1	1
	Queso fingerprint	1	1	1
	oos	5	5	6
	WinGate 1080 Attempt	1	1	1
	UNKNOWN	1	2	29
MY.NET.201.146	NULL	1	1	1
	oos	1	1	1
	Null scan!	2	2	2
	WinGate 1080 Attempt	6	6	33
MY.NET.98.169	UNKNOWN	1	1	2
	oos	1	1	1
	WinGate 1080 Attempt	8	8	26
MY.NET.98.194	UNKNOWN	1	1	3
	oos	1	1	1
	WinGate 1080 Attempt	8	8	17
MY.NET.218.218	NULL	1	1	1
	oos	2	3	3
	WinGate 1080 Attempt	5	5	16
	Null scan!	1	1	1
MY.NET.202.234	INVALIDACK	2	2	4
	oos	3	4	4
	VECNA	1	1	1
	Watchlist 000220 IL-	3	3	11
MY.NET.98.190	FIN	1	1	1
	WinGate 1080 Attempt	4	4	9
	oos	3	3	4
	Watchlist 000220 IL-	1	1	2
	NOACK	1	1	2
MY.NET.202.34	INVALIDACK	1	3	3
	WinGate 1080 Attempt	3	3	10
	oos	3	4	4
MY.NET.98.138	NULL	1	1	1
	oos	3	3	3
	WinGate 1080 Attempt	5	5	8
	Null scan!	1	1	1
MY.NET.217.50	SPAU	1	1	1
	VECNA	1	1	1
	WinGate 1080 Attempt	6	6	8
	oos	2	2	2
MY.NET.140.151	FIN	1	1	1
	INVALIDACK	1	1	1
	Null scan!	3	3	3
	NULL	3	3	4
	oos	2	2	2
MY.NET.212.214	Null scan!	3	3	4
	Queso fingerprint	1	1	1
	WinGate 1080 Attempt	2	2	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	oos	4	4	4
MY.NET.203.98	INVALIDACK	1	1	1
	SPAU	1	1	1
	oos	5	5	5
	WinGate 1080 Attempt	1	1	1
	NOACK	2	2	2
MY.NET.201.90	INVALIDACK	3	3	3
	WinGate 1080 Attempt	1	1	1
	oos	2	2	2
	SPAU	1	1	1
	NOACK	2	3	3
MY.NET.98.197	UDP	3	3	2225
	WinGate 1080 Attempt	2	2	7
MY.NET.208.22	UDP	1	1	12
	WinGate 1080 Attempt	7	7	110
	oos	1	1	1
MY.NET.253.51	NOACK	4	4	5
	Watchlist 000222 NET	3	3	43
	oos	2	2	5
MY.NET.100.203	WinGate 1080 Attempt	9	9	41
MY.NET.98.192	Attempted Sun RPC hi	1	1	10
	oos	2	2	2
	WinGate 1080 Attempt	5	5	7
	Watchlist 000220 IL-	1	1	1
MY.NET.201.206	NOACK	1	2	6
	WinGate 1080 Attempt	4	4	8
	oos	2	2	2
	Watchlist 000220 IL-	1	1	3
MY.NET.217.98	NOACK	1	1	1
	WinGate 1080 Attempt	6	6	13
	oos	2	2	2
MY.NET.201.122	INVALIDACK	1	1	1
	UNKNOWN	1	1	2
	oos	2	2	2
	VECNA	1	1	1
	Null scan!	2	2	3
	NOACK	1	1	3
	NULL	1	1	1
MY.NET.97.242	Back Orifice	1	1	1
	WinGate 1080 Attempt	5	5	9
	oos	3	3	3
MY.NET.98.191	SUNRPC highport acce	1	1	1
	oos	2	5	6
	WinGate 1080 Attempt	3	3	3
MY.NET.215.138	FULLXMAS	1	1	1
	oos	3	5	5
	NOACK	1	1	1
	VECNA	1	1	1
	UNKNOWN	1	1	1
MY.NET.140.2	FIN	1	1	1
	connect to 515 from	1	1	1
	oos	4	4	4
	VECNA	1	1	1
	NOACK	1	1	1
	TCP SMTP Source Port	1	1	1
MY.NET.202.30	Watchlist 000220 IL-	1	1	2288
	WinGate 1080 Attempt	1	1	1
	oos	2	2	2
MY.NET.98.165	UDP	1	1	2244
	oos	1	1	1
	WinGate 1080 Attempt	2	2	3
MY.NET.217.162	NULL	1	1	2
	Null scan!	1	1	2
	Tiny Fragments - Pos	1	1	727
	WinGate 1080 Attempt	1	1	1
	oos	3	3	3
MY.NET.6.34	NMAP TCP ping!	2	2	6
	NOACK	1	1	2
	oos	2	2	2
	Watchlist 000222 NET	3	3	40
MY.NET.98.163	UDP	1	1	17
	UNKNOWN	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	oos	4	4	5
	WinGate 1080 Attempt	2	2	4
MY.NET.98.184	WinGate 1080 Attempt	7	7	19
	oos	1	1	1
MY.NET.202.26	Queso fingerprint	1	1	5
	UNKNOWN	1	1	1
	WinGate 1080 Attempt	1	1	1
	oos	5	5	10
MY.NET.97.173	INVALIDACK	1	1	1
MY.NET.98.117	WinGate 1080 Attempt	7	7	14
	oos	1	1	1
MY.NET.97.173	WinGate 1080 Attempt	3	3	10
	oos	4	4	4
MY.NET.217.238	Null scan!	1	1	1
	WinGate 1080 Attempt	6	6	13
	oos	1	1	1
MY.NET.98.147	NMAP TCP ping!	1	1	3
	Probable NMAP finger	1	1	1
MY.NET.98.168	oos	2	2	2
	WinGate 1080 Attempt	6	6	11
MY.NET.98.147	WinGate 1080 Attempt	4	4	7
	Queso fingerprint	1	1	1
	oos	1	1	1
MY.NET.184.33	NOACK	1	1	1
	NULL	1	1	3
	Null scan!	1	1	3
	connect to 515 from	1	1	1
	oos	3	3	3
	UNKNOWN	1	1	1
MY.NET.206.210	INVALIDACK	1	2	2
	NOACK	1	1	1
	Watchlist 000220 IL-	2	2	6
	oos	2	3	3
MY.NET.98.154	Probable NMAP finger	1	1	1
	WinGate 1080 Attempt	5	5	8
MY.NET.98.202	NOACK	1	1	1
	oos	2	2	2
	WinGate 1080 Attempt	4	4	7
	INVALIDACK	1	1	1
MY.NET.98.154	oos	2	2	2
MY.NET.97.86	WinGate 1080 Attempt	7	7	10
	oos	1	1	1
MY.NET.217.90	INVALIDACK	1	1	1
	NOACK	1	1	1
	Queso fingerprint	1	1	1
	VECNA	1	2	5
	oos	3	3	3
MY.NET.217.114	INVALIDACK	1	2	2
	NOACK	1	1	1
	Null scan!	1	1	1
	NULL	2	2	2
	UNKNOWN	1	1	3
	oos	1	1	1
MY.NET.97.234	Queso fingerprint	1	1	1
	WinGate 1080 Attempt	5	5	7
	oos	2	2	2
MY.NET.75.147	FIN	1	1	3
	INVALIDACK	1	1	1
	NOACK	1	1	1
	oos	4	4	4
	UNKNOWN	1	1	1
MY.NET.98.196	WinGate 1080 Attempt	6	6	7
	oos	2	2	2
MY.NET.97.169	WinGate 1080 Attempt	3	3	4
	oos	5	5	5
MY.NET.106.222	NOACK	2	4	4
	UNKNOWN	1	1	1
	oos	2	2	2
	VECNA	1	1	2
MY.NET.215.106	NULL	1	1	1
	Tiny Fragments - Pos	1	1	2
	oos	4	4	4

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	WinGate 1080 Attempt	2	2	2
MY.NET.201.150	WinGate 1080 Attempt	5	5	6
MY.NET.201.186	oos	1	2	2
MY.NET.201.150	oos	3	3	3
MY.NET.201.186	NULL	1	1	1
	Null scan!	1	1	1
	connect to 515 from	1	1	1
	NOACK	3	3	4
MY.NET.201.190	NULL	2	2	2
	Null scan!	3	3	3
	oos	3	3	3
MY.NET.206.226	NULL	1	1	1
	oos	4	4	4
	WinGate 1080 Attempt	1	1	1
	Null scan!	1	1	1
	UDP	1	1	1
MY.NET.25.135	connect to 515 from	1	1	1
	oos	7	7	7
MY.NET.209.154	INVALIDACK	1	1	1
	oos	1	1	1
	Watchlist 000220 IL-	4	4	858
MY.NET.217.62	INVALIDACK	2	2	2
	NOACK	1	2	2
	oos	1	1	1
	Watchlist 000220 IL-	2	2	340
MY.NET.98.128	UDP	2	2	319
	WinGate 1080 Attempt	2	2	3
	oos	3	3	3
MY.NET.97.213	Attempted Sun RPC hi	1	1	224
	oos	4	4	4
	WinGate 1080 Attempt	1	1	1
	UDP	1	1	22
MY.NET.212.38	Null scan!	1	1	1
	Watchlist 000220 IL-	2	2	724
	oos	3	3	3
MY.NET.6.50	UDP	2	2	191
	oos	5	5	5
MY.NET.98.110	UDP	1	1	1127
	oos	1	1	1
	WinGate 1080 Attempt	3	3	3
MY.NET.224.242	INVALIDACK	1	1	3
	oos	4	4	16
	Queso fingerprint	2	2	37
MY.NET.145.9	NMAP TCP ping!	1	1	1
	Queso fingerprint	1	1	3
	Watchlist 000222 NET	2	2	26
	oos	3	3	3
MY.NET.1.2	Watchlist 000222 NET	3	3	27
	oos	3	3	3
	connect to 515 from	1	1	1
MY.NET.253.52	NOACK	1	1	1
	Watchlist 000222 NET	4	4	22
	oos	2	2	2
MY.NET.130.187	INVALIDACK	1	1	1
	Watchlist 000220 IL-	1	1	1517
	oos	2	2	2

Table - Possible compromised hosts

DST	TYPE	DSRC	CNT	TOTAL
MY.NET.98.200	SYN	3	3	4
	UDP	10462	10462	32402
	spp_portscan: PORTSC	1	0	25
MY.NET.100.230	SYN	1602	1602	8544
	UDP	7081	7081	50219
	spp_portscan: PORTSC	1	0	3009
MY.NET.217.150	FIN	59	51	193
	NMAPID	13	11	26
	NULL	76	63	239
	SYN	58	52	4507

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	UDP	3170	3170	14891
	VECNA	97	50	279
	spp_portscan: PORTSC	1	0	6290
	oos	1394	250	5415
	XMAS	12	11	19
	UNKNOWN	160	55	551
	SYNFIN	15	10	75
	SPAU	10	6	51
	NOACK	418	97	1176
	INVALIDACK	291	78	874
	FULLXMAS	29	25	79
MY.NET.98.238	SYN	5779	5779	7148
	spp_portscan: PORTSC	1	0	7
MY.NET.97.148	UDP	5064	5064	7972
	oos	3	3	12
	spp_portscan: PORTSC	1	0	2
MY.NET.201.46	SYN	1611	1611	3386
	UDP	2558	2558	7464
	spp_portscan: PORTSC	1	0	77
MY.NET.97.19	UDP	4112	4112	4154
	spp_portscan: PORTSC	1	0	2
MY.NET.97.93	SYN	1	1	1
	spp_portscan: PORTSC	1	0	8
	UDP	4037	4037	10043
MY.NET.206.186	UDP	3875	3875	9577
	spp_portscan: PORTSC	1	0	55
MY.NET.224.102	UDP	3713	3713	6411
	spp_portscan: PORTSC	1	0	5
MY.NET.212.34	UDP	3494	3494	7544
	spp_portscan: PORTSC	1	0	3
MY.NET.217.158	FIN	109	90	420
	NULL	124	102	479
	SYN	72	62	460
	spp_portscan: PORTSC	1	0	4935
	oos	1037	153	3900
	XMAS	29	25	112
	VECNA	156	77	533
	UNKNOWN	321	95	1065
	UDP	158	158	935
	SYNFIN	24	20	59
	SPAU	24	21	73
	NOACK	769	135	2230
	FULLXMAS	53	42	133
	INVALIDACK	545	120	1534
	NMAPID	28	22	87
MY.NET.210.70	UDP	3314	3314	5365
	spp_portscan: PORTSC	1	0	10
MY.NET.208.194	UDP	3288	3288	4104
	spp_portscan: PORTSC	1	0	9
MY.NET.98.136	SYN	71	71	273
	UDP	3135	3135	4647
	spp_portscan: PORTSC	1	0	38
MY.NET.213.186	SYN	6	6	7
	spp_portscan: PORTSC	1	0	160
	UDP	3067	3067	54667
MY.NET.210.194	UDP	2839	2839	5786
	spp_portscan: PORTSC	1	0	9
MY.NET.98.194	UDP	2829	2829	3980
	spp_portscan: PORTSC	1	0	28
	UNKNOWN	1	1	3
MY.NET.209.114	SYN	2	2	2
	UDP	2803	2803	5350
	spp_portscan: PORTSC	1	0	11
MY.NET.219.166	SYN	2	2	7
	UDP	2622	2622	6853
	spp_portscan: PORTSC	1	0	19
MY.NET.217.58	SYN	2	2	6
	spp_portscan: PORTSC	1	0	16
	UDP	2590	2590	9049
MY.NET.217.94	SYN	2	2	19
	UDP	2520	2520	33715
	spp_portscan: PORTSC	1	0	31

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.98.106	SYN	254	254	1458
	UDP	2314	2314	3442
	spp_portscan: PORTSC	1	0	33
MY.NET.217.142	SYN	1	1	5
	UDP	2386	2386	10284
	spp_portscan: PORTSC	1	0	69
MY.NET.156.110	SYN	9	9	60
	UDP	2327	2327	16527
	spp_portscan: PORTSC	1	0	156
MY.NET.98.195	UDP	2190	2190	2379
	spp_portscan: PORTSC	1	0	2
MY.NET.97.71	UDP	1930	1930	1950
	spp_portscan: PORTSC	1	0	1
MY.NET.98.141	SYN	1	1	1
	spp_portscan: PORTSC	1	0	1
	UDP	1832	1832	1838
MY.NET.202.94	SYN	1671	1671	5691
	UDP	14	14	29458
	spp_portscan: PORTSC	1	0	155
MY.NET.203.194	UDP	1710	1710	2259
	spp_portscan: PORTSC	1	0	6
MY.NET.201.210	SYN	1142	1142	2053
	spp_portscan: PORTSC	1	0	33
	UDP	501	501	762
MY.NET.97.183	SYN	2	2	3
	spp_portscan: PORTSC	1	0	4
	UDP	1522	1522	1677
MY.NET.97.59	UDP	1496	1496	1504
	spp_portscan: PORTSC	1	0	3
MY.NET.218.130	SYN	21	21	58
	UDP	1420	1420	18420
	spp_portscan: PORTSC	1	0	71
MY.NET.210.82	UDP	1418	1418	1685
	spp_portscan: PORTSC	1	0	6
MY.NET.98.188	SYN	1	1	1
	UDP	1410	1410	1413
	spp_portscan: PORTSC	1	0	4
MY.NET.209.250	UDP	1365	1365	1712
	spp_portscan: PORTSC	1	0	2
MY.NET.225.142	UDP	1333	1333	1767
	spp_portscan: PORTSC	1	0	10
MY.NET.217.186	FIN	1	1	1
	UNKNOWN	1	1	1
	spp_portscan: PORTSC	1	0	9
	oos	1	1	1
	UDP	1316	1316	1832
	INVALIDACK	1	1	1
MY.NET.97.89	UDP	1309	1309	1426
	spp_portscan: PORTSC	1	0	8
MY.NET.253.24	SYN	1233	1233	31840
	spp_portscan: PORTSC	1	0	2002
MY.NET.97.208	SYN	368	368	2149
	UDP	909	909	5026
	spp_portscan: PORTSC	1	0	47
MY.NET.60.8:	spp_portscan: portsc	1282	0	2131
MY.NET.220.58	UDP	1262	1262	2593
	spp_portscan: PORTSC	1	0	18
MY.NET.97.161	SYN	3	3	5
	UDP	1220	1220	3587
	spp_portscan: PORTSC	1	0	7
MY.NET.202.2	UDP	1222	1222	1949
	spp_portscan: PORTSC	1	0	14
MY.NET.97.198	SYN	8	8	9
	UDP	1202	1202	2634
	spp_portscan: PORTSC	1	0	18
MY.NET.98.192	SYN	9	9	9
	spp_portscan: PORTSC	1	0	1
	UDP	1185	1185	1223
MY.NET.201.214	SYN	2	2	4
	UDP	1177	1177	3734
	spp_portscan: PORTSC	1	0	10
MY.NET.219.18	SYN	109	109	136

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	47
	UDP	1061	1061	4365
MY.NET.217.146	UDP	1152	1152	1499
	spp_portscan: PORTSC	1	0	2
MY.NET.217.106	SYN	7	7	16
	UDP	1119	1119	13842
	spp_portscan: PORTSC	1	0	21
MY.NET.217.182	FIN	17	16	38
	SPAU	14	11	33
	spp_portscan: PORTSC	1	0	1328
	oos	447	100	1064
	XMAS	10	9	19
	VECNA	56	35	139
	UNKNOWN	81	31	167
	UDP	4	4	7
	SYNFIN	7	6	14
	SYN	11	9	22
	NULL	42	36	89
	FULLXMAS	21	15	62
	INVALIDACK	193	59	406
	NMAPID	3	3	3
	NOACK	225	64	501
MY.NET.201.50	SYN	9	9	44
	UDP	1065	1065	14120
	spp_portscan: PORTSC	1	0	50
MY.NET.98.177	SYN	310	310	2284
	UDP	729	729	8259
	spp_portscan: PORTSC	1	0	42
MY.NET.226.174	UDP	1049	1049	1459
	spp_portscan: PORTSC	1	0	6
MY.NET.205.254	UDP	993	993	1088
	spp_portscan: PORTSC	1	0	8
MY.NET.209.170	UDP	978	978	1626
	spp_portscan: PORTSC	1	0	12
MY.NET.215.242	UDP	959	959	3060
	spp_portscan: PORTSC	1	0	15
MY.NET.98.157	SYN	51	51	51
	UDP	903	903	1808
	oos	2	2	2
	spp_portscan: PORTSC	1	0	8
MY.NET.98.125	SYN	1	1	3
	UDP	949	949	2175
	spp_portscan: PORTSC	1	0	28
MY.NET.209.6	UDP	936	936	1770
	spp_portscan: PORTSC	1	0	9
MY.NET.220.210	UDP	900	900	1079
	spp_portscan: PORTSC	1	0	2
MY.NET.219.126	FIN	22	19	52
	FULLXMAS	17	12	41
	NMAPID	6	4	12
	INVALIDACK	157	45	400
	NULL	34	28	84
	SYN	49	46	464
	UDP	2	2	26
	VECNA	35	19	84
	oos	237	44	784
	spp_portscan: PORTSC	1	0	2203
	XMAS	8	6	22
	UNKNOWN	92	30	220
	SYNFIN	4	3	7
	SPAU	10	7	22
	NOACK	211	44	636
MY.NET.218.206	UDP	891	891	1094
MY.NET.206.14	UDP	884	884	3296
	spp_portscan: PORTSC	1	0	22
MY.NET.220.162	UDP	877	877	1483
	spp_portscan: PORTSC	1	0	14
MY.NET.218.6	UDP	870	870	1025
	spp_portscan: PORTSC	1	0	1
MY.NET.211.158	UDP	868	868	1149
	spp_portscan: PORTSC	1	0	4
MY.NET.225.194	UDP	850	850	932

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	1
MY.NET.98.127	SYN	10	10	10
	UDP	801	801	1219
	spp_portscan: PORTSC	1	0	12
MY.NET.98.164	SYN	4	4	5
	UDP	799	799	3955
	spp_portscan: PORTSC	1	0	25
MY.NET.98.151	SYN	12	12	99
	connect to 515 from	1	1	9
	spp_portscan: PORTSC	1	0	14
	UDP	771	771	2342
MY.NET.98.158	SYN	9	9	17
	spp_portscan: PORTSC	1	0	4
	UDP	756	756	1203
MY.NET.218.170	UDP	748	748	1357
	spp_portscan: PORTSC	1	0	4
MY.NET.205.30	UDP	748	748	1078
	spp_portscan: PORTSC	1	0	11
MY.NET.205.222	UDP	742	742	1355
	spp_portscan: PORTSC	1	0	17
MY.NET.209.90	UDP	738	738	1324
	spp_portscan: PORTSC	1	0	3
MY.NET.1.8	SYN	1	1	2
	UDP	735	735	1298
	spp_portscan: PORTSC	1	0	16
MY.NET.97.165	SYN	377	377	2030
	UDP	346	346	2336
	spp_portscan: PORTSC	1	0	113
MY.NET.202.254	UDP	720	720	1961
	spp_portscan: PORTSC	1	0	23
MY.NET.98.159	UDP	716	716	1221
	spp_portscan: PORTSC	1	0	14
MY.NET.201.86	SYN	1	1	1
	spp_portscan: PORTSC	1	0	21
	UDP	704	704	6655
MY.NET.98.202	UDP	700	700	860
	oos	1	1	2
	spp_portscan: PORTSC	1	0	2
MY.NET.98.189	UDP	684	684	922
	spp_portscan: PORTSC	1	0	6
MY.NET.202.74	UDP	682	682	1763
	spp_portscan: PORTSC	1	0	15
MY.NET.208.142	UDP	681	681	1609
	spp_portscan: PORTSC	1	0	11
MY.NET.253.52	SYN	675	675	4293
	spp_portscan: PORTSC	1	0	109
MY.NET.97.10	SYN	1	1	1
	UDP	677	677	1028
	spp_portscan: PORTSC	1	0	5
MY.NET.226.234	UDP	676	676	1042
	spp_portscan: PORTSC	1	0	4
MY.NET.97.233	SYN	1	1	1
	UDP	660	660	937
MY.NET.218.158	SYN	5	5	16
	spp_portscan: PORTSC	1	0	12
	oos	1	1	1
	UDP	598	598	25174
MY.NET.97.196	SYN	2	2	2
	spp_portscan: PORTSC	1	0	17
	UDP	651	651	680
MY.NET.202.226	UDP	642	642	2766
	spp_portscan: PORTSC	1	0	5
MY.NET.1.3	UDP	622	622	7410
	spp_portscan: PORTSC	1	0	330
MY.NET.201.242	SYN	3	3	68
	UDP	612	612	3085
	spp_portscan: PORTSC	1	0	28
MY.NET.97.203	SYN	166	166	770
	UDP	448	448	448
	spp_portscan: PORTSC	1	0	25
MY.NET.223.78	UDP	588	588	777
	spp_portscan: PORTSC	1	0	8

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.209.2	UDP	569	569	916
	spp_portscan: PORTSC	1	0	5
MY.NET.217.54	SYN	1	1	1
	UDP	563	563	1118
	spp_portscan: PORTSC	1	0	27
MY.NET.253.53	SYN	549	549	3124
	spp_portscan: PORTSC	1	0	104
MY.NET.202.34	SYN	4	4	5
	spp_portscan: PORTSC	1	0	15
	UDP	540	540	2758
MY.NET.207.250	SYN	2	2	3
	UDP	543	543	967
	spp_portscan: PORTSC	1	0	20
MY.NET.202.6	SYN	1	1	14
	UDP	539	539	3011
	spp_portscan: PORTSC	1	0	50
MY.NET.218.210	UDP	540	540	639
	spp_portscan: PORTSC	1	0	1
MY.NET.97.247	SYN	11	11	25
	spp_portscan: PORTSC	1	0	80
	UDP	519	519	2564
MY.NET.253.51	SYN	528	528	2646
	spp_portscan: PORTSC	1	0	106
MY.NET.217.78	SYN	529	529	1160
	spp_portscan: PORTSC	1	0	40
MY.NET.214.166	UDP	473	473	27825
	spp_portscan: PORTSC	1	0	123
MY.NET.204.134	SYN	1	1	1
	UDP	520	520	547
	spp_portscan: PORTSC	1	0	3
MY.NET.223.246	UDP	510	510	675
	spp_portscan: PORTSC	1	0	1
MY.NET.97.190	SYN	1	1	1
	spp_portscan: PORTSC	1	0	12
	UDP	500	500	792
MY.NET.217.102	SYN	1	1	2
	spp_portscan: PORTSC	1	0	5
	UDP	487	487	1512
MY.NET.209.110	UDP	484	484	672
	spp_portscan: PORTSC	1	0	7
MY.NET.1.5	UDP	465	465	5887
	spp_portscan: PORTSC	1	0	301
MY.NET.203.26	UDP	465	465	465
	spp_portscan: PORTSC	1	0	1
MY.NET.207.78	SYN	1	1	2
	UDP	449	449	1169
	spp_portscan: PORTSC	1	0	17
MY.NET.217.126	FIN	16	13	36
	INVALIDACK	48	28	134
	XMAS	3	3	7
	VECNA	11	8	31
	UNKNOWN	31	19	84
	UDP	1	1	14
	SYNFIN	3	3	9
	SYN	16	15	95
	SPAU	1	1	1
	NULL	18	17	34
	NOACK	61	24	154
	spp_portscan: PORTSC	1	0	1492
	oos	229	52	1132
	FULLXMAS	5	4	16
MY.NET.98.201	UDP	427	427	2661
	spp_portscan: PORTSC	1	0	31
MY.NET.98.191	SYN	7	7	7
	spp_portscan: PORTSC	1	0	13
	UDP	404	404	1223
MY.NET.98.123	SYN	9	9	9
	UDP	401	401	414
	spp_portscan: PORTSC	1	0	14
MY.NET.202.214	UDP	407	407	479
	spp_portscan: PORTSC	1	0	4
MY.NET.97.240	SYN	8	8	16

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	UDP	396	396	1426
	spp_portscan: PORTSC	1	0	18
MY.NET.202.218	UDP	403	403	506
	spp_portscan: PORTSC	1	0	2
MY.NET.97.104	SYN	1	1	1
	UDP	402	402	501
	spp_portscan: PORTSC	1	0	2
MY.NET.98.178	SYN	11	11	11
	spp_portscan: PORTSC	1	0	15
	UDP	389	389	830
MY.NET.212.174	UDP	400	400	524
	spp_portscan: PORTSC	1	0	7
MY.NET.98.110	UDP	396	396	1038
	spp_portscan: PORTSC	1	0	36
MY.NET.201.190	SYN	394	394	521
	spp_portscan: PORTSC	1	0	5
MY.NET.201.14	SYN	8	8	8
	spp_portscan: PORTSC	1	0	10
	UDP	372	372	586
MY.NET.97.34	SYN	25	25	27
	spp_portscan: PORTSC	1	0	5
	UDP	350	350	486
MY.NET.98.198	SYN	217	217	463
	spp_portscan: PORTSC	1	0	40
	UDP	153	153	362
MY.NET.223.82	UDP	367	367	578
	spp_portscan: PORTSC	1	0	6
MY.NET.207.218	SYN	1	1	1
	UDP	362	362	928
	spp_portscan: PORTSC	1	0	13
MY.NET.1.4	UDP	350	350	6081
	spp_portscan: PORTSC	1	0	260
MY.NET.217.14	UDP	358	358	491
	spp_portscan: PORTSC	1	0	2
MY.NET.218.18	UDP	352	352	638
	spp_portscan: PORTSC	1	0	9
MY.NET.217.254	UDP	345	345	1788
	spp_portscan: PORTSC	1	0	3
MY.NET.97.224	SYN	3	3	10
	UDP	319	319	3602
	spp_portscan: PORTSC	1	0	40
MY.NET.201.94	FULLXMAS	1	1	1
	NOACK	7	7	19
	INVALIDACK	1	1	1
	NULL	4	4	7
	UDP	294	294	3015
	spp_portscan: PORTSC	1	0	39
	SYN	7	7	9
MY.NET.224.6	SYN	2	2	2
	UDP	315	315	689
	spp_portscan: PORTSC	1	0	4
MY.NET.203.218	SYN	40	40	52
	spp_portscan: PORTSC	1	0	8
	UDP	265	265	385
MY.NET.97.111	UDP	303	303	1380
	spp_portscan: PORTSC	1	0	5
MY.NET.217.250	SYN	5	5	5
	UDP	291	291	1183
	spp_portscan: PORTSC	1	0	28
MY.NET.71.38	UDP	267	267	11581
	spp_portscan: PORTSC	1	0	2
MY.NET.211.42	UDP	286	286	317
	spp_portscan: PORTSC	1	0	3
MY.NET.202.82	UDP	281	281	444
	spp_portscan: PORTSC	1	0	5
MY.NET.98.150	SYN	10	10	18
	spp_portscan: PORTSC	1	0	9
	oos	1	1	1
	UDP	268	268	502
MY.NET.70.20	UDP	274	274	948
	spp_portscan: PORTSC	1	0	1
MY.NET.224.238	UDP	272	272	474

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	10
MY.NET.203.154	UDP	270	270	669
	spp_portscan: PORTSC	1	0	11
MY.NET.217.218	SYN	3	3	6
	spp_portscan: PORTSC	1	0	6
	UDP	265	265	653
MY.NET.97.128	UDP	262	262	1074
	spp_portscan: PORTSC	1	0	15
MY.NET.210.30	UDP	263	263	282
	spp_portscan: PORTSC	1	0	5
MY.NET.223.118	UDP	262	262	283
	spp_portscan: PORTSC	1	0	1
MY.NET.209.234	UDP	261	261	290
	spp_portscan: PORTSC	1	0	2
MY.NET.201.254	SYN	3	3	3
	UDP	256	256	889
	spp_portscan: PORTSC	1	0	34
MY.NET.217.46	UDP	252	252	1344
	spp_portscan: PORTSC	1	0	21
MY.NET.217.82	SYN	1	1	1
	spp_portscan: PORTSC	1	0	3
	UDP	246	246	3598
MY.NET.220.2	UDP	252	252	691
	spp_portscan: PORTSC	1	0	5
MY.NET.212.150	SYN	1	1	13
	spp_portscan: PORTSC	1	0	24
	UDP	232	232	9402
MY.NET.75.147	SYN	248	248	464
	spp_portscan: PORTSC	1	0	59
MY.NET.221.22	SYN	5	5	8
	spp_portscan: PORTSC	1	0	22
	UDP	236	236	3967
MY.NET.97.207	UDP	243	243	903
	spp_portscan: PORTSC	1	0	14
MY.NET.156.109	UDP	238	238	2035
	spp_portscan: PORTSC	1	0	16
MY.NET.222.206	SYN	2	2	2
	spp_portscan: PORTSC	1	0	2
	UDP	235	235	964
MY.NET.217.90	SYN	235	235	296
	spp_portscan: PORTSC	1	0	8
MY.NET.220.146	UDP	232	232	251
	spp_portscan: PORTSC	1	0	1
MY.NET.202.106	UDP	231	231	279
	spp_portscan: PORTSC	1	0	7
MY.NET.223.226	SYN	229	229	519
	spp_portscan: PORTSC	1	0	11
MY.NET.202.30	SMB Name Wildcard	1	1	3
	UDP	226	226	1017
	spp_portscan: PORTSC	1	0	7
MY.NET.201.218	SYN	1	1	2
	UDP	227	227	506
	spp_portscan: PORTSC	1	0	10
MY.NET.98.128	SYN	11	11	13
	spp_portscan: PORTSC	1	0	47
	UDP	212	212	2741
MY.NET.217.138	SYN	140	140	226
	spp_portscan: PORTSC	1	0	11
	UDP	87	87	110
MY.NET.218.202	SYN	226	226	257
	spp_portscan: PORTSC	1	0	1
MY.NET.201.90	SYN	68	68	90
	UDP	154	154	603
	spp_portscan: PORTSC	1	0	24
MY.NET.70.38	NMAP TCP ping!	130	130	262
!!!!	UDP	1	1	14
	spp_portscan: PORTSC	1	0	255
	oos	1	1	1
	connect to 515 from	89	89	137
MY.NET.98.117	UDP	219	219	1363
	spp_portscan: PORTSC	1	0	21
MY.NET.98.168	SYN	135	135	863

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	UDP	83	83	445
	spp_portscan: PORTSC	1	0	9
MY.NET.97.144	SYN	1	1	1
	spp_portscan: PORTSC	1	0	17
	UDP	218	218	474
MY.NET.98.161	SYN	46	46	144
	spp_portscan: PORTSC	1	0	18
	UDP	168	168	1592
MY.NET.206.230	UDP	215	215	283
	spp_portscan: PORTSC	1	0	1
MY.NET.201.10	UDP	210	210	234
	spp_portscan: PORTSC	1	0	1
MY.NET.204.82	UDP	209	209	306
	spp_portscan: PORTSC	1	0	12
MY.NET.226.46	UDP	201	201	277
	spp_portscan: PORTSC	1	0	1
MY.NET.210.182	UDP	200	200	239
	spp_portscan: PORTSC	1	0	1
MY.NET.222.166	UDP	200	200	232
	spp_portscan: PORTSC	1	0	2
MY.NET.210.146	SYN	196	196	232
	spp_portscan: PORTSC	1	0	9
MY.NET.97.41	SYN	143	143	529
	spp_portscan: PORTSC	1	0	9
	UDP	48	48	759
MY.NET.97.176	SYN	122	122	756
	spp_portscan: PORTSC	1	0	17
	UDP	65	65	2495
MY.NET.201.206	SYN	190	190	254
	spp_portscan: PORTSC	1	0	8
MY.NET.97.172	SYN	9	9	9
	UDP	177	177	1096
	spp_portscan: PORTSC	1	0	18
MY.NET.204.54	UDP	185	185	277
	spp_portscan: PORTSC	1	0	8
MY.NET.98.140	SYN	101	101	404
	UDP	76	76	3178
	spp_portscan: PORTSC	1	0	10
	oos	1	1	6
MY.NET.253.42	SYN	183	183	242
	spp_portscan: PORTSC	1	0	17
MY.NET.98.197	SYN	11	11	12
	UDP	172	172	224
	spp_portscan: PORTSC	1	0	14
MY.NET.97.212	UDP	183	183	228
	spp_portscan: PORTSC	1	0	15
MY.NET.98.181	SYN	7	7	16
	spp_portscan: PORTSC	1	0	12
	UDP	165	165	5556
MY.NET.98.146	SYN	1	1	1
	spp_portscan: PORTSC	1	0	20
	UDP	180	180	360
MY.NET.201.2	SYN	174	174	282
	spp_portscan: PORTSC	1	0	2
	UDP	6	6	8
MY.NET.226.154	UDP	179	179	782
	spp_portscan: PORTSC	1	0	9
MY.NET.98.243	UDP	175	175	2642
	spp_portscan: PORTSC	1	0	6
MY.NET.6.35	SYN	176	176	1437
	spp_portscan: PORTSC	1	0	202
MY.NET.217.142:	spp_portscan: portsc	179	0	480
MY.NET.201.130	SYN	8	8	24
	spp_portscan: PORTSC	1	0	22
	UDP	168	168	858
MY.NET.201.42	SYN	168	168	219
	spp_portscan: PORTSC	1	0	1
	UDP	9	9	15
MY.NET.210.206	UDP	176	176	239
	spp_portscan: PORTSC	1	0	2
MY.NET.217.202	UDP	174	174	518
	spp_portscan: PORTSC	1	0	33

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.253.43	SYN	173	173	640
	spp_portscan: PORTSC	1	0	14
MY.NET.98.196	SYN	7	7	14
	UDP	163	163	220
	spp_portscan: PORTSC	1	0	11
MY.NET.218.10	UDP	167	167	594
	spp_portscan: PORTSC	1	0	11
MY.NET.225.26	UDP	167	167	301
	spp_portscan: PORTSC	1	0	7
MY.NET.206.74	UDP	165	165	453
	spp_portscan: PORTSC	1	0	11
MY.NET.208.70	UDP	164	164	183
	spp_portscan: PORTSC	1	0	15
MY.NET.98.147	SYN	2	2	4
	UDP	161	161	249
	spp_portscan: PORTSC	1	0	12
MY.NET.201.34	UDP	162	162	330
	spp_portscan: PORTSC	1	0	3
MY.NET.201.78	SYN	148	148	227
	spp_portscan: PORTSC	1	0	3
	UDP	12	12	25
MY.NET.98.160	SYN	2	2	2
	spp_portscan: PORTSC	1	0	14
	UDP	153	153	170
MY.NET.202.54	SYN	1	1	2
	spp_portscan: PORTSC	1	0	7
	UDP	152	152	176
MY.NET.97.206	SNMP public access	1	1	7
	UDP	49	49	846
	SYN	97	97	660
	spp_portscan: PORTSC	1	0	9
MY.NET.217.50	UDP	149	149	195
	spp_portscan: PORTSC	1	0	12
MY.NET.217.94:	spp_portscan: portsc	148	0	217
MY.NET.97.246	UDP	147	147	402
MY.NET.201.118	SYN	3	3	4
	spp_portscan: PORTSC	1	0	16
	UDP	143	143	335
MY.NET.98.131	UDP	145	145	662
	spp_portscan: PORTSC	1	0	20
MY.NET.219.118	UDP	144	144	564
	spp_portscan: PORTSC	1	0	22
MY.NET.97.186	SNMP public access	1	1	13
	UDP	140	140	833
	spp_portscan: PORTSC	1	0	15
MY.NET.97.57	UDP	142	142	184
	spp_portscan: PORTSC	1	0	1
MY.NET.98.144	SYN	1	1	1
	oos	1	1	1
	UDP	138	138	174
	spp_portscan: PORTSC	1	0	8
MY.NET.211.90	UDP	137	137	153
	spp_portscan: PORTSC	1	0	3
MY.NET.97.193	SYN	11	11	11
	UDP	122	122	364
	spp_portscan: PORTSC	1	0	11
MY.NET.217.150:	spp_portscan: portsc	112	0	9999
MY.NET.217.110	UDP	129	129	700
	spp_portscan: PORTSC	1	0	15
MY.NET.97.121	SYN	1	1	10
	spp_portscan: PORTSC	1	0	5
	UDP	128	128	225
MY.NET.98.145	UDP	126	126	154
	spp_portscan: PORTSC	1	0	59
MY.NET.201.62	SYN	119	119	152
	UDP	7	7	10
	spp_portscan: PORTSC	1	0	6
MY.NET.160.109	UDP	124	124	950
	spp_portscan: PORTSC	1	0	15
MY.NET.253.41	SYN	125	125	184
	spp_portscan: PORTSC	1	0	11
MY.NET.60.11:	spp_portscan: portsc	123	0	290

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.98.210	UDP	121	121	376
	spp_portscan: PORTSC	1	0	8
MY.NET.6.34	SYN	119	119	861
	spp_portscan: PORTSC	1	0	165
MY.NET.224.150	UDP	117	117	166
	spp_portscan: PORTSC	1	0	5
MY.NET.98.153	UDP	112	112	2572
	spp_portscan: PORTSC	1	0	8
MY.NET.205.162	UDP	116	116	141
	spp_portscan: PORTSC	1	0	3
MY.NET.140.9	UDP	113	113	1356
	spp_portscan: PORTSC	1	0	35
MY.NET.98.200:	spp_portscan: portsc	115	0	496
MY.NET.98.199	SYN	7	7	17
	UDP	107	107	172
	spp_portscan: PORTSC	1	0	39
MY.NET.98.132	SYN	1	1	7
	UDP	111	111	647
	spp_portscan: PORTSC	1	0	13
MY.NET.97.234	SYN	4	4	4
	UDP	90	90	9097
	spp_portscan: PORTSC	1	0	10
MY.NET.98.171	SYN	3	3	3
	UDP	108	108	157
	spp_portscan: PORTSC	1	0	22
MY.NET.218.78	UDP	109	109	319
	spp_portscan: PORTSC	1	0	3
MY.NET.202.10	SYN	5	5	6
	spp_portscan: PORTSC	1	0	15
	oos	1	1	1
	SYNFIN	1	1	1
	UDP	100	100	270
MY.NET.98.130	SYN	83	83	133
	UDP	24	24	61
	spp_portscan: PORTSC	1	0	14
MY.NET.98.201:	spp_portscan: portsc	105	0	1048
MY.NET.6.7	SYN	62	62	314
	spp_portscan: PORTSC	1	0	57
	UDP	41	41	139
MY.NET.210.230	UDP	102	102	124
	spp_portscan: PORTSC	1	0	3
MY.NET.201.46:	spp_portscan: portsc	97	0	1836
MY.NET.98.183	SNMP public access	1	1	3
	spp_portscan: PORTSC	1	0	11
	SYN	2	2	5
	UDP	94	94	137
MY.NET.219.46	UDP	96	96	107
	spp_portscan: PORTSC	1	0	1
MY.NET.201.74	SYN	41	41	49
	UDP	54	54	103
	spp_portscan: PORTSC	1	0	33
MY.NET.209.218	UDP	95	95	122
	spp_portscan: PORTSC	1	0	1
MY.NET.98.119	SYN	2	2	7
	UDP	90	90	1332
	spp_portscan: PORTSC	1	0	13
MY.NET.98.113	SYN	2	2	10
	UDP	88	88	1987
	spp_portscan: PORTSC	1	0	5
MY.NET.218.214	UDP	93	93	136
	spp_portscan: PORTSC	1	0	1
MY.NET.218.130:	spp_portscan: portsc	84	0	4988
MY.NET.97.168	SNMP public access	1	1	33
	spp_portscan: PORTSC	1	0	3
	UDP	90	90	269
	SYN	1	1	1
MY.NET.97.179	SYN	1	1	20
	UDP	92	92	247
MY.NET.6.47	SYN	89	89	979
	spp_portscan: PORTSC	1	0	183
MY.NET.201.54:	spp_portscan: portsc	92	0	119
MY.NET.98.165	UDP	91	91	94

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	20
MY.NET.97.15	SYN	1	1	2
	UDP	88	88	449
	spp_portscan: PORTSC	1	0	7
MY.NET.203.226	UDP	89	89	276
	spp_portscan: PORTSC	1	0	1
MY.NET.98.118	SYN	1	1	1
	UDP	87	87	98
	spp_portscan: PORTSC	1	0	8
MY.NET.98.112	SYN	7	7	7
	spp_portscan: PORTSC	1	0	3
	UDP	80	80	249
MY.NET.97.87	UDP	87	87	97
	spp_portscan: PORTSC	1	0	12
MY.NET.97.216	UDP	87	87	89
	spp_portscan: PORTSC	1	0	1
MY.NET.97.42	UDP	85	85	602
	spp_portscan: PORTSC	1	0	6
MY.NET.6.45	UDP	82	82	1592
	spp_portscan: PORTSC	1	0	196
MY.NET.140.21	UDP	56	56	13684
	spp_portscan: PORTSC	1	0	498
MY.NET.219.166:	spp_portscan: portsc	82	0	214
MY.NET.217.58:	spp_portscan: portsc	82	0	171
MY.NET.98.124	SYN	1	1	3
	spp_portscan: PORTSC	1	0	1
	UDP	79	79	240
MY.NET.218.106	SYN	79	79	157
	spp_portscan: PORTSC	1	0	5
MY.NET.228.18	UDP	79	79	226
MY.NET.217.54:	spp_portscan: portsc	79	0	155
MY.NET.217.118:	spp_portscan: portsc	79	0	117
MY.NET.213.10	UDP	77	77	97
	spp_portscan: PORTSC	1	0	21
MY.NET.204.130	SYN	1	1	1
	UDP	76	76	90
	spp_portscan: PORTSC	1	0	15
MY.NET.206.26	UDP	75	75	327
	spp_portscan: PORTSC	1	0	2
MY.NET.217.66	SYN	75	75	101
	spp_portscan: PORTSC	1	0	2
MY.NET.224.34	UDP	75	75	79
	spp_portscan: PORTSC	1	0	1
MY.NET.98.199:	spp_portscan: portsc	74	0	511
MY.NET.206.186:	spp_portscan: portsc	74	0	450
MY.NET.98.205	SYN	8	8	26
	spp_portscan: PORTSC	1	0	3
	UDP	65	65	340
MY.NET.97.93:	spp_portscan: portsc	73	0	100
MY.NET.60.11	SYN	5	5	560
	spp_portscan: PORTSC	1	0	71
	UDP	64	64	851
MY.NET.97.67	SYN	2	2	2
	UDP	69	69	218
	spp_portscan: PORTSC	1	0	8
MY.NET.60.43	UDP	68	68	1300
	spp_portscan: PORTSC	1	0	151
MY.NET.97.164	UDP	70	70	221
	spp_portscan: PORTSC	1	0	1
MY.NET.206.78	UDP	70	70	89
	spp_portscan: PORTSC	1	0	9
MY.NET.201.202	UDP	66	66	1602
	spp_portscan: PORTSC	1	0	12
MY.NET.100.230:	spp_portscan: portsc	46	0	11891
MY.NET.209.42	spp_portscan: PORTSC	1	0	17
	UDP	66	66	1374
MY.NET.98.107	UDP	67	67	606
	spp_portscan: PORTSC	1	0	12
MY.NET.97.171	SYN	7	7	7
	UDP	62	62	100
MY.NET.98.122	SYN	9	9	9
	spp_portscan: PORTSC	1	0	27

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	oos	4	4	5
	UDP	54	54	71
MY.NET.210.166	UDP	66	66	90
	spp_portscan: PORTSC	1	0	1
MY.NET.208.218	UDP	63	63	1314
	spp_portscan: PORTSC	1	0	5
MY.NET.210.70:	spp_portscan: portsc	66	0	125
MY.NET.218.62	UDP	64	64	267
	spp_portscan: PORTSC	1	0	1
MY.NET.201.250	SYN	3	3	4
	spp_portscan: PORTSC	1	0	18
	UDP	61	61	233
MY.NET.209.114:	spp_portscan: portsc	65	0	134
MY.NET.97.112	SYN	10	10	11
	spp_portscan: PORTSC	1	0	10
	UDP	54	54	102
MY.NET.217.242	UDP	63	63	362
	spp_portscan: PORTSC	1	0	23
MY.NET.212.34:	spp_portscan: portsc	64	0	210
MY.NET.201.170	UDP	63	63	201
	spp_portscan: PORTSC	1	0	2
MY.NET.224.102:	spp_portscan: portsc	63	0	139
MY.NET.219.198	UDP	62	62	110
	spp_portscan: PORTSC	1	0	7
MY.NET.214.166:	spp_portscan: portsc	14	0	24235
MY.NET.201.6	SYN	1	1	1
	UDP	58	58	374
	spp_portscan: PORTSC	1	0	9
MY.NET.97.230	SYN	8	8	9
	UDP	51	51	66
	spp_portscan: PORTSC	1	0	4
MY.NET.75.155	SYN	59	59	65
	spp_portscan: PORTSC	1	0	5
MY.NET.98.138	SYN	7	7	7
	UDP	51	51	167
	spp_portscan: PORTSC	1	0	6
MY.NET.98.115	UDP	58	58	135
	spp_portscan: PORTSC	1	0	6
MY.NET.202.18	SYN	4	4	5
	UDP	54	54	121
	spp_portscan: PORTSC	1	0	6
MY.NET.97.64	SYN	2	2	3
	spp_portscan: PORTSC	1	0	6
	UDP	56	56	97
MY.NET.218.158:	spp_portscan: portsc	56	0	1204
MY.NET.202.142	SYN	21	21	25
	spp_portscan: PORTSC	1	0	4
	UDP	36	36	42
MY.NET.201.98	UDP	57	57	67
	spp_portscan: PORTSC	1	0	3
MY.NET.202.26	SYN	9	9	9
	UDP	48	48	48
	spp_portscan: PORTSC	1	0	13
MY.NET.219.18:	spp_portscan: portsc	57	0	165
MY.NET.98.156	SYN	22	22	28
	UDP	30	30	72
	UNKNOWN	1	1	15
	spp_portscan: PORTSC	1	0	31
	oos	3	3	3
MY.NET.97.146	SYN	2	2	3
	spp_portscan: PORTSC	1	0	8
	UDP	54	54	107
MY.NET.204.130:	spp_portscan: portsc	57	0	106
MY.NET.217.106:	spp_portscan: portsc	53	0	1757
MY.NET.208.194:	spp_portscan: portsc	56	0	130
MY.NET.98.119:	spp_portscan: portsc	54	0	114
MY.NET.98.111	SYN	8	8	8
	UDP	44	44	133
	spp_portscan: PORTSC	1	0	3
MY.NET.97.62	UDP	52	52	92
	spp_portscan: PORTSC	1	0	10
MY.NET.209.162	FIN	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	SYN	42	42	44
	VECNA	2	1	2
	spp_portscan: PORTSC	1	0	11
	UNKNOWN	1	1	1
	NULL	1	1	1
	INVALIDACK	2	2	2
	NOACK	3	3	3
MY.NET.140.21:	spp_portscan: portsc	51	0	787
MY.NET.98.194:	spp_portscan: portsc	52	0	228
MY.NET.98.116	SYN	3	3	11
	UDP	48	48	121
	spp_portscan: PORTSC	1	0	4
MY.NET.201.234:	spp_portscan: portsc	52	0	108
MY.NET.98.155	SYN	9	9	9
	UDP	41	41	185
	spp_portscan: PORTSC	1	0	11
MY.NET.97.217	UDP	50	50	112
	spp_portscan: PORTSC	1	0	3
MY.NET.98.189:	spp_portscan: portsc	51	0	106
MY.NET.205.254:	spp_portscan: portsc	51	0	66
MY.NET.1.3:	spp_portscan: portsc	47	0	1879
MY.NET.98.136:	spp_portscan: portsc	50	0	365
MY.NET.218.102:	spp_portscan: portsc	48	0	1124
MY.NET.215.242:	spp_portscan: portsc	50	0	99
MY.NET.60.8	SYN	3	3	24
	UDP	45	45	312
	spp_portscan: PORTSC	1	0	50
MY.NET.213.186:	spp_portscan: portsc	36	0	6869
MY.NET.217.246	UDP	48	48	190
	spp_portscan: PORTSC	1	0	4
MY.NET.208.66	UDP	48	48	174
	spp_portscan: PORTSC	1	0	2
MY.NET.100.158	UDP	48	48	55
	spp_portscan: PORTSC	1	0	4
MY.NET.97.44:	spp_portscan: portsc	49	0	58
MY.NET.202.78	SYN	40	40	43
	UDP	8	8	10
	spp_portscan: PORTSC	1	0	4
MY.NET.98.167	SYN	3	3	3
	UDP	44	44	77
	spp_portscan: PORTSC	1	0	4
MY.NET.97.65	SYN	1	1	1
	UDP	46	46	49
	spp_portscan: PORTSC	1	0	2
MY.NET.97.188	UDP	47	47	47
	spp_portscan: PORTSC	1	0	4
MY.NET.210.194:	spp_portscan: portsc	47	0	407
MY.NET.153.243	SYN	1	1	1
	spp_portscan: PORTSC	1	0	9
	UDP	45	45	216
MY.NET.97.106	UDP	46	46	148
	spp_portscan: PORTSC	1	0	3
MY.NET.219.210	UDP	46	46	108
	spp_portscan: PORTSC	1	0	3
MY.NET.203.254	UDP	46	46	91
	spp_portscan: PORTSC	1	0	2
MY.NET.97.170	SYN	34	34	39
	spp_portscan: PORTSC	1	0	4
	UDP	12	12	41
MY.NET.98.114	SYN	46	46	72
	spp_portscan: PORTSC	1	0	3
MY.NET.204.58	UDP	46	46	55
	spp_portscan: PORTSC	1	0	2
MY.NET.140.143	UDP	46	46	49
	spp_portscan: PORTSC	1	0	5
MY.NET.1.4:	spp_portscan: portsc	43	0	1267
MY.NET.98.139	UDP	44	44	158
	spp_portscan: PORTSC	1	0	2
MY.NET.1.5:	spp_portscan: portsc	42	0	1519
MY.NET.215.222	SYN	43	43	494
	spp_portscan: PORTSC	1	0	24
MY.NET.203.194:	spp_portscan: portsc	44	0	248

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.225.142:	spp_portscan: portsc	44	0	239
MY.NET.97.186:	spp_portscan: portsc	44	0	195
MY.NET.160.109:	spp_portscan: portsc	43	0	500
MY.NET.217.82:	spp_portscan: portsc	43	0	149
MY.NET.217.158:	spp_portscan: portsc	31	0	6094
MY.NET.206.14:	spp_portscan: portsc	43	0	76
MY.NET.98.106:	spp_portscan: portsc	42	0	463
MY.NET.217.250:	spp_portscan: portsc	41	0	914
MY.NET.201.214:	spp_portscan: portsc	42	0	134
MY.NET.53.60	UDP	41	41	85
	spp_portscan: PORTSC	1	0	5
MY.NET.207.122	UDP	41	41	44
	spp_portscan: PORTSC	1	0	1
MY.NET.201.206:	spp_portscan: portsc	41	0	205
MY.NET.217.202:	spp_portscan: portsc	41	0	97
MY.NET.97.59:	spp_portscan: portsc	41	0	49
MY.NET.207.242	UDP	40	40	41
	spp_portscan: PORTSC	1	0	1
MY.NET.97.103	SYN	39	39	50
	spp_portscan: PORTSC	1	0	4
MY.NET.97.215	UDP	39	39	48
	spp_portscan: PORTSC	1	0	4
MY.NET.97.184	UDP	39	39	40
	spp_portscan: PORTSC	1	0	5
MY.NET.220.50:	spp_portscan: portsc	39	0	127
MY.NET.223.14	SYN	38	38	94
	spp_portscan: PORTSC	1	0	3
MY.NET.53.184	SYN	31	31	42
	UDP	7	7	13
	spp_portscan: PORTSC	1	0	1
MY.NET.226.150	UDP	38	38	46
	spp_portscan: PORTSC	1	0	1
MY.NET.222.22	SYN	35	35	378
	spp_portscan: PORTSC	1	0	12
	UDP	2	2	15
MY.NET.209.170:	spp_portscan: portsc	38	0	244
MY.NET.97.224:	spp_portscan: portsc	37	0	687
MY.NET.202.74:	spp_portscan: portsc	38	0	165
MY.NET.210.82:	spp_portscan: portsc	38	0	62
MY.NET.97.157	UDP	37	37	46
	spp_portscan: PORTSC	1	0	1
MY.NET.97.227	SYN	1	1	1
	oos	1	1	2
	UDP	35	35	35
	spp_portscan: PORTSC	1	0	1
MY.NET.206.90	UDP	37	37	37
	spp_portscan: PORTSC	1	0	2
MY.NET.98.175	UDP	37	37	37
	spp_portscan: PORTSC	1	0	1
MY.NET.220.58:	spp_portscan: portsc	37	0	245
MY.NET.217.102:	spp_portscan: portsc	37	0	88
MY.NET.211.162	SYN	36	36	65
	spp_portscan: PORTSC	1	0	7
MY.NET.201.14:	spp_portscan: portsc	37	0	62
MY.NET.98.186:	spp_portscan: portsc	37	0	40
MY.NET.60.16	UDP	24	24	5430
	spp_portscan: PORTSC	1	0	7
	connect to 515 from	1	1	1
MY.NET.97.89:	spp_portscan: portsc	36	0	90
MY.NET.97.86	UDP	35	35	43
	spp_portscan: PORTSC	1	0	2
MY.NET.201.134	UDP	35	35	37
	spp_portscan: PORTSC	1	0	4
MY.NET.202.6:	spp_portscan: portsc	33	0	1536
MY.NET.219.102	UDP	34	34	268
	spp_portscan: PORTSC	1	0	9
MY.NET.98.179	SYN	5	5	8
	spp_portscan: PORTSC	1	0	18
	UDP	29	29	58
MY.NET.60.17	SYN	33	33	59
	spp_portscan: PORTSC	1	0	7
MY.NET.98.149	UDP	33	33	41

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	23
MY.NET.98.184	SYN	10	10	10
	spp_portscan: PORTSC	1	0	6
	UDP	23	23	43
MY.NET.213.10:	spp_portscan: portsc	34	0	48
MY.NET.98.212:	spp_portscan: portsc	34	0	47
MY.NET.97.111:	spp_portscan: portsc	34	0	44
MY.NET.207.78:	spp_portscan: portsc	33	0	105
MY.NET.217.46:	spp_portscan: portsc	33	0	85
MY.NET.97.31	SYN	2	2	3
	spp_portscan: PORTSC	1	0	4
	UDP	30	30	71
MY.NET.212.174:	spp_portscan: portsc	33	0	62
MY.NET.97.173	SYN	32	32	44
	spp_portscan: PORTSC	1	0	2
MY.NET.226.174:	spp_portscan: portsc	33	0	41
MY.NET.97.27	UDP	31	31	45
	spp_portscan: PORTSC	1	0	2
MY.NET.156.110:	spp_portscan: portsc	24	0	3788
MY.NET.202.226:	spp_portscan: portsc	31	0	65
MY.NET.219.214	NOACK	1	1	1
	SYN	28	28	46
	oos	1	1	1
	spp_portscan: PORTSC	1	0	2
MY.NET.97.45	SYN	7	7	7
	UDP	23	23	23
	spp_portscan: PORTSC	1	0	3
MY.NET.98.176:	spp_portscan: portsc	30	0	458
MY.NET.202.34:	spp_portscan: portsc	30	0	347
MY.NET.207.110	UDP	23	23	3245
	spp_portscan: PORTSC	1	0	11
MY.NET.97.17	SYN	8	8	8
	UDP	21	21	122
	spp_portscan: PORTSC	1	0	3
MY.NET.218.238	SYN	29	29	108
	spp_portscan: PORTSC	1	0	3
MY.NET.207.50	UDP	29	29	43
	spp_portscan: PORTSC	1	0	4
MY.NET.217.186:	spp_portscan: portsc	30	0	42
MY.NET.209.6:	spp_portscan: portsc	30	0	39
MY.NET.97.19:	spp_portscan: portsc	30	0	39
MY.NET.97.183:	spp_portscan: portsc	30	0	37
MY.NET.208.90:	spp_portscan: portsc	30	0	33
MY.NET.207.54	UDP	29	29	30
	spp_portscan: PORTSC	1	0	1
MY.NET.253.24:	spp_portscan: portsc	1	0	14327
MY.NET.97.196:	spp_portscan: portsc	29	0	213
MY.NET.201.210:	spp_portscan: portsc	29	0	206
MY.NET.219.234	UDP	28	28	37
	spp_portscan: PORTSC	1	0	1
MY.NET.219.22	UDP	28	28	28
MY.NET.97.241	spp_portscan: PORTSC	1	0	1
	UDP	28	28	29
MY.NET.219.22	spp_portscan: PORTSC	1	0	2
MY.NET.60.38	SYN	4	4	170
	UDP	21	21	658
	spp_portscan: PORTSC	1	0	18
	connect to 515 from	1	1	3
MY.NET.201.86:	spp_portscan: portsc	26	0	1212
MY.NET.97.74	UDP	27	27	60
	spp_portscan: PORTSC	1	0	3
MY.NET.209.2:	spp_portscan: portsc	28	0	48
MY.NET.217.162	UDP	27	27	30
	spp_portscan: PORTSC	1	0	2
MY.NET.201.246	UDP	27	27	27
	spp_portscan: PORTSC	1	0	2
MY.NET.97.194	SYN	2	2	3
	UDP	24	24	133
	spp_portscan: PORTSC	1	0	5
MY.NET.217.130	SYN	14	14	21
	UDP	12	12	20
	spp_portscan: PORTSC	1	0	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.217.146:	spp_portscan: portsc	27	0	34
MY.NET.97.199:	spp_portscan: portsc	27	0	34
MY.NET.209.90:	spp_portscan: portsc	27	0	33
MY.NET.98.125:	spp_portscan: portsc	25	0	941
MY.NET.202.254:	spp_portscan: portsc	26	0	293
MY.NET.208.142:	spp_portscan: portsc	26	0	107
MY.NET.203.250:	spp_portscan: portsc	26	0	92
MY.NET.219.118:	spp_portscan: portsc	26	0	64
MY.NET.202.2:	spp_portscan: portsc	26	0	61
MY.NET.218.222:	spp_portscan: portsc	26	0	51
MY.NET.202.46	SYN	16	16	19
	UDP	2	2	2
	oos	7	7	7
	spp_portscan: PORTSC	1	0	12
MY.NET.218.210:	spp_portscan: portsc	26	0	35
MY.NET.70.42:	spp_portscan: portsc	26	0	34
MY.NET.98.195:	spp_portscan: portsc	26	0	31
MY.NET.97.148:	spp_portscan: portsc	26	0	30
MY.NET.201.242:	spp_portscan: portsc	23	0	1368
MY.NET.98.165:	spp_portscan: portsc	25	0	349
MY.NET.97.112:	spp_portscan: portsc	25	0	337
MY.NET.98.164:	spp_portscan: portsc	23	0	1305
MY.NET.201.50:	spp_portscan: portsc	22	0	1748
MY.NET.98.123:	spp_portscan: portsc	25	0	199
MY.NET.217.70	UDP	24	24	132
	spp_portscan: PORTSC	1	0	5
MY.NET.1.2	SYN	24	24	106
	spp_portscan: PORTSC	1	0	15
MY.NET.140.9:	spp_portscan: portsc	25	0	118
MY.NET.217.218:	spp_portscan: portsc	25	0	73
MY.NET.97.220	SYN	1	1	1
	UDP	23	23	33
	spp_portscan: PORTSC	1	0	28
MY.NET.209.110:	spp_portscan: portsc	25	0	55
MY.NET.220.2:	spp_portscan: portsc	25	0	45
MY.NET.97.161:	spp_portscan: portsc	25	0	36
MY.NET.98.170	UDP	24	24	27
	spp_portscan: PORTSC	1	0	1
MY.NET.98.191:	spp_portscan: portsc	24	0	374
MY.NET.97.198:	spp_portscan: portsc	23	0	820
MY.NET.97.247:	spp_portscan: portsc	22	0	1307
MY.NET.98.154:	spp_portscan: portsc	24	0	288
MY.NET.220.162:	spp_portscan: portsc	24	0	283
MY.NET.97.20:	spp_portscan: portsc	24	0	207
MY.NET.98.152	SYN	3	3	40
	spp_portscan: PORTSC	1	0	40
	UDP	19	19	58
	oos	1	1	56
MY.NET.97.110	UDP	23	23	134
	spp_portscan: PORTSC	1	0	13
MY.NET.97.10:	spp_portscan: portsc	24	0	92
MY.NET.97.177	SYN	2	2	5
	UDP	22	22	72
MY.NET.212.54	UDP	23	23	68
	spp_portscan: PORTSC	1	0	2
MY.NET.98.213	SYN	2	2	2
	spp_portscan: PORTSC	1	0	5
	UDP	21	21	59
MY.NET.201.66:	spp_portscan: portsc	24	0	63
MY.NET.201.150:	spp_portscan: portsc	24	0	60
MY.NET.201.126	SYN	11	11	15
	UDP	12	12	13
	spp_portscan: PORTSC	1	0	6
MY.NET.211.158:	spp_portscan: portsc	24	0	32
MY.NET.220.210:	spp_portscan: portsc	24	0	32
MY.NET.75.148	SYN	23	23	26
	spp_portscan: PORTSC	1	0	2
MY.NET.97.208:	spp_portscan: portsc	21	0	1382
MY.NET.201.94:	spp_portscan: portsc	22	0	873
MY.NET.202.94:	spp_portscan: portsc	10	0	6863
MY.NET.205.222:	spp_portscan: portsc	23	0	150
MY.NET.221.34	UDP	22	22	108

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	1
MY.NET.205.30:	spp_portscan: portsc	23	0	105
MY.NET.202.130:	spp_portscan: portsc	23	0	90
MY.NET.217.110:	spp_portscan: portsc	23	0	57
MY.NET.218.10:	spp_portscan: portsc	23	0	45
MY.NET.219.206	SYN	22	22	38
	spp_portscan: PORTSC	1	0	2
MY.NET.218.78:	spp_portscan: portsc	23	0	34
MY.NET.98.193	UDP	22	22	22
	spp_portscan: PORTSC	1	0	5
MY.NET.97.190:	spp_portscan: portsc	22	0	260
MY.NET.223.78:	spp_portscan: portsc	22	0	150
MY.NET.207.250:	spp_portscan: portsc	22	0	124
MY.NET.217.242:	spp_portscan: portsc	22	0	82
MY.NET.224.6:	spp_portscan: portsc	22	0	79
MY.NET.98.202:	spp_portscan: portsc	22	0	52
MY.NET.204.82:	spp_portscan: portsc	22	0	47
MY.NET.203.154:	spp_portscan: portsc	22	0	32
MY.NET.97.163	SNMP public access	1	1	5
	UDP	20	20	20
	spp_portscan: PORTSC	1	0	4
MY.NET.181.116	SYN	21	21	21
	spp_portscan: PORTSC	1	0	5
MY.NET.98.145:	spp_portscan: portsc	20	0	753
MY.NET.98.177:	spp_portscan: portsc	21	0	213
MY.NET.202.38:	spp_portscan: portsc	20	0	685
MY.NET.208.70:	spp_portscan: portsc	21	0	118
MY.NET.97.249	SNMP public access	1	1	3
	spp_portscan: PORTSC	1	0	1
	UDP	19	19	90
MY.NET.98.110:	spp_portscan: portsc	20	0	568
MY.NET.97.34:	spp_portscan: portsc	21	0	43
MY.NET.202.30:	spp_portscan: portsc	21	0	31
MY.NET.227.82	SYN	20	20	20
MY.NET.97.21	spp_portscan: PORTSC	1	0	2
	UDP	20	20	20
MY.NET.97.158	UDP	20	20	20
	spp_portscan: PORTSC	1	0	2
MY.NET.227.82	spp_portscan: PORTSC	1	0	2
MY.NET.97.165:	spp_portscan: portsc	18	0	1440
MY.NET.201.254:	spp_portscan: portsc	19	0	753
MY.NET.98.180:	spp_portscan: portsc	19	0	613
MY.NET.6.39	UDP	19	19	64
	spp_portscan: PORTSC	1	0	9
MY.NET.202.214:	spp_portscan: portsc	20	0	57
MY.NET.201.6:	spp_portscan: portsc	20	0	48
MY.NET.70.225:	spp_portscan: portsc	20	0	32
MY.NET.98.157:	spp_portscan: portsc	20	0	29
MY.NET.218.170:	spp_portscan: portsc	20	0	27
MY.NET.97.13	SYN	1	1	1
	UDP	18	18	18
	spp_portscan: PORTSC	1	0	5
MY.NET.208.182	UDP	19	19	21
	spp_portscan: PORTSC	1	0	1
MY.NET.201.158:	spp_portscan: portsc	18	0	892
MY.NET.97.240:	spp_portscan: portsc	19	0	335
MY.NET.98.178:	spp_portscan: portsc	19	0	206
MY.NET.201.90:	spp_portscan: portsc	18	0	673
MY.NET.71.38:	spp_portscan: portsc	18	0	600
MY.NET.201.66	UDP	18	18	57
	spp_portscan: PORTSC	1	0	6
MY.NET.97.229	UDP	18	18	54
	spp_portscan: PORTSC	1	0	7
MY.NET.97.219	SYN	1	1	1
	UDP	17	17	55
	spp_portscan: PORTSC	1	0	4
MY.NET.98.122:	spp_portscan: portsc	19	0	52
MY.NET.201.74:	spp_portscan: portsc	19	0	51
MY.NET.217.254:	spp_portscan: portsc	19	0	48
MY.NET.98.120	UDP	18	18	33
	spp_portscan: PORTSC	1	0	2
MY.NET.220.42	UDP	18	18	28

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	spp_portscan: PORTSC	1	0	5
MY.NET.217.50:	spp_portscan: portsc	19	0	30
MY.NET.224.238:	spp_portscan: portsc	19	0	29
MY.NET.226.26	UDP	18	18	24
	spp_portscan: PORTSC	1	0	2
MY.NET.97.229:	spp_portscan: portsc	19	0	26
MY.NET.207.218:	spp_portscan: portsc	19	0	23
MY.NET.60.182	UDP	18	18	21
	spp_portscan: PORTSC	1	0	1
MY.NET.225.194:	spp_portscan: portsc	19	0	21
MY.NET.221.22:	spp_portscan: portsc	18	0	500
MY.NET.212.150:	spp_portscan: portsc	16	0	1280
MY.NET.98.198:	spp_portscan: portsc	18	0	235
MY.NET.218.18:	spp_portscan: portsc	18	0	155
MY.NET.97.234:	spp_portscan: portsc	16	0	1147
MY.NET.229.30:	spp_portscan: portsc	18	0	112
MY.NET.208.198	UDP	17	17	70
	spp_portscan: PORTSC	1	0	1
MY.NET.226.234:	spp_portscan: portsc	18	0	42
MY.NET.98.183:	spp_portscan: portsc	18	0	38
MY.NET.195.10	UDP	17	17	28
	spp_portscan: PORTSC	1	0	2
MY.NET.97.104:	spp_portscan: portsc	18	0	26
MY.NET.202.106:	spp_portscan: portsc	18	0	25
MY.NET.217.194:	spp_portscan: portsc	18	0	24
MY.NET.98.150:	spp_portscan: portsc	18	0	23
MY.NET.98.188:	spp_portscan: portsc	18	0	19
MY.NET.224.98	UDP	17	17	17
	spp_portscan: PORTSC	1	0	1
MY.NET.98.151:	spp_portscan: portsc	17	0	511
MY.NET.98.117:	spp_portscan: portsc	17	0	440
MY.NET.98.132:	spp_portscan: portsc	17	0	390
MY.NET.98.128:	spp_portscan: portsc	17	0	330
MY.NET.98.131:	spp_portscan: portsc	17	0	325
MY.NET.98.162:	spp_portscan: portsc	16	0	742
MY.NET.209.42:	spp_portscan: portsc	16	0	686
MY.NET.201.130:	spp_portscan: portsc	17	0	161
MY.NET.98.144:	spp_portscan: portsc	17	0	97
MY.NET.97.144:	spp_portscan: portsc	17	0	76
MY.NET.205.154	SYN	16	16	54
	spp_portscan: PORTSC	1	0	2
MY.NET.201.250:	spp_portscan: portsc	17	0	52
MY.NET.97.120	SYN	2	2	4
	spp_portscan: PORTSC	1	0	3
	UDP	14	14	36
MY.NET.98.158:	spp_portscan: portsc	17	0	22
MY.NET.201.82	UDP	16	16	20
	spp_portscan: PORTSC	1	0	1
MY.NET.209.250:	spp_portscan: portsc	17	0	19
MY.NET.218.6:	spp_portscan: portsc	17	0	17
MY.NET.97.197	UDP	17	17	17
MY.NET.97.154	SYN	2	2	2
	spp_portscan: PORTSC	1	0	43
	UDP	12	12	899
MY.NET.97.172:	spp_portscan: portsc	16	0	372
MY.NET.201.218:	spp_portscan: portsc	16	0	344
MY.NET.1.8:	spp_portscan: portsc	16	0	317
MY.NET.97.128:	spp_portscan: portsc	16	0	311
MY.NET.210.30:	spp_portscan: portsc	16	0	249
MY.NET.97.212:	spp_portscan: portsc	16	0	179
MY.NET.226.154:	spp_portscan: portsc	16	0	160
MY.NET.98.181:	spp_portscan: portsc	15	0	651
MY.NET.70.163:	spp_portscan: portsc	16	0	110
MY.NET.97.207:	spp_portscan: portsc	16	0	91
MY.NET.97.220:	spp_portscan: portsc	16	0	90
MY.NET.206.74:	spp_portscan: portsc	16	0	67
MY.NET.201.118:	spp_portscan: portsc	16	0	55
MY.NET.98.171:	spp_portscan: portsc	16	0	42
MY.NET.98.109	UDP	15	15	15
MY.NET.98.184:	spp_portscan: portsc	16	0	21
MY.NET.98.109	spp_portscan: PORTSC	1	0	6
MY.NET.217.14:	spp_portscan: portsc	16	0	19

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.97.71:	spp_portscan: portsc	16	0	18
MY.NET.156.109:	spp_portscan: portsc	15	0	480
MY.NET.98.161:	spp_portscan: portsc	15	0	396
MY.NET.97.87:	spp_portscan: portsc	15	0	250
MY.NET.98.243:	spp_portscan: portsc	14	0	579
MY.NET.98.196:	spp_portscan: portsc	15	0	44
MY.NET.204.54:	spp_portscan: portsc	15	0	36
MY.NET.98.147:	spp_portscan: portsc	15	0	27
MY.NET.223.246:	spp_portscan: portsc	15	0	23
MY.NET.226.46:	spp_portscan: portsc	15	0	21
MY.NET.70.20:	spp_portscan: portsc	15	0	21
MY.NET.206.230:	spp_portscan: portsc	15	0	21
MY.NET.202.218:	spp_portscan: portsc	15	0	20
MY.NET.179.71	SYN	14	14	16
	spp_portscan: PORTSC	1	0	2
MY.NET.203.26:	spp_portscan: portsc	15	0	16
MY.NET.98.141:	spp_portscan: portsc	15	0	16
MY.NET.223.86	UDP	14	14	14
MY.NET.97.242:	spp_portscan: portsc	15	0	15
MY.NET.97.14:	spp_portscan: portsc	15	0	15
MY.NET.223.86	spp_portscan: PORTSC	1	0	1
MY.NET.201.34:	spp_portscan: portsc	15	0	15
MY.NET.217.230:	spp_portscan: portsc	11	0	2005
MY.NET.97.154:	spp_portscan: portsc	10	0	2403
MY.NET.97.176:	spp_portscan: portsc	13	0	855
MY.NET.98.146:	spp_portscan: portsc	14	0	354
MY.NET.97.62:	spp_portscan: portsc	14	0	228
MY.NET.97.244	SNMP public access	1	1	74
	UDP	12	12	58
	spp_portscan: PORTSC	1	0	1
MY.NET.6.7:	spp_portscan: portsc	14	0	73
MY.NET.97.67:	spp_portscan: portsc	14	0	45
MY.NET.206.78:	spp_portscan: portsc	14	0	36
MY.NET.153.243:	spp_portscan: portsc	14	0	29
MY.NET.222.82:	spp_portscan: portsc	14	0	24
MY.NET.202.134:	spp_portscan: portsc	14	0	20
MY.NET.207.158	SYN	13	13	19
	spp_portscan: PORTSC	1	0	1
MY.NET.222.62	INVALIDACK	2	2	2
	NOACK	1	1	1
	oos	6	4	6
	spp_portscan: PORTSC	1	0	4
	NULL	3	3	5
	NMAPID	1	1	1
MY.NET.220.42:	spp_portscan: portsc	14	0	18
MY.NET.218.70	SYN	13	13	14
MY.NET.60.168	UDP	13	13	14
	spp_portscan: PORTSC	1	0	1
MY.NET.218.70	spp_portscan: PORTSC	1	0	1
MY.NET.70.45	UDP	13	13	13
	spp_portscan: PORTSC	1	0	1
MY.NET.212.62	UDP	13	13	13
	spp_portscan: PORTSC	1	0	1
MY.NET.98.109:	spp_portscan: portsc	13	0	512
MY.NET.98.162	UDP	6	6	3410
	spp_portscan: PORTSC	1	0	11
MY.NET.210.134	SYN	1	1	1
	spp_portscan: PORTSC	1	0	9
	UDP	11	11	125
MY.NET.203.218:	spp_portscan: portsc	13	0	94
MY.NET.202.82:	spp_portscan: portsc	13	0	76
MY.NET.98.115:	spp_portscan: portsc	13	0	43
MY.NET.97.36:	spp_portscan: portsc	13	0	35
MY.NET.98.149:	spp_portscan: portsc	13	0	31
MY.NET.98.249	UDP	1	1	18
	spp_portscan: PORTSC	1	0	2
	SYN	11	11	11
MY.NET.98.172	SYN	4	4	8
	UDP	8	8	15
	spp_portscan: PORTSC	1	0	7
MY.NET.98.163	SYN	4	4	6
	spp_portscan: PORTSC	1	0	10

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	UDP	1	1	1
	oos	7	7	10
MY.NET.225.26:	spp_portscan: portsc	13	0	21
MY.NET.204.134:	spp_portscan: portsc	13	0	20
MY.NET.201.154	SYN	12	12	13
	spp_portscan: PORTSC	1	0	6
MY.NET.97.57:	spp_portscan: portsc	13	0	19
MY.NET.202.54:	spp_portscan: portsc	13	0	16
MY.NET.206.234	UDP	12	12	13
	spp_portscan: PORTSC	1	0	1
MY.NET.138.224	UDP	12	12	12
	spp_portscan: PORTSC	1	0	1
MY.NET.210.182:	spp_portscan: portsc	13	0	13
MY.NET.97.88	UDP	12	12	12
	spp_portscan: PORTSC	1	0	1
MY.NET.201.202:	spp_portscan: portsc	12	0	231
MY.NET.97.15:	spp_portscan: portsc	12	0	190
MY.NET.98.113:	spp_portscan: portsc	12	0	126
MY.NET.98.159:	spp_portscan: portsc	12	0	121
MY.NET.97.121:	spp_portscan: portsc	12	0	111
MY.NET.202.10:	spp_portscan: portsc	12	0	88
MY.NET.53.58:	spp_portscan: portsc	12	0	69
MY.NET.98.185	SYN	4	4	6
	spp_portscan: PORTSC	1	0	42
	oos	6	6	9
	UNKNOWN	1	1	8
MY.NET.98.209:	spp_portscan: portsc	12	0	43
MY.NET.202.50:	spp_portscan: portsc	12	0	33
MY.NET.219.102:	spp_portscan: portsc	12	0	28
MY.NET.211.90:	spp_portscan: portsc	12	0	20
MY.NET.217.246:	spp_portscan: portsc	12	0	17
MY.NET.218.218:	spp_portscan: portsc	12	0	17
MY.NET.97.226	UDP	11	11	14
MY.NET.97.82	UDP	11	11	11
MY.NET.97.226	spp_portscan: PORTSC	1	0	1
MY.NET.97.82	spp_portscan: PORTSC	1	0	4
MY.NET.211.142	UDP	11	11	12
	spp_portscan: PORTSC	1	0	1
MY.NET.217.210	SYN	1	1	1
	UDP	10	10	10
	spp_portscan: PORTSC	1	0	1
MY.NET.98.153:	spp_portscan: portsc	11	0	495
MY.NET.98.155:	spp_portscan: portsc	11	0	436
MY.NET.98.210:	spp_portscan: portsc	11	0	189
MY.NET.97.42:	spp_portscan: portsc	11	0	141
MY.NET.97.110:	spp_portscan: portsc	11	0	97
MY.NET.97.203:	spp_portscan: portsc	11	0	75
MY.NET.203.34	UDP	10	10	71
	spp_portscan: PORTSC	1	0	3
MY.NET.98.112:	spp_portscan: portsc	11	0	67
MY.NET.98.130:	spp_portscan: portsc	11	0	63
MY.NET.98.216:	spp_portscan: portsc	11	0	62
MY.NET.201.230:	spp_portscan: portsc	11	0	62
MY.NET.97.175:	spp_portscan: portsc	11	0	51
MY.NET.97.193:	spp_portscan: portsc	11	0	48
MY.NET.60.38:	spp_portscan: portsc	11	0	44
MY.NET.97.187	SYN	1	1	1
	UDP	6	6	33
	spp_portscan: PORTSC	1	0	5
	oos	3	3	4
MY.NET.98.160:	spp_portscan: portsc	11	0	22
MY.NET.217.98:	spp_portscan: portsc	11	0	18
MY.NET.98.173	UDP	10	10	16
	spp_portscan: PORTSC	1	0	1
MY.NET.203.110	UDP	10	10	15
	spp_portscan: PORTSC	1	0	1
MY.NET.205.162:	spp_portscan: portsc	11	0	16
MY.NET.218.214:	spp_portscan: portsc	11	0	14
MY.NET.222.166:	spp_portscan: portsc	11	0	14
MY.NET.218.134:	spp_portscan: portsc	11	0	13
MY.NET.219.150	SYN	10	10	10
	spp_portscan: PORTSC	1	0	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.97.140	spp_portscan: PORTSC	1	0	1
	UDP	1	1	1
	SYN	9	9	9
MY.NET.98.245	UDP	10	10	10
	spp_portscan: PORTSC	1	0	1
MY.NET.98.140:	spp_portscan: portsc	9	0	946
MY.NET.98.107:	spp_portscan: portsc	10	0	336
MY.NET.70.179	UDP	9	9	263
	spp_portscan: PORTSC	1	0	32
MY.NET.209.230	UDP	8	8	750
	spp_portscan: PORTSC	1	0	3
MY.NET.210.250	UDP	9	9	240
	spp_portscan: PORTSC	1	0	1
MY.NET.98.193:	spp_portscan: portsc	10	0	167
MY.NET.208.218:	spp_portscan: portsc	10	0	137
MY.NET.223.82:	spp_portscan: portsc	10	0	65
MY.NET.97.166	SYN	1	1	1
	UDP	8	8	52
	spp_portscan: PORTSC	1	0	1
MY.NET.98.179:	spp_portscan: portsc	10	0	47
MY.NET.98.156:	spp_portscan: portsc	10	0	45
MY.NET.201.18	SYN	9	9	27
	spp_portscan: PORTSC	1	0	1
MY.NET.202.18:	spp_portscan: portsc	10	0	24
MY.NET.202.26:	spp_portscan: portsc	10	0	24
MY.NET.217.70:	spp_portscan: portsc	10	0	19
MY.NET.98.187	SYN	9	9	16
	UDP	1	1	2
MY.NET.6.39:	spp_portscan: portsc	10	0	18
MY.NET.206.26:	spp_portscan: portsc	10	0	16
MY.NET.210.230:	spp_portscan: portsc	10	0	16
MY.NET.219.198:	spp_portscan: portsc	10	0	14
MY.NET.97.230:	spp_portscan: portsc	10	0	12
MY.NET.97.245	SYN	3	3	3
MY.NET.97.49:	spp_portscan: portsc	10	0	12
MY.NET.97.245	spp_portscan: PORTSC	1	0	2
	UDP	6	6	7
MY.NET.98.133	UDP	9	9	9
	spp_portscan: PORTSC	1	0	3
MY.NET.98.108	SYN	9	9	9
	spp_portscan: PORTSC	1	0	2
MY.NET.60.16:	spp_portscan: portsc	10	0	11
MY.NET.97.147	UDP	1	1	1
	spp_portscan: PORTSC	1	0	1
	SYN	8	8	9
MY.NET.1.9	UDP	9	9	9
MY.NET.209.26:	spp_portscan: portsc	10	0	11
MY.NET.1.9	spp_portscan: PORTSC	1	0	2
MY.NET.201.178	SYN	9	9	9
MY.NET.204.174	UDP	9	9	9
MY.NET.201.178	spp_portscan: PORTSC	1	0	1
MY.NET.208.226	spp_portscan: PORTSC	1	0	1
	SYN	9	9	9
MY.NET.204.174	spp_portscan: PORTSC	1	0	1
MY.NET.219.2	oos	10	2	10
MY.NET.220.30:	spp_portscan: portsc	10	0	10
MY.NET.222.138	spp_portscan: PORTSC	1	0	1
	SYN	9	9	9
MY.NET.97.55	UDP	10	10	10
MY.NET.98.15	SYN	8	8	8
	spp_portscan: PORTSC	1	0	1
	UDP	1	1	1
MY.NET.98.229:	spp_portscan: portsc	9	0	482
MY.NET.222.206:	spp_portscan: portsc	9	0	364
MY.NET.219.126:	spp_portscan: portsc	4	0	2772
MY.NET.6.45:	spp_portscan: portsc	9	0	200
MY.NET.98.168:	spp_portscan: portsc	9	0	180
MY.NET.98.237:	spp_portscan: portsc	9	0	64
MY.NET.98.148:	spp_portscan: portsc	9	0	55
MY.NET.98.139:	spp_portscan: portsc	9	0	43
MY.NET.98.197:	spp_portscan: portsc	9	0	37
MY.NET.207.254	NOACK	1	1	3

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	oos	7	5	7
	spp_portscan: PORTSC	1	0	24
MY.NET.204.206	UDP	8	8	27
	spp_portscan: PORTSC	1	0	3
MY.NET.217.178	UDP	8	8	24
	spp_portscan: PORTSC	1	0	1
MY.NET.219.62:	spp_portscan: portsc	9	0	15
MY.NET.97.64:	spp_portscan: portsc	9	0	12
MY.NET.98.190:	spp_portscan: portsc	9	0	11
MY.NET.220.146:	spp_portscan: portsc	9	0	10
MY.NET.201.10:	spp_portscan: portsc	9	0	10
MY.NET.208.150	UDP	8	8	9
	spp_portscan: PORTSC	1	0	1
MY.NET.110.104	UDP	9	9	9
MY.NET.218.146	SYN	8	8	8
	spp_portscan: PORTSC	1	0	1
MY.NET.203.162	spp_portscan: PORTSC	1	0	1
	UDP	8	8	8
MY.NET.220.10	UDP	8	8	8
MY.NET.97.102	UDP	8	8	8
MY.NET.97.11	UDP	8	8	8
	spp_portscan: PORTSC	1	0	1
MY.NET.97.102	spp_portscan: PORTSC	1	0	1
MY.NET.223.190	spp_portscan: PORTSC	1	0	1
MY.NET.220.10	spp_portscan: PORTSC	1	0	1
MY.NET.223.190	SYN	8	8	8
MY.NET.98.192:	spp_portscan: portsc	9	0	9
MY.NET.201.230	UDP	7	7	272
	spp_portscan: PORTSC	1	0	4
MY.NET.207.110:	spp_portscan: portsc	8	0	260
MY.NET.60.43:	spp_portscan: portsc	8	0	166
MY.NET.209.230:	spp_portscan: portsc	8	0	163
MY.NET.98.127:	spp_portscan: portsc	8	0	156
MY.NET.98.182:	spp_portscan: portsc	8	0	140
MY.NET.203.166:	spp_portscan: portsc	8	0	100
MY.NET.98.229	UDP	6	6	571
	spp_portscan: PORTSC	1	0	6
MY.NET.97.90:	spp_portscan: portsc	8	0	75
MY.NET.208.66:	spp_portscan: portsc	8	0	53
MY.NET.97.36	oos	7	7	16
	spp_portscan: PORTSC	1	0	26
MY.NET.97.92:	spp_portscan: portsc	8	0	35
MY.NET.97.146:	spp_portscan: portsc	8	0	24
MY.NET.98.154	SYN	6	6	6
	UDP	1	1	3
	spp_portscan: PORTSC	1	0	13
MY.NET.97.194:	spp_portscan: portsc	8	0	21
MY.NET.219.182	SYN	7	7	17
	spp_portscan: PORTSC	1	0	2
MY.NET.98.138:	spp_portscan: portsc	8	0	19
MY.NET.97.47:	spp_portscan: portsc	8	0	15
MY.NET.98.167:	spp_portscan: portsc	8	0	14
MY.NET.97.80	SYN	2	2	2
	spp_portscan: PORTSC	1	0	1
	UDP	5	5	10
MY.NET.203.254:	spp_portscan: portsc	8	0	13
MY.NET.217.126:	spp_portscan: portsc	4	0	1881
MY.NET.70.38:	spp_portscan: portsc	7	0	299
MY.NET.98.205:	spp_portscan: portsc	7	0	284
MY.NET.97.99:	spp_portscan: portsc	7	0	262
MY.NET.98.219:	spp_portscan: portsc	7	0	53
MY.NET.98.152:	spp_portscan: portsc	7	0	45
MY.NET.98.116:	spp_portscan: portsc	7	0	44
MY.NET.97.17:	spp_portscan: portsc	7	0	26
MY.NET.97.184:	spp_portscan: portsc	7	0	26
MY.NET.211.42:	spp_portscan: portsc	7	0	13
MY.NET.201.98:	spp_portscan: portsc	7	0	12
MY.NET.97.106:	spp_portscan: portsc	7	0	12
MY.NET.97.217:	spp_portscan: portsc	7	0	11
MY.NET.97.65:	spp_portscan: portsc	7	0	10
MY.NET.98.213:	spp_portscan: portsc	7	0	9
MY.NET.217.190	NOACK	2	2	2

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	SYN	1	1	1
	oos	3	3	3
	spp_portscan: PORTSC	1	0	3
MY.NET.97.53	UDP	5	5	300
	spp_portscan: PORTSC	1	0	1
MY.NET.70.163	SYN	1	1	1109
	UDP	1	1	486
	spp_portscan: PORTSC	1	0	4
MY.NET.203.30	INVALIDACK	1	1	1
	spp_portscan: PORTSC	1	0	4
	NOACK	3	3	5
	oos	1	1	1
MY.NET.224.150:	spp_portscan: portsc	6	0	11
MY.NET.225.30	UNKNOWN	1	1	2
	NOACK	1	1	1
	NMAPID	1	1	1
	INVALIDACK	1	1	1
	oos	1	1	1
	spp_portscan: PORTSC	1	0	5
MY.NET.163.17	SYN	1	1	1163
!!!!	connect to 515 from	1	1	1
	spp_portscan: PORTSC	1	0	3
MY.NET.97.206:	spp_portscan: portsc	5	0	146
MY.NET.130.185	SYN	4	4	54
	spp_portscan: PORTSC	1	0	2
MY.NET.208.198:	spp_portscan: portsc	5	0	17
MY.NET.98.190	SYN	1	1	2
	oos	3	3	6
	spp_portscan: PORTSC	1	0	8
MY.NET.222.126	INVALIDACK	1	1	1
	spp_portscan: PORTSC	1	0	2
	SYNFIN	1	1	1
	oos	2	2	2
MY.NET.181.131	spp_portscan: PORTSC	1	0	1
	INVALIDACK	1	1	1
	oos	3	3	3
MY.NET.70.176	SYN	1	1	792
	spp_portscan: PORTSC	1	0	1
	oos	1	1	1
MY.NET.210.250:	spp_portscan: portsc	4	0	110
MY.NET.98.111:	spp_portscan: portsc	4	0	91
MY.NET.70.179:	spp_portscan: portsc	4	0	32
MY.NET.98.169	SNMP public access	1	1	2
	UDP	1	1	12
	spp_portscan: PORTSC	1	0	1
	SYN	1	1	1
MY.NET.98.143:	spp_portscan: portsc	4	0	14
MY.NET.98.126	SYN	2	2	4
	spp_portscan: PORTSC	1	0	6
	UDP	1	1	3
MY.NET.97.13:	spp_portscan: portsc	4	0	13
MY.NET.98.221	UDP	3	3	11
	spp_portscan: PORTSC	1	0	1
MY.NET.202.142:	spp_portscan: portsc	4	0	12
MY.NET.1.9:	spp_portscan: portsc	4	0	11
MY.NET.225.46	INVALIDACK	2	2	2
	NOACK	1	1	1
	spp_portscan: PORTSC	1	0	3
MY.NET.226.134	NOACK	1	1	1
	oos	2	2	2
MY.NET.111.156	SMB Name Wildcard	2	2	17
	SNMP public access	1	1	19
MY.NET.201.78:	spp_portscan: portsc	3	0	27
MY.NET.70.37	UDP	2	2	23
	spp_portscan: PORTSC	1	0	2
MY.NET.97.170:	spp_portscan: portsc	3	0	24
MY.NET.99.51	SYN	2	2	21
	spp_portscan: PORTSC	1	0	2
MY.NET.97.81:	spp_portscan: portsc	3	0	21
MY.NET.212.2	SYN	1	1	1
	UDP	1	1	19
	spp_portscan: PORTSC	1	0	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MY.NET.202.38	UNKNOWN	1	1	1
	oos	1	1	1
	spp_portscan: PORTSC	1	0	18
MY.NET.97.200	SNMP public access	1	1	10
	spp_portscan: PORTSC	1	0	1
	oos	1	1	1
MY.NET.98.174	SYN	2	2	5
	spp_portscan: PORTSC	1	0	6
MY.NET.97.180	SYN	1	1	1
	UNKNOWN	1	1	2
	spp_portscan: PORTSC	1	0	3
MY.NET.209.82	oos	2	2	2
	spp_portscan: PORTSC	1	0	3
MY.NET.207.50:	spp_portscan: portsc	3	0	4
MY.NET.217.194	INVALIDACK	1	1	1
	oos	1	1	1
	spp_portscan: PORTSC	1	0	2
MY.NET.212.102:	spp_portscan: portsc	3	0	4
MY.NET.221.114	INVALIDACK	1	1	1
	UNKNOWN	1	1	1
MY.NET.97.83	spp_portscan: PORTSC	1	0	1
	oos	1	1	1
	SYN	1	1	1
MY.NET.223.194	INVALIDACK	1	1	1
	spp_portscan: PORTSC	1	0	1
	oos	1	1	1
MY.NET.150.139	oos	2	2	2
	spp_portscan: PORTSC	1	0	1
MY.NET.217.166	spp_portscan: PORTSC	1	0	1
	oos	1	1	1
	UNKNOWN	1	1	1
MY.NET.186.16	NULL	1	1	97
	spp_portscan: PORTSC	1	0	165
MY.NET.186.17	NULL	1	1	41
	spp_portscan: PORTSC	1	0	150
MY.NET.97.155	SNMP public access	1	1	60
	spp_portscan: PORTSC	1	0	4
MY.NET.97.52	SNMP public access	1	1	33
	spp_portscan: PORTSC	1	0	1
MY.NET.139.230	SYN	1	1	9
	spp_portscan: PORTSC	1	0	1
MY.NET.219.122	Tiny Fragments - Pos	1	1	7
!!!!	connect to 515 from	1	1	1
MY.NET.97.137	oos	2	2	7
MY.NET.98.174:	spp_portscan: portsc	2	0	7
MY.NET.203.110:	spp_portscan: portsc	2	0	7
MY.NET.219.214:	spp_portscan: portsc	2	0	6
MY.NET.70.42	spp_portscan: PORTSC	1	0	2
	SNMP public access	1	1	4
MY.NET.253.12 !!!!	connect to 515 from	1	1	3
MY.NET.225.186	INVALIDACK	1	1	1
MY.NET.211.130	oos	2	2	3
MY.NET.121.26	UNKNOWN	1	1	1
MY.NET.217.206	UNKNOWN	1	1	1
MY.NET.227.86	oos	2	2	2
	INVALIDACK	1	1	1
MY.NET.205.138	Russia Dynamo - SANS	1	1	442
MY.NET.162.201	SNMP public access	1	1	71
MY.NET.101.160	SMB Name Wildcard	1	1	58
MY.NET.98.134	SNMP public access	1	1	40
MY.NET.97.133	SNMP public access	1	1	26
MY.NET.206.222	SNMP public access	1	1	8
MY.NET.97.107	SNMP public access	1	1	7
MY.NET.99.244!! !!	connect to 515 from	1	1	2
MY.NET.219.194! !!!	connect to 515 from	1	1	1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.