Assignment One – Network Detects

1-1 Network Detect One

'HTTP_Unix_Passwords' event detected by the RealSecure sensor at 'INTERNALENSOR'.

Details:

Source Address: 216.177.16.64
Source Port: 1946
Source MAC Address: 00:E0:FE:7C:30:A0
Destination Address: my.net.6.5
Destination Port: HTTP (80)
Destination MAC Address: 00:10:83:36:04:70
Time: Thursday, July 05, 2001 02:48:34
Protocol: TCP (6)
Priority: high
Actions mask: 0x244
Event Specific Information:

URL: /cgi-bin/pub_affairs/article5.pl?file_dir=././././././././././././././etc/passwd

OBJECT: /cgi-bin/pub_affairs/article5.pl
QUERY: file_dir=././././././././././././././etc/passwd

02:48:29.447596 my.net.6.5.80 > 216.177.16.64.1941: FP 410966647:410967571(924) ack
1267379385 win 32768 (DF)

"GET /cgi-bin/pub_affairs/article5.pl?file_dir=05May2001 HTTP/1.0" 200 6764
"GET /pub_affairs/new_background.jpg HTTP/1.0" 200 6497
"GET /pub_affairs/images/bullet.gif HTTP/1.0" 200 971
1-1-1 Source of Trace:

This alert/trace is from an internal sensor on my employer's network. It has been used with the permission of the Local Computer Incident Response Team (LCIRT).

1-1-2 Detect was generated by:


B. The second piece of the trace is from a Shadow sensor in the DMZ. It shows the connection back to the attacker and how much data was sent.

C. The third piece of the trace is the log from the web server where we see the HTTP/1.0 200 in response to the GET (“GET - an entity corresponding to the requested resource is sent in the response” per RFC1945)

---

1-1-3  Probability the source address was spoofed:

The source address is not spoofed. The attacker is attempting to gain access to the password file of a UNIX host in order to obtain ROOT access on the system. The attacker must use a real IP Address or be located somewhere along the path of the response from the web server using a sniffer which is unlikely. The NSLookup and Whois information indicate this is an ISP Domain which supports an assumption that this is a real IP Address.

NSLOOKUP Information:

07/05/01 13:14:24 dns 64.sun5.dialup.G4.NET
Canonical name: 64.sun5.dialup.G4.NET
Addresses: 216.177.16.64

Whois Information:

whois 64.sun5.dialup.G4.NET  .net is a domain of Network services
Searches for .net can be run at http://www.crsnic.net/
whois -h whois.crsnic.net g4.net ... Redirecting to TUCOWS, INC.
whois -h whois.opensrs.net g4.net ...  Redirecting to TUCOWS, INC.
Registrant: G4 Communications Corp
1 Sundial Avenue
Manchester, NH 03103
US
Domain Name: G4.NET
Administrative Contact: Cav, Cent domreg@cav.net
1 Sundial Avenue
Manchester, NH 03103
US
603-647-2004
Technical Contact: Domain, Administration domreg@g4.net
1 Sundial Avenue, Suite# 114
Manchester, NH 03103
US
603-623-2002
Billing Contact: Cav, Cent domreg@cav.net
1 Sundial Avenue
Manchester, NH 03103
US
603-647-2004
Record last updated on 05-Jul-2001.

notes/rfc1945.txt
1-1-4 Description of the attack:

The attacker is looking for CGI/Perl scripts that are written without consideration to the exploits they provide or the security environment in which they operate. This particular attack is an attempt to exploit a Directory Traversal condition to obtain the /etc/passwd file of the system and eventual control of the system itself. Please note that the amount of data returned as a result of the request for “/etc” and “/etc/passwd” are the same size of 683 bytes. This is a small indication that the attacker may not have gotten what he was looking for and is confirmed when you try the exploit itself with a web browser. The Real Secure alert is just the alarm.

“Vulnerable CGI programs present a particularly attractive target to intruders because they are relatively easy to locate, and they operate with the privileges and power of the web server software itself. Intruders are known to have exploited vulnerable CGI programs to vandalize web pages, steal credit card information, and set up back doors to enable future intrusions, even if the CGI programs are secured.\(^2\)”

There have been several CVE’s released in the past years and continue to be released. Here are a few:

- CVE-1999-0146
- CVE-1999-0149
- CVE-1999-0174
- CVE-1999-0264
- CVE-1999-0744
- CVE-1999-0853
- CVE-2000-0023
- CVE-2000-0731

1-1-5 Attack mechanism:

This is a response, the web server is being targeted, this service has known vulnerabilities and the attacker is trying to exploit a known vulnerability. (Cooper, Page28).

From Network Ice\(^3\):

A common bug with web servers is when a hacker specifies a URL that looks something like /../../../foo/bar.txt. The contents of the website are usually in a subdirectory. The series of "../.." go up the directory structure, then down to the desired file.

The reason this attack works is because the programmer doesn't double-check the URL to see if it is a valid file in the website.

If successful then the attacker would have a list of accounts and account information. Once obtained, a copy of “John the Ripper: Password Cracker” could be used to crack the passwords.

---

\(^2\) SANS Top Ten Vulnerabilities, “2. Vulnerable CGI programs and application extensions (e.g., ColdFusion) installed on web servers”, http://www.sans.org/topten.htm

\(^3\) Network Ice, HTTP URL directory traversal/climbing. http://www.networkice.com/Advice/Intrusions/2000603
in the /etc/passwd file which would provide him with the root password and allow him to access and use the system for any purpose he chooses. If the /etc/passwd file is shadowed, then he would at least have a list of all accounts and account information which may include names and phone numbers. A brute force attack could gain him access in this case or some Social Engineering and a couple of phone calls using names and information from the password file may allow him to simply ask for and receive the password he needs to gain access to the system.

1-1-6 Correlations:

We see this type of attack on a weekly basis on my employers network. There are several on the SANS Incidents.Org site:

http://www.incidents.org/archives/y2k/051400.htm (cgi-bin and /etc/passwd).

1-1-7 Evidence of active targeting:

Yes this is evidence of active targeting. This attacker was targeting a specific host.

1-1-8 Severity:

This is a public web server, setting in the DMZ. All system patches are applied and the /etc/passwd file is shadowed. The network is protected by a firewall but it allows port 80 through. The DMZ Router is using Router Access Control Lists (ACLs). Multiple Intrusion Detection Systems (IDS) are in place.

\[(3 + 5) – (5 + 7) = 1\]

<table>
<thead>
<tr>
<th>Criticality of host:</th>
<th>3</th>
<th>Web Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lethality of attack:</td>
<td>5</td>
<td>Attacker can eventually gain Root Access</td>
</tr>
<tr>
<td>Host Countermeasures:</td>
<td>5</td>
<td>Patched and password file shadowed</td>
</tr>
<tr>
<td>Network Countermeasures:</td>
<td>2</td>
<td>Permissive firewall and IDS sensors</td>
</tr>
</tbody>
</table>

Despite what the log files and the IDS traces above show, the attacker did NOT get a copy of the /etc/passwd file. He was returned a web page telling him that the requested file was unavailable. The web server reported a “200 OK” in response to the GET because the CGI-BIN/ARTICLE5.PL file denies access to it. I could have shown you the alert from six weeks ago where this was not the case, here is what the attacker received in exchange for his request:

1-1-9 Defensive recommendation:

If Perl scripts are not used then do not allow them to run on the web server, if they do run then

---

4 John The Ripper is a password cracker, currently available for UNIX, DOS, Win32. Its intended purpose is to detect weak UNIX passwords. http://www.openwall.com/john
insist on some basic directory and file name checks that check for and deny access to requested critical system directories and files. Use the principle of least privilege whenever possible. The programmer has implemented some simple security in his script. Requests for the /etc/passwd return the user to the main page where a selection must be made. The script reads a list of authorized from a file and provides a simple menu for the user to choose from. If the script is ran by itself it produces the web page that allows the user to make his selection from. Unfortunately, without knowing this, the Intrusion Analyst will (and actually did in this case) go nuts because everything says that the attacker got exactly what he asked for. This demonstrates one method of securing a script, by using static file listings and ignoring all input that does not match the static data provided.

1-1-10 Multiple Choice test Question:

'HTTP_Unix_Passwords' event detected by the RealSecure sensor at 'INTERNALSENSOR'.

Details:
Source Address: 216.177.16.64
Source Port: 1946
Source MAC Address: 00:E0:FE:7C:30:A0
Destination Address: my.net.6.5
Destination Port: HTTP (80)
Destination MAC Address: 00:10:83:36:04:70
Time: Thursday, July 05, 2001 02:48:34
Protocol: TCP (6)
Priority: high
Actions mask: 0x244
Event Specific Information:

URL:
/cgi-bin/pub_affairs/article5.pl?file_dir=../../../../../../../../etc/passwd

OBJECT: /cgi-bin/pub_affairs/article5.pl
QUERY: file_dir=../../../../../../../../etc/passwd

The above Real Secure alert and web server log indicate that an attacker obtained what information?
A. A directory listing of the /etc directory
B. A copy of the /etc/passwd file
C. Both A and B
D. None of the above

ANSWER: D. While both indicate that an attempt was made to access the “/etc” directory and obtain a copy of the “/etc/passwd” file, neither proves this happened. Trying the exploit itself proves that the attacker did not obtain either item requested.
1-2 Network Detect Two

[**] IDS278/dns_named-probe-version [**]
UDP TTL:47 TOS:0x0 ID:35333 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
0x0010: 00 3A 8A 05 00 00 2F 11 95 17 C3 75 E4 51 XX XX ......u.Qxx
0x0020: 3E 81 0D B2 00 35 00 26 50 CB 12 34 00 80 00 01 >....5.&..4...
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 .......version.b
0x0040: 69 6E 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-07:08:39.707309 209.128.96.7:1854 -> MY.NET.60.54:53
UDP TTL:52 TOS:0x0 ID:56939 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
0x0010: 00 3A DE 6B 00 00 34 11 B4 3B D1 80 60 07 XX XX ..k..4..`..xx
0x0020: 3C 36 07 3E 00 00 26 CF C9 12 34 00 80 00 01 <6>..>&..4...
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 .......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:02:09.248945 194.228.83.58:1231 -> MY.NET.136.228:53
UDP TTL:48 TOS:0x0 ID:54698 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
0x0010: 00 3A DE 6B 00 00 30 11 8F B7 C2 8F 60 XX XX ..0..S..xx
0x0020: 3D 11 8F 00 00 26 CF C9 12 34 00 80 00 01 >..&..4...
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 .......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:08:57.897824 63.174.214.200:1918 -> MY.NET.126.201:53
UDP TTL:48 TOS:0x0 ID:49605 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
0x0010: 00 3A C1 C5 00 00 33 11 8F B7 C2 E4 53 3A XX XX .....0.....S:xx
0x0020: 7E C9 07 7E 00 35 00 26 A0 F3 12 34 00 80 00 01 ....&..4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 .......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:10:09.789724 64.160.110.1:4624 -> MY.NET.135.58:53
UDP TTL:51 TOS:0x0 ID:32232 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
0x0010: 00 3A C1 C5 00 00 33 11 8F B7 C2 E4 53 3A XX XX ......0.....S:xx
0x0020: 7E C9 07 7E 00 35 00 26 A8 07 12 34 00 80 00 01 ~.~&..4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 .......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:39:06.209818 64.160.110.1:4624 -> MY.NET.135.58:53
UDP TTL:51 TOS:0x0 ID:32232 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
0x0010: 00 3A C1 C5 00 00 33 11 8F B7 C2 E4 53 3A XX XX ......0.....S:xx
0x0020: 7E C9 07 7E 00 35 00 26 A8 07 12 34 00 80 00 01 ....&..4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 .......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

Page 8 of 181
[**] IDS278/dns_named-probe-version [**]
07/16-10:40:15.614253 209.128.96.7:2880 -> MY.NET.156.10:53
UDP TTL:51 TOS:0x0 ID:10250 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0`...h.
0x0010: 00 3A 28 0A 00 00 33 11 0B C9 D1 80 60 07 XX XX :(...)...
0x0020: 9C 0A 0B 40 00 35 00 26 6B F3 12 34 00 80 00 01 ...@.5&..4
0x0030: 00 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 ......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:40:15.614567 209.128.96.7:2880 -> MY.NET.156.10:53
UDP TTL:50 TOS:0x0 ID:10250 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 00 ...|0`...h.
0x0010: 00 3A B9 A8 00 00 32 11 0C C9 D1 80 60 07 XX XX :(...)...
0x0020: 9C 0A 0B 40 00 35 00 26 6B F3 12 34 00 80 00 01 ...@.5&..4
0x0030: 00 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 ......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:41:12.895110 194.228.57.189:3260 -> MY.NET.181.103:53
UDP TTL:47 TOS:0x0 ID:47528 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 00 ...|0`...h.
0x0010: 00 3A B9 A8 00 00 30 11 05 2E 9D 39 BD XX XX :.....9.xx
0x0020: B5 67 0C BC 00 35 00 26 86 00 12 34 00 80 00 01 ...g..5&..4
0x0030: 00 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 ......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:41:12.895625 194.228.57.189:3260 -> MY.NET.181.103:53
UDP TTL:46 TOS:0x0 ID:47528 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 00 ...|0`...h.
0x0010: 00 3A B9 A8 00 00 30 11 05 2E 9D 39 BD XX XX :.....9.xx
0x0020: B5 67 0C BC 00 35 00 26 86 00 12 34 00 80 00 01 ...g..5&..4
0x0030: 00 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 ......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

[**] IDS278/dns_named-probe-version [**]
07/16-10:41:12.895625 194.228.57.189:3260 -> MY.NET.181.103:53
UDP TTL:46 TOS:0x0 ID:47528 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 00 ...|0`...h.
0x0010: 00 3A B9 A8 00 00 30 11 05 2E 9D 39 BD XX XX :.....9.xx
0x0020: B5 67 0C BC 00 35 00 26 86 00 12 34 00 80 00 01 ...g..5&..4
0x0030: 00 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62 ......version.b
0x0040: 69 6E 64 00 00 10 00 03 ind.....

1-2-1 Source of Trace:
This alert/trace is from an internal sensor on my employers network. It has been used with the permission of the Local Computer Incident Response Team (LCIRT).

1-2-2 Detect was generated by:

Snort v1.7 on Windows NT 4, using the following rule from Whitehats.com

alert UDP $EXTERNAL any -> $INTERNAL 53 (msg: "IDS278/dns_named-probe-version"; content: ":07\version"; offset: 12; nocase; content: ":04\bind"; offset: 12; nocase;)

1-2-3 Probability the source address was spoofed:

The source address is probably not spoofed. This event was logged as a DNS Named Probe. Probes are active reconnaissance. The person conducting the probe needs to either see the response or be on the subnet of the machine receiving the response to see (hear) the results of his probe.

NSLookup and Whois information on the hosts performing the probe follows:

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
Trying 195.117.228.81 at ARIN
Trying 195.117.228 at ARIN
Redirecting to RIPE ...
Trying 195.117.228.81 at RIPE
Trying 195.117.228 at RIPE
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 195.117.228.0 - 195.117.228.31
netname: TFI-PZU
descr: Towarzystwo Funduszy Inwestycyjnych PZU S.A. Warszawa
country: PL
admin-c: AK6009-RIPE
tech-c: BS1071-RIPE
status: ASSIGNED PA
mnt-by: AS5617-MNT
changed: tkielb@cst.tpsa.pl 19991005
source: RIPE

route: 195.117.0.0/16
descr: TPNET (PL)
descr: Provider Local Registry
origin: AS5617
notify: konradpl@zt.piotrkow.tpsa.pl
mnt-by: AS5617-MNT
changed: konradpl@zt.piotrkow.tpsa.pl 19970303
source: RIPE
person: Andrzej Kurzejamski
address: Towarzystwo Funduszy Inwestycyjnych PZU S.A.
address: 00-844 Warszawa
address: ul. Grzybowska 77
phone: +48 501 178959
fax-no: +48 22 6615052
e-mail: a.kurzejmski@tfipzu.com.pl
nic-hdl: AK6009-RIPE
mnt-by: AS5617-MNT
changed: tkielb@cst.tpsa.pl 19991005
source: RIPE

person: Barbara Sarnacka
address: TP S.A.
address: ul. Nowogrodzka 47a
address: 00-695 Warszawa
address: POLAND
phone: +48 22 6252063
e-mail: sarna@cst.tpsa.pl
nic-hdl: BS1071-RIPE
mnt-by: AS5617-MNT
changed: wmalek@cst.tpsa.pl 19980225
source: RIPE

nslookup 209.128.96.7
Canonical name: 209-128-96-007.bayarea.net
Addresses:
  209.128.96.7

Trying 194.228.83.58 at ARIN
Trying 194.228.83 at ARIN
Redirecting to RIPE ...
Trying 194.228.83.58 at RIPE
Trying 194.228.83 at RIPE
Trying 194.228 at RIPE
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 194.228.0.0 - 194.228.0.255
netname: HENNLOCH-NET
descr: Hennlich Industrietechnik s.r.o.
descr: Litomerice
country: CZ
admin-c: PS1950-RIPE
tech-c: PS1950-RIPE
status: ASSIGNED PA
notify: hostmaster@iol.cz
mnt-by: AS5610-MNT
changed: hostmaster@iol.cz 20000321
source: RIPE
route: 194.228.0.0/17
descr: CZ.CZNET
origin: AS5610
notify: hostmaster@iol.cz
mnt-by: AS5610-MTN
changed: vogel@nex.tel.cz 19981120
source: RIPE

person: Pavel Sumera
address: HENNLICH INDUSTRIETECHNIK, spol. s r.o.
address: Turgenevova 19
address: Litomerice
address: 412 01
address: Czech Republic
phone: +420 416 711111
fax-no: +420 416 711999
e-mail: hen.ltm@unl.pvtnet.cz

nic-hdl: PS1950-RIPE
changed: kabelova@pvt.cz 19980406
source: RIPE

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

Trying 63.174.214.200 at ARIN

Trying 63.174.214 at ARIN
Sprint (NETBLK-SPRN-BLKS) SPRN-BLKS 63.160.0.0 - 63.175.255.255
LOWESTFARE.COM (NETBLK-FON-106842265658042) FON-106842265658042
   63.174.214.0 - 63.174.214.127
EPHONES (NETBLK-FON-106842278458103) FON-106842278458103
   63.174.214.128 - 63.174.214.255
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

Trying 64.160.110.1 at ARIN

Trying 64.160.110 at ARIN
Pacific Bell Internet Services,Inc. (NETBLK-PBI-NET-8)
   268 Bush St. #5000
   San Francisco, CA 94104
   US

   Netname: PBI-NET-8
   Netblock: 64.160.0.0 - 64.175.255.255
   Maintainer: PACB

   Coordinator:
      Pacific Bell Internet (PIA2-ORG-ARIN) ip-admin@PBlxxET
          888-212-5411

   Domain System inverse mapping provided by:
      
      NS1.PBlxxET 206.13.28.11
      NS2.PBlxxET 206.13.29.11

      ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

      please send all abuse issue e-mails to abuse@pbi.net
nslookup 194.228.57.189
Canonical name: pha-189.eridan.cz
Addresses:
194.228.57.189

Trying 157.92.15.198 at ARIN
Trying 157.92.15 at ARIN
Universidad Nacional de Buenos Aires (NET-REDUBA)
Ciudad Universitaria
Pabellon, I
AR

Netname: REDUBA
Netblock: 157.92.0.0 - 157.92.255.255

Coordinator:
RED-UBA (ZR38-ARIN) ombu@mail.com
4783-0729

Domain System inverse mapping provided by:
NS1.UBA.AR 157.92.1.1
NS2.UBA.AR 157.92.4.1

1-2-4 Description of the attack:
This is an attempt by seven hosts to scan eight systems to determine the version of BIND they are running.

1-2-5 Attack mechanism:
Once you know the version of BIND, then you can determine if the version is vulnerable and what those vulnerabilities are. Known vulnerabilities are listed at:


CERT.ORG: CA-1998-05, CA-1999-14 and CA-2001-02 (Multiple Vulnerabilities in BIND),

1-2-6 Correlations:
This detect is not new. Several GCIA Practicals contain an analysis of this exploit:
Maria Bianchi GCIA (286)  http://www.sans.org/y2k/practical/Maria_Bianchi_GCIA.doc
Jeff Dell GCIA (312)  http://www.sans.org/y2k/practical/Jeff_Dell_GCIA.doc
Brian Varine (345)  http://www.sans.org/y2k/practical/Brian_Varine_GCIA.doc

It is also discussed in Chapter 3, The Most Critical Internet Security Threats (Part 1), pages 42 thru 46 of the book Intrusion Signatures and Analysis.

1-2-7 Evidence of active targeting:

This is active targeting. Seven Hosts scanned eight systems specifically for the version of BIND they were running.

1-2-8 Severity (See Appendix B):

None of the scanned systems are DNS servers. Had these been DNS Servers, then the formula below would have been (5 + 4) + (4 + 2) = +3

(2 + 1) – (3 + 2) = -2

Criticality of host: 2 (None of the system were DNS Servers).
Lethality of attack: 1 (Because they were not DNS, the attack would not succeed).
Host Countermeasures: 3 (Modern Operating systems with minimum patches)
Network Countermeasures: 2 (Firewall allowed this one to get through).

1-2-9 Defensive recommendation:

There is a firewall in place and it should be configured to block version bind requests. Use Router ACL’s to restrict access to port 53 on specific hosts in the Internal Network. To reduce the possibility of a successful exploit you should keep the version of BIND current and patched. Keep the number of systems running BIND to the minimum.

1-2-10 Multiple Choice test Question:

The previous log entries demonstrate:

A. Nothing, they are all mistakes.
B. Active Targeting
C. Active reconnaissance
D. None of the above.

ANSWER: B. One indication of Active Targeting is the “one-to-one” relationship between the

5 Cooper, Fearnow, Frederick and Northcutt “Intrusion Signatures and Analysis”. Reading: New Riders Publishing 2001
attacker and the intended victim. The clincher is the fact that a specific vulnerability, exploit or piece of information is being used or looked for. This is a prelude to the real attack.
1-3 Network Detect Three

[*] IDS259/web-misc_http-alibaba-overflow [*]
TCP TTL:57 TOs:0x0 ID:21751 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x6DE1AB96 Ack: 0x3DC10DC3 Win: 0x4470 TcpLen: 20
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[*] IDS259/web-misc_http-alibaba-overflow [*]
TCP TTL:56 TOs:0x0 ID:21751 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x6DE1AB96 Ack: 0x3DC10DC3 Win: 0x4470 TcpLen: 20
0x0000: 00 10 83 95 CA 00 00 E0 FE 7C 30 A0 08 00 45 00 .........|0...E.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[*] IDS259/web-misc_http-alibaba-overflow [*]
TCP TTL:57 TOs:0x0 ID:23289 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xF810989 Ack: 0x3E11EF68 Win: 0x4470 TcpLen: 20
0x0000: 00 E0 FE 7C 30 A0 00 60 83 95 19 68 08 00 45 00 ...|0..`...h..E.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[*] IDS259/web-misc_http-alibaba-overflow [*]
TCP TTL:56 TOs:0x0 ID:23289 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xF810989 Ack: 0x3E11EF68 Win: 0x4470 TcpLen: 20
0x0000: 00 10 83 95 CA 00 00 E0 FE 7C 30 A0 08 00 45 00 .........|0...E.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

1-3-1 Source of Trace:

This alert/trace is from an internal sensor on my employers network. It has been used with the permission of the Local Computer Incident Response Team (LCIRT).

1-3-2 Detect was generated by:

Snort v1.7 on Windows NT 4, using the following rule from Whitehats.com

alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS259/web-misc_http-alibaba-overflow"; dsize: >1400; flags: A+; content: "POST");

1-3-3 Probability the source address was spoofed:

It is unlikely that the source address was spoofed. According to the arachNIDS database at Whitehats.com:

The packet that caused this event is normally a part of an established TCP session, indicating that the source IP address has not been spoofed. If you are using a firewall that

---

6 Whitehats.com archNIDS entry for IDS259 Snort Rule, [http://whitehats.com/info/IDS259](http://whitehats.com/info/IDS259)
supports stateful inspection, and are not vulnerable to sequence number prediction attacks, then you can be fairly certain that the source IP address of the event is accurate. Also, it has been noted that due to the nature of this event the attacker does not normally require response traffic. In most cases this means that the event should be analyzed along with other supporting data before acting on the event.

1-3-4 Description of the attack:

Snort flagged this as an attacker attempting to exploite a known buffer overflow on a freeware web server called Alibaba7. The web server is designed to run on Windows 95/98/NT4/2000. The attacker must send a packet with a data payload greater than 1400 Bytes in size (a description of the data packet is in paragraph 1-3-5 below). The description in CAN-2000-06268 states:

Buffer overflow in Alibaba web server allows remote attackers to cause a denial of server via a long GET request.

A comment on the CVE page states that “this is a relatively old Nessus9 plugin, though the exploit uses POST instead of GET”.

A post to the Neohapsis archives concerning the Whitehats.com Snort rule10 that detects this exploit states that a POST or a GET can be used.

The data packet displayed along with the packet is all Snort captured, we don’t see the remainder of the data packet (if there even is a remainder). I believe the DgmLen is what set Snort Off, the DgmLen is set to 1500, yet we only have 16 bits of payload. It’s not fragmentation because the Don’t fragment flag is set. Any Ideas?

1-3-5 Attack mechanism:

This attack was unsuccessful even though it was tried twice. Something is going on, but it is not an Alibaba exploit, because the contents of the data packet contain characters that make this exploit fail as described in the Neohapsis archive below. Even if this were a valid exploit attempt it would have failed since it was attempted against a Unix Web server and not an Alibaba Web Server. This was flagged by Snort as a buffer overflow exploit for the Alibaba web server. A search of the Neohapsis archives produced a very good explanation11 of the exploit:

Tried a little freeware webserver named Alibaba 2.0 today
and found an exploitable overflow. I telnetted to 127.0.0.1:80
and crashed it using

9 Nessus, http://www nessus.org
POST [enter 1028 'x'] / HTTP/1.0

scanf("%s %s %s", szName, szFile, szSomething);

where szFile is a local variable of 0x400 (=1024) bytes on the stack directly above the return address. Coding an exploit for this is going to be a little tricky as it mustn't have any 0x20, 0x00, 0x61-0x7A in it since these bytes are changes by the foregoing function that converts everything into uppercase.

The attacker sends his packet to the Alibaba server causing the Buffer overflow which will him to run his own code on the server and possibly gain administrator access.

1-3-6 Correlations:
I was unable to find an reported incidents of a compromised Alibaba web server.

Security Focus issued BugTraq\(^{12}\) ID 1482 on July 18, 2000. The previous reference to the Neohapsis Archive post was dated q4 1999.

1-3-7 Evidence of active targeting:
This would be considered active targeting - A single host trying an exploit on another host. Although Snort detected this as an Alibaba Buffer Overflow exploit, the data packet we captured does not support this. What are the chances of one host sending two packets with the same data to the same source over a minute apart and they both triggering the same IDS Alert. Two packets with a DgmLen of 1500 but only 16 Bytes of payload.

1-3-8 Severity:
The flagged exploit was designed to work on a Alibaba Web server which is written for the Windows Operating System only, this web server is on a Unix Platform. The Web server is fully patched as is the OS of the machine the web server is hosted on and additional security measures are in use.

\[(4 + 1) - (4 + 2) = -1\]

Criticality of host: 4 (Web server).
Lethality of attack: 1 (Attack not likely to succeed).
Host Countermeasures: 4 (Moden OS, all patches)
Network Countermeasures: 2 (Permissive Firewall)

1-3-9 Defensive recommendation:

If this were an Alibaba web server, I would take the advice of the folks at Nessus.ORG and get another web server. Apache is good and while Alibaba used to be free, it now costs $99.00 to get a copy of Alibaba. The vendor has not supplied a fix to this problem and the product has not been updated in over a year.

1-3-10 Multiple Choice test Question:

The above trace could be considered an example of?

A. Reconnaissance.
B. Probing
C. Active targeting
E. Wrong number.

ANSWER: C. One attacker, one target.
1-4 Network Detect Four

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:18.479588 131.66.108.224:3500 -> MY.NET.108.229:139
TCP TTL:118 TOs:0x4C ID:36702 IpLen:20 DgmLen:223 DF

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:18.479588 131.66.108.224:3500 -> MY.NET.108.229:139
TCP TTL:118 TOs:0x4C ID:36702 IpLen:20 DgmLen:223 DF

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:18.479588 131.66.108.224:3500 -> MY.NET.108.229:139
TCP TTL:118 TOs:0x4C ID:36702 IpLen:20 DgmLen:223 DF

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:18.479588 131.66.108.224:3500 -> MY.NET.108.229:139
TCP TTL:118 TOs:0x4C ID:36702 IpLen:20 DgmLen:223 DF
Key fingerprint = AF19 FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:19.127412 131.66.108.224:3504 -> MY.NET.108.229:139
TCP TTL:117 TOS:0x34 ID:44894 IpLen:20 DgmLen:229 DF
***AP*** Seq: 0x21FE0 Ack: 0x7A79CB69 Win: 0x21D7 TcpLen: 20
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 34 |...8...E4
0x0010: 00 E5 AF 5E 00 6F AB 00 02 1F E7 79 CB 69 50 18 l........zy.i.
0x0020: 21 D7 93 74 00 00 00 00 00 B9 FF 53 4D 42 73 00 !...SMBs.
0x0030: 00 00 00 18 03 80 00 00 65 1A 19 4E C5 31 D8 E3 e..N.1..
0x0040: 00 00 00 00 FE CA 00 00 00 OD 75 80 84 00 04 U....
0x0050: 11 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .2........
0x0060: 00 D4 00 00 00 47 00 00 00 00 00 00 00 00 00 00 ....G......W.i.
0x0070: 00 E0 FE 7C 30 A0 08 00 45 34 |...8...E4
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:19.811984 131.66.108.224:3506 -> MY.NET.108.229:139
TCP TTL:117 TOS:0x34 ID:44894 IpLen:20 DgmLen:229 DF
***AP*** Seq: 0x21FE8 Ack: 0x7A79CE04 Win: 0x21D7 TcpLen: 20
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 34 |...8...E4
0x0010: 00 DF C2 5E 00 6F AB 00 02 1F E7 79 CB 69 50 18 l........zy.i.
0x0020: 21 D7 93 74 00 00 00 00 00 B9 FF 53 4D 42 73 00 !...SMBs.
0x0030: 00 00 00 18 03 80 00 00 65 1A 19 4E C5 31 D8 E3 e..N.1..
0x0040: 00 00 00 00 FE CA 00 00 00 OD 75 80 84 00 04 U....
0x0050: 11 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .2........
0x0060: 00 D4 00 00 00 47 00 00 00 00 00 00 00 00 00 00 ....G......W.i.
0x0070: 00 E0 FE 7C 30 A0 08 00 45 34 |...8...E4
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......

[**] IDS204/netbios_netbios-nt-null-session [**]
07/16-10:06:19.812139 131.66.108.224:3506 -> MY.NET.108.229:139

Harvey Lange
GCIA Practical v2.9
TCP TTL:117 TOS:0x34 ID:49758 IpLen:20 DgmLen:223 DF

**AP** Seq: 0x21FE8 Ack: 0x7A79CE04 Win: 0x21D7 TcpLen: 20
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 34 ....8...[0...E4
0x0010: 00 DF C2 5E 40 00 75 06 5C B1 83 42 6C E0 XX XX ...
0x0020: 6C E5 0D B2 00 82 00 01 2F E8 7A 79 CE 04 50 18 l........zy_.P.
0x0030: 21 D7 E3 4D 00 00 00 00 00 00 B3 FF 53 4D 42 73 00 
0x0040: 00 00 01 00 00 00 FE CA 00 00 00 00 00 D5 75 00 84 00 04 
0x0050: 11 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0060: 0x0070: 00 D4 00 00 00 47 00 00 00 00 00 00 00 00 00 00 
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0090: 00 20 31 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x00A0: 00 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 

---

IP=131.66.108.224:3512 -> MY.NET.108.229:139

TCP TTL:118 TOS:0x34 ID:62814 IpLen:20 DgmLen:223 DF

**AP** Seq: 0x22024 Ack: 0x7A79D30C Win: 0x21D7 TcpLen: 20
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 34 ....8...[0...E4
0x0010: 00 DF F5 5E 40 00 76 06 28 B1 83 42 6C E0 XX XX ...
0x0020: 6C E5 0D B8 00 80 00 02 20 24 7A 79 D3 0C 50 18 l......zy_.P.
0x0030: 21 D7 78 5E 00 00 00 00 00 00 00 B3 FF 53 4D 42 73 00 
0x0040: 00 00 00 13 00 00 00 00 00 00 00 00 00 00 00 00 
0x0050: 00 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0070: 00 D4 00 00 00 47 00 00 00 00 00 00 00 00 00 00 
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0090: 00 20 31 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x00A0: 00 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 

---

TCP TTL:118 TOS:0x34 ID:62814 IpLen:20 DgmLen:223 DF

**AP** Seq: 0x22024 Ack: 0x7A79D30C Win: 0x21D7 TcpLen: 20
0x0000: 00 10 07 17 38 C0 00 E0 FE 7C 30 A0 08 00 45 34 ....8...[0...E4
0x0010: 00 DF 5E 40 00 76 06 28 B1 83 42 6C E0 XX XX ...
0x0020: 6C E5 0D B8 00 80 00 02 20 24 7A 79 D3 0C 50 18 l......zy_.P.
0x0030: 21 D7 78 5E 00 00 00 00 00 00 00 B3 FF 53 4D 42 73 00 
0x0040: 00 00 00 13 00 00 00 00 00 00 00 00 00 00 00 00 
0x0050: 00 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0070: 00 D4 00 00 00 47 00 00 00 00 00 00 00 00 00 00 
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x0090: 00 20 31 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x00A0: 00 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 

---

[**] IDS204/netbios_netbios-nt-null-session [**]
TCP TTL:118 TOS:0x34 ID:62814 IpLen:20 DgmLen:223 DF

---

[**] IDS204/netbios_netbios-nt-null-session [**]
TCP TTL:118 TOS:0x34 ID:62814 IpLen:20 DgmLen:223 DF

---

[**] IDS204/netbios_netbios-nt-null-session [**]
TCP TTL:118 TOS:0x34 ID:62814 IpLen:20 DgmLen:223 DF
1-4-1 Source of Trace:

This alert/trace is from an internal sensor on my employers network. It has been used with the permission of the Local Computer Incident Response Team (LCIRT).

1-4-2 Detect was generated by:

Snort v1.7 on Windows NT 4, using the following rule from Whitehats.com

alert TCP $EXTERNAL any -> $INTERNAL 139 (msg: "IDS204/netbios_netbios-nt-null-session"; flags: A+; content: "(00 00 00 00 57 00 69 00 64 00 77 00 73 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31|;)"

1-4-3 Probability the source address was spoofed:

The source is probably not spoofed. The purpose of the NT NetBios Null Session is to enumerate shares and local users on an NT system. This is data mining at its finest when it comes to NT.

1-4-4 Description of the attack:

NetBIOS services on NT allow users to connect without any username and password (the NULL session). This is accomplished by: Once connected they can interrogate the machine with any number of tools such as Dumpsec and Legion. These tools allow their user to obtain a list of shares, users, groups and their members, and policy information. This will also open the door to allow tools like lOphtCrack (A Password cracker) to be used to obtain user passwords.

1-4-5 Attack mechanism:

The attack is started with the following command:

    NET USE \MY.NET.108.229\IPC$ "" /u:""

This command will establish a session with the IPC$ share (the hidden Interprocess Communication share that allows machines to communicate) on the target machine (IP MY.NET.108.229) without providing a username and password! Once a session is established, then the target can be interrogated.

To obtain a list of shares on the target machine, you could type:

    NET VIEW \MY.NET.108.229
Tools like Dumpsec or Legion can be used to get Registry information, user and group lists, volume and directory information and security settings, policy settings.

1-4-6 Correlations:
Andrew Windsor GCIA\(^{13}\) (349) in his Practical wrote “Anatomy of a Windows 2000 Enumeration” that covers the subject very well.

Al Evans\(^{14}\), GCIA (298) analyzed an NT Null Session as his first Detect for his practical.

Marc Gregoire\(^{15}\) GCIA (249) as his second Detect in his practical analyzed a NetBios Scan on his network and went on to demonstrate a Null Session compromise.

Karen Frederick\(^{16}\) GCIA (248) evaluated a tool called WinFingerprint in her practical. This tool can be used to establish a Null Session and interrogate a target.


1-4-7 Evidence of active targeting:
This is active targeting. One attacker and one host. A tool like Legion and even ShareSniffer can assist you in scanning for NT hosts that allow Null Sessions but a connection or repeated attempts to connect to a single machine is active targeting.

1-4-8 Severity:
There is no evidence in the IDS logs that our host responded.

\[
(1 + 5) – (4 + 2) = 0
\]

Criticality of host: 1 (This is Windows Desktop system with all patches).
Lethality of attack: 5 (Successful connection can lead to administrator access).
Host Countermeasures: 4 (Modern operating system with all patches).
Network Countermeasures: 2 (Firewall allowed the attack to go through).

1-4-9 Defensive recommendation:
Microsoft provides guidance in a Knowledgebase article on how to restrict the amount of information an Anonymous user can access on NT Systems. It is [http://support.microsoft.com/support/kb/articles/Q143/4/74.asp](http://support.microsoft.com/support/kb/articles/Q143/4/74.asp). I highly recommend that you read and apply its settings and recommendations to all NT Systems.

\(^{13}\) Andrew Windsor, GCIA (349) SANS, [http://www.sans.org/y2k/practical/Andrew_Windsor_GCIA.doc](http://www.sans.org/y2k/practical/Andrew_Windsor_GCIA.doc)

\(^{14}\) All Evans, GCIA (298), SANS, [http://www.sans.org/y2k/practical/Al_Evans_GCIA.doc](http://www.sans.org/y2k/practical/Al_Evans_GCIA.doc)

\(^{15}\) Marc Gregoire GCIA (249) SANS, [http://www.sans.org/y2k/practical/marc_gregoire.doc](http://www.sans.org/y2k/practical/marc_gregoire.doc)

\(^{16}\) Karen Frederick GCIA (248), SANS, [http://www.sans.org/y2k/practical/Karen_Frederick_GIAC.doc](http://www.sans.org/y2k/practical/Karen_Frederick_GIAC.doc)
You should also consider blocking Ports 135 through 139 at your border Routers. Blocking these ports at your border routers is your best defense. If you cannot block the port, then at least consider putting an ACL in place to restrict access to specific systems and then monitor them very closely.

1-4-10 Multiple Choice test Question:

What is the command line used to make a Null Session connection to a Microsoft NT host?

A. NET LOGON \MY.NET.NT.PC \IPC$ "" /U:"
B. NET START \MY.NET.NT.PC \IPC$ "" /U:"
C. NET USE \MY.NET.NT.PC \IPC$ "" /U:"
D. NET VIEW \MY.NET.NT.PC \IPC$ "" /U:"

ANSWER: D. NET VIEW \MY.NET.NT.PC \IPC$ "" /U:"" is the correct command. While A, B and D are valid commands, the syntax for each of those commands is not correct.
<table>
<thead>
<tr>
<th>Time</th>
<th>Source:Port</th>
<th>Destination:Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/16-06:56:04</td>
<td>64.124.157.16:8245</td>
<td>MY.NET.65.83:17952 UDP</td>
</tr>
<tr>
<td>07/16-06:56:04</td>
<td>64.124.157.16:8293</td>
<td>MY.NET.65.83:28717 UDP</td>
</tr>
<tr>
<td>07/16-06:56:05</td>
<td>64.124.157.16:13600</td>
<td>MY.NET.65.83:13362 UDP</td>
</tr>
<tr>
<td>07/16-06:56:06</td>
<td>64.124.157.16:8247</td>
<td>MY.NET.65.83:13088 UDP</td>
</tr>
<tr>
<td>07/16-06:56:06</td>
<td>64.124.157.16:8303</td>
<td>MY.NET.65.83:27706 UDP</td>
</tr>
<tr>
<td>07/16-06:56:11</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:16</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:23</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:19</td>
<td>64.124.157.16:18824</td>
<td>MY.NET.65.83:48 UDP</td>
</tr>
<tr>
<td>07/16-06:56:22</td>
<td>64.124.157.16:14901</td>
<td>MY.NET.65.83:13882 UDP</td>
</tr>
<tr>
<td>07/16-06:56:21</td>
<td>64.124.157.16:13365</td>
<td>MY.NET.65.83:8245 UDP</td>
</tr>
<tr>
<td>07/16-06:56:23</td>
<td>64.124.157.16:47175</td>
<td>MY.NET.65.83:49032 UDP</td>
</tr>
<tr>
<td>07/16-06:56:29</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:25</td>
<td>64.124.157.16:14901</td>
<td>MY.NET.65.83:13882 UDP</td>
</tr>
<tr>
<td>07/16-06:56:27</td>
<td>64.124.157.16:12320</td>
<td>MY.NET.65.83:12320 UDP</td>
</tr>
<tr>
<td>07/16-06:56:28</td>
<td>64.124.157.16:12337</td>
<td>MY.NET.65.83:12592 UDP</td>
</tr>
<tr>
<td>07/16-06:56:35</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:31</td>
<td>64.124.157.16:14901</td>
<td>MY.NET.65.83:13882 UDP</td>
</tr>
<tr>
<td>07/16-06:56:35</td>
<td>64.124.157.16:8240</td>
<td>MY.NET.65.83:12320 UDP</td>
</tr>
<tr>
<td>07/16-06:56:41</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:41</td>
<td>64.124.157.16:14901</td>
<td>MY.NET.65.83:13882 UDP</td>
</tr>
<tr>
<td>07/16-06:56:45</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:47</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:47</td>
<td>64.124.157.16:13365</td>
<td>MY.NET.65.83:8245 UDP</td>
</tr>
<tr>
<td>07/16-06:56:47</td>
<td>64.124.157.16:13108</td>
<td>MY.NET.65.83:8243 UDP</td>
</tr>
<tr>
<td>07/16-06:56:48</td>
<td>64.124.157.16:13365</td>
<td>MY.NET.65.83:8245 UDP</td>
</tr>
<tr>
<td>07/16-06:56:53</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:49</td>
<td>64.124.157.16:8244</td>
<td>MY.NET.65.83:13856 UDP</td>
</tr>
<tr>
<td>07/16-06:56:49</td>
<td>64.124.157.16:12576</td>
<td>MY.NET.65.83:13122 UDP</td>
</tr>
<tr>
<td>07/16-06:56:52</td>
<td>64.124.157.16:14901</td>
<td>MY.NET.65.83:13882 UDP</td>
</tr>
<tr>
<td>07/16-06:56:59</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:56:55</td>
<td>64.124.157.16:14901</td>
<td>MY.NET.65.83:13882 UDP</td>
</tr>
<tr>
<td>07/16-06:57:00</td>
<td>64.124.157.16:15659</td>
<td>MY.NET.65.83:15659 UDP</td>
</tr>
<tr>
<td>07/16-06:57:06</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:57:05</td>
<td>64.124.157.16:8245</td>
<td>MY.NET.65.83:12320 UDP</td>
</tr>
<tr>
<td>07/16-06:57:11</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:57:11</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:57:09</td>
<td>64.124.157.16:8293</td>
<td>MY.NET.65.83:29811 UDP</td>
</tr>
<tr>
<td>07/16-06:57:10</td>
<td>64.124.157.16:12832</td>
<td>MY.NET.65.83:12853 UDP</td>
</tr>
<tr>
<td>07/16-06:57:17</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:57:17</td>
<td>64.124.157.16:1894</td>
<td>MY.NET.65.83:1203 UDP</td>
</tr>
<tr>
<td>07/16-06:57:17</td>
<td>64.124.157.16:36471</td>
<td>MY.NET.65.83:56747 UDP</td>
</tr>
<tr>
<td>07/16-06:57:18</td>
<td>64.124.157.16:36471</td>
<td>MY.NET.65.83:56747 UDP</td>
</tr>
<tr>
<td>07/16-06:57:23</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:57:19</td>
<td>64.124.157.16:15659</td>
<td>MY.NET.65.83:15659 UDP</td>
</tr>
<tr>
<td>07/16-06:57:20</td>
<td>64.124.157.16:8240</td>
<td>MY.NET.65.83:12320 UDP</td>
</tr>
<tr>
<td>07/16-06:57:22</td>
<td>64.124.157.16:13880</td>
<td>MY.NET.65.83:8246 UDP</td>
</tr>
<tr>
<td>07/16-06:57:23</td>
<td>64.124.157.16:513</td>
<td>MY.NET.65.83:21843 UDP</td>
</tr>
<tr>
<td>07/16-06:57:29</td>
<td>64.124.157.16:0</td>
<td>MY.NET.65.83:0 UDP</td>
</tr>
<tr>
<td>07/16-06:57:26</td>
<td>64.124.157.16:513</td>
<td>MY.NET.65.83:21843 UDP</td>
</tr>
<tr>
<td>07/16-06:57:27</td>
<td>64.124.157.16:13088</td>
<td>MY.NET.65.83:13873 UDP</td>
</tr>
</tbody>
</table>
Here is the same data as above but only the Port Zero to Port Zero traffic and sorted by time. See
the six second pattern. There is an occasional deviation from this, but not very often and since
we don’t have the milliseconds I can only attribute it to the occasional network delay.

07/16-06:56:11 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:17 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:23 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:29 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:35 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:41 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:47 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:53 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:56:59 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:05 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:11 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:17 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:23 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:29 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:35 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:41 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:57:47 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:03 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:09 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:15 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:21 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:27 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:33 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:39 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:45 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:51 64.124.157.16:0 -> MY.NET.65.83:0 UDP
07/16-06:58:57 64.124.157.16:0 -> MY.NET.65.83:0 UDP

1-5-1 Source of Trace:
This alert/trace is from an internal sensor on my employers network. It has been used with the
permission of the Local Computer Incident Response Team (LCIRT).

1-5-2 Detect was generated by:
Snort v1.7 on Windows NT 4. This a portion of the Portscan log showing multiple UDP
connection attempts from 64.124.157.16 to MY.NET.65.83 between 06:56:04 to 06:58:59 on 16
Jul 2001. The scan continues on until 07:19:39 and looks just like the three minute portion I have
shown above. The second portion of the trace shows a pattern that also continues. The Snort
sensor locked up at 07:19 is the only reason I have no other data on this at the moment.

1-5-3 Probability the source address was spoofed:
The chances are very good that this source address is spoofed. The protocol being used is UDP so no the sender is not looking for a response.

1-5-4 Description of the attack:

Port zero is sometimes used by routers to exchange information. It is a reserved IANA Port as well. The sender is using port zero to bypass firewalls, routers and some IDS systems. It is an attempt to send control signals to systems that have been compromised by Trojans listening to traffic on the network. The original Red Worm (Now the Adore Worm) was rumored to configure itself to listen on port 65535 for a UDP packet of 77 bytes in length that contained commands for it.

1-5-5 Attack mechanism:

This could be a Trojan or a Denial Of Service. We see no outbound connection to the host sending the data at present. We have only seen this one incident. A virus scan of all files and all drives shows nothing is infected. MSN Messenger is installed on the PC, but until I can confirm that MSN Messenger is causing the problem is still under investigation.

1-5-6 Correlations:

I was able to find one article to substantiate my MSN Messenger theory in the Neohapsis Archives. http://archives.neohapsis.com/archives/firewalls/2001-q1/0567.html. Until I can prove otherwise, this all I have to go on.

1-5-7 Evidence of active targeting:

If this is a Denial of Service or a Trojan then this is Active Targeting. Everything is being directed at this one machine on the network.

1-5-8 Severity:

The machine being target is a Windows 95 desktop (It is being replaced, they just haven’t gotten to it yet). All security patches for the OS are applied. All Browser patches are installed and it has current Anti-virus software and signatures.

\[(1 + 4) - (3 + 2) = 0\]

Criticality of host: 1 (Windows 95 Desktop with patches and Antivirus software)
Lethality of attack: 4 (Denial of Service or Trojan)
Host Countermeasures: 3 (Older operating system, patched with Antivirus software)
Network Countermeasures: 2 (Firewall allowed it through).

1-5-9 Defensive recommendation:
At present I have to wait for the user to return from vacation to check on the MSN Messenger option. Until then the machine is off and will remain off until she returns. I booted up the PC and did not see the suspicious activity on the IDS Sensor which adds more support to the MSN Messenager theory.

**1-5-10 Multiple Choice test Question:**

Using the above network trace.

UDP Port zero to UDP Port Zero traffic logged at regular intervals by an IDS is an indication of?

A. Routers exchanging OSPF Data  
B. Updates between Windows 2000 Dynamic DNS servers  
C. Possible Denial Of Service or Trojan activity  
D. None of the above
Assignment Two – Describe the State of Intrusion Detection

NT Event Log Consolidation: A Solution for Centralized Reporting on Windows Based Snort Sensors

Introduction

Why consolidate NT Event Logging? Page 214 and 215 of Hacking Exposed\textsuperscript{17}, Second Edition, talks about Disabling Auditing and Clearing the Event Log. Centralized logging will not help us with disabling logging other than let us know when logging stopped and started, but it can help us with monitoring the clearing of the Event Log. They will have to hack the syslog server and clear it as well to cover their tracks if NT Events are mirrored to a Syslog server. What good is Event Logging if a hacked system doesn’t have any records to indicate it was hacked. What good is a host based IDS system that records events to the Windows NT Event Log if those logs are erased. As an Intrusion Detection Analyst, you should encourage the use of Centralized NT Event Logging as an aid in gathering correlation data for your detects.

This paper is being presented as a possible solution for Centralized NT Event Logging, for use in setting up a centralized reporting system for Windows Based Snort Sensors. This system along with a script (Like SnortLog\textsuperscript{18}) to monitor Snort Events written to Syslogs can provide you with a Near Real Time Notification Intrusion Detection Network. This is an option to adding Windows Based IDS Sensors with Centralized reporting to your network and an addition to SnortNet\textsuperscript{19}

Why Windows?

Why not Windows? I know everyone thinks it has more holes than Swiss Cheese, but if you think about it, while everyone is busy protecting all of those Unix, Sun and HP systems from attackers; who is protecting the machines being used by your users to access those systems. Most of those users are Windows Operating Systems. You don’t have to hack the big machines, just those of the users. Administrators desktops are probably the most neglected. Why, because they are so busy patching the big boxes, they haven’t had time to patch their own desktops. I believe this is one reason for the increase in Worms, Trojans and port scans. Most users synchronize passwords (and a lot of those are weak according to the SANS Top Vulnerabilities List\textsuperscript{20}) on every machine they login to. For a hacker it’s like one stop shopping. You hack one machine and you have access to everything that user has access to, it’s even better if that user is a Unix Administrator or an NT Domain Administrator.

What about Snort on Windows? It has an option to write alerts to the NT Event Log. Problem is, you have to access each machine independently to see the alerts. This may be fine for a host based IDS, but what if you want to deploy Windows based Intrusion Detection Sensors and


\textsuperscript{18} SnortLog, Syslog Analysis Script by Angelos Karageorgiou, \url{http://www.snort.org/Files/snortlog}

\textsuperscript{19} SnortNet, Distributed Logging for Snort by Fyodor Yarochkin, \url{http://www.snort.org/Files/snortnet.tar.gz}

\textsuperscript{20} The Ten Most Critical Internet Security Threats, SANS, \url{http://www.sans.org/topten.htm}
consolidate the alerts. For this you would need a lot of scripts and batch files. Snort as a Host based IDS would be nice for correlation. It would be nice if you could consolidate the NT Event Logs from Windows based IDS systems in one spot. Maybe setup some automated scripts in the central location to check for alerts.

Let’s say you see a Snort alert for a workstation on your network. You know that the machine has Snort as a host based IDS and logs everything to the NT Event Log. You open up your Event Viewer and change your connection to that workstation and see what it has written down. Alternatively, you map a drive to the Snort Log Share and view the detailed event record of the alert.

Below is a drawing that shows the major pieces of the NT Event Log Consolidation System.

The system I will describe will not require any serious programming efforts and is best setup with the cooperation of the Unix Administrator. This system can be implemented as a whole or in parts. Do not forward NT Event Logs from a workstation with a network printer attached!

What resources do we have to setup this system?

There are several books on the subject of NT Event Logging and loads of information on the internet on the subject. O’Reilly & Associates, Windows NT Event Logging\(^{21}\) covers the process very well. It covers the NT Event service, NT Event Viewer, Security Auditing, the Event Logging API, and programming using C/C++, Visual Basic, and Perl (among others). I also used a copy of the Microsoft Windows NT Server Resource Kit\(^{22}\).

Software to perform the task is everywhere, but most of it costs quite a bit of money. For a small business money can be a big issue. We have a very limited budget so we won’t be purchasing

---


any software. We have the Windows NT Resource Kit and it contains the Dumpel (Dump Event Log) utility. We will also be using a free utility from Intersect Alliance\(^{23}\) called Backlog. From their web page:

BackLog is currently configured to deliver audit information to a SYSLOG server running on a remote (or local) machine. A configuration utility allows you to set the appropriate syslog categories, as well as the target server that should receive the audit information.

You will need to configure the Linux Syslog service to receive the log entries being forwarded from the NT servers when we install the software. We would like all the NT Event logs routed to a separate log file if at all possible. For this I worked with my local Unix administrator and with a little tweaking, we have all Syslog entries from NT Servers going to a separate directory. I have also setup my workstation with Snort as a Host Based IDS (Or a network segment IDS if I turn permiscuous mode on), installed BackLog and set the reporting category to Local6 and Information. I use a Linux based Syslog system because Windows based Syslog Utilities at the right price are few and far between.

OK, we have the tools. Now we have to decide what we want to consolidate or what we can afford to consolidate.

NOTE: One weird item about BackLog is that what you configure during setup is not exactly what you get on the Syslog server. I will give you the settings I used when I setup the system I have in place for my Windows 2000 Host Based IDS.

**Configuring the Syslog process on the Syslog server:**

You will have to talk to the Syslog server administrator and ask him to configure the Syslog Local2 Category to log to a separate file and directory. This is what I selected as the Category for all NT Event Logs to be filed under.

**Installing and configuring BackLog**

We know that we want to consolidate the NT Events from Snort Sensors in one location, but what category settings do we need to set?

WARNING! Do NOT use these settings on a server if you are forwarding NT Events to a Syslog server. Trust me, you don’t have enough disk space.

BackLog comes with its own setup routine. Download a copy from the link in the footnotes and run the program. It will install itself and then start the configuration program. You will need the following information to complete the install:

---

Syslog Category: Local6. Yes, I know I told you to set the Syslog process on the Syslog server to Local2, but here I am telling you to set the category on the NT server to Local6. During my initial testing and working with my Syslog administrator we found that if you set both to Local2 then events did not get logged. If they are both set to Local6 they still don’t get logged. Through experimentation, we found that the combination I have given you here works fine. This is only a bug with the Local categories, all other categories work just fine. For free we were willing to live with that until we could come up with something better.

Change Notice to Information. This is the only setting you can use with the Static Version of Snort for Windows when writing to the NT Event Log.

Snort send all alerts to the NT Event Log as Information. If you have the time you could download the Windows Source Code and change this to write it to the NT Event Log as an alert. Setting this as an Alert would allow Snort to be used as a Host Based IDS and NT Event Log to Syslog consolidation to be an acceptable option for servers as well. As an alert, you could configure BackLog to forward all Warnings and higher to the Syslog server.

Why not servers? In my first attempt at setting up BackLog, I had installed it on fifteen NT 4/2000 servers and set BackLog up to send all Syslog Category Information and Higher events to the Syslog server (I did this at 7:00 P.M at night). The next morning at 06:30 I checked and only had about 3.5 Mb of log files. This wasn’t too bad, but I would have to turn it down some if I was...
going to add more servers to the system. I checked an hour and a half later and the Syslog file
had grown to 8 Mb in size! I had installed BackLog on an NT File and Print server and printing
generates tons of Event Log entries. This was not good, even with Network Attached Storage I
was going to fill things up fast. I switched everything to Warnings and higher and things settled
down. If you are just starting out I would recommend that you start with Alerts and higher and
witch to Warnings later, just to see what your disk space requirements are going to be.

What if I want to archive the NT Event Logs?

Unless you configure BackLog to Debug mode (Not recommended at all), you will have
everything you need for a Host Based IDS on the Syslog server. You will have to manually clean
up the local Alert files periodically, but you can use the Task Scheduler to schedule to archive the
files and remove them periodically if you want, or jus delete them if the machine is being backed
up regularly.

If you want to archive the NT Event Logs, then I recommend the DumpEL (Dump Event Log)
utility that is included in the Microsoft NT Server 4.0 Resource Kit. It is a Command Line utility
that dumps the contents of the local or remote event log into a text file. It has many command
line options and supports both tab and comma n delimited file formats. It works with Windows
2000 and Windows NT 4. It can also be used to search for specific events and export them if
needed. An additional tool that I would recommend is DumpEvt from Somarsoft. Their utility
allows you to specify which Event Log you want to export and the format and location you want
it saved in. It supports the Windows 2000 Directory Service, DNS and File Replication Service
log formats as well.

Using either of the tools mentioned above or both if you have the space. You can create a batch
file with the following command(s):

For comma separated format using DumpEL:

DumpEL –l Application –s RAS.MY.NET –c –f APPLICATION.CSV
DumpEL –l Security –s RAS.MY.NET –c –f SECURITY.CSV
DumpEL –l Event –s RAS.MY.NET –c –f EVENT.CSV

For Native Event Log format Using DumpEL

DumpEL –l Application –s RAS.MY.NET –b –f APPLICATION.EVT
DumpEL –l Security –s RAS.MY.NET –b –f SECURITY.EVT
DumpEL –l Event –s RAS.MY.NET –b –f EVENT.EVT

If you are just archiving then I would recommend using the Native Event Log format. You can
always open the log up later with the Event Viewer and save it as text. Here is what I do:

DumpEL –l Application –s RAS.MY.NET –b –f APPLICATION.EVT
DumpEL –l Security –s RAS.MY.NET –b –f SECURITY.EVT
DumpEL –l Event –s RAS.MY.NET –b –f EVENT.EVT

DumpEVT /computer=RAS.MY.NET /logfile=sec /outfile=C:\sec.dev /reg=local_machine
DumpEVT /computer=RAS.MY.NET /logfile=app /outfile=C:\app.dev /reg=local_machine
DumpEVT /computer=RAS.MY.NET /logfile=sys /outfile=C:\sys.dev /reg=local_machine

The above options export each event log in its native format using DumpEL and again with DumpEVT so I can open it with a text editor. You can modify the export format of DumpEVT to any format (your favorite database if you want) which allows for even greater flexibility in archiving.

**Final Step:**

Download a copy of the static version of the Win32\(^{25}\) Port of Snort from the link in the footnotes. Setup your configuration and start it up, if you have problems with the configuration please refer to the FAQ on the Snort web site. You will also need a copy of the Windows NT Server Resource Kit. You will have to install Snort as a service on NT if you are using this as a Network Based IDS Sensor. Instructions for installing and configuring Snort as a Service can be found in the Snort FAQ\(^{26}\), Question 45. You should also download a current copy of the Snort Rule set from Snort.org or Whitehats.com.

**Summary**

I have shown you an NT Event Log consolidation process that works. It is simple and easy to install. Except for the Windows Operating System software and Windows NT Server Resource Kit, it uses OpenSource software and freeware utilities. It allows you to deploy Snort on Windows as a Host Based IDS system using NT Event Logging while allowing you to forward these alerts to a Syslog server for centralized reporting. It is an alternative to Snort and ACID on Windows. There are needed improvements, but if you want something quick and easy then I recommend the above procedure.

In addition to a Windows Host Based IDS, this paper provides you with the beginnings of a Centralized NT Event Log reporting system. There is a need for such a system and several companies are trying to fulfill that need. I have provided a simple solution for small businesses and even home networks.

Finally, I wanted to provide a solution for what I think is a need, an addition to some of other great ideas that abound on the Internet. The need for centralized reporting is there. Fyodor Yarochkin saw it and wrote SnortNet. Marty Roesch saw a need and founded SourceFire\(^{27}\).


\(^{26}\) How to run Snort as a Service on Win32, Snort Faq #45, [http://www.snort.org/FAQ.html#q45](http://www.snort.org/FAQ.html#q45)

\(^{27}\) SourceFire, Inc. Developing appliance-based network security infrastructure systems with Snort as their core. [http://www.sourceforge.com](http://www.sourceforge.com)
Michael Steele\textsuperscript{28} has written a couple of papers on the use of Snort and SnortSnarf as a front end for Snort in a Windows Environment, they are available in the documents section of the Snort.org web site. Jon Bull\textsuperscript{29} makes the following comment in his article on Installing Snort on a Win 2000 Environment:

\begin{quote}
If you plan on doing a decent job of securing your network, you'll want to keep historical records of all your logs. I suggest Snort2HTML to hand keep logs. This use doesn't scale well however and so large outfits may look towards the MySQL.
\end{quote}

Here he mentions historical records. NT Event Logs while not very portable are compact and somewhat easy to use. The DumpEVT utility provides another option to be used in place of Snort2HTML or MySQL or as convenient way to support MySQL if needed.

\textsuperscript{28} Installing Snort on a Win 2000 System - A walkthrough and Snort on Windows 98/ME/NT4/2000 using Snortsnarf to view alerts by Michael Steele from Silicon Defense
\textsuperscript{29} Jon Bull, Snort’s Place in a Windows 2000 Environment
Assignment Three – “Analyze This” Scenario

3-1 Data Description:

File Type: SNORT Alert Logs.
Number of Files: Seven.
Number of Alerts: 17,057
Date of Log Entries: 10, 11, 12, 13, 14, 15 and 16 April 2001.

File Type: SNORT Portscan Logs.
Number of Files: Seven.
Number of scan lines: 193,148.
Date of Log Entries: 10, 11, 12, 13, 14, 15 and 16 April 2001.

File Type: SNORT Out Of Spec (OOS) Logs.
Number of Files: Six
Number of OOS Entries: 905.
Date of Log Entries: 10, 11, 12, 13, 14 and 16 April 2001.

A one week period was examined, Tuesday thru Monday. One day (Sunday, 15 APR 2001) of Out-Of-Spec (OOS) logs was not available and is therefore not included in the analysis.

A description of the Snort fields is in Appendix A of this practical.

3-2 List Of Detects, Descriptions, Correlations and Defensive Recommendations:

Table 1- Alerts By Category

<table>
<thead>
<tr>
<th>Alert Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2655</td>
<td>Attempted Sun RPC high port access/ SUN RPC highport access</td>
</tr>
<tr>
<td>268</td>
<td>Connect to 515 from inside/outside the network</td>
</tr>
<tr>
<td>250</td>
<td>External RPC call</td>
</tr>
<tr>
<td>746</td>
<td>High port 65535 TCP/UDP - Possible Red Worm - traffic</td>
</tr>
<tr>
<td>8</td>
<td>ICMP SRC and DST outside network</td>
</tr>
<tr>
<td>9</td>
<td>NMAP TCP Ping</td>
</tr>
<tr>
<td>2</td>
<td>Probable NMAP fingerprint attempt</td>
</tr>
<tr>
<td>1006</td>
<td>Possible trojan server activity</td>
</tr>
<tr>
<td>142</td>
<td>Queso fingerprint</td>
</tr>
<tr>
<td>1735</td>
<td>Russia Dynamo</td>
</tr>
<tr>
<td>138</td>
<td>SMB Name Wildcard</td>
</tr>
<tr>
<td>1</td>
<td>STATDX UDP Attack</td>
</tr>
<tr>
<td>4</td>
<td>SYN-FIN scan</td>
</tr>
<tr>
<td>50</td>
<td>TCP SRC and DST outside network</td>
</tr>
</tbody>
</table>
Please note that some alerts are later grouped together for analysis.

Table 2 - Alerts By Category and Day

<table>
<thead>
<tr>
<th></th>
<th>01</th>
<th>02</th>
<th>04</th>
<th>05</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/10</td>
<td>0</td>
<td>159</td>
<td>0</td>
<td>29</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>14</td>
<td>14</td>
<td>1725</td>
<td>25</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>222</td>
<td>131</td>
<td>55</td>
<td>25</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>4/11</td>
<td>1</td>
<td>2</td>
<td>22</td>
<td>275</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>10</td>
<td>19</td>
<td>9</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>136</td>
<td>2263</td>
<td>55</td>
<td>18</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>4/12</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>351</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>11</td>
<td>57</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>239</td>
<td>423</td>
<td>19</td>
<td>17</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4/13</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>18</td>
<td>7</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1</td>
<td>286</td>
<td>783</td>
<td>3</td>
<td>46</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4/14</td>
<td>1</td>
<td>1</td>
<td>205</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>5</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>14</td>
<td>656</td>
<td>2968</td>
<td>1</td>
<td>24</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4/15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>904</td>
<td>9</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>229</td>
<td>400</td>
<td>3</td>
<td>18</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4/16</td>
<td>2627</td>
<td>105</td>
<td>22</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>15</td>
<td>31</td>
<td>1</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>326</td>
<td>594</td>
<td>22</td>
<td>17</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2655</td>
<td>268</td>
<td>250</td>
<td>746</td>
<td>8</td>
<td>9</td>
<td>2</td>
<td>1006</td>
<td>142</td>
<td>1735</td>
<td>138</td>
<td>1</td>
<td>4</td>
<td>50</td>
<td>20</td>
<td>2094</td>
<td>7562</td>
<td>158</td>
<td>165</td>
<td>24</td>
<td>20</td>
</tr>
</tbody>
</table>

1  2655  Attempted Sun RPC high port access and SUNRPC highport access
2  268  Connect to 515 from inside / Connect to 515 from outside
4  250  250  External RPC call
5  746  High port 65535 TCP/UDP - Possible Red Worm - traffic
7  8  ICMP SRC and DST outside network
8  9  NMAP TCP Ping
9  2  Probable NMAP fingerprint attempt
10 1006  Possible trojan server activity
11 142  Queso fingerprint
12 1735  Russia Dynamo
13 138  SMB Name Wildcard
14 1 STATDX UDP Attack
15 4  SYN-FIN scan
16 50  TCP SRC and DST outside network
17 20  Tiny Fragments
18 2094  UDP SRC and DST outside network
19 7562  Watchlist 000220
20 158  Watchlist 000222
21 165  WINGATE 1080 Attempt
22 24  Null Scan
23 20  Port 55850 tcp - possible myserver activity
3-2-1 Attempted Sun RPC high port access / SUNRPC highport access

During the period 04/10/2001 to 04/16/2001, there were 2,628 attempts to access Port 32771 recorded in the IDS logs. While a Top Ten Talkers table is provided, the emphasis is placed on the exploit itself since 91 systems on the MY.NET.132, 46 on MY.NET.133, 42 on MY.NET.135 and 3 on MY.NET.137 were scanned during this time period.

![SUN RPC High Port Access Attempts](chart)

Table 3 - Top Ten Talkers (SUN RPC)

<table>
<thead>
<tr>
<th>Count</th>
<th>Source IP</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1508</td>
<td>24.248.185.123</td>
<td>MY.NET.219.34</td>
</tr>
<tr>
<td>1118</td>
<td>172.135.241.112</td>
<td>MY.NET.219.34</td>
</tr>
<tr>
<td>8</td>
<td>198.186.203.77</td>
<td>MY.NET.209.10</td>
</tr>
<tr>
<td>8</td>
<td>64.12.25.115</td>
<td>MY.NET.208.86</td>
</tr>
<tr>
<td>6</td>
<td>64.12.163.199</td>
<td>MY.NET.209.10</td>
</tr>
<tr>
<td>4</td>
<td>199.244.218.40</td>
<td>MY.NET.209.10</td>
</tr>
<tr>
<td>2</td>
<td>163.29.211.66</td>
<td>MY.NET.132.193</td>
</tr>
<tr>
<td>2</td>
<td>210.179.201.196</td>
<td>MY.NET.135.29</td>
</tr>
<tr>
<td>2</td>
<td>210.179.201.196</td>
<td>MY.NET.135.31</td>
</tr>
<tr>
<td>1</td>
<td>128.175.133.84</td>
<td>MY.NET.223.122</td>
</tr>
</tbody>
</table>

3-2-1-1 Description/Discussion:

Attempts to access RPC ports are of a concern because there are several well-known buffer overflow vulnerabilities in various RPC programs. Port map is usually consulted to determine what programs are running on the host before attempting to exploit a vulnerability in one of the programs that is reported (Dell, GCIA Practical). There were 2,628 attempts with a destination
port of 32771. This would indicate that the intruders were attempting to connect to this high port, which is normally used by "yppasswd" to transfer NIS passwords. Source ports include 21 (FTP), 443 (SSL), 4000 (ICQ), 5190, 8080 (Proxy), 9898, 27960, and 32768. Almost all of the activity occurring on 4/16/2001 was from 32768 to 32771.

It appears that there were one of three different rules used, or data from three sensors has been merged into one alert file. A search of the current Whitehats.com and Snort.org current SNORT Rules show that this may be one of the rules that caused this activity to be logged:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 32771 (msg:"MISC-Attempted Sun RPC high port access";)
```

This is not a scan. Of the 2,655 alerts, 2,628 were for MY.NET.219.34. A check of the Portscan Logs show that this host also generated 9,789 return packets with a destination port of 32778 (source port of 327xx). This is a known game port, and the time period in which the activity occurred supports the game conclusion.

3-2-1-2 Correlation(s):

A Keyword ("RPC") search of the Security Focus Vulnerabilities database reveals that this problem has been around since early 1992. Multiple RPC Services (NIS, ToolTalk, SMB, and Portmapper to name a few) on several Operating Systems are affected. Several CERT Advisories warn of problems in specific RPC Services as well.


3-2-1-3 Defenseive Recommendations:

First, if you don’t need the service, then remove or disable it. Second, install ALL patches for the RPC Services you are running. Third, continue to monitor access to all RPC Services ports.
3-2-2 Connect to 515 from inside and Connect to 515 from outside

During the period 04/10/2001 to 04/16/2001, there were 268 attempts to access Port 515 recorded in the IDS logs. The table below shows the number of attempts made by every external host attempting to connect to port 515. There were four attempts by MY.NET hosts to connect to port 515 outside the MY.NET network.

![Graph showing Connect to Port 515 from Inside & Outside of the Network]

<table>
<thead>
<tr>
<th>Table 4 - Top Talkers (Connect to 515)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong></td>
</tr>
<tr>
<td>141</td>
</tr>
<tr>
<td>53</td>
</tr>
<tr>
<td>39</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

3-2-2-1 Description/Discussion:

This is a scan of port 515 on systems on the MY.NET.133.0/32, MY.NET.134.0/32, MY.NET.135.0/32, and MY.NET.137.0/32 subnets.

This is looking for any connections to port 515 which is the Line Printer Daemon (LPD) or Print Spooler. This is an alert for a possible Denial Of Service (DoS) attack. This problem also exists in...
HP JetDirect Firmware x.08.20 and earlier (CAN-2000-1064). It is not restricted to any one operating system.

From the Vigilante WinCom LPD Advisory\(^\text{30}\):

“A continuous stream of LPD options, sent to the LPD port (default TCP port 515) on the host running WinCOM, will eventually consume all the memory on that host”

A search of the current Whitehats.com and Snort.org SNORT Rule Sets did not reveal an exact match to a rule for these events. These are probably the rules for this event:

```plaintext
alert tcp $EXTERNAL any -> $INTERNAL 515 (msg:"Connect to 515 from outside");
alert tcp $INTERNAL any -> $INTERNAL 515 (msg:"Connect to 515 from inside");
```

3-2-2-2 Correlation(s):

A search of the Consensus Intrusion Database\(^\text{31}\) (CID) at Incidents.org for the time period covered by this analysis shows 34 reported incidents of attempted access to port 515. On 6 July 2001, the All Destination Ports Sorted by How Many in the Past 30 Days Chart\(^\text{32}\) at Incidents.org showed there were 946,830 reported attempts to access port 515 in the past 30 days.

There are several Computer Vulnerabilities and Exposures (CVE), CVE Candidates (CAN), and Cert Advisories providing information on Port 515 Line Printer Daemon (LPD) vulnerabilities.

- Box Network - http://neworder.box.sk/showme.php3?id=5025
- Box Network - http://neworder.box.sk/showme.php3?id=2846

3-2-2-3 Defense Recommendations:

First, install ALL patches for the RPC Services you are running. Most Port 515 vulnerabilities and exploits are linked to an RPC Service which can be used to compromise systems. Second, unless you must leave this port open, block it at your border Routers or use Router Access Control Lists to control access to this Port from outside your Intranet. Third, continue to monitor access to all RPC Services ports.

---

\(^{30}\) WinCOM LPD DoS http://www.vigilante.com/netsecurity/advisories/VIGILANTE-20000013.htm

\(^{31}\) WWW INCIDENTS.ORG, Search the Consensus Intrusion Database (CID) http://www.incidents.org/cid/search.php

\(^{32}\) WWW INCIDENTS.ORG, All Destination Ports Sorted by How Many for past 30 days http://www.incidents.org/cid/query/top_port_numc_30.php
3-2-3 External RPC Call

During the period 04/10/2001 to 04/16/2001, there were 250 External RPC Call attempts recorded in the IDS logs.

![External RPC Call Chart]

### Table 5 - External Sources

<table>
<thead>
<tr>
<th>Count</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>210.179.201.196</td>
</tr>
<tr>
<td>64</td>
<td>216.36.36.29</td>
</tr>
<tr>
<td>22</td>
<td>209.247.201.144</td>
</tr>
<tr>
<td>19</td>
<td>200.230.39.5</td>
</tr>
<tr>
<td>15</td>
<td>163.29.211.66</td>
</tr>
<tr>
<td>7</td>
<td>211.46.206.9</td>
</tr>
<tr>
<td>1</td>
<td>24.50.67.77</td>
</tr>
</tbody>
</table>

3-2-3-1 Description/Discussion:

The Remote Procedure Call (RPC) protocol (RFC1831[^33]) is a means by which a host can execute code on a remote host. This appears to be a scan for the SUN Portmapper RPC Service. All RPC Services must register with the Portmapper Service and scanning for this service can provide valuable reconnaissance data, such as a list of the RPC Services registered on the system. Once this list is obtained, an attacker can just pick his favorite exploit to compromise the host. Hosts on the MY.NET.132.0/32, MY.NET.133.0/32, MY.NET.134.0/32, MY.NET.135.0/32, and

MY.NET.137.0/32 subnets were all scanned for port 111 by seven separate hosts. There was one odd connection to MY.NET.5.5 from 216.36.36.29 which occurred 09:55:00. This same external host started to scan 61 hosts on the MY.NET.134.0/32 and MY.NET.135.0/32 subnets in one second at 09:55:12. A reply from any host on the MY.NET network was not detected.

Table 5 shows the number of hosts scanned by each external host causing this alert. A total of 231 MY.NET hosts were scanned 250 times by these seven hosts. This is active reconnaissance.

A search of the Whitehats.com and Snort.org current rule set did not find an exact match for this event. The following rules will produce the log entries we are seeing here:

```plaintext
alert tcp $EXTERNAL any -> $INTERNAL 111 (msg:"External RPC Call");
alert udp $EXTERNAL any -> $INTERNAL 111 (msg:"External RPC Call");
```

### 3-2-3-2 Correlation(s):

A search of the Consensus Intrusion Database\(^{34}\) (CID) at Incidents.org for the time period covered by this analysis shows 289 reported incidents of attempted access to port 111. A search of the Consensus Intrusion Database (CID\(^ {35}\)) for any Source IP and Port to any Destination IP and Port 111 between 01 July 2001 and 06 July 2001 yielded 118,155 matches. Also on 7 July 2001, the All Destination Ports Sorted by How Many in the Past 30 Days Chart\(^ {36}\) at Incidents.org showed there were 854,867 reported attempts to access port 111 in the past 30 days.

Again this vulnerability has been around since 1992. There are several CVE’s, CAN’s, CERT Alerts, and vendor advisories about this exploit and patches for them. Here are a couple.


### 3-2-3-3 Defensive Recommendations:

First, install ALL patches for the RPC Services you are running. Portmapper is a free ticket to a complete list of all RPC Services registered on a host if not properly patched. Once compromised, you no longer own the system. Second, unless you must leave this port open, block it at your border Routers or use Router Access Control Lists to control access to this Port from outside your Intranet. Third, continue to monitor access to all hosts with the Portmapper service active.

---

\(^{34}\) [WWW INCIDENTS.ORG](http://www.incidents.org), Search the Consensus Intrusion Database (CID) [http://www.incidents.org/cid/search.php](http://www.incidents.org/cid/search.php)


3-2-4 High port 65535 TCP/UDP – Possible Red Worm – traffic

During the period 04/10/2001 to 04/16/2001, there were 746 attempts to access port 65535 recorded in the IDS logs.

![High port 65535 TCP/UDP Traffic - Possible Red Worm](image)

### Table 6 - High Port 65535 TCP/UDP Top Five Talkers

<table>
<thead>
<tr>
<th>Count</th>
<th>Source IP</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>139</td>
<td>12.13.129.141</td>
<td>MY.NET.97.175</td>
</tr>
<tr>
<td>70</td>
<td>198.111.138.20</td>
<td>MY.NET.207.118</td>
</tr>
<tr>
<td>44</td>
<td>129.59.51.185</td>
<td>MY.NET.207.54</td>
</tr>
<tr>
<td>17</td>
<td>129.59.51.185</td>
<td>MY.NET.210.130</td>
</tr>
<tr>
<td>14</td>
<td>129.59.51.185</td>
<td>MY.NET.204.66</td>
</tr>
</tbody>
</table>

### 3-2-4-1 Description:

Research revealed a message\(^{37}\) posted by the Emory University Security Office from the Computer Security Office at Dartmouth University to UNISOG at SANS.ORG, here is an excerpt from that posting:

> A trojan'd klogd is compiled and set running on port 65535 waiting for an incoming packet with a data size of 77 bytes

Both SANS\(^{38}\) and Dartmouth University\(^{39}\) describe the exploit as:

---

\(^{37}\) Theory Group, Adore/Red Worm Message Posted 03 April 2001  
[http://theorygroup.com/Archive/Unisog/2001/msg00492.html]
Adore worm replaces only one system binary (ps), with a trojaned version and moves the original to /usr/bin/adore. It installs the files in /usr/lib/lib . It then sends an email to the following addresses: adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com, adore9001@sina.com

Attempts have been made to get these addresses taken offline, but no response so far from the provider. It attempts to send the following information:

/etc/ftpusers
ifconfig
ps -aux (using the original binary in /usr/bin/adore)
/root/.bash_history
/etc/hosts
/etc/shadow

Adore then runs a package called icmp. With the options provided with the tarball, it by default sets the port to listen too, and the packet length to watch for. When it sees this information it then sets a rootshell to allow connections. It also sets up a cronjob in cron daily (which runs at 04:02 am local time) to run and remove all traces of its existence and then reboots your system. However, it does not remove the backdoor.

I only found one reference to the phrase “Red Worm”, but it is most often referred to by its new name ‘Adore Worm’. Everyone agrees that the current Snort Rule set will detect this worm, this alert seems to be one that logs all traffic on port 65535/TCP and 65535/UDP.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Source IP</th>
<th>SRC Port</th>
<th>Destination IP</th>
<th>DST Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/12</td>
<td>12:48:05</td>
<td>MY.NET.253.53</td>
<td>65535</td>
<td>209.36.43.131</td>
<td>25</td>
</tr>
<tr>
<td>04/12</td>
<td>12:48:05</td>
<td>MY.NET.253.53</td>
<td>65535</td>
<td>209.36.43.131</td>
<td>25</td>
</tr>
<tr>
<td>04/14</td>
<td>16:50:54</td>
<td>MY.NET.253.24</td>
<td>65535</td>
<td>206.106.64.12</td>
<td>25</td>
</tr>
<tr>
<td>04/14</td>
<td>16:50:54</td>
<td>MY.NET.253.24</td>
<td>65535</td>
<td>206.106.64.12</td>
<td>25</td>
</tr>
<tr>
<td>04/14</td>
<td>16:50:54</td>
<td>MY.NET.253.24</td>
<td>65535</td>
<td>206.106.64.12</td>
<td>25</td>
</tr>
<tr>
<td>04/14</td>
<td>16:50:54</td>
<td>MY.NET.253.24</td>
<td>65535</td>
<td>206.106.64.12</td>
<td>25</td>
</tr>
<tr>
<td>04/11</td>
<td>18:47:11</td>
<td>MY.NET.100.230</td>
<td>65535</td>
<td>12.6.145.21</td>
<td>25</td>
</tr>
<tr>
<td>04/11</td>
<td>18:47:11</td>
<td>MY.NET.100.230</td>
<td>65535</td>
<td>12.6.145.21</td>
<td>25</td>
</tr>
</tbody>
</table>

There were 371 entries for the TCP Protocol and 376 for the UDP Protocol. Since the Trojan sends email and then listens on port 65535 we should first check for inbound packets to any MY.NET Host with a destination port of 65535. None of the captured packets met that criteria. There were a lot of packets from port 65535. There were no inbound packets to port 65535. A check of outbound Port 25/TCP (SMTP) packets from hosts on the MY.NET network reveals two hosts that each sent two packets each and one

host that sent four packets.

A NSLookup showed that none of the destination hosts were in the 21cn.com or sina.com domains.

**Trying 209.36.43 at ARIN**

AT&T (NETBLK-WORLDNET-MIS2) WORLDNET-MIS2 209.36.0.0 - 209.37.255.255
Giant Food Inc (NETBLK-GIANTFOOD-43) GIANTFOOD-43 209.36.43.0 - 209.36.43.255

**Trying 206.106.64 at ARIN**

US Sprint (NETBLK-NETBLK-SPRINT-BLKG) NETBLK-SPRINT-BLKG
206.104.0.0 - 206.107.255.255
Hoosiers Net, Inc. (NETBLK-SPRINT-CE6A7F) SPRINT-CE6A7F
206.106.64.0 - 206.106.127.255

**Trying 12.6.145 at ARIN**

AT&T ITS (NET-ATT) ATT 12.0.0.0 - 12.255.255.255
CONCERT GLOBAL NETWORKS (NETBLK-CONCERT-145) CONCERT-145
12.6.145.0 - 12.6.145.255

There seems to be some ToolTalk activity between MY.NET.97.175 and 12.13.129.141. Host 12.13.129.141 used port 6112 (Registered to the dtspcd service according to the IANA port list[^40]) to communicate to port 65535 on MY.NET.97.175. A dtspcd vulnerability was reported as part of a Common Desktop Environment (CDE) in CERT Advisory CA-1999-11[^41].

**Trying 12.13.129 at ARIN**

AT&T ITS (NET-ATT) ATT 12.0.0.0 - 12.255.255.255
MULTIPRO NETWORK (NETBLK-MULTIPRO50-129) MULTIPRO50-129
12.13.129.0 - 12.13.129.255

The second host (198.111.138.20, registered as part of Alma College network) listed appears is using port 4443 (registered to pharos according to the IANA port list) to communicate to port 65535 on MY.NET.207.118. Other than a link to a distributed printing management package manufactured by Pharos ([http://www.pharos.com](http://www.pharos.com)) I could find no other information on this. If the Pharos distributed printing management system is in use on this system, then there should be more traffic than just this one host. Investigate this host further.

**Trying 198.111.138 at ARIN**

[^40]: IANA Port List, [http://www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers)
The third host (129.59.51.185, registered as part of the Vanderbilt University network) is listed three times showing communications to three different hosts on the MY.NET network. Multiple ports are used to send data to port 65535 on all three MY.NET hosts.

**Trying 129.59.51 at ARIN**

Vanderbilt University (NET-VANDERBILT)
Computer Center
Box 1577, Station B
Nashville, TN 37235
US

Netname: VANDERBILT
Netblock: 129.59.0.0 - 129.59.255.255
Coordinator:
Zafar, Esfandiar (EZ8-ARIN) zafar@CTRVAX.VANDERBILT.EDU
(615) 343-1610

Domain System inverse mapping provided by:
IP-SRV1.VANDERBILT.EDU 129.59.1.10
IP-SRV2.VANDERBILT.EDU 129.59.2.10
PUNCH.UTCC.UTK.EDU 128.169.201.2

Record last updated on 12-Dec-1996.
Database last updated on 14-Jul-2001 23:02:13 EDT.

### 3-2-4-2 Correlation(s):

A search of the Consensus Intrusion Database\(^42\) (CID) at Incidents.org for the time period covered by this analysis shows 1 reported incident of attempted access to port 65535. A search of the Consensus Intrusion Database (CID\(^43\)) for any Source IP and Port to any Destination IP and Port 65535 between 01 July 2001 and 06 July 2001 yielded 27 matches.

As stated on the SANS and Dartmouth web sites, a more detailed analysis of the Adore package was done by Michael Reiter, GCIH\(^44\) in his practical entitled Exploiting Loadable Kernel Modules.

Links to related CVE’s, CAN’s and CERT bulletins can be found on the SANS and Dartmouth

---

\(^{42}\) [WWW INCIDENTS.ORG, Search the Consensus Intrusion Database (CID) http://www.incidents.org/cid/search.php](http://www.incidents.org/cid/search.php)


\(^{44}\) Reiter, Michael, GCIH Practical, Exploiting Loadable Kernel Modules, [http://www.sans.org/y2k/practical/Michael_Reiter_GCIH.zip](http://www.sans.org/y2k/practical/Michael_Reiter_GCIH.zip)
University web sites as well. A variant was reported to SANS by Lance Dillon\(^{45}\) on 04/10/2001.

### 3-2-4-3 Defensive Recommendations:

Applying all recommended BIND patches from your vendor is the recommended defense against this threat. Blocking all outbound e-mail to the four e-mail addresses should also be done if possible. Use the information on the SANS web site concerning the Lion Worm\(^{46}\) protection measures to protect a host that cannot be updated or patched. This appears to be a detect for the original Red Worm.

A lot has changed since Red Worm (now called the Adore Worm) first appeared and while this detect will alert you of a possible compromise by the original Red Worm, the current Snort Rules available at Whitehats.com and Snort.org provide more refined detection capabilities for this and the new variants of the Adore Worm. The current rule provides a lot of data. Re-evaluate your current Snort Rules Set and consider replacing this rule or dropping it if the new rules set will provide better and more efficient coverage.

---

\(^{45}\) Red Worm Variants Reported, Daily Incidents Analyzed, SANS. [http://www.sans.org/y2k/041001.htm](http://www.sans.org/y2k/041001.htm)

\(^{46}\) Lion Worm v0.1, Chris Benton, SANS, 26 Mar 2001. [http://www.sans.org/y2k/lion_protection.htm](http://www.sans.org/y2k/lion_protection.htm)
3-2-5  NMAP TCP Ping

During the period 04/10/2001 to 04/16/2001, there were 9 NMAP TCP Ping events recorded in the IDS Logs.

![NMAP TCP Ping Chart]

3-2-5-1 Description/Discussion:

From the NMAP47 Manpage:

“Nmap is designed to allow system administrators and curious individuals to scan large networks to determine which hosts are up and what services they are offering. nmap supports a large number of scanning techniques such as: UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), Reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep, IP Protocol, and Null scan. nmap also offers a number of advanced features such as remote OS detection via TCP/IP fingerprinting, stealth scanning, dynamic delay and retransmission calculations, parallel scanning, detection of down hosts via parallel pings, decoy scanning, port filtering detection, direct (non-portmapper) RPC scanning, fragmentation scanning, and flexible target and port specification.”

There were nine recorded alerts from six external hosts to six internal hosts on the MY.NET Network. All were incoming packets. No one responded to the packets. All packets had a source port of 80. Five packets had a destination port of 53, the remaining packets had a destination port of 80. The alert log entries do not indicate that any flags were set. There was no

other traffic from or to the six originating hosts found in the alerts or portscan logs.

Table 7 - NMAP TCP Ping Connections

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Port</th>
<th>NSLookup</th>
<th>Destination IP</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>194.133.58.129</td>
<td>80</td>
<td>bestroute2-t.alcatel.fr</td>
<td>MY.NET.1.4</td>
<td>53</td>
</tr>
<tr>
<td>194.133.58.129</td>
<td>80</td>
<td>bestroute2-t.alcatel.fr</td>
<td>MY.NET.1.3</td>
<td>53</td>
</tr>
<tr>
<td>194.133.58.129</td>
<td>80</td>
<td>bestroute2-t.alcatel.fr</td>
<td>MY.NET.1.5</td>
<td>53</td>
</tr>
<tr>
<td>202.187.24.3</td>
<td>80</td>
<td>No reverse DNS</td>
<td>MY.NET.253.125</td>
<td>80</td>
</tr>
<tr>
<td>202.187.24.3</td>
<td>80</td>
<td>No reverse DNS</td>
<td>MY.NET.1.3</td>
<td>53</td>
</tr>
<tr>
<td>12.40.36.194</td>
<td>80</td>
<td>No reverse DNS</td>
<td>MY.NET.1.5</td>
<td>53</td>
</tr>
<tr>
<td>199.197.130.21</td>
<td>80</td>
<td>No reverse DNS</td>
<td>MY.NET.253.125</td>
<td>80</td>
</tr>
<tr>
<td>63.117.235.7</td>
<td>80</td>
<td>No reverse DNS</td>
<td>MY.NET.100.165</td>
<td>80</td>
</tr>
<tr>
<td>207.30.174.254</td>
<td>80</td>
<td>No reverse DNS</td>
<td>MY.NET.157.150</td>
<td>80</td>
</tr>
</tbody>
</table>

Trying 194.133.58 at RIPE

% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:      194.133.0.0 - 194.133.255.255
netname:      EU-GLOBALONE-OTHER-970109
descr:        Allocated Block
descr:        Provider Local Registry
descr:        this allocation was transferred from eu.sprint
country:      EU
admin-c:      PW269-RIPE
tech-c:       CC3641-RIPE
status:       Allocated PA
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    AS4000-MNT
changed:      hostmaster@ripe.net 19970109
changed:      hostmaster@ripe.net 19980615
changed:      hostmaster@ripe.net 19990510
changed:      hostmaster@ripe.net 19990826
changed:      hostmaster@ripe.net 20000919
source:       RIPE

route:        194.133.58.0/24
descr:        Alcanet
origin:       AS2917
mnt-by:       OLEANE-NOC
changed:      hostmaster@oleane.net 20000302
source:       RIPE
person: Peter Wilmot
address: Equant
address: 13775 McLearen Road
address: Oak Hill, VA 20171
address: USA
phone: +01 703 471-2633
fax-no: +01 703 471-3380
e-mail: peter.wilmot@equant.com
nic-hdl: PW269-RIPE
mnt-by: AS4000-MNT
changed: castelli@hq.si.net 19990408
changed: richard.obengmarnu@globalone.net 19991015
changed: tfischer@rain.fr 20010709
source: RIPE

person: Carrie Costa
address: Equant
address: 13775 McLearen Road
address: Oak Hill, VA 20171
address: USA
phone: +01 703 471-3366
fax-no: +01 703 478-7852
e-mail: Carrie.Costa@equant.com
nic-hdl: CC3641-RIPE
mnt-by: AS4000-MNT
changed: richard.obengmarnu@globalone.net 20000420
changed: tfischer@rain.fr 20010709
source: RIPE

whois -h whois.apnic.net 202.187.24.3 ...

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois6.apnic.net)

netname: JARING-UNITAR2
descr: Universiti Tun Abdul Razak
descr: Plaza CCL, Jalan SS 6/12
descr: Kelana Jaya Urban Centre
descr: 47300 Petaling Jaya Selangor
country: MY
admin-c: AR28-AP
tech-c: AR28-AP
notify: dbmon@apnic.net
notify:    ip-request@jaring.my
mnt-by:   MAINT-JARING-AP
changed:  ip-request@jaring.my 20000509
source:   APNIC

person:   Abdul Razal
address:  Universiti Tun Abdul Razak(410764-P)
address:  Plaza CCL, Jalan SS 6/12
address:  Kelana Jaya Urban Centre
address:  47300 Petaling Jaya Selangor
country:  MY
phone:    +60-3-709-2009
fax-no:   +60-3-704-4421
e-mail:   razal@unitar.edu.my
nic-hdl:  AR28-AP
remarks:  jaring-unitar2
notify:    ip-request@jaring.my
mnt-by:   MAINT-JARING-AP
changed:  ip-request@jaring.my 20000508
source:   APNIC

Trying 12.40.36 at ARIN
AT&T ITS (NET-ATT) ATT 12.0.0.0 - 12.255.255.255
FAIRBANKS SCALES (NETBLK-FANCOR-36-0) FANCOR-36-0 12.40.36.0 - 12.40.36.63
EZIAZ, INC. (NETBLK-SL411-36-64)SL411-36-64 12.40.36.64 - 12.40.36.79
MULTIVAC INC (NETBLK-ATT-MULTIVAC722-36-80) ATT-MULTIVAC722-36-80 12.40.36.80 - 12.40.36.95
DUNBROOKE INC (NETBLK-ATT-36-96)ATT-36-96 12.40.36.96 - 12.40.36.111
CENTRAL STATES THERMAL KING (NETBLK-ATT194154-36-112) ATT194154-36-112 12.40.36.112 - 12.40.36.127
R&D TOOL & ENGINEERING (NETBLK-RDTOOL-36-128) RDTOOL-36-128 12.40.36.128 - 12.40.36.159
WHITE INDUSTRIES (NETBLK-ATT21216-36-160) ATT21216-36-160 12.40.36.160 - 12.40.36.191
HELZBERG DIAMONDS (NETBLK-ATT547-36-192) ATT547-36-192 12.40.36.192 - 12.40.36.199
FARMERS INSURANCE GROUP (NETBLK-FARMERS-IN950-36-200) FARMERS-IN950-36-200 12.40.36.200 - 12.40.36.207
THE MANAGEMENT NETWORK GROUP (NETBLK-A740-36-208) A740-36-208 12.40.36.208 - 12.40.36.223
NETPULSE (NETBLK-NETPULSE-36-224) NETPULSE-36-224 12.40.36.224 - 12.40.36.255
To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

**Trying 199.197.130 at ARIN**
Corning Incorporated (NETBLK-CORNING-CBLK)
  Corning Incorporated
  SP-WW-01-1
  Corning, NY 14831
  US

  Netname: CORNING-CBLK
  Netblock: 199.197.128.0 - 199.197.255.255

  Coordinator:
    Corning Incorporated (ZC107-ARIN) dnsadmin@CORNING.COM
    607-974-9000

  Domain System inverse mapping provided by:

    NS1.CORNING.COM  199.197.130.3
    NS2.CORNING.COM  199.197.135.4
    NS3.CORNING.COM  199.197.135.3
    NS4.CORNING.COM  199.197.130.4

  Record last updated on 29-Jan-2001.
  Database last updated on 14-Jul-2001 23:02:13 EDT.

**Trying 63.117.235 at ARIN**
UUNET Technologies, Inc. (NETBLK-UUNET63) UUNET63 63.64.0.0 - 63.127.255.255
  Manpower International (NETBLK-UU-63-117-235) UU-63-117-235
    63.117.235.0 - 63.117.235.63
  EON Communications (NETBLK-UU-63-117-235-64) UU-63-117-235-64
    63.117.235.64 - 63.117.235.79
  American Ink and Coa (NETBLK-UU-63-117-235-80) UU-63-117-235-80
    63.117.235.80 - 63.117.235.95
  Ibrite, Inc. (NETBLK-UU-63-117-235-96) UU-63-117-235-96
    63.117.235.96 - 63.117.235.103
  Ibrite, Inc. (NETBLK-UU-63-117-235-112) UU-63-117-235-112
    63.117.235.112 - 63.117.235.119

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

**Trying 207.30.174 at ARIN**
Sprint/United Telephone of Florida (NETBLK-UTELFLA-DOM) UTELFLA-DOM
207.30.0.0 - 207.30.255.255
Rollins College (NETBLK-ROLLINS2) ROLLINS2 207.30.174.0 - 207.30.174.255

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

It appears that MY.NET.1.3, MY.NET.1.4 and MY.NET.1.5 are DNS Servers. The remaining three hosts MY.NET.253.125, MY.NET.100.165 and MY.NET.157.150 may be web servers. This traffic appears to be load-balancing queries.

3-2-5-2 Correlation(s):

This appears to be Load-balancing. Additional information is required to completely confirm this. A search at google.com yielded these links that discuss the type of traffic you see here.


A search of the Consensus Intrusion Database (CID) for any Source IP and Source Port 80 to any Destination IP and Port 53 between 01 July 2001 and 06 July 2001 yielded 8 matches. Of these 8 matches, one of them (199.197.130.21) is included in our list above. A search for any Source IP and Source Port 80 to any Destination IP and Destination Source Port 80 produced 12 matches. Two of those twelve matches were from a single host (202.187.24.3) also contained in our list above. This further supports the fact that what we are seeing is Load-balancing.

3-2-5-3 Defensive Recommendations:

Countermeasures for Load-balancing include steps to secure your Domain Name Servers and your Web Servers. Load-balancing is not malicious, but the fact that load balancing traffic is difficult to distinguish from other malicious traffic and the fact that Load-balancing developers are continuously developing ways to by pass firewalls in an effort to enhance their products performance and reliability means you should always be on your guard when you see this type of traffic on your network. Countermeasures for NMAP include blocking all outbound ICMP Unreachable messages at your border routers.

---

3-2-6 Probable NMAP fingerprint activity

During the period 04/10/2001 to 04/16/2001, there were 2 recorded events of *Probable NMAP fingerprint activity* in the IDS Logs. Both of these events occurred on 4/13/2001.

![Probable NMAP fingerprint attempt]

### 3-2-6-1 Description/Discussion:

From the NMAP Manpage:

“Nmap is designed to allow system administrators and curious individuals to scan large networks to determine which hosts are up and what services they are offering. nmap supports a large number of scanning techniques such as: UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), Reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep, IP Protocol, and Null scan. nmap also offers a number of advanced features such as remote OS detection via TCP/IP fingerprinting, stealth scanning, dynamic delay and retransmission calculations, parallel scanning, detection of down hosts via parallel pings, decoy scanning, port filtering detection, direct (non-portmapper) RPC scanning, fragmentation scanning, and flexible target and port specification.”

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Port NSLookup</th>
<th>Destination IP</th>
<th>Port</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.42.5.159</td>
<td>2055cable005159.ciudad.com.ar</td>
<td>MY.NET.221.134</td>
<td>6346</td>
<td>Gnutella</td>
</tr>
<tr>
<td>212.171.49.18</td>
<td>958 See whois information belowMY.NET.223.206</td>
<td>57575</td>
<td>No-Record</td>
<td></td>
</tr>
</tbody>
</table>

---

The connection data is shown in the previous table, NSLookups follow. The first connection listed is probably a Gnutella user. You may want to check the MY.NET.221.134 for Gnutella software. The second connection shown is from port 958 to 57575. The reverse lookup on the IP Address failed, but the Whois lookup revealed that this IP is owned by an Italian ADSL Company.

**Trying 200.42.5 at ARIN**
Prima S.A. (NETBLK-PRIMA-BLK-1) PRIMA-BLK-1 200.42.0.0 - 200.42.127.255
MultiCanal S.A. (NETBLK-PRIMA-BLK-134) PRIMA-BLK-134 200.42.5.0 - 200.42.5.255

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

**Trying 212.171.49 at RIPE**
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 212.171.48.0 - 212.171.49.255
netname: TIN
descr: Telecom Italia Net
descr: TIN ADSL service in OSPF Area 06
descr: PROVIDER
country: IT
admin-c: TAS10-RIPE
tech-c: TAS10-RIPE
status: ASSIGNED PA
remarks: Please send abuse notification to abuse@tin.it
notify: nettin@tin.it
mnt-by: TIN-MNT
changed: cgiadmin.cgi.interbusiness.it 19991215
changed: nettin@tin.it 20010212
source: RIPE

route: 212.171.0.0/16
descr: INTERBUSINESS
origin: AS3269
mnt-by: INTERB-MNT
changed: cgiadmin.cgi.interbusiness.it 19990524
source: RIPE
This host also shows up in the Portscan logs talking to the same MY.NET.223.206 host. Here is the log entry:

```plaintext
This host also shows up in the Portscan logs talking to the same MY.NET.223.206 host. Here is the log entry:
```
He shows up three times in the Out-Of-Spec logs as well:

```
04/13-06:17:31.071408 212.171.49.18:33589 -> MY.NET.223.206:38469
TCP TTL:47 TOS:0x0 ID:1387 DF
*1SFR*** Seq: 0x2465EF4B   Ack: 0x80184401   Win: 0x0
F4 4E
04/13-06:21:52.565063 212.171.49.18:964 -> MY.NET.223.206:16105
TCP TTL:47 TOS:0x0 ID:13460 DF
2*SFRP*U Seq: 0xC0D5215   Ack: 0x501821CC   Win: 0x0
CE B8 00 00 FE D1 5A CC A2 92 BE 41 80 2F 16 28   ......Z....A./.(C4 87
04/13-07:41:46.798578 212.171.49.18:958 -> MY.NET.223.206:57575
TCP TTL:47 TOS:0x0 ID:60351 DF
**SF*P*U Seq: 0xE5C8DA13   Ack: 0x50181EEC   Win: 0x0
50 18 1E EC 23 2B 00 00 F5 7D 00 00 76 3F DC 92 P...#+...}..v..<59 2F D5 11 91 EA 00 B0 D0 24   Y/......$
```

This last entry is the packet from our NMAP Alert. This confirms that active targeting is taking place since this host is only sending packets to a single MY.NET host. The flags set do not conform to any normal combination of IP flags. The source and destination ports remain the same during all of these transactions that occur over an approximate two hour time period (No machine is that slow).

### 3-2-6-2 Correlation(s):

A search of the Consensus Intrusion Database (CID) for any Source IP and Source Port 958 to any Destination IP and Port between 01 July 2001 and 06 July 2001 yields 63 matches. None of these were to port 57575, and they all had the SIN flag only set. None of them were from the 212.171 subnet.

Scans with similar patterns were analyzed by:

Asadoorian, Paul GCIA (337) [http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc](http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc)

[http://www.sans.org/y2k/061000.htm](http://www.sans.org/y2k/061000.htm)

---

"...[*] IDS005 - SCAN-Possible NMAP Fingerprint attempt [*]
06/06-22:56:36.131002 213.6.15.254:38265 -> z.y.w.34:21 TCP
TTL:35 TOS:0x0 ID:30007 **SF*P*U Seq: 0x84E28727 Ack: 0x0 Win: 0xC00 TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL..."

Goodwin, P.J. GCIA (305) http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc

Example of potential source port 1 scanning
http://www.sans.org/y2k/110900-1300.htm
Nov 6 18:41:15 hostre in.telnetd[14093]: refused connect from sweetness.tamu.edu
Nov 6 18:41:15 hostre in.telnetd[14094]: refused connect from sweetness.tamu.edu
Nov 6 18:41:17 hostbe in.telnetd[29543]: refused connect from sweetness.tamu.edu
Nov 6 18:41:17 hostbe in.telnetd[29544]: refused connect from sweetness.tamu.edu
Nov 6 18:41:22 hostp portsentry[542]: attackalert: Connect from host:
sweetness.tamu.edu/165.95.63.130 to TCP port: 1
Nov 6 18:41:22 hostp portsentry[542]: attackalert: Connect from host:
sweetness.tamu.edu/165.95.63.130 to TCP port: 1
Nov 6 18:41:23 hostp portsentry[423]: attackalert: Connect from host:
sweetness.tamu.edu/165.95.63.130 to TCP port: 1
Nov 6 18:41:24 hostp rpcbind: refused connect from 165.95.63.130 to dump()
Nov 6 18:41:26 hostbe portsentry[26278]: attackalert: Connect from host:
sweetness.tamu.edu/165.95.63.130 to TCP port: 1
Nov 6 18:48:36 hoster telned[19024]: refused connect from sweetness.tamu.edu
Nov 6 18:48:36 hoster telned[6926]: refused connect from sweetness.tamu.edu
Nov 6 18:48:45 hoster portsentry[17814]: attackalert: Connect from host:
sweetness.tamu.edu/165.95.63.130 to TCP port: 1
Nov 6 18:55:27 hostmau snort[63106]: SCAN-SYN FIN: 165.95.63.130:4 -> z.y.x.28:111
Nov 6 18:55:33 hostmau snort[63106]: RPC Info Query: 165.95.63.130:1005 -> z.y.x.28:111

3-2-6-3 Defensive Recommendations:

Basic security requirements are all that is required to reduce the chances of your being fingerprinted using NMAP. Ensure that all of the latest patches have been applied to your system. Uninstall applications that are not necessary. Close or block all unneeded ports on your perimeter routers and firewalls.

If you are really worried about fingerprint scanning then consider installing NMAP and/or HPING2 on your own critical systems. You can scan them yourself so you know what the response will be and use that to fine tune your Snort rules or to tweak your border router Access Control Lists some. SnortSnarf from Silicon Defense can even automate this process for you

3-2-7 Possible Trojan server activity

During the period 04/10/2001 to 04/16/2001, there were 1006 recorded alerts of Possible Trojan server activity in the IDS Logs. An analysis of the twenty four MY.NET systems originating outbound traffic from port 27374 is below.

![Graph showing possible Trojan server activity]

3-2-7-1 Description/Discussion:

Of the twenty-four MY.NET systems showing outbound connections from port 27374, eleven of these were in response to outside stimulus. A possible compromise is indicated when a system responds to a stimulus on port 27374. These eleven should be investigated immediately. Each of the twenty-four hosts is discussed below.

This alert appears to be monitoring inbound and outbound traffic from Port 27374 (Probable SubSeven\(^{52}\)). This port is associated with several other Trojans as well. According to the Simovits Consulting Trojan Ports List\(^{53}\) The list includes: Bad Blood, Ramen, Seeker, SubSeven, SubSeven 2.1 Gold, Subseven 2.1.4 DefCon 8, SubSeven Muie, and Ttfloader. All of these Trojans are remote control or backdoor Trojans. Once a host is infected with one of these it will advertise its presence to a controlling host and then wait for instructions. This port has become a very popular open port to scan for. A host should not respond since this port is closed, if it does respond then check it for possible infection. A response does not necessarily mean that a Trojan has compromised the host, it could just be that one port was randomly selected for that particular connection; but it should be checked any. If a host originates a connection from 27374, then this is a very good indication that this host has a Trojan installed on it. Comparing the alert entries

---

\(^{52}\) SubSeven Homepage. [http://subseven.slak.org](http://subseven.slak.org) (Site was in transition of 07/10/2001).

with the Alert, Out-Of-Spec and Portscan logs shows that twenty-four hosts originated a connection to another host with a source port of 27374. If the logs show that a host responded to a stimulus I would recommend that these hosts be immediately checked for a Trojan or at least be more closely monitored.

The results of a search of the Alert, Out-Of-Spec, and Portscan logs for the twenty-four hosts showing outbound connections from port 27374 follow:

**my.net.15.178:27374**

There is no stimulus recorded, so this is not a response to a stimulus. But, an attempt should be made to determine why this host tried to send data to a host outside of the MY.NET network from port 27374.

04/16-15:18:51.426973  [**] Possible trojan server activity  [**] MY.NET.15.178:27374 -> 64.229.171.112:1379
04/16-15:20:55.210506  [**] Possible trojan server activity  [**] MY.NET.15.178:27374 -> 64.229.171.112:1418

------------------------------- Portscan Log entries.
Apr 10 11:38:34 63.163.94.13:1066 -> MY.NET.15.178:21 SYN **S*****
Apr 15 11:04:43 210.52.214.15:21 -> MY.NET.15.178:21 SYN **S*****

**my.net.202.34**

MY.NET.202.34 responded to an outside connection to port 27374, this is an indication of possible compromise and should be investigated immediately. Port 1214 was used 576 times by MY.NET.202.34 while communicating with 207.55.74.26 on port 27374. MY.NET.202.34 sent 288 packets in reply. According to the alert log files, this transaction was originated by 207.55.74.56. This is definitely a response to a stimulus and should be investigated immediately.

04/12-18:16:07.203024  [**] Possible trojan server activity  [**] MY.NET.204.142:27374 -> 62.11.130.144:2566
04/13-02:24:47.991665  [**] Possible trojan server activity  [**] 64.230.147.166:1447 -> MY.NET.204.142:27374
04/13-02:24:48.533305  [**] Possible trojan server activity  [**] 24.42.34.74:3139 ->
No compromise is indicated at this time. But, I do recommend further investigation of this host to determine why it is sending data from port 27374 to a host outside the MY.NET network. Two packets were sent from this host on 4/14/2001 from port 27374 and no response was received. Three queries to this host on 4/15/2001 and one on 4/16/2001 to see if port 27374 was open went unanswered. On 4/16/2001 there was another transmission to a host outside the MY.NET network that went unanswered. This host made at least three attempts to send a packet originating from port 27374 to a host outside the MY.NET network and did not receive a response.

04/14-12:34:01.020119  [**] spp_portscan: PORTSCAN DETECTED from MY.NET.205.218 (THRESHOLD 7 connections in 2 seconds) [**]
04/14-12:34:03.090038  [**] spp_portscan: portscan status from MY.NET.205.218: 13 connections across 10 hosts: TCP(0), UDP(13) [**]
04/14-12:34:04.788364  [**] spp_portscan: End of portscan from MY.NET.205.218 (TOTAL HOSTS:11 TCP:0 UDP:13) [**]
04/14-20:29:59.386806  [**] Possible trojan server activity [**] MY.NET.205.218:27374 -> 164.77.118.15:2413
04/14-20:30:00.571231  [**] Possible trojan server activity [**] MY.NET.205.218:27374 -> 164.77.118.15:2413
04/15-15:24:51.085985  [**] Possible trojan server activity [**] 213.46.196.72:1407 ->
MY.NET.205.218:27374
04/15-19:11:39.064042 [**] Possible trojan server activity [**] 213.46.196.72:4386 -> MY.NET.205.218:27374
04/16-03:12:30.760175 [**] Possible trojan server activity [**] 216.114.16.40:2118 -> MY.NET.205.218:27374
04/16-05:50:03.514106 [**] Possible trojan server activity [**] MY.NET.205.218:27374 -> 194.126.58.37:1601

# Portscan Log entries.
Apr 10 05:30:26 210.220.73.117:3805 -> MY.NET.205.218:21 SYN **S*****
Apr 14 12:18:38 MY.NET.205.218:1421 -> 64.91.13.11:50181 UDP
Apr 14 12:18:38 MY.NET.205.218:1427 -> 64.89.143.5:27018 UDP
Apr 14 12:18:38 MY.NET.205.218:1444 -> 64.81.70.193:443 UDP
Apr 14 12:18:38 MY.NET.205.218:1448 -> 64.81.64.197:27011 UDP
Apr 14 12:18:38 MY.NET.205.218:1498 -> 64.78.201.17:27040 UDP
Apr 14 12:18:38 MY.NET.205.218:1505 -> 64.74.59.7:23117 UDP
Apr 14 12:18:39 MY.NET.205.218:1539 -> 64.7.27.99:61526 UDP
Apr 14 12:18:39 MY.NET.205.218:1540 -> 64.7.27.99:61525 UDP
Apr 14 12:18:39 MY.NET.205.218:1541 -> 64.7.27.99:61519 UDP
Apr 14 12:18:41 MY.NET.205.218:1544 -> 64.7.27.99:61512 UDP
Apr 14 12:18:40 MY.NET.205.218:1636 -> 64.34.31.245:64844 UDP
Apr 14 12:18:40 MY.NET.205.218:1653 -> 64.249.6.250:45075 UDP
Apr 14 12:18:41 MY.NET.205.218:1352 -> 65.2.228.82:62964 UDP

No compromise is indicated at this time. There is no stimulus recorded, so this is not a response to a stimulus. I recommend further investigation to determine why this host is trying to send data to a host outside the MY.NET network using a source port of 27374.

my.net.206.106:27374
04/10-22:19:05.747452 [**] Possible trojan server activity [**] MY.NET.206.106:27374 -> 202.163.100.126:64434

# Portscan log entries.
Apr 14 07:40:59 209.178.22.233:1397 -> MY.NET.206.106:53 SYN **S*****

my.net.206.230:27374

No compromise is indicated at this time. There is no stimulus recorded, so this is not a response to a stimulus. No other activity on this port is indicated in the logs.
my.net.100.82:27374

MY.NET.100.82 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. The log entries show a response to a stimulus. On 4/15/2001, host 198.248.172.184 attempted a connection to this host on port 27374 and it received a reply. One such attempt is suspicious, but this host responded to two such attempts from the same host. No other suspicious activity was indicated. Since it responded twice to external stimulus on a known Trojan port, I would investigate further to determine why this host is responding to queries on this port. Hopefully all that it requires are some patches or that a service be turned off. If there is a service using this port, then every attempt should be made to move the service to another port.

04/15-08:02:25.553259 [**] Possible trojan server activity [**] 198.248.172.184:1699 -> MY.NET.100.182:27374
04/15-08:02:25.553403 [**] Possible trojan server activity [**] MY.NET.100.182:27374 -> 198.248.172.184:1699
04/15-08:02:26.084222 [**] Possible trojan server activity [**] 198.248.172.184:1699 -> MY.NET.100.182:27374
04/15-08:02:26.087304 [**] Possible trojan server activity [**] MY.NET.100.182:27374 -> 198.248.172.184:1699

my.net.146.51:27374

MY.NET.146.51 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. This is a response to an external stimulus. This one shows a second reply sent approximately 1.5 seconds after the first reply. There was no visible stimulus causing this second reply, but it did originate from a know Trojan port. It probably was a retry. I would investigate further to determine why this host is responding to queries on this port. If this is not a normal data exchange, then all that may be required is an operating system patch or a service be turned off. Every attempt should be made to move a service on this port to another port if one is present.

04/14-12:46:35.609896 [**] Possible trojan server activity [**] 211.56.113.59:3526 -> MY.NET.146.51:27374
04/14-12:46:35.621345 [**] Possible trojan server activity [**] MY.NET.146.51:27374 -> 211.56.113.59:3526
04/14-12:46:37.226993 [**] Possible trojan server activity [**] MY.NET.146.51:27374 -> 211.56.113.59:3526

my.net.215.34:27374

MY.NET.215.34 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately, although the gaming activity on this host may be what triggered the alert. This is a response to a stimulus. There is only one exchange on 4/12/2001,
where it sent a packet from port 27374 to 65.199.131.33 port 1512 and received an instantaneous response (a 0.000094 millisecond delay). The pattern of port scans occurring before and after this transaction does not appear to change. The UDP portscans in the 7777 to 7797 range may be Game Traffic (According to a Neohapsis posting\[^{54}\] on SNORT Game Ports this may be a game called Unreal Tournament\[^{55}\]). If this is not a normal data exchange, then all that may be required is an operating system patch or a service be turned off. Every attempt should be made to move a service on this port to another port if one is present.

\[^{55}\text{Unreal Tournament, http://www.unrealtournament.com/}\]
Apr 10 23:44:00 MY.NET.215.34:2006 -> 24.162.170.143:7778 UDP
Apr 10 23:44:00 MY.NET.215.34:2001 -> 212.224.25.206:26301 UDP
Apr 10 23:44:01 MY.NET.215.34:2003 -> 166.70.135.173:7778 UDP
Apr 10 23:44:02 MY.NET.215.34:2010 -> 128.2.153.13:7778 UDP
Apr 10 23:44:02 MY.NET.215.34:2008 -> 209.247.165.214:7778 UDP
Apr 10 23:44:02 MY.NET.215.34:2009 -> 130.89.238.28:8501 UDP
Apr 10 23:44:03 MY.NET.215.34:2006 -> 194.185.88.28:7778 UDP
Apr 10 23:44:06 MY.NET.215.34:2001 -> 194.213.72.20:7778 UDP
Apr 10 23:44:07 MY.NET.215.34:2009 -> 212.115.192.204:7778 UDP
Apr 10 23:44:07 MY.NET.215.34:2006 -> 208.163.74.51:7778 UDP
Apr 10 23:44:10 MY.NET.215.34:2005 -> 206.74.82.109:7778 UDP
Apr 10 23:44:11 MY.NET.215.34:2003 -> 216.125.250.54:7778 UDP
Apr 10 23:44:16 MY.NET.215.34:1070 -> 166.70.135.172:7777 UDP
Apr 14 07:41:08 209.178.22.233:3621 -> MY.NET.215.34:53 SYN ****
Apr 16 19:09:49 MY.NET.215.34:1327 -> 212.137.72.40:7807 UDP
Apr 16 19:09:46 MY.NET.215.34:2016 -> 151.23.31.22:20004 UDP
Apr 16 19:09:47 MY.NET.215.34:2016 -> 216.196.148.9:7778 UDP
Apr 16 19:09:47 MY.NET.215.34:2010 -> 130.89.238.28:10778 UDP
Apr 16 19:09:47 MY.NET.215.34:2003 -> 212.137.72.48:7798 UDP
Apr 16 19:09:48 MY.NET.215.34:2010 -> 66.66.50.46:7778 UDP
Apr 16 19:09:48 MY.NET.215.34:2012 -> 12.32.76.126:7778 UDP
Apr 16 19:09:49 MY.NET.215.34:2011 -> 213.140.4.75:8201 UDP

Page 68 of 181
my.net.217.198:27374

No compromise is indicated at this time. He tried sending packets to two different hosts within the space of four seconds on 04/10/2001 @ 11:36 and received no response. The remainder of the entries in the Alerts Log are portscan entries. A link between the two outbound transmissions and the port scans is not indicated. The Portscan Logs show this host is scanning port 59 on external hosts which is listed as “Any Private File Service” in the port listing on the IANA.ORG Port List web page56. As a minimum, there is probably some type of File Share service installed that should be investigated.

04/10-11:36:50.358961 [**] Possible trojan server activity [**] MY.NET.217.198:27374 -> 62.7.107.166:2966
04/10-11:36:54.425972 [**] Possible trojan server activity [**] MY.NET.217.198:27374 -> 216.252.185.108:2493
04/10-13:50:55.060834 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.217.198 (THRESHOLD 7 connections in 2 seconds) [**]

<-- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange -->
04/10-13:51:11.374798 [**] spp_portscan: End of portscan from MY.NET.217.198 (TOTAL HOSTS:14 TCP:21 UDP:0) [**]
04/10-21:33:11.272624 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.217.198 (THRESHOLD 7 connections in 2 seconds) [**]

<-- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange -->
04/10-21:34:05.198203 [**] spp_portscan: End of portscan from MY.NET.217.198 (TOTAL HOSTS:45 TCP:64 UDP:0) [**]

######################################## Portscan Log entries.

NOTE: There are 87 total entries in the Portscan Log file for this host. Only a portion of those entries are here, but this small portion show the format and destination port listed in all but one entry of the Portscan Logs for this host. The first and last entry are included in this extract since they are the only two entries that are inbound to port 21 from a host outside the MY.NET network and they are the only two inbound connections to this host. All other entries in the Portscan log are outbound connections to port 59 on hosts outside the MY.NET network.

Apr 10 05:32:14 210.220.73.117:4852 -> MY.NET.217.198:21 SYN **S*****
Apr 10 13:36:16 MY.NET.217.198:34416 -> 141.219.84.107:59 SYN **S*****
Apr 10 13:36:17 MY.NET.217.198:55438 -> 65.10.192.102:59 SYN **S*****
Apr 10 13:36:17 MY.NET.217.198:27538 -> 62.254.57.169:59 SYN **S*****

56 IANA Port List, http://www.iana.org/assignments/port-numbers
Apr 10 13:36:22 MY.NET.217.198:44054 -> 144.132.18.100:59 SYN **S*****
Apr 10 13:36:25 MY.NET.217.198:59266 -> 213.243.128.9:59 SYN **S*****
Apr 10 13:36:26 MY.NET.217.198:48232 -> 134.198.246.136:59 SYN **S*****
Apr 10 13:36:28 MY.NET.217.198:26093 -> 194.236.103.99:59 SYN **S*****
Apr 10 13:36:28 MY.NET.217.198:28088 -> 203.164.141.131:59 SYN **S*****
Apr 10 21:17:07 MY.NET.217.198:5433 -> 209.15.87.204:59 SYN **S*****
Apr 10 21:17:07 MY.NET.217.198:31871 -> 64.231.171.81:59 SYN **S*****
Apr 10 21:17:15 MY.NET.217.198:11637 -> 63.22.11.197:59 SYN **S*****
Apr 10 21:17:16 MY.NET.217.198:52941 -> 161.108.185.101:59 SYN **S*****
Apr 10 21:18:00 MY.NET.217.198:5980 -> 64.229.179.165:59 SYN **S*****
Apr 10 21:18:00 MY.NET.217.198:20783 -> 62.155.188.172:59 SYN **S*****
Apr 10 21:18:00 MY.NET.217.198:19899 -> 216.62.157.213:59 SYN **S*****
Apr 10 21:18:01 MY.NET.217.198:19457 -> 216.222.64.63:59 SYN **S*****
Apr 10 21:18:02 MY.NET.217.198:62234 -> 62.108.31.66:59 SYN **S*****
Apr 10 21:18:04 MY.NET.217.198:50853 -> 141.154.48.8:59 SYN **S*****
Apr 10 21:18:06 MY.NET.217.198:47744 -> 130.64.4.153:59 SYN **S*****
Apr 10 21:18:06 MY.NET.217.198:49279 -> 128.151.143.186:59 SYN **S*****
Apr 10 21:18:06 MY.NET.217.198:5980 -> 64.229.179.165:59 SYN **S*****
Apr 10 21:18:06 MY.NET.217.198:21861 -> 141.154.49.151:59 SYN **S*****
Apr 12 05:38:53 24.165.162.34:4626 -> MY.NET.217.198:21 SYN **S*****

my.net.222.226:27374

MY.NET.222.226 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. One incoming transmission on 4/12/2001 was not replied to. At 06:31 on 4/13/2001 the Portscan logs indicate a large amount of traffic originating from this host to port 6346 on several external hosts outside the MY.NET network. Ports 6346 and 6347 are registered as the Gnutella service on the IANA Port List web page. Another incoming transmission to port 27374 on this host was replied to on 4/14/2001 (the day after the Gnutella traffic started). If compromised, I would investigate the possibility of the compromise occurring as a result of the use of Gnutella.

04/12-22:51:25.078523 [**] Possible trojan server activity [**] 142.177.94.46:2755 ->
MY.NET.222.226:27374
04/13-06:47:31.525569 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.226 (THRESHOLD 7 connections in 2 seconds) [**]
  < SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->
04/13-06:47:42.963083 [**] spp_portscan: End of portscan from MY.NET.222.226 (TOTAL HOSTS:16 TCP:17 UDP:2) [**]
04/14-18:49:29.117566 [**] Possible trojan server activity [**] 63.20.223.197:2953 ->
MY.NET.222.226:27374
04/14-18:49:29.117656 [**] Possible trojan server activity [**] MY.NET.222.226:27374 ->

57 Gnutella, Information can be obtained at http://www.gnutellanews.com. An explanation can also be found on this site at http://www.gnutellanews.com/information/what_is_gnutella.shtml
my.net.222.50:27374

MY.NET.222.50 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. This host demonstrates similar behavior as the MY.NET.217.198 host that was discussed previously. There were several individual outbound transmissions from port 27374 on this host to external hosts outside the MY.NET network, but none of them were replied to. There was one inbound packet at 15:59:51.683303 on 4/13/2001 from 210.186.22.114 to port 27374 on this host that was not replied to. On 4/15/2001 beginning at 11:52:07 and ending at 11:52:36 there were four attempts from four hosts outside the MY.NET network to connect to port 27374 on this host. A single reply was sent to 210.186.40.161 on 4/15/2001 at 11:52:12. There were no further communications to or from this host that day or the next. This single reply on 4/15/2001 to 210.186.40.161 was a response to a stimulus.

Because of the one unsolicited transmission from port 27374 on 4/11/2001 and the single response to a query on port 27374 on 4/15/2001, along with the Gnutella traffic; I recommend further investigation to rule out the presence of a trojan.
(THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/11-11:18:32.336696 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:38 TCP:49 UDP:0) [**]
04/11-11:05:04.808014 [**] Possible trojan server activity [**] MY.NET.222.50:27374 -> 193.227.62.21:1901
04/11-18:51:40.069125 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/11-18:52:46.872557 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:75 TCP:101 UDP:0) [**]
04/12-17:40:20.207228 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/12-17:40:31.276330 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:20 TCP:23 UDP:0) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/13-11:46:18.404262 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/13-11:46:37.528364 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:30 TCP:34 UDP:0) [**]
04/13-16:16:30.044612 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/13-16:17:06.513809 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:30 TCP:44 UDP:0) [**]
04/13-16:26:29.471558 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/13-16:27:09.442191 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:77 TCP:88 UDP:0) [**]
04/13-18:01:49.486130 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]

<- SNIP! Portscan entries removed from this Alert Log extract by Harvey Lange ->

04/13-18:02:10.163775 [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:39 TCP:47 UDP:0) [**]
04/15-11:52:07.274179 [**] Possible trojan server activity [**] 203.106.156.31:1516 -> MY.NET.222.50:27374
04/15-11:52:09.274696  [**] Possible trojan server activity [**] 203.54.156.181:3842 -> MY.NET.222.50:27374
04/15-12:07:46.589185  [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]
     <- SNIP!  Portscan entries removed from this Alert Log extract by Harvey Lange ->
04/15-12:08:25.498020  [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:70 TCP:84 UDP:0) [**]
04/15-12:08:46.357444  [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]
     <- SNIP!  Portscan entries removed from this Alert Log extract by Harvey Lange ->
04/15-12:09:07.047627  [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:41 TCP:33 UDP:0) [**]
04/15-13:16:44.539476  [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]
     <- SNIP!  Portscan entries removed from this Alert Log extract by Harvey Lange ->
04/15-13:17:01.629187  [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:19 TCP:18 UDP:0) [**]
04/15-18:53:20.185574  [**] spp_portscan: PORTSCAN DETECTED from MY.NET.222.50 (THRESHOLD 7 connections in 2 seconds) [**]
     <- SNIP!  Portscan entries removed from this Alert Log extract by Harvey Lange ->
04/15-18:53:55.617003  [**] spp_portscan: End of portscan from MY.NET.222.50 (TOTAL HOSTS:107 TCP:109 UDP:0) [**]
 ################################################################# Portscan Log

NOTE: There are 641 total entries in the Portscan Log file. Only a portion of those entries are here, but this small portion show the format and destination port listed in all but one entry of the Portscan Log entries for this host. The one odd entry is from an external host using a source port of 59 to port 38309 on this host with improper flag settings.

Apr 11 11:01:57 MY.NET.222.50:26524 -> 203.103.135.162:59 SYN **S*****
Apr 11 11:01:58 MY.NET.222.50:22791 -> 216.122.40.6:59 SYN **S*****
Apr 11 11:01:58 MY.NET.222.50:38926 -> 202.67.105.229:59 SYN **S*****
Apr 11 11:01:58 MY.NET.222.50:25774 -> 141.164.72.229:59 SYN **S*****
Apr 11 11:01:57 MY.NET.222.50:13488 -> 216.3.114.65:59 SYN **S*****
Apr 11 11:01:57 MY.NET.222.50:43467 -> 216.47.42.228:59 SYN **S*****
Apr 11 11:01:57 MY.NET.222.50:60465 -> 63.225.43.228:59 SYN **S*****
Apr 11 11:01:57 MY.NET.222.50:8097 -> 202.79.126.61:59 SYN **S*****
No compromise is indicated at this time. On 4/15/2001 at 19:52:29 we see one attempt from 213.46.196.72 to contact this host on port 27374 which went unanswered. At 21:33:51 and

my.net.223.50:27374

...
21:33:52 that same day we see two unsolicited transmissions from port 27374 to 65.199.134.83. We can only assume that these packets originated from this host since there is no incoming stimulus recorded. Because of the unsolicited transmissions from port 27374, I recommend additional investigation to rule out the presence of a trojan.

04/15-19:52:29.433803 [**] Possible trojan server activity [**] 213.46.196.72:1384 -> MY.NET.223.50:27374
04/15-21:33:52.866761 [**] Possible trojan server activity [**] MY.NET.223.50:27374 -> 65.199.134.83:3448
04/16-10:53:28.336152 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.223.50 (STEALTH) [**]
04/16-10:53:30.377442 [**] spp_portscan: portscan status from MY.NET.223.50: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH [**]
04/16-10:53:33.034806 [**] spp_portscan: End of portscan from MY.NET.223.50 (TOTAL HOSTS:1 TCP:1 UDP:0) [**]
04/16-13:19:28.544533 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.223.50 (STEALTH) [**]
04/16-13:19:30.298356 [**] spp_portscan: portscan status from MY.NET.223.50: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH [**]
04/16-13:19:32.127082 [**] spp_portscan: End of portscan from MY.NET.223.50 (TOTAL HOSTS:1 TCP:1 UDP:0) [**]

# Portscan Log
Apr 16 10:39:16 MY.NET.223.50:1348 -> 64.4.44.7:443 INVALIDACK *1S**PA*
RESERVEDBITS
Apr 16 13:04:11 MY.NET.223.50:1851 -> 64.4.53.7:443 NULL ********

my.net.225.117:27374

No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 4950 on 211.56.113.59 with no response received. Since this host originated this packet and it was not a response to a stimulus, I recommend additional investigation to rule out the presence of a trojan.

04/14-05:58:40.441578 [**] Possible trojan server activity [**] MY.NET.225.117:27374 -> 211.56.113.59:4950

my.net.229.54:27374

MY.NET.229.54 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. This is a response to a stimulus. We see no further activity on this port for this system, but a response to a stimulus of port 27374 should be investigated.
No compromise is indicated at this time. On 4/12/2001 there was one packet sent which received no response. On 4/16/2001 there was a probe on port 27374 that was not replied to. There was also a Queso Fingerprint scan on 4/12/2001 recorded that was not replied to. The Queso fingerprint scan may have been a scan for Gnutella since it was directed at port 6346. I recommend further investigation of this host because of the one unsolicited transmission from a known Trojan port.

my.net.210.185:27374

No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 4950 on 211.56.112.59 with no response received. I recommend further investigation.
investigation of this host because of the one unsolicited transmission from a known Trojan port.

04/13-07:56:11.744242 [**] Possible trojan server activity [**] MY.NET.210.185:27374 -> 211.57.55.134:4236

Portscan Log entries.
Apr 10 05:31:10 210.220.73.117:3050 -> MY.NET.210.185:21 SYN **S*****

my.net.204.214:27374

MY.NET.204.214 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. This is a response to a stimulus. It has the appearance of two successful probes on port 27374. Host 208.162.229.120 queries port 27374 and receives a reply. It tries again to verify that it received a connection. Finally it terminates the connection. Because of the final push (the paranoid part of me wants to use the word instructions) from the external host with no reply sent and the portscans of port 6112 on hosts outside the MY.NET network that started approximately thirty-six hours after this transaction, I recommend further investigation of this host to determine what data was exchanged during this connection.

04/14-12:42:53.230766 [**] Possible trojan server activity [**] 208.162.229.120:1341 -> MY.NET.204.214:27374
04/14-12:42:53.231165 [**] Possible trojan server activity [**] MY.NET.204.214:27374 -> 208.162.229.120:1341
04/14-12:42:54.125075 [**] Possible trojan server activity [**] 208.162.229.120:1341 -> MY.NET.204.214:27374
04/14-12:42:54.125210 [**] Possible trojan server activity [**] MY.NET.204.214:27374 -> 208.162.229.120:1341
04/14-12:42:55.321893 [**] Possible trojan server activity [**] 208.162.229.120:1341 -> MY.NET.204.214:27374
04/14-12:42:55.321893 [**] Possible trojan server activity [**] 208.162.229.120:1341 -> MY.NET.204.214:27374
04/15-23:36:23.484863 [**] spp_portscan: PORTSCAN DETECTED from MY.NET.204.214 (THRESHOLD 7 connections in 2 seconds) [**]
04/15-23:36:26.002962 [**] spp_portscan: portscan status from MY.NET.204.214: 8 connections across 8 hosts: TCP(0), UDP(8) [**]
04/15-23:36:28.097476 [**] spp_portscan: End of portscan from MY.NET.204.214 (TOTAL HOSTS:8 TCP:0 UDP:8) [**]

Portscan Log entries.
Apr 10 01:06:27 64.48.141.163:4923 -> MY.NET.204.214:53 SYN **S*****
Apr 12 05:36:19 24.165.162.34:1335 -> MY.NET.204.214:21 SYN **S*****
Apr 15 23:21:37 MY.NET.204.214:6112 -> 63.27.117.199:6112 UDP
Apr 15 23:21:37 MY.NET.204.214:6112 -> 63.11.60.77:6112 UDP
Apr 15 23:21:37 MY.NET.204.214:6112 -> 64.243.70.233:6112 UDP
Apr 15 23:21:37 MY.NET.204.214:6112 -> 139.142.118.100:6112 UDP
my.net.98.1193:27374

No compromise is indicated at this time. Shows two outbound connection attempt from port 27374 to port 4058 on 160.79.161.215 with no response received. Since this host originated this packet and it was not a response to a stimulus, I recommend additional investigation to rule out the presence of a trojan.

04/12-20:05:35.229321 [**] Possible trojan server activity [**] MY.NET.98.193:27374 -> 160.79.161.215:4058
04/12-20:05:36.414853 [**] Possible trojan server activity [**] MY.NET.98.193:27374 -> 160.79.161.215:4058

my.net.163.94:27374

No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 4058 on 202.97.219.158 with no response received. Since this host originated this packet and it was not a response to a stimulus, I recommend additional investigation to rule out the presence of a trojan. Also, almost twenty-five hours later a packet is sent to this same host from my.net.217.113 with the same results (no reply).

04/12-06:43:55.439462 [**] Possible trojan server activity [**] MY.NET.163.94:27374 -> 202.97.219.158:3326

my.net.217.113:27374

No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 2239 on 202.97.219.158 with no response received. Since this host originated this packet and it was not a response to a stimulus, I recommend additional investigation to rule out the presence of a trojan. Also, why is this host sending to the same host as my.net.163.94? Further investigation is required.


my.net.60.152:27374

MY.NET.60.152 responded to a direct connection to port, this indicates a possible compromise and should be investigated immediately. This is a response to a stimulus sent by 202.7.184.182. Three packets were received at one second intervals before a reply was sent. There was time to reply between each packet sent, why did our host wait so long to reply? It should not have replied at all if this was a probe. If it was a probe, was the final packet received crafted in such a manner as to prompt the reply? If the answer to the last question is yes, then we may have a victim of a buffer overflow or some other exploit. In any case, further investigation is warranted.
No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 113 on 207.46.186.184 with no response received. This may be an attempt to connect to the ident port on 207.46.186.184. RFC 1413 indicates states that invalid queries may be dropped by the receiving host without sending a response and this may be the case. It is better if we are cautious and investigate this host a little more.

No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 1237 on 64.78.235.14 with no response received. Since this host originated this packet and it was not a response to a stimulus, I recommend additional investigation to rule out the presence of a Trojan.

---

my.net.99.15:27374

No compromise is indicated at this time. Shows one outbound connection attempt from port 27374 to port 2665 on 211.234.149.52 with no response received. Since this host originated this packet and it was not a response to a stimulus, I recommend additional investigation to rule out the presence of a Trojan

04/15-13:06:44.331538  [**] Possible trojan server activity [**] MY.NET.99.15:27374 -> 211.234.149.52:2665

3-2-7-2 Correlation(s):

CERT\(^59\) has been sending alerts about Trojan Horses for years. CERT bulletin 1999-02 (http://www.cert.org/advisories/CA-1999-02.html) contains a short list and protective measures for each.

Here is a link from a search of the Neohapsis Archives\(^60\):

A search of the Consensus Intrusion Database (CID\(^61\)) for any Source IP and Source Port to any Destination IP and Destination Port 27374 between 10 April 2001 and 16 July 2001 yields 52 matches. On 9 July 2001, the thirty chart of top ten ports showed that the number of reported Port 27374 scans was 4048423.

3-2-7-3 Defensive Recommendations:

An unsolicited transmission from source port 27374 can be an indication of a possible SubSeven Trojan. SubSeven infects Windows based hosts only at the present. Until a better detection rule or method for the SubSeven Trojan is found, I recommend that every incidence of an unsolicited transmission from source port 27374 from a Windows operating system be investigated and all other operating systems be closely monitored if not investigated. For those instances where one or two packets are sent and no reply is received I remember a line from the movie “Hunt for Red October\(^62\)” where Commander Marko Ramius is answering a question from the Commander of the USS Dallas and gives the following instruction to his Sonar Operator (I apologize if I spelled the name wrong). “Give me a single ping Vassili. One ping only.” In this case he was signaling that he was willing to do what the Commander of the USS Dallas was asking him to do.

Scripts to scan for and clean some of the Trojans mentioned above and for detecting the SubSeven Trojan are available on the SANS web site and from Antivirus vendors such as Symantec\(^63\) and Network Associates\(^64\). Please check one or all of these web sites for the latest

---

\(^59\) Computer Emergency Response Team, http://www.cert.org
\(^60\) Neohapsis Archives, http://archives.neohapsis.com
\(^61\) Consensus Intrusion Database (CID) Search page at http://www.incidents.org/cid/search.php

Page 80 of 181
information, tools, and instructions on how to detect and clean these Trojans if found on your system.

Invest in a good Trojan scanner to be used to investigate for possible Trojan infections. Encourage your users to avoid downloading and running executables from sites or persons they are not familiar with. Require the use of a good Antivirus package on all your organizations systems and make sure they keep the signatures updated.

You should update your Snort rule set to the newer versions of the Trojan rules. This may reduce the number of false positives.

---

64 Network Associates Inc, McAfee vShield Antivirus Software, [http://vil.nai.com](http://vil.nai.com)
3-2-8 Queso Fingerprint

During the period 04/10/2001 to 04/16/2001, there were 142 recorded alerts for **QUESO Fingerprint activity** in the IDS Logs.

![Queso Fingerprint Activity Graph]

3-2-8-1 Description/Discussion:

Queso\(^{65}\) is a Fingerprinting program similar to NMAP, used for reconnaissance and not for attacks. Julie LeFebvre\(^{66}\) states in her practical that “Queso correctly determines the operating system to be Linux or Windows”. Information concerning the program and its capabilities is available at Matarese.com\(^{67}\). From the Matarese web site, “QueSO means cheese in spanish, but does also mean que-SO or what-OS”.

While searching for a description of Queso, I ran across a whitepaper by Toby Miller\(^{68}\) on the SANS website that raises the question of whether these Queso fingerprint packets are in fact Queso fingerprinting or are they ECN packets. An incoming ECN packet will have the two reserved flags plus the SYN Flag set. A reply to this should have the reserved flag and the SYN-ACK flag combination. In our logs we only see one incoming packet and no reply which would rule out ECN. The following is an extract from RFC 2884\(^{69}\), ECN and IP Networks:

> In the connection setup phase, the source and destination TCPs have to exchange information about their desire and/or capability to use ECN.

---


\(^{66}\) LeFebvre, Julie, GCIA, SANS Practical. [http://www.sans.org/y2k/practical/Julie_lefebvre.doc](http://www.sans.org/y2k/practical/Julie_lefebvre.doc)

\(^{67}\) Matarese.com, Queso Analysis of Queso Performance. [http://www.matarese.com/queso.html](http://www.matarese.com/queso.html)

\(^{68}\) Miller, Toby, ECN and It’s Impact on Intrusion Detection, SANS, 1999. [http://www.sans.org/y2k/ecn.htm](http://www.sans.org/y2k/ecn.htm)

This is done by setting both the ECN-Echo flag and the CWR flag in the SYN packet of the initial connection phase by the sender; on receipt of this SYN packet, the receiver will set the ECN-Echo flag in the SYN-ACK response. Once this agreement has been reached, the sender will thereon set the ECT bit in the IP header of data packets for that flow, to indicate to the network that it is capable and willing to participate in ECN. The ECT bit is set on all packets other than pure ACK's.

A search of the current Snort Rule sets from Whitehats.com and Snort.org revealed the following rules:

- **Whitehats** - alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS29/scan_probe-Queso Fingerprint attempt"; ttl: >225; flags: S12;)
- **SNORT** - alert tcp any any -> $HOME_NET any (msg:"Possible Queso Fingerprint attempt"; flags: S12;)
- **SNORT** - alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS029 - SCAN-Possible Queso Fingerprint attempt";flags:S12;)

Of the 142 recorded alerts:

Eighty-Seven were to port 6346 on various hosts in the MY.NET Network from multiple hosts outside the MY.NET network - This port along with port 6347 are registered to the Gnutella service. This is a file sharing utility that was originally intended to replace Napster. Unlike Napster, this service is capable of sharing more than just MP3 audio files.

Ten were to port 6347 on various hosts in the MY.NET Network - This port along with port 6346 are registered to the Gnutella service. This is a file sharing utility that was originally intended to replace Napster. Unlike Napster, this service is capable of sharing more than just MP3 audio files.

Seven were to port 110 to MY.NET.6.39 & MY.NET.6.44 from 209.150.104.78 – This port is registered as the POP3 port.

Four were to port 113 to MY.NET.202.106, MY.NET.219.42 and MY.NET.219.194 from 209.85.37.71 – According to RFC 1413, Port 113 is used by the Identification Protocol:

    The Identification Protocol (a.k.a., "ident", a.k.a., "the Ident Protocol") provides a means to determine the identity of a user of a particular TCP connection. Given a TCP port number pair, it returns a character string which identifies the owner of that connection on the server's system.

209.85.37.71 is definitely sending crafted packets. Here is an extract from the Out-Of-Spec and Portscan Log files:

---


Page 83 of 181
Checking Alert Log for [209.85.37.71]'s data!
04/13-22:47:36.613555 [**] spp_portscan: portscan status from 209.85.37.71: 1 connections
across 1 hosts: TCP(1), UDP(0) STEALTH [**]
04/14-07:27:24.053471 [**] Queso fingerprint [**] 209.85.37.71:42952 -> MY.NET.202.106:113
04/14-07:39:49.642207 [**] spp_portscan: portscan status from 209.85.37.71: 1 connections
across 1 hosts: TCP(1), UDP(0) STEALTH [**]
04/15-00:55:58.444724 [**] Queso fingerprint [**] 209.85.37.71:52251 -> MY.NET.219.42:113
04/15-01:09:54.419888 [**] spp_portscan: portscan status from 209.85.37.71: 1 connections
across 1 hosts: TCP(1), UDP(0) STEALTH [**]
04/15-01:17:25.969483 [**] spp_portscan: portscan status from 209.85.37.71: 1 connections
across 1 hosts: TCP(1), UDP(0) STEALTH [**]

################################
Checking OOS Log for [209.85.37.71]'s data!

################################
Checking Portscan Log for [209.85.37.71]'s data!

** Trying 209.85.37.37 at ARIN **
SoftAware, Inc. (NETBLK-SOFTASURE-BLK3) SOFTASURE-BLK3
209.85.0.0 - 209.85.255.255
A&S Capital Group, Inc. (NETBLK-ASCAPIT-209-85-37) ASCAPIT-209-85-37
209.85.37.0 - 209.85.37.255

MY.NET.219.194 appears to be the victim of a legitimate Queso Fingerprint scan. As you can see, both reserved flags and the SYN flag are set in the packet sent to MY.NET.219.194. This is not normal, but a further check of the log files indicates that there was no record of a reply.

Unfortunately, it looks like MY.NET.219.42 and MY.NET.202.106 are being used for gaming. A search of the Portscan and OOS logs show that both 219.42 and 219.194 are broadcasting UDP Traffic on several well known\(^7\) game ports.

### Table 9 - Game Ports

#### Popular Game Ports

<table>
<thead>
<tr>
<th>Game</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quake 1/QW</td>
<td>27500 (27500→27600)</td>
</tr>
<tr>
<td>Quake 2</td>
<td>27910 (27900→27930)</td>
</tr>
</tbody>
</table>

A check of the log files shows that MY.NET.219.42 and MY.NET.202.106 did not reply to this fingerprint scan so we don’t have to worry about them being fingerprinted. They have probably already given most of their host information away on the game networks anyway.

27 were to various ports between 1798 and 3386 on MY.NET.225.134 from port 706 on 66.31.48.7 – Except for the changing times and destination ports, they all looked like this:

Apr 16 05:15:40 66.31.48.7:706 -> MY.NET.225.134:1798 SYN 21S***** RESERVEDBITS

There were no replies to any of the packets sent during this scan. Every one of these twenty-seven alerts also showed up in the Out-Of-Spec logs. Remember this host IP and source port, you will see it used again and again.

Two were to ports 2953, 2957, & 2965 on MY.NET.225.134 from 194.182.79.67 ports 1710 – 1712 – As before, except for the changing times and destination ports, they all looked like this:


Notice that this scan took place on 4/11/2001 and the previous scan took place on 4/16/2001. The source port of this scanning host has remained the same. There were no replies to any of these incoming packets logged.

One was to port 2504 on MY.NET.219.134 from port 706 on 66.31.48.7 – Same packet signature as before:

Apr 10 09:15:06 66.31.48.7:706 -> MY.NET.219.134:2504 SYN 21S***** RESERVEDBITS

Another different date but the same source port and IP address. Since the source port is not changing, this adds further weight to the fact that this is some type of fingerprint scan. There were no replies to any of these incoming packets logged.

The remaining three are in the table below:

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Source IP</th>
<th>SRC Port</th>
<th>Destination IP</th>
<th>DST Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/16</td>
<td>09:44:41.556110</td>
<td>158.75.57.4</td>
<td>52947</td>
<td>MY.NET.206.250</td>
<td>6355</td>
</tr>
<tr>
<td>04/11</td>
<td>12:16:15.861042</td>
<td>63.224.52.208</td>
<td>61942</td>
<td>MY.NET.208.54</td>
<td>6700</td>
</tr>
</tbody>
</table>

Table 10 - QUESO Fingerprint Scan Entries
Notice that the flags never change. The two reserved flags are set as well as the SYN Flag.

Notice that the flags never change. The two reserved flags are set as well as the SYN Flag.

Apr 16 09:44:41 158.75.57.4:52947 -> MY.NET.206.250:6355 SYN 21S***** RESERVEDBITS
Apr 11 12:16:15 63.224.52.208:61942 -> MY.NET.208.54:6700 SYN 21S***** RESERVEDBITS
Apr 12 00:18:28 216.5.180.10:1006 -> MY.NET.60.11:22 SYN 21S***** RESERVEDBITS

Of the three, I would worry about the last connection. There is only the one connection from this host and this port, but this is to port 22 (Registered to Secure Shell according to the IANA Port Number list\(^\text{72}\)) and it originates from port 1006 instead of the usual port 1023 (the default source port for most SSH Clients). There is no record of a reply recorded, but a compromise of SSH would be disastrous.

**Trying 216.5.180 at ARIN**

Business Internet, Inc. (NET-ICIX-MD-BLK17)  
3625 Queen Palm Drive  
Tampa, FL 33619  
US

Netname: ICIX-MD-BLK17
Netblock: 216.0.0.0 - 216.5.255.255
Maintainer: IMBI

Coordinator:  
Business Internet, Inc. (ZI44-ARIN) ipreq@icix.net  
240-616-2000

Domain System inverse mapping provided by:  
NS.DIGEX.NET  164.109.1.3  
NS2.DIGEX.NET  64.245.43.14

Record last updated on 02-Jan-2001.  
Database last updated on 14-Jul-2001 23:02:13 EDT.

**3-2-8-2 Correlation(s):**

Mark Scott\(^\text{73}\) compares NMAP and Queso alerts in his practical, but he does not mention the ECN question.

A rather large snort log from a company that was shutdown because of reported queso fingerprinting can be found at Neohapsis – [http://archives.neohapsis.com/archives/postfix/2001-](http://archives.neohapsis.com/archives/postfix/2001-)

---

\(^{72}\) IANA Port List, [http://www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers)

\(^{73}\) Scott, Mark GCIA 253, SANS Practical. [http://www.sans.org/y2k/practical/Mark_Scott.doc](http://www.sans.org/y2k/practical/Mark_Scott.doc)
I found this on Security Focus -
http://www.securityfocus.com/frames/?content=/templates/archive.pike%3Flist%3D75%26mid%3D178231

I found a SnortSnarf\textsuperscript{74} sample from the University of Heidelberg while performing a search with WebFerret\textsuperscript{75} - http://www.gs.uni-heidelberg.de/~malsburg/files/snfout.snort.alert/sig/sig13.html

3-2-8-3 Defensive Recommendations:

How do you defend against a reconnaissance probe? Move all critical servers behind a firewall and use proxy servers whenever possible. You should apply Security patches as operations permit and after careful testing. Install and use Intrusion Detection Systems both network and host based IDS systems are recommended. Monitor your Firewall and IDS logs daily. Check your Syslogs daily. You will never be able to prevent all reconnaissance from being successful, but you can take steps to reduce the amount and type of data obtained from reconnaissance.

You should consider adding the following rule to your rules list as well (Found on the Snort.org\textsuperscript{76} web page, latest news page one):

**New SSH rule from Chris Kuethe, and a new paper on Snort and Win2k** - by Jim Forster @ 14:05:06

This rule will detect SSH traffic on ports other than the standard, port 22.

\begin{verbatim}
alert tcp $EXTERNAL_NET !22 -> $HOME_NET !22 (flags:AP+; msg:"SSH not on port 22"; content:"SSH-"; offset:0; depth:8;)
\end{verbatim}

Thanks Chris

\textsuperscript{74}SnortSnarf, A Snort Log Analyzer from Silicon Defense. http://www.silcondefense.com
\textsuperscript{75}WebFerret, A web search utility from FerreteSoft, a subsidiary of ZD Net. 
http://www.zdnet.com/ferret/download.htm
3-2-9 Russia Dynamo

During the period 04/10/2001 to 04/16/2001, there were 1735 recorded alerts for Russia Dynamo activity in the IDS Logs.

![Russia Dynamo Chart]

3-2-9-1 Description/Discussion:

This appears to be a rule watching all traffic from 194.87.6.xx. The original flash from sans on 7/28/2000\(^{77}\) recommended that if you see a machine transmitting data from or to 194.87.6.X you should pull it from the network immediately. It also recommended that you block traffic to and from 194.87.6.X. The cause for the traffic was given as most likely a Trojan. It was changed shortly after to just watch traffic on ports 80, 8080, and 3128. A reply from the Russian ISP was printed by SANS in the 7/31/00\(^{78}\) Issue of Detects Analyzed stating that the ISP had caught and shutdown the person responsible for the attacks. The Trojan was described by Dan Wangler\(^{79}\) as looking like RingZero.

**Trying 194.87.6 at RIPE**

```
Trying 194.87 at RIPE
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
```


\(^{78}\) SANS GIAC Detects Analyzed - 7/31/00. [http://www.sans.org/y2k/073100-1030.htm](http://www.sans.org/y2k/073100-1030.htm)

\(^{79}\) Wangler, Dan GCIA 0328 SANS Practical. [http://www.sans.org/y2k/practical/Dan_Wangler_GCIA.doc](http://www.sans.org/y2k/practical/Dan_Wangler_GCIA.doc)
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 194.87.0.0 - 194.87.3.255
netname: DEMOS-CORP
descr: DEMOS Corporate Network
descr: Demos Plus Co. Ltd.
descr: Moscow, Russia
country: RU
admin-c: DNOC-ORG
tech-c: DNOC-ORG
status: ASSIGNED PA
mnt-by: AS2578-MNT
changed: eugen@demos.net 19970313
changed: galka@demos.net 19990804
changed: galka@demos.net 20000927
source: RIPE

route: 194.87.0.0/19
descr: DEMOS
origin: AS2578
notify: noc@demos.net
mnt-by: AS2578-MNT
changed: noc@demos.net 20000927
source: RIPE

role: Demos Internet NOC
address: Demos Company Ltd.
address: 6-1 Ovchinnikovskaya nab.
address: Moscow 113035
address: Russia
phone: +7 095 737 0436
phone: +7 095 737 0400
fax-no: +7 095 956 5042
e-mail: ncc@demos.net
admin-c: KEV6-RIPE
admin-c: RVP18-RIPE
admin-c: GK41-RIPE
tech-c: KEV6-RIPE
tech-c: RVP18-RIPE
tech-c: GK41-RIPE
nic-hdl: DNOC-ORG
notify: hm-dbm-msgs@ripe.net
notify: ncc@demos.net
notify: ip-reg@ripn.net
mnt-by: AS2578-MNT
This rule appears to be left over from that incident, since it only identifies traffic sent to and from 194.87.6.X. This rule generated 1725 alerts on 4/10/2001. There were nine more generated on 4/11/2001 and one final alert on 4/16/2001. During these three time periods there were two hosts outside the MY.NET network talking to MY.NET.178.42:

### Table 11 - Russia Dynamo Connections

<table>
<thead>
<tr>
<th>Count</th>
<th>Source IP</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>308</td>
<td>194.87.6.106 MY.NET.178.42</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>194.87.6.201 MY.NET.178.42</td>
<td></td>
</tr>
</tbody>
</table>

Ports used by the MY.NET host include 316, 317, 3146 (x1), 3251 (x2), and 3252 (x2). The source ports on the two destination hosts (94.87.67.201 and 194.87.6.106) started out at 1804 for most of the day on 4/10/2001 and changed to 1802, back to 1030 and then increased somewhat sequentially from then on (1030, 1031 1054, 1057, 1063, 1064, 1065, 1066, 1069, etc). None of the recommended ports to watch were used in this connection.

### 3-2-9-2 Correlation(s):

I have found very little information on this alert. Besides the sources quoted in the description, I found one article in the Neohapsis Archives. A search of the GCIA Practicals greater than 209 provided me with a list of twenty practicals (not counting HTML formatted or zipped archives) that contained the words “Russia Dynamo”

<table>
<thead>
<tr>
<th>Alert Description</th>
<th>Number of Alerts</th>
<th>Number of Source Systems</th>
<th>Number of Destination Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russia Dynamo - SANS Flash 28-jul-00</td>
<td>546</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Loras Even, GCIA 325:

<table>
<thead>
<tr>
<th>Source</th>
<th># Alerts (sig)</th>
<th>Destinations</th>
<th># Alerts (sig)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MY.NET.205.138</td>
<td>442</td>
<td>194.87.6.38</td>
<td>442</td>
</tr>
</tbody>
</table>

---

80 I downloaded them to my hard drive and used the Windows Find tool to search all documents in the folder.
82 Evans, Mark GCIA 350, SANS. http://www.sans.org/y2k/practical/Mark_Evans_GCIA.doc
83 Even, Loras, GCIA 325, SANS. http://www.sans.org/y2k/practical/Loras_Even_GCIA.doc
A search of the SANS Detects Analyzed web page provided the one article with the reply from the Russian ISP which was quoted above.


### 3-2-9-3 Defensive Recommendations:

CHECK THIS MACHINE OUT! Go over it with a magnifying glass. I believe that something automated was in control during these time periods. A large number of alerts were generated by MY.NET.178.42 on the MY.NET network and the source ports were limited to 316 and 317 for the majority of the connect time. The connection on 4/10/2001 lasted for one hour early in the morning (00:07 to 01:01) and again for 2.5 hours (1800 to 2035) that evening. While the evening connection is not exactly, the early morning one is. Either the user on this system never sleeps, or something automated (Trojan?) is doing a lot of work.
3-2-10 SMB Name Wildcard

During the period 04/10/2001 to 04/16/2001, there were 138 recorded alerts for *SMB Name Wildcard* in the IDS Logs. A top ten talkers list is provided, but only to show that a more serious problem is ‘In the weeds’ and did not show up on the top talkers list.

![SMB Name Wildcard Chart]

### Table 12 - SMB Name Wildcard Top Ten Talkers

<table>
<thead>
<tr>
<th>Count</th>
<th>Source IP</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>66.74.68.29</td>
<td>MY.NET.134.57</td>
</tr>
<tr>
<td>4</td>
<td>24.93.44.178</td>
<td>MY.NET.134.144</td>
</tr>
<tr>
<td>4</td>
<td>MY.NET.111.156</td>
<td>MY.NET.125.41</td>
</tr>
<tr>
<td>3</td>
<td>130.13.103.236</td>
<td>MY.NET.135.60</td>
</tr>
<tr>
<td>3</td>
<td>130.157.148.70</td>
<td>MY.NET.134.76</td>
</tr>
<tr>
<td>3</td>
<td>130.67.82.251</td>
<td>MY.NET.132.36</td>
</tr>
<tr>
<td>3</td>
<td>169.254.106.79</td>
<td>MY.NET.135.67</td>
</tr>
<tr>
<td>3</td>
<td>208.190.216.9</td>
<td>MY.NET.135.209</td>
</tr>
<tr>
<td>3</td>
<td>211.106.45.141</td>
<td>MY.NET.134.227</td>
</tr>
<tr>
<td>2</td>
<td>130.13.120.130</td>
<td>MY.NET.133.103</td>
</tr>
</tbody>
</table>

### 3-2-10-1 Description/Discussion:

Here is a short description from Robert Graham’s web site[^84]

> NetBIOS requests to UDP port 137 are the most common item you will see in your

firewall reject logs. This comes about from a feature in Microsoft's Windows: when a program resolves an IP address into a name, it may send a NetBIOS query to IP address. This is part of the background radiation of the Internet, and is nothing to be concerned about.

The third paragraph of this section should also be quoted here as well.

Note that you will see NetBIOS scans, such as from hackers running the Legion NetBIOS scanner or other scanners. In this case, you'll likely see a scan of your entire address range. The important thing to remember is that few NetBIOS packets are from hostile intent.

Mr. Graham even tells us what normal traffic should look like, so I will include that section of his paper as well.

Windows machines use both a source port of 137 as well as a destination port of 137. In contrast, if UNIX machines attempt to resolve NetBIOS names (via SAMBA), they will use dynamic ports above 1024.

If the Windows box is trying to find the name for the IP address 192.0.2.21, it will do the following steps:

- Lookup the DNS "PTR" record for 21.2.0.192.in-addr.arpa; this request is sent to the local DNS server, which recursively forwards the query to the appropriate DNS server as required.

- If the DNS answer comes back, it won't query NetBIOS. If a negative response comes back, it will immediately query NetBIOS. If the DNS server times-out, it will wait 14-seconds, then query NetBIOS.

- When resolving with NetBIOS, it will send out a "NodeStatus" query that is sent to the 192.0.2.12:137 from x.x.x.x:137. (I.e. the query is sent to the IP address being resolved to its port 137, and is sent from the Windows machine port 137).

- The NetBIOS request is a "NodeStatus" query that looks up the name "*". It is 50 bytes worth of data (58 including the UDP header, 78 including the IP header, 92 including an Ethernet header).

- Three NetBIOS queries are sent with a 1.5 second timeout.

There is no evidence of any major subnet scanning from hosts outside the MY.NET network. In those cases where you only see one or two incoming packets, you could be seeing a very slow scan. There are two instances of Private Network Addresses showing up in the scans and one of them indicates the presence of the network.vbs worm. Information on this can be found in the SANS Intrusion Detection FAQ on Port 137 Scans. I have extracted the appropriate paragraph

---

85 Intrusion Detection FAQ, Port 137 Scans, Bryce Alexander, May 2000, SANS.  
http://www.sans.org/newlook/resources/IDFAQ/port_137.htm
An interesting side effect of this worm has been a rather strange pattern that periodically shows up in the scans for port 137. This pattern shows simultaneous scanning from two addresses, one a legitimate address and one a private (RFC1918\textsuperscript{86}) address. It is my speculation that this is caused by systems that are providing proxy services on cable modems in order to share a single IP address on a cable modem. The internal (private) address is leaking out onto the network, most likely due to sharing a single ethernet hub for both internal and external interfaces.

Here is the indicator (notice that none of these are in the Top Ten Talkers list):

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Port</th>
<th>Destination IP Address</th>
<th>Destination Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/16</td>
<td>04:11:41</td>
<td>192.168.1.1</td>
<td>137</td>
<td>MY.NET.134.155</td>
<td>137</td>
</tr>
<tr>
<td>04/16</td>
<td>04:11:44</td>
<td>61.119.188.138</td>
<td>137</td>
<td>MY.NET.134.155</td>
<td>137</td>
</tr>
</tbody>
</table>

MY.NET.134.155 is probably infected with the Network.VBS Worm. Disconnect this machine from the network and clean it immediately.

The second instance of a Private Network address showing up does not show the network.vbs signature and can probably be dropped for now. Keep an eye out for 10.0.0.1, and you may want to add the ‘-e’ switch to your snort command line if it looks like it is becoming a problem.

3-2-10-2 Correlation(s):

A short search of the Neohapsis archive reveals these links, there are more.


CERT has an incident note on the subject http://www.cert.org/incident_notes/IN-2000-02.html

Secure Point has a couple of articles of interest.

http://msgs.securepoint.com/cgi-bin/get/ids-0003/35/1.html
http://msgs.securepoint.com/cgi-bin/get/ids-0003/35.html

3-2-10-3 Defensive Recommendations:

The best defense is to block ports 135 through 139 at the perimeter routers. You may be able to get by with just blocking 135-138.

You should install anti-virus software on all hosts and keep the virus signatures current. Perform

\textsuperscript{86} RFC 1918, Address Allocation for Private Internets. http://www.rfc-editor.org/rfc/rfc1918.txt
a full scan of the servers Bi-weekly. Experience has taught me that having anti-virus software and keeping it current is not enough anymore. I have experienced first-hand that if I update my anti-virus signatures weekly you can still have viruses on the system, they were put on your system before a virus signature was found to detect them. You will not detect these viruses unless you periodically perform a full scan of all files on your systems. You will not detect these older viruses until you access the infected files, which may be months after you acquired them. This is especially true for Macro viruses. I would recommend twice weekly anti-virus signature updates for servers and weekly anti-virus signature updates for desktops if they are available. Apply all recommended security patches to the Microsoft Windows© Operating System platforms. Purchase a Trojan scanner that you can use to scan your hosts periodically and keep it updated as well.

If you can scan incoming electronic mail for attachments, then I would highly recommend that you block all .VBS (VB Script) and .WSH (Windows Script Host) files as well. This has nothing with Port 137 scans, but if you block port 137 at your border routers, then someone will more than likely receive an e-mail message with the network.vbs worm attached to it and then spread it internally.
3-2-11 STATDX UDP Attack

During the period 04/10/2001 to 04/16/2001, there was 1 incidence of a *STATDX UDP Attack* recorded in the IDS Logs.

![STATDX UDP Attack Graph]

3-2-11-1 Description/Discussion:

The description of this exploit comes from George Bakos, GCIA 228 practical:

The rpc.statd is the NFS file lock status reporter. Its function is to track NFS connections with requests to the rpc.lockd. In the event of a server going down, the rpc.statd will attempt to reestablish those locks by communicating the server's status to the NFS client's lock manager.

There is a process of the rpc.statd which passes logging information using the syslog() function. The format string passed is user supplied data, with a UID:GID of 0:0, without any proper bounds checking. It is possible, and proven, that this buffer could be overflowed, placing executable code into the process address space and overwriting the process return address, forcing the execution of that code. This is commonly known as "smashing the stack".

The Alert and Portscan logs for this host are here:

```
04/10-02:44:07.761846  **] STATDX UDP attack [**] 24.43.176.96:2099 -> MY.NET.6.15:32776
#########################################################################
Checking Portscan Log for [MY.NET.6.15]'s data!
```

---

87 SANS, George Bakos GCIA 228 Practical. http://www.sans.org/y2k/practical/George_Bakos.html
The alert show us that something was tried, but we see no replies or acknowledgements. We cannot tell from this one packet that anything really happened. We can assume, based on the two ignored SYN packets sent to this host a little over two hours later that nothing appears to have happened.

**3-2-11-2 Correlation(s):**

Here is a note from the author of the shellcode exploit that was sent to Bugtraq explaining his reasoning behind releasing the exploit - [http://msgs.securepoint.com/cgi-bin/get/bugtraq0008/75.html](http://msgs.securepoint.com/cgi-bin/get/bugtraq0008/75.html).

I have only included two or three links from three search engines, but you can find more.

Bugtraq search for ‘rpc.statd’ on Secure Point:

[http://msgs.securepoint.com/cgi-bin/get/bugtraq0007/209.html](http://msgs.securepoint.com/cgi-bin/get/bugtraq0007/209.html)
[http://msgs.securepoint.com/cgi-bin/get/bugtraq0007/158.html](http://msgs.securepoint.com/cgi-bin/get/bugtraq0007/158.html)

CERT Advisories, Bulletins and Incident Notes:

Widespread Compromises via “ramen” Toolkit –

Problem in rpc.statd –

Widespread Exploitation of rpc.statd and wu-ftpd Vulnerabilities –

Neohapsis Archives search for ‘rpc.statd’ results:


**3-2-11-3 Defensive Recommendations:**

As a minimum you should apply all rpc patches. Read the Cert Advisories and Incident notes for advice and information on securing those systems that cannot be patched.
3-2-12 SYN-FIN Scan

During the period 04/10/2001 to 04/16/2001, there were 4 SYN-FIN Scans recorded in the IDS Logs. A discussion of all four hosts involved is included.

3-2-12-1 Description/Discussion:

From Page 345, Intrusion Signatures and Analysis:

The purpose of the SYN-FIN seems to be twofold, or at least that was the case in 1997. First, because some systems allow FINs to pass through, the attacker uses this technique for network mapping. Second, because FINs tear down connections, some systems do not log these types of packets. Today, every analyst knows to look for SYN-FIN; so why do we still see these packets? Part of the reason is OS fingerprinting. Other forms of Out-Of-Spec are not so obvious.

There are four alerts recorded, each one is shown and examined below.

MY.NET.222.134

We see a lot of packets destined for port 6699 (NAPSTER). We don’t see the replies. OS Fingerprinting maybe, corrupted packets probably. Because of the timing, I would say corrupted packets, it appears that a large file transfer may be going on here which is consistent with the use of Napster.

88 Cooper, Fearnow, Frederick and Northcutt “Intrusion Signatures and Analysis”. Reading: New Riders Publishing 2001
04/10-14:01:07.799338  [**] SYN-FIN scan! [**] 141.30.222.116:1280 -> MY.NET.212.134:6699

Checking Portscan Log for [MY.NET.212.134]’s data!
Apr 10 02:49:19 216.40.195.72:3952 -> MY.NET.212.134:53 SYN **S*****
Apr 10 14:01:07 141.30.222.116:1280 -> MY.NET.212.134:6699 SYNFIN **SF****
Apr 10 14:21:36 141.30.222.116:44 -> MY.NET.212.134:1307 UNKNOWN *1***PAU RESERVEDBITS

Checking Out-Of-Spec Logs for [MY.NET.212.134]’s data!
04/10-14:01:00.308471 141.30.222.116:1280 -> MY.NET.212.134:6699
04/10-14:07:10.185490 141.30.222.116:1295 -> MY.NET.212.134:6699
04/10-14:19:00.125190 141.30.222.116:1307 -> MY.NET.212.134:6699

Gnutella traffic maybe with some corrupted packets. If we look at the Portscan logs however we see that the sending host has also sent a NULL Packet approximately thirty minutes before sending the SYN-FIN packet, and an Out-Of-Spec packet with the SYN-FIN flags that was not logged as a SYN-FIN Scan is present as well. Further analysis of Gnutella traffic indicates that there are a large number of Portscan alerts are generated. Because of the two Out-Of-Spec Packets and the NULL packet occurring in a single thirty time period from port 6346 on one host, I would say this is OS Fingerprinting.
04/11-14:19:29.889612  [**] SYN-FIN scan! [**] 24.64.111.247:6346 -> MY.NET.222.170:2813

Checking Portscan Log for [MY.NET.222.170]'s data!
Apr 10 10:38:43 211.21.104.118:1997 -> MY.NET.222.170:53 SYN **S*****
Apr 10 11:59:26 63.163.94.13:2177 -> MY.NET.222.170:53 SYN **S*****
Apr 11 13:49:19 24.64.111.247:6346 -> MY.NET.222.170:2813 NULL ********
Apr 11 14:19:29 24.64.111.247:6346 -> MY.NET.222.170:2813 SYNFIN **SF****
Apr 14 07:41:15 209.178.22.233:1576 -> MY.NET.222.170:53 SYN **S*****
Apr 14 07:41:20 209.178.22.233:1576 -> MY.NET.222.170:53 SYN **S*****

Checking Out-Of-Spec Logs for [MY.NET.222.170]'s data!
04/11-13:52:15.628288 24.64.111.247:6346 -> MY.NET.222.170:2813
04/11-14:19:22.504419 24.64.111.247:6346 -> MY.NET.222.170:2813

TCP TTL:110 TOS:0x0 ID:54153  DF
**SF**** Seq: 0x69BE9CF   Ack: 0x3F4   Win: 0x5018
TCP Options => EOL EOL

MY.NET.70.27

Looks like some more Gnutella. This SYN-FIN packet is probably OS Fingerprinting. NOT, take a look at the Out-Of-Spec logs at 12:58 that same day. We see the 63.196.167.131 host is sending what appear to be crafted packets with strange flag settings (21**R*** & 21*FRPA*) to MY.NET.70.27. Looks like he found something at 11:23 and came back for more at 12:58.

04/12-11:23:36.405020  [**] SYN-FIN scan! [**] 63.196.167.131:4168 -> MY.NET.70.27:6346

Checking Portscann Log for [MY.NET.70.27]'s data!
Apr 11 15:37:37 MY.NET.70.27:4617 -> 132.248.188.137:6346 SYN **S*****
Apr 11 15:37:37 MY.NET.70.27:4618 -> 193.158.170.57:6346 SYN **S*****
Apr 11 15:37:37 MY.NET.70.27:4619 -> 156.17.213.8:6346 SYN **S*****
Apr 11 15:37:37 MY.NET.70.27:4620 -> 146.201.32.254:6346 SYN **S*****
Apr 11 15:37:37 MY.NET.70.27:4621 -> 61.9.169.135:6346 SYN **S*****
Apr 11 15:37:38 MY.NET.70.27:4613 -> 64.61.25.139:6346 SYN **S*****
Apr 11 15:37:39 MY.NET.70.27:4637 -> 130.231.6.106:6346 SYN **S*****
Apr 11 15:37:39 MY.NET.70.27:4638 -> 65.26.218.244:6346 SYN **S*****
Apr 11 15:37:39 MY.NET.70.27:137 -> 63.148.194.216:137 UDP
Apr 11 15:37:42 MY.NET.70.27:4643 -> 213.65.167.241:17711 SYN **S*****
Apr 11 15:37:44 MY.NET.70.27:4657 -> 149.159.23.34:6346 SYN **S*****
Apr 11 22:05:10 217.136.56.13:3161 -> MY.NET.70.27:21 SYN **S*****
Apr 12 11:23:36 63.196.167.131:4168 -> MY.NET.70.27:6346 SYNFIN **SF****
Apr 14 07:38:38 209.178.22.233:23212 -> MY.NET.70.27:53 SYN **S*****
Apr 15 06:40:16 62.59.129.0:1685 -> MY.NET.70.27:6347 SYN **S*****
Apr 15 06:40:16 62.59.129.0:18254 -> MY.NET.70.27:21844 NOACK 2**FR***
RESERVEDBITS

#########################################################################
# Checking Out-Of-Spec Logs for [MY.NET.70.27]'s data!
04/12-11:19:38.362549 213.97.79.237:107 -> MY.NET.70.27:39801
04/12-11:23:27.916784 63.196.167.131:4168 -> MY.NET.70.27:6346
04/12-12:58:26.985188 63.196.167.131:4168 -> MY.NET.70.27:6346
04/12-12:58:35.516313 63.196.167.131:4168 -> MY.NET.70.27:6346
04/13-17:45:58.681625 213.64.56.9:6346 -> MY.NET.70.27:4515
04/14-03:50:25.717627 194.236.50.60:11743 -> MY.NET.70.27:6347
04/16-00:51:59.922740 61.200.27.90:43962 -> MY.NET.70.27:6347
04/16-05:37:16.637323 192.117.120.140:62584 -> MY.NET.70.27:6347

=========+============================================+
04/12-11:23:27.916784 63.196.167.131:4168 -> MY.NET.70.27:6346
TCP TTL:49 TOs:0x0 ID:49236  DF **SF**** Seq: 0x8A4ED8D  Ack: 0x1F5E2F  Win: 0x3DA0
34 03 3D A0 4E B5 63 B1 23 20 EE DE EA 28 4.=.N.c. ...(=

04/12-12:58:26.985188 63.196.167.131:4168 -> MY.NET.70.27:6346
TCP TTL:49 TOs:0x0 ID:6105  DF 21**R*** Seq: 0x8C85BAA  Ack: 0x45459A  Win: 0xD102
10 48 18 CA D8 C8 5B AA 00 45 45 9A 04 C4 D1 02 .H...[..EE......
82 00 00 48 5D C9 02 00 50 B2 ...H]...P.

04/12-12:58:35.516313 63.196.167.131:4168 -> MY.NET.70.27:6346
TCP TTL:49 TOs:0x0 ID:6369  DF 21*FRPA* Seq: 0x8C85BAA  Ack: 0x45459A  Win: 0xD102
10 48 18 CA D8 C8 62 7A 00 45 4D D7 04 DD D1 02 .H...bz.EM.....
82 00 00 48 5D C9 02 00 A1 D3 ...H].....

MY.NET.97.227
SSL Port to port 1258. One SYN-FIN packet. No other anomalies seen. This alert can be filed.


#########################################################################
Checking Portscan Log for [MY.NET.97.227]'s data!

#########################################################################
Checking Out-Of-Spec Logs for [MY.NET.97.227]'s data!

+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
TCP TTL:113 TOS:0x0 ID:7919  DF
**SF**** Seq: 0x3E1C83  Ack: 0x7AD4FC  Win: 0x7B7C
34 03 7B 7C 9E 41 D1 5B 74 AB F9 12 76 34  4.[].[t...v4
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

3-2-12-3 Correlation(s):

107 GCIA Practicals contain detects of SYN-FIN scans or discuss them. A few of them are listed here:

Terry Bidwell (267), GCIA Practical, Detect #4 –
http://www.sans.org/y2k/practical/Teri_Bidwell_GCIA.doc

Guy Bruneau, (255), GCIA Practical, Detect #1 -
http://www.sans.org/y2k/practical/Guy_Bruneau.doc

David Singer (353), GCIA Practical, Page 32 –
http://www.sans.org/y2k/practical/David_Singer_GCIA.doc

A search of the Neohapsis Archives for “SF Scan” provides the following links:

http://certworks.net/ids/data/snfout.snort_portscan.log/sig/sig59.html
http://komura.net/snort/sig/sig1.html
http://www.ajlc.waterloo.on.ca/snort/sig/sig18.html

3-2-12-3 Defensive Recommendations:

The best defense against a SYN-FIN scan is to apply all patches. Close all non-needed ports and only run services that are absolutely necessary. Keep your drivers current as well.
3-2-13 SRC and DST outside network

NOTE: I have merged the TCP SRC and DST outside network, UDP SRC and DST outside network and ICMP SRC and DST network alerts into one section. There is a separate chart for each alert, but the narrative is combined.
3-2-13-1 Description:

Seeing addresses originating from inside your network with source addresses outside your network is Source Address Spoofing. Page 134 of the book Intrusion Signatures and Analysis[89] contains several uses for spoofed source addresses; an attacker wanting to hide his activities or his identity or he may be conducting a Denial Of Service (DoS) attack. We are dealing with three protocols. The Intrusion Signatures and Analysis book covers this exploit very well. I will

borrow the description of these protocols from the book:

ICMP and UDP are connectionless and stateless protocols. It is often impossible to determine whether a received UDP or ICMP packet has been forged just by looking at the received packet. Pg 137, Intrusion Signatures and Analysis, New Riders Publishing 2001.

Recall that TCP is a connection-oriented protocol that maintains state. If an attacker spoofs the source address in a TCP-SYN packet, how will the attacker be able to respond to the SYN-ACK packet returned? Pg 137, Intrusion Signatures and Analysis, New Riders Publishing 2001.

For more on this subject, you should read Chapter 7 of Intrusion Signatures and Analysis. It goes into much more detail on the subject of source address spoofing.

This is Source Address Spoofing on the MY.NET network. These packets are originating from inside our network with source addresses outside our network. We should not see packets of this nature inbound to our network for obvious reasons (they are not addressed to us). There are several causes for this, some natural and some not-so natural, they include; improperly configured routers, defective/broken routers, or one or more compromised machines on the MY.NET network (Drones/Slaves/Agents).

What services are being attacked? I am not going to list them all, but the biggest were Port 53 (Domain Name Service), 137 (NETBIOS Name Service), and 5190 (AOL). A little over 1400 of the spoofed address packets were UDP packets that had a destination address of 10.10.10.50 and a source address of 192.168.0.53 (or some other private address), and all of them had the same source and destination port of 137. Again, I take an explanation from the book Intrusion Signatures and Analysis:

If a host is using the spoofed IP address, that host silently discards this unexpected ICMP message. If no host is using the spoofed IP address, a router silently discards the ICMP message. Pg 138, Intrusion Signatures and Analysis, New Riders Publishing 2001.

The 169.254.xxx.xxx traffic (623 packets) is best explained by this excerpt from a post found in the Neohapsis archives:

> For last week i sent 4 or 5 complains about UDP scan (138 port). I have
> one answer from iana.org,they wrote: "It is legal traffic and do not
> worry about it and contact to your ISP for more information".It was 2

---

90 Cooper, Fearnow, Frederick and Northcutt “Intrusion Signatures and Analysis”. Reading: New Riders Publishing 2001

Today I sent him a next complain about new scan....

In first: I am the ISP myself ;)

In second: This traffic just has been directed not to one host, in the

log i saw this:

Aug-30-01:37:02 UDP from 169.254.100.72:137 to XXX.XX.XXX.16:137
Aug-30-01:37:06 UDP from 169.254.100.72:137 to XXX.XX.XXX.17:137

169.254.0.0/16 is reserved for auto-configuration of local addresses
in networks where no DHCP server is found[1]. That block is not (or
at least should not) be routed over the internet backbones[2]. Any
traffic from 169.254.0.0/16 is either from your local network, or
forged—and either way, complaining to IANA or ISI is a waste of their
time.

[2] Try a traceroute—you should run into a no-route in a short number
of hops:

% traceroute 169.254.100.72
traceroute to 169.254.100.72 (169.254.100.72), 30 hops max, 40 byte packets
  1 insfw (128.84.44.1) 3 ms 3 ms 3 ms
  2 ccc1-8540-vlt669.cit.cornell.edu (128.253.147.4) 9 ms 14 ms 10 ms
  3 cornellnet4-gig1-0-0.cit.cornell.edu (128.253.222.162) 6 ms !H 5 ms !H 9 ms !H

Maybe someone is using Network Address Translation and it is not configured correctly. An
improperly configured NAT would explain the first 1399 packets we see logged.

This could be a Denial of Service attempt against another network, originating from within your
network. Why are the private IP Addresses showing up in the IDS Logs, are the routers not
properly configured to block outbound private IP Addresses?

3-2-13-2 Correlation(s):

Below are the search results a search on CERT, Neohapsis, and Security Focus Bugtraq.

CERT Advisories/Incident Notes/Bulletins -
http://www.cert.org/incident_notes/IN-99-07.html

Neohapsis Search results -

Security Focus Search results -
http://www.securityfocus.com/frames/?content=/templates/archive.pikel%3Dlist%3D1%26mid%3D
3-2-13-3 Defensive Recommendations:

To prevent your network from being used in this manner you should configure your routers IAW the guidance contained in RFC2267\(^{92}\). Block private IP addresses\(^{93}\) from leaving your network. To prevent your machines from being used for this type of attack you should install Anti-virus software and update the anti-virus signatures often. Perform routine scans of all files for viruses. Get a good Trojan scanner and scan your systems regularly for Trojans. RFC2267 will also help you setup your Router ACL’s to prevent spoofed addresses from entering or leaving your network as well.


3-2-14 Tiny Fragments

During the period 04/10/2001 to 04/16/2001, there were 20 recorded alerts for Tiny Fragments in the IDS Logs.

3-2-14-1 Description/Discussion:

Fragmentation happens when a packet crosses a network that has a Maximum Transmission Unit (MTU) smaller than the size of the packet being transmitted. We have a total of three hosts sending fragmented packets to hosts on the MY.NET network. The following table shows the addresses.

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>202.39.78.124</td>
<td>MY.NET.217.134 **</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MY.NET.228.54 **</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MY.NET.202.86 **</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MY.NET.201.6 **</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MY.NET.211.114 **</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>MY.NET.219.126 **</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MY.NET.203.246 **</td>
<td>2</td>
</tr>
<tr>
<td>63.227.41.165</td>
<td>MY.NET.217.166</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MY.NET.202.106 **</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MY.NET.212.198 **</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>MY.NET.204.90</td>
<td>1</td>
</tr>
</tbody>
</table>
**Table 17 – Popular Game Ports**

<table>
<thead>
<tr>
<th>Game</th>
<th>Port(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quake 1/QW</td>
<td>27500 (27500-&gt;27600)</td>
</tr>
<tr>
<td>Quake 2</td>
<td>27910 (27900-&gt;27930)</td>
</tr>
<tr>
<td>Quake 3</td>
<td>27960 (27960-&gt;27980)</td>
</tr>
<tr>
<td>halfLife</td>
<td>27015 (27010 -&gt; 27050)</td>
</tr>
<tr>
<td>Unreal tournament</td>
<td>7777 (7777-&gt;7797)</td>
</tr>
<tr>
<td>Kingpin</td>
<td>31510 (31500-&gt;31550)</td>
</tr>
<tr>
<td>shogo</td>
<td>27888</td>
</tr>
<tr>
<td>starsiege strike</td>
<td>28000 (28001 &amp; 2 often too)</td>
</tr>
</tbody>
</table>

**Some of destination hosts appear to have GameSpy installed as well. This is an application that probes game servers in order to provide you with the status of available servers for game playing over the internet. It uses UDP Pings on port 13139 to check the status and round trip time to servers.**

A search of the Out-Of-Spec logs shows that the three source hosts have not sent any Out-Of-Spec packets to the MY.NET network. There is no port or protocol information contained in the IDS logs. A check of the Portscan Logs reveals that some of the destination hosts are definitely into gaming. They have used almost every port in the following list of popular network games and their standard ports.

The small number of single fragmented packets is an indication of possible malicious activity. It will difficult to determine that fact with all the game traffic going to and coming from the MY.NET hosts. We don’t have port numbers or data packets in the alert log so the only option is to investigate machine individually for signs of compromise.

### 3-2-14-2 Correlation(s):

Below are links to related IP Fragmentation problems reported by Snort and Firewall-1 users.

**Neohapsis Archives search** –


**Secure Point Archive search** –

http://msgs.securepoint.com/cgi-bin/get/fw1arch97/333.html
http://msgs.securepoint.com/cgi-bin/get/fw1-0005/1037.html

---

94 GameSpy, [http://www.gamespy.com](http://www.gamespy.com)

95 MultiPlayer Total Annihilation behind a firewall, [http://www.estrella.demon.nl/mpfw.htm](http://www.estrella.demon.nl/mpfw.htm)
3-2-14-3 Defensive Recommendations:

Apply all system patches. The use of stateful firewalls will reduce but not completely stop fragmented packets from getting into your network. A stateful IDS will reduce false alarms for fragments and will complement a firewall by catching what is missed by the firewall if the right combination of firewall and IDS are used.
3-2-15 Watchlist 000220

During the period 04/10/2001 to 04/16/2001, there were 7562 recorded alerts for Watchlist 000220 in the IDS Logs.

![Bar chart showing alerts per day for Watchlist 000220.]

3-2-15-1 Description/Discussion:

The complete title of this Alert is “Watchlist 000220 IL-ISDNNET-990517”. It appears to be monitoring the 212.179.0.0 subnet. A quick check of whois shows that most some of this subnet is divided up but the divisions are registered to the same individuals. I have only provided a whois lookup on two of the IP addresses in this list. Please note that the first (212.179.7.2) address lookup contains an entry entitled “Napster Info”; this will play a big part in the analysis.

### Trying 212.179.79.2 at ARIN

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>% This is the RIPE Whois server.</td>
<td>212.179.79.0 - 212.179.79.63</td>
<td>CREOSCITEX</td>
<td>CREOSCITEX-SIFRA</td>
<td>IL</td>
<td>ZV140-RIPE</td>
<td>NP469-RIPE</td>
<td>ASSIGNED PA</td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td>RIPE-NCC-NONE-MNT</td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
</tr>
<tr>
<td>% The objects are in RPSL format.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Please visit <a href="http://www.ripe.net/rpsl">http://www.ripe.net/rpsl</a> for more information.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20001109</td>
<td></td>
</tr>
<tr>
<td>source:</td>
<td>RIPE</td>
<td>inetnum:</td>
<td>212.179.7.0 - 212.179.7.255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>route:</td>
<td>212.179.0.0/17</td>
<td>netname:</td>
<td>FIX-IP-BEZEQINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>descr:</td>
<td>ISDN Net Ltd.</td>
<td>descr:</td>
<td>CUSTOMERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>origin:</td>
<td>AS8551</td>
<td>country:</td>
<td>IL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notify:</td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td>admin-c:</td>
<td>ES4966-RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mnt-by:</td>
<td>AS8551-MNT</td>
<td>tech-c:</td>
<td>NP469-RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>changed:</td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td>status:</td>
<td>ASSIGNED PA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td>notify:</td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td>mnt-by:</td>
<td>RIPE-NCC-NONE-MNT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td>changed:</td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td>19990610</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source:</td>
<td>RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>person:</td>
<td>Zehavit Vigder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address:</td>
<td>bezeq-international</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address:</td>
<td>40 hashacham</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address:</td>
<td>petach tikva 49170 Israel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phone:</td>
<td>+972 52 770145</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fax-no:</td>
<td>+972 9 8940763</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e-mail:</td>
<td><a href="mailto:hostmaster@bezeqint.net">hostmaster@bezeqint.net</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nic-hdl:</td>
<td>ZV140-RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>changed:</td>
<td><a href="mailto:zehavitv@bezeqint.net">zehavitv@bezeqint.net</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:hostmaster@isdn.net.il">hostmaster@isdn.net.il</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source:</td>
<td>RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>person:</td>
<td>Nati Pinko</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address:</td>
<td>Bezeq International</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address:</td>
<td>40 Hashacham St.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address:</td>
<td>Petach Tikva 49170 Israel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phone:</td>
<td>+972 3 9257761</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fax-no:</td>
<td>+972 3 9257726</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e-mail:</td>
<td><a href="mailto:hostmaster@bezeqint.net">hostmaster@bezeqint.net</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nic-hdl:</td>
<td>NP469-RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>changed:</td>
<td><a href="mailto:registrar@ns.il">registrar@ns.il</a> 19990902</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:registrar@ns.il">registrar@ns.il</a> 20000309</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source:</td>
<td>RIPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Trying 212.179.7.2 at ARIN**

Trying 212.179.7 at ARIN
Redirecting to RIPE ...
Trying 212.179.7.2 at RIPE
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/ripsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pubs-services/db/copyright.html

| person:         | Eran Shchori             |                |
| address:        | BEZEQ INTERNATIONAL      |                |
| address:        | 40 Hashacham Street      |                |
| address:        | Petach-Tikva 49170 Israel|                |
| phone:          | +972 3 9257710           |                |
| fax-no:         | +972 3 9257726           |                |
| e-mail:         | hostmaster@bezeqint.net  |                |
| nic-hdl:        | ES4966-RIPE              |                |
| changed:        | registrar@ns.il 19990902 |                |
|                 | registrar@ns.il 20000309 |                |
| source:         | RIPE                     |                |

© SANS Institute 2000 - 2005
Author retains full rights.
As you can see in the chart above, there are 7,562 alerts. Of this I have provided a list of the source hosts and number of connections from each. There are three tables, the first shows incoming Gnutella (Port 6346 & 6347) connections, the second shows incoming Napster (Port 6688, 6699 & 6700), the third has what is left over. I included the destination host that received the majority of the connections as well. NOTE: Most of the senders communicated with more than one host, but I included the single host receiving the most connections.

Table 18 - Watchlist Gnutella Senders

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>212.179.95.5</td>
<td>MY.NET.217.186</td>
<td>1905</td>
</tr>
<tr>
<td>212.179.83.137</td>
<td>MY.NET.227.90</td>
<td>145</td>
</tr>
<tr>
<td>212.179.5.184</td>
<td>MY.NET.209.42</td>
<td>101</td>
</tr>
<tr>
<td>212.179.27.6</td>
<td>MY.NET.209.130</td>
<td>14</td>
</tr>
<tr>
<td>212.179.21.185</td>
<td>MY.NET.225.74</td>
<td>4</td>
</tr>
<tr>
<td>212.179.5.184</td>
<td>MY.NET.225.138</td>
<td>1</td>
</tr>
<tr>
<td>212.179.81.254</td>
<td>MY.NET.227.38</td>
<td>1</td>
</tr>
</tbody>
</table>

Gnutella clients and servers use registered Ports 6346 & 6347. Apply ALL system patches and install Anti-Virus software.

Table 19 - Watchlist Napster Senders

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>212.179.21.187</td>
<td>MY.NET.218.30</td>
<td>1580</td>
</tr>
<tr>
<td>212.179.80.3</td>
<td>MY.NET.222.2</td>
<td>664</td>
</tr>
<tr>
<td>212.179.7.12</td>
<td>MY.NET.225.102</td>
<td>463</td>
</tr>
<tr>
<td>212.179.77.53</td>
<td>MY.NET.224.230</td>
<td>414</td>
</tr>
<tr>
<td>212.179.81.2</td>
<td>MY.NET.218.218</td>
<td>328</td>
</tr>
<tr>
<td>212.179.17.4</td>
<td>MY.NET.205.242</td>
<td>247</td>
</tr>
<tr>
<td>212.179.81.110</td>
<td>MY.NET.219.218</td>
<td>68</td>
</tr>
</tbody>
</table>

Napster clients and serves communicate on three un-registered ports 6688, 6700 & 6699. Apply all patches and use Anti-virus software.

Table 20 - Watchlist Top Senders (Excluding Gnutella & Napster)

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>212.179.80.30</td>
<td>MY.NET.219.38</td>
<td>1010</td>
</tr>
<tr>
<td>212.179.82.119</td>
<td>MY.NET.97.204</td>
<td>317</td>
</tr>
<tr>
<td>212.179.33.168</td>
<td>MY.NET.219.38</td>
<td>106</td>
</tr>
<tr>
<td>212.179.79.2</td>
<td>MY.NET.229.6</td>
<td>71</td>
</tr>
<tr>
<td>212.179.27.6</td>
<td>MY.NET.225.138</td>
<td>31</td>
</tr>
<tr>
<td>212.179.7.182</td>
<td>MY.NET.97.193</td>
<td>22</td>
</tr>
<tr>
<td>212.179.67.192</td>
<td>MY.NET.219.38</td>
<td>16</td>
</tr>
<tr>
<td>212.179.7.10</td>
<td>MY.NET.219.38</td>
<td>11</td>
</tr>
<tr>
<td>212.179.82.68</td>
<td>MY.NET.219.38</td>
<td>8</td>
</tr>
<tr>
<td>212.179.7.41</td>
<td>MY.NET.219.38</td>
<td>6</td>
</tr>
<tr>
<td>212.179.16.228</td>
<td>MY.NET.219.38</td>
<td>6</td>
</tr>
<tr>
<td>212.179.80.60</td>
<td>MY.NET.202.110</td>
<td>4</td>
</tr>
<tr>
<td>212.179.34.215</td>
<td>MY.NET.219.38</td>
<td>4</td>
</tr>
<tr>
<td>212.179.84.121</td>
<td>MY.NET.219.38</td>
<td>3</td>
</tr>
<tr>
<td>212.179.82.225</td>
<td>MY.NET.219.38</td>
<td>3</td>
</tr>
<tr>
<td>212.179.82.30</td>
<td>MY.NET.219.38</td>
<td>2</td>
</tr>
<tr>
<td>212.179.80.20</td>
<td>MY.NET.219.38</td>
<td>2</td>
</tr>
<tr>
<td>212.179.68.226</td>
<td>MY.NET.222.202</td>
<td>2</td>
</tr>
<tr>
<td>212.179.95.5</td>
<td>MY.NET.225.138</td>
<td>1</td>
</tr>
<tr>
<td>212.179.82.55</td>
<td>MY.NET.223.66</td>
<td>1</td>
</tr>
<tr>
<td>212.179.80.38</td>
<td>MY.NET.202.226</td>
<td>1</td>
</tr>
<tr>
<td>212.179.80.102</td>
<td>MY.NET.219.38</td>
<td>1</td>
</tr>
<tr>
<td>212.179.7.230</td>
<td>MY.NET.212.106</td>
<td>1</td>
</tr>
<tr>
<td>212.179.56.5</td>
<td>MY.NET.213.218</td>
<td>1</td>
</tr>
<tr>
<td>212.179.5.92</td>
<td>MY.NET.227.158</td>
<td>1</td>
</tr>
<tr>
<td>212.179.5.184</td>
<td>MY.NET.225.138</td>
<td>1</td>
</tr>
<tr>
<td>212.179.41.141</td>
<td>MY.NET.219.38</td>
<td>1</td>
</tr>
<tr>
<td>212.179.36.68</td>
<td>MY.NET.213.218</td>
<td>1</td>
</tr>
<tr>
<td>212.179.25.27</td>
<td>MY.NET.219.38</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 13 shows us that there are a lot of folks talking to MY.NET.219.38. Check that host for possible compromise, apply all patches, close un-needed services, enforce use of Anti-Virus software on this host.

---


97 IANA Port Numbers, [http://www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers)
3-2-15-2 Correlation(s):

Seventy-three GCIA practicals (minus html and zip archives) contain references to Watchlist 000220 IL-ISDNET\(^{100}\)

From SANS Detects Analyzed –

**NOTE:** On 5/20/2000, John Green (SANS Handler on duty) starts the days Detects Analyzed, 5/20/00 with a statement about Gnutella/Napster and the desensitization of analysts towards this type of traffic. I agree with Mr. Green and would like to add Network games (Quake, Total Anihilation, Doom, etc..) to that list things we have gotten used to.

[http://www.sans.org/y2k/033000-2300.htm](http://www.sans.org/y2k/033000-2300.htm)
[http://www.sans.org/y2k/051900.htm](http://www.sans.org/y2k/051900.htm)
[http://www.sans.org/y2k/052000.htm](http://www.sans.org/y2k/052000.htm)
[http://www.sans.org/y2k/090500-1200.htm](http://www.sans.org/y2k/090500-1200.htm)
[http://www.sans.org/y2k/112600.htm](http://www.sans.org/y2k/112600.htm)

3-2-15-3 Defensive Recommendations:

Check the systems that have the high receive counts to ensure that they have all of the latest system patches. With the heavy use of Gnutella/Napster, I would recommend calling the users and warning them of the dangers involved with Gnutella/Napster. Every host that sent data to a system on the MY.HOST network is listed in the tables above. I would recommend that you check the Router logs for additional correlation of traffic from these hosts.

Apply all operating system patches on all systems communicating with this domain (You should do this to all systems and not just these). Remove un-needed services. Install Anti-Virus software and keep it current. On Unix based systems, you may want to install Tripwire to monitor file activity. Review the syslogs regularly.

---


\(^{100}\) RIPE Registry Information on ISDNet, Ltd. [http://www.ripe.net/ripencc/mem-services/general/indices/data/il.isdnnet.html](http://www.ripe.net/ripencc/mem-services/general/indices/data/il.isdnnet.html)
3-2-16 Watchlist 000222

During the period 04/10/2001 to 04/16/2001, there were 158 recorded alerts for Watchlist 000222 in the IDS Logs.

3-2-16-1 Description:

From the practical of Miika Turkia\(^{101}\)

These are connections from the Computer Network Center Chinese Academy of Sciences. These are alerted since they belong to a watchlist.

MY.NET.253.43, MY.NET.4.3 and MY.NET.6.25 show SMTP connections.

Port 8765 was targeted seven times on MY.NET.70.33. A check of the IANA Port Numbers web site shows that this is registered to the Ultraseek-HTTP service. Checking for exploits on Neohapsis shows that there have been at least three Buffer Overflow Exploits reported (Jan 1999\(^{102}\), Dec 1999\(^{103}\), Oct 2000\(^{104}\)).

The FTP (20 & 21) port on MY.NET.144.54 was accessed 40 times.

MY.NET.6.7 was accessed 52 times on the Telnet port from the same host on 4/10/2001 and

\(^{101}\) Miika Turkia GCIA Practical, SANS, [http://www.sans.org/y2k/practical/Miika_Turkia_GCIA.html](http://www.sans.org/y2k/practical/Miika_Turkia_GCIA.html)


Telnet sessions to ports 21776 and 21817 on MY.NET.110.164 were made on 4/11/2001. There were two sessions that lasted five minutes each between 13:14 – 13:19 and 13:32 – 13:37. Neither of the destination ports is for a registered service. I would investigate this machine to see what happened during those two five minute telnet sessions.

Table 21 - Watchlist 000222 Port and Host Information

<table>
<thead>
<tr>
<th>DST PORTS</th>
<th>HITS</th>
<th>SRC PORTS</th>
<th>HITS</th>
<th>Destination IP</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>52</td>
<td>62100</td>
<td>43</td>
<td>MY.NET.100.230159.226.120.14</td>
<td></td>
</tr>
<tr>
<td>21776</td>
<td>21</td>
<td>21</td>
<td>37</td>
<td>MY.NET.110.164159.226.194.26</td>
<td></td>
</tr>
<tr>
<td>21817</td>
<td>13</td>
<td>23</td>
<td>34</td>
<td>MY.NET.144.54159.226.21.20</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>9</td>
<td>63099</td>
<td>9</td>
<td>MY.NET.253.43159.226.228.1</td>
<td></td>
</tr>
<tr>
<td>8765</td>
<td>7</td>
<td>20</td>
<td>3</td>
<td>MY.NET.253.51159.226.252.11</td>
<td></td>
</tr>
<tr>
<td>1580</td>
<td>4</td>
<td>1081</td>
<td>2</td>
<td>MY.NET.253.52159.226.41.166</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>113</td>
<td>2</td>
<td>MY.NET.4.3159.226.42.180</td>
<td></td>
</tr>
<tr>
<td>1553</td>
<td>3</td>
<td>1243</td>
<td>2</td>
<td>MY.NET.6.35159.226.45.3</td>
<td></td>
</tr>
<tr>
<td>3176</td>
<td>3</td>
<td>2548</td>
<td>2</td>
<td>MY.NET.6.7159.226.47.195</td>
<td></td>
</tr>
<tr>
<td>3679</td>
<td>3</td>
<td>4269</td>
<td>2</td>
<td>MY.NET.70.33159.226.47.5</td>
<td></td>
</tr>
<tr>
<td>4352</td>
<td>3</td>
<td>63931</td>
<td>2</td>
<td>159.226.47.56</td>
<td></td>
</tr>
<tr>
<td>1165</td>
<td>2</td>
<td>1295</td>
<td>1</td>
<td>159.226.5.222</td>
<td></td>
</tr>
<tr>
<td>1227</td>
<td>2</td>
<td>15055</td>
<td>1</td>
<td>159.226.63.200</td>
<td></td>
</tr>
<tr>
<td>1321</td>
<td>2</td>
<td>1987</td>
<td>1</td>
<td>159.226.92.9</td>
<td></td>
</tr>
<tr>
<td>1698</td>
<td>2</td>
<td>2599</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2581</td>
<td>2</td>
<td>26312</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4337</td>
<td>2</td>
<td>2943</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4373</td>
<td>2</td>
<td>3113</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4391</td>
<td>2</td>
<td>32072</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1587</td>
<td>1</td>
<td>32903</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1707</td>
<td>1</td>
<td>36602</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2569</td>
<td>1</td>
<td>37778</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3078</td>
<td>1</td>
<td>38161</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4295</td>
<td>1</td>
<td>38858</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4311</td>
<td>1</td>
<td>3894</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4401</td>
<td>1</td>
<td>62893</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49670</td>
<td>1</td>
<td>63887</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50849</td>
<td>1</td>
<td>63888</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>63889</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>63898</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>63935</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3-2-16-2 Correlation(s):

Seventy-seven GCIA Practicals mention “Watchlist 000222. NET-NCFC”.

From the SANS Detects Analyzed –
3-2-16-3 Defensive Recommendations:

Check your Router logs for additional correlation of traffic from these hosts.

Investigate each machine involved in this watchlist, because of the heavy use of Telnet, FTP, SMTP and UltraSeek. Any or all of them could be compromised. You should install the latest versions of Sendmail, and FTP on each of these machines. Disable Telnet and install Secure Shell.

Apply all operating system patches on all systems communicating with this domain (You should do this to all systems and not just these). Remove un-needed services. Patch those third party services that are installed (Ultraseek-HTTP service on MY.NET.70.33). Install Anti-Virus software and keep it current. On Unix based systems, you may want to install Tripwire to monitor file activity. Review the syslogs regularly.
3-2-17 WINGATE 1080 Attempt

During the period 04/10/2001 to 04/16/2001, there were 165 recorded alerts for WINGATE 1080 Attempt in the IDS Logs.

3-2-17-1 Description/Discussion:

From page 479 of Hacking Exposed: Network Security Secrets & Solutions, 2d Edition:

The popular Windows proxy firewall WinGate (http://wingate.deerfield.com) has been known to have a couple of vulnerabilities. Most of these stem from the lax default parameters including unauthenticated telnet, SOCKS, and Web. While access to these services can be restricted by user (and interface), many simply install the product as is to get it up and running – forgetting about security.

Like many misconfigured proxies, certain WinGate versions (Specifically 2.1d for NT) allow outsiders to browse the Internet completely anonymously. This is important for attackers who target web server applications in particular, as then can hack to their heart’s content with little risk of getting caught.

Also vulnerable in the default configuration is the unauthenticated SOCKS proxy (TCP 1080). As with open Web proxy (TCP 80), an attacker can browse the Internet, remaining almost completely anonymous (especially if logging is turned off).

---

There were systems on the MY.NET network probed by sixty systems outside the MY.NET network. The top five MY.NET systems and the number of times each was probed along with the top five probers are listed in the following table:

<table>
<thead>
<tr>
<th>Destination IP</th>
<th>Hits</th>
<th>Probers IP</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MY.NET.53.89</td>
<td>31</td>
<td>204.117.70.5</td>
<td>24</td>
</tr>
<tr>
<td>MY.NET.98.189</td>
<td>6</td>
<td>217.10.143.54</td>
<td>16</td>
</tr>
<tr>
<td>MY.NET.204.102</td>
<td>5</td>
<td>63.102.227.48</td>
<td>9</td>
</tr>
<tr>
<td>MY.NET.53.99</td>
<td>5</td>
<td>216.179.0.32</td>
<td>7</td>
</tr>
<tr>
<td>MY.NET.202.150</td>
<td>4</td>
<td>195.66.170.8</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 22 - WINGATE 1080 Top Five

I will concentrate on the top three most active MY.NET hosts and the top three most active probers.

MY.NET.53.89 was probed 31 times. Eight hosts accessed this system from outside the MY.NET network a minimum of twice. Two hosts accessed this system six times each. A check of the Alerts Logs also finds alerts for Possible Trojan Server Activity as well. 213.51.32.67 attempted to access port 27374 on 4/13/2001. All of the Wingate 1080 alerts occurred on that day as well. All of these alerts (Wingate 1080 and Trojan Server Activity ) occurred between 09:13 and 10:01 on 4/13/2001. I would investigate this machine in more detail.

MY.NET.98.189 was probed six times. All the probes occurred at various times between 01:43 and 080:10 on 04/10/2001. Five of the six attempts were from 63.102.227.48 (our number three top prober). There are two packets logged in the Portscan Logs from hosts that are not listed in the probers list.

MY.NET.204.102 was probed five times 205.167.47.146. Three times on 04/10/2001 and once on 4/11/2001. There is one entry in the Portscan Log where someone tried to access 5555.

204.117.70.5 (Owned by US Sprint) probed the MY.NET network 24 times. A check of the Alerts Logs show that the Wingate 1080 probe is the only activity recorded on him. Twelve MY.NET hosts were probed.

```
nslookup 204.117.70.5
Canonical name: 204.117.70.5
Addresses:
  204.117.70.5
```

**Trying 204.117.70 at ARIN**

US Sprint (NETBLK-SPRINT-BLKB) SPRINT-BLKB 204.117.0.0 - 204.120.255.255

TELE-TECH COMPANY (NETBLK-FON-343023769634089) FON-343023769634089

204.117.70.0 - 204.117.70.255
217.10.143.54 (UKSolutions Network Operations Centre) probed the MY.NET network sixteen times. A check of the Alerts Logs show that the Wingate 1080 probe is the only activity recorded from this host. Fifteen MY.NET hosts were probed.

**nslookup 217.10.143.54**
Canonical name: 217.10.143.54
Addresses:
217.10.143.54

**Trying 217.10.143.54 at RIPE**
Trying 217.10.143 at RIPE
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:      217.10.143.0 - 217.10.143.3
netname:      UKSOLUTIONS-CORE
descr:        Network routing devices
country:      GB
admin-c:      US5708-RIPE
tech-c:       US5708-RIPE
rev-srv:      ns0.uksolutions.co.uk
rev-srv:      ns1.uksolutions.co.uk
status:       ASSIGNED PA
notify:       ripe@uksolutions.co.uk
mnt-by:       UKS-MNT
changed:      ripe@uksolutions.co.uk 20000928
source:       RIPE
route:        217.10.128.0/20
descr:        UKSOLUTIONS-217.10.128/20
origin:       AS20547
notify:       ripe@uksolutions.co.uk
mnt-by:       UKS-MNT
changed:      ripe@uksolutions.co.uk 20010405
source:       RIPE
role:         UKSolutions Support
address:      UKSolutions Network Operations Centre
address:      CAD Building
address:      Birmingham Road
address:      Studley
address:      Warwickshire
63.102.227.48 (chatspace.com) probed the MY.NET network nine times. Five of those probes are already accounted for above. The remaining four were split evenly between MY.NET.98.186 and MY.NET.98.141.

**Trying 63.102.227.48 at ARIN**

Trying 63.102.227 at ARIN

UUNET Technologies, Inc. (NETBLK-UUNET63) UUNET63 63.64.0.0 - 63.127.255.255

Inflow (NETBLK-UU-63-102-224) UU-63-102-224 63.102.224.0 - 63.102.227.255

chatspace.com (NETBLK-INFLOW-CHT2) INFLOW-CHT2 63.102.226.0 - 63.102.227.255

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

This is reconnaissance. We didn’t see any suspicious outbound connections logged.

**3-2-17-2 Correlation(s):**

Several CVE’s and CAN’s are provided.

CVE-1999-0290 -

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0290

The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number
of connections to localhost.

CVE-1999-0291 -
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-02901
The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.

CVE-1999-0441 -
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0441
Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.

CVE-1999-0494 -
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0494
Denial of service in WinGate proxy through a buffer overflow in POP3.

CAN-1999-0657 -
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0657
WinGate is being used. (Proposed).

CAN-2000-1048
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-1048
Directory traversal vulnerability in the logfile service of Wingate 4.1 Beta A and earlier allows remote attackers to read arbitrary files via a .. (dot dot) attack via an HTTP GET request that uses encoded characters in the URL.

A Neohapsis Archives search provided four pages of links from 1998 to present, here are several.


3-2-17-3 Defensive Recommendations:

You may want to run a port scan on your own to see how many systems have port 1080 or 8080 open. If you find any, configure the security, enable logging and apply all vendor patches to all of them. After this is done, shutdown or disable what you don’t need. Remove the software/service if possible. In case someone turns one of them on again they will at least be patched and properly configured.

Apply all patches on the Operating Systems. Enforce the use of Anti-virus software and keep it updated. Run periodic Anti-virus scans of all files.
3-2-18 Null Scan

During the period 04/10/2001 to 04/16/2001, there were 24 NULL Scans recorded in the IDS Logs.

![Null Scan Chart]

3-2-18-1 Description/Discussion:

From the NMAP\textsuperscript{106} Manpage:

The Null scan turns off all flags. Unfortunately Microsoft (like usual) decided to completely ignore the standard and do things their own way. Thus this scan type will not work against systems running Windows95/NT. On the positive side, this is a good way to distinguish between the two platforms. If the scan finds open ports, you know the machine is not a Windows box.

A search of the current Whitehats.com and Snort.org rules finds one rule:

\begin{verbatim}
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS004 - SCAN-NULL Scan";flags:0; seq:0; ack:0;)
\end{verbatim}

There were 24 different hosts scanned from twenty-four different IP Addresses. I have compiled a list of all of the ports scanned, the number of times each port was scanned and the service that is registered or known to use that port. Of all the ports scanned, one of them is associated with four Trojans. Known Trojan ports were found on the The Trojan List\textsuperscript{107}


\textsuperscript{107} The Trojan Port List - [http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html](http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html)
### Table 23 - NULL Scan Ports Scanned List

<table>
<thead>
<tr>
<th>DST Port</th>
<th>Hits</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>6346</td>
<td>4</td>
<td>Gnutella</td>
</tr>
<tr>
<td>6969</td>
<td>1</td>
<td>Unassigned - (GateCrasher, IRC 3, Net Controller, Priority)</td>
</tr>
<tr>
<td>6688</td>
<td>1</td>
<td>Napster</td>
</tr>
<tr>
<td>6699</td>
<td>2</td>
<td>Napster</td>
</tr>
<tr>
<td>6347</td>
<td>1</td>
<td>Gnutella</td>
</tr>
<tr>
<td>62821</td>
<td>1</td>
<td>Dynamic-Private Port Range</td>
</tr>
<tr>
<td>49631</td>
<td>1</td>
<td>Dynamic-Private Port Range</td>
</tr>
<tr>
<td>4850</td>
<td>1</td>
<td>Unassigned</td>
</tr>
<tr>
<td>4831</td>
<td>1</td>
<td>Unassigned</td>
</tr>
<tr>
<td>4453</td>
<td>1</td>
<td>NSS Alert Manager</td>
</tr>
<tr>
<td>4355</td>
<td>1</td>
<td>Unassigned</td>
</tr>
<tr>
<td>4036</td>
<td>1</td>
<td>WAP Push OTA-HTTP secure</td>
</tr>
<tr>
<td>3619</td>
<td>1</td>
<td>Unassigned</td>
</tr>
<tr>
<td>3004</td>
<td>1</td>
<td>Csoft Agent</td>
</tr>
<tr>
<td>2813</td>
<td>1</td>
<td>IIm-pass</td>
</tr>
<tr>
<td>2696</td>
<td>1</td>
<td>Unify Admin</td>
</tr>
<tr>
<td>1790</td>
<td>1</td>
<td>Narrative Media Streaming Protocol</td>
</tr>
<tr>
<td>1556</td>
<td>1</td>
<td>AshWin CI Tecnologies</td>
</tr>
<tr>
<td>1518</td>
<td>1</td>
<td>Virtual Places Video data</td>
</tr>
<tr>
<td>1147</td>
<td>1</td>
<td>Unassigned</td>
</tr>
</tbody>
</table>

Other than the one scan directed at Port 6969, the only other major scan activity was for Gnutella/Napster. The top source port was 6346 and was used six times. Fifty percent of the NULL Scans had a source or destination port associated with Gnutella/Napster. MY.NET.221.14 was probed on port 6969 and should be investigated at the earliest possible moment.

### 3-2-18-2 Correlation(s):

From a search of SANS.ORG:

- [http://www.sans.org/y2k/011900.htm](http://www.sans.org/y2k/011900.htm)
- [http://www.sans.org/y2k/020800-2300.htm](http://www.sans.org/y2k/020800-2300.htm)
- [http://www.sans.org/y2k/032200-1700.htm](http://www.sans.org/y2k/032200-1700.htm)
- [http://www.sans.org/y2k/053100-1100.htm](http://www.sans.org/y2k/053100-1100.htm)

A Neohapsis Archives search:

3-2-18-3 Defensive Recommendations:

For starters I recommend blocking all Gnutella/Napster ports. Otherwise, apply all operating system patches, close all unneeded ports and shut off all unneeded services. Install Antivirus software and keep it current.
3-2-19 Port 55850 TCP – possible myserver activity

During the period 04/10/2001 to 04/16/2001, there were 20 alerts for Port 55850 TCP – Possible myserver activity recorded in the IDS Logs.

![Graph showing Port 55850 TCP connections]

Table 24 - Port 55850 TCP Connections

<table>
<thead>
<tr>
<th>Count</th>
<th>Source IP</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>MY.NET.6.34</td>
<td>55850165.251.8.76</td>
</tr>
<tr>
<td>3</td>
<td>207.217.120.22</td>
<td>55850MY.NET.253.43</td>
</tr>
<tr>
<td>3</td>
<td>MY.NET.253.43</td>
<td>25207.217.120.22</td>
</tr>
<tr>
<td>2</td>
<td>64.232.129.197</td>
<td>55850MY.NET.253.53</td>
</tr>
<tr>
<td>2</td>
<td>MY.NET.253.24</td>
<td>55850195.241.148.134</td>
</tr>
<tr>
<td>1</td>
<td>165.251.8.76</td>
<td>25MY.NET.6.34</td>
</tr>
<tr>
<td>1</td>
<td>MY.NET.100.201</td>
<td>8080MY.NET.101.140</td>
</tr>
<tr>
<td>1</td>
<td>MY.NET.227.90</td>
<td>6346194.237.76.4</td>
</tr>
<tr>
<td>1</td>
<td>MY.NET.253.24</td>
<td>55850134.220.1.46</td>
</tr>
<tr>
<td>1</td>
<td>MY.NET.253.51</td>
<td>55850204.255.212.10</td>
</tr>
<tr>
<td>1</td>
<td>MY.NET.253.53</td>
<td>5585064.4.56.199</td>
</tr>
</tbody>
</table>

3-2-19-1 Description/Discussion:

This appears to be a rule to alert the possibility of a MyServer exploit. I believe it is a bit broad since the exploit is actually implemented via an RPC exploit. If this triggers, then a simple check of rpc activity to the same host would further indicate that a check of the host in question is needed.

In addition to 55850, you can also use the information in the SANS Detects Analyzed for
082200\textsuperscript{108} (Extracted portion here) and monitor/check ports 9704 and 111 as well:

The following note was sent to us in response to our telling them about an attack originating from their site. I've been in contact with the netadmins at UMass and they saw something similar. They're sending intrusion@sans.org what they found. Anyway, seems like we have a new variant floating around the net these days. The Umass guys found a Trinoo-style tool called MyServer on their linux box. Randy Marchany

>===== Original Message From Joakim Bergkvist <Joakim.F.Bergkvist@telia.se>

Hi Just for your information the status is as follows. We've had (have ??) a hacker in some of our lab servers. The hacker has targeted Linux redhat6.x machines using the RPC stat exploit. Essentially this exploit allows the hacker to send shell commands via the portmapper which will be executed with root privileges. The hacker first scans a list of target addresses watching for the response on port 23 and 25 to try to discern which OS and distribution it is.. The scan script makes another list with all redhat machines and batch runs the exploit on these sending commands to append a line to inetd.conf for starting a shell on port 9704 and restarting inetd. -- When you've seen the RPC info query in your trace watch out for the shell -- On some of the machines the hacker has entered through the shell and patched some files in the distribution. typically 'ps', 'netstat' and 'ls' to filter out the shell and some given file locations and of course 'login'

From a portion of a post\textsuperscript{109} found by doing a search of the Neohapsis Archives:

MyServer is a little known DDOS agent that was running around late in the summer. It binds to UDP 55850, and the rootkit installs trojans of ls and ps, so you won't see it running. You WILL see it with netstat though. The rootkit and ddos tools are stored in "\text{lib/}\"

With the exception of one connection from port 6346 to 55850 and another from 8080 to 55850, the remaining eighteen alerts use ports 25 and 55850. I did a search of the Portscan Logs and found 145 scans for port 9704, none of which contained IP Addresses alerted on in the Port 55850 alert. The same is true for a port 111 search. All of the alerts concern hosts MY.NET.253.43, MY.NET.253.24, MY.NET.253.53, MY.NET.253.51, and MY.NET.6.34.

MY.NET 253.43 has nine Watchlist 000222 alerts and six port 55850 alerts between 04/10/2001 and 04/16/2001.

MY.NET.253.24 has 3 port 55850 and four possible redworm alerts between 04/10/2001 and 04/14/2001.

MY.NET.253.53 has four port 55850 and two possible redworm alerts on 04/11/2001 and 04/12/2001.

\textsuperscript{108} SANS Detects Analyzed for 082200 \url{http://www.sans.org/082200.htm}

\textsuperscript{109} Neohapsis Archives Message, Subj: Connection From Unknown \url{http://archives.neohapsis.com/archives/incidents/2000-10/0136.html}
MY.NET.253.51 has one port 55850 and one watchlist 000222 alert on 04/13/2001 and 04/16/2001.

MY.NET.6.34 has five port 55850 alerts on 04/14/2001.

e see no portmapper activity to any of these hosts, so this ends the discussion.

3-2-19-2 Correlation(s):

A search of the Consensus Intrusion Database\(^{110}\) (CID) at Incidents org shows 1 reported incident of attempted access to port 55850

A search of SecurityFocus found a copy of the message that was sent to SANS (from above) and the following additional link - [http://www.securityfocus.com/archive/75/139765](http://www.securityfocus.com/archive/75/139765).

3-2-19-3 Defensive Recommendations:

Apply all patches. I was not able to find any additional detailed information on this exploit other than the two mentioned in the description above. I would consider dropping this rule and replacing it with a more specific rule that checks for content. It may be you could drop this one completely and update your snort rule set. The current RPC rules may be all that you need.

\(^{110}\) [WWW.INCIDENTS.ORG](http://www.incidents.org/cid/search.php), Search the Consensus Intrusion Database (CID)
3-3 Port scans:

3-3-1 Who’s scanning’ who and the Type of scans performed:

<table>
<thead>
<tr>
<th>Date</th>
<th>Total Scanners</th>
<th>MY.NET</th>
<th>External</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/10</td>
<td>92</td>
<td>69</td>
<td>23</td>
</tr>
<tr>
<td>04/11</td>
<td>86</td>
<td>66</td>
<td>20</td>
</tr>
<tr>
<td>04/12</td>
<td>95</td>
<td>73</td>
<td>22</td>
</tr>
<tr>
<td>04/13</td>
<td>82</td>
<td>63</td>
<td>19</td>
</tr>
<tr>
<td>04/14</td>
<td>89</td>
<td>47</td>
<td>42</td>
</tr>
<tr>
<td>04/15</td>
<td>70</td>
<td>56</td>
<td>14</td>
</tr>
<tr>
<td>04/16</td>
<td>79</td>
<td>60</td>
<td>19</td>
</tr>
<tr>
<td>Totals</td>
<td>593</td>
<td>434</td>
<td>159</td>
</tr>
</tbody>
</table>

Seventy-Three percent of the 593 hosts performing scans were from inside the MY.NET network. If you take into account the repeat offenders on the MY.NET network, then the total number of hosts is 576 and the total number of MY.NET hosts is 417. Repeat offenders are hosts that appear in the top ten port scanners more than once in a week.

<table>
<thead>
<tr>
<th>Number</th>
<th>Type of Scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>151171</td>
<td>UDP</td>
</tr>
<tr>
<td>41594</td>
<td>SYN</td>
</tr>
<tr>
<td>143</td>
<td>INVALIDACK</td>
</tr>
<tr>
<td>137</td>
<td>NOACK</td>
</tr>
<tr>
<td>109</td>
<td>NULL</td>
</tr>
<tr>
<td>47</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>40</td>
<td>VECNA</td>
</tr>
<tr>
<td>18</td>
<td>FULLXMAS</td>
</tr>
<tr>
<td>11</td>
<td>NMAPID</td>
</tr>
<tr>
<td>11</td>
<td>FIN</td>
</tr>
<tr>
<td>7</td>
<td>XMAS</td>
</tr>
<tr>
<td>1</td>
<td>SPAU</td>
</tr>
<tr>
<td>193289</td>
<td>Total Scans</td>
</tr>
</tbody>
</table>

3-3-2 The Top Five:

The Overall Top Five is taken from the merged scan logs for the reporting period. The overall Top Five scanning hosts list is comprised almost entirely of systems from inside the MY.NET network. This is primarily due to the high number of UDP scans reported, which originated entirely from MY.NET hosts. The large amount of UDP scans is due almost entirely to game traffic. Ports 27xxx and 28xxx accounts for 63,493. Port 13139 appears 23,931 times. Ports 7777 & 7778 appear 7,185 times. That alone totals 94,609 entries for games. And those are only the games we know about.
Table 27 - Overall Top Five Scanning Hosts

<table>
<thead>
<tr>
<th>Packets</th>
<th>IP Address</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>7936</td>
<td>MY.NET.220.66</td>
<td>04/15</td>
</tr>
<tr>
<td>7137</td>
<td>MY.NET.228.50</td>
<td>04/12</td>
</tr>
<tr>
<td>7039</td>
<td>MY.NET.224.106</td>
<td>04/11</td>
</tr>
<tr>
<td>6932</td>
<td>MY.NET.211.114</td>
<td>04/16</td>
</tr>
<tr>
<td>6329</td>
<td>210.220.73.117</td>
<td>04/10</td>
</tr>
</tbody>
</table>

Table 28 - Popular Game Ports

- Quake 1/QW: 27500 (27500->27600)
- Quake 2: 27910 (27900->27930)
- Quake 3: 27960 (27960->27980)
- halfLife: 27015 (27010 -> 27050)
- Unreal tournament: 7777 (7777->7797)
- Kingpin: 31510 (31500->31550)
- Shogo: 27888
- Star Siege: 28000 (28001 & 2 often too)

** Some of the MY.NET hosts appear to have GameSpy\(^{111}\) installed as well. This is an application that probes game servers in order to provide you with the status of available servers for game playing over the internet. It uses UDP Pings on port 13139\(^{112}\) to check the status and round trip time to servers.

3-3-3 Repeat Offenders:

The repeat offenders are those hosts that appear in the daily top five lists that we merged into a single list and sorted again on number of packets generated. All duplicates entries were then removed. Three MY.NET hosts that appear in the repeat offenders list below also appear in the top five list above.

Table 29 - Top Five MY.NET Scanners

<table>
<thead>
<tr>
<th>Packets</th>
<th>IP Address</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>7936</td>
<td>MY.NET.220.66</td>
<td>04/15</td>
</tr>
<tr>
<td>7137</td>
<td>MY.NET.228.50</td>
<td>04/12</td>
</tr>
<tr>
<td>7039</td>
<td>MY.NET.224.106</td>
<td>04/11</td>
</tr>
<tr>
<td>6932</td>
<td>MY.NET.211.114</td>
<td>04/16</td>
</tr>
<tr>
<td>4590</td>
<td>MY.NET.219.34</td>
<td>04/16</td>
</tr>
</tbody>
</table>

3-3-4 Top Five External Scanners:

\(^{111}\) GameSpy, [http://www.gamespy.com](http://www.gamespy.com)

\(^{112}\) MultiPlayer Total Annihilation behind a firewall, [http://www.estrella.demon.nl/impfw.htm](http://www.estrella.demon.nl/impfw.htm)
Table 30 - Top Five External Scanners

<table>
<thead>
<tr>
<th>Packets</th>
<th>IP Address</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>6329</td>
<td>210.220.73.117</td>
<td>04/10</td>
</tr>
<tr>
<td>3972</td>
<td>209.178.22.233</td>
<td>04/14</td>
</tr>
<tr>
<td>3349</td>
<td>63.163.94.13</td>
<td>04/10</td>
</tr>
<tr>
<td>2499</td>
<td>210.52.214.15</td>
<td>04/15</td>
</tr>
<tr>
<td>2346</td>
<td>216.40.195.72</td>
<td>04/10</td>
</tr>
</tbody>
</table>

There are no repeat offenders in the External Scanners data. Each one of the five hosts listed above performed a scan for FTP or DNS Servers. 210.220.73.117, 210.52.214.15 performed SYN scans for FTP servers. 209.178.22.233, 63.163.94.13, and 216.40.195.72 performed SYN scans for DNS servers. Whois lookup information (Using Sam Spade\textsuperscript{113}) on each external host is provided below.

Trying 210.220.73.117 at APNIC
Trying 210.220.73 at APNIC
Trying 210.220 at APNIC

\textsuperscript{113} Sam Spade for Windows, Freeware, \url{http://samspade.org/ssw/}

---

\textsuperscript{113} Sam Spade for Windows, Freeware, \url{http://samspade.org/ssw/}
address: Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-ku, Seoul, 137-070, Republic of Korea
country: KR
phone: +82-2-2186-4500
fax-no: +82-2-2186-4496
e-mail: hostmaster@nic.or.kr
nic-hdl: HM127-AP
mnt-by: MNT-KRNIC-AP
changed: hostmaster@nic.or.kr 20010514
source: APNIC

Trying 209.178.22.233 at ARIN
Trying 209.178.22 at ARIN
EarthLink Network, Inc. (NETBLK-EARTHLINK-NET) EARTHLINK-NET
   209.178.0.0 - 209.178.191.255
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0030) CBLPASLAN-USER0030
   209.178.22.8 - 209.178.22.15
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0031) CBLPASLAN-USER0031
   209.178.22.16 - 209.178.22.23
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0032) CBLPASLAN-USER0032
   209.178.22.24 - 209.178.22.31
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0033) CBLPASLAN-USER0033
   209.178.22.32 - 209.178.22.39
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0034) CBLPASLAN-USER0034
   209.178.22.40 - 209.178.22.47
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0035) CBLPASLAN-USER0035
   209.178.22.48 - 209.178.22.55
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0036) CBLPASLAN-USER0036
   209.178.22.56 - 209.178.22.63
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0037) CBLPASLAN-USER0037
   209.178.22.64 - 209.178.22.71
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0038) CBLPASLAN-USER0038
   209.178.22.72 - 209.178.22.79
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0039) CBLPASLAN-USER0039
   209.178.22.80 - 209.178.22.88
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0040) CBLPASLAN-USER0040
209.178.22.89 - 209.178.22.96
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0041) CBLPASLAN-USER0041
209.178.22.97 - 209.178.22.104
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0042) CBLPASLAN-USER0042
209.178.22.105 - 209.178.22.112
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0043) CBLPASLAN-USER0043
209.178.22.113 - 209.178.22.120
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0044) CBLPASLAN-USER0044
209.178.22.121 - 209.178.22.128
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0045) CBLPASLAN-USER0045
209.178.22.129 - 209.178.22.136
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0046) CBLPASLAN-USER0046
209.178.22.137 - 209.178.22.144
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0047) CBLPASLAN-USER0047
209.178.22.145 - 209.178.22.152
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0048) CBLPASLAN-USER0048
209.178.22.153 - 209.178.22.160
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0049) CBLPASLAN-USER0049
209.178.22.161 - 209.178.22.168
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0050) CBLPASLAN-USER0050
209.178.22.169 - 209.178.22.176
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0051) CBLPASLAN-USER0051
209.178.22.177 - 209.178.22.184
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0052) CBLPASLAN-USER0052
209.178.22.185 - 209.178.22.192
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0053) CBLPASLAN-USER0053
209.178.22.193 - 209.178.22.200
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0054) CBLPASLAN-USER0054
209.178.22.201 - 209.178.22.208
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0055) CBLPASLAN-USER0055
   209.178.22.209 - 209.178.22.216
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0056) CBLPASLAN-USER0056
   209.178.22.217 - 209.178.22.224
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0057) CBLPASLAN-USER0057
   209.178.22.225 - 209.178.22.232
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0058) CBLPASLAN-USER0058
   209.178.22.233 - 209.178.22.240
Charter Cable/Pasadena LAN (NETBLK-CBLPASLAN-USER0059) CBLPASLAN-USER0059
   209.178.22.241 - 209.178.22.248

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

**Trying 63.163.94 at ARIN**

Sprint (NETBLK-SPRN-BLKS) SPRN-BLKS   63.160.0.0 - 63.175.255.255
RAF/AMERICAN FRONTEER (NETBLK-FON-106767104042275) FON-106767104042275
   63.163.94.0 - 63.163.94.255

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

**Trying 210.52.214.15 at APNIC**
**Trying 210.52.214 at APNIC**
**Trying 210.52 at APNIC**

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois5.apnic.net)

inetnum:     210.52.0.0 - 210.52.0.63
netname:     BAODING-CABLE-TV
descr:       Baoding Cable TV Network
descr:       No.3 Shidai Road, Baoding
descr:       Hebei Province
country:     CN
admin-c:     ZM28-AP
tech-c:      ZM28-AP
mnt-by:      MAINT-CN-ZM28
changed:     zhaomq@china-netcom.com 20010716
source: APNIC

person: Zhao Mingqun
address: 9/F, Building A, Corporate Square, No. 35 Financial Street,
address: Xicheng District, Beijing 100032, P.R.China
country: CN
phone: +86-10-86011588
fax-no: +86-10-88091446
e-mail: zhaomq@china-netcom.com
nic-hdl: ZM28-AP
mnt-by: MAINT-CN-ZM28
changed: zhaomq@china-netcom.com 20010712
source: APNIC

Trying 216.40.195.72 at ARIN
Trying 216.40.195 at ARIN
Everyones Internet, Inc. (NETBLK-EVRY-BLK-6)
2600 Southwest Frwy Suite 500
Houston, TX 77098
US

Netname: EVRY-BLK-6
Netblock: 216.40.192.0 - 216.40.223.255
Maintainer: EVRY

Coordinator:
Williams, Randy (RW172-ARIN) admin@ev1.net
(713) 400-5400 x255

Domain System inverse mapping provided by:

NS1.EV1.NET   216.88.76.6
NS2.EV1.NET   216.88.77.7

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 07-Feb-2001.
Database last updated on 14-Jul-2001 23:02:13 EDT.

I did a top ten for each day of the week and then merged each daily top ten list into a single list to
see how many hosts showed up in the top on two or more days in the week and found the
following “Repeat Offenders”. These ten hosts account for twenty-two percent of the port scan
traffic for the week 04/10/2001 to 04/16/2001.
Table 31 - Repeat Offenders

<table>
<thead>
<tr>
<th>Packets</th>
<th>IP Address</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>10158</td>
<td>MY.NET.224.106</td>
<td>2</td>
</tr>
<tr>
<td>10043</td>
<td>MY.NET.220.66</td>
<td>2</td>
</tr>
<tr>
<td>6542</td>
<td>MY.NET.228.54</td>
<td>5</td>
</tr>
<tr>
<td>4143</td>
<td>MY.NET.217.230</td>
<td>3</td>
</tr>
<tr>
<td>3676</td>
<td>MY.NET.202.86</td>
<td>4</td>
</tr>
<tr>
<td>3361</td>
<td>MY.NET.219.222</td>
<td>3</td>
</tr>
<tr>
<td>2567</td>
<td>MY.NET.203.150</td>
<td>3</td>
</tr>
<tr>
<td>1981</td>
<td>MY.NET.209.218</td>
<td>2</td>
</tr>
<tr>
<td>518</td>
<td>MY.NET.211.114</td>
<td>2</td>
</tr>
</tbody>
</table>

3-3-5 Defensive Recommendations:

Configure your routers using RFC2267\(^{114}\), Network Ingress Filtering. Use ACL’s on your perimeter routers to restrict inbound access to ports 1-1023 whenever possible. Port scans are active reconnaissance. If you cannot block or restrict access, then employ Firewalls and Proxy servers when possible. You have already deployed perimeter IDS sensors, but you may want to (if you have not already done so) develop and use a host based IDS system.

Keep all systems patched and employ Tripwire on Unix/Solaris systems, IPChains on the latest Linux systems. Check your syslogs regularly.

Check out the top traffic generators, especially those that appear in the repeat offenders list.

3-4 Out-Of-Spec (OOS) Log Entries:

There are 905 Out-Of-Spec (OOS) Log entries. The top five MY.NET OOS talkers and external OOS talkers are in the table below. These ten systems account for 521 of the 905 OOS alerts logged.

Table 32 - Top Five MY.NET & External Out-Of-Spec Packet Generators (Talkers)

<table>
<thead>
<tr>
<th>Connections</th>
<th>MY.NET Host</th>
<th>Connections</th>
<th>External Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
<td>MY.NET.227.130</td>
<td>124</td>
<td>217.80.7.48</td>
</tr>
<tr>
<td>108</td>
<td>MY.NET.217.182</td>
<td>68</td>
<td>66.31.48.7</td>
</tr>
<tr>
<td>7</td>
<td>MY.NET.210.90</td>
<td>37</td>
<td>209.221.200.17</td>
</tr>
<tr>
<td>5</td>
<td>MY.NET.225.42</td>
<td>25</td>
<td>158.75.57.4</td>
</tr>
<tr>
<td>2</td>
<td>MY.NET.222.250</td>
<td>22</td>
<td>150.135.245.171</td>
</tr>
</tbody>
</table>

Whois lookups for each of the external hosts is provided here:

Trying 217.80.7.48 at RIPE
Trying 217.80.7 at RIPE
Trying 217.80 at RIPE
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 217.80.0.0 - 217.89.31.255
netname: DTAG-DIAL14
descr: Deutsche Telekom AG
country: DE
admin-c: RH2086-RIPE
tech-c: AH12705-RIPE
tech-c: ST5359-RIPE
status: ASSIGNED PA
remarks: ******************************************************
remarks: * ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK ATTACKS, *
remarks: * ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC. *
remarks: ******************************************************
notify: auftrag@nic.telekom.de
notify: dbd@nic.dtag.de
mnt-by: DTAG-NIC
changed: auftrag@nic.telekom.de 20010321
source: RIPE

route: 217.80.0.0/12
descr: Deutsche Telekom AG, Internet service provider
origin: AS3320
mnt-by: DTAG-RR
changed: rv@NIC.DTAG.DE 20001027
source: RIPE

% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.

person: Reinhard Hausdorf
address: Deutsche Telekom AG
address: Am Kavalleriesand 3
address: D-64295 Darmstadt
address: Germany
phone: +49
nic-hdl: RH2086-RIPE
notify: auftrag@nic.telekom.de
notify: dbd@nic.dtag.de
mnt-by: DTAG-NIC
changed: auftrag@nic.telekom.de 20010321
source: RIPE

person: Andreas Hengl
address: Deutsche Telekom AG
address: Internetplanung Nuernberg
address: Suedwestpark 26
address: 90449 Nuernberg
address: Germany
phone: +49 911
e-mail: ripe-contact.Darmstadt@telekom.de
nic-hdl: AH12705-RIPE
notify: auftrag@nic.telekom.de
notify: dbd@nic.dtag.de
mnt-by: DTAG-NIC
changed: auftrag@nic.telekom.de 20010528
source: RIPE

person: Security Team
address: Deutsche Telekom AG
address: Am Kavalleriesand 3
address: D-64295 Darmstadt
address: Germany
phone: +49
nic-hdl: RH2086-RIPE
notify: auftrag@nic.telekom.de
notify: dbd@nic.dtag.de
mnt-by: DTAG-NIC
changed: auftrag@nic.telekom.de 20010321
source: RIPE

Trying 66.31.48.7 at ARIN
Trying 66.31.48 at ARIN
ROADRUNNER-NORTHEAST (NETBLK-ROADRUNNER-NORTHEAST)
13241 Woodland Park Road
Herndon, VA 20171
US

Netname: ROADRUNNER-NORTHEAST
Netblock: 66.30.0.0 - 66.31.255.255
Maintainer: RRNE

Coordinator:
ServiceCo LLC (ZS30-ARIN) abuse@rr.com
1-703-345-3416

Domain System inverse mapping provided by:
DNS1.RR.COM 24.30.200.3
DNS2.RR.COM 24.30.201.3
DNS3.RR.COM 24.30.199.7
DNS4.RR.COM 65.24.0.172

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Database last updated on 14-Jul-2001 23:02:13 EDT.

Trying 209.221.200.17 at ARIN
Trying 209.221.200 at ARIN
Quantum Networking Solutions, Inc. (NETBLK-QNET-0)
1529 E Palmdale Blvd Ste 200
Palmdale, CA 93550
US
Netname: QNET-0  
Netblock: 209.221.192.0 - 209.221.223.255  
Maintainer: QNSI  

Coordinator:  
Linstruth, Chris (CL38-ARIN) cj1@QNET.COM  
+1-805-538-2028 (FAX) +1-805-538-2859  

Domain System inverse mapping provided by:  
NS2.QNET.COM  207.155.33.10  
NS1.QNET.COM  207.155.38.11  

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE  

Database last updated on 14-Jul-2001 23:02:13 EDT.  

Trying 158.75.57.4 at ARIN  
Trying 158.75.57 at ARIN  
POLIP (NET-TORUNPOLIP2)  
Computer Centre, Nicolaus Copernicus University  
ul. Chopina 12/18, 87-100 Torun, Poland  
PL  

Netname: TORUNPOLIP2  
Netblock: 158.75.0.0 - 158.75.255.255  

Coordinator:  
Szewczak, Zbigniew S. (ZSS-ARIN) zssz@TORUN.PL  
(56) 260-17 ext. 70  

Domain System inverse mapping provided by:  
ALFA.CS.TORUN.PL  158.75.10.75  
BILBO.NASK.ORG.PL  148.81.16.51  

Record last updated on 11-Oct-1995.  
Database last updated on 14-Jul-2001 23:02:13 EDT.  

Trying 150.135.245.171 at ARIN  
Trying 150.135.245 at ARIN  
University of Arizona (NET-UA-STU-NET)  
CCIT - Telecommunications  
Tucson, AZ 85721  
US
Netname: UA-STU-NET
Netblock: 150.135.0.0 - 150.135.255.255

Coordinator:
   De Young, Chris H (CD503-ARIN)  chd@ARIZONA.EDU
   (520) 626-3213 (FAX) (520) 621-9222

Domain System inverse mapping provided by:
   MAGGIE.TELCOM.ARIZONA.EDU 128.196.128.233
   NS1.ACES.COM   192.195.240.1
   UAZHE0.PHYSICS.ARIZONA.EDU 128.196.188.248
   NS1.SUNQUEST.COM  149.138.1.32

Record last updated on 23-Jul-1999.
Database last updated on 14-Jul-2001 23:02:13 EDT.

The following table also shows the Top Ten source and destination ports used. This shows us that the majority of the 905 OOS alerts were going to or coming from ports commonly used by Gnutella or Napster. That accounts for almost two-thirds of our OOS alerts. One frightening fact to note is that and additional 104 (approximately eleven percent) OOS alerts had a source port of zero. This accounts for seventy-seven percent of the OOS Alerts.

<table>
<thead>
<tr>
<th>Count SRC Port</th>
<th>Count DST Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>168</td>
<td>6346</td>
</tr>
<tr>
<td>104</td>
<td>0</td>
</tr>
<tr>
<td>68</td>
<td>706</td>
</tr>
<tr>
<td>38</td>
<td>18245</td>
</tr>
<tr>
<td>18</td>
<td>6688</td>
</tr>
<tr>
<td>14</td>
<td>2055</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>6699</td>
</tr>
<tr>
<td>4</td>
<td>1061</td>
</tr>
<tr>
<td>4</td>
<td>1107</td>
</tr>
</tbody>
</table>

125 of the OOS Packets from or to MY.NET.227.130 used port 6346 (Gnutella).

All OOS packets from MY.NET.217.182 were to or from port 6346 (Gnutella).

All of MY.NET.225.42 OOS packets were from or to port 6688 (Napster).

Gnutella/Napster related ports were not shown in the OOS packets for MY.NET.210.90 or MY.NET.222.250.

ALL OOS packets from 217.80.7 48 were to port 6346 on MY.NET.227.90 and were all transmitted between 08:30 and 09:01 on 04/12/2001.
All OOS packets from 66.31.48.7 originated from port 706 to sequential ports on
MY.NET.225.134.

All OOS packets from 209.221.200.17 were sent to one of two hosts (MY.NET.225.210 and
MY.NET.217.134).

All OOS Packets from 158.75.57.4 were sent to port 6346 or 6347 (Gnutella) on multiple
MY.NET systems.

ALL OOS Packets from 150.135.245.171 were sent to port 6346 on MY.NET.217.178 and were
all transmitted on 04/11/2001 between 17:35 and 17:37.

3-4-1 Gnutella/Napster (The MY.NET network Boom Box):

This section will actually cover the alerts produced by six of the top ten systems listed in Table 23
above.

Gnutella/Napster traffic accounts for 595 (the table above shows 593) OOS alerts. Sorry, but the
two alerts with a destination port of 6700 didn’t make the top ten. I will concentrate my analysis
on this area since it is in my opinion (and the numbers above support this) the loudest. Related
to the Gnutella/Napster ‘noise’ is the fact that a source port of ZERO was used 104 times. The
relation shows up when you check the destination ports on those 104 alerts and find that the
destination port is again Gnutella/Napster for 85 of the source port zero OOS alerts.

Comparing the OOS packets involved in this I found that 404 out of the 595 Gnutella/Napster
OOS that had a Gnutella destination port only also had a TTL between 43 and 53, a Type Of
Service of 0x0, both urgent flags and the SYN flag were set, the Don’t Fragment flag was set and
they had an ID of Zero. They also had the following additional contents in common (highlighted
in RED):

    TCP TTL:49 TOS:0x0 ID:0  DF
    21S***** Seq: 0x110FA1C1  Ack: 0x0  Win: 0x16D0
    TCP Options => MSS: 1460 SackOK TS: 408013240 0 EOL EOL EOL EOL

None of these ‘similar’ alerts had a source port of zero or one. The 595 Gnutella/Napster OOS
alerts had source or destination ports of 6346 (Gnutella), 6347 (Gnutella), or 6688/6699/6700
(Napster). Here is a list of the Top Five (Or all the talkers if less than five) Gnutella/Napster
talkers.

<table>
<thead>
<tr>
<th>Connections</th>
<th>MY.NET Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>MY.NET.227.130</td>
</tr>
<tr>
<td>108</td>
<td>MY.NET.217.182</td>
</tr>
</tbody>
</table>

Table 34 - Top Five Gnutella/Napster Talkers
Here is a diagram of all Source Port Zero traffic, it depicts all destination ports when the source port is zero. It includes both Internal and External Hosts using a Source Port of Zero. While it shows the number of connections to Source Port Zero. There are three tables below. Table 35 lists all outbound Source and Destination Port Combinations while the tables 36 and 37 depict all inbound source and destination port combinations. These charts cover Gnutella/Napster related traffic only which makes up two-thirds of the total number of Out-Of-Spec packets logged from 04/10/2001 to 04/16/2001.
According to the IANA Port Numbers web page\textsuperscript{115}, Port 0/TCP and 0/UDP is Reserved. All Port Zero OOS Log entries look like the examples below (NOTE: 87 have a TTL of 126 and 88 have one or more reserved flags set). All port Zero packets are TCP packets, they have a Type Of Service of 0x0, and the Don‘t Fragment flag is set.

\begin{verbatim}
04/10-00:48:20.500669 MY.NET.211.130:0 -> 209.11.34.136:1744
TCP TTL:126 TOs:0x0 ID:19224 DF
21*FR*** Seq: 0x50003C  Ack: 0xB167D9BA  Win: 0x5018
TCP Options => EOL EOL
04/10-01:09:09.617654 MY.NET.227.130:0 -> 64.230.75.39:1626
TCP TTL:126 TOs:0x0 ID:814 DF
*1SF*PAU Seq: 0x18CA0CD5  Ack: 0xC3F9026A  Win: 0x5010
04/10-01:12:10.578255 MY.NET.227.130:0 -> 130.113.48.61:6346
TCP TTL:126 TOs:0x0 ID:32119 DF
**SFRAU Seq: 0xB120CE0  Ack: 0x78440066  Win: 0x5010
04/10-02:52:30.354660 MY.NET.227.130:0 -> 132.177.66.198:1163
TCP TTL:126 TOs:0x0 ID:45649 DF
2*SFR*AU Seq: 0x18CA0D3A  Ack: 0x933F000A  Win: 0x8010
TCP Options => EOL EOL NOP NOP
04/10-03:30:32.771571 MY.NET.227.130:0 -> 211.132.49.100:6346
TCP TTL:126 TOs:0x0 ID:41484 DF
21S**** Seq: 0x4C60D5C  Ack: 0x933F000A  Win: 0x5018
38 C2 50 18 1F B5 D6 1D 00 00 34 5B 34 FC B5 79 8.P.......4[4..y
04/10-03:56:07.065768 MY.NET.227.130:0 -> 206.102.239.5:6346
TCP TTL:126 TOs:0x0 ID:11854 DF
21SF**AU Seq: 0x92C50D65  Ack: 0x40310338  Win: 0x5018
TCP Options => EOL EOL
04/10-04:00:33.595298 MY.NET.227.130:0 -> 65.5.197.86:6346
TCP TTL:126 TOs:0x0 ID:31171 DF
**SFRA* Seq: 0x136B0D7F  Ack: 0x82556EB5  Win: 0x5018
TCP Options => EOL EOL
04/10-04:09:53.161004 MY.NET.218.42:4432
TCP TTL:113 TOs:0x0 ID:41203 DF
*1SF*PA* Seq: 0x9F416B1  Ack: 0xE7F50097  Win: 0x5004
\end{verbatim}

\textsuperscript{115} IANA Port List, \url{http://www.iana.org/assignments/port-numbers}

What ports did the remainder of the Gnutella/Napster OOS alerts use? They were spread across the spectrum from port one on up. Here are three tables showing the various source/destination port combinations and the number of times each was used.

### Table 35 - Outgoing Source Port to Destination Port Combinations

<table>
<thead>
<tr>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6346</td>
<td>78</td>
<td>6346</td>
<td>16578</td>
<td>1</td>
<td>6346</td>
<td>2594</td>
<td>1</td>
<td>6346</td>
<td>4516</td>
<td>1</td>
</tr>
<tr>
<td>2370</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>1663</td>
<td>1</td>
<td>6346</td>
<td>2633</td>
<td>1</td>
<td>6346</td>
<td>4551</td>
<td>1</td>
</tr>
<tr>
<td>2954</td>
<td>6688</td>
<td>1</td>
<td>6346</td>
<td>1729</td>
<td>1</td>
<td>6346</td>
<td>2717</td>
<td>1</td>
<td>6346</td>
<td>4568</td>
<td>1</td>
</tr>
<tr>
<td>3685</td>
<td>6346</td>
<td>2</td>
<td>6346</td>
<td>17338</td>
<td>1</td>
<td>6346</td>
<td>3041</td>
<td>1</td>
<td>6346</td>
<td>4586</td>
<td>1</td>
</tr>
<tr>
<td>3812</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>18178</td>
<td>1</td>
<td>6346</td>
<td>3081</td>
<td>3</td>
<td>6346</td>
<td>46359</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1025</td>
<td>3</td>
<td>6346</td>
<td>1835</td>
<td>2</td>
<td>6346</td>
<td>3100</td>
<td>2</td>
<td>6346</td>
<td>4653</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1047</td>
<td>1</td>
<td>6346</td>
<td>1879</td>
<td>1</td>
<td>6346</td>
<td>32032</td>
<td>1</td>
<td>6346</td>
<td>4685</td>
<td>3</td>
</tr>
<tr>
<td>6346</td>
<td>1061</td>
<td>1</td>
<td>6346</td>
<td>19203</td>
<td>1</td>
<td>6346</td>
<td>3209</td>
<td>1</td>
<td>6346</td>
<td>4699</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1068</td>
<td>4</td>
<td>6346</td>
<td>1923</td>
<td>1</td>
<td>6346</td>
<td>3222</td>
<td>1</td>
<td>6346</td>
<td>4833</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>10943</td>
<td>1</td>
<td>6346</td>
<td>1961</td>
<td>1</td>
<td>6346</td>
<td>3226</td>
<td>1</td>
<td>6346</td>
<td>4870</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>10996</td>
<td>1</td>
<td>6346</td>
<td>20177</td>
<td>1</td>
<td>6346</td>
<td>3320</td>
<td>2</td>
<td>6346</td>
<td>49168</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1105</td>
<td>2</td>
<td>6346</td>
<td>2067</td>
<td>1</td>
<td>6346</td>
<td>3328</td>
<td>1</td>
<td>6346</td>
<td>49207</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1114</td>
<td>1</td>
<td>6346</td>
<td>2091</td>
<td>1</td>
<td>6346</td>
<td>33499</td>
<td>1</td>
<td>6346</td>
<td>4971</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1122</td>
<td>1</td>
<td>6346</td>
<td>21044</td>
<td>1</td>
<td>6346</td>
<td>3408</td>
<td>2</td>
<td>6346</td>
<td>49955</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>11418</td>
<td>1</td>
<td>6346</td>
<td>21069</td>
<td>1</td>
<td>6346</td>
<td>3449</td>
<td>2</td>
<td>6346</td>
<td>50244</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1142</td>
<td>2</td>
<td>6346</td>
<td>21245</td>
<td>1</td>
<td>6346</td>
<td>3488</td>
<td>2</td>
<td>6346</td>
<td>51446</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1147</td>
<td>1</td>
<td>6346</td>
<td>21415</td>
<td>1</td>
<td>6346</td>
<td>3493</td>
<td>2</td>
<td>6346</td>
<td>52512</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1208</td>
<td>1</td>
<td>6346</td>
<td>2142</td>
<td>1</td>
<td>6346</td>
<td>3513</td>
<td>1</td>
<td>6346</td>
<td>59481</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1213</td>
<td>1</td>
<td>6346</td>
<td>21813</td>
<td>1</td>
<td>6346</td>
<td>3527</td>
<td>1</td>
<td>6346</td>
<td>60506</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1232</td>
<td>1</td>
<td>6346</td>
<td>2186</td>
<td>1</td>
<td>6346</td>
<td>3655</td>
<td>1</td>
<td>6346</td>
<td>61538</td>
<td>4</td>
</tr>
<tr>
<td>6346</td>
<td>1240</td>
<td>1</td>
<td>6346</td>
<td>2200</td>
<td>2</td>
<td>6346</td>
<td>37390</td>
<td>1</td>
<td>6346</td>
<td>62351</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1259</td>
<td>1</td>
<td>6346</td>
<td>2258</td>
<td>2</td>
<td>6346</td>
<td>3768</td>
<td>4</td>
<td>6346</td>
<td>62568</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1263</td>
<td>2</td>
<td>6346</td>
<td>2279</td>
<td>1</td>
<td>6346</td>
<td>3789</td>
<td>1</td>
<td>6346</td>
<td>62783</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1266</td>
<td>1</td>
<td>6346</td>
<td>2292</td>
<td>1</td>
<td>6346</td>
<td>3909</td>
<td>1</td>
<td>6346</td>
<td>62875</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1309</td>
<td>2</td>
<td>6346</td>
<td>2297</td>
<td>1</td>
<td>6346</td>
<td>4030</td>
<td>1</td>
<td>6346</td>
<td>63633</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1382</td>
<td>1</td>
<td>6346</td>
<td>2372</td>
<td>1</td>
<td>6346</td>
<td>4032</td>
<td>1</td>
<td>6346</td>
<td>64181</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1448</td>
<td>1</td>
<td>6346</td>
<td>2372</td>
<td>1</td>
<td>6346</td>
<td>4154</td>
<td>1</td>
<td>6346</td>
<td>65308</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1485</td>
<td>1</td>
<td>6346</td>
<td>2388</td>
<td>3</td>
<td>6346</td>
<td>4170</td>
<td>1</td>
<td>6688</td>
<td>1528</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1505</td>
<td>1</td>
<td>6346</td>
<td>2429</td>
<td>1</td>
<td>6346</td>
<td>4197</td>
<td>1</td>
<td>6688</td>
<td>1595</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1517</td>
<td>1</td>
<td>6346</td>
<td>2442</td>
<td>1</td>
<td>6346</td>
<td>42133</td>
<td>1</td>
<td>6688</td>
<td>24581</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1533</td>
<td>2</td>
<td>6346</td>
<td>2492</td>
<td>2</td>
<td>6346</td>
<td>4316</td>
<td>1</td>
<td>6688</td>
<td>3694</td>
<td>1</td>
</tr>
<tr>
<td>SRC</td>
<td>DST Counts</td>
<td>SRC</td>
<td>DST Counts</td>
<td>SRC</td>
<td>DST Counts</td>
<td>SRC</td>
<td>DST Counts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
<td>-----</td>
<td>------------</td>
<td>-----</td>
<td>------------</td>
<td>-----</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>6346 2</td>
<td>1052</td>
<td>6699 1</td>
<td>11743</td>
<td>6347 1</td>
<td>1822</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>6688 2</td>
<td>1115</td>
<td>6699 1</td>
<td>1186</td>
<td>6347 1</td>
<td>1182</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>6699 1</td>
<td>1123</td>
<td>6699 1</td>
<td>1220</td>
<td>6346 1</td>
<td>1245</td>
<td>6699 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6346 1</td>
<td>1126</td>
<td>6346 2</td>
<td>1250</td>
<td>6346 1</td>
<td>1280</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6699 1</td>
<td>1156</td>
<td>6699 1</td>
<td>1295</td>
<td>6699 1</td>
<td>1295</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1052</td>
<td>6699 1</td>
<td>1165</td>
<td>6699 1</td>
<td>1304</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1115</td>
<td>6699 1</td>
<td>11743</td>
<td>6347 1</td>
<td>1320</td>
<td>6699 1</td>
<td>1323</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1123</td>
<td>6699 1</td>
<td>1186</td>
<td>6347 1</td>
<td>1348</td>
<td>6699 1</td>
<td>1349</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1126</td>
<td>6346 2</td>
<td>1220</td>
<td>6346 1</td>
<td>1355</td>
<td>6699 1</td>
<td>1355</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1156</td>
<td>6699 1</td>
<td>1245</td>
<td>6699 2</td>
<td>1358</td>
<td>6699 1</td>
<td>1358</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1165</td>
<td>6699 1</td>
<td>1280</td>
<td>6699 1</td>
<td>1419</td>
<td>6346 2</td>
<td>1419</td>
<td>6346 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11743</td>
<td>6347 1</td>
<td>1295</td>
<td>6699 1</td>
<td>1425</td>
<td>6699 1</td>
<td>1425</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1186</td>
<td>6347 1</td>
<td>1304</td>
<td>6699 1</td>
<td>1507</td>
<td>6347 1</td>
<td>1507</td>
<td>6347 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1220</td>
<td>6346 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1516</td>
<td>6346 2</td>
<td>1516</td>
<td>6346 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1245</td>
<td>6699 2</td>
<td>1304</td>
<td>6699 1</td>
<td>1517</td>
<td>6346 1</td>
<td>1517</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1280</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1524</td>
<td>6700 1</td>
<td>1524</td>
<td>6700 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1295</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1561</td>
<td>6699 1</td>
<td>1561</td>
<td>6699 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1304</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1565</td>
<td>6346 1</td>
<td>1565</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1320</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1574</td>
<td>6346 1</td>
<td>1574</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1355</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1580</td>
<td>6346 1</td>
<td>1580</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1358</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1599</td>
<td>6346 1</td>
<td>1599</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1419</td>
<td>6346 2</td>
<td>1307</td>
<td>6699 1</td>
<td>1697</td>
<td>6346 1</td>
<td>1697</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1425</td>
<td>6699 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1784</td>
<td>6688 1</td>
<td>1784</td>
<td>6688 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1507</td>
<td>6347 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1792</td>
<td>6688 1</td>
<td>1792</td>
<td>6688 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1516</td>
<td>6346 2</td>
<td>1307</td>
<td>6699 1</td>
<td>1792</td>
<td>6688 1</td>
<td>1792</td>
<td>6688 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1574</td>
<td>6346 1</td>
<td>1307</td>
<td>6699 1</td>
<td>1795</td>
<td>6346 1</td>
<td>1795</td>
<td>6346 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 36 - Incoming Source Port to Destination Port (Part 1 of 2)
### Table 37 - Incoming Source Port to Destination Port (Part 2 of 2)

<table>
<thead>
<tr>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
<th>SRC</th>
<th>DST</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>61702</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>2596</td>
<td>4</td>
<td>64432</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>2557</td>
<td>1</td>
</tr>
<tr>
<td>61870</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>2813</td>
<td>2</td>
<td>64459</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>2559</td>
<td>1</td>
</tr>
<tr>
<td>61917</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>2813</td>
<td>1</td>
<td>64481</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>1360</td>
<td>1</td>
</tr>
<tr>
<td>61942</td>
<td>6700</td>
<td>1</td>
<td>6346</td>
<td>3855</td>
<td>1</td>
<td>64488</td>
<td>6346</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61947</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>4453</td>
<td>1</td>
<td>64518</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>1586</td>
<td>1</td>
</tr>
<tr>
<td>61993</td>
<td>6346</td>
<td>1</td>
<td>6346</td>
<td>4515</td>
<td>1</td>
<td>64528</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>2285</td>
<td>1</td>
</tr>
<tr>
<td>62174</td>
<td>6346</td>
<td>1</td>
<td>63481</td>
<td>6346</td>
<td>1</td>
<td>64556</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3516</td>
<td>1</td>
</tr>
<tr>
<td>62205</td>
<td>6346</td>
<td>1</td>
<td>63494</td>
<td>6346</td>
<td>1</td>
<td>64624</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62300</td>
<td>6346</td>
<td>1</td>
<td>63525</td>
<td>6346</td>
<td>1</td>
<td>64637</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62383</td>
<td>6346</td>
<td>1</td>
<td>63573</td>
<td>6346</td>
<td>1</td>
<td>64658</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62491</td>
<td>6346</td>
<td>1</td>
<td>63608</td>
<td>6346</td>
<td>1</td>
<td>64735</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62523</td>
<td>6346</td>
<td>1</td>
<td>63674</td>
<td>6346</td>
<td>1</td>
<td>64744</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62584</td>
<td>6347</td>
<td>1</td>
<td>63703</td>
<td>6346</td>
<td>1</td>
<td>64745</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62639</td>
<td>6346</td>
<td>1</td>
<td>63737</td>
<td>6346</td>
<td>1</td>
<td>64760</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62653</td>
<td>6346</td>
<td>1</td>
<td>63767</td>
<td>6346</td>
<td>1</td>
<td>64767</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62730</td>
<td>6346</td>
<td>1</td>
<td>63783</td>
<td>6346</td>
<td>1</td>
<td>64794</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62777</td>
<td>6346</td>
<td>1</td>
<td>63786</td>
<td>6346</td>
<td>1</td>
<td>64800</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62799</td>
<td>6346</td>
<td>1</td>
<td>63801</td>
<td>6346</td>
<td>1</td>
<td>64804</td>
<td>6688</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62834</td>
<td>6346</td>
<td>1</td>
<td>63819</td>
<td>6346</td>
<td>1</td>
<td>64809</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62908</td>
<td>6346</td>
<td>1</td>
<td>63845</td>
<td>6346</td>
<td>1</td>
<td>64820</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62921</td>
<td>6346</td>
<td>1</td>
<td>63877</td>
<td>6346</td>
<td>1</td>
<td>64835</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62941</td>
<td>6346</td>
<td>1</td>
<td>63931</td>
<td>6346</td>
<td>1</td>
<td>64866</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>62964</td>
<td>6346</td>
<td>1</td>
<td>64006</td>
<td>6346</td>
<td>1</td>
<td>64881</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63074</td>
<td>6346</td>
<td>1</td>
<td>64031</td>
<td>6346</td>
<td>1</td>
<td>64908</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63105</td>
<td>6346</td>
<td>1</td>
<td>64105</td>
<td>6346</td>
<td>1</td>
<td>64921</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63137</td>
<td>6346</td>
<td>1</td>
<td>64118</td>
<td>6346</td>
<td>1</td>
<td>64929</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63166</td>
<td>6346</td>
<td>1</td>
<td>64137</td>
<td>6346</td>
<td>1</td>
<td>64938</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63180</td>
<td>6346</td>
<td>1</td>
<td>64167</td>
<td>6346</td>
<td>1</td>
<td>64945</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63193</td>
<td>6699</td>
<td>1</td>
<td>64189</td>
<td>6346</td>
<td>1</td>
<td>64958</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63234</td>
<td>6346</td>
<td>1</td>
<td>64205</td>
<td>6346</td>
<td>1</td>
<td>64966</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63320</td>
<td>6346</td>
<td>1</td>
<td>64214</td>
<td>6346</td>
<td>1</td>
<td>64982</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63400</td>
<td>6346</td>
<td>1</td>
<td>64247</td>
<td>6346</td>
<td>1</td>
<td>64989</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63435</td>
<td>6346</td>
<td>1</td>
<td>64252</td>
<td>6346</td>
<td>1</td>
<td>65005</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63446</td>
<td>6346</td>
<td>1</td>
<td>64290</td>
<td>6346</td>
<td>1</td>
<td>65047</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>63457</td>
<td>6346</td>
<td>1</td>
<td>64307</td>
<td>6346</td>
<td>1</td>
<td>65054</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1188</td>
<td>1</td>
<td>64338</td>
<td>6346</td>
<td>1</td>
<td>65092</td>
<td>6346</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1195</td>
<td>1</td>
<td>64374</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>1712</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1522</td>
<td>2</td>
<td>64393</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>1763</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>1746</td>
<td>2</td>
<td>64399</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>2552</td>
<td>1</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>2244</td>
<td>1</td>
<td>64411</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>2552</td>
<td>2</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>2417</td>
<td>2</td>
<td>64421</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>2554</td>
<td>6</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
<tr>
<td>6346</td>
<td>2499</td>
<td>1</td>
<td>64427</td>
<td>6346</td>
<td>1</td>
<td>6688</td>
<td>2557</td>
<td>2</td>
<td>6699</td>
<td>3586</td>
<td>1</td>
</tr>
</tbody>
</table>

3-4-2 MY.NET.210.90
This system is sending data from Port zero, and all the packets have un-natural flag settings. The Portscan log also show incoming SYN to port 53 on 04/10/2001 and has two outgoing NULL Scans that are not shown in the OOS alerts. None of the ports used are known Trojan ports and the system may not be compromised. The user on this system may be doing things he is not supposed to, or someone else wants us to think this user is doing things he is not supposed to.

Here are the OOS Log and Portscan Log entries:

```
TCP TTL:126 TOS:0x0 ID:12946 DF
2*SFR**U Seq: 0xA1430016 Ack: 0x9CF4037F Win: 0x5010
 2*SFR**U Seq: 0xA1430016 Ack: 0x9CF4037F Win: 0x5010
04/10-19:20:05.257885 MY.NET.210.90:1366 -> 129.32.112.160:41003
TCP TTL:126 TOS:0x0 ID:58036 DF
*1SF**** Seq: 0x1C Ack: 0xED5600FD Win: 0x5010
 33 83 50 10 41 44 60 AB 20 20 20 20 00 3.P.AD`
04/10-19:21:03.097680 MY.NET.210.90:1366 -> 129.32.112.160:41003
TCP TTL:126 TOS:0x0 ID:52921 DF
2*SFRPAU Seq: 0x1C Ack: 0xED560113 Win: 0x5010
05 56 A0 2B 00 00 00 1C ED 56 01 13 08 7F 50 10 .V.+...V....P.
 80 00 4C DD 20 20 20 20 00 ..L.         
TCP TTL:126 TOS:0x0 ID:2751 DF
*1SFRPAU Seq: 0x1C Ack: 0xED56012D Win: 0x5010
 05 56 A0 2B 00 00 00 1C ED 56 01 13 08 7F 50 10 .V.+...V....P.
 80 00 4C DD 20 20 20 20 00 ..L.         
04/10-19:21:44.963471 MY.NET.210.90:0 -> 129.32.112.160:1366
TCP TTL:126 TOS:0x0 ID:21446 DF
**SF*PA* Seq: 0xA02B001C Ack: 0xED560152 Win: 0x5010
 05 56 A0 2B 00 00 00 1C ED 56 01 13 08 7F 50 10 .V.+...V....P.
 80 00 4C DD 20 20 20 20 00 ..L.         
04/11-17:55:35.022868 MY.NET.210.90:1608 -> 134.126.217.97:41069
TCP TTL:126 TOS:0x0 ID:18114 DF
21*F*P** Seq: 0x437 Ack: 0xF04701D5 Win: 0x5010
 05 56 A0 2B 00 00 00 1C ED 56 01 13 08 7F 50 10 .V.+...V....P.
 80 00 4C DD 20 20 20 20 00 ..L.         
04/11-17:59:01.447424 MY.NET.210.90:0 -> 134.126.217.97:1608
TCP TTL:126 TOS:0x0 ID:9193 DF
21*F**AU Seq: 0xA06D0437 Ack: 0xF047024B Win: 0x8010
TCP Options => EOL EOL NOP NOP Sack: 587@51621 EOL EOL EOL EOL
EOL EOL EOL EOL EOL EOL EOL
```

Checking Portscan Log for [MY.NET.210.90]'s data!
The first packet is to an SSL Port, Encrypted data is present. We have the reserved flags set on both OOS alerts. The second packet is from port 240 (A reserved port) to Port 1092 (not a known Trojan port), but again the flags almost look like a christmas tree. Each packet occurred on different days. He received two FTP scans and a DNS scan this week as well. Finally, he sent another packet with a reserved flag set to 209.10.169.37 on 04/15/2001 at 22:38.

Globix Corporation (NETBLK-GLOBIXBLK3)
295 Lafayette St- 3rd Fl
NY, NY 10012
US

Netname: GLOBIXBLK3
Netblock: 209.10.0.0 - 209.11.223.255
Maintainer: PFMC

The 209.10.169.58 address is registered as members.blackplanet.com while the 209.10.169.37 address is unregistered (unregistered.blackplanet.com).

Packet corruption is a very good possibility.

The first packet is to an SSL Port, Encrypted data is present. We have the reserved flags set on both OOS alerts. The second packet is from port 240 (A reserved port) to Port 1092 (not a known Trojan port), but again the flags almost look like a christmas tree. Each packet occurred on different days. He received two FTP scans and a DNS scan this week as well. Finally, he sent another packet with a reserved flag set to 209.10.169.37 on 04/15/2001 at 22:38.

Globix Corporation (NETBLK-GLOBIXBLK3)
295 Lafayette St- 3rd Fl
NY, NY 10012
US

Netname: GLOBIXBLK3
Netblock: 209.10.0.0 - 209.11.223.255
Maintainer: PFMC

The 209.10.169.58 address is registered as members.blackplanet.com while the 209.10.169.37 address is unregistered (unregistered.blackplanet.com).

Packet corruption is a very good possibility.
Checking Portscan Log for [MY.NET.222.250]'s data!

Apr 10 05:33:01 210.220.73.117:4253 -> MY.NET.222.250:21 SYN **S*****
Apr 12 05:39:52 24.165.162.34:2135 -> MY.NET.222.250:21 SYN **S*****
RESERVEDBITS

3-4-4 External Host 66.31.48.7

The alert log entries alone for this host gives me every reason to suspect he is up to no good. The pattern shows a Queso Fingerprint alert followed no later than fourteen to sixteen minutes by an spp. Portscan. All Queso Fingerprint alerts originated from port 706 to port 17989 or higher, had both reserved flags and the SYN flag set. There are twenty-seven entries like this on 04/16/2001 and one on 04/10/2001 being transmitted to MY.NET.225.134. There are sixty-eight OOS packets logged. Except for the TOS: 0x8, these packets have a similar pattern to the one I showed you earlier when examining the Gnutella/Napster traffic. The source port never changes but the destination port is constantly changing and it is directed at a single system on the MY.NET network. This is an indication of active targeting. Take a close look at the destination machine to ensure all patches are in place and if this guy has not broken in yet, then you might want to install a Host based IDS and do some extensive packet logging to attempt to gain information on the type of tools being used the exploits. Only the OOS packets are available and they for all intents and purposes all look the same.

Example Port Scan Log Entry (one of 28 for this report period). Except for the destination port and time stamps, they all looked like this:

Apr 16 05:15:40 66.31.48.7:706 -> MY.NET.225.134:1798 SYN 21S***** RESERVEDBITS

Example OOS Alert logged. Again except for the time stamp, destination port and ACK Number, they all looked like this:

04/10-09:14:56.866168 66.31.48.7:706 -> MY.NET.219.134:2504
TCP TTL:44 TOS:0x8 ID:0  DF
21S***** Seq: 0xE8922086  Ack: 0x0  Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 96560875 0 EOL EOL EOL EOL
04/10-09:19:01.889147 66.31.48.7:706 -> MY.NET.219.134:2545
TCP TTL:44 TOS:0x8 ID:0  DF
21S***** Seq: 0xF84349A7  Ack: 0x0  Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 96585435 0 EOL EOL EOL EOL
TCP TTL:44 TOS:0x8 ID:0  DF
21S***** Seq: 0x26288D38   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 110082814 0 EOL EOL EOL EOL
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+3-4-5 External Host 209.221.200.17

This host generated thirty-seven Out-Of-Spec alerts. Fourteen of them were directed against MY.NET.225.134, the remaining twenty-three were directed at MY.NET.225.210.

This is not the first time that MY.NET.217.134 has appeared. This is a rare case where someone has tried to break in to this machine however. A quick check of the logs gives us 190 alert log entries (all ports), fourteen OOS alerts (all inbound), and 2168 Portscan alerts (only twenty of which are inbound). Most of the port scan traffic was explained as ‘Game’ traffic earlier in this report. Check out MY.NET.217.134 the game traffic seems to have made it an active target. If the host is not compromised then I recommend you take advantage of the situation and use a host based IDS to do some information gathering of your own.

MY.NET.225.210 is another gamer. The OOS traffic is not game related however, it looks like a breakin attempt and a pretty serious one at that. Check out the machine for possible compromise, the gaming has advertised this host as a possible easy target. Again, if the host is not compromised then I recommend you take advantage of the situation and use a host based IDS to do some information gathering of your own.

###############################
Checking Out-Of-Spec Logs for [209.221.200.17]'s data!
37
04/12-14:38:41.880865 209.221.200.17:1634 -> MY.NET.217.134:55482
04/12-14:44:32.297474 209.221.200.17:1645 -> MY.NET.217.134:2422
04/12-14:46:40.010715 209.221.200.17:1652 -> MY.NET.217.134:50986
04/12-14:51:36.428197 209.221.200.17:1660 -> MY.NET.217.134:14315
04/13-12:02:44.759755 209.221.200.17:19 -> MY.NET.217.134:1256
04/16-11:25:03.158244 209.221.200.17:1061 -> MY.NET.225.210:20
04/16-11:30:07.142412 209.221.200.17:166 -> MY.NET.225.210:1091

3-4-6 Defensive Recommendations:

Your best defense is to keep your systems patched and to control access to them as best you can using Router Access Control Lists, Firewalls and Proxies.
3-5 Summary

There is a large amount of Gnutella/Napster activity on the MY.NET network. Add to this an almost equal amount of internet gaming and you have just covered almost seventy-five percent of the traffic on the MY.NET network. Every attempt to reduce if not eliminate this type of activity should be made. Blocking the ports with Firewalls or Routers will not work since most of these applications can be configured to use different ports.

The MY.NET network was scanned 193,148 times in the past week. Print Spooler, Remote Procedure Call, Domain Name Service Server, and FTP servers were the main items being scanned for. These services being scanned for are no different than any other network being scanned. Steps to ensure that access to these services is tightly controlled using Firewalls or Router Access Control Lists is recommended.

Out-Of-Spec packets will always be a problem. Approximately two-thirds of the Out-Of-Spec packets generated were Gnutella/Napster related. Of major concern was the extremely large number (I would call ten percent large) of port zero to port zero packets. These ports are generally used by Routers to transmit routing information to each other and cannot be blocked by firewalls or router ACL’s. Couple this with the fact that almost all of the port zero to port zero activity was also related to Gnutella/Napster and you have a very serious problem on your hands. Gnutella is a file sharing application and is a great way to spread Trojans and Worms. Every attempt to reduce if not eliminate this type of activity should be made.

Alerts are indications of possible hostile activity. I recommend that the current Snort Rule set be re-evaluated and compared to the current Snort Rule sets available from Whitehats.com and Snort.org. The new rule sets are not as general as some of the rules I see in use here.

There are definite indications of misconfigured NATs and Routers. This is an indication of address spoofing. Sixty percent of these spoofed packets were to private addresses and a large portion were for 164.254.0.0 which is reserved for auto-configuration of local addresses where no DHCP server is found. You should not be seeing private network addresses and data packets with source addresses from outside your network originating from the MY.NET network. Configuring your Routers IAW RFC 2267, Network Ingress Filtering will stop these types of packets from leaving your network and should eliminate these types of packets.

3-5-1 Possible/Probable Compromised Systems:

MY.NET.219.34 responded to stimulus of port 32771 (External RPC/SUNRPC High Port Access). Although this host is listed as possible compromise, you will probably find that the gaming activity of the user on this system is what triggered the alerts.

MY.NET.134.55 is probably infected with the Network.VBS Rule (SMB Wildcard Access)

Eleven of twenty four hosts responded to a stimulus of port 27374 (Possible Trojan Server Activity). A response to a stimulus of this port is an indication of Trojan activity on these systems.
systems. They are: MY.NET.202.34, MY.NET.204.142, MUY.NET.100.82, MY.NET 146.51, MY.NET.215.34, MY.NET.222.226, MY.NET.222.50, MY.NET.229.54, MY.NET.204.214, MY.NET.60.152 AND MY.NET.97.147.

MY.NET.178.42 (Russia Dynamo) is producing a lot of traffic at odd hours and should be investigated.

3-5-2 Defensive Recommendations:

The following defensive measures will go a long way in reducing the amount of hostile activity you are seeing on the MY.NET network:

- Control or block access to all critical services (Ports 1 thru 1023) using Firewalls or Router ACLs.
- Configure all Routers IAW guidance contained in RFC 2267 (Network Ingress Filtering).
- Check all hosts and remove all unneeded services.
- Apply all patches to all critical systems immediately.
- Apply patches recommended in all CERT Bulletins to all operating systems.
- Require the use of Anti-Virus software and enforce its use.
- Purchase Trojan Scanner software and use it routinely to scan critical systems.
- Take steps to reduce and/or eliminate the use of Gnutella/Napster.
- Update the Snort Rule set on your snort sensors.
3-6 Analysis Process and Tools Used:

Data Collection

I retrieved the data from http://www.research.umbc.edu/~andy as directed in the assignment guidelines. I also downloaded all the GIAC practicals (10 thru 353). There was four months of data on the download site, I choose one weeks worth of data. The data files I used are listed at the beginning of Section three of this practical.

Tools

Copies of scripts, batch files and special configuration files are provided in Appendix C. Sources and authors are also contained in each script along with modifications I made if any.

A list of the software I used is in Appendix D.

Data Separation

After selecting the files to be analyzed I began by combining all daily files into one large weekly file. I modified the Perl Scripts obtained from Andrew Baker and Michael Bell (the modifications I made are annotated in the scripts listed in Appendix C). From there I extracted data using Perl Scripts or with Grep and eGrep. The extracted data was then pasted into an Excel Spreadsheet for manipulation or captured into an open file in ConText/Programmers File Editor (PFE). ConText/PFE monitor the file on disk for modifications and provide an alert with an option to reload the modified file from disk when this happens. A batch file is listed in Appendix C that allows me to search Alert, Portscan and OOS logs by host.

Data Manipulation

In some cases I just opened the IDS files directly with ConText to do searches, the highlighter configuration was used to provided emphasis on each field of the Snort alerts. Viewing the OOS alerts was a little easier with the fields highlighted.

In some cases I used Microsoft Excel formulas to dissect each alert entry into data, time, source IP, source port, destination IP and destination port. The Perl Scripts I used were modified to output the results in Comma Separated Variable (CSV) format which I could then open in Excel and manipulate. Once in Excel I manipulated/sorted the data to view and analyze it the way I wanted to.

Alert Analysis

Alert descriptions were generated with help from sources from the web, other GCIA Practicals, published works. All sources are listed as footnotes throughout this practical.

A chart depicting the number of alerts per day for each day of the report period was displayed at
the beginning of each Alert (some alerts were combined because of their similarity). All charts and tables were produced using Microsoft Excel 2000.

a. I merged all daily files into one large IDS file in chronological order.
b. I used the anl_ids.pl perl script to get a count of each alert from the merged ids file.
c. I used the anl_ids.pl perl script to get the number of each alert from each daily ids file.
d. These reports were opened in Excel and merged to get the chart and table shown at the beginning of the Alert section of this practical.
e. I grep’d each alert from the large file into separate alert ids files.
f. I used the top_talkers.pl script to get a list of top talkers for each alert and for the merged alert ids file.
g. Each top talkers list was opened in Excel.
h. Each single alert file was opened in Excel and formulas split each alert into Date, Time, Source IP, Source Port, Destination IP, Destination Port.

The G.BAT file was used to extract information on individual systems from the alerts, portscan, and Out-Of-Spec log files. Sam Spade was used to provide Whois information.

Portscan Analysis

For each daily alert file I used the snort_source.pl perl script to generate a list of top talkers.

I merged all daily files into one large portscan file.

a. I used snort_source.pl to generate a list of top talkers for the report period.
b. Each daily top talkers list was opened in Excel.
c. I merged all top talkers lists (daily and weekly) into one list and generated a repeat offenders list and a Top Five External scanners list.
d. I loaded each daily file into Excel and used formulas to extract the scan types, sorted the list and counted each scan type. I repeated this for each daily file and then merged all daily files into the single table in the port scan section.
e. I used grep and the merged file to verify the count of each particular scan type.
f. I used grep to extract all UDP entries and to get a count of the game ports to show the percentage of the weekly total of UDP traffic is generated by what appear to be gamers.
g. I used Sam Spade to do a whois lookup on each of the Top Five External scanners.

The G.BAT file was used to extract information on individual systems from the alerts, portscan, and Out-Of-Spec log files. Sam Spade for Windows was used to provide Whois information.

Out-Of-Spec Analysis

To generate the Top Five MY.NET and External OOS alert generators table I used the oos_TopSourceAddress.pl Perl script to extract the top source and destination address pairs. I then modified this Perl Script and extracted the top source and destination port pairs. The modified Perl script was saved as oos_TopSourcePorts.pl. Each script is listed individually in
Appendix C.

To generate the Top Ten Source and Destination Ports table I used the oos_TopTalkersAddress.pl Perl script to extract a list of the top talkers. I again modified this script and extracted a list of ports used. I saved this script as oos_TopTalkersPorts.pl.

The Top Ten Ports table used indicated a high volume of Gnutella/Napster traffic and the use of Port Zero. I used Grep to count the number of Gnutella/Napster related packets which showed me that Sixty-Percent of the Out-Of-Spec traffic had a Gnutella/Napster source or destination port. Eighty-two percent of the Port Zero packets also had a Gnutella/Napster source or destination port as well.

I used Visio to graph the port Zero connections and Excel to provide tables with all Gnutella/Napster source and destination port combinations used during this evaluation period.

Finally, I finished the evaluation of Out-Of-Spec packets by analyzing the traffic produced by the remaining hosts in the top five talkers category that were not connected to the Gnutella/Napster or Port Zero evaluations.

The G.BAT file was used to extract information on individual systems from the alerts, portscan, and Out-Of-Spec log files. Sam Spade for Windows was used to provide Whois information.

3-7 Published References:


Cooper, Fearnow, Frederick and Northcutt “Intrusion Signatures and Analysis”. Reading: New Riders Publishing 2001


Stevens, W. Richard “TCP/IP Illustrated, Volume 1”. Reading: Addison Wesley 1994
Appendix A
Description of Log Fields

Log formats shown here:

SNORT Alert Log Entry
SNORT Portscan Log Entry
SNORT Out-OF-Spce Log Entry
Shadow Alert Log Entry

SNORT Alert Log Entry:


Intrusion Detection Signature: [**] Null scan! [**]
This Intrusion Detection Signature is a Snort standard of reference.

Date and time:  MM/DD-hh:mm:ss.XXXXXseconds
Source IP address and Source Port:  213.245.17.202:1311
Direction of packet travel:  ->
Indicates direction of packet travel between hosts.
Destination IP address and Destination port:  198.192.223.198:4036

SNORT Portscan Log Entry:

Apr 15 00:10:37 198.192.206.150:2649 -> 200.253.203.246:6346 UDP

Date and time:  MMM DD hh:mm:ss
Source IP address and Source Port:  198.192.206.150:2649
Direction of packet travel:  ->
Indicates direction of packet travel between hosts.
Destination IP address and Destination port:  200.253.203.246:6346
Protocol or Comments:  UDP
The Protocol may be replaced by additional comments such as “NOACK 2**FR*** RESERVEDBITS”, “SYN **S*****”, “SYN 21S***** RESERVEDBITS” to name a

Northcutt, Cooper, Fearnlow and Frederick “Intrusion Signatures and Analysis”. Reading: New Riders Publishing 2001
few.

**SNORT Out Of Spec Log Entry:**

04/14-02:10:23.793710 216.182.20.130:1086 -> 198.192.223.198:4036
TCP TTL:112 TOS:0x0 ID:23059 DF
21S***AU Seq: 0x103E860   Ack: 0xB439AF83   Win: 0x5018
04 3E 0F C4 01 03 E8 60 B4 39 AF 83 00 F2 50 18   .>.....9....P.
D4 B3 CC 04 00 00 47 45 54 20 68 74 74 70 3A 2F   ......GET http:/
2F 77   /w

**Date and time:** MM/DD-hh:mm:ss.XXXXXseconds

**Source IP address and Source Port:** 216.182.20.130:1086

**Direction of packet travel:** ->
Indicates direction of packet travel between hosts.

**Destination IP address and Destination port:** 198.192.223.198:4036

**Protocol or Comments:** TCP

**Time to Live:** TTL:1 12
A field used to prevent packets from traversing the Internet forever. This field is reduced by 1 as it passes through each router. When the packet reaches 0, an ICMP time exceeded during transit is sent to the originating host. (Stevens\(^{117}\), Chapter 13)

**Type of Service:** TOS:0x0
Used to characterize how this IP packet should be handled as to throughput, reliability, etc. (Stevens\(^{118}\), Chapter 3)

**IP Identification number:** ID: 23059
An incrementing value used to identify a datagram.

**Don’t Fragment:** DF
Explicit declaration that this packet is not to be fragmented. If this packet crosses a network that has a maximum packet size smaller than the packet size, then an ICMP Unreachable, fragmentation required and DF set is sent to the originating host. (Stevens\(^{119}\), Chapter 11)

**TCP Flags:** 21S***AU

---


\(^{118}\) Stevens, W. Richard “TCP/IP Illustrated, Volume 1”, Chapter 3. Reading: Addison Wesley 1994

There are 8 bits for flags (of these the first two are reserved). The valid flags are URG, ACK, PSH, RST, SYN, and FIN.

TCP Sequence Number: Seq: 0x103E860
Agreed upon during the TCP three-way handshake and used to help ensure reliable transport.

TCP Acknowledge Number: Ack: 0xB439AF83
Next sequence byte count expected from the session partner.

Window size: Win: 0x5018

Shadow Alert Entry:

02:48:29.447596 my.net.6.5.80 > 216.177.16.64:1941: FP 410966647:410967571(924) ack
1267379385 win 32768 (DF)

Date and time: MM/DD-hh:mm:ss.XXXXXseconds

Source IP address and Source Port: my.net.6.5:80

Direction of packet travel: ->
Indicates direction of packet travel between hosts.

Destination IP address and Destination Port: 216.177.16.64:1941

TCP Flags: FP
There are 8 bits for flags (of these the first two are reserved). The valid flags are URG, ACK, PSH, RST, SYN, and FIN.

TCP Sequence Numbers: 410966647
Agreed upon during the TCP three-way handshake and used to help ensure reliable transport.

TCP Acknowledge Number: 410967571 ack
Next sequence byte count expected from the session partner.

Number Data Bytes Transmitted: (924)
This is the number of bytes of data in the packet.

IP Identification number: 1267379385
An incrementing value used to identify a datagram.

Window size: Win: 32768
Don’t Fragment: (DF)
Explicit declaration that this packet is not to be fragmented. If this packet crosses a
network that has a maximum packet size smaller than the packet size, then an ICMP
Unreachable, fragmentation required and DF set is sent to the originating host.
(Stevens\textsuperscript{120}, Chapter 11)

Netscape Enterprise Server Log Entry

"GET /cgi-bin/pub_affairs/article5.pl?file_dir=05May2001 HTTP/1.0" 200 6764

Source IP: 64.sun5.dialup.G$.NET

Date and Time Stamp: [05/Jul/2001:02:48:06 -0500]
DD/month/Year:hh:mm:ss.GMT Offset

URL and HTTP Protocol Version\textsuperscript{121}: GET /cgi-bin/pub_affairs/article5.pl?file_dir=05May2001
HTTP/1.0

HTTP Result code\textsuperscript{122}: 200

Number of Bytes transmitted: 6764

\textsuperscript{120} Stevens, W. Richard “TCP/IP Illustrated, Volume 1”, Chapter 11. Reading: Addison Wesley 1994
\textsuperscript{121} RFC1945, HTTP/1.0, \url{http://www.rfc-editor.org/rfc/rfc1945.txt}
\textsuperscript{122} RFC2616, HTTP/1.1, \url{http://www.rfc-editor.org/rfc/rfc2616.txt}
## Appendix B

### Severity Evaluation Criteria

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

#### B-1 Criticality:

<table>
<thead>
<tr>
<th>Criticality</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five (5)</td>
<td>Firewall, DNS Server, Core Router</td>
</tr>
<tr>
<td>Four (4)</td>
<td>E-mail relay/Exchanger, Database servers</td>
</tr>
<tr>
<td>Two (2)</td>
<td>Unix Desktop systems.</td>
</tr>
<tr>
<td>One (1)</td>
<td>Windows Desktop systems.</td>
</tr>
<tr>
<td>Zero (0)</td>
<td>Network printers and scanners.</td>
</tr>
</tbody>
</table>

#### B-2 Lethality:

<table>
<thead>
<tr>
<th>Lethality</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five (5)</td>
<td>Can gain root/administrator access across over the network.</td>
</tr>
<tr>
<td>Four (4)</td>
<td>Lockout by Denial Of Service.</td>
</tr>
<tr>
<td>Three (3)</td>
<td>User Access.</td>
</tr>
<tr>
<td>Two (2)</td>
<td>Confidentiality attack.</td>
</tr>
<tr>
<td>One (1)</td>
<td>Attack not likely to succeed.</td>
</tr>
</tbody>
</table>

#### B-3 System Countermeasures:

<table>
<thead>
<tr>
<th>Countermeasures</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five (5)</td>
<td>Modern Operating System (OS), all patches, and added security (TCP Wrappers or Personal Firewall).</td>
</tr>
<tr>
<td>Four (4)</td>
<td>Modern Operating System (OS), minimum patches, added security.</td>
</tr>
<tr>
<td>Three (3)</td>
<td>Older Operating System, some patches, added security.</td>
</tr>
<tr>
<td>Two (2)</td>
<td>Older Operating System, some patches, no added security.</td>
</tr>
<tr>
<td>One (1)</td>
<td>No added security, no patches, allows fixed passwords.</td>
</tr>
</tbody>
</table>

#### B-4 Network Countermeasures:

<table>
<thead>
<tr>
<th>Countermeasures</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five (5)</td>
<td>Validated restrictive firewall, one way in or out.</td>
</tr>
<tr>
<td>Four (4)</td>
<td>Restrictive firewall and some external connections (Dial-ups).</td>
</tr>
<tr>
<td>Three (3)</td>
<td>Permissive firewall (was the attack allowed through?)</td>
</tr>
<tr>
<td>Two (2)</td>
<td>IDS System (was the attack detected).</td>
</tr>
</tbody>
</table>

---

Appendix C
Scripts and Config Files

The scan logs were analyzed using Perl scripts. Some of them were borrowed from Mike Bell’s\(^{124}\) GCIA Practical. They were modified to meet my needs for this practical and to run in a Windows environment.

Snort-sort.pl
Anl_ids.pl
Top_talkers.pl
G.bat
Snort_source.pl
Oos_TopSourceAddress.pl
Oos_TopSourcePorts.pl
Oos_TopTalkersAddress.pl
Oos_TopTalkersPorts.pl
Snort.chl

C-1 **Snort-sort.pl**

```perl
#!/perl
#
# Filename: snort_sort.pl
# Author:   Andrew R. Baker <andrewb@uab.edu>
# Modified: 2000.03.17
# Purpose:  this script produces a sorted list of snort alerts
# from a snort alert file
# Version:  0.03
#
# let me know if you like this and use it -Andrew
#
# Todo: 1) Allow processing of snort alerts from syslog
#       2) Make html output optional
#       3) add specialized processing for portscan alerts
#       4) Make a multi-page hierarchy (not suitable for realtime)
#
# Change History:
# 2000.03.17 handle the new format of "-A fast" alerts
#
# 2000.03.16 changes to process spp_portscan alerts.
#       these need to be rewritten
#
# 2000.03.07 reverse DNS lookup
```

\(^{124}\) Bell, Mike GCIA Practical, SANS. [http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc](http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc)
# derived from snort_stat.pl
# and code donated by Adam Olson <adamo@humbolt1.com>
# cgi link option
# derived from code donated by Adam Olson <adamo@humbolt1.com>
#
# 2000.03.06 Original script
#
# Options:
# -r do reverse DNS lookups (this can slow things down)
# -h produce html output (hardwired)
# -w include cgi links based on IP addresses
#       (implies -h)
# -p include spp_portscan data (uses a special format)
#
use Getopt::Std;
use Socket;
%HOSTS = {};  #hash table for reverse DNS
#$ARGV[0] = "alert.ids";

if($ARGV[0] eq undef)
{
    print STDERR "USAGE: snort-sort <filename>\n";
    exit;
}

getopts('rhwp');
$opt_h = 1;
if($opt_w) {
    $opt_h = 1;
}

# set the cgi query href, you can change this to anything you want
# it gets expanded to "<a href=$cgi_href$ipaddr>$host</a>" in the output.
$cgi_href = "http://www.arin.net/cgi-bin/whois.pl?queryinput=";

open(INFILE,"< $ARGV[0]") || die "Unable to open file $ARGV[0]n";

if($opt_h) {
    print "<html>\n";
    print "<head>\n";
    print "<title>Sorted Snort Alerts</title>\n";
    print "</head>\n";
    print "<body>\n";

# HAL - Centered a few things, added a line to say what file was used in case the default is not.
# HAL - Also added comment about links at the end of the report.

print "<CENTER><h1>Sorted Snort Alerts</h1>File used: $ARGV[0]<BR>Additional References and information at end of page.<hr/></CENTER>

} else {
    #plain old text output goes here
}

while(<INFILE>) {
    chomp;
    # if the line is blank, go to the next one
    if ( $line eq """) { next }

    # we now have multiple formats for the log traffic
    # is this a "new" style fast alert
    if($line =~ /^\[\*\*\]/) {
        # split the alert apart
        ($datentime,$alert,$message) = split(/\[\*\*\]/,"$line");
        $alert =~ s/^\[\*\*\]/\s/; /g;
        $alert =~ s/$/\s/; /g;
        $a = "$datentime $message"
    } elsif ($line =~ /^\[\*\*\]/) {
        # is this an old style alert message
        $a = <INFILE>
        chomp($a);
        unless ( $a eq "") {
            # strip off the [**] from either end.
            s/(\s)\[\*\*\]\(\s)/\s/;g;
            $alert = $a;
        } else {
            print STDERR "Warning, file may be incomplete\n"
            next;
        }
    } else {
        print STDERR "Warning, input not recognized:\n"
        print STDERR "\t(line)\n"
        next;
    }

    # is this output from the portscan preprocessor
    if ( $alert =~ /spp_portscan:/ ) {
        if($spp_portscan_p) {
            # only do the work if we care
            $alert =~ s/spp_portscan:\s/\s/;
            if ( $alert =~ /^PORTSCAN DETECTED/ ) {
                $alert =~ s/^PORTSCAN DETECTED\s/\s/;
                $a = "$a$alert"
            }
        }
    }

    print $a;
    chomp($a);
}

$alert = "PORTSCAN DETECTED";
} elsif ( $alert =~ /^portscan status/ ) {
    $alert =~ s/\s/portscan status\s/;
    $a = "$a$alert";
    $alert = "portscan status";
} elsif ( $alert =~ /^End of portscan/ ) {
    $alert =~ s/\s/End of portscan\s/;
    $a = "$a$alert";
    $alert = "End of portscan";
} else {
    print STDERR "spp_portscan: $_\n";
    next;
}
} else {
    # ignore portscan logs
    next;
}

# put the alert into the hash table
push @{$alerts{$alert} }, $a;
}

if($opt_h) {
    # print out the relative html links to each entry
    foreach $key (keys (%alerts)) {
        $anchor = $key;
        $anchor =~ s/\s/_/g;
        print "<a href=#$anchor>$key</a><br>
";
    }
}

foreach $key (keys (%alerts)) {
    $anchor = $key;
    $anchor =~ s/\s/_/g;
    if($opt_h) {
        print "<hr>\n";
        print "<h3><a name=$anchor>$key</a></h3>\n";
        print "<ul>\n";
    } else {
        #plain text output goes here
    }
    @list = @{$alerts{$key}};
    $size = @list;
    for ( $i = 0 ; $i < $size ; $i++ ) {
    #plain text output goes here
}
$a = $list[$i];
($datentime,$data) = split('','$list[$i]', 2);
# spp_portscan logs look different
if( $data =~ /\^from\s/ ) {
    print "<li>$datentime $data</li>\n";
    next;
}
($datentime,$src,$arrow,$dest) = split('','$list[$i]');
($saddr,$sport) = split(':',"$src");
($daddr,$dport) = split(':',"$dest");
# reverse DNS lookups
if($opt_r) {
    $shost = resolve($saddr);
    $dhost = resolve($daddr);
} else {
    $shost = $saddr;
    $dhost = $daddr;
}
if($opt_w) {
    $shost = "<a href=$cgi_href$saddr>$shost</a>";
    $dhost = "<a href=$cgi_href$daddr>$dhost</a>";
}
if($opt_h) {
    print "<li>$datentime $shost:$sport $arrow $dhost:$dport</li>\n";
} else {
    # plain text output goes here
}
if($opt_h) {
    print "</ul>\n";
} else {
    # plain text output goes here
}

# HAL - Added to provide some (what I think are) useful links at the end of the report.
print "<hr><FONT COLOR="Red"><h2>Additional Reading & Information</FONT></h2>";
print "&nbsp;\n";
HREF="http://dLam.org/security.html">DLAM.ORG Security Links</A>&nbsp;\n";
print "&nbsp;\n";
HREF="http://www.doshelp.com/trojanports.htm">DOSHelp Trojan Port List</A>&nbsp;\n";
print "&nbsp;\n";
HREF="http://www.google.com">Google Search</A>&nbsp;\n";
print "&nbsp;\n";

HREF="http://www.iana.org/assignments/port-numbers">IANA Port List</A>&lt;BR&gt;n;
print "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n}
# Syntax:   %PATH%\perl anl_IDS.PL SNORT_FILE.EXT
# Purpose: Get list of number of attacks.
# #
# Original from GCIA Mike Bell (0318) Practical.
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc
#
foreach $file (@ARGV) {
    open(FILE, $file) || die "Can't open the file aaaaaaaahhhhhhh",
    # HAL - Modified original so the next line goes to screen and not to file.
    print STDERR "Examining File - $file\n";
    while (<FILE>) {
        /\](.*spp.*)\[/ && { next };
        /\](.*)\[/ && do {
            $volume{$1} ++ ;
            next;
        }
    }
    # HAL - Only use one of the following print statements.
    # HAL - Added this when modifying file to create tab seperated column headings.
    # print "Count,Attack Description\n";
    # HAL - Added this when modifying file to create comma seperated column headings.
    print "Count,Attack Description\n";
    #
    foreach $attack (sort keys(%volume)) {
        $parts = $volume{$attack} ;
        foreach $number (split(' ', $parts)) {
            # HAL - Use this line if you want tab seperated columns.
            # print "$number\t$attack\n";
            # HAL - Use this line if you want Comma seperated columns for CSV files.
            print "$number,\$attack\n";
        }
    }
}

C-3 Top_Talkers.pl

#!/perl
#
# File:   top_talkers.PL
# Syntax: %PATH%\perl top_talkers.PL SNORT_FILE.EXT
# Purpose: Count number of Top Talkers in SNORT ALERT Log.
# #
# Original from GCIA Mike Bell (0318) Practical.
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc
#
foreach $file (@ARGV) {
    open(FILE, $file) || die "Can't open the file aaaaaaaahhhhhhh"
    while (<FILE>) {
        /spp_portscan/ && do { next }
        /\d+\d+/ && do {
            $volume{"$1 $2"}++;
            next;
        };
        /\d+\d+/ && do {
            $volume{"$1 $2"}++;
            next;
        };
        /\d+\d+/ && do {
            $volume{"$1 $2"}++;
            next;
        };
        /
        foreach $pair (sort keys(%volume)) {
            $parts = $volume{$pair} ;
            foreach $number (split(' ', $parts)) {
                print "$number $pair";
            }
        }
    }
}
C-4 G.BAT

This is a batch file I used to save my self some typing on the command line when I was extracting and counting lines in each of the logs. When I wanted to get information on one particular host and get it from all three alert logs, then this thing did it for me.

The trick was to have a text editor that detected changes to open files. I would run the report and pipe the results to a file that I already had open in ConText. When the program was done I would switch to the ConText application and just answer Yes to the prompt telling me that my file on disk had changed and did I want to reload from disk. It’s not rocket science, but my fingers need all the relief they can get after typing this practical.

```batch
@echo off
IF "%1"=="X" goto EXTERNAL
IF "%1"=="x" goto EXTERNAL
SET A1=MY
SET A2=NET
SET A3=%1
SET A4=%2
SET F1=%3
SET F2=%4
SET O1=%4
IF "%A3%" == "/h" goto SYNTAX
IF "%A3%" == "/H" goto SYNTAX
IF "%A3%" == "-h" goto SYNTAX
IF "%A3%" == "-H" goto SYNTAX
IF "%A3%" == "/?" goto SYNTAX
IF "%A3%" == "-?" goto SYNTAX
IF "%A3%" == "help" goto SYNTAX
IF "%A3%" == "HELP" goto SYNTAX
IF "%A3%" == "Help" goto SYNTAX
IF "%A3%" == "" goto OCTMSG1
IF "%A4%" == "" goto OCTMSG2
IF "%F1%" == "" goto NOFILE
IF "%O1%" == "x" GOTO ALLLOGS
IF "%O1%" == "X" GOTO ALLLOGS
if "%F2%" == "" goto ALLLOGS
goto ONELOG

:ALLLOGS
echo Output being captured to %F1%.txt!
echo Checking Log files for [%A1%.%A2%.%A3%.%A4%]'s data!
```

```batch
echo ****************************************** >> %F1%.txt
echo Checking Log files for [%A1%.%A2%.%A3%.%A4%]'s data!  >> %F1%.txt
```

```batch
echo %A1%.%A2%.%A3%.%A4% >> %F1%.txt
```

Page 173 of 181

07/20/2001 - 11:37 AM

© SANS Institute 2000 - 2005

Author retains full rights.
echo ######################################## >> %F1%.txt
echo Checking Alert Log file.
echo Checking Alert Log for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
grep -ic "%A1%.%A2%.%A3\.%A4%:" 1-alerts.ids >> %F1%.txt
grep -i "%A1\%.|%A2\%.|%A3\%.|%A4%:" 1-alerts.ids >> %F1%.txt
echo ######################################## >> %F1%.txt
echo Checking OOS Log file.
echo Checking Out-Of-Spec Logs for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
grep -ic "%A1\%.|%A2\%.|%A3\%.|%A4%:" 2-oos.ids >> %F1%.txt
grep -i "%A1\%.|%A2\%.|%A3\%.|%A4%:" 2-oos.ids >> %F1%.txt
echo ######################################## >> %F1%.txt

if %O1%=="" goto ENDOOS
echo Getting OOS Entries.
echo Getting OOS Entries for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
egrep -ic "%A1%.|%A2%.|%A3%.|%A4%" 2-oos.ids >> %F1%.txt
egrep -i -B1 -A4 "%A1%.|%A2%.|%A3%.|%A4%" 2-oos.ids >> %F1%.txt
echo ##################### ################### >> %F1%.txt
:ENDOOS
echo END of Search for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
echo END of Search for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
goto END

:ONELOG
echo Output being captured to %F1%.txt!
echo Checking %F2% Log file for [%A1%.%A2%.%A3%.%A4%]'s data!
echo #################################################### >> %F1%.txt
echo Checking %F2% Log file for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
echo #################################################### >> %F1%.txt
grep -ic "my\.|\net\.|%A3%.|%A4%" %F2% >> %F1%.txt
grep -i "my\.|\net\.|%A3%.|%A4%" %F2% >> %F1%.txt
echo #################################################### >> %F1%.txt
echo END of Search for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt
echo END of Search for [%A1%.%A2%.%A3%.%A4%]'s data! >> %F1%.txt

:EXTERNAL
if "%2" == "" goto NOJOY
if "%3" == "" goto NOJOY
if "%4" == "" goto NOJOY
if "%5" == "" goto NOJOY
if "%6" == "" goto NOJOY
SET A1=%2
SET A2=%3
SET A3=%4
SET A4=%5
SET F1=%6
SET O1=%7
GOTO ALLLOGS

:NOJOY
echo.
echo Searches for external hosts required you supply
echo four octets and a log file to send to.
echo.
echo G 1 2 3 4 OutputFile
PAUSE
goto END

:OCTMSG1
echo.
echo #### ERROR WILL ROBINSON!
echo.
echo This normally happens when all required data elements
echo required to perform the search are missing.
goto SYNTAX

:OCTMSG2
echo.
echo #### YOUR GETTING WARMER BUBBA!
echo.
echo The second octet and output file name required
echo to perform this search are missing.
goto SYNTAX

:NOFILE
echo.
echo #### I ASSUME YOU WANT TO FILE THIS AWAY SOMEWHERE?
echo.
echo No output file name given.
goto SYNTAX

:SYNTAX
echo. echo The correct syntax is:
  echo. echo  G 3rdOctet 4thOctet OutputFile [LogFile]
  echo. echo  A TXT extension is automatically appended to OutputFile name.
  echo. echo  IP Address: %A1%.%A2%.%A3%.%A4%
  echo  Output File: %F1%
  echo. echo  Optional log file to search may be provided.
  echo. echo  [LogFile]: %F2%
:END

C-5 Snort_source.pl

#!/perl
# File: snort_source.PL
# Syntax: %PATH%\perl snort_source.PL SNORT_FILE.EXT
# Purpose: Get Source Addresses by number of scans.
#
# Original from GCIA Mike Bell (0318) Practical.
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc
#
# while (<>) {
# Check for blank line, if so process next line
#   if ( $$_ eq "" ) { next };
# Check for spp_portscan, if it is get the next record
# Tokenize the string so we can use it
#   if ($$_ =~ m/^\w{3}\s+\d+\s+\d+\s+([\[\w\d\.]\+]\:\\s+(\d+)\s+>-\s+(\[\w\d\.]\+]\:\\s+(\d+\s+UDP/) { $saddr = $1;
     $sport = $2;
     $daddr = $3;
     $dport = $4;
     $source{$saddr}++; # end if
#   #
#   if ($$_ =~ m/^\w{3}\s+\d+\s+\d+\s+([\[\w\d\.]\+]\:\\s+(\d+)\s+>-\s+(\[\w\d\.]\+]\:\\s+(\d+)\s+([-\w]+)\s+[*1PUSFAR]+\s+)/ { $saddr = $1;

Page 176 of 181
$sport = $2;
$daddr = $3;
$dport = $4;
$descrp = $5;
$source{$saddr}++; 
} # end if
} # while
# HAL - Use one of the following two lines for the output column headings.
# HAL - Use the following line for tab seperated column headings.
# print "Count\tAddress\n";
# HAL - Use the following line for comma seperated column headings.
print "Count\,Address\n";
#
foreach $num ( sort keys(%source) ) {
    $strings = $source{$num};
    #
    foreach $string (split( ', ', $strings)) {
        # HAL - Use one of the following two lines for the output.
        # HAL - Use the following line for tab seperated columns.
        print "$string\t$num\n";
        # HAL - Use the following line for comma seperated columns.
        print "$string\,\$num\n";
    }
}

C-6  OOS_TopSourceAddress.pl

#!/perl
# File: top_src.PL
# Syntax: %PATH% perl top_src.PL SNORT_FILE.EXT
# Purpose: Get list of top source addresses.
#
# Original from GCIA Mike Bell (0318) Practical.
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc
#
foreach $file (@ARGV) {
    open(FILE, $file) || die "Can't open the file aaaaaaaahhhhhhh";
#
    while (<FILE>) {
        #
        /\d+\d+\-\[\d\.\]+\s+([\w\d\.]+):(\d+)(s+\-\>|s+(\[\w\d\.\]+)):(\d+)/ & do {
# $1 - Source Address.
# $2 - Source Port
# $3 - Destination Address
# $4 - Destination Port

Page 177 of 181
$volume{$1}++;  
next;
}; # end pattern match  

# Comment out the next line if you don't want the column headings.  
# print "Hits,Port\n";  
# Use the next line if getting addresses.  
print "Hits,Source IP\n";  
# 
foreach $pair (sort keys(%volume)) {  
$parts = $volume{$pair} ;  
foreach $number (split(' ', $parts)) {  
# This outputs a Tab Seperated Variable file format.  
# print "$number\t$pair\n";  
# This outputs a Comma Seperated Variable file format.  
print "$number,$pair\n";  
}  
}  
}  

C-7  oosTopSourcePorts.pl

#!perl  
# File: top_src.PL  
# Syntax: %PATH% perl top_src.PL SNORT_FILE.EXT  
# Purpose: Get list of top source addresses.  
# 
# Original from GCIA Mike Bell (0318) Practical.  
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc  
# 
foreach $file (@ARGV) {  
    open(FILE, $file) || die "Can't open the file aaaaaaaahhhhhhhhh";

    while (<FILE>) {  
#  
# /\d+/\d+/\-[\d:\.]\+s+(\[\w\d\.]\+):\(\d+.s+\-\>[\w\d\.]\+):\(\d+)/ & & do {  
# $1 - Source Address.  
# $2 - Source Port  
# $3 - Destination Address  
# $4 - Destination Port  
    $volume{$2}++;  
next;
}; # end pattern match

# Comment out the next line if you don't want the column headings.
# print "Hits\,Port\n";
# Use the next line if getting addresses.
print "Hits\,IP Address\n";
foreach $pair (sort keys(%volume)) {
    $parts = $volume{$pair} ;
    foreach $number (split(' ', $parts)) {
        # This outputs a Tab Seperated Variable file format.
        print "$number\t$pair\n";
        # This outputs a Comma Seperated Variable file format.
        print "$number\,$pair\n";
    }
}

C-8 oosTopTalkersAddress.pl

#!/perl
# File:   top_talkers_oos.PL
# Syntax: %PATH%perl top_talkers.PL SNORT_FILE.EXT
# Purpose: Get source and destination address pairs from
#          Out Of Speck (OOS) SNORT Alert Logs.
#
# Original from GCIA Mike Bell (0318) Practical.
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc
#
foreach $file (@ARGV) {
    open(FILE, $file) || die "Can't open the file aaaaaaahhhhhhhhhhh";
    while (<FILE>) {
        #/\d+\/\d+-\[\d\:\.]+\s+(\[\w\d\.:]+)\:(\d+)\s+\->\s+(\[\w\d\.:]+)\:(\d+)/ && do {
        # $1 - Source Address.
        # $2 - Source Port
        # $3 - Destination Address
        # $4 - Destination Port
        # This line creates a pair of numbers seperated by two spaces.
        # Nice format for text files.
        # $volume{"$1 $4"}++;
        # This line creates comma seperated numbers. If you like CSV files then use
        # this line in conjunction with the CSV line below and pipe the output to
        # a file with a CSV extension. This combination creates a file you can
        # open in Excel with no problems at all.
        $volume{"$1 $3"}++;
        next;
    }
C-9 oosTopTalkersPorts.pl

#!/perl
#
# Original from GCIA Mike Bell (0318) Practical.
# http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc
#
foreach $file (@ARGV) {
   open(FILE, $file) || die "Can't open the file aaaaaaahhhhhhh";
    while (<FILE>) {

    # This line creates a pair of numbers seperated by two spaces.
    # Nice format for text files.
    # $volume("$1 $4")++;
    # This line creates comma seperated numbers. If you like CSV files then use
    # this line in conjunction with the CSV line below and pipe the output to
    # a file with a CSV extension. This combination creates a file you can
    # open in Excel with no problems at all.
$volume{"$2","$4"}++;  
next;
}; # end pattern match 2
}

# Comment out the next line if you don't want the column headings.
print "Hits\,SRC Port\,DST Port\n";
#
  foreach $pair (sort keys(%volume)) {
    $parts = $volume{$pair} ;
    foreach $number (split(\',\,$parts)) {
      # This outputs a Tab Seperated Variable file format.
      #       print "$number\t$pair\n";
      # This outputs a Comma Seperated Variable file format.
      print "$number\,$pair\n";
    }
  }
}

**C-10 SNORT.CHL (ConText Highlighter configuration):**

I use a freeware text editor that allows me to create custom Highlighter files (aka code Templates) for viewing Snort Log files on a Windows PC. It comes with several built in code templates for Perl, PHP, HTML, VBScript to name a few. The highlighter file I used is included here:

```plaintext
C-10 SNORT.CHL (ConText Highlighter configuration):

I use a freeware text editor that allows me to create custom Highlighter files (aka code Templates) for viewing Snort Log files on a Windows PC. It comes with several built in code templates for Perl, PHP, HTML, VBScript to name a few. The highlighter file I used is included here:

```
// multiline
// BlockCommentEnd - block comment end
LineComment:    #
BlockCommentBeg:
BlockCommentEnd:

  // identifier characters
  // note: characters shouldn't be delimited, except arrays
  // array of chars could be defined as from_char..to_char
IdentifierBegChars:  a..z A..Z 0..9 _
IdentifierChars:  a..z A..Z 0..9 _

  // numeric constants begin characters
  // note: characters shouldn't be delimited, except arrays
  // array of chars could be defined as from_char..to_char
  // number always starts with 0..9 except when NumConstBeg
  // defines other
NumConstBegChars:

  // numeric constants characters
  // note: characters shouldn't be delimited, except arrays
  // array of chars could be defined as from_char..to_char
  // number always starts with 0..9 except when NumConstBeg
  // defines other
NumConstChars:

  // escape character
EscapeChar:

  // keyword table
  // note: delimited with spaces, lines could be wrapped
  // you may divide keywords into three groups which can be
  // highlighted differently
KeyWords1:  TROJAN QUESO FINGERPRINT SERVER RAMEN MYSERVER WINGATE
            NMAP HPING HPING2 SMB EXPLOIT SUNRPC HIGHPORT TINY SUN
            FRAGMENTS PROBABLE SYN FIN RUSSIA DYNAMO STATDX STEALTH
            TRACEROUTE HIGH PORT RED WORM NULL SCAN HOSTILE RPC CALL
            EXTERNAL CONNECT OUTSIDE INSIDE
            515 1080 55850 65535

KeyWords2:  SPP_PORTSCAN WATCHLIST ATTEMPT ATTEMPTED POSSIBLE MY
            NET

KeyWords3:  TCP UDP TTL SEQ TOS ID DF ACK WIN MSS TS Options Sack
            SackOK ICMP SRC DST
string delimiter: StringBegChar - string begin char
StringEndChar - string end char
MultilineStrings - enables multiline strings, as perl
has it
StringBegChar: "
StringEndChar: "
MultilineStrings: 0

use preprocessor: 0 - no
1 - yes
note: if yes, '#' and statements after it will be
highlighted with Preprocessor defined colors
UsePreprocessor: 0

highlight line: 0 - no
1 - yes
note: if yes, current line will be highlighted
CurrLineHighlighted: 1

colors
note: first value is foreground, second is background color
and third (optional) represents font attribute:
B - bold
I - italic
U - underline
S - strike out
attributes can be combined: eg. B or BI
as value, it could be used any standard windows color:
clBlack, clMaroon, clGreen, clOlive, clNavy,
clPurple, clTeal, clGray, clSilver, clRed, clLime,
clYellow, clBlue, clFuchsia, clAqua, clLtGray,
clDkGray, clWhite, clScrollBar, clBackground,
clActiveCaption, clInactiveCaption, clMenu, clWindow,
clWindowFrame, clMenuText, clWindowText, clCaptionText,
clActiveBorder, clInactiveBorder, clAppWorkSpace,
clHighlight, clHighlightText, clBtntFace, clBtntShadow,
clGrayText, clBtntText, clInactiveCaptionText,
clBtntHighlight, cl3DDkShadow, cl3DLight, clInfoText,
clInfoBk
as value, it could be used hex numeric constant too:
$BBGGRR - BB: blue, GG: green, RR: red, eg: $FF6A00
SpaceCol: clWindowText clWindow
Keyword1Col: clRed clWindow B
Keyword2Col: clNavy clWindow B
Appendix D
Software Tools Used

ActivePerl for Windows – Larry Wall, GNU General Public License, http://www.perl.com

ConTEXT v0.96.1a - Eden Kinn, Freeware, http://www.fixedsys.com/context.


GNU Utilities for WIN32, K. M. Syring, GNU General Public License, ftp://ftp.uni-koeln.de (I just used egrep from this for now).

Microsoft Word 2000 – Microsoft Corporation.

Microsoft Excel 2000 – Microsoft Corporation.

PowerArchiver v6.11.0, Copyright © 1999-2001 ConeXware, Inc., http://www.powerarchiver.com

Programmers File Editor v1.01, Alan Phillips, Author has stopped development but says on his web page http://www.lancs.ac.uk/people/cpaap/pfe/, that the program is still available at Winsite http://www.winsite.com/info/pc/win95/misc/pfe101i.zip and Simtel http://www.simtel.net/pub/dl/11983.shtml for download.


Visio Technical 5.0 – Now owned by Microsoft Corporation.
## Upcoming Training

<table>
<thead>
<tr>
<th>Training Location</th>
<th>City</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS London April Live Online 2020</td>
<td>London, United Kingdom</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Fort Gordon Cyber Protection Brigade (CPB/ARCYBER) (SEC503) (Don Williams)</td>
<td>Augusta, GA</td>
<td>Apr 20, 2020 - Apr 25, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Instructor-Led Training</td>
<td>Apr 27</td>
<td>Baltimore, MD</td>
<td>Apr 27, 2020 - May 02, 2020</td>
</tr>
<tr>
<td>SANS Security West 2020</td>
<td>San Diego, CA</td>
<td>May 11, 2020 - May 16, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Live Online - SEC503: Intrusion Detection In-Depth</td>
<td>United Arab Emirates</td>
<td>May 28, 2020 - Jul 06, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>SANSFIRE 2020</td>
<td>DC</td>
<td>Jun 13, 2020 - Jun 20, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Rocky Mountain Summer 2020</td>
<td>CO</td>
<td>Jul 20, 2020 - Jul 25, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>Live Online - SEC503: Intrusion Detection In-Depth</td>
<td>United Arab Emirates</td>
<td>Jul 29, 2020 - Sep 04, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>Instructor-Led Training</td>
<td>Aug 3 ET</td>
<td>MA</td>
<td>Aug 03, 2020 - Aug 08, 2020</td>
</tr>
<tr>
<td>SANS vLive - SEC503: Intrusion Detection In-Depth</td>
<td>SEC503 - 202008,</td>
<td>Aug 10, 2020 - Sep 16, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>SANS Melbourne Live Online 2020</td>
<td>Australia</td>
<td>Aug 17, 2020 - Aug 22, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Network Security 2020</td>
<td>Las Vegas, NV</td>
<td>Sep 20, 2020 - Sep 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Northern VA - Reston Fall 2020</td>
<td>Reston, VA</td>
<td>Sep 28, 2020 - Oct 03, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>