Global Information Assurance Certification Paper

Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.3

Nils Reichen

SANS Parliament Square

April 22nd – 27th 2002 London

submitted: October 6, 2002

Table of Contents

Assignment 1 : Describe the State of Intrusion Detection

Cisco PIX subnet address handling vulnerability

Summary

The Cisco PIX firewall is handling the subnet address like a subnet broadcast. Answering to the subnet address, also called all-zero broadcast or Berkeley broadcasts, may be normal in certain situations. Early versions of Berkeley UNIX (BSD) used zeros to indicate the IP broadcast address. But as the PIX is a firewall, we may expect a correct handling of the subnet address and the subnet broadcast address. The goal of this assignment is to describe this issue, how it could be exploited and how to detect it.

Source and format of this trace

A SHADOW IDS installed to get interesting traffic for the practical assignment captured this vulnerability. This SHADOW host and a Snort IDS monitor the LAN segment in front of a Cisco PIX connected to the Internet. The Cisco PIX was running the latest OS version 6.2(2). As SHADOW use tcpdump to capture the network traffic, the traces are in the tcpdump format.

The Trace

[1] 07:32:38.283937 151.100.89.67.22 > MY.NET.97.224.22: S [tcp sum ok] 1345578220:1345578220(0) win 54580 (ttl 112, id 16368, len 40)
[2] 07:32:38.284163 MY.NET.97.224.22 > 151.100.89.67.22: S [tcp sum ok] 1124644420:1124644420(0) ack 1345578221 win 4096 <mss 1460> (ttl 255, id 6171, len 44)
[3] 07:32:38.330852 151.100.89.67.22 > MY.NET.97.253.22: S [tcp sum ok] 1345578220:1345578220(0) win 54580 (ttl 112, id 16368, len 40)
[4] 07:32:38.343561 151.100.89.67.22 > MY.NET.97.224.22: R [tcp sum ok] 1345578221:1345578221(0) win 0 (ttl 239, id 33823, len 40)
[5] 07:32:38.409393 151.100.89.67.4268 > MY.NET.97.224.22: S 1381191936:1381191936(0) win 32120 <mss 1460,sackOK,timestamp 542804848[|tcp]> (DF) (ttl 48, id 33829, len 60)
[6] 07:32:38.409556 MY.NET.97.224.22 > 151.100.89.67.4268: S [tcp sum ok] 872965664:872965664(0) ack 1381191937 win 4096 <mss 1460> (ttl 255, id 6173, len 44)
[7] 07:32:38.452593 151.100.89.67.4268 > MY.NET.97.224.22: . [tcp sum ok] 1:1(0) ack 1 win 32120 (DF) (ttl 48, id 33831, len 40)
[8] 07:32:38.453312 MY.NET.97.253.22 > 151.100.89.67.4268: P 872965665:872965684(19) ack 1381191937 win 4096 (ttl 255, id 6174, len 59)
[9] 07:32:38.497426 151.100.89.67.4268 > MY.NET.97.253.22: R [tcp sum ok] 1381191937:1381191937(0) win 0 (ttl 239, id 33833, len 40)
[10] 07:32:48.482733 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120 (DF) (ttl 48, id 34171, len 40)
…
[15 same TCP FIN/ACK packet with a random IP ID]
…
[26] 07:55:57.401447 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120 (DF) (ttl 48, id 47874, len 40)
…

[TCP SYN scan of all the following addresses of the subnet]

The full trace is included in appendix A.

Introduction

This attack targets the SSH service running on port 22. It's a common scan used to discover SSH servers to try to exploit vulnerabilities afterwards. First, I will not focus on the source address of this attack even if, according to DShield database, the SSH service scan from the same attacker IP address was seen by another IDS.
The first connection attempt for this common SSH scan uses the port 22 as source and destination port. If the attacker get a SYN/ACK answer, the connection is reseted and a new connection is now tried from an ephemeral client port (see the GCIA practical assignment of Kyle Haugsness for more information of this kind of SSH scans).
The trace above has been captured in the subnet MY.NET.97.224/27 used outside a Cisco PIX. The external interface of the PIX is using the address MY.NET.97.253
We will focus on the packets to and from these two IP addresses.

How it works

The trace analysis shows some strange behaviors. The first TCP SYN packet [1] sent to the address MY.NET.97.224 address get a TCP SYN/ACK answer [2]. Then a reset [4] is sent by the attacker to close this half-open connection – common behavior for this kind of SSH scan. The TCP SYN [3] sent to the PIX interface address MY.NET.97.253 gets no answer as expected.

The attacker sends a TCP SYN [5] to the target address MY.NET.97.224 but this time with an ephemeral client ports. As for the previous attempt, a SYN/ACK is answered [6]. Then the attacker host complete the TCP three-way handshake correctly [7]. But just after, a TCP packet with the ACK/PSH flags [8] is sent from MY.NET.97.253 port 22 to the attacker host. This packet uses the same client port (4268 here) as the connection established by the attacker with the host MY.NET.97.224 on port 22. The IP stack of the attacker host sends a TCP RST [9] as not connection was established with the host MY.NET.97.253 on port 22. The attacker finally sends a TCP FIN/ACK [10] to close the connection opened with MY.NET.97.224. This last packet is also common to the "standard" SSH scan pattern.

Taking a close look at the TCP sequence and acknowledgement numbers of the packets 5 to 10, regardless of the IP addresses, some points will come into light.

Packet	Flags	Sequence number	Acknowledgment number	TCP payload length
[5]	SYN	1381191936	0	0
[6]	SYN/ACK	872965664	1381191937	0
[7]	ACK	1381191937	872965665	0
[8]	ACK/PSH	872965665	1381191937	19
[9]	RST	1381191937	0	0
[10]	FIN/ACK	1381191937	872965665	0

The ACK/PSH packet [8] has the correct sequence and acknowledgement numbers used by the connection established by the packets 5 to 7. But the IP source of this packet is not the one that received the SYN [5], sent the SYN/ACK [6] and received the three-way handshake completion [6].

What's wrong there ?
I was first surprise to see an answer to the SSH connection attempts from the subnet address and a bit confused by the packets from the MY.NET.97.253 address. I was expecting to see the next packets coming from the same address. I ran a couple of test and found that the PIX was not handling the subnet address the way it should. The subnet address is used by the old BSD stack as subnet broadcast, it could be ok. But broadcasts are used for connection less protocols like UDP, ICMP, ….
It does not make sense for connection oriented protocol like TCP.

We can now say that the Cisco PIX firewall handle the subnet address like a broadcast. In addition, the TCP stack does not take care of the subnet address.
One point is still not clear, why the PIX answers to the SSH connection attempt on this address and not on the interface address [3]. In fact, the packets sent to the subnet address evade the list of allowed remote hosts.

If at least one remote host is allowed to connect using SSH, the PIX will answer for the connection attempt to the subnet address. The same behavior is also seen with telnet connections, if at least one remote host is allowed to use telnet. This is right even if all allowed hosts are on the inside network.

Today there is three way to remotely manage the Cisco PIX; telnet, SSH and a web interface using SSL. I also tested the web server to see if it was also affected by this vulnerability but it does not respond to the subnet address stimulus.

The only other response received to a stimulus targeting the subnet address is for ICMP echo request/reply but also only if ICMP process is running in the PIX. Other services like SNMP, ISAKMP, … did not respond to the stimulus.

Correlation and vendor information

The trace was posted on the IDS Europe mailing list for feedback. In the meantime I opened a case to the Cisco Systems Technical Assistance Center (TAC) asking them to forward this issue to their security response team (called PSIRT). The vendor security response team confirmed the behavior and issued a high priority patch development process. This issue is internally referenced by Cisco Systems as "CSCdy51810", but unfortunately this bug track detail is not publicly visible even for me. A patched OS build may be requested through the Cisco Systems TAC, the build version number is 6.2.2.111. The patch will be included in next builds.

Vulnerable OS versions:
- Version 6.2.2 has been confirmed as vulnerable.
- Older versions have not been confirmed, but due to the stack level of this vulnerability, they could be supposed vulnerable.

CERT has been notified about this vulnerability. In addition, a post will be done on the bugtraq@securityfocus.com and intrusions@incidents.org mailing lists.

Possible exploits

The Cisco PIX firewall may respond to three different stimulus sent to the subnet address: the TCP port 23 (telnet), TCP port 22 (SSH) and to ICMP echo requests. The Telnet and SSH service may be targeted to gain root access through a buffer overflow. A simpler exploit could be possible, find the concept described below:

- The attacker use a front-end router to drop all packets from his host to the PIX external interface address (drops the TCP RST sendtby the host stack but without blocking the packet sent to the PIX external subnet address)
- Create a script (in language C, Perl, ...) that will:
  - send a TCP SYN to the PIX
  - capture the answer with a capture tool (tcpdump in a filehandle, ...)
  - forge the ACK packet based on the response captured, and send it to the PIX external subnet address and not to the interface address.
  - capture the next packets and send the related responses.

A SSH or telnet like fake application may be built following this concept. The missing "sesame" key is the password. With a good dictionary, an attacker should find quickly some PIX firewall with trivial passwords as the administrator thinks that only internal access is allowed.

The tests performed by attempting to telnet the subnet address were successfully followed after the three-way handshake by the authentication prompt. As the packet is coming from the PIX interface address, it will be dropped by the stack and not shown in the telnet prompt.

How to detect it

The detection of an attempted use of this vulnerability without generating a false positive for another telnet/SSH connection is the goal there. There are at least two manners to accomplish this job.

First the Cisco SSH and telnet answers may be identified by the server banner. But how to distinguish a telnet or SSH connection to a Cisco router from one to a Cisco PIX firewall ?
The first one needs a specific rule to match the response originated from the external subnet IP address. The alert rule needs to match a TCP SYN/ACK response packet with the PIX external subnet IP address as source. Broadcasts cannot be used with TCP, there is no risk of false positive coming from another host on the same subnet and using the subnet address as broadcast.

The Snort IDS rules to detect potential telnet and SSH exploits are:

```
# Define the PIX external subnet and interface addresses to avoid false alarms
var PIX_EXTERNAL_NET <subnet address of the PIX external interface>/32

activate tcp  $PIX_EXTERNAL_NET 23 -> $EXTERNAL_NET any (msg:"TELNET Cisco PIX login
attempted"; flags:SA+; activates:1; classtype: attempted-admin;)
dynamic tcp  $PIX_EXTERNAL_NET 23 <> $EXTERNAL_NET any (activated_by:1; count:50;)
#
activate tcp  $PIX_EXTERNAL_NET 22 -> $EXTERNAL_NET any (msg:"SSH Cisco PIX login
attempted"; flags:SA+; activates:2; classtype: attempted-admin;)
dynamic tcp  $PIX_EXTERNAL_NET 22 <> $EXTERNAL_NET any (activated_by:2; count:50;)
#
```

The activate/dynamic pair of rules will generate an alert and then log the following packet of the session. The "activate" rule above will produce an alert according to the specified packet pattern and activate the "dynamic" rule. This "dynamic" rule will log the following 50 packets matching the pattern specified. It will allow the identification of the attacker activity.

The second manner is more generic and does not need to specify the IP address of the external subnet. However, there is a risk of false positive which needs to be minimized. Telnet sessions to a Cisco device may be identified through the specific banner/prompt and through the protocol version exchange for SSH. At the beginning of the SSH connection, the client and the server have to exchange an identification string (form: "SSH-protoversion-softwareversioncomments") according to the SSH protocol specification. For SSH, the identification strings for a Cisco router and for a Cisco PIX is the same. So there is a risk of false positive if SSH connections to internal Cisco devices are authorized. However, SSH connections from the outside network to manage internal Cisco devices is not current, at least not for firewall directly connected to Internet. For telnet, the Cisco PIX use different telnet options than a Cisco router. The Cisco PIX uses ECHO, SUPPRESS GO AHEAD with repeating the ECHO option after the SUPPRESS GO AHEAD. Including these options in the alert pattern and as incoming telnet access is not current, the false positive may be minimized.

The snort IDS rules to detect potential telnet and SSH exploits for this second manner is:

```
activate tcp  $HOME_NET 23 -> $EXTERNAL_NET any (msg:"TELNET Cisco PIX login attempt";
flags:A+; content:"|FF FB 01 FF FB 03 FF FB 01 0D 0A 0D 0A|User Access Verification";
activates:1; classtype: attempted-admin;)
dynamic tcp  $HOME_NET 23 <> $EXTERNAL_NET any (activated_by:1; count:50;)
#
activate tcp  $HOME_NET 22 -> $EXTERNAL_NET any (msg:"SSH Cisco sshd response"; flags:A+;
content:"SSH-1.5-Cisco-"; activates:2; classtype: attempted-admin;)
dynamic tcp  $HOME_NET 22 <> $EXTERNAL_NET any (activated_by:2; count:50;)
#
```

The activate/dynamic pair of rules is used in the same way than the first manner.

How to protect your network

Cisco Systems released a patch to address this vulnerability in a specific OS build with version 6.2.2.111. The patch will be included in the new official releases. To upgrade your Cisco PIX firewall, a request need to be send to the Cisco TAC (Technical Assistance Center http://www.cisco.com/tac). The PIX may be also protected by adding a filter in the upstream router to deny the SSH and telnet traffic targeting the PIX external interface address and the corresponding subnet address. The access control lists on the PIX interface do not protect the PIX itself. The internal daemons of the PIX are not filtered by the access control lists applied on one "interface". If the upstream router cannot be used to filter the traffic and the PIX is still using a non-patched version. The only way to be protected against this vulnerability, is to remove all the SSH and telnet configuration. It will then remain two ways to manage the PIX: the HTTPS web interface and the console port.

References

IETF Internet Draft "SSH Transport Layer Protocol" Sep 20, 2002
URL: http://www.ietf.org/internet-drafts/draft-ietf-secsh-transport-15.txt (Sep 25, 2002)

IETF RFC "TELNET PROTOCOL SPECIFICATION " May 1983
URL: http://www.rfc-editor.org/rfc/rfc854.txt (Sep 20, 2002)

IETF RFC "TELNET ECHO OPTION" May 1983
URL: http://www.rfc-editor.org/rfc/rfc857.txt (Sep 20, 2002)

IETF RFC "TELNET SUPPRESS GO AHEAD OPTION" May 1983
URL: http://www.rfc-editor.org/rfc/rfc858.txt (Sep 20, 2002)

SSH Communications Security "SSH Secure Shell Specifications"
URL: http://www.ssh.com/products/ssh/specifications.cfm (Sep 25, 2002)

Kyle Haugsness GCIA "GCIA Practical Assignment" January 2002
URL: http://www.giac.org/practical/Kyle_Haugsness_GCIA.zip (Aug 25, 2002)

Cisco Systems "Technical Assistance Center" Sep 23, 2002
URL: http://www.cisco.com/public/support/tac/ (Sep 23, 2002)

IDS Europe mailing list "ids-europe Info Page" Jun 24, 2001
URL: https://ids-europe.alchemistowl.org/mailman/listinfo/ids-europe (Sep 23, 2002)

DShield.org "Search the DShield database" Aug 31, 2002
URL: http://www.dshield.org/search.php (Aug 31, 2002)

DShield.org "IP Info" for 151.100.89.67 Aug 31, 2002
URL: http://www.dshield.org/ipinfo.php?ip=151.100.089.067 (Aug 31, 2002)

Michael Castro GCIH "The NAPTHA Cluster Another Flavor of DoS" Apr 4, 2001
URL: http://www.giac.org/practical/Michael_Castro_GCIH.doc (Sep 24, 2002)

BindView's RAZOR Security Team "The Naptha DoS vulnerabilities" Nov 30, 2000
URL: http://razor.bindview.com/publish/advisories/adv_NAPTHA.html (Sep 24, 2002)

Packet Storm security tools resource "Naptha v1.1" Jan 27, 2001
URL: http://packetstorm.dnsi.info/0101-exploits/naptha-1.1.tgz (Sep 24, 2002)

Naval Surface Warfare Center - Dahlgren Lab "NSWC SHADOW INDEX"
URL: http://www.nswc.navy.mil/ISSEC/CID/ (Aug 22, 2002)

Martin Roesch "Snort User Manual Snort Release: 1.9.x" Apr 26, 2002
URL: http://www.snort.org/docs/SnortUsersManual.pdf (Aug 30,2002)

TCPDump.org "TCPDUMP public repository" Jun 5, 2002
URL: http://www.tcpdump.org (Aug 22, 2002).

Assignment 2 : Network Detects

Detect #1: Strange Gnutella traffic

Trace Log

12:18:24.254488 46.5.180.250.61284 > 24.165.145.34.6385: P 3979059731:3979059785(54) ack 486191126 win 17520 (DF)
12:23:56.744488 24.165.145.34.6385 > 46.5.180.250.61284: . 487265248:487265460(212) win 20504 (DF)
12:39:08.324488 24.165.145.34.6385 > 46.5.180.250.61284: . 489719086:489720042(956) win 20504 (DF)
12:39:19.914488 24.165.145.34.6385 > 46.5.180.250.61284: RWE 3808783643:3808783940(297) ack 4110633783 win 20504 urg 38205 (DF)
12:46:14.694488 24.165.145.34.6385 > 46.5.180.250.61284: . 490699856:490699936(80) win 20504 (DF)
13:21:24.094488 24.165.145.34.6385 > 46.5.180.250.61284: FW 10716463:10716479(16) ack 315968346 win 20496 urg 42160 (DF)
13:31:44.834488 24.165.145.34.6385 > 46.5.180.250.61284: . 498710468:498710705(237) win 20504 (DF)
13:45:29.774488 24.165.145.34.6385 > 46.5.180.250.61284: SRW 501100658:501101027(369) ack 60804 win 20504 (DF)
13:46:41.054488 24.165.145.34.0 > 46.5.180.250.6385: SPE 4016315872:4016316105(233) win 20504 urg 23035 (DF)

The full trace is included in appendix B.

Source of Trace

This trace was extracted from the logs file "2002.5.13" at http://www.incidents.org/log/Raw.

Detect was generated by

This detect was captured by a Snort IDS using an unknown rule set. The monitored network structure is not known. But it seems to be the internet access of a university in USA, probably in New York. The monitored network address was sanitized by replacing them in the packets. The home network in the traces is 46.5.0.0/16 or at least several C classes in the 46.5.0.0/16 range.

Probability the source address was spoofed

The source address is certainly not spoofed. After the first packet of the trace going out of the monitored network, some responses came back. The TCP three-way handshake was certainly completed but it is not part of the trace as it did not generate an alert. This detect was logged because the signature used matches at least the Gnutella protocol traffic and the out of the specification TCP flags and reserved bits.

Description of attack

A tcpdump output with the hex and ASCII of the first packet show the type protocol used:

```
12:18:24.254488 46.5.180.250.61284 > 24.165.145.34.6385: P [bad tcp cksum f9f9!]
3979059731:3979059785(54) ack 486191126 win 17520 (DF) (ttl 124, id 14661, len 94, bad cksum
3e94!)
0x0000  4500 005e 3945 4000 7c06 3e94 2e05 b4fa   E..^9E@.|.>.....
0x0010  18a5 9122 ef64 18f1 ed2b a213 1cfa b016   ...".d...+......
0x0020  5018 4470 8a4c 0000 474e 5554 454c 4c41   P.Dp.L..GNUTELLA
0x0030  2043 4f4e 4e45 4354 2f30 2e36 0d0a 5573   .CONNECT/0.6..Us
0x0040  6572 2d41 6765 6e74 3a20 476e 7563 6c65   er-Agent:.Gnucle
0x0050  7573 2031 2e36 2e30 2e30 0d0a 0d0a        us.1.6.0.0....
```

The bad checksum shown by tcpdump is due to the replacement of the inside IP addresses to keep anonymity. These bad checksums have to be ignored for all packets.

The Gnutella protocol is a peer-to-peer protocol used to exchange binary files. It's not a protocol defined by a RFC but the specifications are widely discussed by all the contributors and then implemented. Big Gnutella networks exist on Internet, the exchanged files are mostly audio in MP3 format.
This first packet logged shows a Gnutella connection request using the version 0.6 of the protocol and done by a client software called "Gnucleus" version 1.6.0.0.

About 5 minutes later, the remote host sends back a first packet. Packets follow with different time between them:

Time	Source ip.port	Destination ip.port	TCP Offset	Reserved bits	TCP flags
12:18:24	46.5.180.250.61284	24.165.145.34.6385	5	0000 00	AP
12:23:56	24.165.145.34.6385	46.5.180.250.61284	0	0000 00	000000
12:39:08	24.165.145.34.6385	46.5.180.250.61284	0	0000 00	000000
12:39:19	24.165.145.34.6385	46.5.180.250.61284	1	0001 11	ARU
12:46:14	24.165.145.34.6385	46.5.180.250.61284	0	0000 00	000000
13:21:24	24.165.145.34.6385	46.5.180.250.61284	1	1011 10	AFU
13:31:44	24.165.145.34.6385	46.5.180.250.61284	0	0000 00	000000
13:45:29	24.165.145.34.6385	46.5.180.250.61284	1	1100 10	SAR
13:46:41	24.165.145.34.0	46.5.180.250.6385	2	1101 01	SPU

Taking a look at the TCP offset (TCP header length in 32-bit word), the reserved bits and the flags, there is no doubt that the packets sent by the attacker are crafted. The timestamps are also interesting, it really looks like someone is behind the screen of the attacker host and forging the packets. Something is still not correct, the TCP Offset is never below the minimum TCP header length. This minimum size is 20 bytes (5 x 32bits word). A TCP offset of 1 word only includes the source and destination port in the header. The packet content may help in this situation, see underneath the packets sent by the attacker with the hex and ASCII:

```
12:23:56.744488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum fa5c!]
487265248:487265460(212) win 20504 (DF) (ttl 110, id 57779, len 232, bad cksum a39b!)
0x0000  4500 00e8 e1b3 4000 6e06 a39b 18a5 9122  E.....@.n......"
0x0010  2e05 b4fa 18f1 ef64 1d0b 13e0 ed35 67ea  .......d.....5g.
0x0020  0000 5018 1f16 1ad7 0000 fdcc d1e1 06a8  ..P.............
0x0030  805b ff5d 1c0f 8ad8 0100 0001 0600 0000  .[.]............
0x0040  0001 c11c ec4b 304b f5ff 01d8 9777 1774  .....K0K.....w.t
0x0050  0000 0304 0000 0000 fec7 0061 dccf 0e46  ...........a...F
0x0060  8871 8ac8 3f7b aec7 8103 035c 0000 0001  .q..?{.....\....
0x0070  ca18 0a00 0004 9500 0000 8803 0000 a233  ...............3
0x0080  2800 4661 6974 686c 6573 7320 2d20 546f  (.Faithless.-.To
0x0090  7461 6c6c 7920 4461 6e63 6520 2d20 3031  tally.Dance.-.01
0x00a0  202d 2057 6520 436f 6d65 2031 2e6d 7033  .-.We.Come.1.mp3
0x00b0  0000 4d52 5048 021e 1901 f010 6427 bf34  ..MRPH......d'.4
0x00c0  7cd6 1185 fc00 a0cc 7bbe 944c b073 961f  |.......{..L.s..
0x00d0  3a66 4797 2e2e 6c21 90aa 1780 0106 0800  :fG...l!........
0x00e0  0000 0000 676f 6c65                      ....gole
12:39:08.324488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum fa2c!]
489719086:489720042(956) win 20504 (DF) (ttl 110, id 21426, len 976, bad cksum 2eb5!)
0x0000  4500 03d0 53b2 4000 6e06 2eb5 18a5 9122  E...S.@.n......"
0x0010  2e05 b4fa 18f1 ef64 1d30 852e ed49 de30  .......d.0...I.0
0x0020  0000 5018 2115 ef97 0000 d093 29e2 c724  ..P.!.......)..$
0x0030  596e ff30 6243 b5c1 bb00 0001 0600 0000  Yn.0bC..........
0x0040  0028 a318 e4af 7ed6 1186 9d44 4553 5400  .(....~....DEST.
0x0050  0000 0106 0000 0000 c91e 18f8 b5fb 9253  ...............S
[snipped for brevity]
12:39:19.914488 24.165.145.34.6385 > 46.5.180.250.61284: RWE [bad tcp cksum f9f9!]
3808783643:3808783940(297) ack 4110633783 win 20504 urg 38205 [RST+
...0...J..P.!..=.....I...B..C.] (DF) (ttl 110, id 64181, len 321, bad cksum 8a40!)
0x0000  4500 0141 fab5 4000 6e06 8a40 18a5 9122  E..A..@.n..@..."
0x0010  2e05 b4fa 18f1 ef64 0000 1d30 e22e ed4a  .......d...0...J
0x0020  11f4 5018 21ce 953d 0000 f783 e649 8608  ..P.!..=.....I..
0x0030  9842 afdb 43da 1bda 126a 8001 0617 0000  .B..C....j......
0x0040  0000 0061 6365 206f 6620 6261 7365 206d  ...ace.of.base.m
0x0050  7033 0075 726e 3a00 099d 266c 587e d611  p3.urn:...&lX~..
[snipped for brevity]
12:46:14.694488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum fa01!]
490699856:490699936(80) win 20504 (DF) (ttl 110, id 17955, len 100, bad cksum 3fb0!)
0x0000  4500 0064 4623 4000 6e06 3fb0 18a5 9122  E..dF#@.n.?...."
0x0010  2e05 b4fa 18f1 ef64 1d3f 7c50 ed4e 45e7  .......d.?|P.NE.
0x0020  0000 5018 1fc0 65f5 0000 2db1 9969 f206  ..P...e...-..i..
0x0030  9f56 fff2 ef12 14d7 6900 0001 0600 0000  .V......i.......
0x0040  00f5 8736 3854 dd00 9924 ddcc 4159 8381  ...68T...$..AY..
0x0050  b401 0205 0e00 0000 ca18 18ea eebc c000  ................
0x0060  0000 cb93                                ....
13:21:24.094488 24.165.145.34.6385 > 46.5.180.250.61284: FW [bad tcp cksum f9f9!]
10716463:10716479(16) ack 315968346 win 20496 urg 42160 (DF) (ttl 110, id 65368, len 40, bad
cksum 86b6!)
0x0000  4500 0028 ff58 4000 6e06 86b6 18a5 9122  E..(.X@.n......"
0x0010  2e05 b4fa 18f1 ef64 1d9e 3544 0000 ed6d  .......d..5D...m
0x0020  1bb1 5010 200c a4b0 0000 0000 0000       ..P...........
13:31:44.834488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum f9f9!]
498710468:498710705(237) win 20504 (DF) (ttl 110, id 62193, len 257, bad cksum 9244!)
0x0000  4500 0101 f2f1 4000 6e06 9244 18a5 9122  E.....@.n..D..."
0x0010  2e05 b4fa 18f1 ef64 1db9 b7c4 ed73 ec5c  .......d.....s.\
0x0020  0000 5018 1d1d 94e4 0000 8b2b 5c2c e177  ..P........+\,.w
0x0030  b165 ff26 a52e f8f2 2d00 0001 0600 0000  .e.&....-.......
0x0040  0067 74fc 750c e751 42ae c43a 0f76 ad83  .gt.u..QB..:.v..
```

```
[snipped for brevity]
13:45:29.774488 24.165.145.34.6385 > 46.5.180.250.61284: SRW [bad tcp cksum f9f9!]
501100658:501101027(369) ack 60804 win 20504 [RST+ ..0r......P."8.O...:.S.......] (DF) (ttl 110, id
32198, len 393, bad cksum 6e8!)
0x0000  4500 0189 7dc6 4000 6e06 06e8 18a5 9122  E...}.@.n......"
0x0010  2e05 b4fa 18f1 ef64 1dde 3072 0000 ed84  .......d..0r....
0x0020  1c96 5018 2238 ed4f 0000 b43a db53 a4b1  ..P."8.O...:.S..
0x0030  f288 dede de0b 91d2 68ad 8001 062c 0000  ........h....,..
0x0040  0000 0073 7461 7274 7265 6b20 6473 3920  ...startrek.ds9.
[snipped for brevity]
13:46:41.054488 24.165.145.34.0 > 46.5.180.250.6385: SPE [bad tcp cksum f9f9!]
4016315872:4016316105(233) win 20504 urg 23035 (DF) (ttl 110, id 46809, len 261, bad cksum
ce58!)
0x0000  4500 0105 b6d9 4000 6e06 ce58 18a5 9122  E.....@.n..X..."
0x0010  2e05 b4fa 0000 18f1 ef64 1de0 757a ed85  .........d..uz..
0x0020  2d6a 5018 20ea 59fb 0000 d281 6400 e1d0  -jP...Y.....d...
0x0030  2d4c a6ca 1100 d633 d28a 8001 0611 0000  -L.....3........
0x0040  0000 006d 7920 7374 6f6e 6579 0075 726e  ...my.stoney.urn
0x0050  3a00 aa40 0969 d72a 2943 90fb 17e2 c749  :..@.i.*)C.....I
0x0060  0414 8001 060e 0000 0000 0062 7261 7a69  ...........brazi
[snipped for brevity]
```

The IP header and the TCP header up to the acknowledgement number seem correct. Taking a specific look at the line 0x0020 of each packet, the pattern '5018' is present in all packets except one. This pattern is like a flashing popup in this trace. It is present in the "window" part of the header but it's more frequent to see it in the "offset/reserved bits/flags" part at the beginning of the line 0x0020.
The packet would be correct if 0x50 0x18 was present in the offset/reserved bits/flags part. In this case, the offset value of 5 words, the null reserved bits and the AP flags would be correct. It looks like the TCP header was wrongly forged and two bytes were inserted just after the acknowledgement number. By snipping these two bytes in the packets from the attacker, the trace might have looked:

Time	Source ip.port	Destination ip.port	TCP Offset	Reserved bits	TCP flags
12:18:24	46.5.180.250.61284	24.165.145.34.6385	5	0000 00	AP
12:23:56	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	AP
12:39:08	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	AP
12:39:19	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	AP
12:46:14	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	AP
13:21:24	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	A
13:31:44	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	AP
13:45:29	24.165.145.34.6385	46.5.180.250.61284	5	0000 00	AP
13:46:41	24.165.145.34.0	46.5.180.250.6385	5	0000 00	AP

Attack mechanism

A detailed analysis of the Gnutella protocol is a good start to understand what the attacker is trying to achieve.

The Gnutella protocol header and specification may be found in the references listed. Basically, after the TCP connection has been established, the initiator sends a "GNUTELLA CONNECT/<proto version>\n\n" ("\n" is line feed character). A "GNUTELLA OK\n\n" is expected in response. In the version 0.6 of the protocol, the "User-Agent: <name>" has been added to the connect message and the line endings is now the carriage-return line feed. Then the initiator should identify to the other member of the network. Sending a Gnutella ping packet to the connected peer does it. This peer will forward the Gnutella ping to all other peers and the Gnutella TTL field will be decremented. This Gnutella ping packet will reach all the hosts of the Gnutella network, up to the Gnutella TTL. Each peers will answer this ping message back to the initiator using a Gnutella pong. The ping message has an empty payload, which is used for routing purpose inside the Gnutella network. Only the file downloads and uploads are done with a direct connection. All other kinds of Gnutella traffic is using the Gnutella routing between the peers. After the ping/pong step, a Gnutella query packet (also called search packet) may be send to search for an item. A query response will come back only from the host which have match the searched item.

From the analysis of the TCP header, it has certainly two bytes inserted after the acknowledgement number that should not be there. By snipping these two bytes, the TCP header of the attacker's packets looks all right. But now, how will look the attacker's Gnutella packets in this case ?

Time	Function	TTL	Hop count	Data length field [byte]	Real data length [byte]	minimum speed field
12:23:56	ping	1	6	0	167	N/A for ping msg
12:39:08	ping	1	6	0	895	N/A for ping msg
12:39:19	query	1	6	232	254	0
12:46:14	ping	1	6	0	35	N/A for ping msg
13:21:24	packet too short, Gnutella header not complete : 0x00 0x00 0x00 0x00					
13:31:44	ping	1	6	0	192	N/A for ping msg
13:45:29	query	1	6	52	326	0
13:46:41	query	1	6	136	194	0

The Gnutella header fields are not far away from the standard, except of course the length field and one short packet. Taking into account what could have been the TCP and Gnutella headers, there is a high probability that the attacker did a mistake in the TCP header structure. The attacker was developing a new tool and tested his code. This tool cannot be a simple Gnutella client as the TCP header was forged. The attacker was developing a new hostile code. Based on the available trace, it's not easy to find what could be the purpose of this tool. The data length does not match the real length. One possible purpose could be the use of a short data length field to create a buffer overflow in one of the well-known Gnutella software.

The full trace is included in appendix B.

Correlations

The attacker IP address is not included in the DShield.org database and no match is present on the Google search engine. No vulnerability and buffer overflow exploit was found concerning Gnutella software. The comment by Eric Chien from Symantec Security Response Team in the "Virus Bulletin" magazine of January 2002 spoke about known viruses propagating through peer-to-peer networks. He speaks also about the possible future exploit of peer-to-peer software bugs to spread viruses.

Evidence of active targeting

Even if an inside host initiated the Gnutella connection request, the response packets are definitively active targeting. The attacker does not succeed to create the right TCP header, but the packets were forged and targeted specifically to one host.

Severity

Criticality: 1
The target network is unknown but seems to be a university campus. The target is a user workstation.

Lethality: 4
The attacker was trying a new tool and testing his development. The target host is running Windows, because Gnucleus software is only available on Windows platform. The bad "offset/reserved bits/flags" may cause a crash on old non-patched Windows systems.

System Countermeasures: 3
The inside host software patches level is unknown. A middle mark may be given, it's highly dependent on the care taken by the security officer. The outgoing traffic from the user workstation did not stop right after the attack. The system seems to not be vulnerable or the attack was stopped by a firewall.

Network Countermeasures: 4
Internal hosts seem to be protected by a state-full firewall.

Attack Severity: (1 + 4) - (3 + 4) = -2

Defensive recommendation

A statefull firewall will block the packets with invalid TCP header. However, the real threat here is the Gnutella protocol misuse. A firewall allowing Gnutella connection will not check the Gnutella protocol header. No application proxy exists for Gnutella protocol. If your internal policy allow you to filter Gnutella, it could be a safe option as well as saving bandwidth. Otherwise take a closer look at the Gnutella traffic for abnormal activity. If in the future a new worm appear, a list of all internal hosts with peer-to-peer software will be a great help.

intrusions@incidents.org mailing list

This detect was posted to the mailing list on Friday 4th October, according to the practical assignment. Unfortunately, I was a bit late for the editing of this paper and no questions have been posted on the mailing list. However, I will answer all coming questions.

Multiple choice test question

Why the TCP "offset/reserved bits/flags" fields are '11f4' and therefor not valid ?
Do not take care of the bad checksum, it's due to the replacement of the destination to keep anonymity.

```
12:39:19.914488 24.165.145.34.6385 > 46.5.180.250.61284: RWE [bad tcp cksum f9f9!]
3808783643:3808783940(297) ack 4110633783 win 20504 urg 38205 [RST+
...0...J..P.!..=.....I...B..C.] (DF) (ttl 110, id 64181, len 321, bad cksum 8a40!)
0x0000  4500 0141 fab5 4000 6e06 8a40 18a5 9122   E..A..@.n..@..."
0x0010  2e05 b4fa 18f1 ef64 0000 1d30 e22e ed4a   .......d...0...J
0x0020  11f4 5018 21ce 953d 0000 f783 e649 8608   ..P.!..=.....I..
0x0030  9842 afdb 43da 1bda 126a 8001 0617 0000   .B..C....j......
0x0040  0000 0061 6365 206f 6620 6261 7365 206d   ...ace.of.base.m
0x0050  7033 0075 726e 3a00 099d 266c 587e d611   p3.urn:...&lX~..
[snipped for brevity]
```

A. The packet has been crafted to elude the IDS
B. The packet has been crafted to try to crash the target
C. The packet was wrongly crafted by the attacker, two bytes was inserted before the correct "offset/reserved bits/flags" bytes
D. The attack is used to pass through a statefull firewall using a TCP header length of 1.

Answer is C.
The packets are all related to same TCP connection according to the sequence and acknowledgement numbers. On the client side, the same port cannot be used for two TCP connections at the same time.

References

Cap'n Bry's "Gnutella Protocol Specifications"
URL: http://capnbry.net/gnutella/protocol.php (Sep 25, 2002)

Gnutelladev.com "The Gnutella Protocol"
URL: http://www.gnutelladev.com/protocol/gnutella-protocol.html (Sep 25, 2002)

Gordon Mohr "Hash/URN Gnutella Extensions (HUGE) v0.94" April 30, 2002
URL: http://groups.yahoo.com/group/the_gdf/files/Proposals/HUGE/draft-gdf-huge-0_94.html (Sep 25,2002)

Gnucleus.com "Gnutella protocol" Jun 29, 2002
URL: http://www.gnucleus.com/research/protocol.html (Sep 25, 2002)

Virus Bulletin "Comment" part by Eric Chien Jan 2002
ISSN 0956-9979
URL: http://www.virusbtn.com/magazine/archives/pdf/2002/200201.PDF (Sep 25, 2002)

DShield.org "IP Info" for 24.165.145.34 Sep 25, 2002
URL: http://www.dshield.org/ipinfo.php?ip=24.165.145.34 (Sep 25, 2002)

Google.com search engine 2002
URL: http://www.google.com (Sep 25, 2002)

TCPDump.org "TCPDUMP public repository" Jun 5, 2002
URL: http://www.tcpdump.org (Aug 22, 2002).

Detect #2: Fragment Scan using TCP RST

Trace Log

```
[1]  17:43:04.382106 61.179.112.130.34562 > MY.NET.97.10.80: S [tcp sum ok]
     679204100:679204100(0) win 16384 <mss 1402,nop,nop,sackOK> (DF) (ttl 110, id 1475, len
     48)
     0x0000    4500 0030 05c3 4000 6e06 34b6 3db3 7082   E..0..@.n.4.=.p.
     0x0010    xxxx 610a 8702 0050 287b d504 0000 0000   ..a....P({......
     0x0020    7002 4000 ec37 0000 0204 057a 0101 0402   p.@..7.....z....
[2]  17:43:04.650628 MY.NET.97.24.137 > 61.179.112.130.137:  >>> NBT UDP Pkt(137): Query;
     REQ; UNICAST TrnID=0x9FB8 OpCode=0 NmFlags=0x0 Rcode=0 QueryCount=1
     AnswerCount=0 AuthorityCount=0 AddressRecCount=0 QuestRecs: Name=
     WARNING: Short packet. Try increasing the snap length
      (ttl 127, id 34550, len 78)
     0x0000    4500 004e 86f6 0000 7f11 e24b xxxx 6118   E..N.......K..a.
     0x0010    3db3 7082 0089 0089 003a 4e19 9fb8 0000   =.p.......:N.....
     0x0020    0001 0000 0000 0000 2043 4b41 4141 4141   ........CKAAAAA
     0x0030    4141 4141 4141                            AAAAAA
[3]  17:43:05.033497 61.179.112.130 > MY.NET.97.24: icmp: 61.179.112.130 udp port 137
     unreachable [tos 0xc0]  (ttl 238, id 6792, len 56)
     0x0000    45c0 0038 1a88 0000 ee01 df1f 3db3 7082   E..8........=.p.
     0x0010    xxxx 6118 0303 ad97 0000 0000 4500 004e   ..a.........E..N
     0x0020    86f6 0000 6611 fb4b xxxx 6118 3db3 7082   ....f..K..a.=.p.
     0x0030    0089 0089 003a                            .....:
[4]  17:43:06.149074 MY.NET.97.24.137 > 61.179.112.130.137:  >>> NBT UDP Pkt(137): Query;
     REQ; UNICAST TrnID=0x9FBA OpCode=0 NmFlags=0x0 Rcode=0 QueryCount=1
     AnswerCount=0 AuthorityCount=0 AddressRecCount=0 QuestRecs: Name=
     WARNING: Short packet. Try increasing the snap length
      (ttl 127, id 34806, len 78)
     0x0000    4500 004e 87f6 0000 7f11 e14b xxxx 6118   E..N.......K..a.
     0x0010    3db3 7082 0089 0089 003a 4e17 9fba 0000   =.p.......:N.....
     0x0020    0001 0000 0000 0000 2043 4b41 4141 4141   ........CKAAAAA
     0x0030    4141 4141 4141                            AAAAAA
[5]  17:43:06.530633 61.179.112.130 > MY.NET.97.24: icmp: 61.179.112.130 udp port 137
     unreachable [tos 0xc0]  (ttl 238, id 6793, len 56)
     0x0000    45c0 0038 1a89 0000 ee01 df1e 3db3 7082   E..8........=.p.
     0x0010    xxxx 6118 0303 ad99 0000 0000 4500 004e   ..a.........E..N
     0x0020    87f6 0000 6611 fa4b xxxx 6118 3db3 7082   ....f..K..a.=.p.
     0x0030    0089 0089 003a                            .....:
[6]  17:43:07.281844 61.179.112.130.34562 > MY.NET.97.10.80: S [tcp sum ok]
     679204100:679204100(0) win 16384 <mss 1402,nop,nop,sackOK> (DF) (ttl 110, id 3948, len
     48)
     0x0000    4500 0030 0f6c 4000 6e06 2b0d 3db3 7082   E..0.l@.n.+.=.p.
     0x0010    xxxx 610a 8702 0050 287b d504 0000 0000   ..a....P({......
     0x0020    7002 4000 ec37 0000 0204 057a 0101 0402   p.@..7.....z....
[7]  17:43:07.649721 MY.NET.97.24.137 > 61.179.112.130.137:  >>> NBT UDP Pkt(137): Query;
     REQ; UNICAST TrnID=0x9FBC OpCode=0 NmFlags=0x0 Rcode=0 QueryCount=1
     AnswerCount=0 AuthorityCount=0 AddressRecCount=0 QuestRecs: Name=
     WARNING: Short packet. Try increasing the snap length
      (ttl 127, id 35062, len 78)
     0x0000    4500 004e 88f6 0000 7f11 e04b xxxx 6118   E..N.......K..a.
     0x0010    3db3 7082 0089 0089 003a 4e15 9fbc 0000   =.p.......:N.....
     0x0020    0001 0000 0000 0000 2043 4b41 4141 4141   ........CKAAAAA
     0x0030    4141 4141 4141                            AAAAAA
[8]  17:43:08.035975 61.179.112.130 > MY.NET.97.24: icmp: 61.179.112.130 udp port 137
     unreachable [tos 0xc0]  (ttl 238, id 6794, len 56)
```

```
        0x0000    45c0 0038 1a8a 0000 ee01 df1d 3db3 7082   E..8.....=.p.
        0x0010    xxxx 6118 0303 ad9b 0000 0000 4500 004e   ..a.........E..N
        0x0020    88f6 0000 6611 f94b xxxx 6118 3db3 7082   ....f..K..a.=.p.
        0x0030    0089 0089 003a                            .....:
[9]  17:44:15.326826 192.9.100.88 > MY.NET.97.10: (frag 0:20@60824+) (ttl 237, len 40)
        0x0000    4500 0028 0000 3db3 ed06 47a1 c009 6458   E..(..=...G...dX
        0x0010    xxxx 610a 0d5e 0050 0172 b8d8 0172 b8d8   ..a..^.P.r...r..
        0x0020    8504 0000 b021 0000 2100 0000 2c00        .....!..!...,.
```

Note: Please ignores the last bytes of the packet smaller than 46 bytes. A tcpdump bug will display part of the Ethernet trailer if the IP packet is smaller than 46 bytes (minimum Ethernet frame size is 64 bytes).

Source of Trace

This trace was captured on a enterprise Internet access using a class C subnet. The capture date is September 24, 2002.

Detect was generated by

A SHADOW IDS installed to get interesting traffic for the practical assignment captured this trace. SHADOW uses tcpdump to record the packets and the default data size captured is 68 bytes (Ethernet header included). The hexadecimal dump displayed in the analysis below may have been truncated due to this size limit. The traces are in the tcpdump format.

Probability the source address was spoofed

The source address of the last packet is certainly spoofed. This address is officially allocated to Sun Microsystems according to the ARIN database. The source address of this IP fragment belongs to a non routed network on Internet. The output of a "looking glass" server, which is a router where some display commands are allowed directly or through a web server, is displayed below:

"Results of query:
Router: Cernh3.cern.ch
Command: show ip bgp 192.9.100.88

% Network not in table"

In addition, the attack description shows evidence of packet forging. The packets from MY.NET.97.24 are outgoing from my network with the right source network, so we may expect the source is not spoofed. About the packets from 61.179.112.130, the source is certainly not spoofed. The TTL value of the packet to/from this host and the ICMP messages are comforting this opinion.

Description of attack

The attack itself is the last packet only, but the previous packets have been included in an attempt to localize the source of the attack.

The first packet is a connection attempt to the HTTP port on an internal host. The second packet is the well-known NetBIOS name table retrieval query, also called NetBIOS wildcard query. Windows operating systems uses it in normal operation and may also be used for reconnaissance to get NetBIOS information.
In this detect, one internal host initiates NetBIOS queries. This internal host MY.NET.97.24 has been identified. After a quick look at the applications running, the program responsible of these NetBIOS query was found. A Syslog server was running and receiving the log of the firewall. This syslog server has an option to resolve IP addresses received to hostnames. Depending on the configuration, Windows systems will do the name query using first DNS then NetBIOS or first NetBIOS and then DNS. The outgoing NetBIOS name queries are explained hereafter. For each outgoing name query, a ICMP port unreachable is received as response.

Using passive OS fingerprinting, the remote OS may be guessed based on the default value used in the protocol headers. The TTL field initial value is dependent on the operating system, but the most current values are not close to each other: 64/128/255. For the incoming TCP SYN to the HTTP port (packet [1] and [6]), the interesting header fields are: TTL=110 => initial TTL=128, TCP window=16384, TCP options=mss, SackOK and two nops. The packet length is 48 bytes. Windows 2000 computers use all these default fields. The TCP mss (maximum segment size) of 1402 could be due to an IPSec tunnel. The TCP mss is 40 bytes below the smallest MTU on the path (with no IP or TCP options). With a TCP mss of 1402, the path MTU should be 1442. Where are the missing 58 bytes to reach 1500 ? Actually, they correspond to the maximum overhead for an IPSec tunnel using ESP encryption.
For the incoming ICMP port unreachable, the interesting header fields are: TTL=238 => initial TTL=255, TOS=192. There are two OS using TTL of 255; Solaris 2.x and IOS 12.x but the TOS of 192 is most likely used by the Cisco IOS. The last packet is the IP fragment that will be analyzed below.

Attack mechanism

From the hex of the last packet, the header gives its secrets. Only the interesting fields are described:
Total length: 40 bytes
IP ID: 0
More Fragment bit set and fragment offset = 60824
TTL: 237
Protocol: TCP
Port source: 3422
Port destination: 80
TCP header length: 8 x 32 bit words = 32 bytes
Reserved bits: 010100

TCP flag: RST
TCP window: 0

There are three interesting things in this packet. First, the IP fragmentation is a 20-bytes packet that should be put in the packet buffer of the target at the offset 60824. A single packet may not cause denial of service, the OS will drop the incomplete packet after a timeout. In this case, an ICMP IP reassembly time exceeded will be turned. The timeout value depends on the OS installed and therefore may be used for fingerprinting. Second, the packet length is 40 bytes, but the TCP header length should be in this case 20 bytes. The TCP header length specifies 12 additional bytes for the TCP options that does not exist. Third, the initial TTL is certainly 255 and so the hop count is 18. The hop counts computed for the packets received from 61.179.112.130 are 18 for the TCP SYN and 17 for the ICMP port unreachable. The shorter hop count of the ICMP port unreachable may be due to a firewall protecting the attacker.

The header analysis and the passive OS fingerprinting give this possible scenario: an attacker using a Windows 2000 OS is doing reconnaissance using connection attempts on TCP port 80. Because the firewall filters the incoming port 80 requests to this target, the packet drop generates a log entry. The internal syslog server, used for the firewall, tries to resolve the attackers IP addresses to hostnames and, in last resort, Windows uses the unicast NetBIOS query. The attacker filters the request to the NetBIOS port and an ICMP port unreachable packets are responded. The attacker tried a last reconnaissance with the fragmented packet. This fragmented packet could elicit an ICMP ip reassembly time exceeded from the target.

Correlations

The source address 192.9.100.88 is in the DShield.org database. The database contains 195 records against this IP address all between September 28th and 29th 2002. The analyzed detect was recorded on September 24th 2002. The IP fragment was certainly generated with a toll like hping2 using the parameters "—count 1 –spoofhostname 192.9.100.88 –ttl 255 –id 0 –frag 40 –morefrag –fragoff 60824 –destport 80 –win 0 –tcpoff 8 –rst". Other tools like Teardrop, Newdrop or Syndrop may produce a similar trace but they use two packets to attempt an OS crash. The IP address 61.179.112.130 is allocated to CHINANET Shandong province network according to the APNIC database:

```
inetnum:      61.179.0.0 - 61.179.255.255
netname:       CHINANET-SD
descr:        CHINANET Shandong province network
descr:        Data Communication Division
descr:        China Telecom
country:       CN
admin-c:       CH93-AP
tech-c:       XZ14-AP
mnt-by:        MAINT-CHINANET
mnt-lower:    MAINT-ZXF
changed:       hostmaster@ns.chinanet.cn.net 20010108
```

status: ALLOCATED PORTABLE
source: APNIC

person: Chinanet Hostmaster
address: No.31 ,jingrong street,beijing
address: 100032
country: CN
phone: +86-10-66027112
fax-no: +86-10-66027334
e-mail: hostmaster@ns.chinanet.cn.net
nic-hdl: CH93-AP
mnt-by: MAINT-CHINANET
changed: hostmaster@ns.chinanet.cn.net 20020814
source: APNIC

person: XIAOFENG ZHANG
address: Shandong Public Information Service Bureau
address: No.77 Jingsan Road,Jinan,Shandong P.R China
country: CN
phone: +86-531-6052163
fax-no: +86-531-6052414
e-mail: ip@pub.sd.cninfo.net
nic-hdl: XZ14-AP
mnt-by: MAINT-ZXF
changed: zxf@sdinfo.net 20001012
source: APNIC

However, if the IP fragment packet also come from China, the possible ICMP IP reassembly time exceeded response will not be routed to the attacker (network 192.9.100.0 is advertised on Internet). The purpose of this packet is therefore not clear in this case. The TTL, the timestamp and the target IP address of the received packets are matching. It might be a coincidence, but the probability is low. If the network 192.9.100.0 was advertised on Internet on September 24th 2002, the ICMP IP reassembly time exceeded message would have been routed to a host. This is the only way to get a response and to use the last packet for reconnaissance. The route for 192.9.100.0 network was maybe present and advertised by the official owner on September 24th 2002. One of their hosts was compromised and was recording the ICMP response. Another possibility is to inject this network on Internet through a compromised ISP router and route it to the attacker host. The above explanation concerning the route 192.9.100.0 is only a supposition. However, it is the only explanation found for this last packet.

Evidence of active targeting

The TCP SYN packets targeting the port 80 and the IP fragment are both send to one address. No other packets from the same sources have been recorded by the SHADOW IDS before or after this detect. It is definitively an active targeting.

Severity

Criticality: 1
The target host is not identifiable because dynamic address translation is used on the firewall. The number of servers using Internet from an inside network is always lower than the number of workstation for an enterprise network. The target address was probably a user workstation.

Lethality: 0
No denial of service using offset 60824 has been found. The IP packet limit is 65535, a packet of 6024 + 20 bytes is below this limit. The attacker was most likely trying to get an ICMP IP reassembly time exceeded for reconnaissance.

System Countermeasures: 5
The inside hosts are running the latest software version and new patches are installed proactively.

Network Countermeasures: 4
Internal hosts are protected by a state-full firewall. The UDP port 137 outgoing traffic should however not be permitted.

Attack Severity: (1 + 0) - (5 + 4) = -8

Defensive recommendation

A statefull firewall is already in place. Some improvements should be performed to avoid unneeded outgoing traffic. The syslog server used for the firewall logs should be reconfigured to not resolve the IP addresses to hostnames. In addition, the firewall should deny outgoing NetBIOS traffic as well as others unneeded flows.

intrusions@incidents.org mailing list

This detect was posted to the mailing list on Friday 4th October, according to the practical assignment. Unfortunately, I was a bit late for the editing of this paper and no questions have been posted on the mailing list. However, I will answer all coming questions.

Multiple choice test question

What is the expected response to this packet ?

```
17:44:15.326826 192.9.100.88 > MY.NET.97.10: (frag 0:20@60824+) (ttl 237, len 40)
0x0000  4500 0028 0000 3db3 ed06 47a1 c009 6458   E..(..=...G...dX
0x0010  xxxx 610a 0d5e 0050 0172 b8d8 0172 b8d8   ..a..^.P.r...r..
0x0020  8504 0000 b021 0000 2100 0000 2c00        .....!..!...,.
```

A. An ICMP Fragmentation needed & DF bit set

B. An ICMP ip reassembly time exceeded
C. An ICMP port 80 unreachable
D. No response is expected

Answer is B.
This kind of packets is used to elicit an ICMP reassembly time exceeded from the target for reconnaissance.

References

Iron Code "Denial of Service Attacks" in Phreedom Magazine issue 21, 1999
URL: http://old.phreedom.org/en/issues/Phm21%20-%20IronCode%20-%20Denial%20of%20Service%20Attacks.txt (Oct 1, 2002)

Toby Miller "Passive OS Fingerprinting: Details and Techniques"
URL: http://www.incidents.org/papers/OSfingerprinting.php (Oct 1, 2002)

Lance Spitzner "Lists of fingerprints for passive fingerprint monitoring" May 23, 2000
URL: http://project.honeynet.org/papers/finger/traces.txt (Oct 1, 2002)

Cisco Systems "IP Fragmentation and PMTUD" Sep 11, 2002
URL: http://www.cisco.com/warp/public/105/pmtud_ipfrag.html (Oct 1, 2002)

ARIN, American Registry for Internet Numbers "Whois tool"
URL: http://ws.arin.net/cgi-bin/whois.pl (Oct 1, 2002)

APNIC, Asia Pacific Network Information Center "Whois tool"
URL: http://www.apnic.net/apnic-bin/whois2.pl (Oct 1, 2002)

DShield.org "IP Info" for 192.9.100.88 Oct1, 2002
URL: http://www.dshield.org/ipinfo.php?ip=192.9.100.88 (Oct 1, 2002)

CERN-CIXP looking glass
URL: http://dxmon.cern.ch/cgi-bin/lg.cgi (Oct 1, 2002)

Richard Sharpe "What is SMB ?" Sep 27, 1999
URL: http://samba.anu.edu.au/cifs/docs/what-is-smb.html (Oct 1, 2002)

Ofir Arkin "Nuisance with small (<46 bytes) IP packets and tcpdump" May 28, 2002
URL: http://www.sys-security.com/archive/bugtraq/ofirarkin2002-03.txt (Oct 1, 2002)

HPING command-line oriented TCP/IP packet assembler/analyzer
URL: http://www.hping.org/ (Oct 1, 2002)

Judy Novak, Stephen Northcutt, Mike Bost, Hal Pomeranz, Jean Triquet, Bill Ralph, Earl Carter, and Bryce Alexander.
SANS Institute "Intrusion Detection In-Depth" courseware 2002

W. Richard Stevens "TCP/IP Illustrated, Volume 1 The Protocols" 20th printing 2001
ISBN 0201633469

TCPDump.org "TCPDUMP public repository" Jun 5, 2002
URL: http://www.tcpdump.org (Aug 22, 2002).

Detect #3: Crafted NetBIOS query

Trace Log

```
22:04:55.446351 195.186.175.109.500 > MY.NET.97.26.137: [|isakmp] (ttl 111, id 29180, len 78)
0x0000  4500 004e 71fc 0000 6f11 ce85 c3ba af6d      E..Nq...o......m
0x0010  xxxx 611a 01f4 0089 003a db33 d872 0000      ..a......:.3.r..
0x0020  0001 0000 0000 0000 2043 4b41 4141 4141  ........CKAAAAA
0x0030  4141 4141 4141                           AAAAAA
22:04:56.964898 195.186.175.109.500 > MY.NET.97.26.137: [|isakmp] (ttl 111, id 39420, len 78)
0x0000  4500 004e 99fc 0000 6f11 a685 c3ba af6d      E..N....o......m
0x0010  xxxx 611a 01f4 0089 003a db2f d876 0000      ..a......:./.v..
0x0020  0001 0000 0000 0000 2043 4b41 4141 4141  ........CKAAAAA
0x0030  4141 4141 4141                           AAAAAA
22:04:58.433557 195.186.175.109.500 > MY.NET.97.26.137: [|isakmp] (ttl 111, id 49916, len 78)
0x0000  4500 004e c2fc 0000 6f11 7d85 c3ba af6d      E..N....o.}....m
0x0010  xxxx 611a 01f4 0089 003a db2d d878 0000      ..a......:.-.x..
0x0020  0001 0000 0000 0000 2043 4b41 4141 4141  ........CKAAAAA
0x0030  4141 4141 4141                           AAAAAA
```

Source of Trace

This trace was captured on an enterprise Internet access using a class C subnet. The capture date is September 22, 2002.

Detect was generated by

A Snort and a SHADOW IDS installed to get interesting traffic for the practical assignment captured this trace. SHADOW uses tcpdump to record the packets and Snort uses the tcpdump format, therefore the traces are in the tcpdump format.

Probability the source address was spoofed

The source address of the packets is certainly not spoofed. The detect is an attempted reconnaissance. The source address is, in most of the cases not spoofed to achieve reconnaissance.

Description of attack

UDP packets on port 137 are well-known to belong to the NetBIOS over IP traffic. The "CKAAAAA.." pattern is the very frequent NetBIOS SMB name wildcard query. However, the source and destination port used are always 137 in normal operation. This attack use the non-common source port 500 allocated to ISAKMP protocol. ISAKMP protocol is used mainly for the key exchange of IPSec. The attacker sending a NetBIOS SMB name wildcard query expect to receive the NetBIOS hostname, domain, user currently logged-in, MAC address and a list of some services running.

Attack mechanism

The particularity of this attack is the source port used. Some security perimeters allow ISAKMP traffic to internal hosts to permit IPSec VPN. This "new" form of the wildcard name query has been done to try to evade weakly configured access control lists. The incoming filter should mainly be based on destination port to block this kind of modified attack.

Correlations

The DShield.org database does not contain any records for the source IP address. No correlation has been found on the web for this specific modified attack with port 500. The "standard" NetBIOS SMB name wildcard attack using source and destination port =137 is a common attack described by a lot of research papers.
A simple "nbtstat" command on Windows will produce the "standard" trace. There are also a number of tools made to scan a network. No tool has been found that may produce the detect analyzed here. According to the RIPE database, the source address is allocated to:

```
inetnum:      195.186.96.0 - 195.186.255.255
netname:      BLUEWINNET
descr:        Bluewin is an internet service provider in CH.
country:      CH
admin-c:      PZ1009-RIPE
tech-c:       AM1626-RIPE
tech-c:       MR1192-RIPE
rev-srv:      dns1.bluewin.ch
rev-srv:      dns2.bluewin.ch
rev-srv:      dns3.bluewin.ch
rev-srv:      dns4.bluewin.ch
status:       ASSIGNED PA
remarks:      In case of hack attacks, spam, scans etc. please
remarks:      send abuse notifications to abuse@bluewin.ch
notify:       ipmaster@bluewin.ch
mnt-by:       RIPE-NCC-NONE-MNT
changed:      ipmaster@bluewin.ch 20020819
source:       RIPE

route:        195.186.0.0/16
descr:        The Blue window
descr:        Provider: Swisscom IP-Plus
origin:       AS3303
mnt-by:       CH-UNISOURCE-MNT
changed:      abuse@bluewin.ch 20000305
source:       RIPE

person:       Peter Zollinger
```

address: Bluewin AG
address: Hardturmstrasse 3
address: P.O. Box 756
address: CH-8037 Zurich
phone: +41 1 274 7111
fax-no: +41 1 274 7499
e-mail: ipmaster@bluewin.ch
nic-hdl: PZ1009-RIPE
notify: ipmaster@bluewin.ch
changed: ipmaster@bluewin.ch 20010528
source: RIPE

[snipped for brevity]

Evidence of active targeting

These special NetBIOS name queries targeted one host. No network activity from the source has been seen before or after this detect. It is a stimulus targeting a host for reconnaissance. Therefore, we are not in presence of active targeting.

Severity

Criticality: 1
The target hosts are not identifiable because dynamic address translation is used on the firewall. The number of servers using Internet from an inside network is always lower than the number of workstations for an enterprise network. The target address was probably a user workstation.

Lethality: 4
This detect is an attempted reconnaissance. The information received by the attacker in case of success is highly valuable for him/her.

System Countermeasures: 5
The inside hosts are running the latest software version and new patches were installed proactively.

Network Countermeasures: 5
Internal hosts are protected by a state-full firewall. The NetBIOS over IP traffic is filtered in inbound and outbound based on destination port.

Attack Severity: (1 + 4) - (5 + 5) = -5

Defensive recommendation

A statefull firewall is protecting the inside network and a specific care has been

taken to filter NetBIOS over IP traffic. This new NetBIOS SMB wildcard name query should be added in the IDS rules for monitoring.

intrusions@incidents.org mailing list

This detect was posted to the mailing list on Friday 4th October, according to the practical assignment. Unfortunately, I was a bit late for the editing of this paper and no questions have been posted on the mailing list. However, I will answer all coming questions.

Multiple choice test question

Which statement best describe the following traffic ?

22:04:55.446351 195.186.175.109.500 > MY.NET.97.26.137: [|isakmp] (ttl 111, id 29180, len 78)
22:04:56.964898 195.186.175.109.500 > MY.NET.97.26.137: [|isakmp] (ttl 111, id 39420, len 78)
22:04:58.433557 195.186.175.109.500 > MY.NET.97.26.137: [|isakmp] (ttl 111, id 49916, len 78)

A. It is a NetBIOS SMB name wildcard query with a source port of 500
B. It is response to a ISAKMP stimulus
C. This trace is not an attack, Windows computer may use port 137 for ISAKMP
D. It is a simple TCP retransmission

Answer is A.
ISAKMP protocol always uses source and destination ports equal to 500. The trace is UDP packet. The only possible answer is A.

References

ARIN, American Registry for Internet Numbers "Whois tool"
URL: http://ws.arin.net/cgi-bin/whois.pl (Oct 1, 2002)

APNIC, Asia Pacific Network Information Center "Whois tool"
URL: http://www.apnic.net/apnic-bin/whois2.pl (Oct 1, 2002)

RIPE, Réseaux IP Européens "Query the Ripe Whois Database"
URL: http://www.ripe.net/perl/whois (Oct 1, 2002)

DShield.org "IP Info" for 203.117.224.22 Oct1, 2002
URL: http://www.dshield.org/ipinfo.php?ip=203.117.224.22 (Oct 1, 2002)

Bryce Alexander "ID FAQ – Port 137 Scan" May 10, 2000
URL: http://www.sans.org/newlook/resources/IDFAQ/port_137.htm (Oct 1, 2002)

Judy Novak, Stephen Northcutt, Mike Bost, Hal Pomeranz, Jean Triquet, Bill Ralph, Earl Carter, and Bryce Alexander.
SANS Institute "Intrusion Detection In-Depth" courseware 2002

W. Richard Stevens "TCP/IP Illustrated, Volume 1 The Protocols" 20th printing 2001
ISBN 0201633469

Naval Surface Warfare Center - Dahlgren Lab "NSWC SHADOW INDEX"
URL: http://www.nswc.navy.mil/ISSEC/CID/ (Aug 22, 2002)

Martin Roesch "Snort User Manual Snort Release: 1.9.x" Apr 26, 2002
URL: http://www.snort.org/docs/SnortUsersManual.pdf (Aug 30,2002)

TCPDump.org "TCPDUMP public repository" Jun 5, 2002
URL: http://www.tcpdump.org (Aug 22, 2002).

Assignment 3 : "Analyze This" Scenario

Executive Summary

The following report was performed for the University by analyzing the security alerts over a five days period. A Snort IDS of an unknown version, running a fairly standard rule set generated these alerts. This security audit aims to identify compromised systems, intrusions and possible host or network configuration problem. Some critical security issues where found and will require immediate actions.

Data Files Analyzed

The data analyzed have been recorded between August 20th to August 24th 2002. Due to an issue with the captured logs on http://www.incidents.org/logs/ , the out of specification logs were not available since August 6th 2002 (at least at the time I write these lines). The last available OOS logs have been used according to GIAC Graders.

To get a whole picture of the trends in alert traffic, the five days logs were merge to be analyzed. SnortSnarf was used to index the alert logs. To be bale to use SnortSnarf, the "MY.NET." was replaced with two valid numbers. The portscan, OOS and some part of the alert logs have been sorted and extracted using small perl scripts. The format of the alert logs seems to be the old Snort format (before version 1.8). I had some problem with SnortSnarf and this old log format and I used a perl script to change the format.

Snort alert files:

Filename	Size
alert.02.08.20	16'890'422
alert.02.08.21	12'969'935
alert.02.08.22	9'076'974
alert.02.08.23	29'763'400
alert.02.08.24	40'293'329

The full raw scan data is provided in the files below. The portscan preprocessor alerts are redundant with the raw scan files. Therefore, the Snort's portscan preprocessor message were ignored in the alert files.

Snort scan files:

Filename	Size
scans.020820	40'846'514
scans.020821	29'464'535
scans.020822	16'572'491
scans.020823	77'548'351
scans.020824	119'005'277

Snort out of spec. files:

Filename	Size
oos_Aug.1.2002	2'004
oos_Aug.2.2002	316'023
oos_Aug.3.2002	141'478
oos_Aug.4.2002	278
oos_Aug.5.2002	261

Network and host profile

Hosts	Service	Hosts	Service
MY.NET.1.201	HTTP	MY.NET.70.14	HTTP
MY.NET.1.204	HTTP	MY.NET.70.164	HTTP
MY.NET.1.205	HTTP	MY.NET.70.170	HTTP
MY.NET.5.14	HTTP	MY.NET.70.172	HTTP
MY.NET.5.15	HTTP	MY.NET.70.18	HTTP
MY.NET.5.55	HTTP	MY.NET.70.183	HTTP
MY.NET.5.92	HTTP	MY.NET.70.191	HTTP
MY.NET.5.95	HTTP	MY.NET.70.205	HTTP
MY.NET.6.34	SMTP	MY.NET.70.225	HTTP
MY.NET.6.35	SMTP	MY.NET.70.231	HTTP
MY.NET.6.35	SMTP	MY.NET.70.41	HTTP
MY.NET.6.40	SMTP, HTTP	MY.NET.70.69	HTTP, FTP, SSH, sunrpc, SMTP, netbios-ssn, 445
MY.NET.6.40	SMTP	MY.NET.70.76	HTTP
MY.NET.6.47	SMTP	MY.NET.70.79	HTTP
MY.NET.6.47	SMTP	MY.NET.80.144	HTTP
MY.NET.6.7	SMTP, POP3	MY.NET.81.6	HTTP
MY.NET.9.9	HTTP	MY.NET.82.135	HTTP
MY.NET.10.17	HTTP	MY.NET.82.2	HTTP
MY.NET.10.179	HTTP	MY.NET.91.0	HTTP
MY.NET.10.24	HTTP	MY.NET.91.104	HTTP
MY.NET.10.32	HTTP	MY.NET.91.154	HTTP
MY.NET.17.13	HTTP	MY.NET.91.156	HTTP
MY.NET.17.14	HTTP	MY.NET.91.8	HTTP
MY.NET.17.15	HTTP	MY.NET.99.122	HTTP
MY.NET.17.16	HTTP	MY.NET.99.133	HTTP
MY.NET.17.2	HTTP	MY.NET.99.174	HTTP
MY.NET.17.3	HTTP	MY.NET.99.42	HTTP
MY.NET.17.4	HTTP	MY.NET.99.43	HTTP
MY.NET.18.18	HTTP	MY.NET.100.217	SMTP, HTTP
MY.NET.18.44	HTTP	MY.NET.100.230	SMTP
MY.NET.18.45	HTTP	MY.NET.139.230	SMTP
MY.NET.18.46	HTTP	MY.NET.157.52	HTTP

MY.NET.29.3	HTTP	MY.NET.179.77	HTTP
MY.NET.30.66	HTTP	MY.NET.179.80	SMTP
MY.NET.40.11	HTTP	MY.NET.253.10	sunrpc
MY.NET.53.228	HTTP	MY.NET.253.12	sunrpc
MY.NET.53.229	HTTP	MY.NET.253.13	sunrpc
MY.NET.53.84	HTTP	MY.NET.253.15	sunrpc
MY.NET.60.10	TFTP	MY.NET.253.17	sunrpc
MY.NET.60.23	HTTP	MY.NET.253.20	FTP
MY.NET.70.103	HTTP	MY.NET.253.41	SMTP
MY.NET.70.113	HTTP	MY.NET.253.43	SMTP

This above list is far from complete. There is more than 500 hosts running a web server (network devices included, printers, …). The subnet that appear to be dedicated for students have not been included for brevity.

Top 10 talkers lists - Alerts

Internal hosts

Top 10 source	Alerts
MY.NET.70.69	20098
MY.NET.116.44	19068
MY.NET.157.239	8200
MY.NET.152.20	7107
MY.NET.91.104	6118
MY.NET.111.226	5675
MY.NET.153.168	5054
MY.NET.85.74	4176
MY.NET.88.106	3648
MY.NET.84.166	2609

Top 10 destination	Alerts
MY.NET.84.147	13433
MY.NET.113.4	9828
MY.NET.87.185	4459
MY.NET.109.104	2431
MY.NET.153.168	2389
MY.NET.70.69	2192
MY.NET.70.113	2151
MY.NET.198.204	1989
MY.NET.100.236	1964
MY.NET.153.185	1729

External hosts

Top 10 source	Alerts
212.179.35.118	14390
212.179.105.146	12583
207.190.151.187	10276
202.106.148.57	6191
200.204.93.238	5566
212.179.32.138	4414
203.148.240.66	4065
212.179.96.242	4038
203.253.128.146	3698
130.126.125.156	3642

Top 10 destination	Alerts
192.244.23.3	50781
202.216.248.253	31372
216.241.219.28	26966
210.162.245.67	22807
62.211.24.31	20333
24.203.199.76	8166
194.87.5.53	5848
198.137.240.91	5679
61.119.9.219	5575
218.43.163.165	5503

Alert events of interest

To analyze the provided logs, the most frequent events have been extracted and will be described in this section. The IDS installed at the University is generating a lot of alerts, certainly due to lack of configuration of the rule base. The alerts analyzed here are due to noisy rules and should be tuned to increase the value of the IDS in place.

Alerts recorded more than 5'000 times

Number of alert	Alert name
107'808	DDOS shaft synflood outgoing
57'200	spp_http_decode: IIS Unicode attack detected
45'059	Watchlist 000220 IL-ISDNNET-990517
38'045	spp_http_decode: CGI Null Byte attack detected
33'545	TFTP - Internal TCP connection to external tftp server
28'577	External RPC call
22'533	SMB Name Wildcard
22'155	beetle.ucs
9'615	Incomplete Packet Fragments Discarded
5'767	Possible trojan server activity
5'679	TCP White House Traffic
5'565	SYN-FIN scan!

DDOS shaft synflood outgoing 107'808 alerts
Severity: high

These alerts show internal hosts participating in a distributed denial of service attack. False positives are not current because the TCP sequence number is fix in the SYN flood used by the shaft DDOS. A total of 254 hosts have been compromised. The list of attacked systems from the University network is:

Targeted systems	SYN sent	# Internal hosts
192.244.23.3	34007	253
202.216.248.253	20985	253
210.162.245.67	15286	253
62.211.24.31	13651	253
194.87.5.53	3966	253
61.119.9.219	3760	253
218.43.163.165	3654	253
210.137.62.3	2659	253
210.254.151.98	1703	251
210.149.120.77	1608	249
133.72.1.88	1586	249
61.122.28.235	1549	247
155.207.19.202	995	230
218.43.0.65	976	230
218.43.14.208	954	231
62.3.73.249	469	161

The list of compromised internal hosts is provided in appendix D.

Correlation: CVE: CAN-2000-0138, arachNIDS IDS253
An Analysis of the Shaft Distributed Denial of Service Tool by Sven Dietrich, Neil Long and David Dittrich

Recommended action: As an emergency workaround, filter this traffic in the security perimeter (on the firewall or on the router). A system administrator have to visit all these hosts as soon as possible to remove the DDOS code and install the latest patches.

spp_http_decode: IIS Unicode attack detected 57'200 alerts
Severity: high

The unicode translation vulnerability of IIS is used by worms like Code Red, Code Red II, Nimda, … to go out of the normal IIS directory. The alerts concerning inbound traffic have to be monitored. The response cannot be seen in the logs. However, all the alerts concerning outgoing traffic are generated by infected internal hosts. A total of 240 internal hosts are compromised and actively targeting other hosts both inside and outside the University. The list of compromised hosts is provided in appendix C.

Correlation: Tod Beardsley noticed 76 compromised hosts in his GCIA practical assignment.

Recommended action: Apply in emergency a filter in the security perimeter (on the firewall or on the router) to stop the spreading of the worms. A system administrator have to visit all these hosts as soon as possible to remove the worm(s) and install the latest IIS and system patches.

Watchlist 000220 IL-ISDNNET-990517 45'059 alerts
Severity: low

The rule generating this alert match the inbound traffic from the network 212.179.0.0/16 allocated to ISDN Net Ltd, Israel. The top 2 hosts are 212.179.35.118 and 212.179.105.146 with a total of 26'951 alerts. Scans to many different ports have been tired, like Kazaa (port 1214), Oraclenames (port 1575), …

Correlation: Alex Stephens also notice this alert in his GCIA practical assignment.

Recommended action: Keep it in the watchlist as the scans are recurrent.

spp_http_decode: CGI Null Byte attack detected 38'045 alerts
Severity: high

These alerts monitor possible use of a null byte in the CGI stream to evade the security check on the server. However, the web browsing from internal host triggered all these alerts. The browsing to the web server 216.241.219.28 and 216.241.219.14 is the main part with a total of 31'302 alerts. No alerts was triggered by external host. It is certainly false positives.

Correlation: Bradley Urwiller analyzed this alert in his GCIA practical assignment.

Recommended action: Tune the rule to minimize the number of false positives.

TFTP - Internal TCP connection to external tftp server 33'545 alerts
Severity: High

Trivial FTP service use UDP protocol, TCP is generally not used. The Slapper worm and its different versions use IRC to download files. The Slapper versions known today, seems to only use few IRC server specified in the code. However, quite all the internal hosts in the MY.NET.70.0/24 subnet are triggering this alert. The TFTP traffic is targeting a lot of different TFTP servers (not part of the list used by the known Slapper version today).

Correlation: No correlation about TFTP TFP traffic was found. The protocol used by the Slapper worm is not explicitly described in the worm analysis available. The Slapper worm certainly use UDP for TFTP transfer. This TFTP/TCP traffic may be a new worm version or another malicious code.

Recommended action: Investigation is needed on the internal hosts to find the source of this traffic. Information about the Slapper worm may be found in the references.

External RPC call 28'577 alerts
Severity: medium

These alerts are triggered due to connection attempt to portmapper service on internal hosts. This is a reconnaissance attempt to list the service offered on a host and the ports used. After this reconnaissance, RPC connections may be attempted. The attack following this reconnaissance will be analyzed in the "Attempted Sun RPC high port access" alert description.

Correlation: Mark Menke describe already this alert in his GCIA practical assignment.

Recommended action: No specific action is required following these alerts. However, check the "Attempted Sun RPC high port access" alert description for more information about the possible use of this reconnaissance.

SMB Name Wildcard 22'533 alerts
Severity: low

SMB Name Wildcard queries are used in normal operation by hosts using SMB protocols like Unix hosts with SAMBA and Windows hosts. No alerts have been triggered by internal hosts. The defensive perimeter is blocking outgoing request or the IDS rule has been tuned to detect incoming request only:

alert udp $EXTERNAL_NET any -> $HOME_NET 137 (msg:"SMB Name Wildcard"; content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";)

This rule is well known to be noisy. However, three external hosts come out of the noise:

Host 65.189.18.177 1770 alerts (388 destination IPs)
Reverse DNS: dsl-65-189-18-177.telocity.com

Host 65.185.217.185 1707 alerts (4 destination IPs)
Reverse DNS: dsl-65-185-217-185.telocity.com

Host 204.96.102.51 7263 alerts (233 destination IPs)
Reverse DNS: 204-96-102-51.rev.worldwebcafe.com, but 204-96-102-51.rev.worldwebcafe.com doesn't resolve to anything.

The first two hosts are not the same DSL host connected at different time with different IP addresses. Both IP were logged at he same times and the traceroute show different source location (maybe Chicago and Atlanta according to the last router names).

Correlation: The IDS FAQ from Bryce Alexander describe this alert and the threat.

Recommended action: Investigation is needed if the University security perimeter permit inbound NetBIOS over IP traffic. In this case, blocking the outgoing NetBIOS over IP traffic is recommended.

beetle.ucs 22'155 alerts
Severity: low

This alerts is a custom alert matching the TCP traffic to beetle.ucs on any ports. The IP address of this host is MY.NET.70.69. This host seems to be used by the students to use shared device like a CD recorder.

Correlation: Jeff Zahr and Edward Peck in their GCIA practical assignments also describe this alert.

Recommended action: The custom rule used provide usage monitoring of this host. No specific security issue is monitored through this rule.

Incomplete Packet Fragments Discarded 9'615 alerts
Severity: low

This alert was generated mainly by four hosts, two internal and two external hosts:

Host MY.NET.157.239	8161 alerts
Host MY.NET.163.117	883 alerts
Host 211.233.22.104	480 alerts
Host 61.138.3.116	60 alerts

Correlation: John Jenkinson in his GCIA practical assignments also describe this alert.

Recommended action: Check the two internal hosts for network connection problem. Concerning the two external hosts, there is not enough information in the Snort "fast" alert log type to investigate further. A Snort "full" log type or a binary capture is needed to check if this traffic has some security issues.

Possible trojan server activity 5'767 alerts
Severity: high

These alerts are triggered by traffic on port 27374, which is well known to be used by the Ramen worm and Sub-7 trojan. A lot of external scans were recorded, however there are strong indication of the presence of one of these malware on the internal host MY.NET.89.147.

Correlation: Stan Hoffman in his GCIA practical assignments describe this alert.

Recommended action: Immediately take this host off the network and check what is generating this traffic to port 27374. If the presence of the Ramen worm is confirmed, follow the instruction provided by Dave Wreski. If the presence of the Sub-7 trojan is confirmed, follow the instruction provided by Hackguard.net.

TCP White House Traffic 5'679 alerts
Severity: high

This rule match the TCP traffic sent to 198.137.240.91 host which is part of the IP range allocated to the White House. This is the old www.whitehouse.gov IP address. The internal host MY.NET.111.226 triggered this rule 5'675 times. The Code Red worm include a denial of service payload against

Correlation: Joni Ramos and Tyler Schacht describe the Code Red worm behavior in their GCIA practical assignments.

Recommended action: Immediately take this host off the network and install an up to date antivirus software. Several guides and software are available to clean infected system. One of these is provided in the references section.

SYN-FIN scan! 5'565 alerts
Severity: low

All these alerts were triggered by the host 200.204.93.238 that did a SYN FIN scan for SSH service on port 22 from MY.NET.1.1 to MY.NET.199.252. The host 200.204.93.238 is located in Brazil according to the LACNIC database. DShield.org database show only few records for this source.

Correlation: Mark Menke describe already this SYN FIN scan in his GCIA practical assignment, however the source was not the same.

Recommended action: This is a noisy reconnaissance attempt, no special action is needed.

Attempted Sun RPC high port access 1'662 alerts
Severity: high

This alert commonly follow a portscan on port 111 used by portmapper. It means the targeted hosts responded to the 'rpcinfo' query. The attacker know on which ports the services are running.

Internal hosts	Alerts	Attackers
MY.NET.154.27	1647	260
MY.NET.117.25	8	3
MY.NET.162.65	4	2
MY.NET.163.113	2	1
MY.NET.117.18	1	1

Correlation: Mark Menke describe already this alert in his GCIA practical assignment.

Recommended action: A system administrator should investigate why these five internal hosts answered to the 'rpcinfo' and fix this issue. He should also check if the hosts have been already compromised.

Top 10 talkers lists - Portscans

Internal hosts

Top 10 source	Alerts
MY.NET.70.207	609974
MY.NET.70.200	323301
MY.NET.165.24	125524
MY.NET.137.18	39502
MY.NET.87.50	32089

Top 10 destination	Alerts
MY.NET.117.25	13440
MY.NET.40.11	1469
MY.NET.106.94	1395
MY.NET.106.93	1385
MY.NET.106.92	1270

MY.NET.137.7	30763
MY.NET.104.104	28428
MY.NET.70.31	27198
MY.NET.88.5	26696
MY.NET.83.146	26266

MY.NET.60.10	1195
MY.NET.106.91	1027
MY.NET.162.67	987
MY.NET.83.72	881
MY.NET.106.90	771

External hosts

Top 10 source	Alerts
217.231.210.103	30166
217.56.39.130	19502
24.209.227.253	16527
212.7.48.35	15131
211.92.142.1	14755
212.50.186.167	13151
211.112.21.156	12796
141.150.70.66	12542
212.157.170.5	12321
211.38.106.8	11806

Top 10 destination	Alerts
192.244.23.3	950738
210.162.245.67	448664
62.211.24.31	344853
202.216.248.253	220864
61.119.9.219	95832
218.43.163.165	95718
210.137.62.3	50992
210.254.151.98	32926
61.122.28.235	32920
210.149.120.77	32872

Portscan log file analysis

Top 10 destination ports

Scans	Destination port	Common use
329487	41170	Blubster P2P
186112	80	HTTP
178234	6257	WinMX P2P
108905	1433	Microsoft SQL Server
83802	21	FTP
55706	6346	Gnutella P2P
41918	111	SUN RPC
38212	53	DNS
34283	22	SSH
29019	139	NETBIOS Session Service

Top 10 source ports

Scans	Source port	Common use (as destination)
379221	12300	
312830	2747	fjippol-swrly
216555	12203	
179727	6257	WinMX P2P
21653	888	AccessBuilder/CDDB Protocol
18365	1207	Softwar trojan/MetaSage
16609	21	FTP
16434	7001	afs3-callback/EverQuest
15451	3005	Genius License Manager
13091	6970	RealPlayer

The top 10 source ports has been included in the report for information. It shows which source port is used for most frequent port scan. It may be specific to a tool signature or used to try to evade the security perimeter.

Port 41170 Blubster P2P 329'487 matches
Severity: -

The port 41170 is not known to be used by a malicious code. This port is the default port used by a peer-to-peer software called Blubster.

Recommended action: None, except if internal policy does not permit this activity.

Port 80 HTTP 186'112 matches
Severity: low

The port 80 is targeted by the "web" worms like Code Red, Nimda, Slapper, … The port 80 scans may be matched with the web alerts recorded.

Recommended action: None.

Port 6257 WinMX P2P 178'234 matches
Severity: -

The port 6257 is not known to be used by a malicious code. This port is the default port used by the WinMX peer-to-peer software.

Recommended action: None, except if internal policy does not permit this activity.

Port 1433 MS-SQL 108'905 matches
Severity: low

Microsoft SQL server have several known vulnerabilities. A worm called SQLSnake has been discovered on May 20th 2002.

Recommended action: Be sure your internal MS-SQL servers have been patched. More information can be found in the references: Incidents.org Handler's Diary about the SQLSnake.

Port 21 FTP 83'802 matches
Severity: low

There were some known FTP daemons vulnerabilities allowing denial of service attacks or exploits. Attackers may also search unrestricted FTP server to store malicious content.

Recommended action: No specific action needed, the FTP servers should be patched regularly as the other actively scanned services.

Port 6346 Gnutella P2P 55'706 matches
Severity: -

This port is the default used by a lot of Gnutella clients. Gnutella is a peer-to-peer protocol for file sharing.

Recommended action: None, except if internal policy does not permit this activity.

Port 111 SUN RPC 41'918 matches
Severity: low

The port 111 is commonly used by portmapper. A scan for this port, is followed by a reconnaissance query to find the service running. The description of the "External RPC call" and "Attempted Sun RPC high port access" alerts in this paper gives more information.

Recommended action: No specific action needed

Port 53 DNS 38'212 matches
Severity: medium

The DNS servers are one of the most important service of a network. DNS servers are one of the first host to be targeted by an attacker. Commonly used BIND DNS server has several vulnerability in his history.

Recommended action: Be sure your DNS servers are running the latest software release.

Port 22 SSH 34'283 matches
Severity: medium

The SSH Secure Shell is a remote control protocol widely used on unix platforms. Several exploits are known on some implementation like the popular OpenSSH software.

Recommended action: Be sure your hosts use a non-vulnerable software version.

Port 139 NetBIOS Session Service 29'019 matches
Severity: medium

NetBIOS Session Service is used for file and printer sharing with the SMB protocol. A "SMB name wildcard" query is often issued before a port 139 connection attempt.

Recommended action: The NetBIOS over IP ports (137, 138 and 139) should be blocked on the edge firewall or router.

Top 10 talkers lists – Out of Specification

The out of specification logs only contain inbound traffic.

Top 10 external sources	Alerts
68.32.126.64	652
62.76.241.129	345
209.116.70.75	214
212.35.180.17	83
65.210.154.210	48
213.250.44.19	29
61.132.74.239	18
202.155.91.142	18
209.132.232.101	18
211.154.85.159	17

Top 10 internal destinations	Alerts
MY.NET.6.7	660
MY.NET.97.217	241
MY.NET.97.238	104
MY.NET.100.217	95
MY.NET.253.20	85
MY.NET.111.198	54
MY.NET.100.165	43
MY.NET.253.125	41
MY.NET.253.114	37
MY.NET.6.40	34

Out of Specification events of interest

Host 68.32.126.64 652 matches
Severity: -

```
08/01-00:03:02.100571 68.32.126.64:26052 -> MY.NET.6.7:110
TCP TTL:48 TOS:0x0 ID:432  DF
21S***** Seq: 0x1D18A45   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 51874805 0 EOL EOL EOL EOL

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

All the alerts triggered by this host are concerning the same destination host and port and do not have a different pattern. This host seems to use TCP Explicit Congestion notification. The bits Congestion Window Reduced (CWR) and Explicit Congestion Notification (ECN-Echo) are set. Some operating system support TCP ECN and may produce false positives.

Recommended action: No specific action needed

The others hosts in the top 10 OOS lists are all using TCP ECN except the last one which did some hostile activities.

Host 211.154.85.159 **17** matches
Severity: medium

```
08/01-03:20:12.727003 211.154.85.159:0 -> MY.NET.111.140:1663
TCP TTL:107 TOS:0x0 ID:9575  DF
21S**PAU Seq: 0x500170   Ack: 0x24010A75   Win: 0x8010
TCP Options => EOL EOL NOP NOP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

```
08/01-03:34:42.575935 211.154.85.159:0 -> MY.NET.111.140:1676
TCP TTL:107 TOS:0x0 ID:56950  DF
2*SF**A* Seq: 0x50017D   Ack: 0x95F442CC   Win: 0x8010
3C 53 80 10 B5 80 32 B5 00 00 01 01 05 0A 42 CC  <S....2.......B.
4D 57 42 CC                                      MWB.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-03:41:31.177676 211.154.85.159:1681 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:62599  DF
21S***A* Seq: 0x18388F1   Ack: 0x145B3C   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-03:41:52.984402 211.154.85.159:182 -> MY.NET.111.140:1681
TCP TTL:107 TOS:0x0 ID:45962  DF
2*SFRPA* Seq: 0x500183   Ack: 0x8E9B5B48   Win: 0x5010
00 B6 06 91 00 50 01 83 8E 9B 5B 48 07 5F 50 10  .....P....[H._P.
B5 80 E5 91 00 00 00 00 00 00                    ..........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-03:46:53.173239 211.154.85.159:1684 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:47505  DF
*1SFR*** Seq: 0x10187   Ack: 0xDE6E6D93   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-03:47:11.134688 211.154.85.159:0 -> MY.NET.111.140:1685
TCP TTL:107 TOS:0x0 ID:44435  DF
21*FRPAU Seq: 0x500188   Ack: 0x19436DB3   Win: 0x5010
B2 AD 08 AB 00 00 00 00 00 00                    ..........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-04:03:22.663985 211.154.85.159:1694 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:55472  DF
**SFRPAU Seq: 0x197   Ack: 0xF031AEE7   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK NOP NOP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-04:21:15.877198 211.154.85.159:1722 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:21989  DF
21S*R*AU Seq: 0x1A7   Ack: 0xC35BF035   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-05:19:16.649660 211.154.85.159:1754 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:1658  DF
21S***AU Seq: 0x1DB6FB5   Ack: 0xC5A0B5B1   Win: 0x5010
06 DA 00 50 01 DB 6F B5 C5 A0 B5 B1 00 F2 50 10  ...P..o.......P.
B5 80 EB 2B 00 00 00 00 00 00                    ...+......
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-05:19:16.920123 211.154.85.159:1753 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:3194
**SF***U Seq: 0x1DB1677   Ack: 0xF0C438   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-05:34:08.929013 211.154.85.159:1787 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:33153  DF
2*SF**** Seq: 0x1EA57B7   Ack: 0x3915778   Win: 0x5010
06 FB 00 50 01 EA 57 B7 03 91 57 78 00 43 50 10  ...P..W...Wx.CP.
B4 CD 23 F6 00 00 00 00 00 00                    ..#.......
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-05:38:52.122492 211.154.85.159:1798 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:53641  DF
2*SFR*A* Seq: 0x1EE7F29   Ack: 0x149BC537   Win: 0x5010
07 06 00 50 01 EE 7F 29 14 9B C5 37 00 57 50 10  ...P...)...7.WP.
```

```
B0 8C 81 EC 00 00 00 00 00 00                     ..........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-05:46:41.855061 211.154.85.159:0 -> MY.NET.111.140:1816
TCP TTL:107 TOS:0x0 ID:24217  DF
*1SF**AU Seq: 0x5001F5   Ack: 0x344D3196   Win: 0x5010
00 00 07 18 00 50 01 F5 34 4D 31 96 09 B3 50 10  .....P..4M1...P.
B5 80 66 45 00 00 00 00 00 00                     ..fE......
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-05:57:29.626865 211.154.85.159:20 -> MY.NET.111.140:1852
TCP TTL:107 TOS:0x0 ID:44972
21S***A* Seq: 0x5001FE   Ack: 0xE56958A0   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL WS: 1 NOP TS: 3604480 0 EOL EOL EOL EOL EOL
EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-06:13:00.731738 211.154.85.159:1893 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:59605  DF
21S*R*** Seq: 0x20DB060   Ack: 0x94F7   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-06:42:01.892932 211.154.85.159:1959 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:10526  DF
*1SF**A* Seq: 0x2281695   Ack: 0x2ADA817   Win: 0x5010
07 A7 00 50 02 28 16 95 02 AD A8 17 00 93 50 10  ...P.(........P.
B5 80 13 C0 00 00 00 00 00 00                     ..........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-06:49:03.188702 211.154.85.159:1975 -> MY.NET.111.140:80
TCP TTL:107 TOS:0x0 ID:46631  DF
21S**P** Seq: 0x22DC827   Ack: 0x1B1F351F   Win: 0x5010
07 B7 00 50 02 2D C8 27 1B 1F 35 1F 00 CA 50 10  ...P.-.'..5...P.
B5 80 BC 9E 00 00 00 00 00 00                     ..........
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This trace show evidence of reconnaissance and target a specific host. The attacker is doing active OS fingerprinting. According to APNIC database, the source address is allocated to Cable OnLine Network Xuhui2 pop., Shangai China.

Recommended action: Check the criticality of the destination host and check this hosts if appropriate.

Host 61.170.132.27 5 matches
Severity: medium

```
08/01-02:48:18.258649 61.170.132.27:1363 -> MY.NET.111.140:103
TCP TTL:46 TOS:0x0 ID:57131  DF
21*FR*** Seq: 0x500013   Ack: 0x318C9256   Win: 0xA010
TCP Options => EOL EOL NOP NOP
C2 D1                                             ..
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-02:48:20.657415 61.170.132.27:1365 -> MY.NET.111.140:80
TCP TTL:46 TOS:0x0 ID:17452  DF
*1SF*P*U Seq: 0x13318F   Ack: 0x91D70001   Win: 0x5010
00 00 00 00 00 00                                 ......
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-02:48:50.620950 61.170.132.27:1355 -> MY.NET.111.140:80
TCP TTL:46 TOS:0x0 ID:30512  DF
```

```
21*F**** Seq: 0x1332AC   Ack: 0x92690000   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-02:49:01.364194 61.170.132.27:1361 -> MY.NET.111.140:80
TCP TTL:46 TOS:0x0 ID:64561  DF
2*SF**** Seq: 0x1332B4   Ack: 0x91990001   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/01-02:49:05.227124 61.170.132.27:1356 -> MY.NET.111.140:80
TCP TTL:46 TOS:0x0 ID:37682  DF
21SFR*A* Seq: 0x1332AD   Ack: 0x92090000   Win: 0x5010
00 00 00 00 00 00                               ......
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This trace show evidence of reconnaissance and target a specific host. The attacker is doing active OS fingerprinting. According to APNIC database, the source address is allocated to Cable OnLine Network Xuhui2 pop., Shangai China.

Recommended action: Check the criticality of the destination host and check this hosts if appropriate.

Five external hosts

The registration information of the following five external hosts may be used for further investigation. The technical contacts may be noticed of the issue. The first two were chosen because the actively targeted the internal network for OS fingerprinting. The last three are part of the DDOS Shaft alerts, however they did not triggered the TCP TFTP alert.

211.154.85.159 – OS Fingerprinting

```
inetnum:     211.154.85.1 - 211.154.85.255
netname:     XUHUI2POPNET
descr:       Cable OnLine Network Xuhui2 pop.
descr:       Internet Service Provider
descr:       Shanghai China
country:     CN
admin-c:     HL6-CN
tech-c:      YM2-CN
mnt-by:      MAINT-CNNIC-AP
changed:     leion@cableplus.com.cn 20010615
status:      ASSIGNED NON-PORTABLE
source:      APNIC
changed:     hm-changed@apnic.net  20020827

person:      Huaiyu Li
address:     Computer Center
address:     Shanghai Cable TV Station
address:     487#, East Luo Chuan Road, Shanghai 200072, China
country:     CN
phone:       +86 21 56729282
e-mail:      fyma@shnet.edu.cn
nic-hdl:     HL6-CN
mnt-by:      MAINT-CN-CJJ
```

```
changed:    cjj@cableplus.com.cn 20010609
source:     APNIC

person:     Yougang Min
address:    Computer Center
address:    Shanghai Cable TV Station
address:    487#, East Luo Chuan Road, Shanghai 200072, China
phone:      +86 21 56729282
e-mail:     fyma@shnet.edu.cn
nic-hdl:    YM2-CN
mnt-by:     MAINT-CN-CJJ
changed:    cjj@cableplus.com.cn 20010611
source:     APNIC
```

61.170.132.27 – OS Fingerprinting

```
inetnum:    61.169.0.0 - 61.171.255.255
netname:    CHINANET-SH
descr:      CHINANET Shanghai province network
descr:      Data Communication Division
descr:      China Telecom
country:    CN
admin-c:    CH93-AP
tech-c:     XI5-AP
mnt-by:     MAINT-CHINANET
mnt-lower:  MAINT-CHINANET-SH
changed:    hostmaster@ns.chinanet.cn.net 20001201
status:     ALLOCATED PORTABLE
source:     APNIC

person:     Chinanet Hostmaster
address:    No.31 ,jingrong street,beijing
address:    100032
country:    CN
phone:      +86-10-66027112
fax-no:     +86-10-66027334
e-mail:     hostmaster@ns.chinanet.cn.net
nic-hdl:    CH93-AP
mnt-by:     MAINT-CHINANET
changed:    hostmaster@ns.chinanet.cn.net 20020814
source:     APNIC

person:     Wu Xiao Li
address:    Room 805,61 North Si Chuan Road,Shanghai,200085,PRC
country:    CN
phone:      +86-21-63630562
fax-no:     +86-21-63630566
e-mail:     ip-admin@mail.online.sh.cn
nic-hdl:    XI5-AP
mnt-by:     MAINT-CHINANET-SH
changed:    ip-admin@mail.online.sh.cn 20010510
source:     APNIC
```

133.72.1.88 – DDOS Shaft

The network is allocated by Japan NIC and unfortunately, the result of the whois is in Japanese. The remaining information was extracted.

Network Information:
a. [Network Number] 133.72.0.0
b. [Network Name] MIYAMO-NET
g. [Organization] Kanagawa University
m. [Administrative Contact] FM002JP
n. [Technical Contact] HM454JP
n. [Technical Contact] HS3321JP
n. [Technical Contact] KO437JP
n. [Technical Contact] SO006JP
p. [Nameserver] heaven.hiratsuka.kanagawa-u.ac.jp
p. [Nameserver] jingw.kanagawa-u.ac.jp
p. [Nameserver] ns4.ttnet.ad.jp
y. [Reply Mail] hiro@kanagawa-u.ac.jp
y. [Reply Mail] makino@kanagawa-u.ac.jp
y. [Reply Mail] ntn@kanagawa-u.ac.jp
[Assigned Date]
[Return Date]
[Last Update] 2002/03/22 16:33:05 (JST) hiro@kanagawa-u.ac.jp

210.254.151.98 – DDOS Shaft

The network is allocated by Japan NIC and unfortunately, the result of the whois is very small for this network.

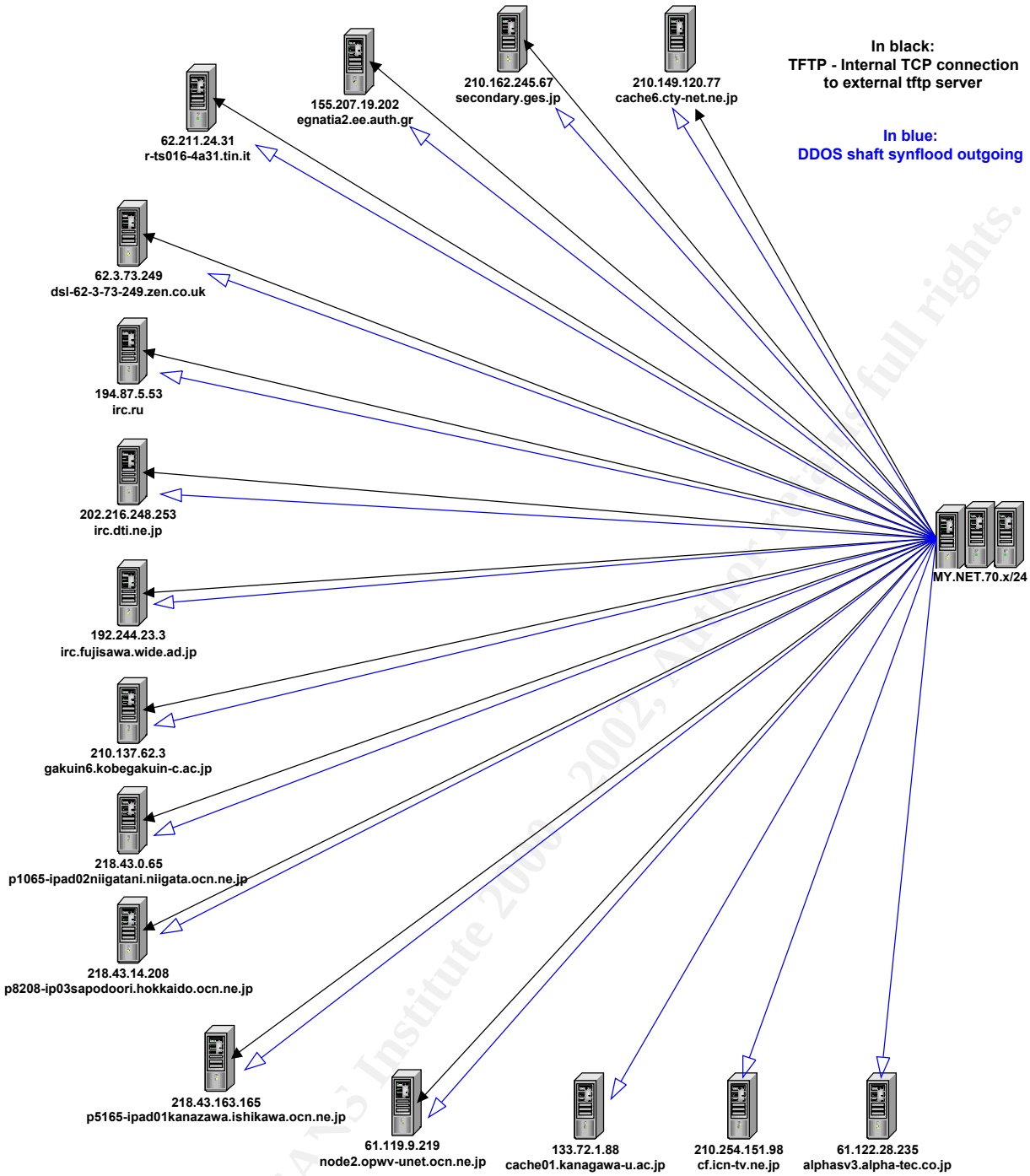
Open Computer Network
SUBA-131-K62 [Sub Allocation] 210.254.151.0
ICN Corporation
ICNTV-NET [210.254.151.96 <-> 210.254.151.127] 210.254.151.96/27

61.122.28.235 – DDOS Shaft

The network is allocated by Japan NIC and unfortunately, the result of the whois is very small for this network.

New Century GlobalNet Corporation
SUBA-475-A06 [Sub Allocation] 61.122.28.0
ALPHA TECHNOLOGY CO., LTD.
ALPHANET [61.122.28.232 <-> 61.122.28.239] 61.122.28.232/29

Link graph



From the alert log analysis, it was not clear that the "TFTP - Internal TCP connection to external tftp server" and "DDOS shaft synflood outgoing" alerts are related. However, no correlation has been found on the web. TCP connection on the TFTP port seems to be used now with a DDOS Shaft code.

Defensive recommendations and conclusions

This security audit aims to display the actual security level of the network and to list the actual weakness. The recommendations to improve the security are provided for each issues. It include a list of high severity problems that will need to be handled as soon as possible. The University network need to be more protected, the security perimeter of the Internet connection need strong improvements.

References

Bryce Alexander "ID FAQ – Port 137 Scan" May 10, 2000
URL: http://www.sans.org/newlook/resources/IDFAQ/port_137.htm (Oct 1, 2002)

Common Vulnerabilities and Exposures (CVE) "CAN-2000-0138" May 2, 2000
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138 (Oct 1,2002)

arachNIDS IDS253
URL: http://www.whitehats.com/info/IDS253 (Oct 1,2002)

Sven Dietrich, Neil Long and David Dittrich
"An Analysis of the Shaft Distributed Denial of Service Tool" May, 2000
URL: http://www.chi-publishing.com/isb/backissues/ISB_2000/ISB0504/ISB0504SDNLDD.pdf (Oct 5, 2002)

Dave Wreski "Ramen Linux Worm Propagation" Jan 18, 2001
URL: http://www.linuxsecurity.com/articles/network_security_article-2335.html (Oct 6,2002)

HackGuard.net "Advanced Trojan Removal Information" 2002
URL: http://www.hackguard.net/advanced.htm (Oct 6,2002)

Stan Hoffman "GCIA Practical Assignment" Oct 28, 2001
URL:
http://www.whitehats.ca/main/members/Packet_Geek/GCIA_Practical/GCIA_Practical.html (Oct 6,2002)

Tod Beardsley "GCIA Practical Assignment" May 8, 2002
URL: http://www.giac.org/practical/Tod_Beardsley_GCIA.doc (Oct 6,2002)

Alex Stephens "GCIA Practical Assignment" Mar 25, 2001
URL: http://www.sans.org/y2k/practical/Alex_Stephens_GCIA.htm (Oct 6,2002)

Bradley Urwiller "GCIA Practical Assignment" Apr 23, 2002
URL: http://www.giac.org/practical/Bradley_Urwiller_GCIA.pdf (Oct 6,2002)

Mark Menke "GCIA Practical Assignment"
URL: http://www.giac.org/practical/Mark_Menke_GCIA.doc (Oct 6,2002)

Jeff Zahr "GCIA Practical Assignment" Nov 15, 2001
URL: http://www.giac.org/practical/Jeff_Zahr_GCIA.doc (Oct 6,2002)

Edward Peck "GCIA Practical Assignment" Aug 4, 2001
URL: http://www.giac.org/practical/Edward_Peck_GCIA.doc (Oct 6,2002)

John Jenkinson "GCIA Practical Assignment" Aug 13, 2001
URL: http://www.giac.org/practical/John_Jenkinson_GCIA.doc (Oct 6,2002)

Joni Ramos "GCIA Practical Assignment" 2001
URL: http://www.giac.org/practical/Joni_Ramos_GCIA.doc (Oct 6,2002)

Tyler Schacht "GCIA Practical Assignment" Aug 16, 2001
URL: http://www.giac.org/practical/Tyler_Schacht_GCIA.doc (Oct 6,2002)

Judy Novak, Stephen Northcutt, Mike Bost, Hal Pomeranz, Jean Triquet, Bill Ralph, Earl Carter, and Bryce Alexander.
SANS Institute "Intrusion Detection In-Depth" courseware 2002

W. Richard Stevens "TCP/IP Illustrated, Volume 1 The Protocols" 20th printing 2001
ISBN 0201633469

ARIN, American Registry for Internet Numbers "Whois tool"
URL: http://ws.arin.net/cgi-bin/whois.pl (Oct 1, 2002)

APNIC, Asia Pacific Network Information Center "Whois tool"
URL: http://www.apnic.net/apnic-bin/whois2.pl (Oct 1, 2002)

RIPE, Réseaux IP Européens "Query the Ripe Whois Database"
URL: http://www.ripe.net/perl/whois (Oct 1, 2002)

JNIC, Japan Network Information Center "Whois tool"
URL: http://www.nic.ad.jp/ (Oct 1,2002)

DShield.org "IP Info"
URL: http://www.dshield.org/ipinfo.php?ip= (Oct 1, 2002)

Silicon Defense "SnortSnarf snort alert browser"
URL: http://www.silicondefense.com/software/snortsnarf/index.htm (Oct 1, 2002)

Martin Roesch "Snort User Manual Snort Release: 1.9.x" Apr 26, 2002
URL: http://www.snort.org/docs/SnortUsersManual.pdf (Aug 30,2002)

Appendix A – Full trace of Assignment 1

Cisco PIX subnet address handling vulnerability

The trace was captured by a SHADOW IDS using tcpdump with the default snap length of 68 bytes (including layer 2 header). Therefor the packets are not all complete.

```
07:32:38.283937 151.100.89.67.22 > MY.NET.97.224.22: S [tcp sum ok]
1345578220:1345578220(0) win 54580 (ttl 112, id 16368, len 40)
0x0000  4500 0028 3ff0 0000 7006 f548 9764 5943     E..(?...p..H.dYC
0x0010  xxxx 61e0 0016 0016 5033 e4ec 3dbb b4e9     ..a.....P3..=...
0x0020  5002 d534 9d25 0000 0000 0000 0000          P..4.%........
07:32:38.284163 MY.NET.97.224.22 > 151.100.89.67.22: S [tcp sum ok]
1124644420:1124644420(0) ack 1345578221 win 4096 <mss 1460> (ttl 255, id 6171, len 44)
0x0000  4500 002c 181b 0000 ff06 8e19 xxxx 61e0     E..,..........a.
0x0010  9764 5943 0016 0016 4308 b644 5033 e4ed  .dYC....C..DP3..
0x0020  6012 1000 43e5 0000 0204 05b4 0000          `...C.........
07:32:38.330852 151.100.89.67.22 > MY.NET.97.253.22: S [tcp sum ok]
1345578220:1345578220(0) win 54580 (ttl 112, id 16368, len 40)
0x0000  4500 0028 3ff0 0000 7006 f52b 9764 5943     E..(?...p..+.dYC
0x0010  xxxx 61fd 0016 0016 5033 e4ec 3dbb b4e9     ..a.....P3..=...
0x0020  5002 d534 9d08 0000 0000 0000 0000          P..4..........
07:32:38.343561 151.100.89.67.22 > MY.NET.97.224.22: R [tcp sum ok]
1345578221:1345578221(0) win 0 (ttl 239, id 33823, len 40)
0x0000  4500 0028 841f 0000 ef06 3219 9764 5943     E..(......2..dYC
0x0010  xxxx 61e0 0016 0016 5033 e4ed 0000 0000     ..a.....P3......
0x0020  5004 0000 64fc 0000 0000 0000 0000          P...d.........
07:32:38.409393 151.100.89.67.4268 > MY.NET.97.224.22: S 1381191936:1381191936(0) win
32120 <mss 1460,sackOK,timestamp 542804848[|tcp]> (DF) (ttl 48, id 33829, len 60)
0x0000  4500 003c 8425 4000 3006 b0ff 9764 5943     E..<.%@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5100 0000 0000     ..a.....RSQ.....
0x0020  a002 7d78 5517 0000 0204 05b4 0402 080a     ..}xU...........
0x0030  205a 8b70 0000                              .Z.p..
07:32:38.409556 MY.NET.97.224.22 > 151.100.89.67.4268: S [tcp sum ok]
872965664:872965664(0) ack 1381191937 win 4096 <mss 1460> (ttl 255, id 6173, len 44)
0x0000  4500 002c 181d 0000 ff06 8e17 xxxx 61e0     E..,..........a.
0x0010  9764 5943 0016 10ac 3408 6620 5253 5101  .dYC....4.f.RSQ.
0x0020  6012 1000 2440 0000 0204 05b4 0402          `...$@........
07:32:38.452593 151.100.89.67.4268 > MY.NET.97.224.22: . [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 33831, len 40)
0x0000  4500 0028 8427 4000 3006 b111 9764 5943  E..(.'@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621     ..a.....RSQ.4.f!
0x0020  5010 7d78 ce84 0000 2100 0000 2c00          P.}x....!...,.
07:32:38.453312 MY.NET.97.253.22 > 151.100.89.67.4268: P 872965665:872965684(19) ack
1381191937 win 4096 (ttl 255, id 6174, len 59)
0x0000  4500 003b 181e 0000 ff06 8dea xxxx 61fd     E..;..........a.
0x0010  9764 5943 0016 10ac 3408 6621 5253 5101  .dYC....4.f!RSQ.
0x0020  5018 1000 a68b 0000 5353 482d 312e 352d  P.......SSH-1.5-
0x0030  4369 7363 6f2d                           Cisco-
07:32:38.497426 151.100.89.67.4268 > MY.NET.97.253.22: R [tcp sum ok]
1381191937:1381191937(0) win 0 (ttl 239, id 33833, len 40)
0x0000  4500 0028 8429 0000 ef06 31f2 9764 5943     E..(.)....1..dYC
0x0010  xxxx 61fd 10ac 0016 5253 5101 0000 0000     ..a.....RSQ.....
0x0020  5004 0000 e615 0000 5353 482d 312e          P.......SSH-1.
07:32:48.482733 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
```

```
(DF) (ttl 48, id 34171, len 40)
0x0000  4500 0028 857b 4000 3006 afbd 9764 5943   E..(.{@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:32:51.461928 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 34268, len 40)
0x0000  4500 0028 85dc 4000 3006 af5c 9764 5943   E..(..@.0..\.dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 0101 0512 7c83        P.}x........|.
07:32:57.450108 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 34478, len 40)
0x0000  4500 0028 86ae 4000 3006 ae8a 9764 5943   E..(..@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:33:09.453163 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 34862, len 40)
0x0000  4500 0028 882e 4000 3006 ad0a 9764 5943   E..(..@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:33:33.453593 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 35582, len 40)
0x0000  4500 0028 8afe 4000 3006 aa3a 9764 5943   E..(..@.0.:.dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:34:21.450910 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 37448, len 40)
0x0000  4500 0028 9248 4000 3006 a2f0 9764 5943   E..(.H@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:35:57.447524 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 44125, len 40)
0x0000  4500 0028 ac5d 4000 3006 88db 9764 5943   E..(.]@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:37:57.439245 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 49737, len 40)
0x0000  4500 0028 c249 4000 3006 72ef 9764 5943   E..(.I@.0.r..dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:39:57.438335 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 56306, len 40)
0x0000  4500 0028 dbf2 4000 3006 5946 9764 5943   E..(..@.0.YF.dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00        P.}x....!...,.
07:41:57.430543 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 64049, len 40)
0x0000  4500 0028 fa31 4000 3006 3b07 9764 5943   E..(.1@.0.;..dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 0021 8e1e 8005        P.}x.....!....
07:43:57.429004 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 6612, len 40)
0x0000  4500 0028 19d4 4000 3006 1b65 9764 5943   E..(..@.0..e.dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 0101 050a 91bd        P.}x..........
07:45:57.422307 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 13391, len 40)
0x0000  4500 0028 344f 4000 3006 00ea 9764 5943   E..(4O@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621   ..a.....RSQ.4.f!
```

```
0x0020  5011 7d78 ce83 0000 2100 0000 2c00          P.}x....!...,.
07:47:57.425104 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 21657, len 40)
0x0000  4500 0028 5499 4000 3006 e09f 9764 5943  E..(T.@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621  ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00          P.}x....!...,.
07:49:57.413605 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 28342, len 40)
0x0000  4500 0028 6eb6 4000 3006 c682 9764 5943  E..(n.@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621  ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00          P.}x....!...,.
07:51:57.412920 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 36169, len 40)
0x0000  4500 0028 8d49 4000 3006 a7ef 9764 5943  E..(.I@.0....dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621  ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00          P.}x....!...,.
07:53:57.406662 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 40658, len 40)
0x0000  4500 0028 9ed2 4000 3006 9666 9764 5943  E..(..@.0..f.dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621  ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00          P.}x....!...,.
07:55:57.401447 151.100.89.67.4268 > MY.NET.97.224.22: F [tcp sum ok] 1:1(0) ack 1 win 32120
(DF) (ttl 48, id 47874, len 40)
0x0000  4500 0028 bb02 4000 3006 7a36 9764 5943  E..(..@.0.z6.dYC
0x0010  xxxx 61e0 10ac 0016 5253 5101 3408 6621  ..a.....RSQ.4.f!
0x0020  5011 7d78 ce83 0000 2100 0000 2c00          P.}x....!...,.
```

Appendix B – Full trace of detect #1 Strange Gnutella traffic

```
12:18:24.254488 46.5.180.250.61284 > 24.165.145.34.6385: P [bad tcp cksum f9f9!]
3979059731:3979059785(54) ack 486191126 win 17520 (DF) (ttl 124, id 14661, len 94, bad cksum
3e94!)
0x0000  4500 005e 3945 4000 7c06 3e94 2e05 b4fa   E..^9E@.|.>.....
0x0010  18a5 9122 ef64 18f1 ed2b a213 1cfa b016   ...".d...+......
0x0020  5018 4470 8a4c 0000 474e 5554 454c 4c41   P.Dp.L..GNUTELLA
0x0030  2043 4f4e 4e45 4354 2f30 2e36 0d0a 5573   .CONNECT/0.6..Us
0x0040  6572 2d41 6765 6e74 3a20 476e 7563 6c65   er-Agent:.Gnucle
0x0050  7573 2031 2e36 2e30 2e30 0d0a 0d0a        us.1.6.0.0....
12:23:56.744488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum fa5c!]
487265248:487265460(212) win 20504 (DF) (ttl 110, id 57779, len 232, bad cksum a39b!)
0x0000  4500 00e8 e1b3 4000 6e06 a39b 18a5 9122   E.....@.n......"
0x0010  2e05 b4fa 18f1 ef64 1d0b 13e0 ed35 67ea   .......d.....5g.
0x0020  0000 5018 1f16 1ad7 0000 fdcc d1e1 06a8   ..P.............
0x0030  805b ff5d 1c0f 8ad8 0100 0001 0600 0000   .[.]............
0x0040  0001 c11c ec4b 304b f5ff 01d8 9777 1774   .....K0K.....w.t
0x0050  0000 0304 0000 0000 fec7 0061 dccf 0e46   ...........a...F
0x0060  8871 8ac8 3f7b aec7 8103 035c 0000 0001   .q..?{.....\....
0x0070  ca18 0a00 0004 9500 0000 8803 0000 a233   ...............3
0x0080  2800 4661 6974 686c 6573 7320 2d20 546f   (.Faithless.-.To
0x0090  7461 6c6c 7920 4461 6e63 6520 2d20 3031   tally.Dance.-.01
0x00a0  202d 2057 6520 436f 6d65 2031 2e6d 7033   .-.We.Come.1.mp3
0x00b0  0000 4d52 5048 021e 1901 f010 6427 bf34   ..MRPH......d'.4
0x00c0  7cd6 1185 fc00 a0cc 7bbe 944c b073 961f   |.......{..L.s..
0x00d0  3a66 4797 2e2e 6c21 90aa 1780 0106 0800   :fG...l!........
0x00e0  0000 0000 676f 6c65                       ....gole
12:39:08.324488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum fa2c!]
489719086:489720042(956) win 20504 (DF) (ttl 110, id 21426, len 976, bad cksum 2eb5!)
0x0000  4500 03d0 53b2 4000 6e06 2eb5 18a5 9122   E...S.@.n......"
0x0010  2e05 b4fa 18f1 ef64 1d30 852e ed49 de30   .......d.0...I.0
0x0020  0000 5018 2115 ef97 0000 d093 29e2 c724   ..P.!.......)..$
0x0030  596e ff30 6243 b5c1 bb00 0001 0600 0000   Yn.0bC..........
0x0040  0028 a318 e4af 7ed6 1186 9d44 4553 5400   .(....~....DEST.
0x0050  0000 0106 0000 0000 c91e 18f8 b5fb 9253   ...............S
0x0060  ff22 cd61 9d75 f300 8001 0635 0000 0000   .".a.u.....5....
0x0070  0068 6f6f 7469 6520 616e 6420 7468 6520   .hootie.and.the.
0x0080  626c 6f77 6669 7368 2063 6c6f 7369 6e67   blowfish.closing
0x0090  2074 696d 6520 756e 706c 7567 6765 6420   .time.unplugged.
0x00a0  6d70 3300 acc3 7480 c1f3 8c4a 2cb4 7f43   mp3...t....J,..C
0x00b0  8fac 5841 8001 0642 0000 0000 0064 6176   ..XA...B.....dav
0x00c0  6964 2062 7972 6e65 206c 6f6f 6b20 696e   id.byrne.look.in
0x00d0  746f 2074 6865 2065 7965 6261 6c6c 2030   to.the.eyeball.0
0x00e0  3320 7468 6520 6772 6561 7420 696e 746f   3.the.great.into
0x00f0  7869 6361 7469 6f6e 206d 7033 001e 68b4   xication.mp3..h.
0x0100  6395 7ed6 119a 6caa 0004 00d9 0480 0205   c.~...l.........
0x0110  1100 0000 0000 7769 636b 6564 206f 6e65   ......wicked.one
0x0120  2061 7366 00dc 64b4 6395 7ed6 119a 6caa   .asf..d.c.~...l.
0x0130  0004 00d9 0480 0205 0c00 0000 0000 6e69   ..............ni
0x0140  6b6b 6920 6d70 6700 06e4 e4f4 9de4 22e8   kki.mpg.......".
0x0150  ff04 aa36 b9eb cd00 0002 0500 0000 001f   ...6............
0x0160  cfae 0e1b 3e1e 3eff c36e 1fca 2476 0000   ....>.>..n..$v..
0x0170  0304 0000 0000 bea9 e676 7073 9f75 ff3f   .........vps.u.?
0x0180  187e b1cb 8500 0001 0600 0000 00dc 0788   .~..............
0x0190  9d8b 7ed6 11a0 1200 50ba 4fe5 6680 0205   ..~.....P.O.f...
0x01a0  1700 0000 0000 696e 2074 6872 6f75 6768   ......in.through
```

```
0x01b0  2074 6865 206e 6967 6874 00dd 0788 9d8b  .the.night......
0x01c0  7ed6 11a0 1200 50ba 4fe5 6680 0205 0e00  ~.....P.O.f.....
0x01d0  0000 0000 6465 6620 6c65 7070 6172 6400  ....def.leppard.
0x01e0  f24f 7090 d69b 7d48 ffbd 98f4 97e6 e000  .Op...}H........
0x01f0  0002 0500 0000 0029 6c4c 3c7a 04eb 7e6f  .......)lL<z..~o
0x0200  76f2 b32d fec9 8980 0106 1c00 0000 0000  v..-............
0x0210  7468 6973 2069 7320 7370 696e 616c 2074  this.is.spinal.t
0x0220  6170 2064 7620 6176 6900 22a7 b61e dc7e  ap.dv.avi."....~
0x0230  d611 9c5c 4445 5354 0000 8001 060a 0000  ...\DEST........
0x0240  0000 0061 7afd 7a63 616e 0014 6e75 49e8  ...az.zcan..nuI.
0x0250  4fcc 4e98 0627 9316 d410 7e80 0304 1300  O.N..'....~.....
0x0260  0000 0000 6a61 6465 206d 6172 6365 6c61  ....jade.marcela
0x0270  206d 7067 00de a1bb c3ff 6601 4f8e 8ffa  .mpg......f.O...
0x0280  031b 4d4f 4880 0304 1000 0000 0000 776f  ..MOH.........wo
0x0290  6e67 2066 6179 6520 6d70 3300 2387 e86d  ng.faye.mp3.#..m
0x02a0  6fb3 6447 900f dbac 33c4 4711 8003 040e  o.dG....3.G.....
0x02b0  0000 0000 0073 7761 6c6c 6f77 206d 7067  .....swallow.mpg
0x02c0  0050 f8d3 5f41 866d 4eb2 e8ce 7d24 6129  .P.._A.mN...}$a)
0x02d0  3f80 0304 1500 0000 0000 7570 2077 6865  ?.........up.whe
0x02e0  7265 2077 6520 6265 6c6f 6e67 0084 a196  re.we.belong....
0x02f0  5f41 7583 4984 e64c 4fd3 19b5 ea80 0304  _Au.I..LO.......
0x0300  0b00 0000 0000 7265 756e 6974 6564 000b  ......reunited..
0x0310  30d9 eb5e f385 4a91 d80b a6eb 6e21 c080  0..^..J.....n!..
0x0320  0304 1200 0000 0000 646f 6e27 7420 6b6e  ........don't.kn
0x0330  6f77 206d 7563 6800 69b2 8b1b 4ca7 2d3e  ow.much.i...L.->
0x0340  ff92 08f0 5dc1 3500 0003 0400 0000 00a3  ....].5.........
0x0350  9978 d4ac 342e 8e1c b05c 34b4 ddaf fc00  .x..4....\4.....
0x0360  0304 0000 0000 3551 1939 9a21 03db ff72  ......5Q.9.!...r
0x0370  2a4f 55cc 4c00 8001 0655 0000 0000 006f  *OU.L....U.....o
0x0380  7a7a 7920 6f73 626f 726e 6520 646d 7820  zzy.osborne.dmx.
0x0390  6f6c 2064 6972 7479 2062 6173 7461 7264  ol.dirty.bastard
0x03a0  2074 6865 2063 7279 7374 616c 206d 6574  .the.crystal.met
0x03b0  686f 6420 6675 7a7a 6275 6262 6c65 206e  hod.fuzzbubble.n
0x03c0  6f77 6865 7265 2074 6f20 7275 6e20 6d70  owhere.to.run.mp
12:39:19.914488 24.165.145.34.6385 > 46.5.180.250.61284: RWE [bad tcp cksum f9f9!]
3808783643:3808783940(297) ack 4110633783 win 20504 urg 38205 [RST+
...0...J..P.!..=.....I...B..C.] (DF) (ttl 110, id 64181, len 321, bad cksum 8a40!)
0x0000  4500 0141 fab5 4000 6e06 8a40 18a5 9122  E..A..@.n..@..."
0x0010  2e05 b4fa 18f1 ef64 0000 1d30 e22e ed4a  .......d...0...J
0x0020  11f4 5018 21ce 953d 0000 f783 e649 8608  ..P.!..=.....I..
0x0030  9842 afdb 43da 1bda 126a 8001 0617 0000  .B..C....j......
0x0040  0000 0061 6365 206f 6620 6261 7365 206d  ...ace.of.base.m
0x0050  7033 0075 726e 3a00 099d 266c 587e d611  p3.urn:...&lX~..
0x0060  9215 0040 f452 4988 8001 060e 0000 0000  ...@.RI.........
0x0070  0053 6572 656e 6469 7069 7479 0054 6ef9  .Serendipity.Tn.
0x0080  709f d156 16ff 83e3 73e4 3e9c 0000 0106  p..V....s.>.....
0x0090  0000 0000 a3d9 e949 324a 0307 fff7 ce1c  .......I2J......
0x00a0  b721 b300 0001 0600 0000 0078 9fe7 9302  .!.........x....
0x00b0  441a 4592 963e 65fd d45b c601 0601 0e00  D.E..>e..[......
0x00c0  0000 ca18 4219 022b fb04 0000 a364 fe00  ....B..+.....d..
0x00d0  f882 e949 6d74 0e04 ff53 6af1 2acf 2d00  ...Imt...Sj.*.-.
0x00e0  8002 0517 0000 0000 0067 6c6f 6265 2063  .........globe.c
0x00f0  7265 616d 7920 6461 7920 6d70 3300 ee43  reamy.day.mp3..C
0x0100  686c 96b9 684b 9737 a600 0102 b4a4 0002  hl..hK.7........
0x0110  0500 0000 00b6 4b8e a31f 7ed6 1187 3695  ......K...~...6.
0x0120  8bb4 ee66 4500 0205 0000 0000 4239 2f0f  ...fE.......B9/.
0x0130  cef9 62e1 ff87 27cd bc81 3b00 0001 0600  ..b...'...;.....
0x0140  00                                       .
```

```
12:46:14.694488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum fa01!]
490699856:490699936(80) win 20504 (DF) (ttl 110, id 17955, len 100, bad cksum 3fb0!)
0x0000  4500 0064 4623 4000 6e06 3fb0 18a5 9122   E..dF#@.n.?...."
0x0010  2e05 b4fa 18f1 ef64 1d3f 7c50 ed4e 45e7   .......d.?|P.NE.
0x0020  0000 5018 1fc0 65f5 0000 2db1 9969 f206   ..P...e...-..i..
0x0030  9f56 fff2 ef12 14d7 6900 0001 0600 0000   .V......i.......
0x0040  00f5 8736 3854 dd00 9924 ddcc 4159 8381   ...68T...$..AY..
0x0050  b401 0205 0e00 0000 ca18 18ea eebc c000   ................
0x0060  0000 cb93                                 ....
13:21:24.094488 24.165.145.34.6385 > 46.5.180.250.61284: FW [bad tcp cksum f9f9!]
10716463:10716479(16) ack 315968346 win 20496 urg 42160 (DF) (ttl 110, id 65368, len 40, bad
cksum 86b6!)
0x0000  4500 0028 ff58 4000 6e06 86b6 18a5 9122   E..(.X@.n......"
0x0010  2e05 b4fa 18f1 ef64 1d9e 3544 0000 ed6d   .......d..5D...m
0x0020  1bb1 5010 200c a4b0 0000 0000 0000        ..P.............
13:31:44.834488 24.165.145.34.6385 > 46.5.180.250.61284: . [bad tcp cksum f9f9!]
498710468:498710705(237) win 20504 (DF) (ttl 110, id 62193, len 257, bad cksum 9244!)
0x0000  4500 0101 f2f1 4000 6e06 9244 18a5 9122   E.....@.n..D..."
0x0010  2e05 b4fa 18f1 ef64 1db9 b7c4 ed73 ec5c   .......d.....s.\
0x0020  0000 5018 1d1d 94e4 0000 8b2b 5c2c e177   ..P........+\,.w
0x0030  b165 ff26 a52e f8f2 2d00 0001 0600 0000   .e.&....-.......
0x0040  0067 74fc 750c e751 42ae c43a 0f76 ad83   .gt.u..QB..:.v..
0x0050  0300 0106 0000 0000 e03c afaf 6b58 cb18   .........<..kX..
0x0060  ffc6 ca8f 318c 6300 0001 0600 0000 007d   ....1.c........}
0x0070  a2e7 fc19 b3a5 4482 f4a7 65e5 8a34 6780   ......D...e..4g.
0x0080  0205 1700 0000 0000 6c65 6420 7a65 7070   ........led.zepp
0x0090  6c69 6e20 6d70 3300 7572 6e3a 00e2 f4c9   lin.mp3.urn:....
0x00a0  04a1 4cdb 78cc 88bc d229 c6c9 0d00 0106   ..L.x....).......
0x00b0  0000 0000 857a ff7e 5d63 ab71 e600 0201   .....z.~]c.q....
0x00c0  0003 0000 0002 0500 0000 0000 a2fa 0dce   ................
0x00d0  7ed6 11b8 9be3 b39d 2473 7380 0106 0a00   ~.......$ss.....
0x00e0  0000 0000 6c65 7665 6c34 3200 5052 be25   ....level42.PR.%
0x00f0  332b 8f0a ffe8 46d2 8072 db00 0001 0600   3+....F..r......
0x0100  00                                        .
13:45:29.774488 24.165.145.34.6385 > 46.5.180.250.61284: SRW [bad tcp cksum f9f9!]
501100658:501101027(369) ack 60804 win 20504 [RST+ ..0r......P."8.O...:.S........] (DF) (ttl 110, id
32198, len 393, bad cksum 6e8!)
0x0000  4500 0189 7dc6 4000 6e06 06e8 18a5 9122   E...}.@.n......"
0x0010  2e05 b4fa 18f1 ef64 1dde 3072 0000 ed84   .......d..0r....
0x0020  1c96 5018 2238 ed4f 0000 b43a db53 a4b1   ..P."8.O...:.S..
0x0030  f288 dede de0b 91d2 68ad 8001 062c 0000   ........h....,..
0x0040  0000 0073 7461 7274 7265 6b20 6473 3920   ...startrek.ds9.
0x0050  3578 3136 2064 7220 6261 7368 6972 2069   5x16.dr.bashir.i
0x0060  2070 7265 7375 6d65 2061 7669 00dd a1fc   .presume.avi....
0x0070  bc83 c7eb ceff 9fe3 8190 cc8b 0080 0106   ................
0x0080  1d00 0000 0000 626c 6f6e 6469 6520 6865   ......blondie.he
0x0090  6172 7420 6f66 2067 6c61 7373 206d 7033   art.of.glass.mp3
0x00a0  00de 4bc5 1514 6064 bbff 0bce cfd3 4a04   ..K...`d......J.
0x00b0  0080 0205 2d00 0000 0000 736f 6e69 6320   ....-.....sonic.
0x00c0  666f 756e 6472 7920 736f 756e 6420 666f   foundry.sound.fo
0x00d0  7267 6520 7636 2030 7265 6e65 6761 6465   rge.v6.0renegade
0x00e0  2072 6172 00d6 f1c4 f48b 5955 eaff 43e5   .rar......YU..C.
0x00f0  384d 802c 0000 0205 0000 0000 5699 5ef2   8M.,........V.^.
0x0100  18bd 67ce 6886 7169 37bc 022e 8003 041c   ..g.h.qi7.......
0x0110  0000 0000 0072 7573 7369 616e 2061 6d61   .....russian.ama
0x0120  7475 7265 2072 6170 6520 6d70 6567 00b0   ture.rape.mpeg..
0x0130  9afa 8a18 acc0 ceff 1573 7e6b 160e 0000   .........s~k....
0x0140  0106 0000 0000 8ea4 7426 f8d3 e597 2b28   ........t&....+(
```

```
0x0150  53bd 977a a80c 0001 0600 0000 00d2 8326  S..z...........&
0x0160  65d0 32e5 458e 507a 8382 7a86 7e00 0106  e.2.E.Pz..z.~...
0x0170  0000 0000 1aba 1727 4a4b 4187 ffda 0334  .......'JKA....4
0x0180  758c c000 0001 0600 00                   u.........
13:46:41.054488 24.165.145.34.0 > 46.5.180.250.6385: SPE [bad tcp cksum f9f9!]
4016315872:4016316105(233) win 20504 urg 23035 (DF) (ttl 110, id 46809, len 261, bad cksum
ce58!)
0x0000  4500 0105 b6d9 4000 6e06 ce58 18a5 9122  E.....@.n..X..."
0x0010  2e05 b4fa 0000 18f1 ef64 1de0 757a ed85  .........d..uz..
0x0020  2d6a 5018 20ea 59fb 0000 d281 6400 e1d0  -jP...Y.....d...
0x0030  2d4c a6ca 1100 d633 d28a 8001 0611 0000  -L.....3........
0x0040  0000 006d 7920 7374 6f6e 6579 0075 726e  ...my.stoney.urn
0x0050  3a00 aa40 0969 d72a 2943 90fb 17e2 c749  :..@.i.*)C.....I
0x0060  0414 8001 060e 0000 0000 0062 7261 7a69  ...........brazi
0x0070  6c20 6d70 6567 0022 b370 e96e 0f9c a57a  l.mpeg.".p.n...z
0x0080  2b88 2146 0734 5d80 0205 1100 0000 0000  +.!F.4].........
0x0090  7175 6974 7465 2070 6173 206d 7033 0044  quitte.pas.mp3.D
0x00a0  a8df c1b1 23e2 2ca3 a423 d8ee 6d62 5d80  ....#.,..#..mb].
0x00b0  0106 0c00 0000 0000 646f 7562 6c65 6461  ........doubleda
0x00c0  7900 c248 91e1 ec5f c0bf ffe9 f479 91cd  y..H..._.....y..
0x00d0  d200 0001 0600 0000 005e c03f 4f5a 46ca  .........^.?OZF.
0x00e0  3aff 7d43 72b8 7dfa 0000 0106 0000 0000  :.}Cr.}.........
0x00f0  8762 3121 0367 d930 ffb2 aa9f d009 6000  .b1!.g.0......`.
0x0100  0002 0500 00                             .....
```

Appendix C – Infected internal hosts by a Code Red/Code Red II/Nimda virus

Host	Alerts
MY.NET.91.104	6116
MY.NET.153.168	5053
MY.NET.85.74	4170
MY.NET.84.166	2586
MY.NET.183.26	1882
MY.NET.153.190	1390
MY.NET.153.106	1127
MY.NET.153.112	1064
MY.NET.163.125	1052
MY.NET.15.212	1031
MY.NET.153.135	940
MY.NET.87.193	880
MY.NET.116.84	850
MY.NET.153.143	706
MY.NET.117.133	700
MY.NET.145.199	698
MY.NET.153.136	666
MY.NET.153.177	569
MY.NET.85.53	534
MY.NET.153.205	517
MY.NET.91.103	511
MY.NET.183.25	495
MY.NET.162.68	443
MY.NET.153.179	425
MY.NET.153.153	418
MY.NET.153.71	416
MY.NET.153.120	409
MY.NET.145.27	400
MY.NET.153.196	384
MY.NET.143.107	379
MY.NET.91.106	369
MY.NET.53.164	367
MY.NET.153.118	355
MY.NET.153.167	353
MY.NET.190.16	350
MY.NET.87.54	326
MY.NET.53.120	325
MY.NET.153.137	321
MY.NET.133.17	315
MY.NET.178.181	312
MY.NET.130.73	246
MY.NET.53.55	244
MY.NET.104.204	232
MY.NET.153.119	227
MY.NET.178.140	222
MY.NET.153.114	221
MY.NET.15.71	219

MY.NET.53.56	209
MY.NET.111.48	208
MY.NET.153.127	197
MY.NET.80.134	162
MY.NET.153.110	161
MY.NET.117.202	153
MY.NET.18.36	145
MY.NET.84.5	142
MY.NET.87.201	138
MY.NET.53.196	137
MY.NET.118.11	137
MY.NET.178.78	134
MY.NET.84.200	133
MY.NET.53.39	126
MY.NET.153.189	126
MY.NET.153.169	119
MY.NET.84.216	119
MY.NET.84.162	118
MY.NET.153.188	117
MY.NET.153.108	117
MY.NET.153.105	108
MY.NET.53.49	106
MY.NET.87.121	106
MY.NET.150.86	103
MY.NET.88.5	102
MY.NET.90.43	94
MY.NET.150.67	92
MY.NET.15.179	92
MY.NET.70.50	92
MY.NET.153.116	91
MY.NET.88.243	84
MY.NET.113.4	83
MY.NET.84.225	83
MY.NET.100.141	79
MY.NET.153.172	78
MY.NET.106.107	78
MY.NET.153.113	77
MY.NET.84.142	76
MY.NET.139.51	74
MY.NET.153.161	74
MY.NET.18.30	72
MY.NET.99.185	72
MY.NET.116.75	72
MY.NET.140.196	71
MY.NET.153.122	69
MY.NET.84.244	61
MY.NET.80.159	60
MY.NET.53.47	60
MY.NET.182.60	60
MY.NET.111.197	59
MY.NET.116.52	58
MY.NET.88.78	56

MY.NET.106.176	56
MY.NET.153.185	56
MY.NET.130.60	55
MY.NET.84.245	55
MY.NET.10.175	54
MY.NET.84.141	53
MY.NET.88.201	50
MY.NET.153.211	50
MY.NET.84.147	49
MY.NET.99.190	49
MY.NET.114.88	45
MY.NET.141.67	40
MY.NET.110.52	39
MY.NET.157.239	38
MY.NET.153.117	38
MY.NET.83.247	37
MY.NET.109.99	37
MY.NET.100.224	36
MY.NET.83.235	36
MY.NET.178.57	34
MY.NET.153.111	32
MY.NET.84.196	31
MY.NET.117.151	31
MY.NET.115.66	29
MY.NET.70.232	28
MY.NET.111.196	28
MY.NET.83.118	28
MY.NET.153.206	28
MY.NET.111.178	27
MY.NET.144.51	27
MY.NET.153.125	27
MY.NET.153.123	27
MY.NET.178.56	27
MY.NET.183.55	26
MY.NET.153.124	25
MY.NET.153.121	25
MY.NET.130.52	25
MY.NET.53.53	24
MY.NET.184.40	24
MY.NET.88.151	24
MY.NET.81.6	23
MY.NET.85.52	22
MY.NET.151.64	22
MY.NET.83.169	22
MY.NET.116.37	21
MY.NET.153.152	20
MY.NET.150.97	19
MY.NET.177.44	18
MY.NET.178.75	18
MY.NET.84.190	17
MY.NET.53.51	17
MY.NET.100.6	16

MY.NET.84.240	15
MY.NET.53.40	15
MY.NET.157.105	15
MY.NET.153.126	14
MY.NET.89.154	14
MY.NET.145.55	12
MY.NET.180.169	12
MY.NET.153.146	12
MY.NET.53.128	12
MY.NET.83.48	12
MY.NET.183.17	12
MY.NET.88.137	11
MY.NET.84.233	11
MY.NET.84.157	11
MY.NET.162.22	11
MY.NET.168.92	11
MY.NET.53.52	10
MY.NET.153.210	10
MY.NET.84.146	9
MY.NET.153.151	9
MY.NET.130.132	9
MY.NET.53.44	8
MY.NET.168.181	8
MY.NET.145.220	8
MY.NET.100.4	8
MY.NET.153.187	7
MY.NET.87.6	6
MY.NET.87.53	6
MY.NET.53.100	6
MY.NET.113.7	6
MY.NET.104.47	6
MY.NET.153.165	6
MY.NET.153.160	6
MY.NET.130.133	6
MY.NET.84.156	6
MY.NET.110.227	5
MY.NET.177.48	5
MY.NET.109.11	5
MY.NET.168.59	5
MY.NET.84.150	4
MY.NET.153.195	4
MY.NET.168.185	4
MY.NET.17.43	4
MY.NET.110.168	3
MY.NET.87.37	3
MY.NET.53.160	3
MY.NET.163.91	3
MY.NET.130.131	3
MY.NET.140.171	3
MY.NET.84.203	3
MY.NET.105.151	3
MY.NET.111.30	3

MY.NET.168.103	3
MY.NET.53.45	3
MY.NET.88.246	3
MY.NET.84.242	3
MY.NET.180.62	3
MY.NET.162.193	3
MY.NET.15.219	2
MY.NET.81.125	2
MY.NET.53.35	2
MY.NET.150.235	2
MY.NET.109.10	2
MY.NET.152.251	2
MY.NET.152.171	2
MY.NET.153.174	2
MY.NET.108.250	2
MY.NET.84.165	2
MY.NET.153.148	2
MY.NET.104.155	2
MY.NET.84.223	2
MY.NET.84.226	1
MY.NET.53.57	1
MY.NET.153.141	1
MY.NET.84.167	1
MY.NET.145.155	1
MY.NET.178.118	1
MY.NET.88.251	1
MY.NET.153.176	1
MY.NET.152.164	1
MY.NET.87.41	1
MY.NET.140.79	1
MY.NET.109.25	1
MY.NET.110.76	1
MY.NET.145.214	1
MY.NET.153.107	1
MY.NET.153.109	1
MY.NET.116.47	1
MY.NET.198.17	1

Appendix D – Compromised internal hosts running the Shaft DDOS tool

Source	Alerts	Attacked hosts
MY.NET.70.74	590	16
MY.NET.70.54	557	16
MY.NET.70.63	547	16
MY.NET.70.186	540	16
MY.NET.70.109	537	16
MY.NET.70.12	535	15
MY.NET.70.45	531	15
MY.NET.70.193	531	16
MY.NET.70.70	529	15
MY.NET.70.79	525	16
MY.NET.70.199	525	16
MY.NET.70.35	523	15
MY.NET.70.217	521	16
MY.NET.70.173	519	16
MY.NET.70.116	519	15
MY.NET.70.227	519	16
MY.NET.70.245	519	16
MY.NET.70.18	519	16
MY.NET.70.86	518	16
MY.NET.70.101	516	15
MY.NET.70.189	515	16
MY.NET.70.195	515	15
MY.NET.70.146	514	16
MY.NET.70.121	513	16
MY.NET.70.13	511	16
MY.NET.70.31	510	16
MY.NET.70.151	508	15
MY.NET.70.174	508	16
MY.NET.70.111	507	15
MY.NET.70.143	507	15
MY.NET.70.178	503	16
MY.NET.70.112	499	16
MY.NET.70.136	499	15
MY.NET.70.47	494	16
MY.NET.70.134	493	16
MY.NET.70.44	492	13
MY.NET.70.124	491	16
MY.NET.70.226	491	16
MY.NET.70.29	491	15
MY.NET.70.171	489	15
MY.NET.70.162	488	16
MY.NET.70.16	487	15
MY.NET.70.222	484	16
MY.NET.70.97	482	15
MY.NET.70.233	481	14
MY.NET.70.56	481	15
MY.NET.70.142	481	16

MY.NET.70.201	480	16
MY.NET.70.25	478	16
MY.NET.70.182	477	15
MY.NET.70.84	477	16
MY.NET.70.137	477	16
MY.NET.70.72	475	16
MY.NET.70.126	474	16
MY.NET.70.165	474	13
MY.NET.70.228	473	16
MY.NET.70.170	473	16
MY.NET.70.90	472	15
MY.NET.70.11	471	16
MY.NET.70.139	471	16
MY.NET.70.234	470	16
MY.NET.70.68	469	16
MY.NET.70.208	468	16
MY.NET.70.145	468	15
MY.NET.70.240	466	16
MY.NET.70.6	465	16
MY.NET.70.175	465	15
MY.NET.70.169	465	15
MY.NET.70.188	464	16
MY.NET.70.21	464	15
MY.NET.70.113	464	16
MY.NET.70.123	462	14
MY.NET.70.108	462	16
MY.NET.70.252	461	16
MY.NET.70.14	461	14
MY.NET.70.22	460	16
MY.NET.70.30	460	16
MY.NET.70.106	460	16
MY.NET.70.141	460	16
MY.NET.70.237	459	16
MY.NET.70.207	459	16
MY.NET.70.219	458	15
MY.NET.70.152	457	16
MY.NET.70.225	456	14
MY.NET.70.105	456	16
MY.NET.70.154	456	16
MY.NET.70.122	455	15
MY.NET.70.144	455	15
MY.NET.70.94	454	15
MY.NET.70.55	452	16
MY.NET.70.179	451	15
MY.NET.70.220	451	16
MY.NET.70.118	447	16
MY.NET.70.206	447	16
MY.NET.70.120	447	16
MY.NET.70.95	446	16
MY.NET.70.128	443	15
MY.NET.70.34	442	16
MY.NET.70.147	442	14

MY.NET.70.160	442	16
MY.NET.70.251	442	15
MY.NET.70.82	441	15
MY.NET.70.127	441	14
MY.NET.70.117	441	16
MY.NET.70.185	441	16
MY.NET.70.36	440	16
MY.NET.70.8	439	16
MY.NET.70.19	438	13
MY.NET.70.209	438	14
MY.NET.70.241	438	14
MY.NET.70.66	438	16
MY.NET.70.32	437	15
MY.NET.70.73	437	15
MY.NET.70.28	437	16
MY.NET.70.99	437	14
MY.NET.70.53	436	15
MY.NET.70.177	436	16
MY.NET.70.89	436	15
MY.NET.70.181	435	13
MY.NET.70.67	434	16
MY.NET.70.238	434	16
MY.NET.70.1	434	15
MY.NET.70.242	431	16
MY.NET.70.75	428	16
MY.NET.70.214	426	15
MY.NET.70.33	425	16
MY.NET.70.125	423	15
MY.NET.70.203	422	13
MY.NET.70.172	421	15
MY.NET.70.167	421	15
MY.NET.70.129	420	15
MY.NET.70.100	420	16
MY.NET.70.98	420	16
MY.NET.70.103	420	16
MY.NET.70.130	418	16
MY.NET.70.52	418	14
MY.NET.70.253	418	15
MY.NET.70.229	417	15
MY.NET.70.132	416	16
MY.NET.70.196	415	15
MY.NET.70.231	414	16
MY.NET.70.57	412	16
MY.NET.70.159	412	16
MY.NET.70.224	412	16
MY.NET.70.161	411	14
MY.NET.70.212	411	14
MY.NET.70.78	411	16
MY.NET.70.247	410	16
MY.NET.70.4	409	16
MY.NET.70.3	409	16
MY.NET.70.149	409	15

MY.NET.70.184	409	16
MY.NET.70.83	409	16
MY.NET.70.20	409	16
MY.NET.70.71	408	16
MY.NET.70.223	407	15
MY.NET.70.168	407	16
MY.NET.70.194	406	15
MY.NET.70.164	405	16
MY.NET.70.180	405	15
MY.NET.70.163	404	16
MY.NET.70.65	404	16
MY.NET.70.249	404	16
MY.NET.70.131	403	16
MY.NET.70.157	402	16
MY.NET.70.2	402	16
MY.NET.70.133	401	15
MY.NET.70.58	401	15
MY.NET.70.7	398	15
MY.NET.70.102	397	16
MY.NET.70.204	396	15
MY.NET.70.192	396	13
MY.NET.70.104	395	14
MY.NET.70.198	395	14
MY.NET.70.244	394	15
MY.NET.70.114	393	16
MY.NET.70.37	393	14
MY.NET.70.211	392	15
MY.NET.70.202	392	16
MY.NET.70.148	389	15
MY.NET.70.43	389	15
MY.NET.70.191	389	16
MY.NET.70.27	389	16
MY.NET.70.155	388	16
MY.NET.70.92	387	14
MY.NET.70.110	387	16
MY.NET.70.190	387	14
MY.NET.70.50	385	14
MY.NET.70.62	385	15
MY.NET.70.135	384	15
MY.NET.70.215	384	16
MY.NET.70.197	384	16
MY.NET.70.232	383	13
MY.NET.70.140	383	16
MY.NET.70.236	382	16
MY.NET.70.15	382	16
MY.NET.70.246	381	16
MY.NET.70.5	381	15
MY.NET.70.243	381	16
MY.NET.70.60	381	15
MY.NET.70.24	380	15
MY.NET.70.41	380	14
MY.NET.70.119	379	15

MY.NET.70.221	378	13
MY.NET.70.17	378	15
MY.NET.70.166	378	16
MY.NET.70.9	377	13
MY.NET.70.205	376	14
MY.NET.70.85	375	15
MY.NET.70.153	370	15
MY.NET.70.213	368	15
MY.NET.70.88	368	15
MY.NET.70.150	368	13
MY.NET.70.218	367	16
MY.NET.70.23	367	15
MY.NET.70.235	366	16
MY.NET.70.138	364	15
MY.NET.70.26	364	15
MY.NET.70.87	363	16
MY.NET.70.40	362	13
MY.NET.70.77	358	15
MY.NET.70.107	356	16
MY.NET.70.93	355	16
MY.NET.70.250	355	15
MY.NET.70.64	354	15
MY.NET.70.216	351	13
MY.NET.70.10	350	14
MY.NET.70.210	350	16
MY.NET.70.59	348	14
MY.NET.70.200	347	15
MY.NET.70.42	347	15
MY.NET.70.49	342	16
MY.NET.70.76	341	15
MY.NET.70.254	341	16
MY.NET.70.46	340	15
MY.NET.70.248	340	15
MY.NET.70.91	334	14
MY.NET.70.187	332	16
MY.NET.70.81	332	15
MY.NET.70.176	324	16
MY.NET.70.183	324	15
MY.NET.70.51	324	14
MY.NET.70.230	323	14
MY.NET.70.39	317	15
MY.NET.70.96	315	16
MY.NET.70.115	304	15
MY.NET.70.158	303	15
MY.NET.70.61	296	15
MY.NET.70.239	294	13
MY.NET.70.156	293	16
MY.NET.70.80	289	13
MY.NET.70.38	275	16
MY.NET.70.48	274	15