Global Information Assurance Certification Paper

Intrusion Detection in Depth

GIAC Practical Assignment (v.3.0)

Roland Lee
26 February 2002

Table of Content

Assignment 1 – Describe the State of Intrusion Detection P.3

Assignment 2 – Network Detects P.12

Assignment 3 – "Analyze This" Scenario P.28

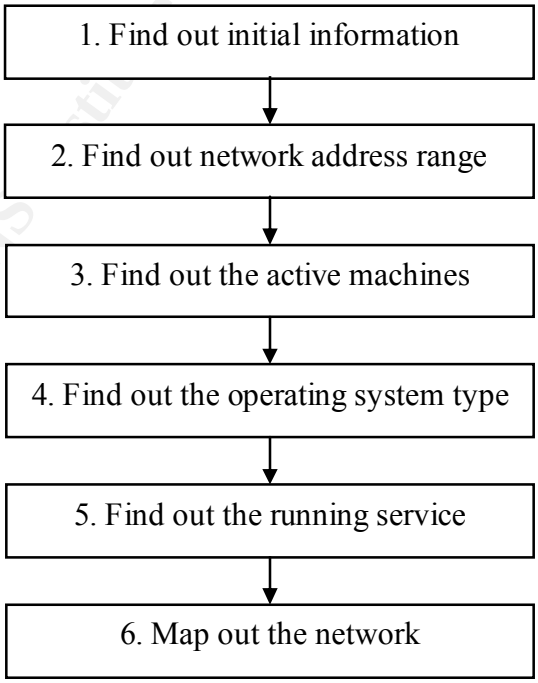
Assignment 1 – Describe the State of Intrusion Detection

Intrusion Detection for Firewall

Nowadays, nearly all companies have installed firewalls at their perimeter to control network access and protect its networks from malicious attack. Located at the outmost perimeter of the network, firewall can act as an effective safeguard. Not only it provides access control to internal and external network resources, but it also blocks thousands of malicious attacks every minute, or nearly every second. With proper logging, the firewall log can record the very detailed intrusion attempt. In this paper, I will implement a Perl script to find out port scan activities logged by Check Point Firewall-1 NG. My work is based on the paper "Intrusion Detection for FW-1" written by Lance Spitzner. In the paper, Lance has shared his experience in using a Unix shell script to check for port scan activities logged by Check Point Firewall-1 NG. The shell script runs on Solaris and can be configured to send an e-mail to the administrator when a port scan activity is detected. I have implemented a Perl script to do the same task for Check Point Firewall-1 NG under Windows NT/2000 environment, to turn Check Point Firewall-1 into an efficient intrusion detection system for port scan activity.

In most cases, for an exploit to occur, an attacker needs to understand the target environment before he can run an exploit to his target systems. He can do so by gathering preliminary information about the target systems. It is commonly known as reconnaissance phase. The information that the attacker interested in usually include operating system type, operating system version, running services, patch level of the systems, etc. As a rule of thumb to the attacker, the richer the information he can gather, the more likely his attack will succeed.

The following are the basic steps usually an attacker will take during reconnaissance phase:



During the first stage finding out initial information, the attacker will try to gather network information such as the IP address, domain names of the systems. Usually, if an attacker is specifically interested in your network, he will certainly obtain the information beforehand. In the second stage, the attacker will try to obtain the network address range, in the sake of concentrating his attack on his target network but not the others. He can do so by using some of the techniques like utilizing ARIN whois search. After knowing the IP address range, he will go to find out all the active machines in the network. The most common technique used is to send ICMP echo request to each IP address in the network address range. If a machine is alive, it will respond with an ICMP echo reply, provided that the firewall doesn't block outbound ICMP echo reply. After knowing which host is alive, the attacker will try to find out the operating system of the host. He can do so by some freely available tools like Nmap and Queso. In the stage of finding out running service, the attacker will try to find out some specific ports that the host is listening and he is specifically interested in. He can easily do so by using some port scanning tools. And finally, after obtaining all the above information, he can map out the target network.

Detecting port scan activity is thus important because port scanning activity indicates that someone is interested in your hosts, and he is trying to find out what applications are currently running at your hosts. Certainly, a network-based intrusion detection system can be used to effectively detect port scan activity. However, with proper logging, a firewall log can also be a valuable source for finding out port scan activity.

When an attacker performs port scanning to the network, he will either:

- Scan a list of ports on a specific host
- Scan a list of hosts for a specific port

Undoubtedly, to have a clearer picture of the target network, an attacker may choose to do the two above at the same time. However, it is a very time-consuming job. If an attacker intends to look for a specific vulnerability in the network, he will prefer scanning a list of hosts for a specific port, to see which hosts are running with that vulnerable application.

It is feasible to implement a mechanism to acknowledge the administrator when port scan activities are found. The keys in this solution are "User Defined Alert" and a user-defined program. A firewall rule is created to look for traffic going to a list of ports. For instance, if we don't have ftp service running on the hosts, we put on TCP Port 21. The rule is specified with "User Defined Alert" in the Track column. When an attacker tries to connect to any ports on the list, the rule will trigger the user-defined program, and parse the corresponding firewall log entry to it. The user-defined program can be developed to store the firewall log entry in a centralized repository, and send an alert to the network administrator for further investigation.

I have implemented the above by a Perl script. The Perl script is based on the shell script alert.sh published in the paper "Intrusion Detection for FW-1" written by Lance Spitzner. The Perl script is used as the user-defined program, which handles the log recording and e-mail notification for port scan activities.

The software configuration of the firewall machine is as follows:

- Check Point Firewall-1 NG Management Console
- Windows 2000 Server with SP2
- ActivePerl 5.6.1.631
  (*ActivePerl is a Perl interpreter available for Linux, Solaris and Windows. It is covered by its community license, which states that it can be used for commercial or non-commercial purposes without charge. It can be downloaded at http://www.activeperl.com.*)

The Check Point Firewall-1 NG Firewall Module can be installed at the same machine or other machines with other operating systems.

A rule for detecting port scan activities is necessary. The following rule is inserted at the top of the rulebase:

The ports specified in the rule will include all the ports that your hosts are not listening or should not be exposed to the external network, and are usually considered interesting to attackers, like the following:

- 7 (*echo*)
- 19 (*chargen*)
- 79 (*finger*)
- 111 (*PortMapper*)
- 139 (*NetBIOS-SSN*)
- 12345 (*NetBus*)
- 27374 (*SubSeven*)
- 31337 (*Black Orifice*)

The "Top 10 Target Ports" report at Dshield.org often provides you a good start to consider which ports should be put on the list.

The rule specified that any traffic originated from external network to internal servers, with the ports specified in the rule will all be dropped. The rule will then trigger a user-defined program.

The Perl script which, will be trigged when the rule is matched, is named "`scanalert.pl`". The source code of this program is attached at the end of this assignment. Basically it does the followings:

1. When the program is triggered by the firewall, it reads the standard input STDIN from Check Point Firewall. The standard input STDIN is similar to the following log entry:

```
15:34:20 drop   152.102.194.154 >EL90BC0 useralert product VPN-1 &
FireWall-1 src 152.102.194.95 s_port 2487 dst 152.102.194.154 service
ftp proto tcp rule 1
```

2. The program extracts the important fields from the log entry, which include the time, source IP address, destination IP address, source port and destination port, and put them into the `alert.log` file for later investigation.

3. The program counts the number of port scans originated from the source IP address in the `alert.log` file.

4. The program checks the source IP address against the addresses stored in the `attacker.log` file. If the source IP address does not exist in the file, it is appended to the file. Therefore regardless of the number of ports are scanned, there will only be one entry per unique source recorded in this file. It facilitates tracking of who scanned your network.

5. The program checks whether the number of port scans originated from the source IP address is greater than the scan limit threshold (default is 50). If it is the case, the program issues a command to the firewall to block the source IP address for 15 minutes, and send an e-mail to notify the network administrator about that.

6. The program checks whether the number of port scans originated from the source IP address is smaller than the mail limit threshold (default is 5). If it is the case, the program sends an e-mail to notify the network administrator about the port scan activity and the detailed information about that scan. The threshold is used to prevent network administrator e-mail account from being bombed by alert e-mails.

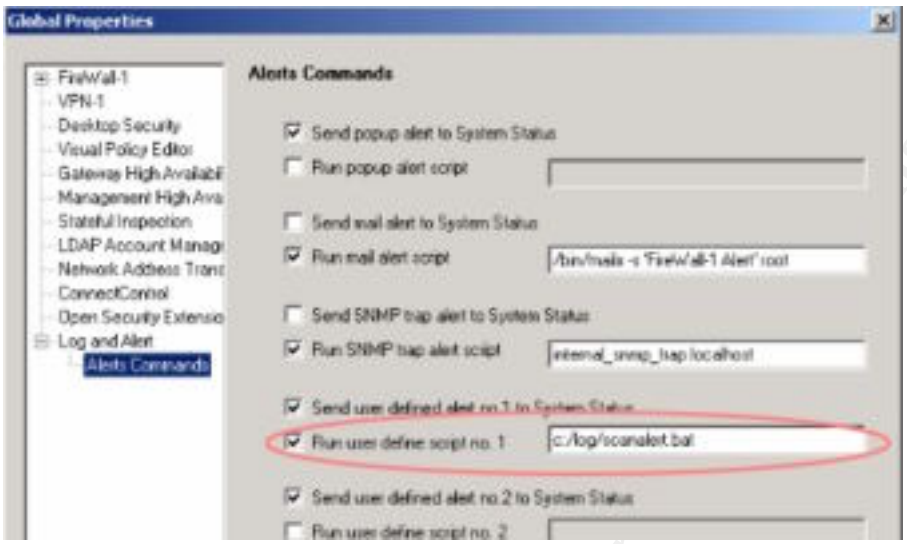
Because Perl script running under ActivePerl cannot handle I/O redirection (in ActivePerl document, it states that it is a limitation of Windows NT/2000), so the `pl2bat` utility distributed with ActivePerl is used to convert the Perl script into a batch file. What the utility does is to tag some Win32 batch language to the front of the script so that the system calls the Perl interpreter on the file. In this way, the standard input STDIN generated from Check Point can be probably read by the program.

To convert the Perl script to a batch file, we need to do the following:

```
C:\> pl2bat scanalert.pl
```

Then the corresponding batch file `scanalert.bat` will be created and located in the same directory with the Perl script.

As "User Defined Alert" is specified in the Track column for this rule, the program needed to be specified under Global Properties:



It should be noticed that the program is specified by "`c:/log/scanalert.bat`' but not "`c:\log\scanalert.bat`'. Check Point will not launch the program if it is specified in the latter way.

After completing all the configurations, install the policy on the firewall. From now on, when the ports specified in the rule is probed, the network administrator will receive an e-mail like the following:

```
Someone at IP address 152.102.194.99 is probably performing a port
scanning to your network. At most 5 e-mails regarding this IP address will
be sent to you. The following is the detailed information of this scan:

-------- Critical Information --------
Date: 22-12-2001
Time: 12:14:31
Source IP: 152.102.194.99
Source Port: 1590
Destination IP: 10.10.10.21
Service: echo

----- Check Point FW-1 Log Entry -----
12:14:31 drop   152.102.194.1 >EL90BC0 useralert product VPN-1 & FireWall-
1 src 152.102.194.99 s_port 1590 dst 10.10.10.21 service echo proto tcp
rule 1
```

By default, the network administrator will only receive 5 e-mails regarding the port scan activities originated from the same IP address.

The port scan activity will be recorded in the `alert.log` file in the following format:

```
22-12-2001, 12:14:31, 152.102.194.99, 10.10.10.21, 1590, echo
```

The information stored in the file can be used for later investigation and reporting. A simple batch program can be created to archive the `alert.log` and `attacker.log` files daily for these purposes.

If the number of port scans originated from the same IP address exceeds the threshold, the attacker IP address will be blocked for 15 minutes, and the administrator will receive an e-mail notice like the following:

```
The IP address 152.102.194.99 is blocked for 15 minutes because it has
exceeded the threshold (50) in scanning your network.
```

To conclude, by using a simple Perl script, a firewall can be easily customized to generate port scan alert and block the attacker when port scan activities are detected. The Perl script can be further improved to directly store the port scan log entries into a database for better archival and reporting. To cite a maxim of security, "Prevention is ideal, but detection is a must." When no network-based IDS are installed in the network, it will be a good practice for network administrator to drill into the firewall logs to look for malicious activities every day.

Reference:

1. Spitzner, Lance. "Intrusion Detection for FW-1"
   URL: http://www.enteract.com/~lspitz/intrusion.htm (22 December 2001).
2. DShield.org. "Top 10 Target Ports"
   URL: http://www.dshield.org/topports.html
3. Cole, Eric. "Hackers Beware"
   New Riders, August 2001: 63-102.
4. Northcutt, Stephen "Network Intrusion Detection – An Analyst's Handbook 2nd Ed"
   New Riders, September 2000
5. Scambray, Joel "Hacking Exposed 3rd Ed"
   McGraw-Hill Professional Publishing, September 2001

Appendix - Source code of scanalert.pl

```
#
# scanalert.pl
# Written By: Roland Lee
# Date: 5 February 2001
#

#########################################################
# User customization
#########################################################

#The FromAddress variable is the e-mail address of the firewall management
console
$FromAddress = 'Firewall';

#The ToAddress variable is the e-mail address of where the alert should be
sent to
$ToAddress = 'YOUR_E-MAIL_ADDRESS';

#The MailServer variable is the IP address of the Mail Server
$MailServer = 'YOUR_MAILSERVER_IP';

# The AttackerLog file keeps track of the IP addresses of the attackers
$AttackerLog = 'c:\log\attacker.log';

# The AlertLog file contains the detailed log information of the scan
$AlertLog  = 'c:\log\alert.log';

# The maximum number of e-mail alerts
$MailLimit = 5;

# The maximum number of scan before blocking the attacker
$ScanLimit = 50;

# This command is to block the attacker for 900 seconds
$block = $ENV{SystemRoot}.'\fw1\5.0\bin\fw sam -t 900 -i src';

#########################################################
# The main program starts at here
#########################################################

# Read the standard input from Check Point Firewall-1 NG
$message = <STDIN>;

# Manipulate the standard input and put the information to the corresponding
variables
@message = split / src /, $message;
@message1 = split /[ ]+/, @message[0];
@message2 = split /[ ]+/, @message[1];

# ICMP traffic will generate a log entry in different format
$icmp = 1 if ( $message =~ / proto icmp / );

if ( $icmp == 1 ) {
   $time = @message1[0];
```

```
   $source = @message2[0];
   $destination = @message2[2];
   $sport = @message2[4];
   $service = @message2[5]." ".@message2[6]." ".@message2[7]."
".@message2[8];
}
else {
   $time = @message1[0];
   $source = @message2[0];
   $destination = @message2[4];
   $sport = @message2[2];
   $service = @message2[6];
}

# Check Point won't pass the date information, we need to get it here. Format
is DD-MM-YYYY.
use Time::localtime;
$year = localtime->year()+1900;
$mon = localtime->mon()+1;
$mday = localtime->mday();
$date = $mday.'-'.$mon.'-'.$year;

# Record the log information into the alert.log file
open file, ">>$AlertLog";
print file  "$date, $time, $source, $destination, $sport, $service\n";
close file;

# Count the number of times the source has scanned us
open file, "<$AlertLog";
while (@alert = <file>){
   chomp @alert;
   $NumberOfScan = grep {/$source/} @alert;
}
close file;

# Check whether the source has scanned us before, if not, log it in the
attacker.log file
open file, "<$AttackerLog";
while (@attacker = <file>){
   chomp @attacker;
   $NumberOfAttack = grep {/$source/} @attacker;
}
close file;

if ( $NumberOfAttack == 0 ) {
   open file, ">>$AttackerLog";
   print file "$source\n";
   close file;
}

# Block the attacker if the number of scan exceeds the limit and send an e-
mail to the admin
if ( $NumberOfScan%$ScanLimit == 0 ) {

   system "$block $source\n";

   use Net::SMTP;
```

```
   $smtp = Net::SMTP->new($MailServer);
   $smtp->mail($FromAddress);
   $smtp->to($ToAddress);

   $smtp->data();
   $smtp->datasend("To: $ToAddress\n");
   $smtp->datasend("Subject: ***** Block Notice ***** \n\n");
   $smtp->datasend("The IP address $source is blocked for 15 minutes because
it has ");
   $smtp->datasend("exceeded the threshold ($ScanLimit) in scanning your
network.\n");
   $smtp->dataend();

   $smtp->quit;

}

# Send an e-mail to acknowledge the administrator if the number of scan does
not exceed the limit
if ($NumberOfScan <= $MailLimit) {

   use Net::SMTP;

   $smtp = Net::SMTP->new($MailServer);
   $smtp->mail($FromAddress);
   $smtp->to($ToAddress);

   $smtp->data();
   $smtp->datasend("To: $ToAddress\n");
   $smtp->datasend("Subject: ***** Port Scan Alert ***** \n\n");
   $smtp->datasend("Someone at IP address $source is probably performing a
port scanning to your network. ");
   $smtp->datasend("At most $MailLimit e-mails regarding this IP address will
be sent to you. ");
   $smtp->datasend("The following is the detailed information of this
scan:\n\n");
   $smtp->datasend("-------- Critical Information --------\n");
   $smtp->datasend("Date: $date\n");
   $smtp->datasend("Time: $time \n");
   $smtp->datasend("Source IP: $source\n");
   $smtp->datasend("Source Port: $sport\n");
   $smtp->datasend("Destination IP: $destination\n");
   $smtp->datasend("Service: $service\n\n");
   $smtp->datasend("----- Check Point FW-1 Log Entry -----\n");
   $smtp->datasend("$message \n");
   $smtp->dataend();

   $smtp->quit;

}
```

Assignment 2 – Network Detects

Analysis tool and log format

Snort is a freeware intrusion detection system (IDS). More information of the software can be obtained at http://www.snort.org. All of the log files collected in this assignment were from Snort. The snort log format is as below:

```
[**] [1:247:1] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
02/18-18:25:16.345098 207.246.136.130:80 ->
MY.CORP.NET.243:12754
TCP TTL:43 TOS:0x0 ID:29663 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xFBF9848 Ack: 0x5BAE08EB Win: 0x40B0 TcpLen: 20
```

- The first line is the IDS alert signature.
- The second line is optional. It contains the classification and the priority of the alert.
- The third line contains the date and time, source IP address, source port, destination address and service port.
- The forth line contains various IP header fields, including protocol, time-to-live (TTL), TOS, IP Identification Number (ID), IP Header Length (IpLen), datagram length (DgmLen), fragment flags and other fragment offset information (if any).
- The fifth line contains TCP flags, sequence number, acknowledgement number, window size and TCP Header Length (TcpLen).
- The lines (optional) after the fifth line contain the whole or part of the datagram.

Source of network detects

The network detects were collected by a Snort IDS machine attached to my company external network. The Snort IDS was installed on a Red Hat 7.2 Linux machine. My company network was connecting to the Internet through an ADSL modem, and my company employed 2-tier firewalls to protect the internal network. The first tier firewall is a Cisco PIX 515 firewall and the second tier firewall is a Nokia IP330 Check Point Firewall-1. The ADSL modem and the first tier Cisco PIX were connecting to the same VLAN on a Cisco switch, and the Snort machine was attached to the SPAN port for that VLAN on the switch. Therefore all network traffic coming from and going out to the Internet would pass through the Snort IDS machine.

Here are the IP addresses of the devices:

- MY.CORP.NET.242 – External interface of Cisco PIX firewall
- MY.CORP.NET.243 – NAT address of all internal users accessing the Internet
- MY.CORP.NET.245 – IP address of the Snort IDS machine
- MY.CORP.NET.253 – External interface of Nokia IP330 Check Point Firewall-1
- MY.CORP.NET.254 – Internal interface of Cisco PIX firewall

The subnet mask for all of them is 255.255.255.248. The IP address range MY.CORP.NET.240 – MY.CORP.NET.247 located between the ADSL modem and Cisco PIX firewall, and the IP

address range MY.CORP.NET.248 – MY.CORP.NET.255 located between the Cisco PIX firewall and the Check Point Firewall-1.

Analysis of network detects

1. DDOS mstream client to handler

1.1 Source of Trace:

The source of trace was obtained from my company network.

1.2 Detect was generated by:

This detect was generated by Snort intrusion detection system 1.8.3.

```
[**] [1:247:1] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
02/18-18:25:16.345098 207.246.136.130:80 -> MY.CORP.NET.243:12754
TCP TTL:43 TOS:0x0 ID:29663 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xFBF9848 Ack: 0x5BAE08EB Win: 0x40B0 TcpLen: 20
[**] [1:247:1] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
02/18-18:25:16.355098 207.246.136.130:80 -> MY.CORP.NET.243:12754
TCP TTL:43 TOS:0x0 ID:29664 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xFBF9DAC Ack: 0x5BAE08EB Win: 0x40B0 TcpLen: 20
[**] [1:247:1] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
02/18-18:25:16.635098 207.246.136.130:80 -> MY.CORP.NET.243:12754
TCP TTL:43 TOS:0x0 ID:30192 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xFBFA310 Ack: 0x5BAE08EB Win: 0x40B0 TcpLen: 20
[**] [1:247:1] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
02/18-18:25:16.635098 207.246.136.130:80 -> MY.CORP.NET.243:12754
TCP TTL:43 TOS:0x0 ID:30194 IpLen:20 DgmLen:775 DF
***AP**F Seq: 0xFBFADD8 Ack: 0x5BAE08EB Win: 0x40B0 TcpLen: 20
```

The corresponding Snort rule that triggered this alert:

```
ddos.rules:alert tcp $EXTERNAL_NET any -> $HOME_NET 12754 (msg:"DDOS
mstream client to handler"; content: ">"; flags: A+; reference:cve,CAN-
2000-0138; classtype:attempted-dos; sid:247; rev:1;)
```

1.3 Probability the source address was spoofed:

The source address was probably not spoofed.

1.4 Description of attack:

This is probably a false positive. The first tier Cisco PIX firewall performed network address translation (NAT) for the internal users and used TCP port 12754 to connect to the web server at 207.246.136.130.

1.5 Attack mechanism:

"msteam" is one of the popular seven major recognized distributed denial of service (DDoS) tools discovered for executing DDoS attacks. DDoS tools in general are capable of producing high magnitude packet flooding denial of service attacks. The "mstream" tool is found capable of producing a severe denial of service condition against one or more victim sites, including sites being used as hosts for portions of a "mstream" DDoS network. The "mstream" tool consists of a handler and a client (attacker) portion. Remote control of the "mstream" handler is accomplished via a TCP connection to port 6723/tcp, or 15104/tcp, or 12754/tcp, etc.

Whenever Cisco PIX perform NAT for internal clients, it will initiate a connection to the target host by a high port.

1.6 Correlations:

A lookup of the source 207.246.136.130 at Dshield.org gives the following information:

```
IP Address: 207.246.136.130
Hostname: cbird10.sextracker.com
DShield Profile:
```

Country:	US
Contact E-mail:	edmond@FLYINGROC.COM
Total Records against IP:	575
Number of targets:	107
Date Range:	-

```
Whois:
Accretive Technology Group, Inc. (NET-ANI-001)
   2001 Sixth Ave, Ste 3302
   Seattle, WA 98121
   US

   Netname: ANI-001
   Netblock: 207.246.128.0 - 207.246.159.255
   Maintainer: ACCR
```

From the hostname of the IP address, it seems to be a porn web site. Obviously, an internal user was surfing on a porn web site, or perhaps the web site was connected by an advertisement link automatically.

- CERT Incident Note IN-2000-05
  http://www.cert.org/incident_notes/IN-2000-05.html

- The "mstream" distributed denial of service attack tool
  http://staff.washington.edu/dittrich/misc/mstream.analysis.txt

1.7 Evidence of active targeting:

There was no specific target as this alert is a false positive.

1.8 Severity:

If this is a real attack and not a false positive,

Criticality = 2 (user desktop system)
Lethality = 1 (unlikely to succeed)
System Countermeasures = 5 (all systems were applied with up-to-date patches)
Network Countermeasures = 5 (restrictive firewall, external connection can't be initiated to internal host directly)

Severity = (Criticality + Lethality) – (Network Countermeasures + System Countermeasures)
= (2 + 1) – (5 + 5)
= -7

1.9 Defensive recommendations

No defensive measure is necessary as this is a false positive.

If it is the case that a company employee was browsing web sites with porn stuff, establish a clear policy to state that it is an illegal act in the company. Besides, consider deploying a URL filtering system to block all access to those porn web sites.

1.10 Multiple choice test question

Which of the following is a common communication port between client and handler for "mstream"?

A. 80
B. 161
C. 1080
D. 12754

Answer: D

2. SYN FIN Scan

2.1 Source of Trace:

The source of trace was obtained from my company network.

2.2 Detect was generated by:

This detect was generated by Snort intrusion detection system 1.8.3.

```
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-02:51:00.885098 209.61.158.39:21 -> MY.CORP.NET.242:21
TCP TTL:20 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x4D901DBD Ack: 0x37D65117 Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-02:51:00.905098 209.61.158.39:21 -> MY.CORP.NET.243:21
TCP TTL:20 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x4D901DBD Ack: 0x37D65117 Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-02:51:00.915098 209.61.158.39:21 -> MY.CORP.NET.245:21
TCP TTL:20 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x4D901DBD Ack: 0x37D65117 Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-02:51:00.995098 209.61.158.39:21 -> MY.CORP.NET.253:21
TCP TTL:20 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x4D901DBD Ack: 0x37D65117 Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-10:37:57.065098 210.5.19.166:22 -> MY.CORP.NET.242:22
TCP TTL:34 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x32BFC843 Ack: 0x659D42FF Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-10:37:57.075098 210.5.19.166:22 -> MY.CORP.NET.243:22
TCP TTL:34 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x32BFC843 Ack: 0x659D42FF Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-10:37:57.115098 210.5.19.166:22 -> MY.CORP.NET.245:22
TCP TTL:34 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x32BFC843 Ack: 0x659D42FF Win: 0x404 TcpLen: 20
[**] [111:13:1] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]
02/18-10:37:57.275098 210.5.19.166:22 -> MY.CORP.NET.253:22
TCP TTL:34 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x32BFC843 Ack: 0x659D42FF Win: 0x404 TcpLen: 20
```

These alerts are triggered by the Snort portscan preprocessor.

2.3 Probability the source address was spoofed:

The source addresses were probably not spoofed, as the purpose of this scan is for reconnaissance so the attacker needs to receive the response.

2.4 Description of attack:

Two attackers were trying to find out whether ftp and ssh services were running in my company by stealth scans. The stealth scan is identified by the following special characteristics:

- SYN and FIN flags are both set
- IP Identification Number (IPID) is always 39426
- Source port is always the same as destination port
- Windows size is always 0x404 (1028)

2.5 Attack mechanism:

The purpose of the scan is for reconnaissance. The attackers are interested to know whether ftp and ssh services are running in my network.

2.6 Correlations:

- According to the GCIA assignment of Roland Gerlach (http://www.giac.org/practical/Roland_Gerlach_GCIA.html), this SYN FIN scan was generated through the old version of synScan, which can be downloaded at http://www.psychoid.lam3rz.de/synscan.html. He has mentioned that the new version of synScan 1.6a has already fixed the same IP ID problem. Besides, the flags are set to SYN only, not SYN and FIN again.

  A scan performed by synScan 1.6a captured by tcpdump:

```
17:27:04.885614 MY.CORP.INT.160.ftp > MY.CORP.INT.95.ftp: S [tcp sum ok] 1255773
590:1255773590(0) win 9628 (ttl 126, id 40157, len 40)
17:27:26.905614 MY.CORP.INT.160.ftp > MY.CORP.INT.98.ftp: S [tcp sum ok] 5165333
20:516533320(0) win 14047 (ttl 141, id 29044, len 40)
17:27:39.035614 MY.CORP.INT.160.ftp > MY.CORP.INT.100.ftp: S [tcp sum ok] 144366
9210:1443669210(0) win 34002 (ttl 126, id 3649, len 40)
17:27:44.395614 MY.CORP.INT.160.ftp > MY.CORP.INT.101.ftp: S [tcp sum ok] 397127
228:397127228(0) win 26548 (ttl 126, id 42660, len 40)
17:27:57.965614 MY.CORP.INT.160.ftp > MY.CORP.INT.104.ftp: S [tcp sum ok] 133499
2518:1334992518(0) win 56406 (ttl 137, id 30913, len 40)

Note: MY.CORP.INT is the internal address of my company network.
```

  From the above capture, the source port and the destination port are still the same for every scan. In this scan, port 21 ftp service is targeted. However, in every scan, the window size is different (can't prove it is really randomized or not), and the IP ID varies.

- A lookup of the source 209.61.158.39 at Dshield.org gives the following information:

```
IP Address: 209.61.158.39
Hostname: aquariumventures.com
DShield Profile:
```

Country:	US
Contact E-mail:	hostmaster@rackspace.com
Total Records against IP:	984
Number of targets:	984
Date Range:	2002-02-16 to 2002-02-17

```
Whois:
Rackspace.com (NETBLK-RSPC-NET-2)
   112 East Pecan St.
   San Antonio, TX 78205
   US

   Netname: RSPC-NET-2
   Netblock: 209.61.128.0 - 209.61.191.255
   Maintainer: RSPC
```

- A lookup of the source 210.5.19.166 at Dshield.org gives the following information:

```
IP Address: 210.5.19.166
Hostname: 210.5.19.166
DShield Profile:
```

Country:	AU
Contact E-mail:	UNASIGNED
Total Records against IP:	21
Number of targets:	21
Date Range:	2002-02-17 to 2002-02-17

```
Whois:
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255
```

2.7 Evidence of active targeting:

All external real IP addresses of my company network were scanned by the attackers. It is clear that the attackers were scanning a range of IP addresses for the services.

2.8 Severity:

Criticality = 5 (firewalls)
Lethality = 1 (stealth port scan)
System Countermeasures = 5 (ftp and ssh services were disabled)
Network Countermeasures = 5 (restrictive firewall, traffic to ftp and ssh services were not allowed and blocked by firewall)

Severity = (Criticality + Lethality) – (Network Countermeasures + System Countermeasures)
= (5 + 1) – (5 + 5)
= -4

2.9 Defensive recommendations:

No defensive measure is necessary as ftp and ssh services are not running in my company network. If ftp and ssh services are running in the firewall boxes, ensure that they are applied with the up-to-date patches. Besides, apply firewall rule to restrict access to the ftp and ssh services on the firewall boxes itself.

2.10 Multiple choice test question:

Which of the following is a valid combination of TCP flags?

```
A. ****PR**
B. ******SF
C. ***AP***
D. ***A*R*F
```

Answer: C

3. WEB-IIS CodeRed v2 root.exe

3.1 Source of Trace:

The source of trace was obtained from my company network.

3.2 Detect was generated by:

This detect was generated by Snort intrusion detection system 1.8.3.

```
[**] WEB-IIS CodeRed v2 root.exe access [**]
02/16-07:14:43.698009 202.64.166.20:4938 -> MY.CORP.NET.242:80
TCP TTL:126 TOS:0x0 ID:5742 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x951D690D  Ack: 0xD9FEF0EA  Win: 0x4248  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 72 6F 6F  GET /scripts/roo
74 2E 65 78 65 3F 2F 63 2B 64 69 72 20 48 54 54  t.exe?/c+dir HTT
50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77  P/1.0..Host: www
0D 0A 43 6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63  ..Connnection: c
6C 6F 73 65 0D 0A 0D 0A                          lose....
[**] WEB-IIS CodeRed v2 root.exe access [**]
02/16-07:15:31.758009 202.64.166.20:4938 -> MY.CORP.NET.242:80
TCP TTL:126 TOS:0x0 ID:37282 IpLen:20 DgmLen:112 DF
***AP**F Seq: 0x951D690D  Ack: 0xD9FEF0EB  Win: 0x4248  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 72 6F 6F  GET /scripts/roo
74 2E 65 78 65 3F 2F 63 2B 64 69 72 20 48 54 54  t.exe?/c+dir HTT
50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77  P/1.0..Host: www
0D 0A 43 6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63  ..Connnection: c
6C 6F 73 65 0D 0A 0D 0A                          lose....
[**] WEB-IIS CodeRed v2 root.exe access [**]
02/16-07:17:07.848009 202.64.166.20:4938 -> MY.CORP.NET.242:80
TCP TTL:126 TOS:0x0 ID:35009 IpLen:20 DgmLen:112 DF
***AP**F Seq: 0x951D690D  Ack: 0xD9FEF0EB  Win: 0x4248  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 72 6F 6F  GET /scripts/roo
74 2E 65 78 65 3F 2F 63 2B 64 69 72 20 48 54 54  t.exe?/c+dir HTT
50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77  P/1.0..Host: www
0D 0A 43 6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63  ..Connnection: c
6C 6F 73 65 0D 0A 0D 0A                          lose....
```

The corresponding Snort rule that triggered this alert:

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

3.3 Probability the source address was spoofed:

The source address was probably not spoofed, as TCP connection is required between the attacker and the target host.

3.4 Description of attack:

An attacker was trying to find out whether my company hosts were infected with "Code Red II" worm. The "Code Red II" worm is self-propagating malicious code that exploits a known vulnerability in Microsoft IIS servers. An infected host will leave open to attackers. Anyone can execute arbitrary commands within the `LocalSystem` security context in the infected systems through crafted URLs.

3.5 Attack mechanism:

The attack mechanism is as follows:

- The "Code Red II" worm attempts to connect to TCP port 80 on a randomly chosen host assuming that a web server will be found. Upon a successful connection to port 80, the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit the buffer overflow in the Indexing Service.
- The same exploit is sent to each of the randomly chosen hosts due to the self-propagating nature of the worm. However, there are varied consequences depending on the configuration of the host which receives this request.
- Affected targets include unpatched Windows 2000 servers running IIS 4.0 or 5.0 with Indexing Service installed. Unpatched Windows NT servers running IIS 4.0 or 5.0 with Indexing Server 2.0 installed and unpatched Cisco 600-series DSL routers will stop function properly.

If the exploit is successful, the worm begins executing on the victim host:

- Checks to see if it has already infected this system by verifying the existence of the Code Red II atom. If the worm finds this atom it sleeps forever. Otherwise it creates this atom and continues the infection process.
- Checks the default system language, and spawns threads for propagation. If the default system language is "Chinese (Taiwanese)" or "Chinese (PRC)", 600 threads will be spawned to scan for 48 hours. Otherwise, 300 threads will be created which will scan for 24 hours.
- Copies `%SYSTEM%\CMD.EXE` to `root.exe` in the IIS scripts and MSADC folders. Placing `CMD.EXE` in a publicly accessible directory may allow an intruder to execute arbitrary commands on the compromised machine with the privileges of the IIS server process.
- Creates a Trojan horse copy of explorer.exe and copies it to `C:\` and `D:\`. The Trojan horse `explorer.exe` calls the real `explorer.exe` to mask its existence, and creates a virtual mapping which exposes the `C:` and `D:` drives.
- On systems not patched against the "Relative Shell Path" vulnerability, the Trojan horse copy of `explorer.exe` will run every time a user logs in. In this fashion, certain pieces of the worm's payload have persistence even after a reboot of the compromised machine.

After the host is infected, anyone can execute arbitrary command at the victim host through crafted URLs like the followings:

- `http://victim_host/scripts/root.exe?/c+dir`
- `http://victim_host/MSADC/root.exe?/c+dir`

The above two URLs will return the directory information of `C:` drive of the victim host. The attacker from 202.64.166.20 was using the first crafted URL to obtain the `C:` drive directory information of my company's host to test whether the host was infected with Code Red II worm or not.

Please notice that this maybe also a scan from Nimda. However, as there were no other URLs related to cmd.exe being scanned (this is a signature of scanning activity of Nimda worm), likely the scan was not because of Nimda.

3.6 Correlations:

- A lookup of the source 202.64.166.20 at Dshield.org gives the following information:

```
IP Address: 202.64.166.20
Hostname: ip20.dyn166.pacific.net.hk
DShield Profile:
```

Country:	HK
Contact E-mail:	charlesl@pacific.net.hk
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
Asia Pacific Network Information Center (APNIC2)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK
   Netblock: 202.0.0.0 - 203.255.255.255
   Maintainer: AP
```

Probably, the IP address was assigned temporarily to a home user who has an ADSL connection to the Internet. The user was a script kiddies and using a downloaded malicious program to search for Code Red II infected hosts on the Internet.

- CERT Incident Note IN-2001-09 – "Code Red II:" Another Worm Exploiting Buffer Overflow in IIS Indexing Service DLL: http://www.cert.org/incident_notes/IN-2001-09.html

- CERT Advisory CA-2001-26 Nimda Worm

http://www.cert.org/advisories/CA-2001-26.html

3.7 Evidence of active targeting:

The target IP address belonged to the external interface of the first tier Cisco PIX firewall. It was very strange, as the external interface should not be listening to TCP port 80. Because of this, I tried to make a connection from my home to the external interface to the Cisco PIX firewall through this port and I could get connected. The web management daemon of Cisco PIX was listening and accepting connections at its external interface. I then talked to my company network administrator and he confirmed that it was a configuration fault. It seemed the target was randomly selected by the attacker.

3.8 Severity:

Criticality = 5 (firewall)
Lethality = 1 (unlikely to succeed because Cisco PIX is not vulnerable to Code Red II)
System Countermeasures = 5 (hardware appliance firewall)
Network Countermeasures = 2 (permissive firewall, web traffic is allowed to go inside)

Severity = (Criticality + Lethality) – (Network Countermeasures + System Countermeasures)
= (5 + 1) – (5 + 2)
= -1

3.9 Defensive recommendations:

No defensive measure is necessary in my company network as there were no Microsoft IIS web servers and Cisco DSL routers. The only thing should be done immediately is to ensure the web management daemon was not listening at the external interface of the Cisco PIX firewall.

3.10 Multiple choice test question:

Which is the most indicative that it is a scanning for back doors left behind by "Code Red II"?

```
10/10-10:15:31.000000 w1.x1.y1.z1:2186 -> w2.x2.y2.z2.:80
TCP TTL:126 TOS:0x0 ID:37282 IpLen:20 DgmLen:112 DF

47 45 54 20 2F 73 63 72 69 70 74 73 2F 72 6F 6F  GET /scripts/roo
74 2E 65 78 65 3F 2F 63 2B 64 69 72 20 48 54 54  t.exe?/c+dir HTT
50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77  P/1.0..Host: www
0D 0A 43 6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63  ..Connnection: c
6C 6F 73 65 0D 0A 0D 0A                          lose....
```

A. GET /scripts/root.exe
B. Source Port is 2186
C. Destination Port is 80
D. IP ID is 37282

Answer: A

4. WEB-CGI csh access

4.1 Source of Trace:

The source of trace was obtained from my company network.

4.2 Detect was generated by:

This detect was generated by Snort intrusion detection system 1.8.3.

```
[**] [1:862:2] WEB-CGI csh access [**]
[Classification: Attempted Information Leak][Priority: 2]
02/16-04:46:58.658009 128.42.44.98:61876 -> MY.CORP.NET.242:80
TCP TTL:125 TOS:0x0 ID:22015 IpLen:20 DgmLen:331 DF
***AP*** Seq: 0x58F6DC64 Ack: 0x7F34048F Win: 0x2058 TcpLen: 20
```

The corresponding Snort rules that triggered these alerta:

```
web-cgi.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-
CGI csh access";flags: A+; uricontent:"/csh"; nocase; reference:cve,CAN-
1999-0509;classtype:attempted-recon; sid:862; rev:2;)
```

4.3 Probability the source address was spoofed:

The source address was probably not spoofed as the attacker needed to establish TCP connection to the web server and obtain the response.

4.4 Description of attack:

An attacker is trying to access csh in the PIX server's CGI bin directory. If csh is accessible, then the attacker can execute any command the interpreters can execute on that server.

4.5 Attack mechanism:

The vulnerability lies on the fact that certain Unix shells may reside in the CGI bin directory of a Unix web server. A Unix shell can act as the user interface to the system, and it handles inputs from the user and the pass them to the kernel for processing. The attacker can try to access the some popular shell programs such as csh, sh and bash in the CGI bin directory. If the attacker succeeds, he might be able to interact with the system just as any authorized user would be able to do. If the attacker can gain root privileges, he can even do whatever to the system.

The web server of Cisco PIX is not susceptible to this attack, as it doesn't contain any shell program.

4.6 Correlations:

- A lookup of the source 128.42.44.98 at Dshield.org gives the following information:

```
IP Address: 128.42.44.98
Hostname: 128.42.44.98
DShield Profile:
```

Country:	US
Contact E-mail:	hostmaster@RICE.EDU
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
Rice University (NET-RICE-NET)
   Office of Networking and Planning
   Houston, TX 77251-1892
   US

   Netname: RICE-NET
   Netblock: 128.42.0.0 - 128.42.255.255
```

- CERT Advistory CA-1996-11 Interpreters in CGI bin Directories
  http://www.cert.org/advisories/CA-1996-11.html

4.7 Evidence of active targeting:

Again, the target IP address is the external interface of the 1st tier Cisco PIX firewall. Because of a mis-configuration fault, the web management daemon of the PIX firewall was listening at its external interface. It seems the target was randomly selected by the attacker.

4.8 Severity:

Criticality = 5 (firewall)
Lethality = 1 (unlikely to succeed as no csh in firewall)
System Countermeasures = 5 (hardware appliance firewall)
Network Countermeasures = 2 (permissive firewall, web traffic is allowed to go inside)

Severity = (Criticality + Lethality) – (Network Countermeasures + System Countermeasures)
= (5 + 1) – (5 + 2)
= -1

4.9 Defensive recommendations:

No defensive measure is necessary in my company network as there were no external web servers in my company network. The only thing should be done immediately is to ensure the web management daemon was not listening at the external interface of the Cisco PIX firewall.

4.10 Multiple choice test question:

Which of the following(s) can be used to execute arbitrary commands on a Web server system if located in the CGI bin directory?

A. Perl
B. csh
C. cmd.exe
D. Tcl
E. All of the above

Answer: E

5. WEB-FRONTPAGE shtml.dll access

5.1 Source of Trace:

The source of trace is from my company network.

5.2 Detect was generated by:

This detect was generated by Snort intrusion detection system 1.8.3.

```
[**] WEB-FRONTPAGE shtml.dll access [**]
02/15-02:07:05.618009 61.15.10.198:51314 -> MY.CORP.NET.242:80
TCP TTL:125 TOS:0x0 ID:9009 IpLen:20 DgmLen:564 DF
***AP*** Seq: 0x7A43B5EA  Ack: 0xE1D4ED74  Win: 0x1EB0  TcpLen: 20
50 4F 53 54 20 2F 5F 76 74 69 5F 62 69 6E 2F 73  POST /_vti_bin/s
68 74 6D 6C 2E 64 6C 6C 2F 73 65 61 72 63 68 2E  html.dll/search.
68 74 6D 6C 20 48 54 54 50 2F 31 2E 31 0D 0A 41  html HTTP/1.1..A
63 63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69 66  ccept: image/gif
[ truncated ]
```

The corresponding Snort rule that triggered this alert:

```
web-frontpage.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80
(msg:"WEB-FRONTPAGE shtml.dll access"; uricontent: "/_vti_bin/shtml.dll";
nocase; flags:A+; reference:arachnids,292; classtype:web-application-
activity; sid:940; rev:3;)
```

5.3 Probability the source address was spoofed:

The source address was probably not spoofed as the attacker needed to establish TCP connection to the web server and obtain the response.

5.4 Description of attack:

An attacker can obtain a web server physical path simply by making an invalid html request through `shtml.dll`. `shtml.dll` is a component in Microsoft FrontPage 2000 Server Extensions.

5.5 Attack mechanism:

The Microsoft FrontPage 2000 Server Extensions are a set of programs installed on a web server that support administering, authoring and browsing a Frontpage-extended web site. The extension `shtml.dll` handles user interactions with web forms and must be accessible to the users of a web site. An attacker can gain the information of the web server path by a simple request through `shtml.dll`. This can be done by requesting an invalid html page just like the following:

```
http://victim_host/_vti_bin/shtml.dll/anything_invalid.htm
```

The victim host with old version of `shtml.dll` will return something like the following:

```
Cannot open "D:\Inetpub\virtuals\powerasp\anything_invalid.htm": no such file or folder.
```

From the above returned information, the attacker can know the physical path of the web server.

5.6 Correlations:

- A lookup of the source 61.15.10.198 at Dshield.org gives the following information:

```
IP Address: 61.15.10.198
Hostname: cm61-15-10-198.hkcable.com.hk
DShield Profile:
```

Country:	-
Contact E-mail:	-
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
Asia Pacific Network Information Center (NETBLK-APNIC2)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC3
   Netblock: 61.0.0.0 - 61.255.255.255
   Maintainer: AP
```

  Probably, the IP address was assigned temporarily to a home user who has an ADSL connection to the Internet.

- Why to upgrade to: Front Page 2000 Server Extensions 1.2
  http://www.securityfocus.com/archive/1/68331

- Other attacks related to shtml.dll – IIS 5.0 cross site scripting vulnerability - Georgi Guniski security advisory #19, 2000

http://www.guninski.com/iis50shtml.html

- Other attacks related to shtml.dll – ISS X-Force Database frontpage-ext-shtml-multiple-dos (4899)
  http://www.iss.net/security_center/static/4899.php

5.7 Evidence of active targeting:

Again, the target IP address is the external interface of the 1st tier Cisco PIX firewall. Because of a mis-configuration fault, the web management daemon of the PIX firewall was listening at its external interface. It seems the target was randomly selected by the attacker.

5.8 Severity:

Criticality = 5 (firewall)
Lethality = 1 (unlikely to succeed)
System Countermeasures = 5 (hardware appliance firewall)
Network Countermeasures = 2 (permissive firewall, web traffic is allowed to go inside)

Severity = (Criticality + Lethality) – (Network Countermeasures + System Countermeasures)
= (5 + 1) – (5 + 2)
= -1

5.9 Defensive recommendations:

No defensive measure is necessary in my company network as there were no external web servers in my company network. The only thing should be done immediately is to ensure the web management daemon was not listening at the external interface of the Cisco PIX firewall.

5.10 Multiple choice test question:

What is the most possible intention of an attacker by requesting the following URL?

http://www.victim.com/_vti_bin/shtml.dll/an_html_do_not_exist.html

A. The attacker is performing a buffer overflow attack to the web server
B. The attacker is searching for the shtml.dll file
C. The attacker is mapping out the physical path of the web server
D. The attacker is locating files in _vti_bin

Answer: C

Assignment 3 – "Analyze This" Scenario

Security Audit Report for GIAC University

Executive Summary

A security audit was performed for GIAC University by its invitation. Five consecutive days of Snort log files were provided by the university, and they were used for detailed analysis in this security audit exercise. These log files included alert log files, scan log files, and out-of-spec (OOS) log files.

After detailed analysis, the followings needed your immediate action:

- The default community string "public" for SNMP devices should be changed. Besides, all SNMP traffic should be bounded to internal network. SNMP traffic coming from external network must be dropped at the firewall. Besides, to protect from the new SNMP implementation vulnerabilities, update patches should be downloaded from the vendor web sites and applied to the devices immediately.

- If real-time broadcast is not allowed to come into your network, your firewall should block all the traffic from the sites which offers this kind of service. Besides, your firewall should block all the traffic originated from UDP port 0 and destined to UDP port 0. Though these traffics may not be harmful, they are not normal traffic and may cause abnormalities in your host systems.

- Perform thorough checking on the internal hosts MY.NET.153.45, MY.NET.151.63, MY.NET.153.171, MY.NET.153.144, MY.NET.153.185, MY.NET.153.196 and MY.NET.150.143 to make sure that they are not installed with any unauthorized or malicious software. Any unauthorized or malicious software can become loopholes in your network. It seems that KaZaA, a peer-to-peer file sharing software, is actively running in your network.

- Block external host 66.38.185.141 as it was actively transmitting Red Worm to your internal hosts. Perform thorough checking on all internal destination hosts related to the Red Worm alert to ensure that they were not infected. Besides, ensure the software LPRng, rpc-statd, wu-ftpd and BIND installed in all Linux boxes in the internal network are updated.

- Perform thorough checking of virus / Trojan on all Microsoft IIS web servers. Ensure that they are properly patched to protect from the IIS Unicode attack.

The followings are recommended to implement in your network:

- As lpd printer daemon is running inside the network, if the machine is a Linux box or a BSD variant, ensure the printer daemon patched to the update version. Besides, ensure that your firewall block all the port 515 access from external network.

- Install software metering software, like Microsoft SMS and CA AimIT to assist in preventing unauthorized software from running at your desktop computers and recording the software that the computers run.

- Install server anti-virus software to ensure all the critical servers are protected from virus and Trojan attack.

- Install desktop anti-virus software to ensure all the desktop computers are protected from virus and Trojan attack.

- Block outgoing ICMP fragment reassembly time exceeded error messages at the router to protect from internal host discovery by attackers.

- Block hosts 193.144.127 and 195.77.24.2 as they were actively scanning your network.

Introduction

The Snort log files of the five consecutive days in the period 21 Jan 2002 – 25 Jan 2002 were used for analysis. The log files included alert log files, scan log files, and out-of-spec (OOS) log files. The names of the log files are as follows:

Alert	Scan	OOS
alert.020221	scans.020221	oos_Jan.21.2002
alert.020222	scans.020222	oos_Jan.22.2002
alert.020223	scans.020223	oos_Jan.23.2002
alert.020224	scans.020224	oos_Jan.24.2002
alert.020225	scans.020225	oos_Jan.25.2002

By using Snortsnarf, the alert log files were analyzed and processed. The number of alerts, number of sources and destinations corresponding to each alert are shown below:

Rank	Signature	# Alerts	# Sources	# Dests
1	**Connect to 515 from inside**	**31425**	**82**	**1**
2	**SNMP public access**	**25657**	**15**	**140**
3	**spp_http_decode: IIS Unicode attack detected**	**18668**	**94**	**368**
4	**MISC Large UDP Packet**	**16804**	**18**	**7**
5	**INFO MSN IM Chat data**	**5189**	**72**	**71**
6	**spp_http_decode: CGI Null Byte attack detected**	**3716**	**13**	**25**
7	**High port 65535 udp – possible Red Worm – traffic**	**3622**	**77**	**120**
8	**ICMP Router Selection**	**1701**	**139**	**1**
9	**ICMP Echo Request CyberKit 2.2 Windows**	**1486**	**4**	**5**
10	**ICMP Fragment Reassembly Time Exceeded**	**1116**	**19**	**31**
11	Null scan!	1021	61	5
12	Watchlist 000220 IL-ISDNNET-990517	802	23	7
13	ICMP Echo Request BSDtype	728	5	7
14	SMB Name Wildcard	700	54	50
15	ICMP Echo Request L3retriever Ping	432	25	7
16	FTP DoS ftpd globbing	375	6	4
17	ICMP Echo Request Windows	266	10	23
18	WEB-IIS view source via translate header	194	6	2
19	ICMP Destination Unreachable (Communication Administratively Prohibited)	164	1	10
20	NMAP TCP ping!	121	10	4
21	ICMP Destination Unreachable (Host Unreachable)	112	1	35
22	WEB-MISC Attempt to execute cmd	106	10	7
23	SCAN Proxy attempt	91	14	5
24	ICMP Echo Request Nmap or HPING2	80	7	3
25	Incomplete Packet Fragments Discarded	75	7	3
26	INFO FTP anonymous FTP	69	6	20
27	EXPLOIT NTPDX buffer overflow	68	13	7
28	Possible trojan server activity	58	5	5
29	INFO Inbound GNUTella Connect request	57	51	2
30	INFO Possible IRC Access	56	13	13
31	ICMP Destination Unreachable (Protocol Unreachable)	52	2	3
32	MISC traceroute	47	3	3
33	WEB-CGI scriptalias access	42	3	1
34	SCAN Synscan Portscan ID 19104	34	33	2

35	WEB-IIS _vti_inf access	23	12	2
36	WEB-FRONTPAGE _vti_rpc access	22	12	2
37	INFP Inbound GNUTella Connect accept	21	2	20
38	ICMP traceroute	13	7	5
39	Port 55850 tcp – Possible myserver activity – ref. 010313-1	12	4	4
40	SCAN FIN	11	4	2
41	TCP SRC and DST outside network	11	4	8
42	SUNRPC highport access!	9	2	2
43	Attempted Sun RPC high port access	6	3	5
44	WEB-MISC compaq nsight directory traversal	6	5	5
45	Queso fingerprint	6	5	2
46	INFO Outbound GNUTella Connect accept	5	5	2
47	EXPLOIT x86 setuid 0	5	3	4
48	High port 65535 tcp – possible Red Worm – traffic	4	3	3
49	SYN-FIN scan!	4	2	2
50	TFTP – External UDP connection to internal tftp server	4	1	1
51	WEB-MISC 403 Forbidden	4	2	3
52	EXPLOIT x86 NOOP	3	3	3
53	INFO – Possible Squid Scan	3	2	2
54	EXPLOIT x86 setgid 0	3	3	3
55	ICMP SRC and DST outside network	3	1	1
56	ICMP Source Quench	3	1	1
57	Back Orifice	2	2	2
58	ICMP Echo Request Cisco Type.x	2	1	1
59	Probable NMAP fingerprint attempt	2	1	1
60	Port 55850 udp – Possible myserver activity – ref. 010313-1	1	1	1
61	TFTP – Internal UDP connection to external tftp server	1	1	1
62	MISC Large ICMP packets	1	1	1
63	WEB-MISC http directory traversal	1	1	1
64	MISC source port 53 to <1024	1	1	1
65	RFB – Possible WinVNC – 010708-1	1	1	1
66	INFO Napster Client Data	1	1	1
67	RPC udp traffic contains bin sh	1	1	1

In total, 67 different alerts were found in the 5 consecutive days of alert log files. The top 10 alerts, in terms of number of occurrences, found in the alert log files are selected for further in-depth analysis as they represent 95% of the total number of alerts.

Detailed Analysis of the top 10 alerts

1. connect to 515 from inside

31425 alerts with this signature were found. The earliest was found at 08:43:45 on 01/22/2002 and the latest was found at 17:24:03 on 01/25/2002.

◈ Sources triggering this attack signature

There were 82 sources triggered this attack signature. The top 20 are listed below:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.114	3011	3770	1	31
MY.NET.153.202	2384	2385	1	2
MY.NET.153.118	2285	2297	1	2
MY.NET.153.112	1720	1763	1	4
MY.NET.153.113	1645	1747	1	10
MY.NET.153.137	1538	1546	1	2
MY.NET.153.119	1232	1281	1	9
MY.NET.153.121	1139	1177	1	7
MY.NET.153.123	1060	1243	1	13
MY.NET.153.173	894	894	1	1
MY.NET.153.204	866	868	1	3
MY.NET.153.124	686	1277	1	21
MY.NET.153.120	671	946	1	21
MY.NET.88.148	651	656	1	2
MY.NET.153.140	618	656	1	2
MY.NET.153.160	604	604	1	2
MY.NET.153.105	553	627	1	7
MY.NET.153.143	504	962	1	20
MY.NET.153.110	447	1176	1	18
MY.NET.153.111	439	447	1	2

◈ Destinations receiving this attack signature

There was only 1 destination received this attack signature:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.150.198	31425	31425	82	82

◈ Brief description of the attack

This alert represents an attempt to connect to port 515, the lpd, which is the printer daemon of a Unix host within the internal network.

There were advisories released regarding vulnerabilities for the LPR service, for many distributions of Linux and for the BSD variants. The LPRng port, versions prior to 3.6.24, contains a potential vulnerability, which may allow root compromise from both local and remote

systems. The vulnerability is due to incorrect usage of the syslog function. Local and remote users can send string-formatting operators to the printer daemon to corrupt the daemon's execution, potentially gaining root access.

As all of the sources were within the internal network, there is no sign of port probing on port 515 from outsiders. Obviously, the host MY.NET.150.198 is a Unix box running with the lpd printer daemon. If not, it is better to check whether the computers' printer settings are incorrect.

◈ Defensive recommendations

This is not an attack so no defensive measure is necessary. However, as lpd printer daemon is running inside the network, if the machine is a Linux box or a BSD variant, it is better to check whether the version of the Printer Daemon is vulnerable or not. Patch the printer daemon to the update version. Besides, ensure that your firewall block all the port 515 access from external network.

◈ Correlation

- CERT Advisotry CA-2000-22 Input Validation Problems in LRRng: http://www.cert.org/advisories/CA-2000-22.html
- Increased probes to TCP port 515: http://www.sans.org/newlook/alerts/port515.htm
- Thomas Rodriguez GIAC Certified Intrusion Analyst Practical Assignment: http://www.giac.org/practical/Thomas_Rodriguez_GCIA.doc

2. SNMP public access

25657 alerts with this signature were found. The earliest was found at 00:01:23 on 21 Jan 2002 and the latest was found at 23:59:59 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 15 sources triggered this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.70.177	10603	10627	25	25
MY.NET.88.240	7128	7128	1	1
MY.NET.150.198	2850	2850	102	102
MY.NET.150.41	2256	2256	1	1
MY.NET.153.220	1538	1538	1	1
MY.NET.186.10	670	670	1	1
MY.NET.150.245	259	259	1	1
MY.NET.84.155	246	246	17	17
MY.NET.183.11	46	46	5	5
MY.NET.150.49	20	1493	5	9

MY.NET.111.197	19	19	9	9
MY.NET.86.22	15	15	9	9
MY.NET.111.139	4	4	2	2
MY.NET.150.112	2	22	1	2
MY.NET.150.179	1	43	1	3
MY.NET.104.200	42	42	1	1
MY.NET.150.243	37	41	4	7
MY.NET.150.84	34	38	3	6
MY.NET.88.240	33	33	1	1

◈ Destinations receiving this attack signature

There were 140 destinations triggered this attack signature. The top 20 are listed below:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.150.195	7152	7155	6	8
MY.NET.152.109	4073	4073	4	4
MY.NET.5.37	1733	1733	1	1
MY.NET.5.96	1716	2041	1	28
MY.NET.5.128	1683	1683	1	1
MY.NET.5.127	1669	1669	1	1
MY.NET.5.249	1352	1409	2	6
MY.NET.151.114	1323	1327	3	6
MY.NET.5.141	929	963	1	7
MY.NET.153.219	699	703	4	7
MY.NET.5.92	638	648	1	6
MY.NET.5.83	338	780	1	13
MY.NET.5.97	217	220	1	2
MY.NET.5.95	61	96	1	8
MY.NET.150.14	57	57	3	3
MY.NET.151.52	51	51	6	6

◈ Brief description of the attack

This attack represents an attempt to send or receive SNMP messages using the standard community string “public”. The community string is a token passed from the manager to the agent, you might want to think of it as a remarkably weak password (it is passed around in clear-text). The device you're talking to will use the community name you give it to decide what data you should have access to. The default community is "public," which should theoretically give access only to safe information. In practice, vendors have an unfortunate tendency to allow *all* SNMP to the community "public"; this may include the ability to get information you might not want given out to anybody in the universe, like the names of all the accounts on your machine, or worse yet it may include the ability to do sets on arbitrary variables.

Therefore, by using common community strings such as “public” and “private”, attackers can gain information about devices on your network by polling them for their SNMP properties. Attackers can also modify devices by sending SNMP commands.

As the sources and the destinations associated with this alert were all within the internal network, this is not an external attack. It is unlikely that an attacker was using SNMP to gain information

of your network. The problem is that the standard community string "public" is being used in the SNMP devices.

There are a lot of SNMP devices found in the network. Recently, numerous vulnerabilities have been reported in multiple vendors' SNMP implementations. These vulnerabilities may allow unauthorized privileged access, denial-of-service attacks or cause unstable behavior.

◈ Defensive recommendations

The default community string "public" should be changed. Besides, all SNMP traffic should be bounded to internal network. SNMP traffic coming from external network must be dropped at the firewall. To block SNMP access, block traffic to ports 161 and 162 for tcp and udp. In addition, if you are using Cisco, block udp for port 1993. Besides, to protect from the new SNMP implementation vulnerabilities, update patches should be downloaded from the vendor web sites and applied to the devices immediately. SANS provided a free self-test tool to find out the affected SNMP devices in your network. The tool can be downloaded at http://www.sans.org/snmp/tool.php.

◈ Correlation

- SNMP Vulnerabilities FAQ:
  http://www.cert.org/tech_tips/snmp_faq.html

- CERT Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementations of the SNMP:
  http://www.cert.org/advisories/CA-2002-03.html

- SANS Flash Alert: Widespread SNMP Vulnerability:
  http://www.sans.org/alerts/SNMP.php

3. spp_http_decode: IIS Unicode attack detected

18668 alerts with this signature were found. The earliest was found at 00:51:11 on 21 Jan 2002 and the latest was found at 23:47:05 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 94 sources triggered this attack signature. The top 20 sources are listed below:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.171	1998	2416	85	97
MY.NET.152.14	1764	1836	12	15
MY.NET.153.141	1608	1608	13	13
MY.NET.153.193	1371	1378	41	46
MY.NET.153.185	980	1227	40	46
MY.NET.153.197	720	797	35	38

MY.NET.153.110	712	1176	16	18
MY.NET.153.114	644	3770	27	31
MY.NET.152.168	586	971	12	15
MY.NET.153.124	583	1277	19	21
MY.NET.153.115	579	694	29	31
MY.NET.151.108	464	465	10	11
MY.NET.153.143	458	962	19	20
MY.NET.153.144	449	468	20	21
MY.NET.153.125	432	672	13	15
MY.NET.153.151	384	384	24	24
MY.NET.153.152	327	328	11	12
MY.NET.153.200	318	318	12	12
MY.NET.88.245	312	315	3	4
MY.NET.153.177	289	340	3	7

◈ Destinations receiving this attack signature

There were 368 destinations received this attack signature. The top 20 are listed below.

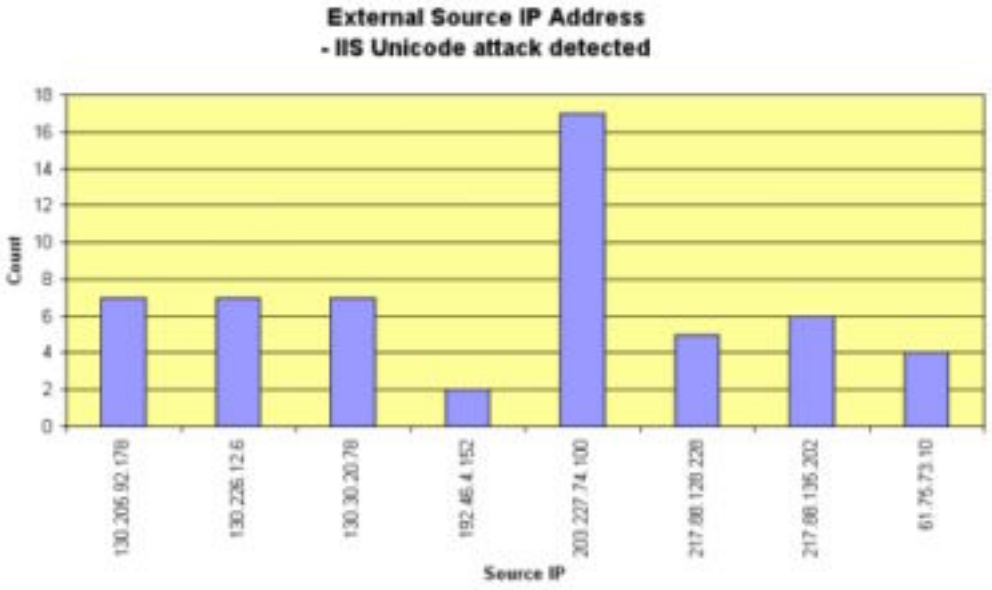
Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
211.115.213.202	2166	2166	3	3
64.12.184.141	1249	1249	12	12
211.115.213.207	697	697	2	2
211.32.117.27	647	647	4	4
211.32.117.37	626	626	8	8
211.32.117.26	569	569	1	1
64.12.180.21	477	477	7	7
211.233.28.80	465	465	6	6
211.32.117.31	434	434	4	4
216.33.148.250	362	362	4	4
205.188.138.85	296	296	5	5
199.244.218.42	270	270	5	5
211.32.117.227	262	262	5	5
211.233.29.209	257	257	2	2
207.200.86.66	208	208	1	1
211.233.29.252	207	207	10	10
207.200.89.225	201	201	6	6
211.176.60.147	195	195	1	1
211.32.117.228	187	187	8	8
211.174.58.36	183	183	1	1

◈ Brief description of the attack

Due to a canonicalization error in Microsoft IIS 4.0 and 5.0, a particular type of malformed URL could be used to access files and folders that lie anywhere on the logical drive that contains the web folders. This would potentially enable a malicious user who visited the web site to gain additional privileges on the machine. Specifically, it could be used to gain privileges commensurate with those of a locally logged-on user. Gaining these permissions would enable the malicious user to add, change or delete data, run code already on the server, or upload new code to the server and run it.

The request would be processed under the security context of the IUSR_machinename account, which is the anonymous user account for IIS. Within the web folders, this account has only privileges that are appropriate for untrusted users. However, it is a member of the Everyone and Users groups and, as a result, the ability of the malicious user to access files outside the web folders becomes particularly significant. By default, these groups have execute permissions to most operating system commands, such as `cmd.exe`, and this would give the malicious user the ability to cause widespread damage. If the permissions of Everyone and Users groups have been proactively removed on the server, or the web folders are hosted on a different drive from the operating system, it would be at significantly less risk from the vulnerability.

Among the 94 sources, 8 of them are located externally:



The source 203.227.74.100 initiated the highest number of attacks to the network. An IP Info search of the address 203.227.74.100 at Dshield.org gives the following information:

```
IP Address: 203.227.74.100
Hostname: -
DShield Profile:
```

Country:	KR
Contact E-mail:	master_AT_hawk.daesung.co.kr (bounced)
Total Records against IP:	192
Number of targets:	112
Date Range:	2002-02-13 to 2002-02-14

```
Whois:
Asia Pacific Network Information Center (APNIC2)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU
```

```
Netname: APNIC-CIDR-BLK
Netblock: 202.0.0.0 - 203.255.255.255
Maintainer: AP
```

Among the 368 destinations, 8 of them are located in internal network. The alerts associated with these 8 hosts may indicate that they are targets of the attackers. The 8 hosts are web servers and maybe vulnerable to this attack.

The two internal hosts MY.NET.5.249 and MY.NET.5.96 received the highest number of this alerts. They should be paid with special attention. It is also found that hosts MY.NET.150.83, MY.NET.5.141, MY.NET.5.249, MY.NET.5.92, MY.NET.5.95, MY.NET.5.96 and MY.NET.5.97 received the alert "*WEB-MISC Attempt to execute cmd*". It is a sign that the attackers were trying to utilize this IIS Unicode vulnerability to gain system access through `cmd.exe`. However, this IIS Unicode alert is subject to a considerably high false positives rate. Very often, you may see false positives with sites that use cookies with urlencoded binary data, or if you are scanning port 443 and picking up SSLencrypted traffic. Your own internal users normal surfing can trigger these alerts. Netscape in particular has been known to trigger them. Having the packet dumps through a network sniffer, like tcpdump, is a good way to tell for sure if you have a real attack on your hands. Besides, you can also investigate the Web server log files to perform a thorough investigation on the suspected malformed URLs.

◈ Defensive recommendations

Ensure that all the web servers are properly patched to protect from the IIS Unicode attack.

For Microsoft IIS 4.0, download the patch at:
http://www.microsoft.com/ntserver/nts/downloads/critical/q269862/default.asp)

For Microsoft IIS 5.0, download the patch at:
http://www.microsoft.com/windows2000/downloads/critical/q269862/default.asp

The IIS 4.0 patch can be installed on systems running Windows NT 4.0 Service Packs 5 and 6a. The IIS 5.0 patch can be installed on systems running either Windows 2000 or Service Pack 1. The patch is already included in Windows 2000 Service Pack 2.

A convenient way to check the web server patch level is to use the hfnetchk utility provided by Microsoft. It is a command-line tool that enables an administrator to check the patch status of Windows NT 4.0, Windows 2000, and Windows XP machines. The current version is 3.3 and it can be downloaded at http://www.microsoft.com/downloads/release.asp?releaseid=31154.

◈ Correlation

- A search of the IP address 203.227.74.100 at Google returns three related web pages containing that IP. From them, I found that in a message posted on 21 Dec 2001 in a guest book of a Korea web site, the IP address is related to the e-mail address lee0406@hanmail.net.
  http://db3.protectsite.net/babykims/gbook/CrazyGuestbook.cgi?db=guest

- Microsoft Security Bulletin MS00-78:
  http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-078.asp

- Common Vulnerabilities and Exposures CVE2000-0884:
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884

- CERT Vulnerability Note VU#111677 – Microsoft IIS 4.0 / 5.0 vulnerable to directory traversal via extended Unicode in url (MS00-78):
  http://www.kb.cert.org/vuls/id/111677

- Snort FAQ:
  http://www.snort.org/docs/faq.html

- Thomas Rodriguez GIAC Certified Intrusion Analyst Practical Assignment:
  http://www.giac.org/practical/Thomas_Rodriguez_GCIA.doc

4. MISC Large UDP Packet

16804 alerts with this signature were found. The earliest was found at 10:52:24 on 22 Jan 2002 and the latest was found at 11:54:54 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 18 sources triggered this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)

63.210.47.81	5007	5008	1	1
63.250.208.34	4970	4970	1	1
211.172.232.21	1540	1609	1	1
211.202.0.47	1012	1012	1	1
210.181.96.14	905	905	1	1
216.106.166.212	767	767	1	1
62.253.169.246	662	664	1	1
211.233.70.163	615	615	1	1
217.15.64.179	313	313	1	1
211.174.63.108	241	241	1	1
211.233.70.162	219	219	1	1
211.233.70.165	217	217	1	1
216.106.166.164	142	142	1	1
211.174.63.106	104	105	1	1
64.124.157.58	53	53	1	1
211.233.70.172	26	26	1	1
68.55.200.56	10	11	1	1
63.250.209.34	1	1	1	1

◈ Destinations receiving this attack signature

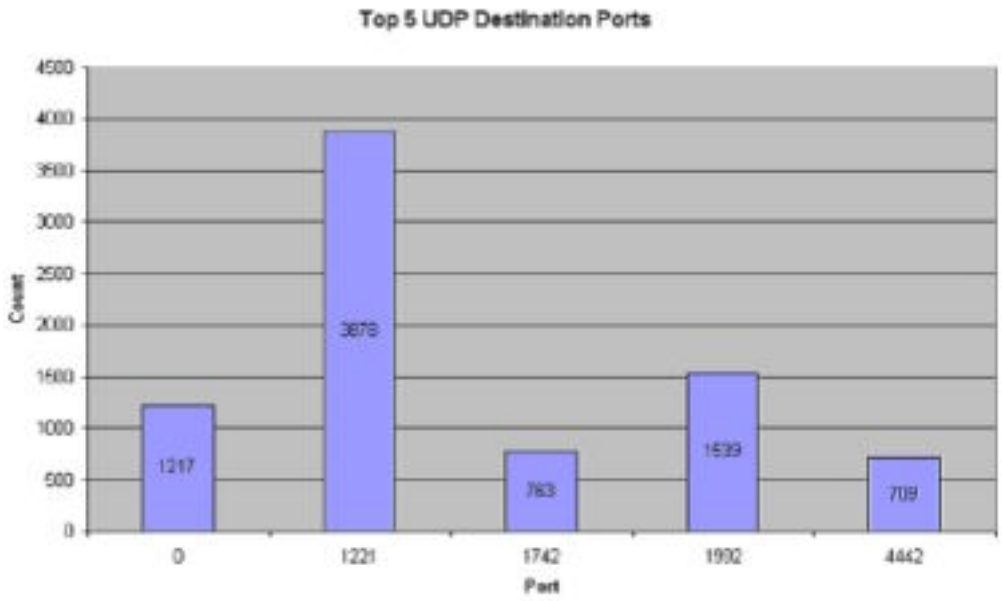
There were 7 destinations received this attack signature:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.153.45	5916	5917	3	3
MY.NET.151.63	4971	4974	2	4
MY.NET.153.171	2262	2297	4	14
MY.NET.153.144	1540	1618	1	5
MY.NET.153.185	1390	1398	5	10
MY.NET.153.196	715	958	2	7
MY.NET.150.143	10	13	1	3

◈ Brief description of the attack

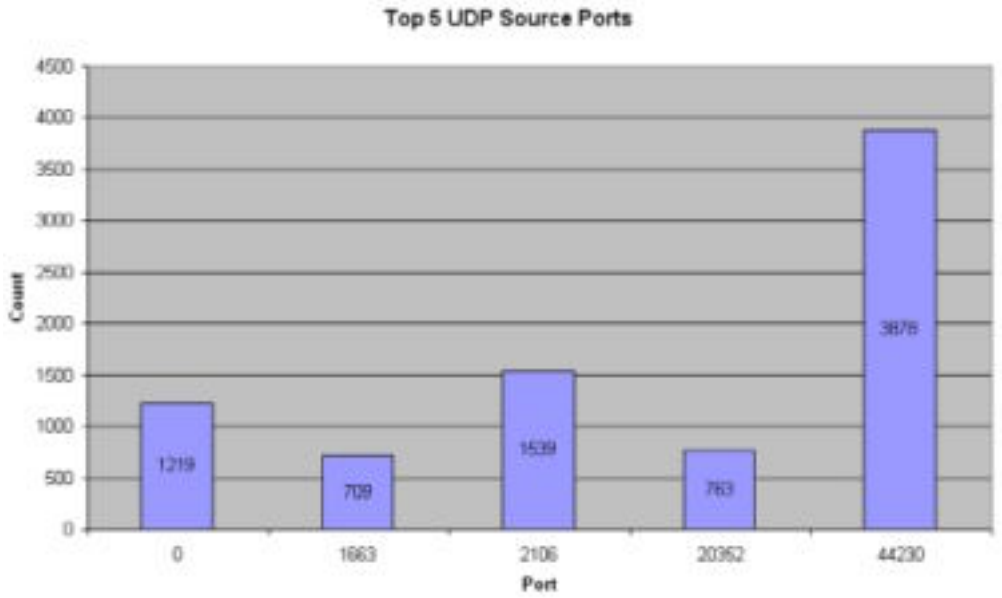
Large UDP packet with size over 4000 bytes will trigger this alert. Many real-time applications, and those that require no reliability, use unicast UDP instead of TCP for data transfer.

414 different UDP destination ports were found. The top 5 UDP destination ports are shown below:



The highest number of appearance, UDP port 1221, is associated with SweetWARE Apps. Not much information can be found on SweetWARE, maybe it is related to http://www.sweetware.com. The SweetWARE Company develops and markets software for the food industry. It was founded in 1989 to do custom software development for bakeries. The second highest, UDP port 1992 is associated with service IPsendmsg and Cisco STUN Priority 3 port. UDP port 1742 is associated with 3Com-nsd. UDP port 4442 is associated with Saris. However, I found no information on Saris on the Internet and other sources. From my search, the ports are not associated with any known attacks.

406 different UDP source ports were found. The top 5 UDP source ports are shown below:

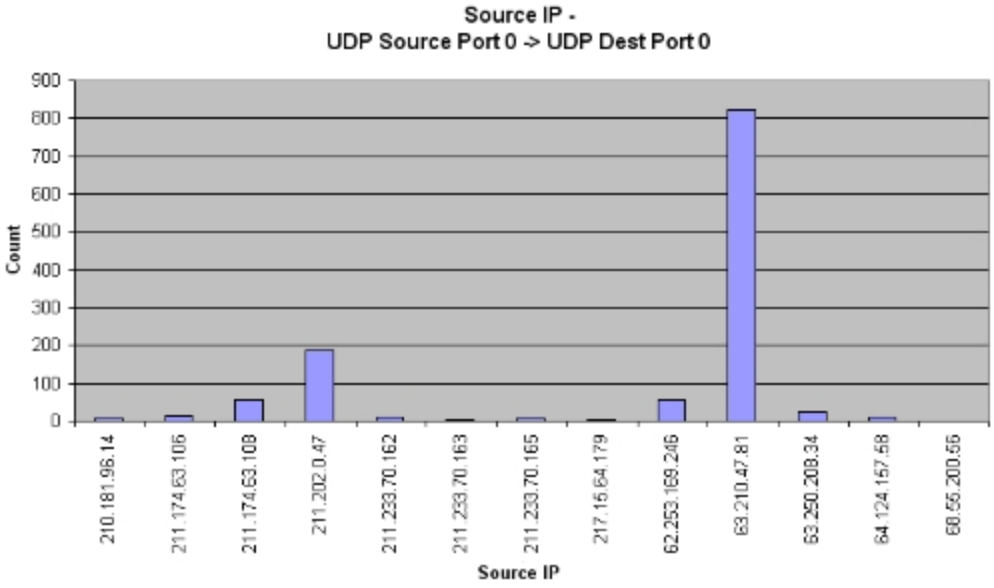


Port 1663 is associated with netview-aix-3, 2106 is associated with MZAP, while ports 20352 and 44230 are not associated with any services / attacks.

There are numerous mysterious alerts found. They are those with UDP traffic originated from port 0 to port 0:

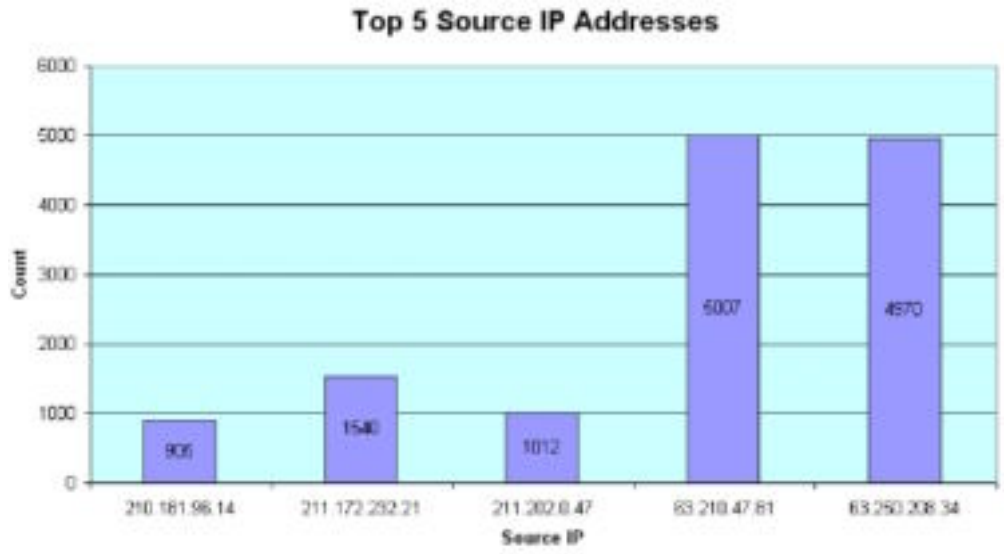
01/22-10:53:18.355134 [**] MISC Large UDP Packet [**] 68.55.200.56:0 -> MY.NET.150.143:0

In total 1206 of this alert is found. 13 of the 18 sources had initiated this kind of traffic to the internal hosts, where the source 63.210.47.81 has initiated the highest number of traffic among them:



From the above, over half of the sources had initiated this kind of traffic. I suspect that there should be something wrong, and the source never sends this kind of traffic to the internal hosts. A network sniffer, for example, tcpdump should be used for further investigation.

The top 5 sources triggered this alerts are shown below graphically:



A lookup of the source 63.210.47.81 at Dshield.org gives the following information:

```
IP Address: 63.210.47.81
Hostname: unknown.Level3.net
DShield Profile:
```

Country:	US
Contact E-mail:	spamtool@level3.com
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
Level 3 Communications, Inc. (NETBLK-LEVEL4-CIDR)
   1450 Infinite Drive
   Louisville, CO 80027
   US

   Netname: LEVEL4-CIDR
   Netblock: 63.208.0.0 - 63.215.255.255
   Maintainer: LVLT
```

Using a web browser to access the server address 63.210.47.81 indicates that the server is offering online multimedia service. However, I cannot find out what exactly the service is being offered. (I have already tried to connect by RealPlayer and Windows Media Player.)

A lookup of the source 63.250.208.34 at Dshield.org gives the following information:

```
IP Address: 63.250.208.34
Hostname: 63.250.208.34
DShield Profile:
```

Country:	US
Contact E-mail:	netops@broadcast.com
Total Records against IP:	1
Number of targets:	1
Date Range:	2002-02-01 to 2002-02-01

```
Whois:
Yahoo! Broadcast Services, Inc. (NETBLK-NETBLK2-YAHOOBS)
   2914 Taylor st
   Dallas, TX 75226
   US

   Netname: NETBLK2-YAHOOBS
   Netblock: 63.250.192.0 - 63.250.223.255
   Maintainer: YAHO
```

As the source IP address is owned by Yahoo! broadcast service, it is likely that someone was enjoying online broadcast service offered by Yahoo!.

◈ Defensive recommendations

If real-time broadcast is not allowed to come into your network, your firewall should block all the traffic from the sites which offers this kind of service. Besides, your firewall should block all the traffic originated from UDP port 0 and destined to UDP port 0. Though these traffics may not be related to any attacks, they are not normal traffic and may cause abnormalities in your host systems. Besides, you should perform a thorough checking on the seven destination hosts to make sure that they are not installed with any unauthorized or malicious software. Any unauthorized or malicious software can become loopholes in your network.

◈ Correlation

- No further information on the source IP addresses can be found on the Internet and other assignments.

5. INFO MSN IM Chat Data

5189 alerts with this signature were found. The earliest was found at 10:40:45 on 21 Jan 2002 and the latest was found at 18:49:01 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 72 sources triggered this attack signature. The top 20 sources are listed below:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.150.242	431	437	18	19
MY.NET.150.232	301	318	17	20
MY.NET.150.165	279	615	16	28
MY.NET.88.181	275	619	17	32
MY.NET.153.194	235	591	10	11
64.4.12.193	209	209	4	4
MY.NET.153.106	172	542	3	5
64.4.12.177	162	162	7	7
MY.NET.150.241	145	184	8	9

64.4.12.184	140	140	5	5
MY.NET.152.19	135	135	9	9
64.4.12.188	126	126	4	4
64.4.12.155	125	125	5	5
64.4.12.185	123	123	3	3
MY.NET.152.167	113	233	7	8
64.4.12.157	88	88	2	2
64.4.12.160	86	86	5	5
64.4.12.178	85	85	6	6
MY.NET.152.215	84	84	9	9
64.4.12.190	78	78	3	3

◈ Destinations receiving this attack signature

There were 71 destinations received this attack signature. The top 20 are listed below.

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.150.242	384	384	19	19
MY.NET.150.165	278	278	11	11
MY.NET.88.181	264	269	16	18
MY.NET.153.106	239	239	2	2
MY.NET.153.194	195	491	9	22
64.4.12.184	189	189	5	5
64.4.12.184	189	189	5	5
64.4.12.183	173	173	3	3
64.4.12.177	170	170	8	8
MY.NET.150.241	149	149	6	6
64.4.12.193	134	134	4	4
64.4.12.185	133	133	4	4
64.4.12.161	129	129	4	4
64.4.12.160	126	126	5	5
64.4.12.186	120	120	3	3
64.4.12.155	118	118	6	6

◈ Brief description of the attack

MSN Messenger is an instant messaging program that allows user to send messages or chat with several friends when they are online at once. It can allow online file transfer, application sharing, and voice chatting, etc. For Windows XP, a new version "Windows Messenger" is already bundled in the OS. The newest version offers more advance features like video conferencing (without the use of Netmeeting), remote desktop administration, etc.

The MSN messenger protocol is an ASCII based protocol. The first phase involves connecting to an MSN messenger server .In this case the MSN Messenger client shall connect to the server on port 1863.

For file transfer, both incoming and outgoing TCP connections use this range of ports: 6891 to 6900. This allows up to 10 simultaneous file transfers per sender.

For voice communication, the client establishes an outgoing TCP connection from port 6901. In the case of computer-to-computer communications, the call recipient also used TCP port 6901. All voice traffic uses UDP packets. The user's computer sends and receives UDP packets at port 6901.

A lookup of the source IP address 64.4.12.184 at Dshield.org gives the following information:

```
IP Address: 64.4.12.184
Hostname: msgr-sb35.msgr.hotmail.com
DShield Profile:
```

Country:	US
Contact E-mail:	icon@HOTMAIL.COM
Total Records against IP:	1
Number of targets:	1
Date Range:	2002-02-05 to 2002-02-05

```
Whois:
MS Hotmail (NETBLK-HOTMAIL)
   1065 La Avenida
   Mountain View, CA 94043
   US

   Netname: HOTMAIL
   Netblock: 64.4.0.0 - 64.4.63.255
```

This information indicates that the whole netblock 64.4.0.0/18 is owned by Hotmail. Hotmail (http://www.hotmail.com) is a free web mail company and is now owned by Microsoft. It is likely that the netblock 64.4.0.0/18 is used by Microsoft to provide mail and messaging service.

All the source and destination addresses associated with this alert are either internal address or external address fall in the range 64.4.0.0/18. Therefore the machines involved are either internal machines or messaging servers. It is obvious that some user desktop machines are installed with MSN Messenger and the users are using it to chat with their friends or colleagues online through the Internet. If any files are being transferred through MSN messenger, it is highly risky as files containing virus may transmit into the desktop computer and spreading to other computers in the internal network.

Recently, there is a new computer worm, known as "Cool Worm", "Menger" or "JS Exploit-Messenger" discovered. It could infect MSN Messenger users. Though it does little more than sending itself to other instant messaging users on a victim's address list, it is a sign that more exploits related to instant messaging software will be appearing soon.

◈ Defensive recommendations

To block MSN messenger traffic completely is nearly not feasible, as MSN messenger will revert to use TCP port 80 to continue communication, though functionality such as file transfer will be lost. At least, at the firewall gateway, block all the traffic directing to TCP port 1863 of the network 64.4.0.0/18. Ensure that all desktop machines are installed with anti-virus software.

Using of software metering software like Microsoft SMS and CA AimIT could prevent the program from running at your desktop computers.

◈ Correlation

- MSN Messenger – Guide for Network Administrators: http://messenger.msn.com/support/firewall.asp
- MSN Messenger Worm Marks Troubling Trend: http://www.ecommercetimes.com/perl/story/16355.html
- Symantec Advisory - JS.Menger.Worm: http://securityresponse.symantec.com/avcenter/venc/data/js.menger.worm.html

6. spp_http_decode: CGI Null Byte Attack

3716 alerts with this signature were found. The earliest was found at 11:52:09 on 22 Jan 2002 and the latest was found at 10:03:17 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 13 sources triggered this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.150.121	2982	2988	1	2
MY.NET.153.194	356	591	1	11
MY.NET.150.165	205	615	2	28
MY.NET.153.114	74	3770	1	31
MY.NET.153.171	60	2416	6	97
MY.NET.153.121	15	1177	3	7
MY.NET.153.197	7	797	2	38
MY.NET.153.193	5	1378	4	46
MY.NET.153.47	4	4	1	1
MY.NET.88.162	4	59	1	4
MY.NET.153.179	2	208	1	5
MY.NET.88.189	1	1	1	1
MY.NET.150.103	1	12	1	3

◈ Destinations receiving this attack signature

There were 25 destinations triggered this attack signature. The top 20 are listed below:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
216.241.219.14	2982	2982	1	1
209.143.193.79	356	356	1	1
205.226.241.199	203	203	1	1
144.266.116.29	74	74	1	1
216.33.88.53	43	199	1	3

216.115.102.40	13	13	1	1
211.169.246.50	6	6	1	1
208.184.29.210	5	20	1	1
63.210.31.72	4	6	1	1
216.203.49.219	4	6	1	1
17.254.3.22	4	4	1	1
209.157.71.37	4	4	1	1
205.188.180.57	2	19	1	5
208.184.29.190	2	7	1	1
211.169.246.29	2	2	1	1
208.184.29.130	2	3	1	1
205.188.180.25	2	49	1	8
203.199.93.4	1	1	1	1
205.158.62.59	1	1	1	1
211.169.246.3	1	1	1	1

◈ Brief description of the attack

"%00" is the hex value of a null byte. It can be used to fool a web application into thinking a different file type has been requested. By doing this, an attacker may be able to access system files in the web server.

All the sources were in your internal network and all the destinations were outside your network. As such, your site is probably not under this attack.

Very often, you may see false positives with sites that use cookies with urlencoded binary data, or if you are scanning port 443 and picking up SSLencrypted traffic. Your own internal users normal surfing can trigger these alerts. Netscape in particular has been known to trigger them. Having the packet dumps through a network sniffer, like tcpdump, is a good way to tell for sure if you have a real attack on your hands.

◈ Defensive recommendation

No defensive measure is necessary, as the attacks found were not targeted to your network.

◈ Correlation

- Fingerprinting Port 80 Attacks:
  http://www.cgisecurity.com/papers/fingerprint-port80.txt

- Snort FAQ:
  http://www.snort.org/docs/faq.html

7. High port 65535 udp – possible Red Worm - traffic

3622 alerts with this signature were found. The earliest was found at 03:04:10 on 21 Jan 2002 and the latest was found at 17:52:55 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 77 sources triggered this attack signature. The top 20 sources are listed below:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.6.48	809	810	36	36
MY.NET.6.49	793	798	45	45
MY.NET.6.50	435	437	32	33
MY.NET.6.52	375	399	1	1
66.38.185.141	375	399	1	1
MY.NET.6.60	89	89	35	35
MY.NET.6.53	80	82	26	27
216.107.173.149	66	74	4	4
216.106.172.155	65	76	3	3
216.106.172.148	57	60	3	3
216.106.173.148	52	53	4	4
216.106.173.147	51	66	3	4
216.106.172.149	43	46	3	4
MY.NET.6.45	33	33	20	20
216.106.172.147	33	33	4	4
66.77.13.122	22	23	1	1
216.106.172.156	21	22	2	3
MY.NET.60.43	20	20	15	15
216.106.172.157	19	19	3	3
63.146.181.119	18	18	1	1

◈ Destinations receiving this attack signature

There were 120 destinations received this attack signature. The top 20 are listed below:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.88.163	375	399	1	1
MY.NET.152.10	280	280	1	1
MY.NET.153.194	275	491	12	22
MY.NET.153.196	242	958	5	7
MY.NET.152.14	155	204	2	6
MY.NET.152.175	134	134	2	2
MY.NET.153.210	134	143	11	12
MY.NET.153.189	133	133	4	4
MY.NET.152.19	133	222	10	20
MY.NET.88.155	121	130	17	18
MY.NET.152.248	87	87	2	2
MY.NET.152.176	86	86	2	2
MY.NET.152.12	85	85	1	1
MY.NET.153.142	85	87	3	4
MY.NET.152.161	54	54	1	1
MY.NET.153.179	54	54	3	3
MY.NET.153.173	47	47	5	5
MY.NET.152.179	46	46	3	3
MY.NET.153.177	39	68	2	4
MY.NET.152.166	36	36	1	1

◈ Brief description of the attack

Adore is a worm that was originally called the Red Worm, which is similar to the Ramen and Lion worms. Adore scans the Internet checking Linux hosts to determine whether they are vulnerable to any of the following well-known exploits: LPRng, rpc-statd, wu-ftpd and BIND.

When a system is infected with Adore, the worm will replace the system binary (`ps`), with a trojaned version and moves the original to `/usr/bin/adore`. It installs the files in /usr/lib/lib, and then sends an email with information includes `/etc/ftpusers`, `ifconfig`, `ps -aux`, `/root/bash_history`, `/etc/hosts`, `/etc/shadow` to the e-mail addresses adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com and adore9001@sina.com.

Adore then set up a ping backdoor. By default, the ping backdoor sets the port to listen to UDP 65535, with a packet length of 77 bytes to watch for. When it sees this kind of packet it then sets a root shell to allow connections. It also sets up a cronjob in cron daily (which runs at 04:02 am local time) to run and remove all traces of its existence and then reboots your system. However, it does not remove the backdoor.

The detected traffic is quite abnormal. The UDP port 65535 should not frequently appear in network traffic, as the port is not associated with any known services. Maybe some attackers were probing the systems for the ping backdoor or even have successfully connected to it.

A lookup of the source IP address 66.38.185.141 at Dshield.org gives the following information:

```
IP Address: 66.38.185.141
Hostname: 141.185.38.66.gt-est.net
DShield Profile:
```

Country:	CA
Contact E-mail:	hostmaster@gt.ca
Total Records against IP:	8
Number of targets:	2
Date Range:	2002-02-09 to 2002-02-09

```
Whois:
GT Group Telecom Services Corp. (NETBLK-GROUPTELECOM-BLK-3)
   20 BAY STREET SUITE 700
   TORONTO, ON M5J 2N8
   CA

   Netname: GROUPTELECOM-BLK-3
   Netblock: 66.38.128.0 - 66.38.255.255
   Maintainer: GTGR
```

◈ Defensive recommendations

The software LPRng, rpc-statd, wu-ftpd and BIND installed in all Linux boxes in the internal network should be ensured updated. Besides, you should block for outbound emails to the 4

email addresses, i.e. adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com, adore9001@sina.com and block the website address http://go.163.com.

William Stearns has written a script Adorefind to detect the Adore worm and it can be downloaded at http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm

◈ Correlation

- No further information on the source IP addresses can be found on the Internet and other assignments.

- GIAC Adore Worm Version 0.8
  http://www.sans.org/y2k/adore.htm

8. ICMP Router Selection

1701 alerts with this signature were found. The earliest was found at 01:39:32 on 21 Jan 2002 and the latest was found at 23:12:22 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 139 sources triggered this attack signature. The top 20 sources are listed below:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.71	76	185	1	10
MY.NET.150.165	69	615	1	28
MY.NET.88.181	65	619	1	32
MY.NET.153.45	40	126	1	4
MY.NET.150.241	39	184	1	9
MY.NET.153.114	38	3770	1	31
MY.NET.151.33	28	29	1	2
MY.NET.150.72	26	26	1	1
MY.NET.153.112	26	1763	1	4
MY.NET.153.105	25	627	1	7
MY.NET.150.79	25	25	1	1
MY.NET.150.223	24	24	1	1
MY.NET.150.63	22	22	1	1
MY.NET.151.90	21	26	1	2
MY.NET.150.100	20	20	1	1
MY.NET.153.46	20	24	1	3
MY.NET.153.115	18	694	1	31
MY.NET.151.89	18	18	1	1
MY.NET.88.188	18	18	1	1
MY.NET.150.37	18	18	1	1

◈ Destination receiving this attack signature

There was only 1 destination triggered this attack signature:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
224.0.0.2	1701	1701	139	139

◈ Brief description of the attack

IP hosts typically learn about routes through manual configuration of the default gateway parameter and redirection messages. If a host boots up without a default gateway setting, that host may issue an ICMP Router Solicitation packet to locate a local router.

This process is referred to as ICMP Router Solicitation and ICMP Router Discovery. IP hosts send ICMP Router Solicitations, and routers reply with ICMP Router Advertisements. By default, the ICMP Router Solicitation packet is sent to the all-routers IP multicast address 224.0.0.2. Although RFC 1812 dictates that IP routers "must support the router part of the ICMP Router Discovery Protocol on all connected networks on which the router supports either IP multicast or IP broadcast addressing," many IP routers do not. If an IP router does not support the router portion of ICMP Router Discovery, the host's Router Solicitation Requests will not be answered.

If an IP host resides on a network that supports multiple IP routers, the IP host may receive multiple replies, i.e. one reply from each of the locally connected IP routers. Typically, the hosts accept and use the first reply received as the default gateway.

◈ Defensive recommendation

No defensive measure is necessary. You can consider assigning default gateways to those hosts manually or by DHCP to prevent them from sending ICMP Router Solicitation messages. It is more preferable as you can make sure the default routers that your hosts use are valid and without problems.

◈ Correlation

- Routing Sequences for ICMP
  http://www.ncmag.com/2001_03/ICMP/

9. ICMP Echo Request CyberKit 2.2 Windows

1486 alerts with this signature were found. The earliest is found at 23:54:54 on 21 Jan 2002 and the latest is found at 16:29:33 on 24 Jan 2002.

◈ Sources triggering this attack signature

There were 4 sources triggered this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.150.49	1463	1493	3	9

MY.NET.88.181	16	619	3	32
MY.NET.150.232	4	328	2	20
MY.NET.150.145	3	11	3	6

◈ Destinations receiving this attack signature

There were 5 destinations received this attack signature:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
204.71.200.33	744	744	4	4
204.71.200.34	739	739	4	4
216.136.175.132	1	1	1	1
216.136.130.46	1	1	1	1
216.136.225.36	1	1	1	1

◈ Brief description of the attack

CyberKit is a collection of network tools for Windows 9X/NT/2000/ME. The following tools are includes: Ping, TraceRoute, Finger, WhoIS, Quote of the Day, NSLookUP, Time Synchronizer, PortScanner, NetInfo and MailChecker. The latest stable release is 2.5. It can be downloaded for free from http://www.cyberkit.net/index.html.

Because of the rich functionality, CyberKit is a very effective tool for reconnaissance purpose.

Some users have probably installed the software in their machines, and use the software to ping external machines. All the sources are in internal network, and a lookup on the destination IP addresses 204.71.200.33 and 204.71.200.34, which received the highest number of alerts, at Dshield.org give the following information:

```
IP Address: 204.71.200.33
Hostname: ns1.yahoo.com
DShield Profile:
```

Country:	US
Contact E-mail:	ipadmin@cw.net
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
[No name] (NS5365-HST)

   Hostname: NS1.YAHOO.COM
   Address: 204.71.200.33
   System: ? running ?

   Record last updated on 17-Aug-2000.
   Database last updated on  28-Dec-2001 19:54:46 EDT.


----------
```

```
Cable & Wireless USA (NETBLK-CW-PROVIDER)
   9000 Regency Parkway, Suite 200
   Cary, NC 27511
   US

   Netname: CW-PROVIDER
   Netblock: 204.71.0.0 - 204.71.255.255
   Maintainer: CWUS

IP Address: 204.71.200.34
Hostname: dns1.snv.yahoo.com
DShield Profile:
```

Country:	US
Contact E-mail:	ipadmin@cw.net
Total Records against IP:	180
Number of targets:	26
Date Range:	-

```
Whois:
Cable & Wireless USA (NETBLK-CW-PROVIDER)
   9000 Regency Parkway, Suite 200
   Cary, NC 27511
   US

   Netname: CW-PROVIDER
   Netblock: 204.71.0.0 - 204.71.255.255
   Maintainer: CWUS
```

◈ Defensive recommendation

No defensive measure is necessary. However, if CyberKit is installed in machines without prior notice to the administrator and being used with malicious intention, it should be removed immediately.

◈ Correlation

- The CyberKit homepage:
  http://www.cyberkit.net/

10. ICMP Fragment Reassembly Time Exceeded

1116 alerts with this signature were found. The earliest was found at 10:59:13 on 22 Jan 2002 and the latest was found at 12:37:31 on 25 Jan 2002.

◈ Sources triggering this attack signature

There were 19 sources triggered this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.171	343	2416	7	97

MY.NET.153.159	334	342	1	3
MY.NET.153.185	149	1227	5	46
MY.NET.88.137	88	153	1	13
MY.NET.153.45	89	126	3	4
MY.NET.153.197	70	797	1	38
MY.NET.152.10	15	16	1	2
MY.NET.152.180	6	105	2	4
MY.NET.151.63	6	130	1	5
MY.NET.88.162	4	59	1	4
MY.NET.152.20	4	4	2	2
MY.NET.6.50	2	437	1	33
MY.NET.88.159	2	33	1	9
MY.NET.153.210	2	36	1	2
MY.NET.6.53	1	82	1	27
MY.NET.6.52	1	406	1	43
MY.NET.88.155	1	1	1	1
MY.NET.153.196	1	276	1	5
MY.NET.152.179	1	82	1	3

◈ Destination receiving this attack signature

There were 31 destinations received this attack signature. The top 20 are listed below:

Destination	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
211.234.110.20	404	404	2	2
211.174.63.106	119	119	1	1
210.158.194.98	88	88	1	1
208.172.128.163	84	84	1	1
211.174.63.108	59	59	1	1
211.171.202.142	58	58	1	1
211.233.70.162	48	48	1	1
211.233.70.165	46	46	1	1
211.233.70.163	45	45	1	1
211.233.27.144	34	34	1	1
211.233.38.109	32	32	1	1
211.233.70.161	28	28	1	1
211.233.50.56	15	15	1	1
211.115.220.73	15	15	2	2
64.132.47.201	7	7	1	1
63.250.208.34	6	6	1	1
MY.NET.6.52	5	12	1	8
162.83.145.85	4	4	1	1
MY.NET.6.60	3	11	1	9
211.233.70.172	3	3	1	1

◈ Brief description of the attack

Each of the internal host received IP fragments with some pieces missing. As after a certain period of time it still did not get the missing pieces, it discarded all the received pieces and generated an ICMP fragment reassembly time exceeded error message back to the external source. The missing pieces may be actually due to network problem. However, an attacker can

deliberately use this technique to discover the hosts who are alive in the network, as only active hosts will send back ICMP fragment reassembly time exceeded error message.

A lookup of the destination IP address 211.234.110.20, which was associated with the highest number, at Dshield.org gives the following information:

```
IP Address: 211.234.110.20
Hostname: 211.234.110.20
DShield Profile:
```

Country:	-
Contact E-mail:	-
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255
```

The destination IP address is found to be assigned to an Asia-Pacific user.

◈ Defensive recommendation

You should block outgoing ICMP fragment reassembly time exceeded error messages at the router. This can be done through filtering by access control list at router.

◈ Correlation

- No further information on the destination IP addresses can be found on the Internet and other assignments.

- Host Detection – Generating arbitrary responses to identify inter-networked nodes: http://www.synnergy.net/downloads/papers/responses-tisc.txt

Top 10 Talkers and Destinations

1a. Top 10 Talkers (for all alert log files)

The following two tables show the top 10 talkers for all alert log files. The first table shows the top 10 talkers from all source IP addresses, including both external and internal addresses. More than 99% of the alerts for the top two source IP addresses MY.NET.70.177 and MY.NET.88.240 are "*SNMP public access*" alerts.

The second table shows the top 10 talkers from external source IP addresses only. It clearly indicates the top 10 outsiders who initiated the most alerts in your network. More than 99% of the alerts of the top two external source IP addresses, 63.210.47.81 and 63.250.208.34 are "*MISC Large UDP Packet*" alerts.

All

Rank	Source IP	Count
1	MY.NET.70.177	10627
2	MY.NET.88.240	7128
3	63.210.47.81	5008
4	63.250.208.34	4970
5	MY.NET.153.114	3770
6	MY.NET.150.121	2988
7	MY.NET.150.198	2850
8	MY.NET.153.171	2416
9	MY.NET.153.202	2385
10	MY.NET.153.118	2297

External Source IP Addresses Only

Rank	Source IP	Count
1	63.210.47.81	5008
2	63.250.208.34	4970
3	211.172.232.21	1609
4	211.202.0.47	1012
5	210.181.96.14	905
6	216.106.166.212	767
7	62.253.169.246	664
8	217.80.164.95	627
9	211.233.70.163	615
10	212.179.35.118	449

1b. Top 10 Talkers (for all scan log files)

The following two tables show the top 10 talkers for all scan log files. The first table shows the top 10 talkers from all source IP addresses, including both external and internal addresses. The second table shows the top 10 talkers from external source IP addresses only. It indicates the outsider from address 66.38.185.141, 205.188.228.0/24, 216.106.172.0/24 and 216.106.173.0/24 are particular interested in your network.

All

Rank	Source IP	Count
1	MY.NET.60.43	352739
2	MY.NET.6.49	79371
3	MY.NET.6.45	73707
4	MY.NET.6.48	45883
5	MY.NET.6.52	41698
6	MY.NET.6.50	34740
7	MY.NET.6.60	31839
8	MY.NET.153.17	28542
9	MY.NET.6.53	25589
10	66.38.185.141	23798

External Source IP Addresses Only

Rank	Source IP	Count
1	66.38.185.141	23798
2	205.188.228.33	12173
3	205.188.228.65	8326
4	205.188.228.17	7303
5	205.188.228.1	6617
6	216.106.172.148	5099
7	216.106.173.149	4918
8	216.106.172.149	4800
9	216.106.173.147	3612
10	216.106.173.148	3308

2a. Top 10 Destinations (from all alert log files)

The following two tables show the top 10 destinations for all alert log files. The first table shows the top 10 destinations from all destination IP addresses, including both external and internal addresses. All the alerts of the top destination IP address MY.NET.150.198 are "*connect to 515 from inside*" alerts. More than 99% of the alerts for the second top destination IP address MY.NET.150.195 are "*SNMP public access*" alerts.

The second table shows the top 10 talkers from internal destination IP addresses only. It clearly indicates the top 10 internal hosts who received the most alerts in your network. Again, the top two destination IP addresses are the same as the previous table.

All

Rank	Destination IP	Count
1	MY.NET.150.198	31425
2	MY.NET.150.195	7155
3	MY.NET.153.45	5917
4	MY.NET.151.63	4974
5	MY.NET.152.109	4073
6	216.241.219.14	2982
7	MY.NET.153.171	2297
8	211.115.213.202	2166
9	MY.NET.5.96	2041
10	MY.NET.5.37	1733

Internal Destination IP Addresses Only

Rank	Destination IP	Count
1	MY.NET.150.198	31425
2	MY.NET.150.195	7155
3	MY.NET.153.45	5917
4	MY.NET.151.63	4974
5	MY.NET.152.109	4073
6	MY.NET.153.171	2297
7	MY.NET.5.96	2041
8	MY.NET.5.37	1733
9	MY.NET.5.128	1683
10	MY.NET.5.127	1669

2b. Top 10 Destinations (for all scan log files)

The following two tables show the top 10 destinations for all scan log files. The first table shows the top 10 destinations from all destination IP addresses, including both external and internal addresses. The two hosts MY.NET.1.3 and MY.NET.1.4 received the most numbers of scan. A detailed look in the scan log files indicates that all the scans are actually from internal hosts to port 53. The two are likely DNS servers of the internal network. A detailed look in the scan log files for third top destination IP MY.NET.88.163 indicates that an external host, 66.38.185.141, initiated a lot of suspicious connection to a total 6648 different ports, where 297 of them are below 1024, to the MY.NET.88.163 host. It is noticeable that the external host 66.38.185.141 is the top talker (external IP addresses) for the scan log files. More information of this external host can be obtained in the previous analysis, *"7. High Port 65535-udp possible Red Worm-traffic*".

All

Rank	Destination IP	Count
1	MY.NET.1.3	34906
2	MY.NET.1.4	24879
3	MY.NET.88.163	23837
4	MY.NET.6.45	22019
5	MY.NET.153.194	21476
6	MY.NET.60.43	19654
7	MY.NET.152.10	19229
8	MY.NET.153.210	18368
9	MY.NET.152.19	14438
10	MY.NET.153.173	14077

Internal Destination IP Addresses Only

Rank	Destination IP	Count
1	MY.NET.1.3	34906
2	MY.NET.1.4	24879
3	MY.NET.88.163	23837
4	MY.NET.6.45	22019
5	MY.NET.153.194	21476
6	MY.NET.60.43	19654
7	MY.NET.152.10	19229
8	MY.NET.153.210	18368
9	MY.NET.152.19	14438
10	MY.NET.153.173	14077

3a. Top 10 Destination Ports (from all alert log files)

The following table shows the top 10 destination ports for all alert log files. The first table shows the top 10 destination ports from both external and internal IP addresses to internal IP address. The top destination port 515 is related to the alert "*connect to 515 from inside*". The second top destination port 161 is related to the alert "*SNMP public access*". More information can be obtained for both of them in the analysis section. The second table shows the top 10 destination ports from external IP address to internal IP address only. Many of them are related to the alert "*MISC Large UDP Packet*". More information can be again obtained in the previous analysis section.

All

Rank	Port	Count
1	515	31425
2	161	25657
3	1221	3878
4	65535	3434
5	0	1884
6	1992	1539
7	1742	763
8	4442	709
9	137	703
10	3276	620

From external IP to internal IP

Rank	Port	Count
1	1221	3878
2	0	1878
3	1992	1539
4	65535	863
5	1742	763
6	4442	709
7	3276	620
8	2925	607
9	3282	602
10	4146	576

3b. Top 10 Destination Ports (from all scan log files)

The following table shows the top 10 destination ports for all scan log files. The first table shows the top 10 destination ports from both external and internal IP addresses to internal IP address. The top two destination ports 7001 and 7000 are related to UDP traffic only, and related to Remote Grab Trojan and af3-callback cache manager respectively. The second table shows the top 10 destination ports from external IP address to internal IP address only. The port 6970 is again related to UDP traffic only, and no known service is known to related to this port.

All

Rank	Port	Count
1	7001	139627
2	7000	77160
3	53	61837
4	0	54497
5	6970	34306
6	7003	26967
7	111	7129
8	123	5414
9	1214	3937
10	514	3243

From external IP to internal IP

Rank	Port	Count
1	6970	34300
2	0	15934
3	7001	5192
4	1214	3469
5	7000	2275
6	21	1612
7	22	821
8	65535	539
9	1221	533
10	1269	517

It is noticeable that a lot of scan is targeted on the destination port 1214, which is related to KaZaA. The KaZaA Media Desktop is a peer-to-peer file-sharing service with which enable searching and downloading media files among KaZaA users. KaZaA supports audio, video, software, games, images, and documents. (http://www.kazaa.com/en/index.htm)

Statistics on port 1214 scan found in the scan log files (from all source IP addresses):

Scan	Count
INVALIDACK	26
NOACK	19
NULL	34
SYN	20131
UDP	518
UNKNOWN	15
VECNA	3158

This large amount of scan attempt towards port 1214 may indicate that some hosts in your network are / were assigned as KaZaA SuperNodes. As the destination IP addresses of this scan are targeted to only one to two hosts, not a range of hosts. Any KaZaA Media Desktop can become a SuperNode if they have a modern computer and are accessing the Internet with a broadband connection. By default, all users have the chances to be chosen as a SuperNode. (http://www.kazaa.com/en/help_connect.htm#port1214). Other KaZaA users using the same Internet Service provider or located in the same region as the SuperNode host, will automatically upload to the SuperNode a small list of files they are sharing. When they search they send the search request to the SuperNode. The actual download will be directly from the computer who is sharing the file, not from the SuperNode. (http://www.kazaa.com/en/help_supernd.htm)

From the scan log files, 47 internal hosts are found to initiate traffic to port 1214 of other hosts. It indicates that many internal hosts were installed with the KaZaA Media Desktop software. Special attention should be paid to the hosts as this kind of peer-to-peer file sharing software can be backdoor to your internal network and result in information theft and leakage. Files infected with virus can also be transmitted into the internal hosts by this mean. You are highly suggested using a stateful firewall to block incoming to port 1214. Besides, you should consider performing a thorough virus scan on the internal hosts, and removing the software installed at the hosts.

Correlated with the OOS log files, hosts MY.NET.88.162 and MY.NET.150.133 received a lot of malicious scan, i.e. with reserved bits set and invalid combination of flags, on TCP port 1214. Checking with the alert log files on the alert signature "*NMAP TCP Ping!*", it is also found that the two hosts are the top 2 destinations receiving this alert signature. Probably, the KaZaA Media Desktop software has exposed the two internal hosts to the external attackers, and the external attackers were gathering information on the two hosts by Nmap.

A lookup of the top two source IP addresses, 193.144.127.9 and 195.77.24.2 associated with the alert signature "*NMAP TCP Ping!*" gives the following information:

```
IP Address: 193.144.127.9
Hostname: 193.144.127.9
DShield Profile:
```

Country:	ES
Contact E-mail:	abuse@rediris.es
Total Records against IP:	2179
Number of targets:	672
Date Range:	2002-02-14 to 2002-02-16

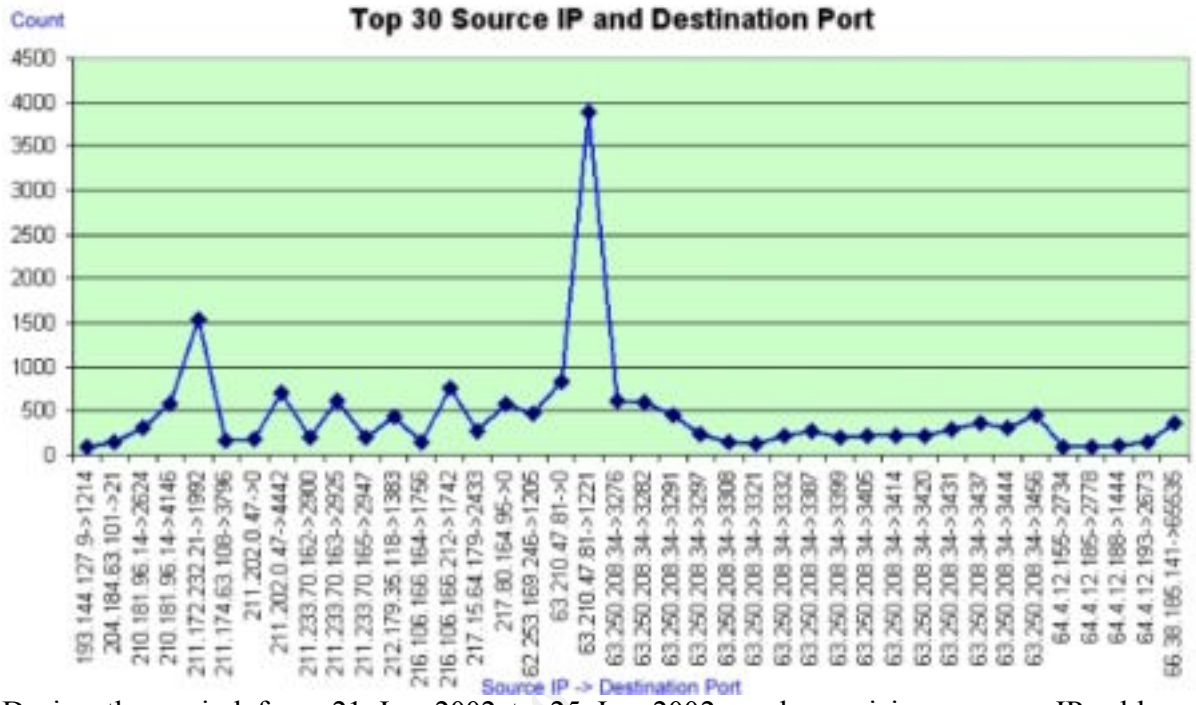
```
Whois:
inetnum:      193.144.104.0 - 193.144.127.255
netname:      GVA
descr:        Red GVA De La Generalitat Valenciana
descr:        Valencia
country:      ES

IP Address: 195.77.24.2
Hostname: 195.77.24.2
DShield Profile:
```

Country:	ES
Contact E-mail:	jgaleano@gva.es
Total Records against IP:	122
Number of targets:	58
Date Range:	2002-02-15 to 2002-02-15

```
Whois:
inetnum:      195.77.24.0 - 195.77.24.255
netname:      GVANET
descr:        Generalitat Valenciana
descr:        Internet access for Valencia State (NCC#1998103531)
country:      ES
```

Other analysis through graphical methods

The number of alerts that the top 30 source IP and destination port pairs associated with are shown in the following graph:



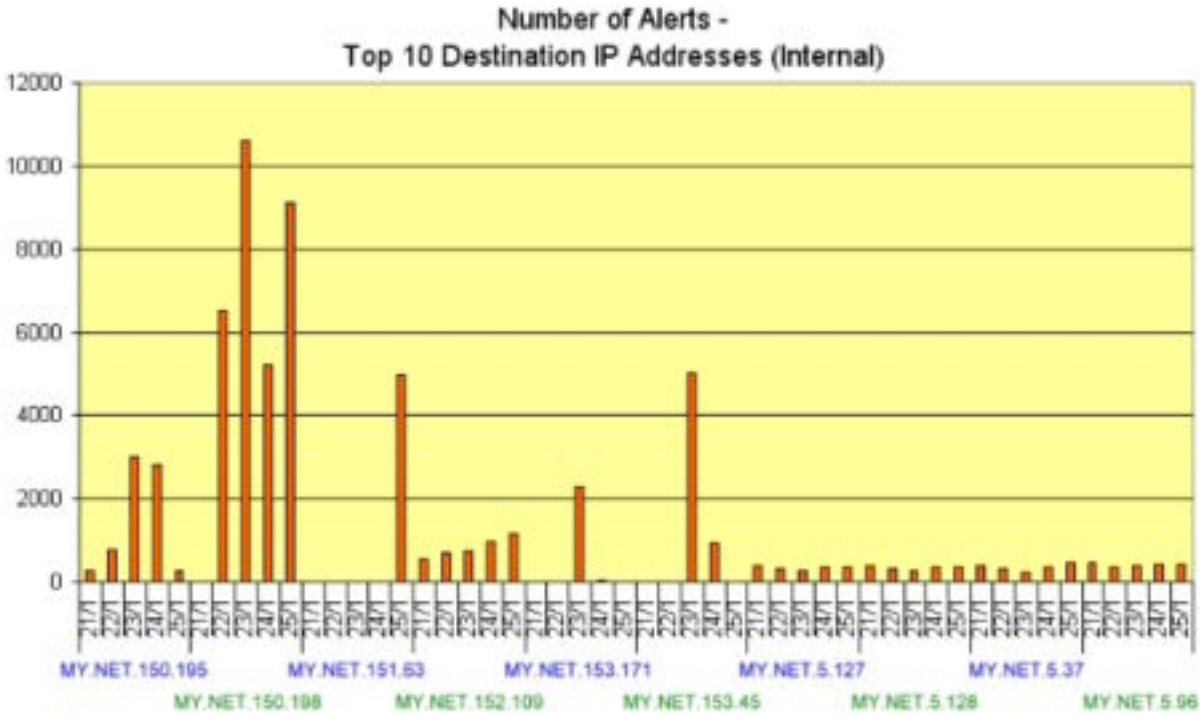
During the period from 21 Jan 2002 to 25 Jan 2002, each suspicious source IP address and destination port pair generally generated around 500 alerts. However, two exceptional pairs, source IP address 63.210.47.81 and destination port 1221 and source IP address 211.172.232.21 and destination port 1992, generated alerts a lot more than the others. This behavior deviates from the others. The two pairs are found associated with the alerts "*MISC Large UDP Packet*". Information of the host 63.210.47.81 can be found in the previous analysis section. with the alert. A lookup on the source IP address 211.174.63.108 at Dshield.org gives the following information:

```
IP Address: 211.172.232.21
Hostname: 211.172.232.21
DShield Profile:
```

Country:	KR
Contact E-mail:	dkkim@kci.co.kr
Total Records against IP:	-
Number of targets:	-
Date Range:	-

```
Whois:
IP Address          : 211.172.232.0-211.172.232.255
Connect ISP Name    : HANNET
Connect Date        : 20001031
Registration Date   : 20001101
Network Name        : HANNET-INFRA
```

The number of alerts that the top 10 internal destination hosts, during the period 21 Jan 2002 to 25 Jan 2002 are shown in the following graph:



The graph shows that for hosts MY.NET.152.109, MY.NET.5.127, MY.NET.5.128, MY.NET.5.37, MY.NET.5.96, the amount of alerts they received are fairly steady. There is a high chance that the alerts they received are actually false-positives. However, for hosts MY.NET.151.63, MY.NET.153.171 and MY.NET.153.45, each of them normally received very few or even no alerts, but suddenly received a very high amount of alerts in one of the days. If these hosts are servers that are offering services around the clock, they needed to be thoroughly checked to see whether they are targeted by attackers.

Appendix: Analysis Process - Log files manipulation

Snortsnarf was used to analyze the alerts detected in the period during 21 Jan 2002 – 25 Jan 2002. These alerts files were:

- alert.020121
- alert.020122
- alert.020123
- alert.020124
- alert.020125

Snortsnarf was downloaded from the Silicon Defense Web site at http://www.silicondefense.com/software/snortsnarf/index.htm. The version downloaded was 020126.1. The software was installed at a desktop PC, which was a Pentium III 1.0G machine with 512MB memory and a 30GB hard drive. The operating system of the PC was Red Hat Linux 7.2. Before running Snortsnarf, the followings were performed on the log files:

1. Remove the header portion in each alert log file
2. Concatenate them into a single alert.log file
3. Use *vi* editor to replace all MY.NET internal network address to 10.1 as Snortsnarf cannot properly handle the MY.NET string.

Then I used the following command to launch Snortsnarf:

```
> snortsnarf.pl alert.log
```

The generated web pages were used for brief analysis.

The single alert.log file was then transferred to another PC, with Microsoft Windows operating system, ActivePerl, Microsoft Excel and Microsoft Access installed. A Perl script was written and used to convert the alert log file into csv format:

```
----------------------
#
# convert_alert.pl
#
$SnortLog = 'c:\snort\alert.log';

print ("Date;Alert;Src IP;Src Port;Dst IP;Dst Port\n");
open file, "<$SnortLog";

while ($line = <file>){

   chomp $line;
   @message = split /[ ]+\[\*\*\] /, $line;
   @message1 = split /\./, @message[0];
   @message2 = split / -> /, @message[2];
   @message3 = split /:/, @message2[0];
   @message4 = split /:/, @message2[1];
   @message5 = split /-/, @message1[0];
```

```
#  This is for date format DD/MM/YYYY
   @message6 = split /\//, @message5[0];
   print ("@message6[1]/@message6[0]/2002
@message5[1];@message[1];@message3[0];@message3[1];@message4[0];@message4[1]\
n");
}
close file;
--------------------
```

Here are some entries in the original alert log file:

```
01/21-00:01:28.256289  [**] SNMP public access [**] MY.NET.84.155:1498 ->
MY.NET.150.14:161
01/21-00:01:28.356522  [**] SNMP public access [**] MY.NET.84.155:1498 ->
MY.NET.150.14:161
01/21-00:01:28.456846  [**] SNMP public access [**] MY.NET.84.155:1498 ->
MY.NET.150.14:161
```

The new csv file was generated by:

```
> convert_alert.pl > alert.csv
```

The log entries in the new csv file become:

```
Date;Alert;Src IP;Src Port;Dst IP;Dst Port
21/01/2002 00:01:28;SNMP public access;MY.NET.84.155;1498;MY.NET.150.14;161
21/01/2002 00:01:28;SNMP public access;MY.NET.84.155;1498;MY.NET.150.14;161
21/01/2002 00:01:28;SNMP public access;MY.NET.84.155;1498;MY.NET.150.14;161
```

The alert.csv file in csv format was then imported into Microsoft Excel and Microsoft Access for further analysis. The graphs included in this assignment were all generated by using pivot tables and graphs, while queries on the data were conducted in Microsoft Access.

Similarly, the scan log files were also processed in this way. The scan log files were:

- scans.020121
- scans.020122
- scans.020123
- scans.020124
- scans.020125

First, their headers were removed, and the five scan log files were concatenated into a single file named scans.log. A Perl program was written and used to convert the scan log file into csv format:

```
--------------------
#
# convert_scan.pl
#
$ScanLog = 'c:\snort\scans.log';
```

```
print ("Date;Time;Src IP;Src Port;Dst IP;Dst Port;Proto;Other\n");
open file, "<$ScanLog";

while ($line = <file>){

   chomp $line;
   @message = split /[ ]+/, $line;
   @message2 = split /:/, @message[3];
   @message3 = split /:/, @message[5];

   print ("@message[0]
@message[1];@message[2];@message2[0];@message2[1];@message3[0];@message3[1];@
message[6];@message[7]\n");

}
close file;
--------------------
```

Here are some entries in the original scan log file:

```
Jan 21 00:00:04 MY.NET.60.43:123 -> MY.NET.153.193:1473 UDP
Jan 21 00:00:04 MY.NET.60.43:123 -> MY.NET.153.200:1489 UDP
Jan 21 00:13:45 MY.NET.5.50:13892 -> MY.NET.5.83:7938 SYN ******S*
Jan 21 00:13:43 MY.NET.5.50:29171 -> MY.NET.5.83:7937 SYN ******S*
Jan 21 00:13:43 MY.NET.5.50:29172 -> MY.NET.5.83:13880 SYN ******S*
```

The new csv file was generated by:

```
> convert_scan.pl > scans.csv
```

The log entries in the new csv file become:

```
Date;Time;Src IP;Src Port;Dst IP;Dst Port;Proto;Other
Jan 21;00:00:04;MY.NET.60.43;123;MY.NET.153.193;1473;UDP;
Jan 21;00:00:04;MY.NET.60.43;123;MY.NET.153.200;1489;UDP;
Jan 21;00:13:45;MY.NET.5.50;13892;MY.NET.5.83;7938;SYN;******S*
Jan 21;00:13:43;MY.NET.5.50;29171;MY.NET.5.83;7937;SYN;******S*
Jan 21;00:13:43;MY.NET.5.50;29172;MY.NET.5.83;13880;SYN;******S*
```

Again, the scans.csv file in csv format were imported into Microsoft Excel and Microsoft Access for further analysis. The graphs included in this assignment were all generated by using pivot tables and graphs, while queries on the data were conducted in Microsoft Access.

Because of the few entries found in the OOS files, the files were simply removed with headers and concatenated together for manual inspection. The OOS files were:

- oos_Jan.21.2002
- oos_Jan.22.2002
- oos_Jan.23.2002
- oos_Jan.24.2002
- oos_Jan.25.2002