SANS Intrusion Detection Practical for Parliament Hill 2001

Level Two - Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.0 (revised August 13, 2001)

Chris Payne

Contents

Assignment 1 - Describe the State of Intrusion Detection

For this portion of the practical, I have chosen to write about a network reconnaissance utility. The tool I am writing about is actually a tool I have recently written myself called PortProbe. Basically, this program is what I call a "banner grabber". Now its sole purpose in life is to connect to a user specified IP address on a user selected port, and if that port is accepting connections on the remote host, PortProbe will retrieve the banner information displayed by the service operating on that port. Why would this be useful you might ask? Well as my previous job consisted of me auditing my departments own networks for vulnerabilities at all levels, this is one of the first tools that could be used during the reconnaissance phase of an attack. What a potential attacker wants to determine is, are there vulnerable versions of software that are exploitable and running on the network? Once PortProbe returns a version of software being used as a service to the user, this information can then be used to research known exploits or vulnerabilities by using a site like http://www.securityfocus.com.

There are actually other good uses for this tool as well. In my daily duties of System Administrator there have been times when I have wanted to know what hosts have not been patched to the latest version of whatever software needed to be pushed out at that time. We all know that as our networks grow, and our duties increase, it gets harder and harder to keep a good grip on our networks, so some hosts tend to be forgotten during upgrades to newer versions of various packages like SSH. For those not familiar with SSH, it is a replacement for inherently insecure programs such as telnet, rlogin, ftp, and other such programs that transmit their password across the Internet unencrypted. SSH encrypts all traffic. To save me from walking around to everyone's computer and typing something like `ssh -V` to display the version, I wanted a tool capable of probing networks with in the hopes of compiling a report on service versions. Now I realize that there are probably scripts like this already available, but I wanted the challenge of creating my own.

When I actually started to code this, I starting thinking why stop with just port 22? Why not a more robust, less restrictive utility? So I set out to code a small tool I could use to run against either one particular IP, or against a few non-sequential IP addresses or even against a large range (from 1 - 254 hosts within the same subnet) probing any port the person running the program wanted and retrieve some banner information from the service running on that port in question. One of the little issues I had was with retrieving the http header information and format that in the report but I think I have it pretty much straightened out now.

I decided to write the program in a scripting language called Winbatch. Wilson Windoware is the company distributing Winbatch, and for those that have never used or heard of this product before, it is a full-featured programming language, with the ability to produce independent .EXE files that you can distribute freely. The core of the product is the Windows Interface Language (WIL) which is actually a powerful general-purpose batch style programming language. It is used in a lot of companies to create logon scripts, push updates to clients and is capable of automating pretty much any task you currently perform on a Windows based machine.

The version of PortProbe used was PortProbe v0.10 BETA. For those interested in this tool, please check http://www.whitehats.ca/main/members/Chris/Chris.html for more information.

For the data presented in this portion of the practical, I have used my network as presented below with my "main" computer running the Windows 2000 Professional operating system while running VMware using a virtual Windows NT 4 Workstation as the "shooter" as well as a virtual Slackware Linux 8.0 workstation as one of the "targets". There were 4 targets used: 1) my firewall, 2) my "main computer" 3) my laptop, 4) and a virtual Linux host. The computer targeting them all was the virtual NT 4 computer. Some of the computers are running the SSH daemon, and some are not. It is my intention to display to the reader what PortProbe traffic looks like so they can appreciate the difficulty of detecting a tool like this using and IDS. The problem is, when the utility finds a host that actually runs a service on a port, it will look like a normal connection that gets torn down almost immediately.

If you are unfamiliar with VMware, the literature from the company says that "With VMware Workstation, operating systems and applications run inside virtual machines. So you can create a whole set of computers - whether you operate under Linux, Windows NT or Windows 2000." Basically it enables users to run different operating systems on the same computer with no need to reboot between different operating systems, and these virtual hosts can all be active on the network at the same time. This is an amazing product for network programming and testing.

The shooter for this exercise was the NT 4 Workstation virtual computer with an IP of 192.168.30.30. This host was going to probe 192.168.30.1-192.168.30.5 looking for SSH version banners. My real firewall Shadow IDS v1.5 (192.168.30.1) was included in the scan as well as my main computer (192.168.30.2), my laptop running Slackware Linux 8.0 (192.168.30.3), and finally, the Slackware Linux 8.0 virtual computer (192.168.30.5).

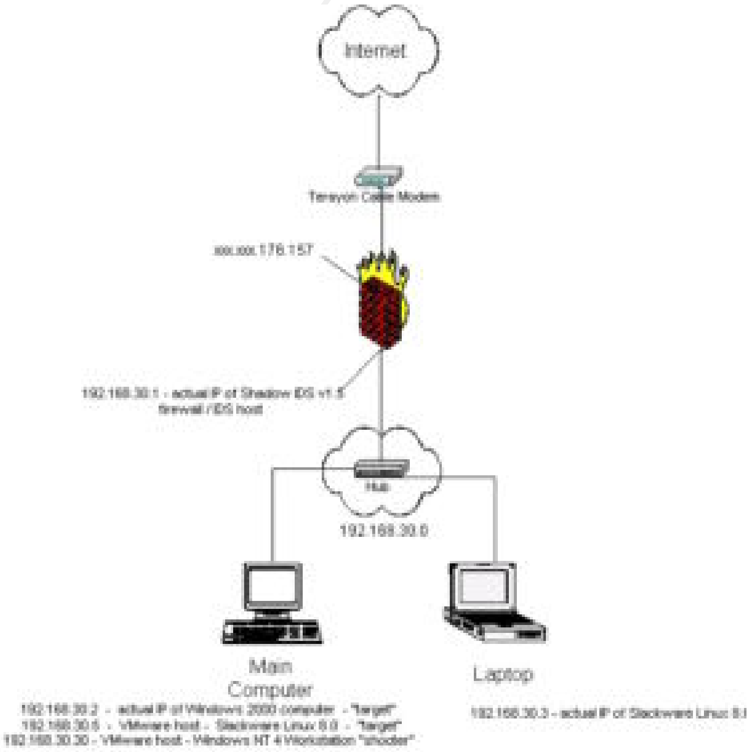
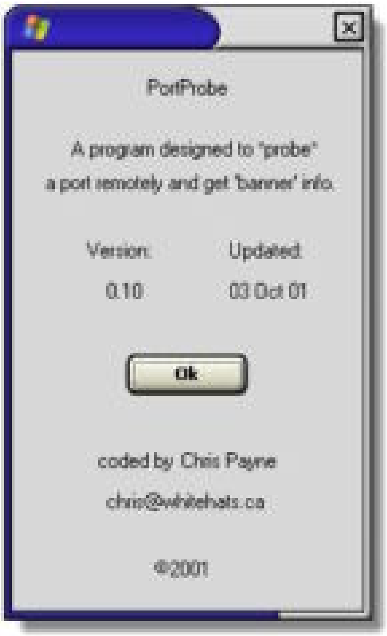

Internet
Terayon Cable Modem
xxx.xxx.176.157
192.168.30.1 - actual IP of Shadow IDS v1.5 firewall / IDS host
Hub
192.168.30.0
Main Computer
192.168.30.2 - actual IP of Windows 2000 computer - "target"
192.168.30.5 - VMware host - Slackware Linux 8.0 - "target"
192.168.30.30 - VMware host - Windows NT 4 Workstation "shooter"
Laptop
192.168.30.3 - actual IP of Slackware Linux 8.0

Fig. 1-1 My Test Network Environment

Before we take a look at what kind of network traffic this tool generates, I would like to give the reader an understanding of how this tool works. I will do this with the help of some screen captures to display the process of scanning a small range of hosts. After I cover that, we can thought go into a little detail of the network traffic that tcpdump was able to capture.

The following example is probably the most typical use. It is just a couple of IP addresses with no fancy options.



PortProbe

A program designed to "probe"
a port remotely and get 'banner' info.

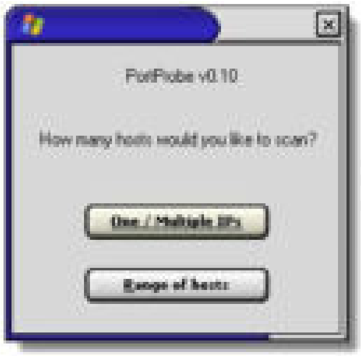
Version: 0.10

Updated: 03 Oct 01

coded by Chris Payne
chris@whitehats.ca

©2001

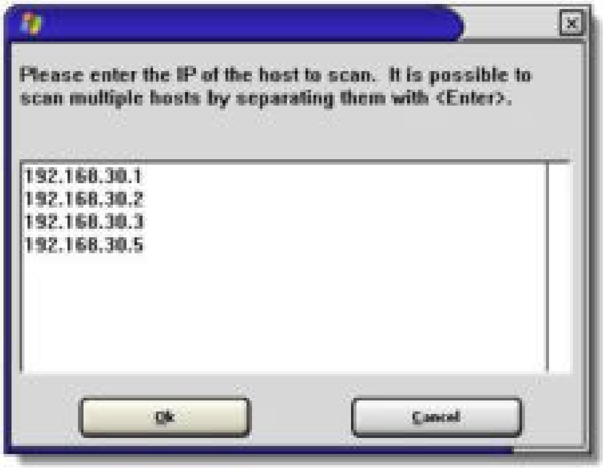
Fig. 1-2 Starting up PortProbe.



PortProbe v0.10

How many hosts would you like to scan?

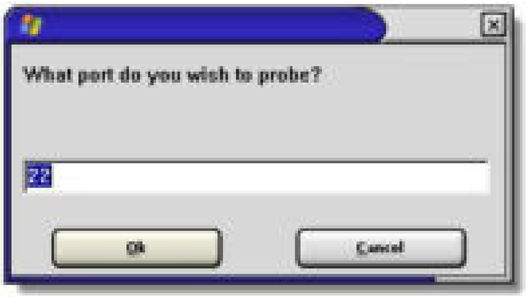
Fig. 1-3 There is an option to probe one IP, a few non-sequential IP's or even a range (from 1 to 254) hosts. In this example I will select "One/Multiple IPs".



Please enter the IP of the host to scan. It is possible to scan multiple hosts by separating them with <Enter>.

```
192.168.30.1
192.168.30.2
192.168.30.3
192.168.30.5
```

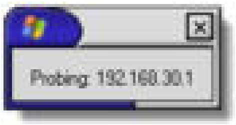
Fig. 1-4 The interface to enter a single IP or few IP addresses.



What port do you wish to probe?

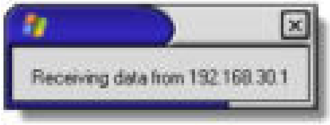
22

Fig. 1-5 The program will prompt you on which remote port to try and obtain "banner" information from. The default port is 22.



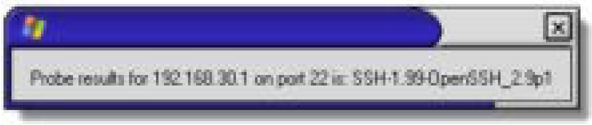
Probing: 192.168.30.1

Fig. 1-6
Probing the host.



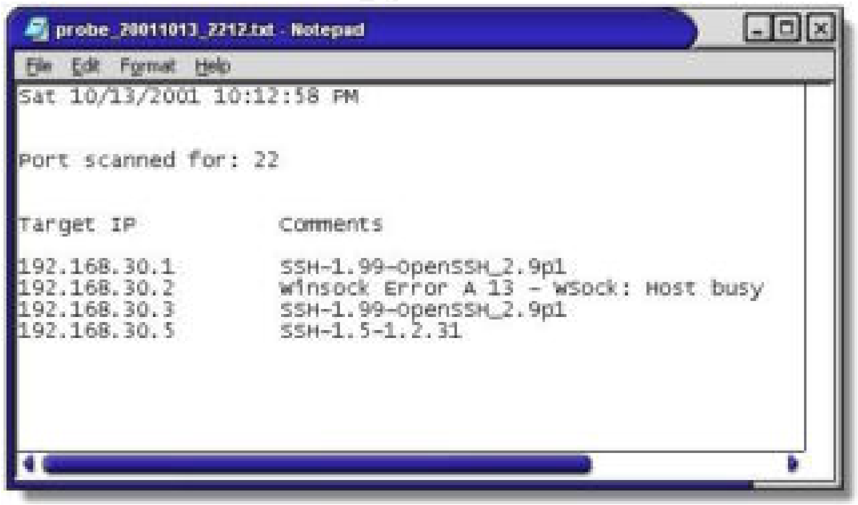
Receiving data from 192.168.30.1

Fig. 1-7
Receive data from the host.



Probe results for 192.168.30.1 on port 22 is: SSH-1.99-OpenSSH_2.9p1

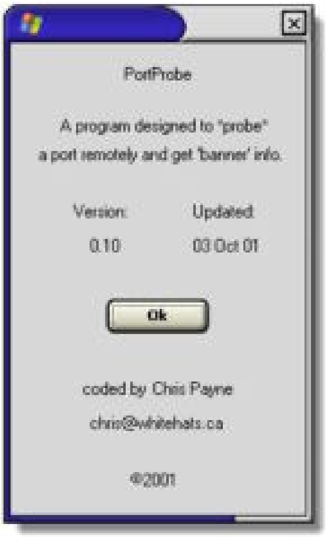
Fig. 1-8
Display a quick blurb containing any information found.



```
Sat 10/13/2001 10:12:58 PM


Port scanned for: 22


Target IP           Comments

192.168.30.1        SSH-1.99-OpenSSH_2.9p1
192.168.30.2        Winsock Error A 13 - WSock: Host busy
192.168.30.3        SSH-1.99-OpenSSH_2.9p1
192.168.30.5        SSH-1.5-1.2.31
```

Fig. 1-9 The final results are displayed in a handy notepad window for future reference.

The following example attempts to be a little stealthier.


PortProbe

A program designed to "probe" a port remotely and get 'banner' info.

Version: 0.10

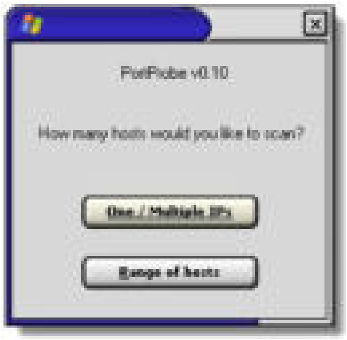
Updated: 03 Oct 01

coded by Chris Payne

chris@whitehats.ca

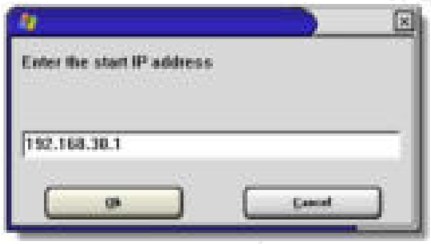
©2001

Fig. 1-10 Starting up PortProbe


PortProbe v0.10

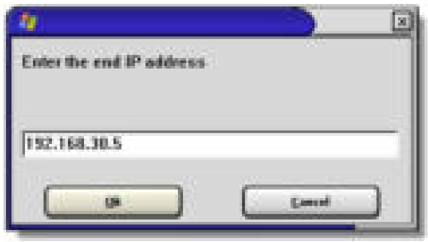
How many hosts would you like to scan?

Fig. 1-11 In this example I will select a “Range of Hosts”


Enter the start IP address

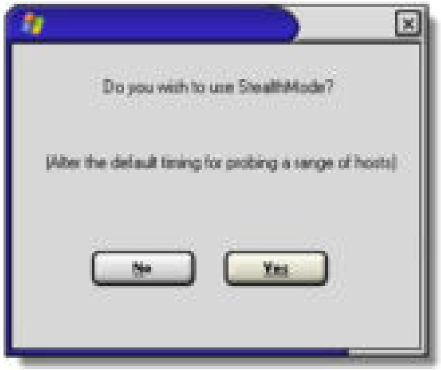
192.168.30.1

Fig. 1-12 We are prompted for beginning of the IP range.


Enter the end IP address

192.168.30.5

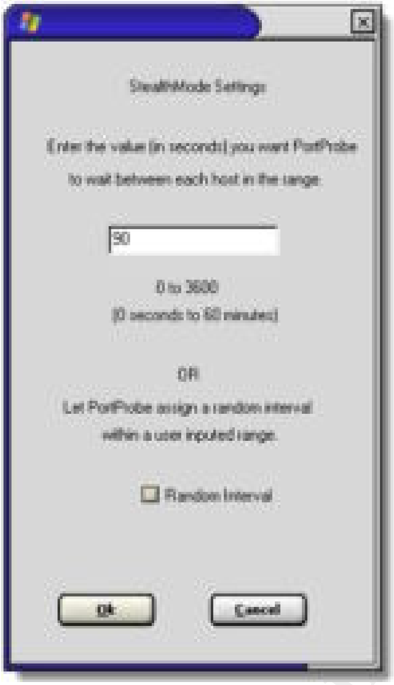
Fig. 1-13 We are prompted for the end of the IP range.



Do you wish to use StealthMode?

(Alter the default timing for probing a range of hosts)

Fig. 1-14 When you are scanning a range, we have an additional option. The use of "StealthMode" is basically a random interval between probing each host to try to hide the probes.



StealthMode Settings

Enter the value (in seconds) you want PortProbe to wait between each host in the range.

90

0 to 3600
(0 seconds to 60 minutes)

OR

Let PortProbe assign a random interval within a user inputed range.

□ Random Interval

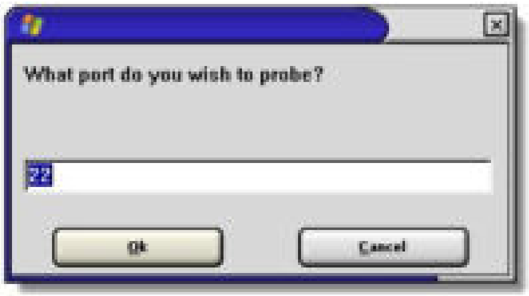
Fig. 1-15 In StealthMode, there is an option for a "static" timeout interval between probing multiple hosts, or there is an option for a newly generated random interval between each host.

I am just going to interrupt these screen shots for a couple of minutes and discuss what StealthMode means to PortProbe. I had mentioned earlier that it used to be my job to audit our own networks at work, having said that, part of the hopes of performing the pre-attack host analysis is to not give away your intentions and end up getting caught by generating too much traffic that you stand out like a blinking beacon to any analyst sitting in front of an IDS.

My plan to deal with this was to implement some sort of variable timeouts between probing each host. I have decided to only include this feature when scanning a range of hosts (not just a couple hosts) as I felt this is where it would be most obvious. The way it works is, if you are looking a at a range (lets say 192.168.30.1 – 192.168.30.10) we are presented with the option to either 1) enter the value (in seconds) you want PortProbe to wait between each host in the range, or 2) let PortProbe assign a random interval within a user defined range.

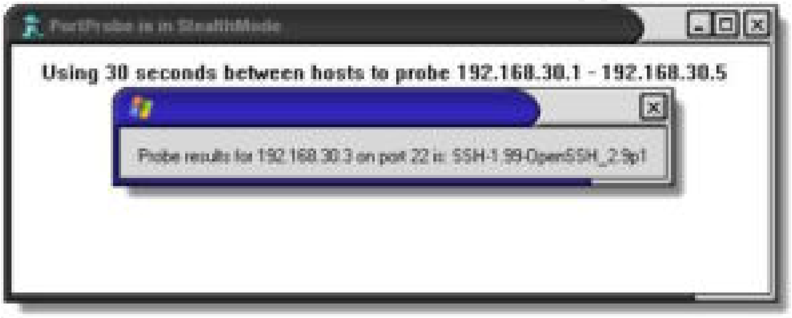
What does all that mean? Well, if we selected option number 1 from the paragraph above, PortProbe would wait X seconds after probing 192.168.30.1 before it moved on to 192.168.30.2. While if we select option 2 from above, PortProbe would wait a randomly generated timeout after one host before moving on the next in the list. The plan is each timeout for option 2 would be different (possibly anywhere from a 0 second to 3600 second delay).

The thinking behind adding this "stealth" feature was so that the IDS doesn't see a glaring pattern and flag it immediately. By having random timeouts (some as long as 60 minutes) between hosts, the analyst might not notice a probe here and there in the logs.


What port do you wish to probe?

22

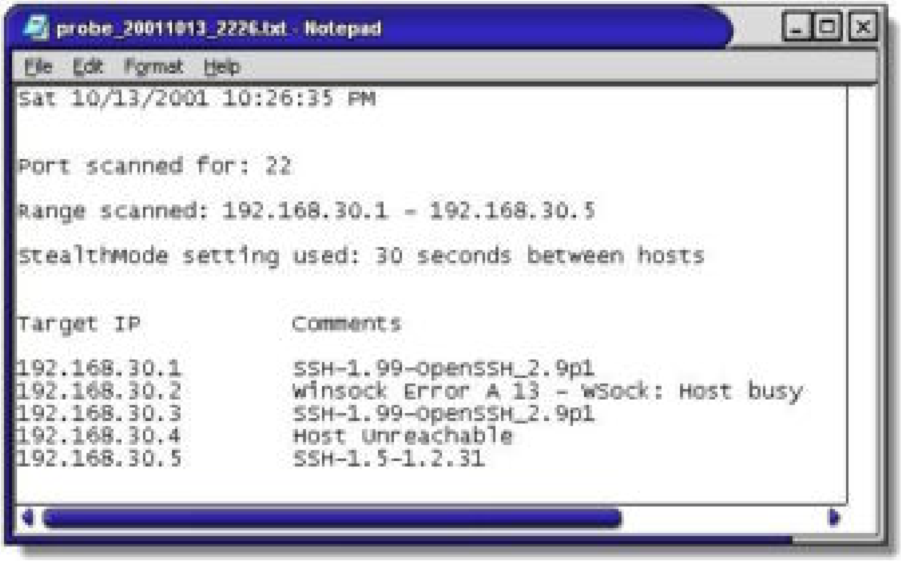
Fig. 1-16 We need to specify the port to scan.



Using 30 seconds between hosts to probe 192.168.30.1 - 192.168.30.5

Probe results for 192.168.30.3 on port 22 is: SSH-1.99-OpenSSH_2.9p1

Fig. 1-17 The program now probes the IP range.



```
Sat 10/13/2001 10:26:35 PM

Port scanned for: 22

Range scanned: 192.168.30.1 - 192.168.30.5

StealthMode setting used: 30 seconds between hosts

Target IP          Comments

192.168.30.1       SSH-1.99-OpenSSH_2.9p1
192.168.30.2       Winsock Error A 13 - WSock: Host busy
192.168.30.3       SSH-1.99-OpenSSH_2.9p1
192.168.30.4       Host Unreachable
192.168.30.5       SSH-1.5-1.2.31
```

Fig. 1-18 The final results from the StealthMode probe.

Figure 1-19 shows the user the results of the competed probe. For each IP address in the range scanned, it shows a comment. We can see for host 192.168.30.1, it is running SSH-1.99-OpenSSH_2.9p1, but what is that message for host 192.168.30.2? PortProbe is saying "Winsock Error A 13 – Wsock: Host Busy" this is PortProbe/Winbatch's way of saying that the IP address is "alive" (pingable) on the network but is not offering a service on that port. Basically, you can think of it as a closed port. The comment displayed for host 192.168.30.4 means that IP address was not "alive" (or pingable) on the network during the time that PortProbe ran.

The following is a capture of the noise generate by PortProbe while probing one host on my test network. As we can see, it is not too noisy and doesn't really look too much out of the ordinary.

- The probe to host 192.168.30.1 which happens to be Linux host running the SSH daemon.

```
16:49:46.217734 192.168.30.30.1028 > 192.168.30.1.22: S 131218:131218(0) win 8192 <mss 1460> (DF)
(ttl 128, id 25856)
0x0000   4500 002c 6500 4000 8006 d85b c0a8 1e1e     E..,e.@....[....
0x0010   c0a8 1e01 0404 0016 0002 0092 0000 0000     ................
0x0020   6002 2000 b608 0000 0204 05b4 0000          `............

16:49:46.217769 192.168.30.1.22 > 192.168.30.30.1028: S 382513760:382513760(0) ack 131219 win
16060 <mss 1460> (DF) (ttl 64, id 7492)
0x0000   4500 002c 1d44 4000 4006 6018 c0a8 1e01     E..,.D@.@.`.....
0x0010   c0a8 1e1e 0016 0404 16cc b260 0002 0093     ...........`....
0x0020   6012 3ebc ce0e 0000 0204 05b4 0000          `.>...........

16:49:46.220829 192.168.30.30.1028 > 192.168.30.1.22: . 1:1(0) ack 1 win 8760 (DF) (ttl 128, id
26112)
0x0000   4500 0028 6600 4000 8006 d75f c0a8 1e1e     E..(f.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0093 16cc b261     ...............a
0x0020   5010 2238 0250 0000 0000 0000 0000          P."8.P........

16:49:46.225201 192.168.30.30.1028 > 192.168.30.1.22: . 1:1(0) ack 1 win 8760 (DF) (ttl 128, id
26112)
0x0000   4500 0028 6600 4000 8006 d75f c0a8 1e1e     E..(f.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0093 16cc b261     ...............a
0x0020   5010 2238 0250 0000                         P."8.P..
```

{ Now that we have established our TCP 3 way handshake, this next capture shows the target sending back to the prober the ssh banner version we had hoped to recover.

```
16:49:46.225921 192.168.30.1.22 > 192.168.30.30.1028: P 1:24(23) ack 1 win 16060 (DF) (ttl 64, id
7493)
0x0000   4500 003f 1d45 4000 4006 6004 c0a8 1e01     E..?.E@.@.`.....
0x0010   c0a8 1e1e 0016 0404 16cc b261 0002 0093     ...........a....
0x0020   5018 3ebc 78d7 0000 5353 482d 312e 3939     P.>.x...SSH-1.99
0x0030   2d4f 7065 6e53 5348 5f32 2e39 7031 0a       -OpenSSH_2.9p1.

16:49:46.393621 192.168.30.30.1028 > 192.168.30.1.22: . 1:1(0) ack 24 win 8737 (DF) (ttl 128, id
26368)
0x0000   4500 0028 6700 4000 8006 d65f c0a8 1e1e     E..(g.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0093 16cc b278     ...............x
0x0020   5010 2221 0250 0000                         P."!.P..

16:49:46.394395 192.168.30.30.1028 > 192.168.30.1.22: . 1:1(0) ack 24 win 8737 (DF) (ttl 128, id
26368)
0x0000   4500 0028 6700 4000 8006 d65f c0a8 1e1e     E..(g.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0093 16cc b278     ...............x
0x0020   5010 2221 0250 0000 0000 0000 0000          P."!.P........
```

{ Once the prober has the requested information, it attempts to gracefully close the connection.

```
16:49:50.304315 192.168.30.30.1028 > 192.168.30.1.22: F 1:1(0) ack 24 win 8737 (DF) (ttl 128, id
26624)
0x0000   4500 0028 6800 4000 8006 d55f c0a8 1e1e     E..(h.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0093 16cc b278     ...............x
0x0020   5011 2221 024f 0000                         P."!.O..

16:49:50.304968 192.168.30.30.1028 > 192.168.30.1.22: F 1:1(0) ack 24 win 8737 (DF) (ttl 128, id
26624)
0x0000   4500 0028 6800 4000 8006 d55f c0a8 1e1e     E..(h.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0093 16cc b278     ...............x
0x0020   5011 2221 024f 0000 0000 0000 0000          P."!.O........

16:49:50.305007 192.168.30.1.22 > 192.168.30.30.1028: . 24:24(0) ack 2 win 16060 (DF) (ttl 64, id
7494)
0x0000   4500 0028 1d46 4000 4006 601a c0a8 1e01     E..(.F@.@.`.....
0x0010   c0a8 1e1e 0016 0404 16cc b278 0002 0094     ...........x....
0x0020   5010 3ebc e5b3 0000 0000 0000 0000          P.>...........

16:49:50.306234 192.168.30.1.22 > 192.168.30.30.1028: F 24:24(0) ack 2 win 16060 (DF) (ttl 64, id
7495)
0x0000   4500 0028 1d47 4000 4006 6019 c0a8 1e01     E..(.G@.@.`.....
0x0010   c0a8 1e1e 0016 0404 16cc b278 0002 0094     ...........x....
0x0020   5011 3ebc e5b2 0000 0000 0000 0000          P.>...........

16:49:50.309821 192.168.30.30.1028 > 192.168.30.1.22: . 2:2(0) ack 25 win 8737 (DF) (ttl 128, id
26880)
0x0000   4500 0028 6900 4000 8006 d45f c0a8 1e1e     E..(i.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0094 16cc b279     ...............y
0x0020   5010 2221 024e 0000                         P."!.N..

16:49:50.310250 192.168.30.30.1028 > 192.168.30.1.22: . 2:2(0) ack 25 win 8737 (DF) (ttl 128, id
26880)
0x0000   4500 0028 6900 4000 8006 d45f c0a8 1e1e     E..(i.@...._....
0x0010   c0a8 1e01 0404 0016 0002 0094 16cc b279     ...............y
0x0020   5010 2221 024e 0000 0000 0000 0000          P."!.N........
```

There isn't really anything outstanding from this traffic. It just looks like a normal SSH session (albeit a short one, with a teardown right after the connection).

Now let's check out the traffic generated in one of my favourite uses of the tool, checking web server types. I will send a quick probe to a website that shall remain anonymous and see just what kind of noise PortProbe will generate.

{ Initial SYN request to initiate a connection.

```
22:23:14.507424 192.168.30.7.1584 > xxx.xxx.96.36.80: S 397155:397155(0) win 8192 <mss 1460> (DF)
(ttl 128, id 42750)
0x0000   4500 002c a6fe 4000 8006 3c51 c0a8 1e07     E..,..@...<Q....
0x0010   xxxx 6024 0630 0050 0006 0f63 0000 0000     ..`$.0.P...c....
0x0020   6002 2000 4ac1 0000 0204 05b4               `...J.......

22:23:14.541675 xxx.xxx.96.36.80 > 192.168.30.7.1584: S 2875459461:2875459461(0) ack 397156 win
16060 <mss 1460> (DF) (ttl 51, id 39035)
0x0000   4500 002c 987b 4000 3306 97d4 xxxx 6024     E..,.{@.3.....`$
0x0010   c0a8 1e07 0050 0630 ab64 0785 0006 0f64     .....P.0.d.....d
0x0020   6012 3ebc 790a 0000 0204 05b4 0000          `.>.y.........

22:23:14.541850 192.168.30.7.1584 > xxx.xxx.96.36.80: . 1:1(0) ack 1 win 8760 (DF) (ttl 128, id
43006)
0x0000   4500 0028 a7fe 4000 8006 3b55 c0a8 1e07     E..(..@...;U....
0x0010   xxxx 6024 0630 0050 0006 0f64 ab64 0786     ..`$.0.P...d.d..
0x0020   5010 2238 ad4b 0000                         P."8.K..
```

{ Great, we received our TCP 3 way handshake, lets send our GET request to the www server.

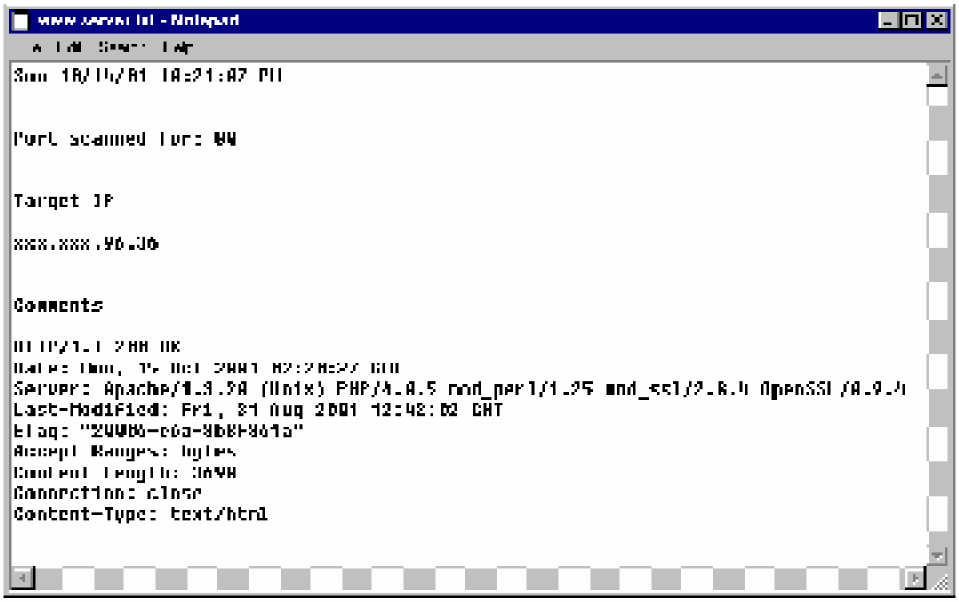
```
22:23:14.547561 192.168.30.7.1584 > xxx.xxx.96.36.80: P 1:18(17) ack 1 win 8760 (DF) (ttl 128, id
43262)
0x0000   4500 0039 a8fe 4000 8006 3a44 c0a8 1e07     E..9..@...:D....
0x0010   xxxx 6024 0630 0050 0006 0f64 ab64 0786     ..`$.0.P...d.d..
0x0020   5018 2238 d372 0000 4845 4144 202f 2048     P."8.r..HEAD./.H
0x0030   5454 502f 312e 300d 0a                      TTP/1.0..

22:23:14.600734 xxx.xxx.96.36.80 > 192.168.30.7.1584: . 1:1(0) ack 18 win 16060 (DF) (ttl 51, id
39037)
0x0000   4500 0028 987d 4000 3306 97d6 xxxx 6024     E..(.}@.3.....`$
0x0010   c0a8 1e07 0050 0630 ab64 0786 0006 0f75     .....P.0.d.....u
0x0020   5010 3ebc 90b6 0000 0000 0000 0000          P.>...........

22:23:14.600899 192.168.30.7.1584 > xxx.xxx.96.36.80: P 18:54(36) ack 1 win 8760 (DF) (ttl 128,
id 43518)
0x0000   4500 004c a9fe 4000 8006 3931 c0a8 1e07     E..L..@...91....
0x0010   xxxx 6024 0630 0050 0006 0f75 ab64 0786     ..`$.0.P...u.d..
0x0020   5018 2238 b3fb 0000 4163 6365 7074 3a20     P."8....Accept:.
0x0030   2a2f 2a0d 0a48 6f73 743a 2032 3136 2e31     */*..Host:.xxx.x
0x0040   3638 2e39 362e 3336 0d0a 0d0a               xx.96.36....
```

{ Great, the www server likes our request and is sending us the goods below.

```
22:23:14.659826 xxx.xxx.96.36.80 > 192.168.30.7.1584: P 1:301(300) ack 54 win 16060 (DF) (ttl 51,
id 39038)
0x0000   4500 0154 987e 4000 3306 96a9 xxxx 6024     E..T.~@.3.....`$
0x0010   c0a8 1e07 0050 0630 ab64 0786 0006 0f99     .....P.0.d......
0x0020   5018 3ebc a259 0000 4854 5450 2f31 2e31     P.>..Y..HTTP/1.1
0x0030   2032 3030 204f 4b0d 0a44 6174 653a 204d     .200.OK..Date:.M
0x0040   6f6e 2c20 3135 204f 6374 2032 3030 3120     on,.15.Oct.2001.
0x0050   3032 3a32 303a 3237 2047 4d54 0d0a 5365     02:20:27.GMT..Se
0x0060   7276 6572 3a20 4170 6163 6865 2f31 2e33     rver:.Apache/1.3
0x0070   2e32 3020 2855 6e69 7829 2050 4850 2f34     .20.(Unix).PHP/4
0x0080   2e30 2e35 206d 6f64 5f70 6572 6c2f 312e     .0.5.mod_perl/1.
0x0090   3235 206d 6f64 5f73 736c 2f32 2e38 2e34     25.mod_ssl/2.8.4
0x00a0   204f 7065 6e53 534c 2f30 2e39 2e34 0d0a     .OpenSSL/0.9.4..
0x00b0   4c61 7374 2d4d 6f64 6966 6965 643a 2046     Last-Modified:.F
0x00c0   7269 2c20 3331 2041 7567 2032 3030 3120     ri,.31.Aug.2001.
0x00d0   3132 3a34 323a 3032 2047 4d54 0d0a 4554     12:42:02.GMT..ET
0x00e0   6167 3a20 2232 3030 3036 2d65 3661 2d33     ag:."20006-e6a-3
0x00f0   6238 6638 3631 6122 0d0a 4163 6365 7074     b8f861a"..Accept
```

```
0x0100   2d52 616e 6765 733a 2062 7974 6573 0d0a     -Ranges:.bytes..
0x0110   436f 6e74 656e 742d 4c65 6e67 7468 3a20     Content-Length:.
0x0120   3336 3930 0d0a 436f 6e6e 6563 7469 6f6e     3690..Connection
0x0130   3a20 636c 6f73 650d 0a43 6f6e 7465 6e74     :.close..Content
0x0140   2d54 7970 653a 2074 6578 742f 6874 6d6c     -Type:.text/html
0x0150   0d0a 0d0a                                   ....

22:23:14.659963 xxx.xxx.96.36.80 > 192.168.30.7.1584: F 301:301(0) ack 54 win 16060 (DF) (ttl 51,
id 39043)
0x0000   4500 0028 9883 4000 3306 97d0 xxxx 6024     E..(..@.3.....`$
0x0010   c0a8 1e07 0050 0630 ab64 08b2 0006 0f99     .....P.0.d......
0x0020   5011 3ebc 8f65 0000 0000 0000 0000          P.>..e........

22:23:14.660139 192.168.30.7.1584 > xxx.xxx.96.36.80: . 54:54(0) ack 302 win 8460 (DF) (ttl 128,
id 43774)
0x0000   4500 0028 aafe 4000 8006 3855 c0a8 1e07     E..(..@...8U....
0x0010   xxxx 6024 0630 0050 0006 0f99 ab64 08b3     ..`$.0.P.....d..
0x0020   5010 210c ad15 0000                         P.!.....

22:23:14.720111 192.168.30.7.1584 > xxx.xxx.96.36.80: F 54:54(0) ack 302 win 8460 (DF) (ttl 128,
id 44030)
0x0000   4500 0028 abfe 4000 8006 3755 c0a8 1e07     E..(..@...7U....
0x0010   xxxx 6024 0630 0050 0006 0f99 ab64 08b3     ..`$.0.P.....d..
0x0020   5011 210c ad14 0000                         P.!.....

22:23:14.750484 xxx.xxx.96.36.80 > 192.168.30.7.1584: . 302:302(0) ack 55 win 16060 (DF) (ttl 51,
id 39046)
0x0000   4500 0028 9886 4000 3306 97cd xxxx 6024     E..(..@.3.....`$
0x0010   c0a8 1e07 0050 0630 ab64 08b3 0006 0f9a     .....P.0.d......
0x0020   5010 3ebc 8f64 0000 0000 0000 0000          P.>..d........
```

{ Once PortProbe receives the data it requested, it is finished with this host and gracefully closes the connection. The following text file is the result of the above probe.



```
Sun 10/14/01 10:21:07 PM


Port scanned for: 80


Target IP

xxx.xxx.96.36


Comments

HTTP/1.1 200 OK
Date: Mon, 15 Oct 2001 02:20:27 GMT
Server: Apache/1.3.20 (Unix) PHP/4.0.5 mod_perl/1.25 mod_ssl/2.8.4 OpenSSL/0.9.6
Last-Modified: Fri, 31 Aug 2001 12:42:02 GMT
ETag: "2a0b6-e6a-3b8f961a"
Accept-Ranges: bytes
Content-Length: 3690
Connection: close
Content-Type: text/html
```

Fig. 1-19 Results of PortProbe against a webserver.

In conclusion, I think it is easy to see the potential for this tool. Unfortunately though, it doesn't come without some "quirks". Some of the deficiencies in this tool I hope to address in the future are such items as the currently somewhat limited options of changing the timeout values when scanning a range of hosts and to clean up some of the interfaces. Of course, I hope to add more functionality and improve its “stealthiness” so it can evolve to become an even more useful utility.

Whitepaper References

PortProbe v0.10 BETA
URL: http://www.whitehats.ca/main/members/Chris/Chris.html

VMware. "Desktop Products -- VMware Workstation -- Features."
URL: http://www.vmware.com/products/desktop/ws_features.html

SecurityFocus
URL: http://www.securityfocus.com

SSH Communications Security
URL: http://www.ssh.com

OpenSSH
URL: http://www.openssh.org

WinDump: TCPDump for Windows
URL: http://netgroup-serv.polito.it/windump/

WinBatch
URL: http://www.winbatch.com

Assignment 2 - Network Detects

Some of the data presented in this section of the practical are the results of detects collected on the authors firewall computer using a combination of TCPDump, Snort and logs from Shadow IDS v1.5 powered by Slackware Linux packaged for distribution by Guy Bruneau.

My network configuration is as follows. I am a cable modem user, so I have a cat 5 connection running from the cable modem to the eth0 port of my dual-homed firewall computer which currently is a P200 with 256MB RAM running the Shadow IDS v1.5 powered by Slackware Linux mentioned earlier. The firewall box is using a default installation of the Shadow IDS package which installs Shadow v1.7. This same host is also running Snort Version 1.8.1-RELEASE (Build 74). Unfortunately, do to configuration issues, I don't have both Shadow and Snort captures for all traces. I then have a cat 5 cable leaving the firewall computer on eth1 and going into a small Clear Signal MicroHub-4. From this hub I have my main computer plugged in, which is as dual Celeron 533 with 640MB of RAM dual booting between Microsoft Windows 2000 and Slackware Linux 8.0. Also currently hooked up the hub is a P266 laptop running Slackware Linux 8.0.

Any of my personal IP addresses in the logs have been obfuscated. I have left the IPs and host names of the "questionable" hosts unaltered

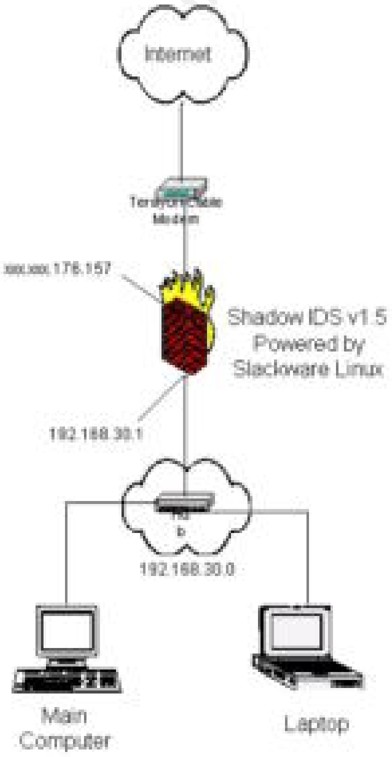


Fig. 2-1 My Network Configuration

Detect 1 - TCP port 27374 probe.

Capture 1 -

```
17:14:54.430660 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 62: 10.10.0.12.2536 > xx.xxx.xxx.157.27374: S
[tcp sum ok] 3348854476:3348854476(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 109, id
55643, len 48)
0x0000 4500 0030 d95b 4000 6d06 611d 0a0a 000c E..0.[@.m.a.....
0x0010 xxxx xx9d 09e8 6aee c79b 76cc 0000 0000 ......j...v.....
0x0020 7002 4000 bc91 0000 0204 05b4 0101 0402 p.@.............

17:14:57.232636 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 62: 10.10.0.12.2536 > xx.xxx.xxx.157.27374: S
[tcp sum ok] 3348854476:3348854476(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 109, id
57593, len 48)
0x0000 4500 0030 e0f9 4000 6d06 597f 0a0a 000c E..0..@.m.Y.....
0x0010 xxxx xx9d 09e8 6aee c79b 76cc 0000 0000 ......j...v.....
0x0020 7002 4000 bc91 0000 0204 05b4 0101 0402 p.@.............

17:15:03.220100 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 62: 10.10.0.12.2536 > xx.xxx.xxx.157.27374: S
[tcp sum ok] 3348854476:3348854476(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 109, id
59421, len 48)
0x0000 4500 0030 e81d 4000 6d06 525b 0a0a 000c E..0..@.m.R[....
0x0010 xxxx xx9d 09e8 6aee c79b 76cc 0000 0000 ......j...v.....
0x0020 7002 4000 bc91 0000 0204 05b4 0101 0402 p.@.............
```

Capture 2 -

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

10/07-17:14:54.430660 0:0:77:95:5D:56 -> X:X:X:XX:XX:XX type:0x800 len:0x3E
10.10.0.12:2536 -> xx.xxx.xxx.157:27374 TCP TTL:109 TOS:0x0 ID:55643 IpLen:20 DgmLen:48 DF
******S* Seq: 0xC79B76CC Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
0x0000: 00 01 02 3C 62 BB 00 00 77 95 5D 56 08 00 45 00 ...<b...w.]V..E.
0x0010: 00 30 D9 5B 40 00 6D 06 61 1D 0A 0A 00 0C xx xx .0.[@.m.a.......
0x0020: xx 9D 09 E8 6A EE C7 9B 76 CC 00 00 00 00 70 02 ....j...v.....p.
0x0030: 40 00 BC 91 00 00 02 04 05 B4 01 01 04 02 @............

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

10/07-17:14:57.232636 0:0:77:95:5D:56 -> X:X:X:XX:XX:XX type:0x800 len:0x3E
10.10.0.12:2536 -> xx.xxx.xxx.157:27374 TCP TTL:109 TOS:0x0 ID:57593 IpLen:20 DgmLen:48 DF
******S* Seq: 0xC79B76CC Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
0x0000: 00 01 02 3C 62 BB 00 00 77 95 5D 56 08 00 45 00 ...<b...w.]V..E.
0x0010: 00 30 E0 F9 40 00 6D 06 59 7F 0A 0A 00 0C xx xx .0..@.m.Y.......
0x0020: xx 9D 09 E8 6A EE C7 9B 76 CC 00 00 00 00 70 02 ....j...v.....p.
0x0030: 40 00 BC 91 00 00 02 04 05 B4 01 01 04 02 @............

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

10/07-17:15:03.220100 0:0:77:95:5D:56 -> X:X:X:XX:XX:XX type:0x800 len:0x3E
10.10.0.12:2536 -> xx.xxx.xxx.157:27374 TCP TTL:109 TOS:0x0 ID:59421 IpLen:20 DgmLen:48 DF
******S* Seq: 0xC79B76CC Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
0x0000: 00 01 02 3C 62 BB 00 00 77 95 5D 56 08 00 45 00 ...<b...w.]V..E.
0x0010: 00 30 E8 1D 40 00 6D 06 52 5B 0A 0A 00 0C xx xx .0..@.m.R[......
0x0020: xx 9D 09 E8 6A EE C7 9B 76 CC 00 00 00 00 70 02 ....j...v.....p.
0x0030: 40 00 BC 91 00 00 02 04 05 B4 01 01 04 02 @............

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

1. Source of Trace:

These captures have been detected on my firewall system. Please see the configuration above.

2. Detect was generated by:
The filter used on shadow to obtain Capture 1 was: "`-nveX ip and host 10.10.0.12 and xx.xxx.xxx.157`". Capture 1 was captured through Shadow while Capture 2 was obtained from the Snort IDS software running on the same host. A breakdown of all the applicable fields from these logs can be found later in this document in the Log Files Explained section. The rule that made snort take notice of this is a basic rule as follows:

```
alert tcp any any -> $HOME_NET 27374 (flags: S; msg: "Possible Trojan probe to port 27374";)
```

3. Probability the source address was spoofed:
It is my belief that this traffic is probably not spoofed. Although upon initial examination of the traffic, seeing the source as coming being from the 10. address block I had thought it might be a spoofed address. As RFC 1918 states

```
"Because private addresses have no global meaning, routing
information about private networks shall not be propagated on
inter-enterprise links, and packets with private source or
destination addresses should not be forwarded across such links.
Routers in networks not using private address space, especially
those of Internet service   providers, are expected to be
configured to reject (filter out) routing information about
private networks."
```

I do know that some Internet providers actually do route 10. addresses as the external IP for my modem is in the 10. block. Now, if this was spoofed, it would be very difficult, but not impossible, to see the response back as the paper by Tom Chmielarski called "Reconnaissance Techniques" dated April 4, 2001 illustrates some of the issues involved with spoofing IP addresses. We know that the source is wanting a reply from this reconnaissance probe, so I do not believe the address to be spoofed.

4. Description of attack:
This traffic appears to be a reconnaissance probe for Trojans as per CVE CAN-1999-0660 (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660) such as SubSeven v2.1 (http://www.nipc.gov/warnings/advisories/2000/00-056.htm) or the Ramen worm (http://xforce.iss.net/alerts/advise71.php).

5. Attack Mechanism:
This activity from the source IP address is a stimulus. By probing a host on port 27374 the source host is trying to gather a list of hosts that reply back with a SYN|ACK, probably for later action. SANS has some information on SubSeven, as well, here is an archived e-mail to a mailing list that contained some good links. SANS also has a good page with a write-up on the Ramen worm.

6. Correlations:
http://www.incidents.org/archives/y2k/021901.htm contains an archive of questionable activity from different source IP addresses submitted by subscribers the intrusions mailing list. SubSeven

is currently one of the most prolific Trojans currently on the Internet, it is very common to see probes for this backdoor appearing daily in log files.

7. Evidence of active targeting:
This is probably not active targeting, but a case of reconnaissance against a range of IPs looking for infected hosts.

8. Severity:
The severity formula as described at http://www.sans.org/giactc/ID_assignment_guidelines.htm is as follows:
Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
Each severity variable is a value of either 1 (being the Lowest) to 5 (being the Highest).
Criticality: **5**, the target hit was a firewall.
Lethality: **1**, this connection attempt for a Trojan hit the Linux box.
System Countermeasures: **5**, the firewall should block inbound access to that port as it is not a required service I am running.
Network Countermeasures: **4**, the probe was blocked at the firewall.
(5 + 1) - (5 + 4) = -3

9. Defensive recommendation:
As this activity was stopped at the firewall, I feel no additional defensive recommendations are required. As Jamie Crapanzano states in the paper "Deconstructing SubSeven, the Trojan Horse of Choice" January 8, 2001, the best way for users to protect themselves is to disable, or have stringent access control to any shares, keep their antivirus definitions up to date and run a personal firewall.

10. Multiple choice test question:

```
17:14:57.232636 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 62: 10.10.0.12.2536 > xx.xxx.xxx.157.27374: S
[tcp sum ok] 3348854476:3348854476(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 109, id
57593, len 48)
0x0000 4500 0030 e0f9 4000 6d06 597f 0a0a 000c E..0..@.m.Y.....
0x0010 xxxx xx9d 09e8 6aee c79b 76cc 0000 0000 ......j...v.....
0x0020 7002 4000 bc91 0000 0204 05b4 0101 0402 p.@.............
```

What Trojan is probably being targeted here?
a) NetBus v1.2
b) SubSeven v2.1
c) Hack a Tack
d) Back Orifice

Answer: B – TCP port 27374 is the default for the SubSeven v2.1 Trojan.

Detect 2 – Large ICMP packets.

Capture 1 -

```
14:27:33.442389 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 1514: 195.241.50.76 > xx.xxx.176.157: icmp:
```

```
echo request (DF) (ttl 233, id 4898, len 1500)
0x0000   4500 05dc 1322 4000 e901 b987 c3f1 324c     E...."@.......2L
0x0010   xxxx b09d 0800 7e52 9abc def0 0000 0000     ......~R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0040   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0060   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0070   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0080   0000 0000 0000 0000                         ........

14:27:40.969466 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 1514: 195.241.50.76 > xx.xxx.176.157: icmp:
echo request (DF) (ttl 233, id 26818, len 1500)
0x0000   4500 05dc 68c2 4000 e901 63e7 c3f1 324c     E...h.@...c...2L
0x0010   xxxx b09d 0800 7e52 9abc def0 0000 0000     ......~R........
0x0020   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0040   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0060   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0070   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0080   0000 0000 0000 0000                         ........
```

Capture 2 -

```
 10/16-14:27:33.442389 0:0:77:95:5D:56 -> X:X:X:XX:XX:XX type:0x800 len:0x5EA
195.241.50.76 -> xx.xxx.176.157 ICMP TTL:233 TOS:0x0 ID:4898 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:48282   Seq:61662  ECHO
0x0000: 00 01 02 3C 62 BB 00 00 77 95 5D 56 08 00 45 00  ...<b...w.]V..E.
0x0010: 05 DC 13 22 40 00 E9 01 B9 87 C3 F1 32 4C xx xx  ..."@.......2L..
0x0020: B0 9D 08 00 7E 52 9A BC DE F0 00 00 00 00 00 00  ....~R..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0x02A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0300: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0310: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0330: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0340: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0350: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0380: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0410: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0420: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0430: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0440: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0450: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0460: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0470: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0480: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0490: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0520: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0530: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0540: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0550: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0560: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0570: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0590: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05E0: 00 00 00 00 00 00 00 00 00 00                    ..........

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

10/16-14:27:40.969466 0:0:77:95:5D:56 -> X:X:X:XX:XX:XX type:0x800 len:0x5EA
195.241.50.76 -> xx.xxx.176.157 ICMP TTL:233 TOS:0x0 ID:26818 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:48282   Seq:61662  ECHO
0x0000: 00 01 02 3C 62 BB 00 00 77 95 5D 56 08 00 45 00  ...<b...w.]V..E.
0x0010: 05 DC 68 C2 40 00 E9 01 63 E7 C3 F1 32 4C xx xx  ..h.@...c...2L..
0x0020: B0 9D 08 00 7E 52 9A BC DE F0 00 00 00 00 00 00  ....~R..........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x01F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x02F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0300: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0310: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0330: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0340: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0350: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0380: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x03F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0410: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0420: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0430: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0440: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0450: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0460: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0470: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0480: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0490: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x04F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0520: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0x0530: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0540: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0550: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0560: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0570: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0590: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05E0: 00 00 00 00 00 00 00 00 00 00                    ..........

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

1. Source of Trace:
These captures have been detected on my firewall system. Please see the configuration above.

2. Detect was generated by:
Capture 1 was pulled from shadow with: "`-n -vvv -e -X host 195.241.50.76 and ip proto \icmp`". Capture 2 was gleaned from my snort logs by replaying the correct date's file and using the options: "host `195.241.50.76`". A breakdown of all the applicable fields from this log can be found later in this document in the Log Files Explained section.

3. Probability the source address was spoofed:
There is good possibility this source address has been spoofed. Since the protocol being used is icmp, there is the possibility the source address is not expecting any kind of return data back from the destination address. According to http://www.securityspace.com/swhois/whois.html the source IP address is registered to the following:

Domain Query Results

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

inetnum:      195.241.50.0 - 195.241.50.255
netname:      WOL-NET-ROUTE-2
descr:        World Online
descr:        Routing equipment
country:      NL
admin-c:      WON2-RIPE
tech-c:       NV133-RIPE
status:       ASSIGNED PA
mnt-by:       WOLTECH-MNT
changed:      n.vogels@nl.worldonline.com 20000421
source:       RIPE

route:        195.241.0.0/16
descr:        World Online BV
origin:       AS5615
mnt-by:       WOLTECH-MNT
```

```
changed:       niels@worldonline.nl 19980811
source:        RIPE

role:          World Online Networks
address:       World Online B.V.
address:       Ir. D.S. Tuynmanweg 10
address:       4131 PN VIANEN
phone:         +31 347 358700
fax-no:        +31 347 358799
e-mail:        beheer@worldonline.nl
admin-c:       JN1555-RIPE
tech-c:        JW100-RIPE
nic-hdl:       WON2-RIPE
mnt-by:        WOLADM-MNT
changed:       n.vogels@nl.worldonline.com 20000215
source:        RIPE

person:        Nils Vogels
address:       Henk Sneevlietweg 2
address:       1066VH Amsterdam
address:       The Netherlands
phone:         +31 20 7986699
fax-no:        +31 20 8629469
e-mail:        nivo@worldonline.nl
nic-hdl:       NV133-RIPE
remarks:       No longer working for WorldOnline.
remarks:       Contact abuse@worldonline.nl if you have any
complains about
remarks:       abusive behaviour of WorldOnline subscribers.
notify:        nivo@worldonline.nl
changed:       nivo@worldonline.nl 20010908
source:        RIPE
```

Fig. 2-2 SecuritySpace whois Results

4. Description of attack:

This is a very large icmp packet received from the source address. This particular icmp packet is 1500 bytes in length when the expected usual icmp packet is around 64 to 128 bytes long. The website whitehats.com has a good description of a large icmp packet event located at http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids246&view=event. It has also been given the CVE designation CVE-1999-0128. There is a very good article by Karen Frederick called "Abnormal IP Packets" last updated Friday, October 13, 2000 located at http://www.securityfocus.com/infocus/1200. In it, Karen talks about the characteristics of abnormal Internet Protocol (IP) packets, and in specific about ICMP traffic, states the following:

> "Most ICMP packets are composed of a small header and payload; for example, most ICMP echo request packets have an 8-byte header and a 56-byte payload. ICMP packets that are significantly larger than normal should be considered suspicious."

The SecurityFocus webpage has many posts from people talking about receiving large icmp packets and is located at http://www.securityfocus.com/cgi-bin/search.pl.

5. Attack Mechanism:
This is an echo request ping with a payload of a lot of padded 0's. Looking closer at the ICMP header, we can break it down a little bit as follows:

```
0x0010   xxxx b09d 0800 7e52 9abc def0 0000 0000     ......~R........
```

The ICMP header starts at HEX0800. Which breaks out to ICMP Type 8, ICMP Code 0 (Echo Request). HEX7e52 is the 16-bit checksum which converts to decimal value of 32338. The values after that (9abc def0 0000 0000) are the start of the data/padding. It is interesting for me to see a pattern of 9abc def0. As I was thumbing through my book by Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Reading: Addison Wesley Longman, Inc, 1994. page 152, I started thinking that one possibility for this traffic might be for Maximum Transmission Unit (MTU) path size discovery as the Don't Fragment (DF) bit is also set on these packets.

6. Correlations:
I was able to find an email from someone on the Internet who has done a lot of work related to ICMP usage on the Internet talking about about large packets, and mtu discovery. Arkin, Ofir "RE: [Snort-users] Large ICMP packets." Sep 29 2000. You can read the text here. I also did a search on the incidents.org website and was able to come up with a number of posts from Internet users talking about receiving large ICMP packets at http://www.incidents.org/cgi-bin/htsearch?method=and&config=htdig&words=large+icmp+packets.

7. Evidence of active targeting:
With so little information to go on, it is hard to say whether this is a case of active targeting. It could be part of a larger event and I am only seeing one little part of the data.

8. Severity:
The severity formula as described at http://www.sans.org/giactc/ID_assignment_guidelines.htm is as follows:
Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
Each severity variable is a value of either 1 (being the Lowest) to 5 (being the Highest).
Criticality: **5**, the destination IP address was a firewall.
Lethality: **1**, it is an ICMP echo request.
System Countermeasures: **5**, the firewall is filtering ICMP traffic.
Network Countermeasures: **4**, the probe was blocked at the firewall.
(5 + 1) - (5 + 4) = -3

9. Defensive recommendation:
This topic can become pretty controversial so I will try to to make it brief. If the decision to block ICMP is chosen, it must be done wisely and carefully. Certain ICMP should be allowed, a minimum of "host unreachable – need to defrag" messages say Stephen Northcutt and Judy Novak suggest in "Network Intrusion Detection: An Analyst's Handbook." 2nd ed. Indianapolis: New Riders, 2000. The bottom line is, you must know what you are allowing and disallowing.

If you actually filter any out, yes you potentially are disallowing malicious ICMP traffic, but a poor configuration can lead to legitimate traffic being denied as well.

10. Multiple choice test question:

```
14:27:40.969466 0:0:77:95:5d:56 X:X:X:XX:XX:XX 0800 1514: 195.241.50.76 > xx.xxx.176.157: icmp:
echo request (DF) (ttl 233, id 26818, len 1500)
```

Given the above traffic, is there anything usual for ICMP traffic?
a) A Time To Live (TTL) of 233 is too high.
b) The (DF) flag set in an icmp packet.
c) The presence of the TTL, ID and LEN flags.
d) The length of 1500 bytes seems suspicious.

Answer: D – Typical ICMP traffic is between 64 to 128 bytes long.

Detect 3 – TCP Ports 1080/23 probe.

Capture -

```
00:27:18.711028 130.227.3.123.1512 > xxx.xxx.114.179.1080: S 2519795512:2519795512(0) win 16384
(DF)
0x0000     4500 002c dfff 4000 2f06 ef31 82e3 037b        E..,..@./..1...{
0x0010     xxxx 72b3 05e8 0438 9631 0738 0000 0000        ..r....8.1.8....
0x0020     6002 4000 3402 0000 0204 05b4 0101             `.@.4.........

00:27:18.711311 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000     4500 0038 9294 0000 ff01 27cb xxxx f77d        E..8......'....}
0x0010     82e3 037b 030d 5569 0000 0000 4500 002c        ...{..Ui....E..,
0x0020     dfff 4000 2e06 f031 82e3 037b xxxx 72b3        ..@....1...{..r.
0x0030     05e8 0438 9631 0738                            ...8.1.8

00:27:21.706544 130.227.3.123.1512 > xxx.xxx.114.179.1080: S 2519795512:2519795512(0) win 16384
(DF)
0x0000     4500 002c e1a9 4000 2f06 ed87 82e3 037b        E..,..@./......{
0x0010     xxxx 72b3 05e8 0438 9631 0738 0000 0000        ..r....8.1.8....
0x0020     6002 4000 3402 0000 0204 05b4 0103             `.@.4.........

00:27:21.706621 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000     4500 0038 92ca 0000 ff01 2795 xxxx f77d        E..8......'....}
0x0010     82e3 037b 030d 5569 0000 0000 4500 002c        ...{..Ui....E..,
0x0020     e1a9 4000 2e06 ee87 82e3 037b xxxx 72b3        ..@........{..r.
0x0030     05e8 0438 9631 0738                            ...8.1.8

00:27:27.719843 130.227.3.123.1512 > xxx.xxx.114.179.1080: S 2519795512:2519795512(0) win 16384
(DF)
0x0000     4500 002c e4d4 4000 2f06 ea5c 82e3 037b        E..,..@./..\...{
0x0010     xxxx 72b3 05e8 0438 9631 0738 0000 0000        ..r....8.1.8....
0x0020     6002 4000 3402 0000 0204 05b4 xxxx             `.@.4.........

00:27:27.719918 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000     4500 0038 92fc 0000 ff01 2763 xxxx f77d        E..8......'c...}
0x0010     82e3 037b 030d 5569 0000 0000 4500 002c        ...{..Ui....E..,
0x0020     e4d4 4000 2e06 eb5c 82e3 037b xxxx 72b3        ..@....\...{..r.
0x0030     05e8 0438 9631 0738                            ...8.1.8
```

```
00:27:39.705115 130.227.3.123.1512 > xxx.xxx.114.179.1080: S 2519795512:2519795512(0) win 16384
(DF)
0x0000    4500 002c eb4d 4000 2f06 e3e3 82e3 037b        E..,.M@./......{
0x0010    xxxx 72b3 05e8 0438 9631 0738 0000 0000        ..r....8.1.8....
0x0020    6002 4000 3402 0000 0204 05b4 0101             `.@.4.........

00:27:39.705381 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000    4500 0038 933a 0000 ff01 2725 xxxx f77d        E..8.:....'%...}
0x0010    82e3 037b 030d 5569 0000 0000 4500 002c        ...{..Ui....E..,
0x0020    eb4d 4000 2e06 e4e3 82e3 037b xxxx 72b3        .M@........{..r.
0x0030    05e8 0438 9631 0738                            ...8.1.8

00:28:03.704496 130.227.3.123.1512 > xxx.xxx.114.179.1080: S 2519795512:2519795512(0) win 16384
(DF)
0x0000    4500 002c f4d4 4000 2f06 da5c 82e3 037b        E..,..@./..\...{
0x0010    xxxx 72b3 05e8 0438 9631 0738 0000 0000        ..r....8.1.8....
0x0020    6002 4000 3402 0000 0204 05b4 0101             `.@.4.........

00:28:03.704804 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000    4500 0038 93c0 0000 ff01 269f xxxx f77d        E..8......&....}
0x0010    82e3 037b 030d 5569 0000 0000 4500 002c        ...{..Ui....E..,
0x0020    f4d4 4000 2e06 db5c 82e3 037b xxxx 72b3        ..@....\...{..r.
0x0030    05e8 0438 9631 0738                            ...8.1.8


00:28:03.828724 130.227.3.123.4211 > xxx.xxx.114.179.telnet: S 2879031709:2879031709(0) win 16384
(DF)
0x0000    4500 002c f4e2 4000 2f06 da4e 82e3 037b        E..,..@./..N...{
0x0010    xxxx 72b3 1073 0017 ab9a 899d 0000 0000        ..r..s..........
0x0020    6002 4000 95c9 0000 0204 05b4 6768             `.@.........gh

00:28:06.825416 130.227.3.123.4211 > xxx.xxx.114.179.telnet: S 2879031709:2879031709(0) win 16384
(DF)
0x0000    4500 002c f608 4000 2f06 d928 82e3 037b        E..,..@./..(...{
0x0010    xxxx 72b3 1073 0017 ab9a 899d 0000 0000        ..r..s..........
0x0020    6002 4000 95c9 0000 0204 05b4 4452             `.@.........DR

00:28:06.825663 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000    4500 0038 93e3 0000 ff01 267c xxxx f77d        E..8......&|...}
0x0010    82e3 037b 030d b730 0000 0000 4500 002c        ...{...0....E..,
0x0020    f608 4000 2e06 da28 82e3 037b xxxx 72b3        ..@....(...{..r.
0x0030    1073 0017 ab9a 899d                            .s......

00:28:12.825258 130.227.3.123.4211 > xxx.xxx.114.179.telnet: S 2879031709:2879031709(0) win 16384
(DF)
0x0000    4500 002c f874 4000 2f06 d6bc 82e3 037b        E..,.t@./......{
0x0010    xxxx 72b3 1073 0017 ab9a 899d 0000 0000        ..r..s..........
0x0020    6002 4000 95c9 0000 0204 05b4 ffff             `.@...........

00:28:12.825335 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000    4500 0038 93f8 0000 ff01 2667 xxxx f77d        E..8......&g...}
0x0010    82e3 037b 030d b730 0000 0000 4500 002c        ...{...0....E..,
0x0020    f874 4000 2e06 d7bc 82e3 037b xxxx 72b3        .t@........{..r.
0x0030    1073 0017 ab9a 899d                            .s......

00:28:24.824532 130.227.3.123.4211 > xxx.xxx.114.179.telnet: S 2879031709:2879031709(0) win 16384
(DF)
0x0000    4500 002c fdc2 4000 2f06 d16e 82e3 037b        E..,..@./..n...{
0x0010    xxxx 72b3 1073 0017 ab9a 899d 0000 0000        ..r..s..........
0x0020    6002 4000 95c9 0000 0204 05b4 0d0a             `.@...........

00:28:24.824787 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000    4500 0038 9442 0000 ff01 261d xxxx f77d        E..8.B....&....}
0x0010    82e3 037b 030d b730 0000 0000 4500 002c        ...{...0....E..,
0x0020    fdc2 4000 2e06 d26e 82e3 037b xxxx 72b3        ..@....n...{..r.
0x0030    1073 0017 ab9a 899d                            .s......
```

```
00:28:48.824938 130.227.3.123.4211 > xxx.xxx.114.179.telnet: S 2879031709:2879031709(0) win 16384
(DF)
0x0000     4500 002c 0842 4000 2f06 c6ef 82e3 037b        E..,.B@./......{
0x0010     xxxx 72b3 1073 0017 ab9a 899d 0000 0000        ..r..s..........
0x0020     6002 4000 95c9 0000 0204 05b4 0101             `.@...........

00:28:48.825196 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000     4500 0038 94d4 0000 ff01 258b xxxx f77d        E..8......%....}
0x0010     82e3 037b 030d b730 0000 0000 4500 002c        ...{...0....E..,
0x0020     0842 4000 2e06 c7ef 82e3 037b xxxx 72b3        .B@........{..r.
0x0030     1073 0017 ab9a 899d                            .s......
```

1. Source of Trace:

This trace was actually logged on my network at work between the border router and internal firewalls.

2. Detect was generated by:

Traffic was pulled from Shadow v1.6 IDS host. A breakdown of all the applicable fields from these logs can be found later in this document in the Log Files Explained section.

3. Probability the source address was spoofed:

It is my feeling this source address is not being spoofed. This appears to be reconnaissance activity possibly searching for proxy servers (commonly using TCP port 1080), further on in the detect we notice the source host try a telnet connection to one of our monitored hosts. For this connection to be successful, TCP requires a successful 3 way handshake between the source and the destination host. As this is connection oriented, it would be very difficult to spoof this connection.

The following is an excerpt from the http://www.ripe.net whois database containing the registration information for the source IP address. The RIPE Network Coordination Center (RIPE NCC) is one of 3 Regional Internet Registries that exist in the world today. It covers Europe, The Middle East, The North of Africa and parts of Asia.

130.227.3.123

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


inetnum:      130.227.0.0 - 130.227.255.255
netname:      DK-NETCOM-19971002
descr:        UNI2 Internet for professionelle
             Gl. Koege landevej 55
              DK-2500 Valby
country:      DK
admin-c:      UNI2-DK
tech-c:       UNI2-DK
status:       ALLOCATED PA
```

```
remarks:       was DENET-227
              If you have any complaints regarding a user from this
              IP range, please contact abuse@uni2.dk regarding this
              issue.
mnt-by:        RIPE-NCC-HM-MNT
mnt-lower:     AS5492-MNT
changed:       domain@uni2.dk 19971002
changed:       hostmaster@ripe.net 20010419
changed:       hostmaster@ripe.net 20010507
source:        RIPE


route:         130.227.0.0/16
descr:         TELE2 A/S Danmark
origin:        AS5492
mnt-by:        AS5492-MNT
changed:       jo@uni2.dk 20001227
source:        RIPE


role:          UNI2 Internet for professionelle
address:       UNI2
address:       Gl. Koege landevej 55
address:       DK-2500 Valby, Denmark
phone:         +45 77 30 12 00
fax-no:        +45 77 30 10 00
e-mail:        domain@uni2.dk
admin-c:       MLO-RIPE
admin-c:       JO67-RIPE
admin-c:       HBH2-RIPE
tech-c:        MLO-RIPE
tech-c:        JO67-RIPE
tech-c:        HBH2-RIPE
nic-hdl:       UNI2-DK
```

Fig. 2-3 RIPE whois Results

4. Description of attack:

We have a remote host sending SYN packets to destination TCP ports 1080 and 23 our host. Typically, TCP port 1080 is used for the SOCKS proxy service but according to NetworkICE, in reference to connection attempts to port 1080, "Most scans for port 1080 are actually looking for WinGate, a popular firewall/proxy for Windows. BugTraq vulnerability 509 "Qbik WinGate Buffer Overflow DoS Vulnerability" contains a description of the vulnerability probably being searched for. CVE-1999-0441 is the reference a vulnerability in the Wingate service. After a number of attempts to connect to port 1080, the remote source tries a telnet connection to our host. Users want to find vulnerable web proxy servers so they can use it to hide there identity. If they are able to use a proxy server, it will not log their real IP address, but the address of the proxy server in the course of their activities.

5. Attack Mechanism:

In this detect, we have a remote host sending packets to an IP address in our protected subnet. Each packet sent to destination port 1080 has a source port of 1512 and as noted in para 4, the most likely reason for probing this port is an attempt to perform a recon for hosts acting as proxies. After 5 connection attempts to port 1080, the source computer switches to attempting to connect on port 23 which is the well known port for the telnet service. Again, there are 5 connection attempts to this port, all with the same source port of 4211. I felt that the Time to Live (TTL) for the source host was a low number (47) which might indicate the use a Linux machine as the source computer.

6. Correlations:
The incidents.org website has quite a number of messages from people reporting remote connection attempts destination IP addresses under the control. This information can be found at http://www.incidents.org/cgi-bin/htsearch?method=and&config=htdig&words=port+1080.

7. Evidence of active targeting:
Without more data to base my opinion on, it is hard to say if this is for sure active targeting. Although all of the traffic was destined for one specific destination IP address, this IP address wasn't offering any of the services (proxy or telnet) that connection attempts were made for. I would venture to say that this is reconnaissance activity.

8. Severity:
The severity formula as described at http://www.sans.org/giactc/ID_assignment_guidelines.htm is as follows:
Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
Each severity variable is a value of either 1 (being the Lowest) to 5 (being the Highest).
Criticality: **2**, the target was a users computer.
Lethality: **1**, as this is a users computer, and not a server.
System Countermeasures: **2**, unable to confirm exactly what operating system the user is running.
Network Countermeasures: **4**, the activity was stopped at the border router.
(2 + 1) - (2 + 4) = -3

9. Defensive recommendation:
While the border did not allow access to our protected host, it is alarming that the router responded back with the icmp error messages, even though that is what the RFC's say should happen. As J. Reynolds and J. Postel note in RFC 1700 STD: 2 ICMP Type 3 Code 9 refers to a destination host unreachable due to "Communication with Destination Network is Administratively Prohibited."

It is a question of policy whether this kind of error messages should be allowed back through the border routers and back to the source host. In RFC 1009, R. Braden and J. Postel state:

"Net unreachable implies that an intermediate gateway was unable to forward a datagram, as its routing data-base gave no next hop for the datagram, or all paths were down. Host Unreachable implies that the destination network was reachable, but that a gateway on that network was unable to reach the destination host."

They further go on say:

```
"Gateways should send Host Unreachable messages whenever other
hosts on the same destination network might be reachable;
otherwise, the source host may erroneously conclude that ALL
hosts on the network are unreachable, and that may not be the
case."
```

The problem with this information leaving your network is that is very valuable information for a potential attacker and can be used to provide a very accurate picture of your network configuration and architecture. It's possible to "silence" a Cisco router by entering the statement "no ip unreachables" in the configuration file. This will prevent the router from broadcasting these icmp unreachable messages back to hosts on the Internet. The less information a potential attacker can get from our network, the better.

10. Multiple choice test question:

```
00:27:27.719843 130.227.3.123.1512 > xxx.xxx.114.179.1080: S 2519795512:2519795512(0) win 16384
(DF)
0x0000     4500 002c e4d4 4000 2f06 ea5c 82e3 037b       E..,..@./..\...{
0x0010     xxxx 72b3 05e8 0438 9631 0738 0000 0000       ..r....8.1.8....
0x0020     6002 4000 3402 0000 0204 05b4 xxxx            `.@.4.........

00:27:27.719918 xxx.xxx.247.125 > 130.227.3.123: icmp: host xxx.xxx.114.179 unreachable - admin
prohibited filter
0x0000     4500 0038 92fc 0000 ff01 2763 xxxx f77d       E..8......'c...}
0x0010     82e3 037b 030d 5569 0000 0000 4500 002c       ...{..Ui....E..,
0x0020     e4d4 4000 2e06 eb5c 82e3 037b xxxx 72b3       ..@....\...{..r.
0x0030     05e8 0438 9631 0738                           ...8.1.8
```

Looking at this traffic, what configuration changes could be made in the interest of security?
a) Turn of ip unreachables from the router.
b) Block incoming ICMP.
c) Install a firewall / IDS.
d) Block outgoing access to port 1080.

Answer: A – This is valuable information for remote hosts to use in mapping out a network.

Detect 4 - TCP probe to port 515.

Capture -

```
00:36:39.359899 255.255.255.255.31337 > xxx.xxx.132.131.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 cd12 ffff ffff E..(...........
0x0010 xxxx 8483 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 2906 0000 3232 3620 5472 P...)...226.Tr

00:45:19.788763 255.255.255.255.31337 > xxx.xxx.42.71.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 274f ffff ffff E..(......'O....
0x0010 xxxx 2a47 7a69 0203 0000 0064 0000 0000 ..*Gzi.....d....
0x0020 5002 0200 8342 0000 0204 05b4 0101 P....B........
```

```
00:59:15.412564 255.255.255.255.31337 > xxx.xxx.155.192.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 b5d5 ffff ffff E..(............
0x0010 xxxx 9bc0 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 11c9 0000 0204 0218 0364 P............d

01:07:36.658826 255.255.255.255.31337 > xxx.xxx.101.174.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 ebe7 ffff ffff E..(............
0x0010 xxxx 65ae 7a69 0203 0000 0064 0000 0000 ..e.zi.....d....
0x0020 5002 0200 47db 0000 0101 080a 0b9c P...G.........

02:42:07.655987 255.255.255.255.31337 > xxx.xxx.94.172.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 f2e9 ffff ffff E..(............
0x0010 xxxx 5eac 7a69 0203 0000 0064 0000 0000 ..^.zi.....d....
0x0020 5002 0200 4edd 0000 0204 05b4 0101 P...N.........

03:15:36.416114 255.255.255.255.31337 > xxx.xxx.41.156.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 27fa ffff ffff E..(......'.....
0x0010 xxxx 299c 7a69 0203 0000 0064 0000 0000 ..).zi.....d....
0x0020 5002 0200 83ed 0000 34ea 9373 c076 P.......4..s.v

03:36:08.506440 255.255.255.255.31337 > xxx.xxx.140.208.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 c4c5 ffff ffff E..(............
0x0010 xxxx 8cd0 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 20b9 0000 0204 05b4 0101 P.............

04:11:52.504278 255.255.255.255.31337 > xxx.xxx.65.2.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 1094 ffff ffff E..(............
0x0010 xxxx 4102 7a69 0203 0000 0064 0000 0000 ..A.zi.....d....
0x0020 5002 0200 6c87 0000 0204 05b4 38d8 P...l.......8.

04:29:48.201037 255.255.255.255.31337 > xxx.xxx.80.7.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 018f ffff ffff E..(............
0x0010 xxxx 5007 7a69 0203 0000 0064 0000 0000 ..P.zi.....d....
0x0020 5002 0200 5d82 0000 3c04 05b4 0101 P...]...<.....

05:14:08.964111 255.255.255.255.31337 > xxx.xxx.71.57.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 0a5d ffff ffff E..(.......]....
0x0010 xxxx 4739 7a69 0203 0000 0064 0000 0000 ..G9zi.....d....
0x0020 5002 0200 6650 0000 6134 e131 ec69 P...fP..a4.1.i

06:50:25.296348 255.255.255.255.31337 > xxx.xxx.174.98.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 a333 ffff ffff E..(.......3....
0x0010 xxxx ae62 7a69 0203 0000 0064 0000 0000 ...bzi.....d....
0x0020 5002 0200 ff26 0000 0204 05b4 0103 P....&........

08:23:17.119842 255.255.255.255.31337 > xxx.xxx.5.69.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 4c51 ffff ffff E..(......LQ....
0x0010 xxxx 0545 7a69 0203 0000 0064 0000 0000 ...Ezi.....d....
0x0020 5002 0200 a844 0000 0204 05b4 0101 P....D........

09:01:04.856761 255.255.255.255.31337 > xxx.xxx.1.144.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 5006 ffff ffff E..(......P.....
0x0010 xxxx 0190 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 abf9 0000 4238 e866 8389 P.......B8.f..

09:08:44.794645 255.255.255.255.31337 > xxx.xxx.151.227.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 b9b2 ffff ffff E..(............
0x0010 xxxx 97e3 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 15a6 0000 0204 05b4 e36f P...........o

10:11:29.068832 255.255.255.255.31337 > xxx.xxx.151.102.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 ba2f ffff ffff E..(......./....
0x0010 xxxx 9766 7a69 0203 0000 0064 0000 0000 ...fzi.....d....
0x0020 5002 0200 1623 0000 ce53 7195 123c P....#...Sq..<
```

```
10:27:16.936955 255.255.255.255.31337 > xxx.xxx.142.221.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 c2b8 ffff ffff E..(...........
0x0010 xxxx 8edd 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 1eac 0000 0204 0564 0101 P..........d..

11:00:08.827426 255.255.255.255.31337 > xxx.xxx.41.29.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 2879 ffff ffff E..(......(y....
0x0010 xxxx 291d 7a69 0203 0000 0064 0000 0000 ..).zi.....d....
0x0020 5002 0200 846c 0000 07f3 ba00 964f P....l.......O

11:28:45.575980 255.255.255.255.31337 > xxx.xxx.78.214.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 02c0 ffff ffff E..(...........
0x0010 xxxx 4ed6 7a69 0203 0000 0064 0000 0000 ..N.zi.....d....
0x0020 5002 0200 5eb3 0000 0232 3803 3133 P...^....28.13

11:51:43.998554 255.255.255.255.31337 > xxx.xxx.230.152.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f006 6dfd ffff ffff E..(......m.....
0x0010 xxxx e698 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 c6f0 0000 0577 696e 6573 P........wines

11:55:45.706337 255.255.255.255.31337 > xxx.xxx.134.203.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 caca ffff ffff E..(...........
0x0010 xxxx 86cb 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 26be 0000 35fb eb6b bb29 P...&...5..k.)
```

1. Source of Trace:

This trace was actually logged on my network at work between the border router and internal firewalls.

2. Detect was generated by:

This capture was detected on a Shadow v1.6 sensor. A breakdown of all the applicable fields from these logs can be found later in this document in the Log Files Explained section.

3. Probability the source address was spoofed:

I feel it is obvious that in this case, the source address is being spoofed. It is the opinion of Patrick Nolan in his post related to similar traffic dated Wed, 2 May 2001 14:02:43 -0400 to incidents.org that "The spoofed source address is trying to trigger a compromised machine to send a response. The response desired is obviously not back to the source. The response on a compromised machine is most likely a service on the local machine that will send legitimate looking outbound traffic with the headers carrying the payload." Paragraph (a), section 3.2.1.3 of RFC 1122 specifically states that an address: “that contains all 1 bits.” (ie: convert the decimal bitmask to binary) “MUST NOT be used as a source address.”   The difficulty in using such a spoofed source address is that no response can be received, except by a host on the same broadcast domain that has their network card in promiscuous mode.

4. Description of attack:

This capture displays the results of a mysterious source IP address of 255.255.255.255 probing individual hosts from different subnets over the course of a 24 hour period. The destination port being targeted in this detect usually runs the in.lpd service for Unix hosts while Windows hosts can have the "Windows Services for Unix" installed. CVE-2000-0232: states an attacker can cause a denial of service via a malformed TCP/IP print request for Windows hosts. CVE-2001-0353: refers to Solaris 8 and earlier allowing local and remote attackers to gain root privileges via a "transfer job" routine under Unix. Something to note about this capture is that even over such a large time period, the packets being sent from the source are **exactly** the same for each host it is probing. If you were to break out the packets, you would find that up until you reach the 16-bit window size in the TCP header that the **only** difference in the whole packet is the destination address. This is obviously the work of some kind of tool using the same source port (31337), same ISN (100), same TTL (243). When someone sees the port 31337 in traffic, they usually automatically think of BackOrifice but that is 31337/UDP. Port 31337/TCP has been associated the Elite before. "31337" reads "ELEET" (Elite) in the hacker lingo.

5. Attack Mechanism:

A very interesting capture is displayed here. There are a number of packets being sent from a 255.255.255.255 source address with a source port of 31337 to a range of different subnets and hosts looking for a response on port 515. Each connection attmempt is from the same port (31337) and has the exact same Initial Sequence Number of 100 and a small window size of 512. As the SANS course material "Network Traffic Analysis Using TCP Dump" book 3.2 states:

```
 "Since TCP is a connection-oriented, reliable protocol , we
have to have a mechanism to account for data being sent and
received.  In part, that is done using TCP sequence numbers.
These sequence number should never be repeated unless there is a
retry of the same connection. The initial sequence number (ISN)
is the first sequence number that is used in the TCP exchange
between the sending hosts.  Each host in the exchange selects a
unique sequence number when sending the initial SYN connection
to the other host."
```

This capture is stimulus showing many SYN connection attempts. It is hoped that by sending SYN packets, this probe will get a response back in order to get a three way handshake performed. One hypothesis here is that as the source IP addresses are being spoofed, there is no intention or requirement for the source IP to get any response back, that this might be something of client/server scenario where commands or instructions are being sent to random hosts, in the hopes of finding a listening host that can carry out the desired action.

6. Correlations:

The first search I performed on the incidents.org website, I found a post that exactly depicts the traffic I am seeing. It is dated Wed, 02 May 2001 16:32:24 -0400 and is from Fred Portnoy and can be found at http://www.incidents.org/archives/intrusions/msg00020.html which goes into a little detail about the traffic. The incidents.org website contains a number of other posts correlating the interest from remote hosts trying to recon networks offering port 515 services.

7. Evidence of active targeting:

From the research I have done with this particular capture, and with the knowledge of the other sites receiving some reconnaissance activity as well with the same characteristics, I do not feel this to be active targeting against this site in particular, but part of a bigger probe for any hosts able to respond.

8. Severity:

The severity formula as described at http://www.sans.org/giactc/ID_assignment_guidelines.htm

is as follows:
Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
Each severity variable is a value of either 1 (being the Lowest) to 5 (being the Highest).
Criticality: **3**, the targets are users systems.
Lethality: **4**, if a user was operating a printer service, there is a potential for compromise.
System Countermeasures: **5**, the firewall should block inbound access to that port from outside the perimeter.
Network Countermeasures: **3**, a borderline restrictive/permissive firewall is in place.
(3 + 4) - (5 + 3) = -1

9. Defensive recommendation:

It is recommended to ensure that if this service is not required on any hosts, it is disabled and this port is being blocked at the perimeter.

10. Multiple choice test question:

```
06:50:25.296348 255.255.255.255.31337 > xxx.xxx.174.98.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 a333 ffff ffff E..(.......3....
0x0010 xxxx ae62 7a69 0203 0000 0064 0000 0000 ...bzi.....d....
0x0020 5002 0200 ff26 0000 0204 05b4 0103 P....&........

08:23:17.119842 255.255.255.255.31337 > xxx.xxx.5.69.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 4c51 ffff ffff E..(......LQ....
0x0010 xxxx 0545 7a69 0203 0000 0064 0000 0000 ...Ezi.....d....
0x0020 5002 0200 a844 0000 0204 05b4 0101 P....D........

09:01:04.856761 255.255.255.255.31337 > xxx.xxx.1.144.printer: S 100:100(0) win 512
0x0000 4500 0028 f2b0 0000 f306 5006 ffff ffff E..(......P.....
0x0010 xxxx 0190 7a69 0203 0000 0064 0000 0000 ....zi.....d....
0x0020 5002 0200 abf9 0000 4238 e866 8389 P.......B8.f..
```

Looking at this traffic, which of the following options doesn’t show evidence of packet crafting?
a) Destination port of 515.
b) Source address of 255.255.255.255.
c) Same Initial Sequence Number (ISN) for all packets.
d) Same source port for all packets, even though different destination addresses.

Answer: A – The use of an invalid source address, non-changing sequence numbers and a source port of 31337 for all packets really looks like forged packets.

Detect 5 - TCP Port 111 probe.

Capture -

```
13:36:59.586881 216.221.215.20.4281 > xxx.xxx.xxx.51.sunrpc: S 1969667479:1969667479(0) win 32120
(DF)
0x0000 4500 003c c933 4000 3806 03c1 d8dd d714 E..<.3@.8.......
0x0010 xxxx xx33 10b9 006f 7566 bd97 0000 0000 ...3...ouf......
0x0020 a002 7d78 e808 0000 0204 05b4 0402 080a ..}x............
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........


13:36:59.588088 216.221.215.20.4283 > xxx.xxx.xxx.53.sunrpc: S 1958127494:1958127494(0) win 32120
(DF)
0x0000 4500 003c c935 4000 3806 03bd d8dd d714 E..<.5@.8.......
```

```
0x0010 xxxx xx35 10bb 006f 74b6 a786 0000 0000 ...5...ot.......
0x0020 a002 7d78 fec5 0000 0204 05b4 0402 080a ..}x............
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........

13:36:59.589084 216.221.215.20.4285 > xxx.xxx.xxx.55.sunrpc: S 1958190074:1958190074(0) win 32120
(DF)
0x0000 4500 003c c937 4000 3806 03b9 d8dd d714 E..<.7@.8.......
0x0010 xxxx xx37 10bd 006f 74b7 9bfa 0000 0000 ...7...ot.......
0x0020 a002 7d78 0a4d 0000 0204 05b4 0402 080a ..}x.M..........
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........

13:36:59.589699 216.221.215.20.4287 > xxx.xxx.xxx.57.sunrpc: S 1966877905:1966877905(0) win 32120
(DF)
0x0000 4500 003c c939 4000 3806 03b5 d8dd d714 E..<.9@.8.......
0x0010 xxxx xx39 10bf 006f 753c 2cd1 0000 0000 ...9...ou<,.....
0x0020 a002 7d78 78ed 0000 0204 05b4 0402 080a ..}xx...........
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........

13:36:59.591719 216.221.215.20.4289 > xxx.xxx.xxx.59.sunrpc: S 1964382413:1964382413(0) win 32120
(DF)
0x0000 4500 003c c93b 4000 3806 03b1 d8dd d714 E..<.;@.8.......
0x0010 xxxx xx3b 10c1 006f 7516 18cd 0000 0000 ...;...ou.......
0x0020 a002 7d78 8d13 0000 0204 05b4 0402 080a ..}x............
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........

13:36:59.592358 216.221.215.20.4290 > xxx.xxx.xxx.60.sunrpc: S 1964541347:1964541347(0) win 32120
(DF)
0x0000 4500 003c c93c 4000 3806 03af d8dd d714 E..<.<@.8.......
0x0010 xxxx xx3c 10c2 006f 7518 85a3 0000 0000 ...<...ou.......
0x0020 a002 7d78 2039 0000 0204 05b4 0402 080a ..}x.9..........
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........

13:36:59.593545 216.221.215.20.4292 > xxx.xxx.xxx.62.sunrpc: S 1958873814:1958873814(0) win 32120
(DF)
0x0000 4500 003c c93e 4000 3806 03ab d8dd d714 E..<.>@.8.......
0x0010 xxxx xx3e 10c4 006f 74c2 0ad6 0000 0000 ...>...ot.......
0x0020 a002 7d78 9b58 0000 0204 05b4 0402 080a ..}x.X..........
0x0030 02c5 25d3 0000 0000 0103 0300 ..%.........

13:37:02.547826 216.221.215.20.4280 > xxx.xxx.xxx.50.sunrpc: S 1963718756:1963718756(0) win 32120
(DF)
0x0000 4500 003c cfc9 4000 3806 fd2b d8dd d714 E..<..@.8..+....
0x0010 xxxx xx32 10b8 006f 750b f864 0000 0000 ...2...ou..d....
0x0020 a002 7d78 ac6c 0000 0204 05b4 0402 080a ..}x.l..........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.548613 216.221.215.20.4281 > xxx.xxx.xxx.51.sunrpc: S 1969667479:1969667479(0) win 32120
(DF)
0x0000 4500 003c cfca 4000 3806 fd29 d8dd d714 E..<..@.8..)....
0x0010 xxxx xx33 10b9 006f 7566 bd97 0000 0000 ...3...ouf......
0x0020 a002 7d78 e6dc 0000 0204 05b4 0402 080a ..}x............
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.549469 216.221.215.20.4282 > xxx.xxx.xxx.52.sunrpc: S 1964527505:1964527505(0) win 32120
(DF)
0x0000 4500 003c cfcb 4000 3806 fd27 d8dd d714 E..<..@.8..'....
0x0010 xxxx xx34 10ba 006f 7518 4f91 0000 0000 ...4...ou.O.....
0x0020 a002 7d78 552f 0000 0204 05b4 0402 080a ..}xU/..........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.550008 216.221.215.20.4283 > xxx.xxx.xxx.53.sunrpc: S 1958127494:1958127494(0) win 32120
(DF)
0x0000 4500 003c cfcc 4000 3806 fd25 d8dd d714 E..<..@.8..%....
0x0010 xxxx xx35 10bb 006f 74b6 a786 0000 0000 ...5...ot.......
0x0020 a002 7d78 fd99 0000 0204 05b4 0402 080a ..}x............
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........
```

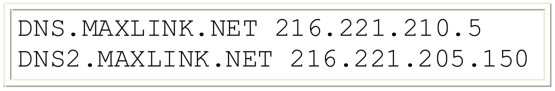
```
13:37:02.550611 216.221.215.20.4288 > xxx.xxx.xxx.58.sunrpc: S 1960996598:1960996598(0) win 32120
(DF)
0x0000 4500 003c cfd1 4000 3806 fd1b d8dd d714 E..<..@.8.......
0x0010 xxxx xx3a 10c0 006f 74e2 6ef6 0000 0000 ...:...ot.n.....
0x0020 a002 7d78 35f4 0000 0204 05b4 0402 080a ..}x5...........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.551401 216.221.215.20.4286 > xxx.xxx.xxx.56.sunrpc: S 1965581948:1965581948(0) win 32120
(DF)
0x0000 4500 003c cfcf 4000 3806 fd1f d8dd d714 E..<..@.8.......
0x0010 xxxx xx38 10be 006f 7528 667c 0000 0000 ...8...ou(f|....
0x0020 a002 7d78 3e2c 0000 0204 05b4 0402 080a ..}x>,..........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.551954 216.221.215.20.4291 > xxx.xxx.xxx.61.sunrpc: S 1964554565:1964554565(0) win 32120
(DF)
0x0000 4500 003c cfd4 4000 3806 fd15 d8dd d714 E..<..@.8.......
0x0010 xxxx xx3d 10c3 006f 7518 b945 0000 0000 ...=...ou..E....
0x0020 a002 7d78 eb68 0000 0204 05b4 0402 080a ..}x.h..........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........
```

1. Source of Trace:
This trace was actually logged on my network at work between the border router and internal firewalls.

2. Detect was generated by:
This detect was picked up from a Shadow v1.6 sensor. The capture shows a remote host probing a network block attempting to locate hosts that will respond to a connection attemp on tcp port 111 by responding with a SYN|ACK to the requesting host. A breakdown of the applicable fields in the above file can be found later in this document in the Log Files Explained section.

3. Probability the source address was spoofed:
As the source is expecting responses from these connection attempts, I feel the source address in this detect has not been spoofed. According to http://www.securityspace.com/swhois/whois.html the source IP address is registered to the following:

```
Maxlink Communications Inc.
(NETBLK-MAXLINK-BLK1)
1 Yonge Street Suite 2415M5E1E5
CA

Netname: MAXLINK-BLK1
Netblock: 216.221.192.0 -
216.221.223.255
Maintainer: MXLN

Coordinator:
Maxlink Communications Inc.
(ZM104-ARIN) ipadmin@maxlink.net
+1-416-775-5252 (FAX) 416 775-
5501

Domain System inverse mapping
provided by:
```



```
DNS.MAXLINK.NET 216.221.210.5
DNS2.MAXLINK.NET 216.221.205.150
```

Fig. 2-4 SecuritySpace whois Results

4. Description of attack:
Our logs show a remote host probing incrementing IP addresses within a range that is being watched by our Shadow sensor. The destination host IP addresses are incrementing, although somewhat out of order which is probably due the network congestion at the time. Remote Procedure Call or (RPC) is a network technology developed by Sun. It is mostly used in the UNIX environment as way to build network applications. The CVE project has also made attempts to standardize the list of vulnerabilities.

5. Attack Mechanism:
This action from the source IP address is definitely meant as a stimulus as it appears to be a hit-and-miss attempt to find hosts offering rpc services. The remote host is sending a SYN packet to an IP address in the hopes of discovering a host that is offering rpc services. The X-Force team over at Internet Security Systems (ISS) have a list of vulnerabilities with the rpcbind services.

6. Correlations:
The incidents.org website contains many posts from people reporting connection attempts to port 111. DShield.org contains a list of recent activity that shows just how active port 111 probing is today.

7. Evidence of active targeting:
The capture as displayed above represents all activity from that particular source IP address. Since only a few IP addresses have been targeted, it would appear to be a form of active reconnaissance more that anything.

8. Severity:
The severity formula as described at http://www.sans.org/giactc/ID_assignment_guidelines.htm is as follows:
Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
Each severity variable is a value of either 1 (being the Lowest) to 5 (being the Highest).
Criticality: **3**, targeted at users systems.
Lethality: **1**, looking for rpcbind, portmapper which our windows users won't be running.
System Countermeasures: **3**, currently unable to verify if all our users are up to date.
Network Countermeasures: **3**, a borderline restrictive/permissive firewall is in place.
(3 + 1) - (3 + 3) = -2

9. Defensive recommendation:
Definitely there should me measures in place to block access to port 111. We must not believe this is all that is required to be safe from rpc being exploited though. As the paper titled "Information Security Paper: "Rpcbind and Portmapper"" by David P. Reece, 26 February 2000 located at http://www.sans.org/newlook/resources/IDFAQ/blocking.htm states "On Solaris 2.x operating systems, rpcbind listens not only on TCP port 111, and UDP port 111, but also on a

port greater than 32770. This results in a large number of packet filters, which intend to block access to rpcbind/portmapper, being ineffective. Instead of sending requests to TCP or UDP port 111, the attacker simply sends them to a UDP port greater than 32770 on which rpcbind is listening."

10. Multiple choice test question:

```
13:37:02.547826 216.221.215.20.4280 > xxx.xxx.xxx.50.sunrpc: S 1963718756:1963718756(0) win 32120
(DF)
0x0000 4500 003c cfc9 4000 3806 fd2b d8dd d714 E..<..@.8..+....
0x0010 xxxx xx32 10b8 006f 750b f864 0000 0000 ...2...ou..d....
0x0020 a002 7d78 ac6c 0000 0204 05b4 0402 080a ..}x.l..........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.548613 216.221.215.20.4281 > xxx.xxx.xxx.51.sunrpc: S 1969667479:1969667479(0) win 32120
(DF)
0x0000 4500 003c cfca 4000 3806 fd29 d8dd d714 E..<..@.8..)....
0x0010 xxxx xx33 10b9 006f 7566 bd97 0000 0000 ...3...ouf......
0x0020 a002 7d78 e6dc 0000 0204 05b4 0402 080a ..}x............
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........

13:37:02.549469 216.221.215.20.4282 > xxx.xxx.xxx.52.sunrpc: S 1964527505:1964527505(0) win 32120
(DF)
0x0000 4500 003c cfcb 4000 3806 fd27 d8dd d714 E..<..@.8..'....
0x0010 xxxx xx34 10ba 006f 7518 4f91 0000 0000 ...4...ou.O.....
0x0020 a002 7d78 552f 0000 0204 05b4 0402 080a ..}xU/..........
0x0030 02c5 26ff 0000 0000 0103 0300 ..&.........
```

What are the intentions of the source IP address from this detect do you think?
a) Obtain a map of active hosts.
b) Locate hosts offering SunRPC services for possible attack later.
c) Detect Sun computers on the Internet.
d) Answers A & B.

Answer: D – It would actually be a combination of the two answers. By mapping a network this way, it is possible to get a list of active hosts, to learn the configuration a little and find out some important information about a potentially exploitable service running on the network.

Log Files Explained.

The following tables are actually from a previous students practical. The author (Jamie French) did such a good job, I figured (after asking first) that I would use them here.

Snort

Example:	09/12-10:16:59.420396 22:22:22:22:22:22 -> 66:66:66:66:66:66 type:0x800 len:0x1F4 1.1.1.1:23 -> 2.1.1.1:23 TCP TTL:60 TOS:0x0 ID:51966 DF**S***** Seq: 0x3039 Ack: 0x0 Win: 0x0	
Field	Description	Sample Value
Day and Month	The Day and month of the capture.	09/12
Time	Sensors local computer time, logged in HH:mm:ss.milisec format.	10:16:59.420396
Src Ethernet Address	The address in hex (MAC) from originating host.	22:22:22:22:22:22
Seperator		->
Dest Ethernet Address	The address in hex (MAC) of destination host.	66:66:66:66:66:66
Type	Value determined from 10 bits (hardware, proto, size) of ether frame.Value of an IP datagram (0x0800)	0x800
Length	Total length of the IP datagram.	0x1F4
Source IP	The source IP address logged.	1.1.1.1
Source Port	The source port.	23
Seperator		->
Destination IP	The destination IP address logged. If -n switch is not used it will be resolved if possible as seen in the sample value.	2.1.1.1
Destination Port	The destination port.	23
Protocol	The protocol used.	TCP
Time To Live	This is the number of hops remaining before the packet ceases to be routed.	60
Type of Service	Values Min Delay, Max Throughput, Max Reliability, Min Cost, or None. (0x0 = None)	0x0
Fragmentation	This field is either set to on or off. DF means don't fragment. (DF means don't fragment)	DF
ID	This is the identification number.	51966
Flag Set	URG, ACK, PSH, RST, SYN, FIN or any combination	**S*****
Sequence #	Identifies the sequence in which packets are received. They are determined by the host and number 1 up from this initial sequence number for the same connection for every packet sent until termination of that session. (in hex)	0x18CD
Acknowledgement Sequence #	Same as above sequence # except from destination host.	0x303A
Window Size	This is the amount of buffer space that will be alloted for the reconstruction of packets received out of order. It may be negotiated.	0x0
Different switches will produce different output. This example used the -e switch to record ethernet headers too.Recommended reading for further TCP/IP packet breakdown is TCP/IP Illustrated, Volume 1 by Richard Stevens, ISBN 0-201-63346-9		

Fig. 2-5 snort log breakdown.

Tcpdump

Example:	12:33:38.339480 172.16.0.119.50289 > 205.158.26.242.www: S 2864488374:2864488374(0) win 8760 <mss 1460> (DF)	
Field	**Description**	**Sample Value**
Time	Sensors local computer time, logged in HH:mm:ss.milisec format.	06:57:50.734869
Source IP	The source IP address logged.	63.197.4.191
Source Port	The source port.	111
Seperator		>
Destination IP	The destination IP address logged. If -n switch is not used it will be resolved if possible as seen in the sample value.	host1.goodguys.com
Destination Port	The destination port.	111
Flag Set	URG, ACK, PSH, RST, SYN, FIN or any combination	SF
Sequence #	Identifies the sequence in which packets are received. They are determined by the host and number 1 up from this initial sequence number for the same connection for every packet sent until termination of that session.	665720017:6657200 17
Size of Data	This is the number of bytes sent in this packet	(0)
Window Size	This is the amount of buffer space that will be alloted for the reconstruction of packets received out of order.Â It may be negotiated.	1028
Maximum Segment Size	This is the maximum size of data in bytes that may be sent to the host.	1460
Fragmentation	This field is either set to on or off. DF means don't fragment.	DF
The Hex and ASCII are collected and displayed depending on switches used to initiate the caputre. Recommended reading for further TCP/IP packet breakdown is TCP/IP Illustrated, Volume 1 by Richard Stevens, ISBN 0-201-63346-9		

Fig. 2-6 TCPDump output breakdown.

Shadow

Example:	06:57:50.734869 63.197.4.191.111 > host1.goodguys.com.111: SF 665720017:665720017(0) win 1028	
Field	**Description**	**Sample Value**
Time	Sensors local computer time, logged in HH:mm:ss.milisec format.	06:57:50.734869
Source IP	The source IP address logged.	63.197.4.191
Source Port	The source port.	111
Seperator		>
Destination IP	The destination IP address logged. If -n switch is not used it will be resolved if possible as seen in the sample value.	host1.goodguys.com
Destination Port	The destination port.	111
Flag Set	URG, ACK, PSH, RST, SYN, FIN or any combination	SF
Sequence #	Identifies the sequence in which packets are received. They are determined by the host and number 1 up from this initial sequence number for the same connection for every packet sent until termination of that session.	665720017:6657200 17
Size of Data	This is the number of bytes sent in this packet	(0)
Window Size	This is the size of a packet that may be handled during communications by the hosts involved. It may be negotiated.	1028

Fig. 2-7 Shadow output breakdown.

Assignment 3 - "Analyze This" Scenario

Executive Summary

For this assignment, the author was responsible for providing an audit of traffic logged from a Snort Intrusion Detection System sensor using a fairly standard ruleset. The following is a result of the analysis of five consecutive days worth of traffic. Not being given a topology diagram of the network, and armed with no real background of the infrastructure in place will not enable me to perform the most accurate audit.

Overall, the network appears to be in bad shape. There was evidence of compromised hosts, users running such questionable services as chat clients, peer-to-peer file sharing, and even network games. It is recommended that the network owners review their acceptable use policy to analyze the requirements of running such services. The snort rules could use a tweaking to cut down on the number of false positives generated.

I did have some troubles getting all of the necessary files to audit from the University. It was difficult to find the required 3 files for 5 consecutive days. Many files are either missing or seemed to have an incorrect time/date stamp. This could indicate potential problems with the current firewall/IDS archiving or backup procedures in place. The data files I had to work with are as follows:

Alert Files	*Scan Files*	*OOS Files*
alert.010901.clean	scans.010901.clean	oos_Sep.1.2001
alert.010902.clean	scans.010902.clean	oos_Sep.2.2001
alert.010903.clean	scansscans.010903.clean	oos_Sep.3.2001
alert.010904.clean	scansscans.010904.clean	oos_Sep.4.2001
alert.010905.clean	scansscans.010905.clean	oos_Sep.5.2001

Fig. 3-1 Snort files used during the analysis.

The Alert Files are snort generated alerts recorded in *full* data capture mode.
The Scan Files are snort generated alerts recorded in *fast* data capture mode.
The OOS Files are snort generated alerts using a fully decoded output.

Analysis Process

The first order of business I felt was to concatenate all of the files into one large file of each different capture type. This meant that all of the alert data gets put into one large alert file for processing. This resulted in an alert file of over 100 Megabytes of data! I had a scans data file of just over 30 Megabytes and these files really gave me grief when trying to do any parsing with them. My first instinct was to feed the snort log file analysis program SnortSnarf-010821.1 the 100MB data file, but after crunching away for a little while, the program would just die, saying it had ran out of memory on a dual Pentium 533 system with 640MB of RAM.

I then decided I needed more number crunching power so I fed the smaller daily files into a Sun Ultra 10 computer with 512MB of RAM as I knew the RISC processor could better handle the

number manipulation. This went on for a couple of days and then I just tallied up all of the daily totals to reach the numbers presented in this report.
The analysis presented in this report is a result of the information obtained through a variety of attempts to learn some scripting using UNIX commands such as grep, awk, and sed.

Various other students SANS GCIA practicals have been reviewed for correlation, and to try and make sense of what I was looking at. Based on my research, it was discovered that a total of 779963 alerts had been logged over the time period analyzed. During the analysis, I was able to identify a total of 80257 distinct source addresses, a total of 192665 alerts generated from spp portscan events.

There was a total of 500692 scans performed. My research revealed a total of 134 distinct alerts that snort identified. I will only cover a few of them in this paper.

Data Summary

I have compiled a list of the top 10 top talkers seen on the network. The IP Owner field of the table has been resolved using the whois services provided by the following two websites: http://www.securityspace.com/swhois/whois.html and http://www.ripe.net/perl/whois.

The following table shows the Top Talkers overall detected on the network.

IP Address	Count
MY.NET.160.114	70655
MY.NET.218.78	31409
MY.NET.218.50	27329
MY.NET.202.102	21385
MY.NET.234.198	17837
212.199.28.76	15469
MY.NET.233.202	15110
216.162.3.20	14869
MY.NET.201.42	15110
MY.NET.212.150	12955

Fig. 3-2 Top 10 Destination addresses.

The following table displays the top 10 source addresses.

IP Address	IP Owner	Count
211.90.176.59	China United Telecommunications Corporation	21934
MY.NET.14.1	Our Network	16091
MY.NET.16.5	Our Network	14701
211.90.164.34	China United Telecommunications Corporation	11358
211.90.88.43	China United Telecommunications Corporation	9813

61.153.17.244	Ningbo Telecommunication Corporation, China	8898
200.250.65.1	Comite Gestor da Internet no Brasil	7468
211.96.99.59	GD-SZ-UNICOMSZ, China	6976
217.57.15.133	S.C.P. CALCOLATORI SRL, Italy	6677
61.153.17.24	Ningbo Telecommunication Corporation, China	6654

Fig. 3-3 Table showing the Top 10 Talkers by source address.

The following table shows the top 10 destination addresses which resolve to MY.NET network.

IP Address	Count
MY.NET.140.9	24086
MY.NET.100.165	15752
MY.NET.253.114	12251
MY.NET.111.221	6910
MY.NET.1.3	6646
MY.NET.219.154	5895
MY.NET.111.142	5712
MY.NET.1.4	5091
MY.NET.1.5	4296
MY.NET.178.236	3421

Fig. 3-4 Top 10 Destination addresses.

The following table displays the top 5 source addresses found in the OOS logs.

IP Address	IP Owner	Count
151.38.11.166	Infostrada, Italy	71
198.186.202.147	Dandelion Digital, NV	58
128.46.156.155	Purdue University, West Lafayette, IN	20
212.194.4.183	T-Online France - Club Internet	13
151.38.84.194	Infostrada, Italy	11

Fig. 3-5 Top 5 Source addresses form the OOS logs.

The following table displays the top 5 destination addresses found in the OOS logs.

IP Address	Count
MY.NET.280.62	73
MY.NET.253.53	31
MY.NET.253.52	27
MY.NET.99.85	23
MY.NET.218.194	14

Fig. 3-6 Top 5 Destination addresses form the OOS logs.

The following table displays the top 10 destination ports as contained in the alerts files and the possible service for that port as researched at http://www.snort.org/ports.html?port.

Port	Service	Count
80	World Wide Web HTTP	604748
53	Domain Name Server	19588
0		12570
1863	MSN Messenger Protocol	9014
8888	Possibly NewsEDGE server or Sun Answerbook HTTP server	8614
27374	[trojan] SubSeven	5574
1214	KAZAA	4520
3128	[trojan] RingZero or Squid http	3614
6699	Napster Music Sharing Client	1600
6667	Internet Relay Chat	1185

Fig. 3-7 Top 10 Destination ports from the alerts file.

As we can see, the majority of the traffic from the alerts file is related to www services. Given the current proliferation of IIS worms on the Internet, this is not too surprising. We see a lot of domain name service (DNS) activity as well. This could be related to an abundance of BIND vulnerabilities. The rest of the top 10 destination ports is related to such things as Instant Messaging (chat), IRC, peer-to-peer file sharing (Kazaa and Napster), and more worrisome is the Trojan activity.

The following table displays the top 5 destination ports as contained in the OOS files and researched at http://www.snort.org/ports.html?port.

Port	Service	Count
6346	gnutella-svc	148
113	ident tap Authentication Service	59
80	World Wide Web HTTP	44
1214	KAZAA	29
27970	Unknown	11

Fig. 3-8 Top 5 Destination ports from the OOS logs.

Looking at the top ports from the OOS file, we see fairly the same pattern as in the alerts file. A lot of peer-to-peer file sharing, more www related traffic and port 27970 which is unkown to me.

scans file

```
Sep  4 13:29:07 MY.NET.233.42:1100 -> 209.155.226.5:27970 UDP
Sep  4 13:29:11 MY.NET.233.42:1097 -> 212.40.5.36:27970 UDP
Sep  4 13:29:14 MY.NET.233.42:1113 -> 202.12.147.60:27970 UDP
Sep  4 13:29:15 MY.NET.233.42:1114 -> 195.149.21.39:27970 UDP
Sep  4 22:32:44 MY.NET.230.30:1657 -> 198.135.234.35:27970 UDP
Sep  5 11:02:20 151.38.84.194:27960 -> MY.NET.235.94:27970 NOACK *1SFR*** RESERVEDBITS
Sep  5 11:04:29 151.38.84.194:27960 -> MY.NET.235.94:27970 NOACK *1SFR*** RESERVEDBITS
Sep  5 11:07:05 151.38.84.194:27960 -> MY.NET.235.94:27970 NOACK *1SFR*** RESERVEDBITS
Sep  5 11:07:16 151.38.84.194:27960 -> MY.NET.235.94:27970 NOACK *1SFR*** RESERVEDBITS
Sep  5 15:16:04 MY.NET.228.138:1664 -> 194.126.124.66:27970 UDP
Sep  5 15:16:21 MY.NET.228.138:2316 -> 212.137.72.31:27970 UDP
Sep  5 19:20:28 MY.NET.228.138:4888 -> 194.126.124.66:27970 UDP
Sep  5 19:20:49 MY.NET.228.138:1602 -> 212.137.72.31:27970 UDP
```

oos file

```
09/05-11:00:21.548861 151.38.84.194:27960 -> MY.NET.235.94:27970
09/05-11:00:21.592768 151.38.84.194:27960 -> MY.NET.235.94:27970
09/05-11:01:17.971080 151.38.84.194:27960 -> MY.NET.235.94:27970
09/05-11:01:24.356224 151.38.84.194:27960 -> MY.NET.235.94:27970
09/05-11:02:02.236341 151.38.84.194:27960 -> MY.NET.235.94:27970
```

While going through the scans file I noticed outgoing connection attempts from MY.NET.97.191 from source port UDP 6112 to various destination addresses on destination port 6112. After some quick research, the source host is probably looking to exploit a vulnerable dtspcd service on remote hosts. More information on this vulnerability can be found at the following web pages http://www.securiteam.com/unixfocus/2LUQ5QUSAS.html and http://www.cert.org/advisories/CA-1999-11.html.

Also, while perusing the scans logs, I picked up a very large UDP port 137 NETBIOS Name Service scan from an internal host. The following is just a sample of one of the scans they performed.

```
Sep  5 21:31:22 MY.NET.218.78:1854 -> 24.23.230.183:137 UDP
Sep  5 21:31:22 MY.NET.218.78:1854 -> 142.163.15.77:137 UDP
Sep  5 21:31:22 MY.NET.218.78:1854 -> 213.200.165.213:137 UDP
Sep  5 21:31:22 MY.NET.218.78:1854 -> 64.111.37.49:137 UDP
Sep  5 21:31:22 MY.NET.218.78:1854 -> 24.229.11.133:137 UDP
```

Also of interest in the scans file is a snip of the following traffic making one think there might be some ECN compliant hosts out there. RFC 3168 has more information about this possibility.

```
Sep  3 07:37:42 198.186.202.147:53711 -> MY.NET.253.52:113 SYN 21S***** RESERVEDBITS
Sep  3 08:57:03 208.178.176.216:43032 -> MY.NET.182.91:6346 SYN 21S***** RESERVEDBITS
Sep  3 12:38:33 65.9.152.192:0 -> MY.NET.225.182:6346 SYN 21S***** RESERVEDBITS
Sep  3 12:52:24 198.186.202.147:59980 -> MY.NET.253.52:113 SYN 21S***** RESERVEDBITS
Sep  3 13:08:01 198.186.202.147:60426 -> MY.NET.253.53:113 SYN 21S***** RESERVEDBITS
```

One host in particular is very prevelant in the scans file because of their love of multiplayer network games. The MY.NET.212.50 machine is vary active in this log for checking his friends activities on the GameSpy network and has generated loads of UDP traffic with all of the games they have been playing over the network.

Link Graph

The following table is a link graph depicting the traffic detected with a source or destination of host MY.NET.235.14. The regular lines depict a singular packet seen traversing the wire. The lightly dotted line from source 211.90.176.59 on port 30419 was 3 packets sent while the heavy dashed line from MY.NET.235.14 on port 6346 to host 149.2.31.6 represents a data count of 2178. This shows us that the host MY.NET.235.14 has been the recipient of a couple of www probes that he has not responded to, but that this host does seem to be actively using the GNUTella peer-to-peer file sharing program.

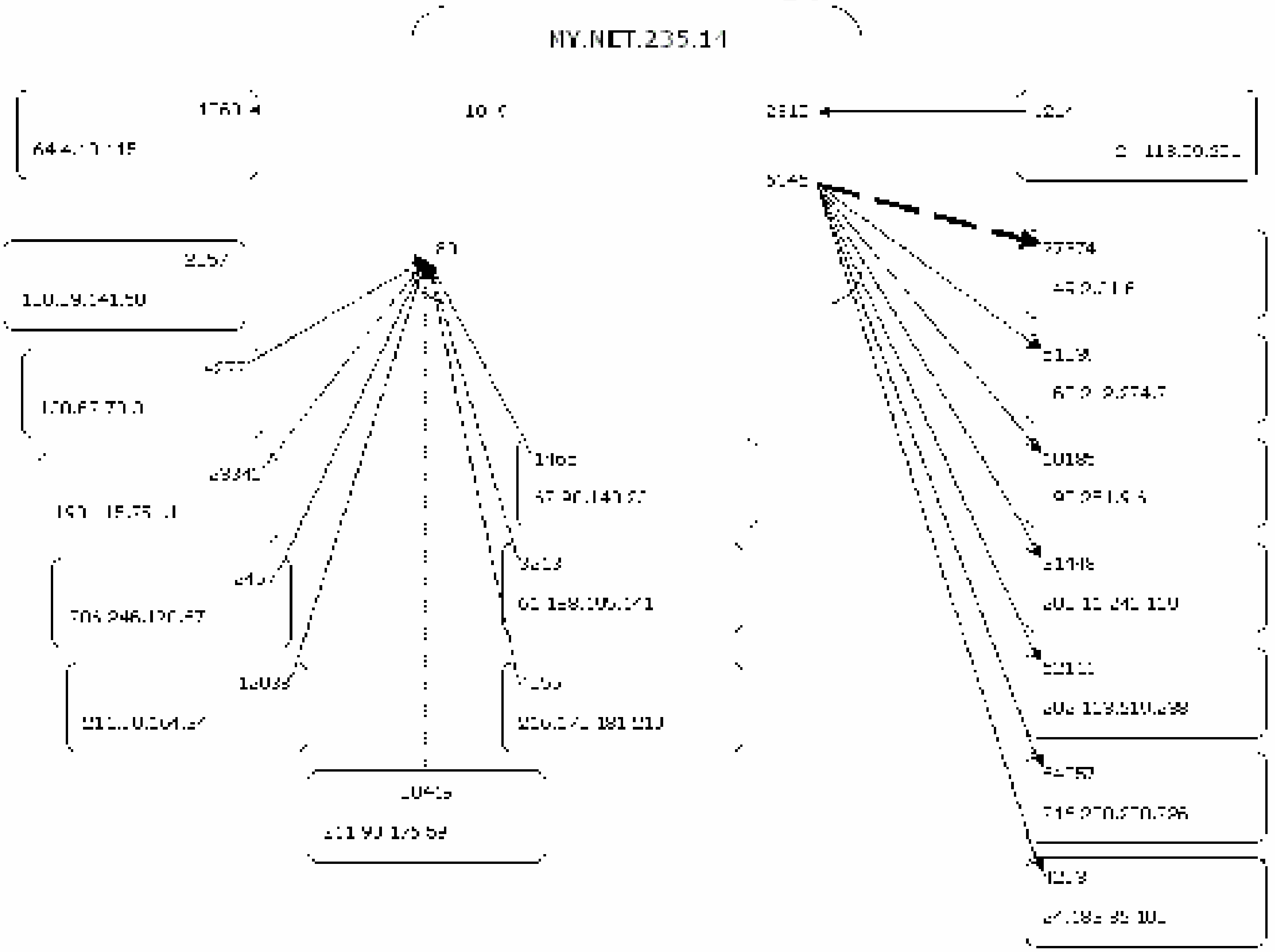


Fig. 3-9 Link Graph representing data seen with a source and destination of MY.NET.235.14.

Top Snort Alerts (Overall Total)

The following table highlights the most significant amount of network traffic seen (incoming and outgoing from the MY.NET network) as detected by snort.

Alerts	Count
WEB-MISC Attempt to execute cmd	305468
IDS552/web-iis_IIS ISAPI Overflow ida nosize	268112
ICMP Destination Unreachable (Communication Administratively Prohibited)	32311
MISC Large UDP Packet	20678
MISC Traceroute	20453
MISC source port 53 to <1024	19590
CS WEBSERVER – external web traffic	16079
INFO MSN IM Chat data	14853
WEB-MISC prefix-get //	12258
ICMP Echo Request Nmap or HPING2	10805

Fig. 3-10 Representation of all traffic.

1)

Alert	Alert Count
WEB-MISC Attempt to execute cmd	305468

Traffic Sample

```
09/01-00:00:03.329644  [**] WEB-MISC Attempt to execute cmd [**] 211.96.99.59:13049 -> MY.NET.224.234:80
09/01-00:00:06.216134  [**] WEB-MISC Attempt to execute cmd [**] 195.23.79.174:33936 -> MY.NET.9.171:80
09/01-00:00:06.273124  [**] WEB-MISC Attempt to execute cmd [**] 210.250.111.50:3430 -> MY.NET.142.217:80
09/01-00:00:08.968484  [**] WEB-MISC Attempt to execute cmd [**] 211.90.223.220:13528 -> MY.NET.181.208:80
09/01-00:00:10.776361  [**] WEB-MISC Attempt to execute cmd [**] 200.26.105.130:2009 -> MY.NET.191.197:80
```

Top 5 Source IP Addresses

Count	IP Address
11522	211.90.176.59
5887	211.90.164.34
5148	211.90.88.43
3837	200.25.65.1
3624	217.57.15.133

Top 5 Destination Addresses

Count	IP Address
89	MY.NET.152.219
62	MY.NET.94.69
60	MY.NET.12.170
59	MY.NET.106.6
57	MY.NET.183.240

Top Source Ports

Count	Port
196	1025
112	3121
110	3262
110	4014
108	3198

Top Destination Ports

Count	Port
305468	80

This detect may have been caused by a snort rule such as:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access"; flags: A+;
content:"cmd.exe"; nocase; classtype:attempted-user; sid:1002; rev:1;)
```

Conclusion

On a webserver, any remote access to the command prompt could prove to be fatal as it would give outsiders the ability to execute commands on your web server. As discussed in the article by Russ Cooper called "10 Steps To Better IIS Security" which he wrote in August 2001 (http://www.infosecuritymag.com/articles/september01/features_IIS_security.shtml) he makes a good comment about the existence of cmd.exe on a server as found in paragraph 2 (Don't Let Hackers Exploit DOS).

"So, if CMD.EXE isn't where it's expected to be or doesn't exist at all, the overwhelming majority of exploits that rely on it are going to fail. In such cases, an attacker will likely move on to another target.

On NT 4.0 systems, CMD.EXE can be deleted, renamed or moved to another directory. Also, remove the COMSPEC environment variable, since it points directly to the location of CMD. EXE. If you renamed or moved CMD.EXE, you don't want to re-point COMSPEC, which would help an attacker. If you delete CMD.EXE, COMSPEC has nothing to point to.

On Windows 2000 systems, removing CMD.EXE is a little more difficult because of Windows File Protection (WFP). CMD.EXE will automatically be replaced by WFP if you delete, rename or move it. However, you can assign explicit access permissions to members of the Administrators group. You should explicitly deny all access to the SYSTEM and IUSR/IWAM accounts (see http://www.infosecuritymag.com/articles/september01/features IIS security.shtml#8), as well as any other accounts that you use in your Web site."

This detect could potentially be related to the Dos.Storm.Worm, although there are many worms circulating around the Internet currently, targeting unpatched Microsoft Internet Information Servers (IIS). For more information on this specific worm, please see the following website http://www.incidents.org/react/dosstormworm.php.

By far, the biggest offender was 211.90.176.59, generating 11522 alerts for this signature alone in a 5 day period! The registration information follows:

```
inetnum:    211.90.0.0 - 211.91.255.255
netname:    UNICOM
descr:      China United
Telecommunications Corporation
country:    CN
admin-c:    XL31-AP
tech-c:     XL31-AP
mnt-by:     MAINT-CNNIC-AP
changed:    xiaqing@cnnic.net.cn
20000414
source:     APNIC

person:     XiaoMing Li
address:    6F Office Tower 3,
Henderson Centre, Beijing China
country:    CN
phone:      +86-10-65181800-291
fax-no:     +86-10-65181800-777
e-mail:     lxmlxm@public3.bta.net.cn
nic-hdl:    XL31-AP
mnt-by:     MAINT-CNNIC-AP
changed:    wangch@cnnic.net.cn
20000331
source:     APNIC
```

Fig. 3-11 Largest external source of WEB-MISC Attemp to execute cmd.

Recommendation

Make sure the program cmd.exe is not accessible to be executed by external users.

2)

Alert	Alert Count
IDS552/web-iis_IIS ISAPI Overflow ida nosize	268112

Traffic Sample

```
09/01-00:00:10.587983  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 200.26.105.130:2009 -> MY.NET.191.197:80
09/01-00:00:13.551790  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 200.26.105.130:2009 -> MY.NET.191.197:80
09/01-00:00:15.187039  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 203.74.136.149:2344 -> MY.NET.54.60:80
09/01-00:00:18.466898  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 213.167.132.65:63975 -> MY.NET.110.161:80
09/01-00:00:19.001142  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 211.96.99.59:13456 -> MY.NET.202.174:80
```

Top 5 Source IP Addresses

Count	IP Address
10412	211.90.176.59
5471	211.90.164.34
4663	211.90.88.43
3631	200.250.65.1
3379	211.96.99.59

Top 5 Destination Addresses

Count	IP Address
68	MY.NET.100.220
65	MY.NET.152.219
62	MY.NET.12.170
51	MY.NET.179.223
50	MY.NET.142.65

Top Source Ports

Count	Port
169	1025
98	3449
95	3465
95	3837
3656	3198

Top Destination Ports

Count	Port
268112	80

This detect may have been caused by a snort rule such as:

```
alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS552/web-iis_IIS ISAPI Overflow ida"; dsize: >239;
flags: A+; uricontent: ".ida?"; classtype: system-or-info-attempt; reference: arachnids,552;)
```

Conclusion

This alert will be generated when a remote user attempts to exploit the IIS Index Server ISAPI vulnerability as outlined by Microsoft in their security bulletin MS01-033. This vulnerability is the result of an unchecked buffer in the ISAPI extensions with the potential for a remote buffer overflow and compromise of the host. The use of an IDS to monitor such malicious requests is recommended as well as keeping abreast of the latest patches for your systems.

The biggest offender was 211.90.176.59, generating 10412 alerts for this signature over 5 days. The registration information follows:

```
inetnum:    211.90.0.0 - 211.91.255.255
netname:    UNICOM
descr:      China United
Telecommunications Corporation
country:    CN
admin-c:    XL31-AP
tech-c:     XL31-AP
mnt-by:     MAINT-CNNIC-AP
changed:    xiaqing@cnnic.net.cn
20000414
source:     APNIC

person:     XiaoMing Li
address:    6F Office Tower 3,
Henderson Centre, Beijing China
country:    CN
phone:      +86-10-65181800-291
fax-no:     +86-10-65181800-777
e-mail:     lxmlxm@public3.bta.net.cn
nic-hdl:    XL31-AP
mnt-by:     MAINT-CNNIC-AP
changed:    wangch@cnnic.net.cn
20000331
source:     APNIC
```

Fig. 3-12 Largest external source of IDS552/web-iis_IIS ISAPI Overflow ida.

Recommendation

If this is a default installation of Windows NT 4, there is nothing to worry about unless you have installed the Windows NT 4 Option Pack. If you are running Windows 2000 Server, a default installation is vulnerable. If you are running a default installation of Windows 2000 Professional you are not vulnerable unless you install IIS5.0 after the fact. If you are running a version of Windows XP prior to Release Candidate 1 (which would be a very bad thing to be still doing) you are vulnerable. If you fall under any of these categories which makes your system vulnerable, you must download the patch from Microsoft and patch your systems.

3)

Alert	**Alert Count**
ICMP Destination Unreachable (Communication Administratively prohibited)	32311

Traffic Sample

```
09/01-00:00:19.471910  [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 ->
MY.NET.201.58
09/01-00:00:53.760042  [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.14.1 ->
MY.NET.182.250
09/01-00:00:56.097355  [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 ->
MY.NET.5.74
09/01-00:01:31.138783  [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 ->
MY.NET.5.79
09/01-00:02:27.968899  [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 ->
MY.NET.202.238
```

Top 5 Source IP Addresses

Count	IP Address
16091	MY.NET.14.1
14700	MY.NET.16.5
654	192.80.53.46
250	152.61.1.10
248	192.5.89.62

Top 5 Destination Addresses

Count	IP Address
32311	MY.NET.16.5

Top Source Ports

Count	Port

Top Destination Ports

Count	Port

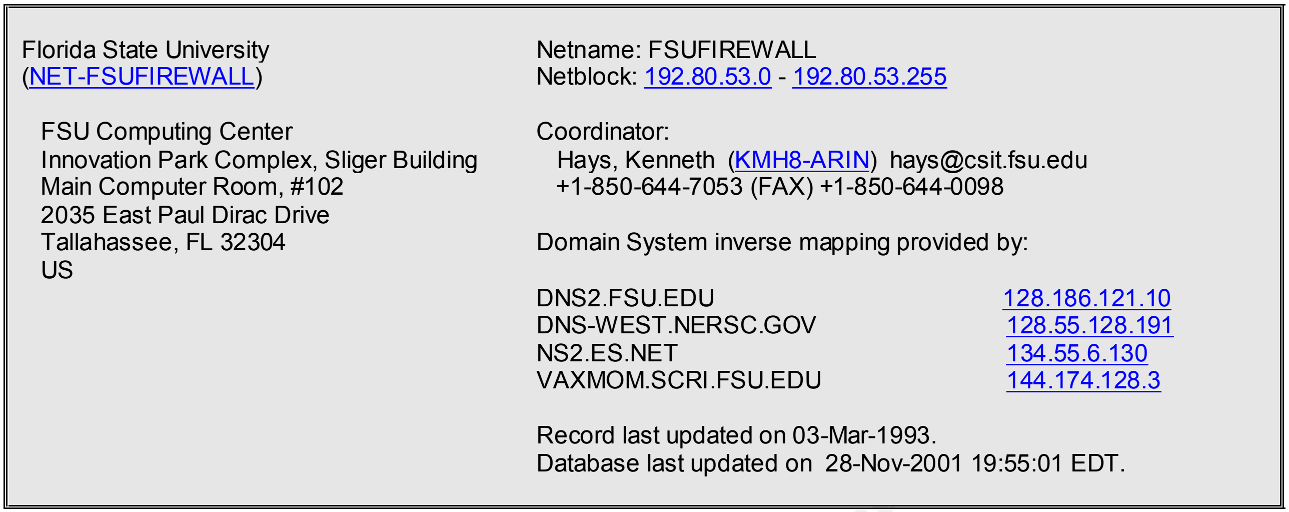
This detect may have been caused by a snort rule such as:

alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Communication Administratively Prohibited)"; itype: 3; icode: 13; sid:485; rev:1;)

Conclusion

This message is generated if a router cannot forward a packet due to administrative filtering at the router. These types of messages are sent back to the originator and can be used for reconnaissance of a network as they provide valuable information as to the network configuration. There is the possiblitiy of configuring the routers to not send out these ICMP error messages for this situation, or block this type of ICMP traffic from leaving this site.

The biggest external offender was 192.80.53.46, generating 654 alerts for this signature over 5 days. The registration information follows:



Florida State University
(NET-FSUFIREWALL)

FSU Computing Center
Innovation Park Complex, Sliger Building
Main Computer Room, #102
2035 East Paul Dirac Drive
Tallahassee, FL 32304
US

Netname: FSUFIREWALL
Netblock: 192.80.53.0 - 192.80.53.255

Coordinator:
Hays, Kenneth (KMH8-ARIN) hays@csit.fsu.edu
+1-850-644-7053 (FAX) +1-850-644-0098

Domain System inverse mapping provided by:

DNS2.FSU.EDU	128.186.121.10
DNS-WEST.NERSC.GOV	128.55.128.191
NS2.ES.NET	134.55.6.130
VAXMOM.SCRI.FSU.EDU	144.174.128.3

Record last updated on 03-Mar-1993.
Database last updated on 28-Nov-2001 19:55:01 EDT.

Fig. 3-13 Largest external source of ICMP Destination Unreachable (Communication Administratively Prohibited).

Recommendation

As shown in the Cisco article "Configure IP Services" found at the URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdip.htm#xtocid2722854 you can enter the command "no ip unreachables" in your configuration file.

4)

Alert	**Alert Count**
MISC Large UDP Packet	20678

Traffic Sample

```
09/01-11:48:41.900824  [**] MISC Large UDP Packet [**] 61.153.19.95:0 -> MY.NET.153.113:0
09/01-11:48:44.590463  [**] MISC Large UDP Packet [**] 61.153.19.95:2506 -> MY.NET.153.113:2767
09/01-11:48:55.687650  [**] MISC Large UDP Packet [**] 61.153.19.95:0 -> MY.NET.153.113:0
09/01-11:48:55.786480  [**] MISC Large UDP Packet [**] 61.153.19.95:2506 -> MY.NET.153.113:2767
09/02-12:58:37.307092  [**] MISC Large UDP Packet [**] 64.132.43.122:0 -> MY.NET.104.209:0
```

Top 5 Source IP Addresses

Count	IP Address
8898	61.153.17.244
6654	61.153.17.24
1215	61.153.19.95
651	64.157.10.118
636	61.153.17.210

Top 5 Destination Addresses

Count	IP Address
6870	MY.NET.111.221
5700	MY.NET.111.142
2982	MY.NET.144.51
1248	MY.NET.153.110
651	MY.NET.140.136

Top Source Ports

Count	Port
10215	0
2172	3563
906	1790
820	1631
627	3439

Top Destination Ports

Count	Port
10202	0
2172	1548
906	1680
820	2643
627	2889

This detect may have been caused by a snort rule such as:

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large UDP Packet"; dsize: >4000;
reference:arachnids,247; classtype:bad-unknown; sid:521; rev:1;)
```

Conclusion

It is hard to determine from the provided detects, exactly what is going on here. The availability of logs from a higher fidelity sensor might help shed some light on this traffic.

The biggest external offender was 61.153.17.244, generating 8898 alerts for this signature over 5 days. The registration information follows:

```
inetnum:    61.153.17.0 - 61.153.17.255
netname:     NINGBO-ZHILAN-NET
descr:      NINGBO
TELECOMMUNICATION CORPORATION
,ZHILAN APPLICATION SERVICE
PROVIDER
descr:      Ningbo, Zhejiang Province
country:     CN
admin-c:     CZ61-AP
tech-c:     CZ61-AP
mnt-by:     MAINT-CHINANET-ZJ
changed:     master@dcb.hz.zj.cn
20010512
source:     APNIC

person:     CHINANET ZJMASTER
address:     no 378,yan an road,hangzhou,zhejiang
country:     CN
phone:      +86-571-7015441
fax-no:     +86-571-7027816
e-mail:     master@dcb.hz.zj.cn
nic-hdl:     CZ61-AP
mnt-by:     MAINT-CHINANET-ZJ
changed:     master@dcb.hz.zj.cn 20001219
source:     APNIC
```

Fig. 3-14   Largest external source of MISC Large UDP Packet.

Recommendation

More analysis of this traffic should be peformed to determine what is going on. This traffic could be generated from such UDP transport programs as media streaming, instant messaging chat, or even some network games.

5)

Alert

MISC Traceroute

Alert Count

20453

Traffic Sample

```
09/02-12:58:58.549825  [**] MISC traceroute [**] 132.198.101.254:39495 -> MY.NET.140.9:33463
09/02-12:59:57.775533  [**] MISC traceroute [**] 138.26.220.46:35862 -> MY.NET.140.9:33456
09/02-13:00:07.976660  [**] MISC traceroute [**] 206.220.240.230:33890 -> MY.NET.140.9:33460
09/02-13:00:18.522281  [**] MISC traceroute [**] 129.119.224.250:61399 -> MY.NET.140.9:33470
09/02-13:00:38.460720  [**] MISC traceroute [**] 137.78.21.22:45212 -> MY.NET.140.9:33483
```

Top 5 Source IP Addresses

Count	IP Address
410	129.79.20.239
401	128.114.129.62
396	199.249.169.82
392	128.138.213.35
391	129.89.70.20

Top 5 Destination Addresses

Count	IP Address
20386	MY.NET.140.9
9	MY.NET.150.220
7	MY.NET.1.8
6	MY.NET.150.133
4	MY.NET.204.18

Top Source Ports

Count	Port
15	53
9	61868
8	60137
8	48890
7	61404

Top Destination Ports

Count	Port
1119	33461
1119	33459
1097	33460
1039	33462
1018	33463

This detect may have been caused by a snort rule such as:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute ";ttl:1;itype:8;
reference:arachnids,118; classtype:attempted-recon; sid:385; rev:1;)
```

Conclusion

Traceroute is used to discover the path from the source to a destination, essentially providing a map. Now it is possible to block this at the border, but the problem is that the Windows version of tracert uses the ICMP echo request while UNIX traceroute uses UDP so this would get through if you only thought to block the ICMP. Care must be taken when blocking ICMP traffic at the border as certain error messages are needed by internal hosts when communicating with the outside.

The biggest external offender was 129.79.20.239, generating 410 alerts for this signature over 5 days. The registration information follows:

```
Indiana University (NET-INDIANA-NET)          Coordinator:
  2711 E 10th St                              Indiana University Computing Services  (IUD-ORG-ARIN)
  Bloomington, IN 47408                       dns-admin@indiana.edu
  US                                              812 855-9255

  Netname: INDIANA-NET                          Domain System inverse mapping provided by:
  Netblock: 129.79.0.0 - 129.79.255.255
                                                NS.INDIANA.EDU                 129.79.1.1
                                                NS2.INDIANA.EDU                129.79.5.100
                                                DNS1.CSO.UIUC.EDU              128.174.5.103

                                                Record last updated on 03-Mar-1999.
                                                Database last updated on  28-Nov-2001 19:55:01 EDT.
```

Fig. 3-15 Largest external source of MISC Traceroute.

Recommendation

This is not necessarily “a vulnerability” but is used during the recon phase which could be used to map out a network for a possible later attack. In the big scheme of things, it is hard to stop reconnaissance activity and has to be looked at in the context of kind of information is the “attacker” getting. If you can minimize the information that is available to malicious users, that is a good step to securing your assets.

6)

Alert	**Alert Count**
MISC source port 53 to <1024	19590

Traffic Sample

```
09/02-13:00:55.560416  [**] MISC source port 53 to <1024 [**] 216.136.227.241:53 -> MY.NET.1.5:53
09/02-13:01:03.003354  [**] MISC source port 53 to <1024 [**] 204.134.124.2:53 -> MY.NET.1.4:53
09/02-13:01:09.414415  [**] MISC source port 53 to <1024 [**] 207.217.77.82:53 -> MY.NET.1.3:53
09/02-13:01:11.395730  [**] MISC source port 53 to <1024 [**] 24.69.255.213:53 -> MY.NET.1.3:53
09/02-13:01:38.085218  [**] MISC source port 53 to <1024 [**] 208.242.128.11:53 -> MY.NET.1.4:53
```

Top 5 Source IP Addresses

Count	IP Address
2420	134.93.19.12
922	53.122.1.10
328	207.171.178.5
310	159.230.4.2
289	192.115.189.10

Top 5 Destination Addresses

Count	IP Address
6644	MY.NET.1.3
5091	MY.NET.1.4
4296	MY.NET.1.5
2421	MY.NET.130.122
289	MY.NET.88.88

Top Source Ports

Count	Port
19590	53

Top Destination Ports

Count	Port
19578	53
11	1024
1	777

This detect may have been caused by a snort rule such as:

```
alert tcp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to <1024"; flags:S; reference:arachnids,07; classtype:bad-unknown; sid:504; rev:2;)
```

in combination with:

```
alert udp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to <1024"; classtype:bad-unknown; sid:515; rev:2;)
```

Conclusion

This alert can be generated when a connection is made to a destination “privileged port” (below 1024) on a machine from a source port of 53 which is commonly used for domain name queries (DNS). The concern with this alarm is that these are TCP connections we are dealing with. To cut down on the false positives this type of alarm can generate, it is suggested that the snort rule be modified to be triggered only for the IP addresses uses by the actual networks DNS servers and not just matching any internal machine. Chris Brenton touches lightly on the topic of some legitimate uses for TCP port 53 in the article entitled “Lion Worm Version 0.1” dated March 26, 2001 and available at the http://www.incidents.org/react/lion_protection.php. As pointed out in the article “Securing Your Internet Access Router” by Richard Langley (January 23, 2001) and found at http://www.sans.org/infosecFAQ/firewall/router.htm that one of the services that should be filtered is TCP port 53 “DNS Zone Transfers except from external secondary DNS servers” which must be carefully configured to avoid false positives or a misconfigured access route.

The biggest external offender was 134.93.19.12, generating 2420 alerts for this signature over 5 days. The registration information follows:

```
inetnum:    134.93.0.0 - 134.93.255.255          route:     134.93.0.0/16
netname:    UNI-MAINZ-B                          descr:     UNI-MAINZ-B
descr:      Johannes Gutenberg-Universitaet Mainz origin:   AS2857
country:    DE                                   mnt-by:    AS2857-MNT
admin-c:    FN                                   changed:   weiss@uni-mainz.de 20001212
tech-c:     FN                                   source:    RIPE
rev-srv:    ns-extern.zdv.Uni-Mainz.DE
rev-srv:    DENEB.DFN.DE                         person:    Friedrich H. Neugebauer
rev-srv:    WS-WAS.WIN-IP.DFN.DE                 address:   Johannes Gutenberg-Universitaet
status:     ASSIGNED PI                          address:   Zentrum fuer Datenverarbeitung
mnt-by:     AS2857-MNT                           address:   Saarstrasse 21
mnt-by:     DFN-NTFY                             address:   D-55099 Mainz
```

```
changed:    nipper@xlink.net 19940717                  address:    Germany
changed:    Oliver.Doll@Germany.EU.net 19940727        phone:      +49 6131 39 26357
changed:    rv@Informatik.Uni-Dortmund.DE 19940805     fax-no:     +49 6131 39 26407
changed:    knocke@nic.de 19941127                     e-mail:     neugebauer@uni-mainz.de
changed:    weiss@uni-mainz.de 19941230                nic-hdl:    FN
changed:    weiss@uni-mainz.de 19951031                mnt-by:     AS2857-MNT
changed:    weiss@uni-mainz.de 19960112                changed:    neugebauer@vzdmzc.zdv.uni-mainz.de
changed:    schweikh@noc 19990819                      19931223
changed:    weiss@uni-mainz.de 20000124                changed:    knocke@nic.de 19941127
source:     RIPE                                       changed:    weiss@uni-mainz.de 19941230
                                                       changed:    weiss@uni-mainz.de 20000124
                                                       source:     RIPE
```

Fig. 3-16 Largest external source of MISC source port 53 to <1024.

Recommendation

Tighten the snort rule to watch only the networks DNS servers and not have a generalized rule that will alert when this rule matches any host. This will significantly cut down on false positives coming up.

7)

Alert	Alert Count
CS WEBSERVER – external web traffic	16079

Traffic Sample

```
09/02-13:04:00.272591  [**] CS WEBSERVER - external web traffic [**] 62.252.64.5:64586 -> MY.NET.100.165:80
09/02-13:04:02.581471  [**] CS WEBSERVER - external web traffic [**] 199.172.149.188:62545 -> MY.NET.100.165:80
09/02-13:04:50.307388  [**] CS WEBSERVER - external web traffic [**] 216.239.46.151:29698 -> MY.NET.100.165:80
09/02-13:05:41.628536  [**] CS WEBSERVER - external web traffic [**] 62.252.64.5:49255 -> MY.NET.100.165:80
09/02-13:05:43.712802  [**] CS WEBSERVER - external web traffic [**] 62.252.64.5:49592 -> MY.NET.100.165:80
```

Top 5 Source IP Addresses

Count	IP Address
554	200.199.99.143
426	206.156.10.102
246	216.239.46.26
244	66.7.131.154
242	204.123.28.40

Top 5 Destination Addresses

Count	IP Address
15571	MY.NET.100.165

Top Source Ports

Count	Port
15	1115
11	1066
11	1241
11	1138
11	1207

Top Destination Ports

Count	Port
15458	80

This detect may have been caused by a snort rule such as:

alert tcp any any -> $HOME_NET 80 (msg:"CS WEBSERVER – external web traffic";)

Conclusion

I am unfamiliar with this snort alarm. It appears to be a custom rule triggering on outside access to an internal web server.

The biggest external offender was 200.199.99.143, generating 554 alerts for this signature over 5 days. The registration information follows:

```
Comite Gestor da Internet no Brasil                    Coordinator:
(NETBLK-BRAZIL-BLK2)                                    Registro.br  (NF-ORG-ARIN)  blkadm@nic.br
                                                          +55 19 9119-0304
  R. Pio XI, 1500
  Sao Paulo, SP 05468-901                              Domain System inverse mapping provided by:
  BR
                                                       NS.DNS.BR                  143.108.23.2
  Netname: BRAZIL-BLK2                                 NS1.DNS.BR                 200.255.253.234
  Netblock: 200.128.0.0 - 200.255.255.255               NS2.DNS.BR                200.19.119.99
  Maintainer: BR
```

Fig. 3-17 Largest external source of CS WEBSERVER – external web traffic.

Recommendation

This activity should be continued to be monitored and if deemed to be malicious or undesired, it should be blocked at the firewall.

8)

Alert	Alert Count
INFO MSN IM Chat data	14853

Traffic Sample

```
09/02-13:05:43.814173  [**] INFO MSN IM Chat data [**] MY.NET.98.193:2181 -> 64.4.13.162:1863
09/02-13:05:48.743567  [**] INFO MSN IM Chat data [**] MY.NET.97.190:1595 -> 64.4.13.128:1863
09/02-13:05:53.754072  [**] INFO MSN IM Chat data [**] MY.NET.97.190:1595 -> 64.4.13.128:1863
09/02-13:05:54.003573  [**] INFO MSN IM Chat data [**] MY.NET.53.31:2532 -> 64.4.13.168:1863
09/02-13:06:09.909293  [**] INFO MSN IM Chat data [**] MY.NET.53.51:3090 -> 64.4.13.136:1863
```

Top 5 Source IP Addresses

Count	IP Address
375	64.4.13.161
312	64.4.13.132
253	64.4.13.121
245	64.4.13.197
229	64.4.13.137

Top 5 Destination Addresses

Count	IP Address
442	64.4.13.164
402	64.4.13.115
394	64.4.13.121
374	64.4.13.117
318	64.4.13.139

Top Source Ports

Count	Port
5840	1863
163	1675
108	2906
107	2951
103	2577

Top Destination Ports

Count	Port
9013	1863
88	2577
75	2685
69	1038
68	1492

This detect may have been caused by a snort rule such as:

```
alert tcp $HOME_NET any -> $any 1863 (msg:"INFO MSN IM Chat data";flags: A+;
content:"|746578742F706C61696E|"; depth:100; classtype:not-suspicious; sid:540; rev:1;)
```

Conclusion

This snort detect relates to the Microsoft MSN Instant Messenger software.

The biggest external offender was 64.4.13.161, generating 375 alerts for this signature over 5 days. The registration information follows:

```
MS Hotmail (NETBLK-HOTMAIL)                                Coordinator:
   1065 La Avenida                                            Myers, Michael  (MM520-ARIN)  icon@HOTMAIL.COM
   Mountain View, CA 94043                                    650-693-7072
   US
                                                             Domain System inverse mapping provided by:
   Netname: HOTMAIL
   Netblock: 64.4.0.0 - 64.4.63.255                          NS1.HOTMAIL.COM                216.200.206.140
                                                             NS3.HOTMAIL.COM                209.185.130.68
```

Fig. 3-18   Largest external source of INFO MSN IM Chata.

Recommendation

While this does not in itself constitute ”a vulnerability” its use on the network might be a violation of the networks policy. The presense of this software on the network should be questioned and if found to be acceptable, be monitored for abuse.

9)

Alert	Alert Count
WEB-MISC prefix-get //	12258

Traffic Sample

```
09/02-13:09:55.006709  [**] WEB-MISC prefix-get // [**] 208.240.209.184:1091 -> MY.NET.253.114:80
09/02-13:09:56.436181  [**] WEB-MISC prefix-get // [**] 208.240.209.184:1093 -> MY.NET.253.114:80
09/03-07:54:23.794709  [**] WEB-MISC prefix-get // [**] 211.155.162.26:2023 -> MY.NET.253.114:80
09/03-08:14:16.911075  [**] WEB-MISC prefix-get // [**] 210.214.45.81:51480 -> MY.NET.253.114:80
09/03-08:27:45.772968  [**] WEB-MISC prefix-get // [**] 165.247.104.240:1280 -> MY.NET.253.114:80
```

Top 5 Source IP Addresses

Count	IP Address
181	24.3.0.33
148	141.157.92.101
147	24.180.140.140
132	24.184.104.136
117	64.20.68.8

Top 5 Destination Addresses

Count	IP Address
12214	MY.NET.253.114
33	MY.NET.99.85
9	MY.NET.253.115
1	MY.NET.60.14
1	MY.NET.253.18

Top Source Ports

Count	Port
15	1190
15	1278
15	1295
14	1294
14	1331

Top Destination Ports

Count	Port
12258	80

This detect may have been caused by a snort rule such as:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC prefix-get //";flags: A+;
content:"get //"; nocase; classtype:attempted-recon; sid:1114; rev:1;)
```

Conclusion

The destination host is being probed from remote hosts with the hopes of obtaining server specific information.

The biggest external offender was 24.3.0.33, generating 181 alerts for this signature over 5 days. The registration information follows:

```
@Home Network (NETBLK-ATHOME)                ATHOME                          24.0.0.0 - 24.23.255.255
@Home Network (NETBLK-MD-COMCAST-TWSN-1) MD-COMCAST-TWSN-1                   24.3.0.0 - 24.3.15.25
```

Fig. 3-19 Largest external source of WEB-MISC prefix-get //.

Recommendation

If the destination machine in question is not an actual web server that is serving hosts outside of the internal network, access to it should be blocked at the firewall.

10)

Alert	Alert Count
ICMP Echo Request Nmap or HPING2	10805

Traffic Sample

```
09/03-08:51:59.745896  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.137.7 -> 216.158.50.240
09/03-09:44:38.987754  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.97.203 -> 206.251.6.192
09/03-09:51:08.317032  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.97.193 -> 24.234.76.207
09/03-09:51:08.527078  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.97.193 -> 213.89.200.239
09/03-09:51:08.669342  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.97.193 -> 24.80.119.40
```

Top 5 Source IP Addresses

Count	IP Address
5302	MY.NET.226.18
3393	MY.NET.208.82
355	MY.NET.201.78
231	MY.NET.97.181
110	MY.NET.98.183

Top Destination Address

Count	IP Address
2696	206.79.171.51
2608	204.71.200.75
1262	128.197.213.103
1157	168.122.171.197
1087	130.91.233.199

Top Source Ports

Count	Port

Top Destination Ports

Count	Port

This detect may have been caused by a snort rule such as:

alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS162/scan_ping-nmap-icmp"; dsize: 0; itype: 8; classtype: info-attempt; reference: arachnids,162;)

Conclusion

The use of tools like HPING2 or NMAP, which are security related tools were noticed to be very prevelant coming from internal hosts on this network. These tools are capable of such things as sending TCP, UDP, ICMP or even raw IP protocols. They are generally used for creating custom packets for operating system fingerprinting with the end result of discovering vulnerabilities on remote hosts. With such a large number of probes coming from MY.NET.226.18 and MY.NET.208.82, these machines should be taken a look at to make sure they have not been compromised or are being used for malicious intent by attempting to discover potential new targets.

The destination receiving the most traffic was 206.79.171.51, receiving 2696 alerts for this signature over 5 days. The registration information follows:

```
Exodus Communications (NETBLK-ECI-2)        Domain System inverse mapping provided by:
  948 Benecia Ave
  Sunnyvale, CA 94086                         DNS01.EXODUS.NET                 209.1.222.244
  US                                          DNS02.EXODUS.NET                 209.1.222.245
                                              DNS03.EXODUS.NET                 209.1.222.246
  Netname: ECI-2                              DNS04.EXODUS.NET                 209.1.222.247
  Netblock: 206.79.0.0 - 206.79.255.255
  Maintainer: ECI                             * Rwhois reassignment information for this block is
                                            available at:
Coordinator:                                  * rwhois.exodus.net 4321
    Center, Network Control  (NOC44-ARIN)
CompServ@Exodus.net                           ADDRESSES WITHIN THIS BLOCK ARE NON-
    (888) 239-6387 (FAX) (888) 239-6387     PORTABLE

                                              Record last updated on 03-Sep-1998.
                                              Database last updated on  28-Nov-2001 19:55:01 EDT.
```

Fig. 3-20 Largest external source of ICMP Echo Request NMAP or HPING2.

Recommendation

Such activity is not usually authorized for internal users and should be investigated.

Top Snort Alerts (Incoming to MY.NET)

The following table identifies traffic specifically destined for the network MY.NET.

Alert	Count
WEB-MISC Attempt to execute cmd	305468
IDS552/web-iis_IIS ISAPI Overflow ida nosize	268112
MISC Large UDP Packet	20678
MISC traceroute	20453
MISC source port 53 to <1024	19590
CS WEBSERVER - external web traffic	15458

WEB-MISC prefix-get //	12258
INFO MSN IM Chat data	5841
Watchlist 000220 IL-ISDNNET-990517	5315
High port 65535 tcp - possible Red Worm – traffic	3650

Fig. 3-20 Traffic coming into network MY.NET.

Alerts not covered in the Top Snort Alerts List

Alert	**Alert Count**
Watchlist 000220 IL-ISDNNET-990517	5315

Traffic Sample

```
09/03-09:51:32.331123  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.82.106:2163 -> MY.NET.222.74:4349
09/03-09:51:32.333787  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.82.106:2163 -> MY.NET.222.74:4349
09/03-10:14:31.041425  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.65.3:1981 -> MY.NET.221.138:1214
09/03-10:14:31.097181  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.65.3:1981 -> MY.NET.221.138:1214
09/03-10:14:31.421728  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.65.3:1981 -> MY.NET.221.138:1214
```

Top 5 Source IP Addresses

Count	IP Address
2327	212.179.85.27
653	212.179.43.225
379	212.179.86.6
354	212.179.34.114
269	212.179.82.106

Top 5 Destination Addresses

Count	IP Address
2327	MY.NET.202.58
652	MY.NET.213.150
378	MY.NET.224.186
347	MY.NET.210.6
316	MY.NET.222.74

Top Source Ports

Count	Port
1620	1776
652	55746
376	1802
331	1806
191	1046

Top Destination Ports

Count	Port
4049	1214
652	4467
304	4349
206	80
35	

This detect may have been caused by a snort rule such as:

Conclusion

I wasn't familiar with a "Watchlist" alert so I had to do a little research. I searched http://www.securityspace.com/swhois/whois.html and found the following:

```
inetnum:     212.179.80.0 -
212.179.94.255
netname:     L2TP-PROJECT
descr:       2st-pool-Dailup-L2TP-client.
country:     IL
admin-c:     NP469-RIPE
tech-c:      NP469-RIPE
status:      ASSIGNED PA
notify:      hostmaster@isdn.net.il
mnt-by:      RIPE-NCC-NONE-MNT
changed:     hostmaster@isdn.net.il
20000402
source:      RIPE
```

```
route:       212.179.0.0/17
descr:       ISDN Net Ltd.
origin:      AS8551
notify:      hostmaster@isdn.net.il
mnt-by:      AS8551-MNT
changed:     hostmaster@isdn.net.il
19990610
source:      RIPE
```

```
person:      Nati Pinko
address:     Bezeq International
address:     40 Hashacham St.
address:     Petach Tikvah  Israel
phone:       +972 3 9257761
e-mail:      hostmaster@isdn.net.il
nic-hdl:     NP469-RIPE
changed:     registrar@ns.il 19990902
source:      RIPE
```

Fig. 3-21   Largest external source of Watchlist 000220 IL-ISDNNET-990517.

The majority of the data seen seems to be from the KaZaa peer-to-peer media file sharing program. (http://www.kazaa.com) In the practical by Simon Whiting called "SANS GIAC – Intrustion Detection Assignments – Darling Harbour 2001" http://www.sans.org/y2k/practical/Simon_Whiting_GCIA.doc the author makes reference to the Watchlist 000220 IL-ISDNNET-990517 alert. While reading the previous practicals on the SANS webpage, I found the paper of Khan, Faud "GCIA Practical" February 19, 2001. URL: http://www.sans.org/y2k/practical/Faud_Khan_GCIA.doc This analyst also detected a questionable amount of traffic getting flagged by snort for the alert "Watchlist 000220 IL-ISDNNET-990517".

Recommendation

The fact that these hosts are coming from a "Watchlist" which usually denotes they have "questionable" intentions, the network should be configured to block this address block at the firewall.

Alert	Alert Count
High port 65535 tcp – possible Red Worm – traffic	3650

Traffic Sample

```
09/03-10:36:30.163533  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.150.133:1214 ->
193.251.91.101:65535
09/03-13:44:17.044602  [**] High port 65535 tcp - possible Red Worm - traffic [**] 209.185.123.128:25 -> MY.NET.6.47:65535
09/03-14:42:10.428598  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.234.138:4682 ->
24.114.117.16:65535
09/03-15:06:28.278549  [**] High port 65535 tcp - possible Red Worm - traffic [**] 211.90.88.43:65535 -> MY.NET.242.218:80
09/03-15:06:28.278616  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.242.218:80 -> 211.90.88.43:65535
```

Top 5 Source IP Addresses

Count	IP Address
3601	130.161.37.101
16	216.45.89.78
7	62.22.33.169
6	MY.NET.253.43
4	MY.NET.234.138

Top 5 Destination Addresses

Count	IP Address
641	130.161.37.101
16	MY.NET.236.110
6	206.210.69.141
5	MY.NET.6.47
4	62.22.33.169

Top Source Ports

Count	Port
3647	65535
641	3128
13	25
7	80
2	4349

Top Destination Ports

Count	Port
3601	3128
672	65535
15	33117
10	80
9	25

This detect may have been caused by a snort rule such as:

```
alert tcp any any -> $HOME_NET 65535 (msg:"High port 65535 tcp – possible Red Worm - traffic";
classtype:bad-unknown; rev:1;)
```

Conclusion

This worm exploits and old vulnerability that should have been patched ages ago. By actively monitoring the traffic on the network, I think you will see that anything to port 65535 should be suspect.

I believe the following traffic to be of some concern. It appears an internal host (MY.NET.6.47) might be infected and is currently being exploited. Without full fidelity logs though, we can only guess, especially when we see a low port such as 25 involved. We need more data to make a statement about this host.

```
09/01-03:49:06.944322  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.6.47:65535 -> 209.96.210.81:25
09/01-03:49:07.006849  [**] High port 65535 tcp - possible Red Worm - traffic [**] 209.96.210.81:25 -> MY.NET.6.47:65535
09/03-13:44:16.554898  [**] High port 65535 tcp - possible Red Worm - traffic [**] 209.185.123.128:25 -> MY.NET.6.47:65535
09/03-13:44:16.870319  [**] High port 65535 tcp - possible Red Worm - traffic [**] 209.185.123.128:25 -> MY.NET.6.47:65535
09/03-13:44:17.044602  [**] High port 65535 tcp - possible Red Worm - traffic [**] 209.185.123.128:25 -> MY.NET.6.47:65535
09/03-13:44:17.110223  [**] High port 65535 tcp - possible Red Worm - traffic [**] 209.185.123.128:25 -> MY.NET.6.47:65535
09/04-06:00:58.013034  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.6.47:3128 -> 130.161.37.101:65535
```

Recommendation

The best advise I can give is to sign up for the various security related mailing lists to stay on top of all of the latest available software patches and implement them promptly. This action will save you countless headaches in this business.

Top Snort Alerts (Outgoing from MY.NET)

The following table identifies traffic with its source being the network MY.NET.

Alert	Count
ICMP Destination Unreachable (Communication Administratively Prohibited)	31054
ICMP Echo Request Nmap or HPING2	10805
INFO MSN IM Chat data	9012
INFO napster login	8603
Possible trojan server activity	5574
ICMP Destination Unreachable (Network Unreachable)	4690
INFO Inbound GNUTella Connect accept	1782
INFO Napster Client Data	1727
ICMP traceroute	1278
INFO Possible IRC Access	1214

Fig. 3-22 Traffic leaving the network MY.NET.

Alerts not covered in the previous lists (Top Snort Alerts)

Alert	Alert Count
INFO napster login	8603

Traffic Sample

```
09/03-15:07:27.312145  [**] INFO napster login [**] MY.NET.235.106:4623 -> 208.184.216.98:8888
09/03-15:07:36.718800  [**] INFO napster login [**] MY.NET.201.246:1890 -> 64.124.41.157:8888
09/03-15:08:15.297016  [**] INFO napster login [**] MY.NET.235.106:4629 -> 208.184.216.38:8888
09/03-15:09:02.040539  [**] INFO napster login [**] MY.NET.201.246:1899 -> 64.124.41.152:8888
09/03-15:11:27.306087  [**] INFO napster login [**] MY.NET.235.106:4654 -> 208.184.216.16:8888
```

Top 5 Source IP Addresses

Count	IP Address
2400	MY.NET.226.118
1991	MY.NET.235.106
1281	MY.NET.207.110
567	MY.NET.227.94
524	MY.NET.201.246

Top 5 Destination Addresses

Count	IP Address
180	208.184.216.10
114	208.184.216.98
111	208.184.216.84
106	208.184.216.55
105	208.184.216.32

Top Source Ports

Count	Port
8	3083
8	1025
8	2660
8	1457
8	1964

Top Destination Ports

Count	Port
8603	8888

This detect may have been caused by a snort rule such as:

```
alert tcp $HOME_NET !80 -> $EXTERNAL_NET 8888 (msg:"INFO napster login"; flags: A+; content:"|00
0200|"; offset: 1; depth: 3;  classtype:bad-unknown; sid:549; rev:1;)
```

Conclusion

This is much like the MSN Chat alert listed earlier. It is more of a question of acceptable use than anything else. There are concerns related to the sucking up of bandwidth resources, and the constant fear of any kind of a virus outbreak from shared files whenever these file sharing technology programs are used in an intranet/Internet environment.

Recommendation

Activity like this can be blocked at the firewall, but the policy for the network should be reviewed to see if this is an acceptable product to be running on the network or not.

Alert	Alert Count
Possible Trojan server activity.	5574

Traffic Sample

```
09/03-23:52:39.609743  [**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
09/03-23:52:46.417494  [**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
09/03-23:53:46.735540  [**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
09/03-23:53:50.242293  [**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
09/03-23:53:58.499274  [**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
```

Top 5 Source IP Addresses

Count	IP Address
3377	MY.NET.98.190
2178	MY.NET.235.14
29	172.130.79.50
6	MY.NET.60.14
6	199.174.122.13

Top 5 Destination Addresses

Count	IP Address
2178	149.2.31.6
476	MY.NET.98.190
29	MY.NET.205.142
19	129.177.122.17
16	142.163.126.17

Top Source Ports

Count	Port
2178	6346
523	27374
15	80
6	4547
6	2568

Top Destination Ports

Count	Port
5574	27374
28	3642
11	80
3	4788
	4663

This detect may have been caused by a snort rule such as:

```
alert tcp any 27374 -> $HOME_NET any (msg “Possible Trojan server activity”;)
```

Conclusion

The traffic displayed in the sample for this detect shows activity on source port 27374. According to Network Ice, “This is the most commonly probed port on the Internet right now…” (http://advice.networkice.com/advice/Exploits/Ports/27374/default.htm)
This is the default port for SubSeven Trojan. It is also used for the Lion and the Ramen worm. For more information, see the pages located at http://www.incidents.org/react/lion.php and http://service2.symantec.com/SARC/sarc.nsf/html/Linux.Ramen.Worm.html respectively. One host I recommend that gets looked at is MY.NET.98.190 who seems to be doing some pretty heavy scanning of many external IP addresses for the Subseven Trojan. On first analysis of the traffic, it looks like MY.NET.235.14 is controlling 149.2.31.6 but upon further analysis, we see the source port of 6346 to destination port 27374. TCP port 6436 is the default port for the GNUTella file sharing program. Due to the low fidelity of the logs, we are unable to observe the actual TCP handshake and the circumstances surrounding the connection so we cannot say for sure we are observing infections.

Recommendation

Monitoring the IDS will help the administrators be aware of this activity, but by actively working to secure the internal network through virus scanning, policy enforcement, and as a last resort even blocking known Trojan ports from coming in at the border router, this kind of activity can be minimized. This activity must be verified and in the meantime, these machines should be considered compromised until it can be confirmed otherwise.

Alert	Alert Count
ICMP Destination Unreachable (Network Unreachable)	4690

Traffic Sample

```
09/04-00:00:13.042590  [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 195.78.199.37
09/04-00:00:58.355181  [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 200.161.65.101
09/04-00:00:58.537609  [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 211.90.176.59
09/04-00:01:00.717673  [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 211.90.176.59
09/04-00:03:14.393228  [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 200.204.148.162
```

Top 5 Source IP Addresses

Count	IP Address
4690	MY.NET.30.2
13	131.118.255.17
1	139.134.52.22
1	152.63.7.145
1	198.59.55.1

Top 5 Destination Addresses

Count	IP Address
263	212.199.28.76
210	211.90.176.59
174	200.250.65.1
110	211.90.88.43
99	217.128.232.163

Top Source Ports

Count	Port

Top Destination Ports

Count	Port

This detect may have been caused by a snort rule such as:

alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Network Unreachable)"; itype: 3; icode: 0; sid:401; rev:1;)

Conclusion

Again, this leads back to a router configuration issue. By the router returning these types of messages to the originator (that is out of our network) we are allowing valuable network configuration information out that can be used against us later.

Recommendation

Messages like this should be quietly dropped at the router. As noted in the Stephen Northcutt and Judy Novak book "Network Intrusion Detection: An Analyst's Handbook." 2nd ed. Indianapolis: New Riders, 2000. "It is possible to silence some Cisco routers by putting a statement sucah as "no ip unreachables" in the access control list."

Alert	Alert Count
INFO Inbound GNUTella Connect accept	1782

Traffic Sample

```
09/04-00:05:39.281892  [**] INFO Inbound GNUTella Connect accept [**] MY.NET.219.138:6346 -> 66.1.228.43:2965
09/04-00:06:21.934286  [**] INFO Inbound GNUTella Connect accept [**] MY.NET.205.146:6346 -> 208.239.76.100:1149
09/04-00:07:10.528384  [**] INFO Inbound GNUTella Connect accept [**] MY.NET.205.146:6346 -> 198.82.108.183:3352
09/04-00:14:34.342058  [**] INFO Inbound GNUTella Connect accept [**] MY.NET.234.42:6346 -> 66.31.35.31:3047
09/04-00:15:24.571152  [**] INFO Inbound GNUTella Connect accept [**] MY.NET.219.138:6346 -> 172.148.12.70:1127
```

Top 5 Source IP Addresses

Count	IP Address
160	MY.NET.108.42
135	MY.NET.202.102
95	MY.NET.202.102
87	MY.NET.203.66
78	MY.NET.223.78

Top 5 Destination Addresses

Count	IP Address
86	208.239.76.100
11	64.61.25.140
9	128.211.205.61
6	142.177.194.22
4	148.61.242.38

Top Source Ports

Count	Port
1698	6346
41	5634
28	6347
4	6390
4	6357

Top Destination Ports

Count	Port
11	1025
6	2596
4	4249
4	4737
4	1151

This detect may have been caused by a snort rule such as:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"INFO Inbound GNUTella Connect accept";
content: "GNUTELLA OK"; nocase; depth: 40; classtype:bad-unknown; sid:557; rev:1;)
```

Conclusion

This is another of the many peer-to-peer file sharing technologies available as alternatives to Napster now. The risks associated with this activity are such things as virii, and users having misconfigured network shares, loss of productivity, etc.

Recommendation

This is an issue for the acceptable use policy and if required, blocking at the firewall for such activity.

Alert	**Alert Count**
INFO Napster Client Data	1727

Traffic Sample

```
09/04-00:20:29.007564  [**] INFO Napster Client Data [**] MY.NET.205.166:1059 -> 64.81.224.227:6699
09/04-00:27:44.265148  [**] INFO Napster Client Data [**] MY.NET.205.102:1055 -> 24.129.213.136:6699
09/04-00:29:18.266739  [**] INFO Napster Client Data [**] MY.NET.219.6:1301 -> 216.129.74.254:6699
09/04-00:33:45.817589  [**] INFO Napster Client Data [**] MY.NET.236.102:1152 -> 24.248.154.159:6699
09/04-00:36:01.122929  [**] INFO Napster Client Data [**] MY.NET.205.102:1060 -> 24.129.213.136:6699
```

Top 5 Source IP Addresses

Count	IP Address
728	MY.NET.219.86
114	MY.NET.201.246
74	MY.NET.236.250
47	MY.NET.224.150
43	MY.NET.205.102

Top 5 Destination Addresses

Count	IP Address
726	64.129.230.53
47	65.34.30.109
28	128.119.33.225
24	213.46.106.88
20	MY.NET.219.178

Top Source Ports

Count	Port
728	1025
47	3565
36	7777
27	6699
19	6666

Top Destination Ports

Count	Port
1594	6699
73	6666
60	7777
20	40798
11	1610

This detect may have been caused by a snort rule such as:

```
alert tcp $HOME_NET any <> $EXTERNAL_NET 6699 (msg:"INFO Napster Client Data"; flags: A+;
content:".mp3"; nocase; classtype:bad-unknown; sid:561; rev:1;)
```

Conclusion

Again, it appears that more peer-to-peer file sharing basically is what is going on. There is always the potential for malicious users with these types of activities. It is an acceptable use issue.

Recommendation

Attention should be paid to the IDS to identify such users and watch for any suspicious activities. There is always the possibility of blocking this activity from coming in at the firewall.

Alert	Alert Count
ICMP traceroute	1278

Traffic Sample

```
09/04-00:49:09.349236  [**] ICMP traceroute  [**] MY.NET.221.22 -> MY.NET.14.1
09/04-00:54:28.638113  [**] ICMP traceroute  [**] MY.NET.228.150 -> MY.NET.14.1
09/04-01:02:41.062864  [**] ICMP traceroute  [**] MY.NET.208.86 -> 130.244.141.251
09/04-01:19:02.033489  [**] ICMP traceroute  [**] MY.NET.222.78 -> 202.232.85.151
09/04-01:29:30.151651  [**] ICMP traceroute  [**] MY.NET.211.254 -> 130.244.215.243
```

Top 5 Source IP Addresses

Count	IP Address
12	MY.NET.209.42
11	MY.NET.234.102
11	MY.NET.209.246
8	MY.NET.220.118
8	MY.NET.223.174

Top 5 Destination Addresses

Count	IP Address
531	MY.NET.14.1
7	MY.NET.70.135
4	MY.NET.134.1
4	MY.NET.132.1
4	209.255.109.160

Top Source Ports

Count	Port

Top Destination Ports

Count	Port

This detect may have been caused by a snort rule such as:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute ";ttl:1;itype:8;
reference:arachnids,118; classtype:attempted-recon; sid:385; rev:1;)
```

Conclusion

As mentioned earlier, traceroute is used to map out a path from a source to its destination. A potential attacker now has a list of active hosts that can be used for a more active reconnaissance at a later time.

Recommendation

This activity can be blocked at the border router or firewall.

Alert	Alert Count
INFO Possible IRC Access	1214

Traffic Sample

```
09/04-01:58:36.938891  [**] INFO Possible IRC Access [**] MY.NET.98.199:3108 -> 216.177.89.36:6667
09/04-05:48:44.953143  [**] INFO Possible IRC Access [**] MY.NET.212.86:2935 -> 151.189.12.20:6668
09/04-05:52:50.269237  [**] INFO Possible IRC Access [**] MY.NET.212.86:2935 -> 151.189.12.20:6668
09/04-08:45:57.802891  [**] INFO Possible IRC Access [**] MY.NET.153.171:1334 -> 207.46.216.29:6667
09/04-09:14:23.095681  [**] INFO Possible IRC Access [**] MY.NET.60.8:41618 -> 128.138.129.31:6667
```

Top 5 Source IP Addresses

Count	IP Address
955	MY.NET.134.14
38	MY.NET.206.186
15	MY.NET.60.8
13	MY.NET.60.11
10	MY.NET.221.206

Top 5 Destination Addresses

Count	IP Address
955	216.138.228.204
38	206.139.136.5
19	216.177.89.36
13	151.189.12.20
12	207.46.216.29

Top Source Ports

Count	Port
6	1048
5	38583
4	2481
4	1366
3	2935

Top Destination Ports

Count	Port
1183	6667
14	6666
10	7000
7	6668

This detect may have been caused by a snort rule such as:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 6666:6669 (msg:"INFO Possible IRC Access"; flags: A+;
content: "NICK "; classtype:not-suspicious; sid:542; rev:1;)
```

Conclusion

IRC is a great place to chat and hangout if you are having issues with something but its usefulness is outweighed probably by its security issues when used on an internal network.

Recommendation

This activity should be investigated to determine if it is required. Just going to http://www.securityfocus.com/cgi-bin/search.pl and performing a search on IRC returns a handful of vulnerabilities against various IRC clients and servers.

Overall Conclusion

In closing, this network has several issues. There are computers that have are running questionable services, routers leaking potentially valuable information, and even some trojaned computers. It is recommended that there be some investigation into the following types of traffic to determine the acceptable use of such programs like: MSN Instant Messaging, napster, GNUTella, and IRC. It appears there is not currently a good strong security policy in place on site. The requirement for well maintained and up to date patched systems cannot be overestimated as most of the vulnerabilities being exploited these days seem to come from systems that have had patches available for exploitable services for some time now. The procurement of some sort of backup power supply should be looked into to ensure that the log integrity is kept to a maximumThere currently appears to be some issues with not being able to find all of the files for each day, leading me to believe that there are backup problems. I see a definite requirement for an improved sensor configuration. There exists a need to better identify and analyze attacks against the network. I talked a lot about an Acceptable Use Policy (AUP). This document sets out what is the level of acceptable standards for users to perform on the network. This should be reviewed and heavily enforced to minimize potential breaches. I strongly encourage the network owners to review the recommendations presented here and consider implementing some of the defensive policies talked about.

References

Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Reading: Addison Wesley Longman, Inc, 1994.

Stephen Northcutt, Judy Novak, and Donald McLachlan. Network Intrusion Detection: An Analyst's Handbook.
2nd ed. Indianapolis: New Riders, 2000.

Roesch, Martin "Snort Users Manual - Snort Release: 1.8.1"
URL: http://snort.sourcefire.com/docs/writing_rules/

Roesch, Martin "SNORT FAQ Version 1.8 - July 10 2001 v1.8.1" 10 July 2001.
URL: http://www.snort.org/docs/faq.html

Braden, R; Editor.
“Request for Comments: 1122 - Requirements for Internet Hosts – Communication Layers”.
Oct 1989. URL: http://www.faqs.org/rfcs/std/std3.html

Chmielarski, Tom. "Reconnaissance Techniques using Spoofed IP Addresses." 4 Apr 2001.
URL: http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm

Bruneau, Guy. "Build Securely a Shadow Sensor Step-by-Step Powered by Slackware Linux "
URL: http://members.home.com/gbruneau1/webdoc1.htm

Bruneau, Guy. "Guy Bruneau: GCIA Practical Assignment"
URL: http://www.sans.org/y2k/practical/Guy_Bruneau.doc

French, Jamie. "Jamie French: GCIA Practical Assignment" 23 Sep 2000.
URL: http://www.sans.org/y2k/practical/Christopher_French_GCIA.zip

French, Jamie. "Whitehats.ca" URL: http://www.whitehats.ca

Galvin, Bradley. "Bradley Galvin: GCIA Practical Assignment." April 2001.
URL: http://www.sans.org/y2k/practical/Bradley_Galvin_GCIA.doc

Kuenthe, Chris. "Chris Kuethe: GCIA Practical Assignment." 24 Jun 2000.
URL: http://www.sans.org/y2k/practical/chris_kuethe_gcia.html

Lenny, Lenny. "Lenny Lenny: GCIA Practical Assignment." 15 August 2000.
URL: http://www.sans.org/y2k/practical/Lenny_Zeltser.htm

Bayerkohler, Marc. "Marc Bayerkohler : GCIA Practical Assignment."
URL: http://www.sans.org/y2k/practical/Marc_Bayerkohler_GCIA.html

Rekhter, Y; Moskowitz, B; Karrenberg, D; de Groot, G.J.; Lear, N
“Request for Comments: 1918 - Address Allocation for Private Internets.” February 1996.
URL: http://www.faqs.org/rfcs/rfc1918.html

Chris Brenton “Lion Worm Version 0.1” March 26 2001
URL: http://www.incidents.org/react/lion_protection.php

Novak, Judy. “Network Traffic Analysis Using TCP Dump book 3.2” 2000,2001
2001, The SANS Institute training material from Intrusion Detection – In Depth.

Reynolds, J.; Postel, J. “Request for Comments: 1700 STD: 2” October 1994
URL: http://www.faqs.org/rfcs/std/std2.html

Braden, R.; Postel, J. “Request for Comments: 1009” June 1987
URL: http://www.faqs.org/rfcs/std/std4.html

Frederick, Karen “Abnormal IP Packets” October 13 2000
URL: http://www.securityfocus.com/infocus/1200

Langley, Richard “Securing Your Internet Access Router” January 23, 2001
URL: http://www.sans.org/infosecFAQ/firewall/router.htm