GIAC Intrusion Detection

Level II Practical Assignment

SANS Monterey

October 15 – 22, 2000

Oliver Viitamaki

Table of Contents

Introduction

I work in a commercial business environment. The business practices require that where ever possible commercially available tools are implemented. The use of tools freely available on the Internet is heavily discouraged, in other than when used for a proof of concept project. This has placed me in a position where the vast majority of the traces have been gathered and analyzed using the tools made available to me. In the interest of making the best use of my time, I have chosen to use those tools that I am familiar with. Naturally the presentation of the information will be different and in some cases more colorful, than that of Snort, TCPDump, or Shadow, but the relevant details are there. The following Uniform Resource Locators (URLs) point to the various sites where information on the products can be found.

Network ICE Black ICE: http://www.networkice.com/html/small_home_office.html
AgGroup Now known as WildPackets (Etherpeek): http://www.wildpackets.com/products
Network Associates: http://www.sniffer.com/asp_set/products/tnv/das.asp

1.0 Assignment 1 - Network Detects

1.1 Detect 1 – Network ICE, Black ICE “Intrusion” detection on a Home Dialup connection

Summary: This is an example of a false detect. It occurs because of the choice of Internet Service provider, and associated traffic generated by that Service Provider, due to the Service agreement with the home user, and the user chosen settings established in the setup of the Firewall.

Background: The following detects are from a Network ICE BlackICE Firewall installed on a Home computer. The machine is running Windows 98 Second Edition, with current patches as recommended by Microsoft. The connection to the Internet is made through a 56Kbps dialup modem, the Internet provider is Juno. The actual Destination IP addresses have been replaced by *home.ip*. Netscape 4.75 is the browser used during the connected sessions. The current, version of McAfee VirusScan is running.

```
Attack-List.csv
#Severit  timestamp (GMT)      issue Id   Issue Name              Intruder IP      Intruder Name          Victim IP          parameters
y
      39  2000-10-22 22:33:42   2003102   TCP port probe          63.211.172.77    m14.boston.juno.com    home.ip.134.162    port=1037
      39  2000-10-22 22:38:25   2003102   TCP port probe          63.211.172.77    m14.boston.juno.com    home.ip.136.216    port=1031
      59  2000-10-23 00:47:29   2000318   TCP Invalid Urgent offset 205.188.140.185 ads.web.aol.com       home.ip.136.216    port=3059&flags=FAU&o
      59  2000-10-23 00:58:23   2000318   TCP Invalid Urgent offset 152.163.180.25 ads.web.aol.com        home.ip.136.216    port=1244|3367&flags=F
```

```
Date.enc
Severity  Source IP Name      Source IP Address    Source Port   Destination IP Address   Destination Port   Packet size  Protocol
      39 IP-63.211.172.77     IP-63.211.172.77     IP-64162      IP-home.ip.134.162       IP-1037                     62  IP TCP
      39 IP-63.211.172.77     IP-63.211.172.77     IP-64162      IP-home.ip.134.162       IP-1037                     62  IP TCP

      39 IP-63.211.172.77     IP-63.211.172.77     IP-65020      IP-home.ip.136.216       IP-1031                     62  IP TCP
      39 IP-63.211.172.77     IP-63.211.172.77     IP-65020      IP-home.ip.136.216       IP-1031                     62  IP TCP

      59 IP-205.188.140.185   IP-205.188.140.185   IP-80         IP-home.ip.136.216       IP-3059                     58  TCP HTTP
      59 IP-205.188.140.185   IP-205.188.140.185   IP-80         IP-home.ip.136.216       IP-3059                     58  TCP HTTP
      59 IP-205.188.140.185   IP-205.188.140.185   IP-80         IP-home.ip.136.216       IP-3059                     58  TCP HTTP
      59 IP-205.188.140.185   IP-205.188.140.185   IP-80         IP-home.ip.136.216       IP-3059                     58  TCP HTTP

      59 ads.web.aol.com      IP-152.163.180.25    IP-80         IP-home.ip.136.216       IP-1244                     58  TCP HTTP
      59 ads.web.aol.com      IP-152.163.180.25    IP-80         IP-home.ip.136.216       IP-1244                     58  TCP HTTP
      59 ads.web.aol.com      IP-152.163.180.25    IP-80         IP-home.ip.136.216       IP-1244                     58  TCP HTTP

      59 ads.web.aol.com      IP-152.163.180.25    IP-80         IP-home.ip.136.216       IP-3367                     58  TCP HTTP
      59 ads.web.aol.com      IP-152.163.180.25    IP-80         IP-home.ip.136.216       IP-3367                     58  TCP HTTP
```

59 ads.web.aol.com	IP-152.163.180.25	IP-80	IP-home.ip.136.216	IP-3367	58	TCP HTTP
59 ads.web.aol.com	IP-152.163.180.25	IP-80	IP-home.ip.136.216	IP-3367	58	TCP HTTP

The following packet decode displays the generic features of the First warning in the "Attack-List" File, and the First and Second entries in the *Date*.enc file.

```
 Flags:      0x20  Runt                                      Analysis and tracking information inserted by Etherpeek
 Status:      0x00                                           before the Ethernet header.
 Packet Length:62
 Timestamp:    15:33:21.334000 10/22/2000
Ethernet Header                                                      Start of actual Ethernet Packet
 Destination:  44:45:53:54:00:00  [0-5]
 Source:      20:53:52:43:00:00  [6-11]
 Protocol Type:0x0800  IP  [12-13]
IP Header - Internet Protocol Datagram
 Version:          4  [14 Mask 0xF0]
 Header Length:      5  (20  bytes)  [14 Mask 0x0F]
 Type of Service:     %00000000  [15]
 Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
 Total Length:       44  [16-17]
 Identifier:        28178  [18-19]
 Fragmentation Flags:  %010  Do Not Fragment   Last Fragment  [20 Mask 0xE0]
 Fragment Offset:      0  (0  bytes)  [20-21 Mask 0x1FFF]
 Time To Live:       243  [22]
 Protocol:          6  TCP  [23]
 Header Checksum:     0x66EC  [24-25]
 Source IP Address:    63.211.172.77  [26-29]
 Dest. IP Address:    home.ip.134.162  [30-33]
 No IP Options
TCP - Transport Control Protocol
 Source Port:     64162  [34-35]
 Destination Port: 1037  [36-37]
 Sequence Number:  2325994121  [38-41]
 Ack Number:       0  [42-45]
 Offset:          6  [46 Mask 0xF0]
 Reserved:        %000000  [46-47 Mask 0x0FC0]
 Code:          %000010  [47 Mask 0x3F]
         Synch Sequence
 Window:         8760  [48-49]
 Checksum:        0x5F44  [50-51]
 Urgent Pointer:  0  [52-53]
 TCP Options:
  Option Type:   2  Maximum Segment Size  [54]
    Length:    4  [55]
    MSS:       1460  [56-57]
 TCP Data Area:    No more data.
Frame Check Sequence:  0x04004800  [58-61]
```

The following packet decode displays the generic features of the Second warning in the “Attack-List” File, and the Third and Fourth entries in the *Date*.enc file.

```
 Flags:       0x20  Runt
 Status:      0x00
 Packet Length:58
 Timestamp:    15:55:43.275000 10/22/2000
```

Analysis and tracking information inserted by Etherpeek before the Ethernet header.

Start of actual Ethernet Packet

```
Ethernet Header
 Destination:  44:45:53:54:00:00  [0-5]
 Source:      20:53:52:43:00:00  [6-11]
 Protocol Type:0x0800  IP  [12-13]
IP Header - Internet Protocol Datagram
 Version:            4  [14 Mask 0xF0]
 Header Length:       5  (20  bytes)  [14 Mask 0x0F]
 Type of Service:     %00000000  [15]
 Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
 Total Length:       40  [16-17]
 Identifier:       7199  [18-19]
 Fragmentation Flags:  %000  May Fragment   Last Fragment  [20 Mask 0xE0]
 Fragment Offset:     0  (0  bytes)  [20-21 Mask 0x1FFF]
 Time To Live:       49  [22]
 Protocol:           6  TCP  [23]
 Header Checksum:     0x5812  [24-25]
 Source IP Address:   152.163.180.25  ads.web.aol.com  [26-29]
 Dest. IP Address:    home.ip.136.216  [30-33]
 No IP Options
TCP - Transport Control Protocol
 Source Port:      80  World Wide Web HTTP  [34-35]
 Destination Port: 1244  [36-37]
 Sequence Number:  2942350989  [38-41]
 Ack Number:      1185074  [42-45]
 Offset:         5  [46 Mask 0xF0]
 Reserved:        %000000  [46-47 Mask 0x0FC0]
 Code:           %110001  [47 Mask 0x3F]
          Urgent is valid
           Ack is valid
           FIN (Sender End of Byte Stream)
 Window:         16384  [48-49]
 Checksum:        0xD9B4  [50-51]
 Urgent Pointer:   1  [52-53]
 No TCP Options
   No More HTTP Data
Frame Check Sequence:  0x04004400  [54-57]
```

The following packet decode displays the generic features of the Third warning in the “Attack-List” File, and the Fifth through Twelfth entries in the *Date*.enc file.

Analysis and tracking information inserted by Etherpeek before the Ethernet header.

```
 Flags:      0x20  Runt
 Status:     0x00
 Packet Length:58
 Timestamp:    17:58:05.562000 10/22/2000
```

Start of actual Ethernet Packet

```
Ethernet Header
 Destination:  44:45:53:54:00:00  [0-5]
 Source:      20:53:52:43:00:00  [6-11]
 Protocol Type:0x0800  IP  [12-13]
IP Header - Internet Protocol Datagram
 Version:           4  [14 Mask 0xF0]
 Header Length:       5  (20  bytes)  [14 Mask 0x0F]
 Type of Service:     %00000000  [15]
 Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
 Total Length:       40  [16-17]
 Identifier:        17564  [18-19]
 Fragmentation Flags:  %000  May Fragment   Last Fragment  [20 Mask 0xE0]
 Fragment Offset:     0  (0  bytes)  [20-21 Mask 0x1FFF]
 Time To Live:       49  [22]
 Protocol:          6  TCP  [23]
 Header Checksum:     0x2F95  [24-25]
 Source IP Address:   152.163.180.25  ads.web.aol.com  [26-29]
 Dest. IP Address:    home.ip.136.216  [30-33]
 No IP Options
TCP - Transport Control Protocol
 Source Port:      80  World Wide Web HTTP  [34-35]
 Destination Port: 3367  [36-37]
 Sequence Number:  3765563481  [38-41]
 Ack Number:       8526254  [42-45]
 Offset:         5  [46 Mask 0xF0]
 Reserved:        %000000  [46-47 Mask 0x0FC0]
 Code:          %110001  [47 Mask 0x3F]
         Urgent is valid
          Ack is valid
```

FIN (Sender End of Byte Stream)
Window: 16384 [48-49]
Checksum: 0x61A0 [50-51]
Urgent Pointer: 1 [52-53]
No TCP Options
No More HTTP Data
Frame Check Sequence: 0x04004400 [54-57]

1.1.1 Source of Trace:

The detects are from a Network ICE BlackICE Firewall Version 2.1.cn installed on a Home computer running Windows 98 Second Edition with updates. Black ICE generates three files as evidence, the two files which are of interest in this situation are Attack-list.csv, and *Date*.enc.
The Attack-list.csv description below is from the Network ICE site at http://advice.networkice.com/Advice/Support/KB/q000018/

The file "Attack-list.csv" contains the list of intrusions that the product found. The primary information lists the attack and the suspected intruder. This article explains the file format. The columns are, from left to right:

"Severity"
This is a number from 1-99 that indicates the severity of an attack, where 1 is not very severe, and 99 is the most severe attack. Unfortunately, these levels do not have any precise meaning. Even an attack at level 1 may result in a compromise of the machine, whereas an attack at level 99 could be harmless. The assigned level is just a best-guess.

"timestamp"
This indicates the time and date of the last time the attack occurred. Attacks are "coalesced", meaning that if the same attack occurs multiple times, earlier attacks are sometimes removed from the list and simply merged with the latest one. A count of the number of times an attack has occurred is kept in another column. This timestamp is kept in GMT (aka UTC), and is probably several hours off from the time you see in the user interface. The ISP will want the time in this format so they don't have to worry about what timezone you are in.

"issueId"
A numeric identifier for this attack type. Each of the more than 300 attacks that the intrusion-detection component detects is assigned a unique number. This number is used for all internal processing of events. This number may also be pasted at the end of the URL http://advice.networkice.com/advice/intrusions/<num> in order to get help on the event.

"issueName"
The name of the attack. Each of the unique "issueId" numbers has a name associated with it.
"intruderIp "The IP address of the attacker. Remember that IP addresses can sometimes be "spoofed" (forged), or that an intrusion may be a "false-positive", so there isn't a 100% chance that this is actually a hostile

```
        person.

        "intruderName"
        The name of the intruder. We scan both Internet databases like DNS as well as the attacker itself in order to
        find the "best-name" of the machine, then display it here.

        "victimIp"
        This is the IP address of who the intruder was attacking. For example, if a user is running the product and
        gets attacked on a dial-up, then this will be the IP address assigned to that machine during that dialup
        session.

        "parameters"
        This contains some detailed information about the attack. For example, in a "TCP port probe" scan, this will
        contain a list of "ports" the attacker was scanning. The meaning of this information is documented in the
        "advICE" database.


        "count"
        The number of times this attack was seen.
```

The *Date*.enc file actually has a much longer name in the form of evdyyyymmdd-nn.enc identifying the following yyyy=year, mm=month, dd=date, nn= version. It has been shortened in the interest of clarity The *Date*.enc file contains, sections of the actual packet traces, in standard network sniffing program format, these have been opened using Etherpeek for Windows, Version 4.0.2. The information included at the beginning of the packet, before the Ethernet Header has been placed there by Etherpeek. Network ICE does not save a full packet, of data, which is similar to TCPDump, therefore Etherpeek refers to it as a Runt. The packet decode is visually more appealing than TCPDump, but contains the same detail of information. The field naming is conventional, the numbers contained in brackets are the offset values in the packet, the flag fields are shown printed out, the mask values are displayed in Hexadecimal. If there are questions about the remainder of the data, in the files, the following references can be used resolve those questions:

RFC 793, which contains the original specification of TCP. http://www.isi.edu/in-notes/rfc793.txt
TCP/IP Illustrated Volume 1 by W. Richard Stevens, Chapters 17 through 24
To access the Etherpeek manual, requires downloading and installing the demonstration version of Etherpeek, if one does not have it, it can be found at http://www.wildpackets.com/products/etherpeek

1.1.2 Detect was generated by:

Network Ice BlackICE was used to generate the detects, There are 4 configurable options for detects, “Paranoid – blocks all unsolicited inbound traffic (the chosen setting), Nervous - blocks most inbound traffic, Cautious - blocks some inbound traffic, Trusting – allows all inbound traffic”. The “Paranoid” mode of operation produces the largest number of alerts, but also provides the highest degree of protection. The BlackICE QuickStart Guide does not describe how the Firewall works, or what the designers of the product placed in each of the 4 categories of options. Therefore it is difficult to determine exactly what the decision process that the firewall went through to create these “detects’.. The two issues that caused the alert to occur from the Firewall are Issue ID 2000318 and Issue ID 2003102. They are described as follows, from the BlackICE documentation:

```
“Issue ID 2000318
TCP Invalid Urgent offset
Summary: Some TCP/IP implementations will hang when receiving many such frames.
```

```
The intruder sends a TCP frame with an Urgent pointer which points past the end of the data.
This may cause some TCP/IP implementations to become unstable or crash.

The TCP flags from the offending frame.
The flags are: S (SYN), F (FIN), R (RESET), P (PUSH), A (ACK), U(URGENT),
4 (low-order unused bit), 8 (high-order unused bit)
The TCP options from the offending frame.
The options are displayed as "option-value", separated by commas.
No-ops are not displayed”
```

“TCP port probe (2003102)
Summary

Somebody has tried to access your machine and failed.

Details

This is the most common intrusion detected on the Internet.
This is so common because hackers do frequent wide-spread scans looking for one specific exploit they can use to break into systems. The typical hacker scans thousands or millions of machines in a typical scan. In other words, the hacker isn't targeting you personally. In particular, this event is generated upon failed attempts, so there is no reason to worry. Probes like this result from "script-kiddies", hackers just above the skill level of trained monkeys. They download attack programs (called "scripts") from various sites on the net, then run them against millions of machines. There are thousands of script-kiddies out there, so if you have an always-on connection (cable-modem, DSL), then you can expect about one of these scans per day. About 10% of these scans are from forged addresses. This means the indicated IP address in the attack is probably from the real attack, but a small percentage of the time the indicated person is completely innocent. <p> About 20% of these scans are from machines already compromised by a hacker. In other words, if you report this scan back to the originator, they may thank you, because you've discovered a hacked system on their network they didn't know about. Information on reporting the hacker can be found in our support Knowledge Base article

Ports

A port is a point of entry into a system. Each program running on a system is reached through its own ports. You rarely see this detail because most port assignments are automatic. For example, most websites run at port 80 on a machine, so you never have to specify it yourself. This means that if you see a TCP port probe for port 80, then a hacker is most likely testing your system to see if you've installed your own web server. The exact port the intruder probed for is listed on your system in the file "attack-list.csv".

False Positives

The system errs on the side of caution. When your machine attempts to connect to a remote site and fails, sometimes this alert will trigger. Carefully watch the source of the attack in case it is your own machine. The system triggers on any failed connection. Some web-sites will attempt to contact your machine. For example, chat servers, FTP servers, and multimedia servers (video, audio) often open connections directed at your machine. If the firewall settings block this, then these will be reported as port probes.”

1.1.3 Probability the Source Address was Spoofed:

Domain Name Service lookups were performed with SamSpade Version1.14

```
nslookup 63.211.172.77
Canonical name: m14.boston.juno.com
Addresses: 63.211.172.77

nslookup 152.163.180.25
Canonical name: ads.web.aol.com
Addresses:152.163.180.25

nslookup 205.188.140.185
Canonical name: ads.web.aol.com
Addresses: 205.188.140.185
```

Very Low, all of the connections are TCP, and come from the following 3 addresses, 63.211.172.77 which resolves to m14.boston.juno.com, the service provider, as identified in the Juno.ini file, and the DNS lookup provided through SamSpade. 152.163.180.25, 205.188.140.185 resolve to ads.web.aol.com. It is prudent to view the two AOL machines as part of an advertising server cluster. The advertisement servers produce the constantly updating advertisements displayed, on the display of the home user.

Probability that the source address is looking for a Trojan, such as RingZero, or Sub Seven is very low. The ports in the group that raise suspicion are port 80, and 1244. They is referenced on several Trojan Lists, such as:

Simovits Consulting: http://www.simovits.com/nyheter9902.html
DoS Help listing of Trojan Ports: http://www.doshelp.com/trojanports.htm
Robert Graham Listing of ports: http://www.robertgraham.com/pubs/firewall-seen.html

The reason that this is considered low, is that the general traffic and port usage patterns ***do not*** match with that of Ring Zero, or SubSeven. The server at AOL would have to be infected, and participating in the spread of RingZero, or SubSeven and its versions, for this to be a concern. The Common Vulnerabilities and Exposures (CVE) project has assigned the name CAN-1999-0660 to this group of vulnerabilities. It can be found at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660 in addition to the well documented features of RingZero at the SANS web site at http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm .

1.1.4 Attack Mechanism:

The owner of the machine in question subscribes to a free Internet service. As a part of that subscription, in order for the service to be free, the owner has to agree to set the Juno website as the first address contacted when a connection is made to the Internet Service Provider. When user initiates a connection to the Internet, the program, Juno.exe executes in conjunction with the file Juno.ini (which contains the user specific settings, and the default initial connection to the host m14.boston.juno.com). An outgoing connection is established as a part of this process. The completion of the successful connection to the Internet elicits a response from the Juno server 63.211.172.77 on port 64162 in this example, other times other ports on the Juno server will be used. The connection from the Juno server is made back to home.ip.134.162 on port 1037 as it will be using the first available port on the home machine, this port changes from one connection to the internet, to the next. The port was negotiated in the background, by the Juno.exe, and supporting programs when the initial connection to the Internet was made. One additional program that starts at connection time is the advertisement display program. It is responsible for contacting the advertisement servers 152.163.180.25, 205.188.140.185, ads.web.aol.com, which is where the advertisements, which are displayed on the video display come from. These advertisements are configured to arrive and be displayed, on the machine regardless of what other things may be taking place, and on top of all other programs.

1.1.5 Description of Attack:

This set of detected events turn out to be a false positive to Network BlackICE. The BlackICE program has been designed and implemented, to watch for incoming connects to the machine on which it is installed. It does not keep track of, or assist the user in validating outgoing requests, therefore there is no recorded history of the outgoing SYN, return SYN/ACK (expected, normal event to BlackICE), outgoing ACK, and subsequent incoming and outgoing packets with the ACK bit set. A packet sequence normally associated with a successful 3 way TCP handshake and following communication finally ending with a FIN. Therefore it is blind to outgoing activity generated by the host machine on which it runs, which causes incoming requests to be generated. In all of the cases above the requests are due to user actions. The first two entries in the Attack-List.csv are a result of the machine completing two different connections to the Internet, over a short period of time (the connection dropped the first time). The entries following the first 2 in the ATTACK-List.csv file, are as a result of the ad-server contacting the client machine with advertising information which is to be displayed once again due to BlackICE design it does not record the initial 3 way handshake. The Urgent flag is shown as being set in the last 3 entries in the ATTACK-List.csv file, and as well, in the last two decodes, to insure that the advertising information receives a higher priority transport than any other network activity that the host machine may be engaging in. The Fin and ACK are set as a normal termination of the 3 way handshake between the client and the server.

1.1.6 Correlation's:

In this example a direct correlation would require connecting another machine to the same Internet Service Provider, with the same products and verifying that the same events occur. This is impractical. Thus we can make other observations, the events presented by BlackICE are not an attack, the Juno and AOL servers are connecting to the home machine as expected, and the DNS lookups using SamSpade, are consistent with what is expected.

1.1.7 Evidence of Active Targeting:

None at this time. The detection of numerous false positives in this situation, of using a Firewall which does not keep track of the TCP 3 way handshake, as well as of connection state, will tend to disguise an attack when it does occur. This will be caused by the user becoming accustomed to the BlackICE program indicating an attack is underway, the user checking it, and finding out that it is either initial connection validation, or advertising push, coming at the users machine,

thereby learning to ignore the attack indicator. The home user could choose to place these machines in the "Trusted Addresses", this would eliminate the warning messages, assuming that the machines in question were never spoofed or compromised.

1.1.8 Severity:

Target Criticality, 5, this is the only computer that the home user has
Attack Lethality, 0, this was a false positive
System Countermeasures, 5, the machine is current with patches
Network Countermeasures, 2, there is a firewall in place

Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)
Severity = (5 + 0) - (5 + 2)

Severity = -2

1.1.9 Defense Recommendations:

1. Insure there are good and frequent system backups made. Check frequently that they can be restored.
2. Consider changing to a different Firewall structure, settings, or product, one which is able to maintain state and validate outgoing connections.
3. Continue to keep the system patches current, and the Virus protection current.
4. Limit the amount of time that the machine is connected to the Internet. As an example download e-mail, disconnect, read, compose replies, re-connect and reply, then disconnect, as soon as possible.

1.1.10 Multiple Choice Question:

Choose the most correct statement from the four below:

a) A properly configured filtering router is all I need to protect my network.
b) A properly configured state aware Firewall is all I need to protect my network.
c) I need the most appropriate tools, to my business environment which I can afford, to protect my network.
d) None of the above.

c) is the correct answer

Assignment 1 - Network Detects

1.2 Detect 2 – Collateral Damage

1.2.1 Source of Trace: The trace was graciously provided by a company where I was a Network Engineer for, 7 years. This trace and the analysis following demonstrate the effect of “collateral damage” . The IP addresses have been disguised where appropriate to *class.b*. The site was receiving ICMP Protocol Unreachable messages to that was never sent out from their site, but instead TCP traffic sent by a third party, to IP address 24.66.45.1, which generated the ICMP Protocol Unreachable Messages, and returned them to the “spoofed” (*class.b)* IP address. The ISP was provided with the full trace information.

Background: The following detects have been captured by a Network General Distributed Sniffer System (DSS), and have been analyzed using Etherpeek 4.0.2 for this exercise. The Sniffer was set up with a special filter, and placed on the network connecting the network referred to as “class.b” to the Internet Service Provider’ s (ISP’s) router.

```
Packet   Source Logical   Dest.Logical      Size     Time-Stamp        Protocol        Plug-in Info
       1 IP-24.66.45.1    IP-class.b.81.9            94 00:38.11.661509  ICMP DUnr       Protocol unreachable 24.66.45.1
       2 IP-24.66.45.1    IP-class.b.81.9            94 00:38.11.664446  ICMP DUnr       Protocol unreachable 24.66.45.1
       3 IP-24.66.45.1    IP-class.b.81.9            94 00:38.11.665029  ICMP DUnr       Protocol unreachable 24.66.45.1
       4 IP-24.66.45.1    IP-class.b.81.9            94 00:38.11.668027  ICMP DUnr       Protocol unreachable 24.66.45.1
       5 IP-24.66.45.1    IP-class.b.81.9            94 00:38.11.747702  ICMP DUnr       Protocol unreachable 24.66.45.1
       6 IP-24.66.45.1    IP-class.b.81.9            94 00:38.11.750717  ICMP DUnr       Protocol unreachable 24.66.45.1
```

..this continues for a total of 50 packets and goes quiet until 02:26 …. the destination port changes from 99 to 98 as in the packets above

```
      56 IP-24.66.45.1    IP-class.b.104.47          94 02:26:09:665094  ICMP DUnr       Protocol unreachable 24.66.45.1
      57 IP-24.66.45.1    IP-class.b.104.47          94 02:26:09:665760  ICMP DUnr       Protocol unreachable 24.66.45.1
      58 IP-24.66.45.1    IP-class.b.104.47          94 02:26:09:668678  ICMP DUnr       Protocol unreachable 24.66.45.1
      59 IP-24.66.45.1    IP-class.b.104.47          94 02:26:09:669265  ICMP DUnr       Protocol unreachable 24.66.45.1
      60 IP-24.66.45.1    IP-class.b.104.47          94 02:26:09:756945  ICMP DUnr       Protocol unreachable 24.66.45.1
```

……this continues for a total of 50 packets and goes quiet until 03:45 ….

```
110 IP-24.66.45.1    IP-class.b.206.17     94 03:45:12:727758   ICMP DUnr    Protocol unreachable 24.66.45.1
                     Destination Port 50
111 IP-24.66.45.1    IP-class.b.147.64     94 03:45:39:456724   ICMP DUnr    Protocol unreachable 24.66.45.1
                     Destination Port 30
112 IP-24.66.45.1    IP-class.b.79.78      94 04:01:13:644314   ICMP DUnr    Protocol unreachable 24.66.45.1
                     Destination Port 36
113 IP-24.66.45.1    IP-class.b.55.148     94 04:19:20:330331   ICMP DUnr    Protocol unreachable 24.66.45.1
                     Destination Port 88
114 IP-24.66.45.1    IP-class.b.111.157    94 05:33:56:199385   ICMP DUnr    Protocol unreachable 24.66.45.1
                     Destination Port
                     107
```

……this continues with the address *class.b.111.157* for 49, more packets (total of 50), changes addresses, changes ports in what appear to be random order , but once the group of 50 similar packets start, the port and address remain the same. This continues at 18:41 for another 50 packets, with another address and port from *class.b*….. the whole pattern of activity finally ends the next day at 06:31

Analysis and tracking information inserted by Etherpeek before the Ethernet header.

```
  Flags:         0x00
  Status:        0x01
  Packet Length:94
  Timestamp:     00:38:11.661509 10/23/1999
```

Start of actual Ethernet Packet

```
Ethernet Header
  Destination:   00:D0:BA:D9:DC:21  [0-5]
  Source:        08:00:20:A7:73:AF  [6-11]
  Protocol Type:0x0800   IP  [12-13]
IP Header - Internet Protocol Datagram
  Version:               4  [14 Mask 0xF0]
  Header Length:         5  (20  bytes)  [14 Mask 0x0F]
  Type of Service:       %00000000  [15]
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:          76  [16-17]
  Identifier:            45187  [18-19]
  Fragmentation Flags:   %000  May Fragment   Last Fragment  [20 Mask 0xE0]
  Fragment Offset:       0  (0  bytes)  [20-21 Mask 0x1FFF]
  Time To Live:          50  [22]
  Protocol:              1  ICMP  [23]
  Header Checksum:       0xB398  [24-25]
  Source IP Address:     24.66.45.1  [26-29]
  Dest. IP Address:      class.b.81.9  [30-33]
  No IP Options
```

```
ICMP - Internet Control Messages Protocol
  ICMP Type:              3  Destination Unreachable  [34]
  Code:                   2  Protocol Unreachable  [35]
  Checksum:               0xBF68  [36-37]
  Unused (must be zero):0x00000000  [38-41]

Header of packet that caused error follows.
IP Header - Internet Protocol Datagram
  Version:                4  [42 Mask 0xF0]
  Header Length:          5  (20  bytes)  [42 Mask 0x0F]
  Type of Service:        %00000000  [43]
  Precedence: Routine,  Normal Delay,   Normal Throughput,  Normal Reliability
  Total Length:           40  [44-45]
  Identifier:             26883  [46-47]
  Fragmentation Flags:    %010  Do Not Fragment   Last Fragment  [48 Mask 0xE0]
  Fragment Offset:        0  (0  bytes)  [48-49 Mask 0x1FFF]
  Time To Live:           244  [50]
  Protocol:               6  TCP  [51]
  Header Checksum:        0xFA36  [52-53]
  Source IP Address:      class.b.81.9  [54-57]
  Dest. IP Address:       24.66.45.1  [58-61]
  No IP Options
TCP - Transport Control Protocol
  Source Port:        12467  [62-63]
  Destination Port:   99  Metagram Relay  [64-65]
  Sequence Number:    0  [66-69]
  Ack Number:         1380012832  [70-73]
  Offset:             0  [74 Mask 0xF0]
  Reserved:           %000001  [74-75 Mask 0x0FC0]
  Code:               %111011  [75 Mask 0x3F]
            Urgent is valid
            Ack is valid
            Push Request
            Synch Sequence
            FIN (Sender End of Byte Stream)
  Window:             44752  [76-77]
  Checksum:           0xF7CF  [78-79]
  Urgent Pointer:     48641  [80-81]
  No TCP Options
  TCP Data Area:      No more data.
Extra bytes (Padding):
  ........            01 00 00 00 00 00 00 00  [82-89]
Frame Check Sequence:  0x00000000  [90-93]
```

1.2.2 Detect was generated by:

This trace was captured by a Network General Sniffer version 3.5.4 (now Network Associates). The Sniffer was located outside the protected network's screening router, on the link connecting the screening router to the Internet Service Provider's (ISP's) router. There were only the 3 nodes physically present, on that leg of the network, the two routers and the one Sniffer interface. That piece of the network is in a controlled access environment. The Sniffer had been installed with a filter looking specifically for IP packets with both the SYN and FIN flags set. This was a proof of concept exercise, therefore, the filter mechanism was designed to be relatively simple. It identifies a packet as being Protocol Type, IP. Then looks in the packet 33 bytes (20 bytes, IP datagram with no options and 13 bytes for TCP, from the front of the IP header). It then masks where the SYN and FIN bits are (in the 33rd byte), or would normally be in the case of ICMP, or UDP packets. If the bits are set then the packet is recorded. It will catch any IP packet that has those bits set, regardless of protocol. This is a small source of false positives. In this case the ICMP packet is captured because the position of the bits required to be set, in the filter, turn out to be in the identifier field in the original crafted TCP packet that caused the ICMP packet to be generated. It is a decimal value of 26883, Hexadecimal 6903, the 3 being important. This is in the normal position for the TCP SYN & FIN flags. The other packets captured as a result of the activity all have this same feature. This indicates that many other crafted packets used in this same attack on 24.66.45.1 were not captured. It may partially explain why the activity was not recorded by the Sniffer, during the day.

1.2.3 Probability the Source Address was Spoofed:

For the ICMP Protocol Unreachable traffic, the Source address was not Spoofed. The ICMP Protocol Unreachable messages were a result of the original TCP packets, crafted by a 3rd party this is the packet with the spoofed source IP address (*class.b.x.x* network). In all cases the addresses were not in use at that time, and were not routed from the inside of the network. If a packet from the inside of the network were generated with the addresses, as used by the 3rd party, there would not be a route to the Internet, and therefore would not escape the network named *class.b*.

1.2.4 Attack Mechanism:

There are three possibilities here:

One 24.66.45.1 is actually trying to create a DoS attack against *class.b*.
Second, that 24.66.45.1 is being attacked by a 3rd party.
Third, a 3rd party is trying to create a DoS against *class.b* using 24.66.45.1 to do so, and in the process hiding his identity.

It is unlikely that 24.66.45.1 is the attacker. It would require much extra effort on the behalf of 24.66.45.1 to create the crafted ICMP packets, and would make finding that host far to easy for *class.b.* It is not impossible, just less likely than the second case.

In the second possibility, the intent of the attack against 24.66.45.1 by the 3rd party is unclear, due to the lack of a complete packet capture in the ICMP Protocol Unreachable message. ICMP packets contain only the first 64 bytes of the message that forced the creation of the ICMP message, this is intended to allow diagnosis of why the ICMP message was created, the destination computer does not support the protocol requested by the 3rd party, therefore the Protocol unreachable message. In this case one has to assume that the 3rd party wanted to create a DoS attack against 24.66.45.1, or to temporarily disable that machine as there is an invalid combination of flags *(*Urgent, Ack, Push Request, Synch Sequence, FIN) set in the crafted TCP packet. The 3rd party did not want to complete the 3 way handshake in this case, and therefore chose to spoof the address. The 3rd party may also have been sending other crafted packets at 24.66.45.1, at the same time just using addresses from another address space than *class.b.* This cannot be determined from this trace as packets with other destination addresses, out side of the range assigned to *class.b* would not have returned to this detector.

In the third case, *class.b* is suffering the effects, of a partial DoS, it is taking place at non business hours, and has no real effect. The 3rd party is effective in disguising his identity, but ineffective in the DoS. For this case to be taken seriously, other machines would have had to have been involved in the attack, and it would have been a Distributed Denial of Service (DDoS), or coordinated DoS.

1.2.5 Description of Attack:

The network where the detector was placed is a class b network of publicly valid IP addresses. A large number of the addresses were not used at the time that this trace was taken. A 3rd party took advantage of this availability of unused address space to “steal” some of the address for his own needs. The packets were then crafted and sent in groups of 50 to 24.66.45.1 It is alive, and it rejects the message, with a protocol unreachable message, it is this message which is returned to *class.b* . The Internet connection to *class.b* had sufficient bandwidth to tolerate this traffic, otherwise it would have been a Denial of Service (DoS) attack against *class.b.*

1.2.6 Correlation’s:

There was a slight increase in network traffic at *class.b* during the time the ICMP packets were being sent. This was only apparent after the discovery of the Sniffer traces, but would not have been noticed or investigated otherwise. The ISP was notified.

1.2.7 Evidence of Active Targeting:

The network *class.b* was obviously chosen as the one to “borrow” the addresses from for the attack against 24.66.45.1.

1.2.8 Severity:

Target Criticality = 0 (System doesn’t exist)
Attack Lethality = 1 (limited congestion caused by attack)
System Countermeasures = 0 (System doesn’t exist)
Network Countermeasures = 4 (Rudimentary Intrusion Detection, proof of concept exercise + Filtering Router + Firewall)

Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)
Severity = (0 + 1) - (0 + 4)

The Severity in this case for *class.b,* is a –3.

1.2.9 Defense Recommendation:

There is no fool proof opportunity today to directly prevent spoofing of IP addresses other than requiring each and every network connecting to the Internet to implement anti-spoofing egress rules in their routers. This was already implemented at *class.b*

A limited defense against this type of activity could be to choose to make all of the addresses appear to be in use, to the Internet, by implementing a Honeypot. This would only be effective in situations where the 3rd party wanted to use an address, which would not respond. It would have limited effect in this case, as the ICMP protocol has been designed such that the machine receiving the ICMP Protocol Unreachable message will not generate a response and we cannot determine how the *class.b*. network was evaluated to determine which addresses to choose to use.

A related concern is that the 3rd party may have been successful in mapping *class.b’s* IP address range at some time in the past. The only way to solve this would be to completely change *class.b’s* IP address assignment, a huge effort. Instead it was recommended to implement a more fully featured Intrusion Detection System (IDS) based on the information gathered during the proof of concept exercise.

1.2.10 Multiple Choice Question:

Select the most correct answer from the four below

a) ICMP is the Internet Control Message Protocol
b) ICMP is used to communicate error messages between machines.
c) There are 14 ICMP message type codes.
d) a) and b) only
e) a), b) and c)

d) is the correct answer

Assignment 1 - Network Detects

1.3 Detect 3 – IMAP Scan

1.3.1 Source of Trace: The trace was graciously provided by a company where I was a Network Engineer for, 7 years. This trace and the analysis following demonstrate an attacker testing IMAP vulnerabilities . The IP addresses have been disguised where appropriate to *class.b*.

Background: The following detects have been captured by a Network General Distributed Sniffer System (DSS), and have been analyzed using Etherpeek 4.0.2 for this exercise. The Sniffer was set up with a special filter, and placed on the network connecting the network referred to as "class.b" to the Internet Service Provider' s (ISP's) router.

```
Packet      Source Logical    Dest. Logical       Size   Time-Stamp   Protocol     Etherpeek Plug-in Information
          1 IP-212.216.13.8   IP-class.b.219.255     64         09:31.8 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          2 IP-212.216.13.8   IP-class.b.220.0       64         09:31.9 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          3 IP-212.216.13.8   IP-class.b.220.1       64         09:31.9 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          4 IP-212.216.13.8   IP-class.b.220.2       64         09:31.9 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          5 IP-212.216.13.8   IP-class.b.220.3       64         09:31.9 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
Continues testing each node consecutively until it reaches
          6 IP-212.216.13.8   IP-class.b.255.251     64         11:04.0 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          7 IP-212.216.13.8   IP-class.b.255.252     64         11:04.0 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          8 IP-212.216.13.8   IP-class.b.255.253     64         11:04.0 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
          9 IP-212.216.13.8   IP-class.b.255.254     64         11:04.0 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
         10 IP-212.216.13.8   IP-class.b.255.255     64         11:04.0 TCP IMAP   S= 141885440,L=   0,A=       0,W=  512
```

Analysis information inserted by Etherpeek before the Ethernet header.

```
  Flags:         0x00
  Status:        0x01
  Packet Length:64
  Timestamp:     12:09:31.847713 09/08/1999
```

Start of actual Ethernet Packet

```
Ethernet Header
  Destination:   08:00:20:A7:73:AF   [0-5]
  Source:        00:D0:BA:D9:DC:21   [6-11]
  Protocol Type:0x0800   IP  [12-13]
IP Header - Internet Protocol Datagram
  Version:                4   [14 Mask 0xF0]
  Header Length:          5   (20  bytes)   [14 Mask 0x0F]
  Type of Service:        %00000000   [15]
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:           40   [16-17]
  Identifier:             23042   [18-19]
  Fragmentation Flags:  %000  May Fragment   Last Fragment  [20 Mask 0xE0]
  Fragment Offset:      0  (0  bytes)  [20-21 Mask 0x1FFF]
  Time To Live:         232  [22]
  Protocol:             6  TCP  [23]
  Header Checksum:      0x2CA4  [24-25]
  Source IP Address:    212.216.13.8  [26-29]
  Dest. IP Address:     class.b.219.255  [30-33]
  No IP Options
TCP - Transport Control Protocol
  Source Port:        0  Reserved  [34-35]
  Destination Port: 143  IMAP -  Internet Message Access  [36-37]
  Sequence Number:  141885440  [38-41]
  Ack Number:         0  [42-45]
  Offset:             5  [46 Mask 0xF0]
  Reserved:           %000000  [46-47 Mask 0x0FC0]
  Code:               %000011  [47 Mask 0x3F]
          Synch Sequence
          FIN (Sender End of Byte Stream)
  Window:             512  [48-49]
  Checksum:           0x58B4  [50-51]
  Urgent Pointer:     0  [52-53]
  No TCP Options
  TCP Data Area:      No more data.
Extra bytes (Padding):
  h.~~.F              68 12 7E 7E 08 46  [54-59]
Frame Check Sequence:  0x00000000  [60-63]
```

This activity continues counting upwards through the addresses until it reaches the top of *class.b*

Analysis information inserted by Etherpeek before the Ethernet header.

```
Flags:         0x00
  Status:        0x01
  Packet Length:64
  Timestamp:     12:11:04.031875 09/08/1999
```

Start of actual Ethernet Packet

```
Ethernet Header
  Destination:   08:00:20:A7:73:AF  [0-5]
  Source:        00:D0:BA:D9:DC:21  [6-11]
  Protocol Type:0x0800  IP  [12-13]
IP Header - Internet Protocol Datagram
  Version:              4  [14 Mask 0xF0]
  Header Length:        5  (20  bytes)  [14 Mask 0x0F]
  Type of Service:      %00000000  [15]
  Precedence: Routine,  Normal Delay,  Normal Throughput,  Normal Reliability
  Total Length:         40  [16-17]
  Identifier:           49154  [18-19]
  Fragmentation Flags:  %000  May Fragment   Last Fragment  [20 Mask 0xE0]
  Fragment Offset:      0  (0  bytes)  [20-21 Mask 0x1FFF]
  Time To Live:         232  [22]
  Protocol:             6  TCP  [23]
  Header Checksum:      0xA2A3  [24-25]
  Source IP Address:    212.216.13.8  [26-29]
  Dest. IP Address:     class.b.255.255  [30-33]
  No IP Options
TCP - Transport Control Protocol
  Source Port:          0  Reserved  [34-35]
  Destination Port: 143  IMAP -  Internet Message Access  [36-37]
  Sequence Number:      141885440  [38-41]
  Ack Number:           0  [42-45]
  Offset:               5  [46 Mask 0xF0]
  Reserved:             %000000  [46-47 Mask 0x0FC0]
  Code:                 %000011  [47 Mask 0x3F]
            Synch Sequence
            FIN (Sender End of Byte Stream)
  Window:               512  [48-49]
  Checksum:             0x34B4  [50-51]
  Urgent Pointer:       0  [52-53]
  No TCP Options
  TCP Data Area:        No more data.
Extra bytes (Padding):
  ^Ã~~á.                5E C3 7E 7E E1 9D  [54-59]
Frame Check Sequence:   0x00000000  [60-63]
```

1.3.2 Detect was generated by:

This trace was captured by a Network General Sniffer version 3.5.4 (now Network Associates). The Sniffer was located outside the protected network's screening router, on the link connecting the screening router to the Internet Service Provider's (ISP's) router. There were only the 3 nodes physically present, on that leg of the network, the two routers and the one Sniffer interface. That piece of the network is in a controlled access environment. The Sniffer had been installed with a filter looking specifically for IP packets with both the SYN and FIN flags set. This was a proof of concept exercise, therefore, the filter mechanism was designed to be relatively simple. It identifies a packet as being Protocol Type, IP. Then looks in the packet 33 bytes (20 bytes, IP datagram with no options and 13 bytes for TCP, from the front of the IP header). It then masks where the SYN and FIN bits are (in the 33rd byte), or would normally be in the case of ICMP, or UDP packets. If the bits are set then the packet is recorded. It will catch any IP packet that has those bits set, regardless of protocol. This is a small source of false positives. In this case the packet is captured because it has the TCP SYN & FIN flags set in the packet.

1.3.3 Probability the Source Address was Spoofed:

The probability that the source address was spoofed in this case is very low. The individual was performing a mapping operation, and would have been expecting a response. If the source address was spoofed, the attacker would have had to intercept the return packets along the way, in order for this effort to be effective.

```
inetnum:     212.216.0.0 - 212.216.31.255
netname:     TIN
descr:       Telecom Italia Net
descr:       PROVIDER, TIN elite customers in OSPF Area 01
```

person: Enzo Berti
address: Via Val Cannuta, 182
phone: +39 06 36888592
fax-no: +39 06 36889863
e-mail: e.berti@tin.it

1.3.4 Attack Mechanism:

The attacker chose Internet Message Access Protocol (IMAP) as the protocol due to vulnerabilities as noted in CERT* Summary CS-97.06 "The impact of an IMAP attack is that the remote user (e.g., intruder) will be able to gain root-level access on a vulnerable host" http://www.cert.org/summaries/CS-97.06.html and CERT® Advisory CA-1998-09 Buffer Overflow in Some Implementations of IMAP Servers http://www.cert.org/advisories/CA-1998-09.html discusses specific issues with the IMAP protocol. The attack is also using port 0, with SYN and FIN, this is a common IMAP exploit.

1.3.5 Description of Attack:

The network where the detector was placed is a class b network of publicly valid IP addresses. A large number of the addresses were not used at the time that this trace was taken. The group of addresses that the attacker was evaluating was not used, there would not have been a response from the end node as it would not be there to respond. The router interfacing *class.b* to the Internet had been configured not to respond to unreachables, therefore the attacker would not have received any response for this string of attacks. The intruder chose IMAP and had the SYN and FIN flags set as an attempt to evade screening routers and Firewalls which do not maintain state. The Three Way Handshake works as follows, a node (client) wishing to communicate with another node (server) establishes the TCP connection with a TCP packet containing a SYN, the initial sequence number it wishes to use, to the well known port it wishes to connect to, the Maximum Segment Size (MSS) and Maximum Transmission Unit (MTU) size. If the server is willing and able to establish a connection, it responds with a packet, which contains a SYN-ACK, increments the client's initial sequence number by 1, supplies the server's initial sequence number, MSS and MTU. If that port is not active on the server, a reset is sent instead to the client node. The client responds to the server's SYN with an ACK, and increments the server's initial sequence number by 1. The initial sequence numbers are incremented as appropriate to each node, with each transfer of data. Communication then continues until the session is complete. The start of session tear down is initiated with a FIN from, which ever node has completed first. The other node responds with an ACK, and if it has completed its portion of the session, as well, it responds with a FIN, if it has not completed its side of the communication it continues until it is complete. The node that has sent its FIN will continue to respond even though it may already have sent a FIN, until both ends of the conversation have sent a FIN, to which the other node responds with an ACK.. As can be observed, the combination of SYN and FIN do not naturally occur together at the same time, in a normal session.

1.3.6 Correlation's:

In this case there was no other correlation. The traffic generated by this intruder was so low that it was lost in the other traffic. This was a single event from this address to a block of unused addresses, therefore no other action was taken.

1.3.7 Evidence of Active Targeting:

The intruder targeted a block of unused addresses, therefore received no response. This was the only group of attacks seen from that group of addresses with this set of features. The attack was considered unsuccessful.

1.3.8 Severity:

Target Criticality = 0 (System doesn't exist)
Attack Lethality = 0 (can be Lethal, to IMAP servers, not used/ shutdown at this site)
System Countermeasures = 0 (System doesn't exist)
Network Countermeasures = 4 (Rudimentary Intrusion Detection, proof of concept exercise + Filtering Router + Firewall)
(0 + 0) - (0 + 4)
Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

The Severity in this case is a –4.

1.3.9 Defense Recommendation:

The concern with this attack is that the intruder could have started to develop a network map for the site. The solution here is to keep up to date with all of the recommended patches, and keep the firewalls, and routers as quiet as possible about reachable address and port. The attack appears to have been crafted by a "script-kiddy", it was poorly directed, at the target, and very little information was obtained.

1.3.10 Multiple Choice Question:

Select the most correct answer from the four below

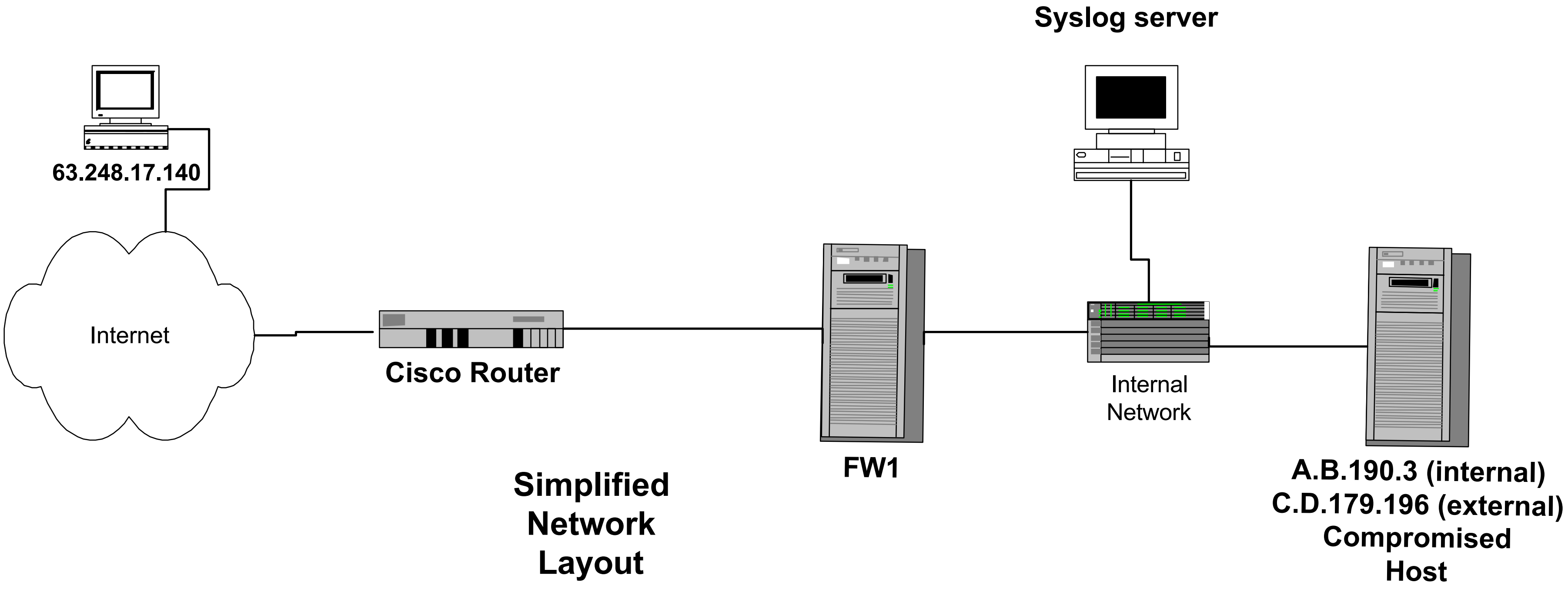
a) IMAP uses UDP as the transport protocol
b) IMAP uses TCP as the transport protocol
c) IMAP connects to the server on port 143
d) both b) and c)

d) is the most correct answer

Assignment 1 - Network Detects

1.4 Detect 4 – Successful Intrusion

The data below has been sanitized. A.B.190.3 and C.D 179.196 are not the real addresses of the machines, the Firewall data has been organized in a more readable fashion. The Syslog (Cisco Router log) information has also been sanitized, and reorganized.



The Firewall performs address translation as well as performing the Firewall function. The first line of information shows the following:
The Original Source Address A.B.190.3 is translated to the destination address C.D.179.196, this translation remains the same throughout, it is bi-directional in this case, and the Cisco logs refer to the machine as C.D.179.196, with the associated port pairings.
The Original Destination Address does not undergo translation
The Original Source port is translated, in the first line, from 3760 to 44527, the source port and Translated source port change in the next line.
The Original Destination Port does not undergo translation

Information from Firewall Logs

```
 Time     Status  Device  Interface   proto   Orig Source    Orig Dest Addr    Orig Dest Port  Orig Source Port  Length  Rule   Translated Source Addr    Translated Dest Addr   Translated Source Port  Translated Dest Port
0:44:29 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3760 len 44  rule 7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44527 xlatedport 80
0:44:30 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3761 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44528 xlatedport 80
```

```
0:44:31 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3762 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44529 xlatedport 80
0:44:31 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3763 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44530 xlatedport 80
0:44:31 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3764 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44531 xlatedport 80
0:44:31 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3765 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44532 xlatedport 80
0:44:31 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3766 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44533 xlatedport 80
0:44:32 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3767 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44534 xlatedport 80
0:44:33 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3768 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44535 xlatedport 80
0:44:44 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3769 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44536 xlatedport 80
0:45:33 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3795 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 44562 xlatedport 80

3:25:23 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1972 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46195 xlatedport 80
3:25:24 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1973 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46196 xlatedport 80
3:25:24 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1974 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46197 xlatedport 80
3:25:25 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1975 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46198 xlatedport 80
3:25:25 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1976 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46199 xlatedport 80
3:25:25 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1977 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46200 xlatedport 80
3:25:25 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1978 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46201 xlatedport 80
3:25:25 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1979 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46202 xlatedport 80
3:25:25 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1980 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46203 xlatedport 80
3:25:27 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 1981 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 46204 xlatedport 80

6:24:48 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3507 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47344 xlatedport 80
6:24:49 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3508 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47345 xlatedport 80
6:24:49 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3509 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47346 xlatedport 80
6:24:49 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3510 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47347 xlatedport 80
6:24:50 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3511 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47348 xlatedport 80
6:24:50 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3512 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47349 xlatedport 80
6:24:50 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3513 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47350 xlatedport 80
6:24:50 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3514 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47351 xlatedport 80
6:24:50 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3515 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47352 xlatedport 80
6:24:53 accept FW1 >lan2     tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3516 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 47353 xlatedport 80

19:48:16 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3446 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51385 xlatedport 80
19:48:17 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3447 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51386 xlatedport 80
19:48:17 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3448 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51387 xlatedport 80
19:48:18 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3449 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51388 xlatedport 80
19:48:18 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3450 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51389 xlatedport 80
19:48:18 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3451 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51390 xlatedport 80
19:48:18 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3452 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51391 xlatedport 80
19:48:18 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3453 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51392 xlatedport 80
19:48:19 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3454 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51393 xlatedport 80
19:48:20 accept FW1 >lan2    tcp src  A.B.190.3 dst 63.248.17.140  d_port 80  s_port 3455 len 44 rule  7 xlatesrc C.D.179.196  xlatedst 63.248.17.140 xlatesport 51394 xlatedport 80
```

The Logs from the Cisco Router

```
Date and Time                              Access List  Status     proto  Source address and       Destination address and  Number of
                                                                          port                     port                     Packets
Oct  19  0:44:29 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44527),     1 packet
Oct  19  0:44:30 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44528),     1 packet
Oct  19  0:44:31 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44529),     1 packet
Oct  19  0:44:31 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44530),     1 packet
Oct  19  0:44:31 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44531),     1 packet
Oct  19  0:44:31 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44532),     1 packet
Oct  19  0:44:31 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44533),     1 packet
Oct  19  0:44:32 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44534),     1 packet
Oct  19  0:44:33 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44535),     1 packet
Oct  19  0:44:43 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44536),     1 packet
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44527),     4 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44528),     3 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44529),     3 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44530),     3 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44531),     3 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44532),     3 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44534),     3 packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44533),    20  packets
Oct  19  0:44:48 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44535),     3 packets
Oct  19  0:44:49 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44536),     3 packets
Oct  19  0:45:33 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44533),     1 packet
Oct  19  0:45:33 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44562),     1 packet
Oct  19  0:46:21 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44533),     1 packet
Oct  19  0:46:21 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(44562),     3 packets

Oct  19  3:25:23 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46195),     1 packet
Oct  19  3:25:24 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46196),     1 packet
Oct  19  3:25:24 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46197),     1 packet
Oct  19  3:25:24 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46198),     1 packet
Oct  19  3:25:24 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46199),     1 packet
Oct  19  3:25:24 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46200),     1 packet
Oct  19  3:25:25 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46201),     1 packet
Oct  19  3:25:25 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46202),     1 packet
Oct  19  3:25:25 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46203),     1 packet
Oct  19  3:25:27 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46204),     1 packet
Oct  19  3:25:32 %SEC-6-IPACCESSLOGP:  list      103 permitted   tcp   63.248.17.140(80)  ->  C.D.179.196(46195),     3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46196),  3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46197),  3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46198),  3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46199),  3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46200),  3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46201),  3 packets
Oct  19   3:25:32 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46202),  3 packets
Oct  19   3:25:33 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46203),  4 packets
Oct  19   3:25:33 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(46204),  3 packets

Oct  19   6:24:48 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47344),  1 packet
Oct  19   6:24:49 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47345),  1 packet
Oct  19   6:24:49 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47346),  1 packet
Oct  19   6:24:49 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47347),  1 packet
Oct  19   6:24:49 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47348),  1 packet
Oct  19   6:24:50 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47349),  1 packet
Oct  19   6:24:50 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47350),  1 packet
Oct  19   6:24:50 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47351),  1 packet
Oct  19   6:24:50 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47352),  1 packet
Oct  19   6:24:52 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47353),  1 packet
Oct  19   6:24:56 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47344),  3 packets
Oct  19   6:24:57 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47345),  3 packets
Oct  19   6:24:57 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47346),  3 packets
Oct  19   6:24:57 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47347),  3 packets
Oct  19   6:24:57 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47348),  3 packets
Oct  19   6:24:57 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47349),  3 packets
Oct  19   6:24:57 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47350),  3 packets
Oct  19   6:24:58 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47351),  3 packets
Oct  19   6:24:58 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47352),  4 packets
Oct  19   6:24:58 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(47353),  3 packets

Oct  19  19:48:17 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51386),  1 packet
Oct  19  19:48:17 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51387),  1 packet
Oct  19  19:48:18 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51388),  1 packet
Oct  19  19:48:18 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51389),  1 packet
Oct  19  19:48:18 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51390),  1 packet
Oct  19  19:48:18 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51391),  1 packet
Oct  19  19:48:18 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51392),  1 packet
Oct  19  19:48:19 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51393),  1 packet
Oct  19  19:48:20 %SEC-6-IPACCESSLOGP:  list  103 permitted  tcp  63.248.17.140(80)  ->  C.D.179.196(51394),  1 packet
Oct   19   19:48:28 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51385),       3 packets
Oct   19   19:48:28 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51386),       3 packets
Oct   19   19:48:28 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51387),       3 packets
Oct   19   19:48:28 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51388),       3 packets
Oct   19   19:48:29 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51389),       3 packets
Oct   19   19:48:29 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51390),       3 packets
Oct   19   19:48:29 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51391),       3 packets
Oct   19   19:48:29 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51392),       4 packets
Oct   19   19:48:29 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51393),       3 packets
Oct   19   19:48:29 %SEC-6-IPACCESSLOGP:   list      103 permitted    tcp    63.248.17.140(80)   ->   C.D.179.196(51394),       3 packets
```

```
Addresses:63.248.17.140 Canonical name: 3ff8118c.dsl.flashcom.net

Flashcom, Inc. (NETBLK-NETBLK-FLASHCOM-2)
Huntington Beach, CA 92649 US
  Netname: NETBLK-FLASHCOM-2
  Netblock: 63.248.0.0 - 63.248.255.255
  Coordinator: Benton, Curtis  (CB373-ARIN)  curtisb@flashcom.com
    (714) 891-7891
```

1.4.1 Source of Trace:

The traces above were collected from a Firewall-1 Version 4.1 SP3 and from a Cisco Router running version11.3. The information was provided by an associate of mine, with the full knowledge, and very helpful co-operation of the system owners.

1.4.2 Detect was generated by:

The actual detect was made by an alert System Administrator. The traces above are the supporting evidence discovered, once the initial compromise was recognized..

1.4.3 Probability the Source Address was Spoofed:

It is unlikely that the source address was spoofed. The protocol in use is TCP, in order for the system to be compromised, the spoof would require a "man in the middle" style of spoof for it to be successful. The intent of the attack was to use A.B.190.3 as an intermediate node to attack other machines, perhaps to install other Trojan Warez, to work in a Distributed Denial of Service.

1.4.4 Attack Mechanism:

The system runs Netscape proxy server version 3.53, and an Oracle Database. The attacker probably used a known compromise, any combination of the following could have been used:

Buffer Overflow in Netscape Enterprise and FastTrack Authentication Procedure http://xforce.iss.net/alerts/advise39.php

CVE-1999-0853 Buffer overflow in Netscape Enterprise Server and Netscape FastTrack Server allows remote attackers to gain privileges via the HTTP Basic Authentication procedure. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0853

A Stateful Inspection of FireWall-1 http://www.dataprotect.com/bh2000/blackhat-fw1.txt This paper discusses vulnerabilities of Firewall-1, and was presented at the Black Hat Briefings 2000 July 26th and 27th, Las Vegas

Oracle setting UTL_FILE_DIR is set to * allowing file I/O package to write anywhere http://xforce.iss.net/static/3547.php

Stale accounts provide a point of attack for unauthorized users http://xforce.iss.net/static/3428.php

1.4.5 Description of Attack:

The system, A.B.190.3 was compromised. The detect was made after the System Administrator noticed another machine was down, other than A.B.190.3, looked in the Syslogs and noticed that A.B.190.3 was originating traffic that was not part of the Proxy service on that machine. The traffic was not passthrough, it originated on the Proxy machine and was destined for a DSL line, communication that this machine does not normally do. After probing the DSL line it was discovered that the node at 63.248.17.140 was not a Web Server. The Root password was discovered to have been modified on A.B.190.3. It is undergoing a total rebuild, with current software, from original installation media, official vendor supplied patches, verification that known exposures have been closed and installation of Tripwire.

1.4.6 Correlation's:

The System Administrator made the call, the traces above are the correlating evidence.

1.4.7 Evidence of Active Targeting:

The target was well chosen, identified for its possibility of a compromise then the takeover was executed, the net result a compromised system.

1.4.8 Severity:

Target Criticality = 4 (Provides Proxy services)
Attack lethality = 5 (It was successful)
System Countermeasures = 1 (System not kept at current approved levels, of software)
Network Countermeasures = 3 (screening router + logging + firewall + logging)

Severity = (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures)

(4 + 5) – (1 + 3)

Severity = 5

1.4.9 Defense Recommendation:

In order to be successful in this case, in addition what has already been done, it requires that:

- the software installed on the machine is kept up to date,
- limited services are installed, or enabled,
- the system frequently checked for new/known vulnerabilities,
- if at all possible the access rules tightened down further, on both the screening router and Firewall

1.4.10 Multiple Choice Question:

Please select the most correct statement from the ones below

a) A screening router, firewall and up to date secured system will always prevent an attack from being successful.
b) A screening router will always prevent an attack from being successful.
c) A firewall will always prevent an attack from being successful.
d) A screening router, firewall and up to date secured system will only buy time in the face of a determined attack.

d) is the most correct answer

2.0 Assignment 2 – Evaluate an Attack – DumpSec by Somarsoft, formerly DumpAcl

This demonstration is offered in the hope of helping those who might have any questions about port 139 being open to the Internet, understand the exposure to their machines. The product DumpSec version 2.8.1 was downloaded from http://www.somarsoft.com/ it is advertised as a security auditing program for Windows NT. It can be used to dump and display the permissions and audit settings for a system in an easily readable format. It is a newer, updated version of DumpAcl. This security evaluation tool was mentioned in the book *Hacking Exposed* , by Stuart McClure and Joel Scambray. It is also demonstrated in the SANS course “*Contemporary Hacking Tools and Their use in Penetration Testing.”*

2.1 Description:

For this demonstration, I installed the program on a Windows NT Workstation V4.0 Sp6a+ Hotfixes, and used it to evaluate a Toshiba Laptop, running Windows NT V4.0 Sp6a+ Hotfixes. Both of the machines are in the same subnet, same NT domain. They have been secured as noted in “*Windows Security Guidelines”*, by Trusted Systems Services http://www.trustedsystems.com where ever reasonably possible. The deviation from the recommendations is due to corporate requirements to allow the Administrators to administrate the machine. This evaluation was run against a single user machine, but could and would be run against a Primary Domain Controller, or the backup, in the case of an intrusion. All that is needed for this “evaluation” of the NT machine to function is port 139 to be open on the machine being evaluated.

2.2 Issue:

In this case I can down load a program, which is easy to use, and in less than 30 seconds, and 64 packets, I have a list of the users on a reasonably secure NT machine. The information output from the DumpSec program, allows me to know that for this machine I have 2 users to attack, (Guest has been disabled as indicated in the display) with Administrator being the one of choice. The information is all obtained using normal NT commands, there is nothing suspicious in the exchange of information between the two machines, as can be seen from the Packet Exchange and the display of packets 9, 23,27,39, and 51. 10.100.50.8 is running the DumpSec program, 10.100.50.87 is the Toshiba Laptop being evaluated.
The following is the output of DumpSec, that can be viewed on the program main window. Other NT options are available, to be viewed as selected in the configuration screen of the DumpSec program. I chose this group of items as it contains the minimum amount of useful information that an intruder would want to have. My User Name has been modified to Aaayyyyzzz, although it may not take long to guess what it might be.

```
11/6/00 1:43 PM - Somarsoft DumpSec (formerly DumpAcl) - \\pc872
UserName    FullName        AccountType PswdCanBeChanged PswdLastSetTime   PswdRequired PswdExpires PswdExpiresTime  AcctDisabled AcctLockedOut
Administrator                  User         Yes              3/15/00 10:42 PM   Yes           No          Never              No           No
Guest                          User         No               4/27/00 10:45 AM   Yes           Yes         6/26/00 10:45 AM   Yes          No
Aaayyyzz Oliver Viitamaki      User         Yes              10/10/00 12:54 PM  Yes           Yes         12/9/00 11:54 AM   No           No

UserName       AcctExpires   Time LastLogonTime   LastLogonServer LogonHours
 Administrator  Never          10/28/00 9:28 PM      pc872          All
 Guest          Never           Never                pc872          All
 Aaayyyzz       Never          11/4/00 6:15 PM       pc872          All
```

2.3 Packet Exchange

Packet #	Source IP Address	Source Port	Destination IP Address	Destination Port	Packet Size	Time	Protocol Information as Decoded by Etherpeek
1	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	162	13:43:26.337063	TCP NB SMB
2	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	165	13:43:26.337892	TCP NB SMB
3	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	218	13:43:26.338725	SMB NBIO
4	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	186	13:43:26.339344	SMB NBIO
5	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	206	13:43:26.340115	SMB NBIO
6	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	218	13:43:26.340613	SMB NBIO
7	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	104	13:43:26.341252	SMB CloF
8	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	97	13:43:26.341610	SMB CloF
9	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	64	13:43:26.462956	TCP NB SessMsg

To this point DumpSec has contacted the machine being analyzed. The contact by Pc Name is actually made in packet number 5, displayed below. Packets 10 through 64 are used for gathering all of the information displayed in the main program window.

Packet #	Source IP Address	Source Port	Destination IP Address	Destination Port	Packet Size	Time	Protocol Information as Decoded by Etherpeek
10	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	222	13:43:50.824266	SMB NBIO
11	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	166	13:43:50.824996	SMB NBIO
12	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	192	13:43:50.825840	SMB NBIO
13	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	178	13:43:50.826257	SMB NBIO
14	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	192	13:43:50.827030	SMB NBIO
15	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	162	13:43:50.827360	SMB NBIO
16	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	190	13:43:50.828112	SMB NBIO
17	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	166	13:43:50.828442	SMB NBIO
18	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	210	13:43:50.829723	SMB NBIO
19	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	166	13:43:50.847031	SMB NBIO
20	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	222	13:43:50.848048	SMB NBIO
21	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	166	13:43:50.848495	SMB NBIO
22	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	202	13:43:50.849309	SMB NBIO
23	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	298	13:43:50.850389	SMB NBIO

In packet 23 all of the available users on the machine are identified That packet is displayed below.

Packet #	Source IP Address	Source Port	Destination IP Address	Destination Port	Packet Size	Time	Protocol Information as Decoded by Etherpeek
24	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	198	13:43:50.851329	SMB NBIO
25	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	166	13:43:50.851810	SMB NBIO
26	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	192	13:43:50.852548	SMB NBIO
27	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	642	13:43:50.853592	SMB NBIO

In packet 27 the first of the administrator settings are sent to the machine running DumpSec. That packet is displayed below

Packet #	Source IP Address	Source Port	Destination IP Address	Destination Port	Packet Size	Time	Protocol Information as Decoded by Etherpeek
28	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	194	13:43:50.854863	SMB NBIO
29	IP-10.100.50.87	IP-139	IP-10.100.50.8	IP-2242	270	13:43:50.855311	SMB NBIO
30	IP-10.100.50.8	IP-2242	IP-10.100.50.87	IP-139	190	13:43:50.856184	SMB NBIO

```
31   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     170   13:43:50.856537    SMB NBIO
32   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      274   13:43:50.857470    SMB NBIO
33   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     162   13:43:50.857827    SMB NBIO
34   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      190   13:43:50.858715    SMB NBIO
35   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     166   13:43:50.859058    SMB NBIO
36   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      198   13:43:50.859838    SMB NBIO
37   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     166   13:43:50.860304    SMB NBIO
38   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      192   13:43:50.861053    SMB NBIO
39   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     630   13:43:50.861851    SMB NBIO
```

In packet 39 the first of the Guest settings are sent to the machine running DumpSec. That packet is displayed below

```
40   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      194   13:43:50.863086    SMB NBIO
41   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     258   13:43:50.863516    SMB NBIO
42   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      190   13:43:50.864365    SMB NBIO
43   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     170   13:43:50.864712    SMB NBIO
44   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      274   13:43:50.865624    SMB NBIO
45   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     158   13:43:50.865971    SMB NBIO
46   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      190   13:43:50.867261    SMB NBIO
47   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     166   13:43:50.867623    SMB NBIO
48   IP-10.100.50.8  IP-2242       IP-10.100.50.87  IP-139            198   13:43:50.868408    SMB NBIO
49   IP-10.100.50.87 IP-139        IP-10.100.50.8  IP-2242      166   13:43:50.868867    SMB NBIO
50   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      192   13:43:50.869626    SMB NBIO
51   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     686   13:43:50.870754    SMB NBIO
```

In packet 51, the first of O. Viitamaki settings are sent to the machine running DumpSec. That packet is displayed below.

```
52   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      194   13:43:50.872056    SMB NBIO
53   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     270   13:43:50.872505    SMB NBIO
54   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      190   13:43:50.873365    SMB NBIO
55   IP-10.100.50.87 IP-139        IP-10.100.50.8  IP-2242      170   13:43:50.873711    SMB NBIO
56   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      274   13:43:50.874624    SMB NBIO
57   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     174   13:43:50.874983    SMB NBIO
58   IP-10.100.50.8  IP-2242       IP-10.100.50.87  IP-139            190   13:43:50.875800    SMB NBIO
59   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     166   13:43:50.876143    SMB NBIO
60   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      190   13:43:50.877004    SMB NBIO
61   IP-10.100.50.87 IP-139        IP-10.100.50.8  IP-2242      166   13:43:50.877333    SMB NBIO
62   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139      190   13:43:50.878085    SMB NBIO
63   IP-10.100.50.87        IP-139        IP-10.100.50.8  IP-2242     166   13:43:50.878411    SMB NBIO
64   IP-10.100.50.8  IP-2242       IP-10.100.50.87        IP-139       64   13:43:50.996832    TCP NB SessMsg
```

Packet #5

```
  Flags:         0x00
  Status:        0x00
  Packet Length:206
  Timestamp:     13:43:26.340115 11/06/2000
```

Analysis and tracking information inserted by Etherpeek before the Ethernet header.

Start of actual packet

```
Ethernet Header
  Destination:   00:10:A4:ED:47:0D
  Source:        00:50:8B:67:9C:FB
  Protocol Type:0x0800  IP
IP Header - Internet Protocol Datagram
  Version:               4
  Header Length:         5  (20  bytes)
  Type of Service:       %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:          188
  Identifier:            51169
  Fragmentation Flags:   %010  Do Not Fragment   Last Fragment
  Fragment Offset:       0  (0  bytes)
  Time To Live:          128
  Protocol:              6  TCP
  Header Checksum:       0xB933
  Source IP Address:     10.100.50.8
  Dest. IP Address:      10.100.50.87
  No IP Options
TCP - Transport Control Protocol
  Source Port:       2242
  Destination Port:  139  NETBIOS Session Service
  Sequence Number:   1190518053
  Ack Number:        74370
  Offset:            5
  Reserved:          %000000
  Code:              %011000
            Ack is valid
            Push Request
  Window:            8525
  Checksum:          0xB532
  Urgent Pointer:    0
  No TCP Options
NetBIOS Session Service - Network Basic Input/Output System
  Packet Type:           0x00  Session Message
  Flags:                 0x00  Length Extension Off
  Length:                144
SMB - Server Message Block
  Protocol ID:           SMB
  Command Code:          37  Transaction - Name, Bytes In/Out
  Error Code Class:      0x00  Success
  Reserved:              0x00
  Error Code:             0   Success
  Flags:                  0x18
            Request
            Pathnames Are Without Case
            Pathnames Are Already In Canonicalized Format
  Flags2:                 0x8003
            Application Understands Long File Names
            Application Understands Extended Attributes
            Application Understands Unicode Strings
  Reserved:
  Ø,..........        D8 82 00 00 00 00 00 00 00 00 00 00
  Tree ID (TID):          0x0800
  Process ID (PID):       0x1180
  User ID (UID):          0x0800
  Multiplex ID (MID):     0x1A80
SMB Transaction - Name, Bytes In/OutRequest
  Word Count:             16
  Total Param Bytes:      0
  Total Data Bytes:       60
  Param Bytes To Recv:    0
  Data Bytes To Recv:     1024
  Setup Bytes To Recv:    0
  Reserved:               0x00
  Flags:                  0x0000
  Timeout (millisec.):    0
  Reserved:               0x0000
  Params This Buffer:     0
  Params Bytes Offset:    84
  Data This Buffer:       60
  Data Bytes Offset:      84
  Setup Word Count:       2
  Reserved:               0x00
  Additional Setup Bytes:
  &.   .                26 00 09 08
  Byte Count:             77
  File Pathname:

  Parameter And Data Bytes:
  .~........<.....    00 7E 05 00 00 03 10 00 00 00 3C 00 00 00 01 00
  ..$.......è.©...    00 00 24 00 00 00 00 00 15 00 E8 04 A9 00 08 00
  ..........\.\.p.    00 00 00 00 00 00 08 00 00 00 5C 00 5C 00 70 00
  c.8.7.2...e....     63 00 38 00 37 00 32 00 00 00 65 00 00 00 00
```

Packet #23

```
  Flags:        0x00
```

Analysis and tracking information inserted by Etherpeek before the Ethernet header.

```
  Status:        0x00
  Packet Length:298
  Timestamp:     13:43:50.850389 11/06/2000
```

Start of actual packet

```
Ethernet Header
  Destination:   00:50:8B:67:9C:FB
  Source:        00:10:A4:ED:47:0D
  Protocol Type:0x0800  IP
IP Header - Internet Protocol Datagram
  Version:               4
  Header Length:         5  (20  bytes)
  Type of Service:       %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:          280
  Identifier:            6712
  Fragmentation Flags:   %010  Do Not Fragment   Last Fragment
  Fragment Offset:       0  (0  bytes)
  Time To Live:          128
  Protocol:              6  TCP
  Header Checksum:       0x6681
  Source IP Address:     10.100.50.87
  Dest. IP Address:      10.100.50.8
  No IP Options
TCP - Transport Control Protocol
  Source Port:       139  NETBIOS Session Service
  Destination Port:  2242
  Sequence Number:   75225
  Ack Number:        1190519271
  Offset:            5
  Reserved:          %000000
  Code:              %011000
           Ack is valid
           Push Request
  Window:            7690
  Checksum:          0x9271
  Urgent Pointer:    0
  No TCP Options
NetBIOS Session Service - Network Basic Input/Output System
  Packet Type:           0x00  Session Message
  Flags:                 0x00  Length Extension Off
  Length:                236
SMB - Server Message Block
  Protocol ID:           SMB
  Command Code:          37  Transaction - Name, Bytes In/Out
  Error Code Class:      0x00  Success
  Reserved:              0x00
  Error Code:            0  Success
  Flags:                 0x98
```

```
            Response
            Pathnames Are Without Case
            Pathnames Are Already In Canonicalized Format
  Flags2:                 0x8003
            Application Understands Long File Names
            Application Understands Extended Attributes
            Application Understands Unicode Strings
  Reserved:
  Ø,.........         D8 82 00 00 00 00 00 00 00 00 00 00
  Tree ID (TID):          0x0800
  Process ID (PID):       0x1180
  User ID (UID):          0x0800
  Multiplex ID (MID):     0x1C80
SMB Transaction - Name, Bytes In/OutResponse
  Word Count:             10
  Total Param Bytes:      0
  Total Data Bytes:       180
  Reserved:               0x0000
  Params This Buffer:     0
  Params Bytes Offset:    56
  Params Displacement:    0
  Data This Buffer:       180
  Data Bytes Offset:      56
  Data Displacement:      0
  Setup Word Count:       0
  Reserved:               0x00
  Byte Count:             181
  Parameter And Data Bytes:
  8........´...>..    38 05 00 02 03 10 00 00 00 B4 00 00 00 3E 00 00
  .œ...........pp.    00 9C 00 00 00 00 00 00 00 03 00 00 00 70 70 14
  .....po......ô..    00 03 00 00 00 70 6F 14 00 03 00 00 00 F4 01 00
  ...  ..¨..õ.....    00 1A 00 20 00 10 A8 14 00 F5 01 00 00 0A 00 20
  .....è.....  .Ho.   00 10 00 14 00 E8 03 00 00 10 00 20 00 48 6F 14
  .............A.d    00 10 00 00 00 00 00 00 00 0D 00 00 00 41 00 64
  .m.i.n.i.s.t.r.a    00 6D 00 69 00 6E 00 69 00 73 00 74 00 72 00 61
  .t.o.r..........    00 74 00 6F 00 72 00 00 05 10 00 00 00 00 00 00
  .....G.u.e.s.t..    00 05 00 00 00 47 00 75 00 65 00 73 00 74 00 00
  .............a.a    01 10 00 00 00 00 00 00 00 08 00 00 00 aa 00 bb --- aa, bb substituted for the actual entry
  .a.y.y.y.z.z....    00 cc 00 dd 00 ee 00 ff 00 gg 00 hh 00 ii 00 00 --- cc through ii on this line
  .....               00 00 00 00 00
Frame Check Sequence:    0x00000000
```

Packet # 27 Administrator

The packet has been edited, in the interest of saving space, the Ethernet header, TCP header, and NetBios Session Service header are similar to packet 23, and do not significantly add to the discussion

SMB Transaction - Name, Bytes In/OutResponse

```
  Word Count:              10
  Total Param Bytes:       0
  Total Data Bytes:        524
  Reserved:                0x0000
  Params This Buffer:      0
  Params Bytes Offset:     56
  Params Displacement:     0
  Data This Buffer:        524
  Data Bytes Offset:       56
  Data Displacement:       0
  Setup Word Count:        0
  Reserved:                0x00
  Byte Count:              525
  Parameter And Data Bytes:
  .............@..  2E 05 00 02 03 10 00 00 00 0C 02 00 00 40 00 00
  .ô........|.....  00 F4 01 00 00 00 00 00 00 10 7C 14 00 15 00 18
  L@IÀÀ`AÀ.€BÜ‰”.À  4C 40 49 C0 C0 60 41 C0 01 80 42 DC 89 94 0B C0
  .Ð-O¿..¿........  01 D0 2D 4F BF 12 8F BF 01 00 00 00 00 00 00 00
  .Ðí¸éÛ.¿.ÿÿÿÿÿÿÿ  00 D0 ED B8 E9 DB 8F BF 01 FF FF FF FF FF FF FF
  .....Ho......pp.  7F 1A 00 1A 00 48 6F 14 00 00 00 00 00 70 70 14
  .....ÀP......8.   00 00 00 00 00 C0 50 14 00 00 00 00 00 38 09 14
  .....ÐP......`©.  00 00 00 00 00 D0 50 14 00 00 00 00 00 60 A9 14
  .l.l.øu......8..  00 6C 00 6C 00 F8 75 14 00 00 00 00 00 38 13 14
  .....è.......°P.  00 00 00 00 00 E8 90 14 00 00 00 00 00 B0 50 14
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .ô...........ÿÿÿ  00 F4 01 00 00 01 02 00 00 10 02 00 00 FF FF FF
  .¨...ˆ..........  00 A8 00 00 00 88 02 14 00 00 00 13 00 00 00 00
  ................  00 00 00 00 00 0D 00 00 00 00 00 00 00 0D 00 00
  .A.d.m.i.n.i.s.t  00 41 00 64 00 6D 00 69 00 6E 00 69 00 73 00 74
  .r.a.t.o.r......  00 72 00 61 00 74 00 6F 00 72 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .........6......  00 00 00 00 00 00 00 00 00 36 00 00 00 00 00 00
  .6...B.u.i.l.t.-  00 36 00 00 00 42 00 75 00 69 00 6C 00 74 00 2D
  .i.n. .a.c.c.o.u  00 69 00 6E 00 20 00 61 00 63 00 63 00 6F 00 75
  .n.t. .f.o.r. .a  00 6E 00 74 00 20 00 66 00 6F 00 72 00 20 00 61
  .d.m.i.n.i.s.t.e  00 64 00 6D 00 69 00 6E 00 69 00 73 00 74 00 65
  .r.i.n.g. .t.h.e  00 72 00 69 00 6E 00 67 00 20 00 74 00 68 00 65
  . .c.o.m.p.u.t.e  00 20 00 63 00 6F 00 6D 00 70 00 75 00 74 00 65
  .r./.d.o.m.a.i.n  00 72 00 2F 00 64 00 6F 00 6D 00 61 00 69 00 6E
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .....ì..........  00 00 00 00 00 EC 04 00 00 00 00 00 00 15 00 00
  .ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ  00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  ÿÿÿÿÿÿ. .....     FF FF FF FF FF FF 00 20 00 00 00 00 00
```

```
Frame Check Sequence:   0x00000000
```

Packet # 39 Guest
The packet has been edited, in the interest of saving space, the Ethernet header, TCP header, and NetBios Session Service header are similar to packet 23, and do not significantly add to the discussion

```
SMB Transaction - Name, Bytes In/OutResponse
  Word Count:             10
  Total Param Bytes:      0
  Total Data Bytes:       512
  Reserved:               0x0000
  Params This Buffer:     0
  Params Bytes Offset:    56
  Params Displacement:    0
  Data This Buffer:       512
  Data Bytes Offset:      56
  Data Displacement:      0
  Setup Word Count:       0
  Reserved:               0x00
  Byte Count:             513
  Parameter And Data Bytes:
  .............F..  2E 05 00 02 03 10 00 00 00 00 02 00 00 46 00 00
  .è........|....À  00 E8 01 00 00 00 00 00 00 10 7C 14 00 15 00 C0
  >...............  3E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .@pucp°¿........  00 40 70 75 63 70 B0 BF 01 00 00 00 00 00 00 00
  .........@p>T–ß¿  00 00 00 00 00 00 00 00 00 40 70 3E 54 96 DF BF
  .....@s......°P.  01 0A 00 0A 00 40 73 14 00 00 00 00 00 B0 50 14
  .....8.......`©.  00 00 00 00 00 38 13 14 00 00 00 00 00 60 A9 14
  .....ÐP......8..  00 00 00 00 00 D0 50 14 00 00 00 00 00 38 09 14
  .p.p.øu......ÀP.  00 70 00 70 00 F8 75 14 00 00 00 00 00 C0 50 14
  .....è.......pp.  00 00 00 00 00 E8 90 14 00 00 00 00 00 70 70 14
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .õ...........ÿÿÿ  00 F5 01 00 00 01 02 00 00 11 00 00 00 FF FF FF
  .¨...ˆ..........  00 A8 00 00 00 88 02 14 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 05 00 00 00 00 00 00 05 00 00
  .G.u.e.s.t.i....  00 47 00 75 00 65 00 73 00 74 00 69 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .........8......  00 00 00 00 00 00 00 00 00 38 00 00 00 00 00 00
  .8...B.u.i.l.t.-  00 38 00 00 00 42 00 75 00 69 00 6C 00 74 00 2D
  .i.n. .a.c.c.o.u  00 69 00 6E 00 20 00 61 00 63 00 63 00 6F 00 75
  .n.t. .f.o.r. .g  00 6E 00 74 00 20 00 66 00 6F 00 72 00 20 00 67
  .u.e.s.t. .a.c.c  00 75 00 65 00 73 00 74 00 20 00 61 00 63 00 63
  .e.s.s. .t.o. .t  00 65 00 73 00 73 00 20 00 74 00 6F 00 20 00 74
  .h.e. .c.o.m.p.u  00 68 00 65 00 20 00 63 00 6F 00 6D 00 70 00 75
  .t.e.r./.d.o.m.a  00 74 00 65 00 72 00 2F 00 64 00 6F 00 6D 00 61
  .i.n............  00 69 00 6E 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .........ì......  00 00 00 00 00 00 00 00 00 EC 04 00 00 00 00 00
  .....ÿÿÿÿÿÿÿÿÿÿÿ  00 15 00 00 00 FF FF FF FF FF FF FF FF FF FF FF
  ÿÿÿÿÿÿÿÿÿÿÿÿÿ...  FF FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00
  .                 00
Frame Check Sequence:  0x00000000
```

Packet #51 Oliver Viitamaki

The packet has been edited, in the interest of saving space, the Ethernet header, TCP header, and NetBios Session Service header are similar to packet 23, and do not significantly add to the discussion

```
SMB Transaction - Name, Bytes In/OutResponse
  Word Count:            10
  Total Param Bytes:     0
  Total Data Bytes:      568
  Reserved:              0x0000
  Params This Buffer:    0
  Params Bytes Offset:   56
  Params Displacement:   0
  Data This Buffer:      568
  Data Bytes Offset:     56
  Data Displacement:     0
  Setup Word Count:      0
  Reserved:              0x00
  Byte Count:            569
  Parameter And Data Bytes:
  .........8...L..  2E 05 00 02 03 10 00 00 00 38 02 00 00 4C 00 00
  . ........|....š  00 20 02 00 00 00 00 00 00 10 7C 14 00 15 00 9A
  ØðJy9ÎFÀ.ð½ž’ìEÀ  D8 F0 4A 79 39 CE 46 C0 01 F0 BD 9E 92 EC 45 C0
  ...hÛó2À........  01 00 03 68 DB F3 32 C0 01 00 00 00 00 00 00 00
  ..ÃÑ.½3À...1Ì.bÀ  00 00 C3 D1 05 BD 33 C0 01 00 03 31 CC 19 62 C0
  .....@s..  .  .Ho.  01 10 00 10 00 40 73 14 00 20 00 20 00 48 6F 14
  .....è.......ÀP.  00 00 00 00 00 E8 90 14 00 00 00 00 00 C0 50 14
  .....8.......ÐP.  00 00 00 00 00 38 09 14 00 00 00 00 00 D0 50 14
  .".".po......`©.  00 22 00 22 00 70 6F 14 00 00 00 00 00 60 A9 14
  .....pp..`.`.pÕ.  00 00 00 00 00 70 70 14 00 60 00 60 00 70 D5 13
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .è...........ÿÿÿ  00 E8 03 00 00 01 02 00 00 10 00 00 00 FF FF FF
  .¨...ˆ..........  00 A8 00 00 00 88 02 14 00 00 00 1E 00 00 00 00
  ................  00 00 00 00 00 08 00 00 00 00 00 00 00 08 00 00
  .A.A.A.Y.Y.Y.Z.Z  aa bb cc dd ee ff gg hh ii jj kk ll mm nn oo pp ---    This line modified to protect the guilty…, otherwise its
. .............O.l  00 10 00 00 00 00 00 00 00 10 00 00 00 4F 00 6C Intact.
  .i.v.e.r. .V.i.i  00 69 00 76 00 65 00 72 00 20 00 56 00 69 00 69
  .t.a.m.a.k.i....  00 74 00 61 00 6D 00 61 00 6B 00 69 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 11 00 00
  ........N.e.t.w  00 00 00 00 00 11 00 00 00 4E 00 65 00 74 00 77
  .o.r.k. .A.r.c.h  00 6F 00 72 00 6B 00 20 00 41 00 72 00 63 00 68
  .i.t.e.c.t.o....  00 69 00 74 00 65 00 63 00 74 00 6F 00 00 00 00
  ................  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  .....0.......0..  00 00 00 00 00 30 00 00 00 00 00 00 00 30 00 00
  .m.:. . . . . .   00 6D 00 3A 00 20 00 20 00 20 00 20 00 20 00 20
  . . . . . . . .   00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20
  . . . . . . .d.    00 20  00  20  00  20  00  20  00  20  00  20  00  64  00  09
  . . . . . . . .   00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20
  . . . . . . . .   00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20
  . . . . . . . .   00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20
  .ì...........ÿÿÿ  00 EC 04 00 00 00 00 00 00 15 00 00 00 FF FF FF
  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
  ÿÿ.......         FF FF 00 00 00 00 00 00 00
Frame Check Sequence:  0x00000000
```

3.0 Assignment 3 – “Analyze This” Scenario

Syn-Apps Consulting was asked to provide a bid for Security Services at GIAC Enterprises The data provided was analyzed, the following issues were noted. A group of decisions need to be made, by management, in order to proceed further. The cost of services will be quoted based on what services are agreed upon by management and Syn-Apps Consulting as being required.

Background

The Network at GIAC Enterprises has received numerous, repeated probes from various outside addresses. Some of the probing has been with “Hacking” tools crafted specifically for that purpose. It is well understood that once a specific machine has been identified with a specific vulnerability that the machine can be compromised in less than 60 seconds.

GIAC Enterprises has not provided a Network Map, a copy of their Security Policy, a list of their installed Operating System types or configuration of their screening router(s) or Firewall(s). That information would go a long way in helping to provide a tailored analysis of the data.

3.1 Executive Summary

Assumptions:

- GIAC Enterprises has known but undisclosed (to this analyst) issues with the Chinese Academy of Sciences, therefore the "Watchlist 222"
- GIAC Enterprises has known but undisclosed (to this analyst) issues with, NV-PICTUREVISION of Israel therefore "Watchlist 220"
- MY.NET.253.41, MY.NET.253.42, MY.NET.253.43 are the official E-mail servers for GIAC Enterprises.
  MY.NET.6.7 is not an E-mail server
- MY.NET.1.3, MY.NET.1.4, & MY.NET.1.5, are DNS Servers for GIAC Enterprises.

Issues:

Compromised Hosts:

- MY.NET.219.142, MY.NET.218.242, and MY.NET.217.218. Packets from these hosts have been captured with invalid combinations of flags. **Analysis of Type 3 Data, Suspicious Traffic** contains details.
- MY.NET.1.13 **Analysis of Type 2 Data, Second Generic Type** contains details

Potentially Compromised Hosts:

- MY.NET.6.35, and MY.NET.179.80 should be checked for signs of the Happy99 Virus, **Analysis of Type 1 Data** the section **Happy 99 Virus** contains the details.
- A large number of nodes on MY.NET.X.Y are participating in IRC type activities, and will soon be compromised, if they are not now, without strong defenses. **Analysis of Type 3 Data, Suspicious Traffic** contains details
- MY.NET.6.7, due to E-mail services scanning, from the Chinese Academy of Sciences**, Analysis of Type 1 Data** the section **Watchlist 000222 NET-NCFC** contains the details
- MY.NET.60.11, due to repeated Telnet accesses from the Chinese Academy of Sciences, **Analysis of Type 1 Data** the section **Watchlist 000222 NET-NCFC** contains the details
- MY.NET.217.42, MY.NET.219.26, MY.NET.218.218, MY.NET.217.82, MY.NET.211.2, MY.NET.105.2, and MY.NET.6.15 due to large numbers of attempts for RPC accesses from various sources**. Analysis of Type 1 Data** the sections **SUNRPC highport access!, Attempted Sun RPC high port access,** and **External RPC call** contain the details
- MY.NET.60.8, MY.NET.60.11, MY.NET.100.2, MY.NET.60.16, MY.NET.98.197, MY.NET.98.124, MY.NET.97.237, MY.NET.98.162, and MY.NET.98.193 due to large numbers of attempted accesses to port 1080, **Analysis of Type 1 Data** the section **WinGate1080 Attempt** contains the details.
- MY.NET.101.192 repeated attempted accesses to well known Microsoft ports **Analysis of Type 1 Data** the section **SMB Name Wildcard** contains the details
- MY.NET.208.178, MY.NET.221.94, and MY.NET.181.87 appear to be participating in Internet Relay Chat (IRC) activity with NV-PICTUREVISION, **Analysis of Type 1 Data** the section **Watchlist 000220 IL-ISDNNET-990517** contains the details
- Numerous Scanning tools have been used against the MY.NET. X.Y Network. The sections **SYN-FIN scan!**, **Null scan!, NMAP TCP ping!**, **Queso fingerprint,** and **Probable NMAP fingerprint attempt** in the section **Analysis of Type 1 Data,** virtually the whole of **Analysis of Type 2 Data,** and **Analysis of Type 3 Data** show evidence of the tools.

- The network device at MY.NET.101.192 has a default SNMP Password**, Analysis of Type 1 Data** the section **SNMP public access** contains the details

Targeted Hosts:

- The following hosts show signs of being directly targeted MY.NET.97.119, MY.NET.60.11, MY.NET.253.42, MY.NET. 219, MY.NET219.118, MY.NET207.74, MY.NET.5.7, MY.NET.98.160, MY.NET.220.190, MY.NET.217.206 MY.NET.202.150, MY.NET.253.112, MY.NET.97.230, MY.NET.218.34, MY.NET.98.188 and MY.NET.208.18. This observation is based on evidence in section **Analysis of Type 2 Data** subsection **Third Generic Type.**

Intrusion Detection Sensor:

- Detected data, is intermittently available, due to various reasons, examples are, runs out of disk space, crashes, power failure
- Large Volumes of data, when the data is available.

Actions:

- Compromised Hosts,
  1. remove compromise,
  2. when possible, install at a new IP address
- Potentially Compromised Hosts,
  1. validate respective compromise,
  2. remove compromise, when found
  3. consider installing at another IP address
- Intrusion Detection Sensor (IDS),
  1. consider installing more disk,
  2. removing detected data more frequently,
  3. putting the IDS on an UnInterruptable Power Supply
  4. analyzing the detected data on a more frequent (hourly) basis, with automated tools
  5. troubleshoot any remaining issues with IDS
  6. dedicate a full time Analyst to IDS

Decision Matrix:

- Leave things as they are, GIAC Enterprises to assume consequences.

or

- Decide which items internal resources can be used to troubleshoot and repair
- Decide which items will require attention from external resources

Action Plan:

The plan will be developed based on the output of the Decision Matrix. Naturally, due to the large number of issues a prioritized approach will be required. Syn-Apps Consulting is prepared to go ahead once a decision is made and direction is given.

3.2 Detailed analysis

3.2.0 Background

The Snort rules used as examples in the detects below have been obtained from the Snort 1.6.3 Ruleset Updated -- 10/10/2000. These rules do not in all cases directly match the Ruleset in use by GIAC Enterprises, and are used here as a basis for developing the hypothesis of how the detects were created. GIAC Enterprises is invited to make their Ruleset available, in cases where there is a strong divergence between the rule that I have assumed to be in place, and the rule actually in place, in the Snort Intrusion Detection System.

Supplementary Information used to develop the analysis of the Snort Rules

Identify and install tools that aid in detecting signs of intrusion. http://www.cert.org/security-improvement/practices/p042.html
Writing rules and understanding alerts for Snort, a network intrusion detection system http://www.cert.org/security-improvement/implementations/i042.14.html
Writing Snort Rules How To write Snort rules and keep your sanity Current as of version 1.6 by Martin Roesch http://www.snort.org/

Three types of data files were provided for analysis. It is assumed that the start of the period to be analyzed is August 15th at 00:00:00 and ends on September 14th at 24:00:00 as this is the time which is to some degree covered by all of the provided data sets.

One type was labeled SnortA## where ## is a randomly assigned numeric value. This file type contained data with the following characteristics:

```
08/11-00:33:44.374672  [**] Watchlist 000222 NET-NCFC [**] 159.226.23.155:37822 -> MY.NET.6.7:25
08/11-00:33:46.103627  [**] Watchlist 000222 NET-NCFC [**] 159.226.23.155:113 -> MY.NET.6.7:28835
08/11-00:33:47.338274  [**] Watchlist 000222 NET-NCFC [**] 159.226.23.155:37822 -> MY.NET.6.7:25
```

This will be referred to as **Type 1** data.

Type 1 data sets were unavailable for August 21,22,23,24,25,26,27,28,29,30, 31, September 1, and 4. Most data sets covered the whole day, on September 9th the data set ran over to September 10th by 2 hours, this may provide duplicate data for those 2 hours as this was not excluded. This should be a minor issue judging by the overall traffic provided.

The second file type was labeled SnortS##, where ## is again a randomly assigned numeric value. This file type contained data with the following characteristics:

```
Aug 15 00:46:11 195.114.226.41:2244 -> MY.NET.1.2:21 SYN **S*****
Aug 15 00:46:12 195.114.226.41:2250 -> MY.NET.1.8:21 SYN **S*****
Aug 15 00:46:14 195.114.226.41:2252 -> MY.NET.1.10:21 SYN **S*****
```

This will be referred to as **Type 2** Data.

Type 2 data sets were unavailable for August 19,20,21,22,23,24,25,26,27,29,30,31, September 1,and 12. The Data set for September 5th did not cover the whole day.

The third file type was labeled SOOS## where ## is again a randomly assigned numeric value. This file type contained data with the following characteristics:

```
08/28-00:27:14.211201 128.194.9.94:1575 -> MY.NET.201.190:6699
TCP TTL:116 TOS:0x0 ID:41583  DF
**SFR**U Seq: 0xB3D2B34   Ack: 0x80025E   Win: 0x5010
TCP Options => EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-00:47:45.028963 128.194.9.94:1578 -> MY.NET.201.190:6699
TCP TTL:116 TOS:0x0 ID:49529  DF
2*SFRP*U Seq: 0xB3FD0DF   Ack: 0x150278   Win: 0x5010
06 2A 1A 2B 0B 3F D0 DF 00 15 02 78 02 6F 50 10  .*.+.?.....x.oP.
22 36 AF E9 00 00 15 96 DC A6 BC 6F 89 DD E3 FB  "6.........o....
A7 73                                            .s
```

This will be referred to as **Type 3** Data

Type 3 data sets were unavailable for August 15 to the 27th, 30th, September 3,4. Some data sets which were provided, did not cover the whole day.

In order to understand, and help determine an approach to analyze the data, it was considered appropriate to break down the data, and determine the number and types of attacks, that the network was under during the sampling period. The following duplicate files have not been included in the data. SnortA20 contains the same data set as SnortA21, therefore SnortA21 was excluded, SnortS20 contains the same data set as SnortS21, therefore S21 was excluded, SOOS9 contains the same data set as SOO10, and therefore SOO10 was excluded.

3.2.1 Analysis of Type 1 Data

TOP Detects in the Alert (Type 1) Files

Type of Alert	Total of this Type
Watchlist 000222 NET-NCFC	15405
Watchlist 000220 IL-ISDNNET-990517	4480
WinGate 1080 Attempt	3857
SYN-FIN scan!	3065
Attempted Sun RPC high port access	1869
SNMP public access	607
SMB Name Wildcard	315
Null scan!	154
NMAP TCP ping!	131
SUNRPC highport access!	63
Queso fingerprint	46
Probable NMAP fingerprint attempt	41
External RPC call	40
TCP SMTP Source Port traffic	8
Possible wu-ftpd exploit - GIAC000623	8
Happy 99 Virus	1
Total Detected Types of Attacks	30090

Top 20 Addresses Attracting Attention in the Alert (Type1) Files

Detection and Destination Address	Number of attempts
Attempted Sun RPC high port access at MY.NET.217.42	1054
SNMP public access at MY.NET.101.192	788
Attempted Sun RPC high port access at MY.NET.219.26	391
WinGate 1080 Attempt at MY.NET.60.8	258
SMB Name Wildcard at MY.NET.101.192	253
Attempted Sun RPC high port access at MY.NET.218.218	240
Attempted Sun RPC high port access at MY.NET.217.82	174
WinGate 1080 Attempt at MY.NET.60.11	167
WinGate 1080 Attempt at MY.NET.100.2	91
SUNRPC highport access! at MY.NET.211.2	57
Attempted Sun RPC high port access at MY.NET.220.58	52
Attempted Sun RPC high port access at MY.NET.105.2	45
WinGate 1080 Attempt at MY.NET.60.16	39
External RPC call at MY.NET.6.15	27
WinGate 1080 Attempt at MY.NET.98.197	25
WinGate 1080 Attempt at MY.NET.98.124	21
WinGate 1080 Attempt at MY.NET.97.237	18
WinGate 1080 Attempt at MY.NET.98.162	18
NMAP TCP ping! at MY.NET.1.8	16
WinGate 1080 Attempt at MY.NET.98.193	15

3.2.1.1 Watchlist 000222 NET-NCFC

Snort Rule

Custom - This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from a version other than 1.6.3.

possibly, an earlier version of Snort. It has been set up to capture any traffic from159.226.X.Y to MY.NET.X.Y It will be similar to the following, although the actual rule may have more options specified.

```
alert TCP 159.226.0.0/16 any -> $HOME_NET any (msg:" Watchlist 000222 NET-NCFC ";)
```

Description: This rule has been generated to log any TCP traffic with a source address belonging to The Computer Network Center, Chinese Academy of Sciences , Institute of Computing Technology, Chinese Academy of Sciences. GIAC Enterprises obviously has a special interest in keeping track of traffic from Institute of Computing Technology Chinese Academy of Sciences.

Known Issues: The standard issues involved with network scanning, Operating System evaluation for exposures, exposure of proprietary corporate data to potential competitors, Denial of Service (DoS) etc. There is more information required from GIAC Enterprises, in order to better understand their concerns with this address block, and perhaps be able to adjust (tune) the snort rules accordingly. As an example see Figure 1. This drawing depicts how the E-mail system between GIAC Enterprises and the Chinese Academy of Sciences is working, based on sensor data. Port 113 traffic incoming from the Chinese Academy of Sciences to MY.NET.253.41, MY.NET.253.42, MY.NET.253.43 are Ident (Identify) requests. The Ident requests are as a result of E-mail sent from GIAC Enterprises to the Chinese Academy of Sciences. The Sendmail program at the Chinese Academy of Sciences, has been configured to attempt to identify the individual sending the E-mail from GIAC Enterprises therefore the Ident traffic. The Sendmail program at NV-PICTUREVISION has not been configured to identify an individual, therefore it does not generate the Ident traffic . The rule above does not track the outgoing traffic to the Chinese Academy of Sciences, just the incoming traffic, therefore we do not see the outgoing E-mail traffic, and have one half of the information required. This E-mail/ident traffic alone generates 13532 detects out of the 15405. Has GIAC Enterprises received many Virii from this site through E-mail?

MY.NET.6.7 is assumed not to be an E-mail server. It is being probed for E-mail Services. This is a concern.

Known SMTP Exposures:

CAN-2000-0738 WebShield SMTP 4.5 allows remote attackers to cause a denial of service by sending e-mail with a From: address that has a . (period) at the end,
which causes WebShield to continuously send itself copies of the e-mail.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0738

CAN-2000-0657 Buffer overflow in AnalogX proxy server 4.04 and earlier allows remote attackers to cause a denial of service via a long HELO command in the SMTP protocol.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0657

CVE-2000-0582 Check Point FireWall-1 4.0 and 4.1 allows remote attackers to cause a denial of service by sending
a stream of invalid commands (such as binary zeros) to the SMTP Security Server proxy.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0582

CVE-1999-0203 In Sendmail, attackers can gain root privileges via SMTP by specifying an improper “mail from”
address and an invalid “rcpt to” address that would cause the mail to bounce to a program.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0203

CVE-1999-0047 MIME conversion buffer overflow in Sendmail versions 8.8.3 and 8.8.4. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0047

CVE-1999-0130 Local users can start Sendmail in daemon mode and gain root privileges. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0130

CVE-1999-0131 Buffer overflow and denial of service in Sendmail 8.7.5 and earlier through GECOS field gives root access to local users.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0131

CVE-1999-0203 In Sendmail, attackers can gain root privileges via SMTP by specifying an improper "mail from" address and an invalid "rcpt to" address

that would cause the mail to bounce to a program. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0203
CVE-1999-0204 Sendmail 8.6.9 allows remote attackers to execute root commands, using ident. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0204
CVE-1999-0206 MIME buffer overflow in Sendmail 8.8.0 and 8.8.1 gives root access. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0206
Advisory CA-1997-05 MIME Conversion Buffer Overflow in Sendmail Versions 8.8.3 and 8.8.4 http://www.cert.org/advisories/CA-1997-05.html
SANS CVE Entries http://www.sans.org/y2k/CVE.htm

The vast majority of the remaining traffic (the last two sections below) is made up of two different types, first from the Chinese Academy of Sciences to MY.NET.X.Y port 23 and from MY.NET.X.Y port 23 to the Chinese Academy of Sciences **is a major concern**. This is **Telnet** traffic it is used to create a Virtual Terminal service on a remote machine. It allows the user who logs on, to run programs on that machine as well as **potentially become ROOT** on the machine.
BugtraqID: 459 http://www.securityfocus.com/bid/459.html
BugtraqID: 594-Possible to set the TERM environmental variable before connecting. http://www.securityfocus.com/bid/594.html
Nt4.0 Telnet to port 53 vulnerability http://support.microsoft.com/support/kb/articles/Q169/4/61.ASP
Win2k Telnet.exe malicious server vulnerability http://www.insecure.org/sploits/NT.NTLM.auto-authentication.html
Windows 2000 Telnet Client NTLM Authentication" Vulnerability http://www.insecure.org/sploits/NT.NTLM.auto-authentication.html

Detection:

```
08/11-00:33:47.338274  [**] Watchlist 000222 NET-NCFC [**] 159.226.23.155:37822 -> MY.NET.6.7:25
08/11-02:13:17.167679  [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:113 -> MY.NET.253.43:55219
08/11-02:13:19.466406  [**] Watchlist 000222 NET-NCFC [**] 159.226.63.200:1843 -> MY.NET.253.43:25

08/11-01:51:05.043450  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1051 -> MY.NET.6.7:23 to
08/11-02:23:28.664773  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1051 -> MY.NET.6.7:23 then
08/11-02:23:44.834384  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1054 -> MY.NET.60.8:23 to
08/11-02:25:17.278243  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1054 -> MY.NET.60.8:23 then
08/11-02:25:24.710017  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1055 -> MY.NET.6.7:23 to
08/11-03:01:53.618330  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.108:1057 -> MY.NET.6.7:23 then
08/16-20:42:09.047163  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:4628 -> MY.NET.6.7:23 to
08/16-20:50:55.336179  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:4628 -> MY.NET.6.7:23

09/11-10:58:58.817818  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:23 -> MY.NET.163.32:1060 to
09/11-11:17:22.505410  [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:23 -> MY.NET.163.32:1060 then
08/11-16:11:43.712817  [**] Watchlist 000222 NET-NCFC [**] 159.226.41.166:23 -> MY.NET.60.11:10593 to
08/11-16:21:53.440338  [**] Watchlist 000222 NET-NCFC [**] 159.226.41.166:23 -> MY.NET.60.11:10593
```

There are 15405 total detects logged to various addresses on MY.NET.X.X. they are not listed here in the interests of keeping this reasonably brief.

The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
Institute of Computing Technology Chinese Academy of Sciences
Beijing, China
Netname: NCFC
Netnumber: 159.226.0.0
Coordinator:Qian, Haulin (QH3-ARIN) hlqian@NS.CNC.AC.CN
+86 1 2569960

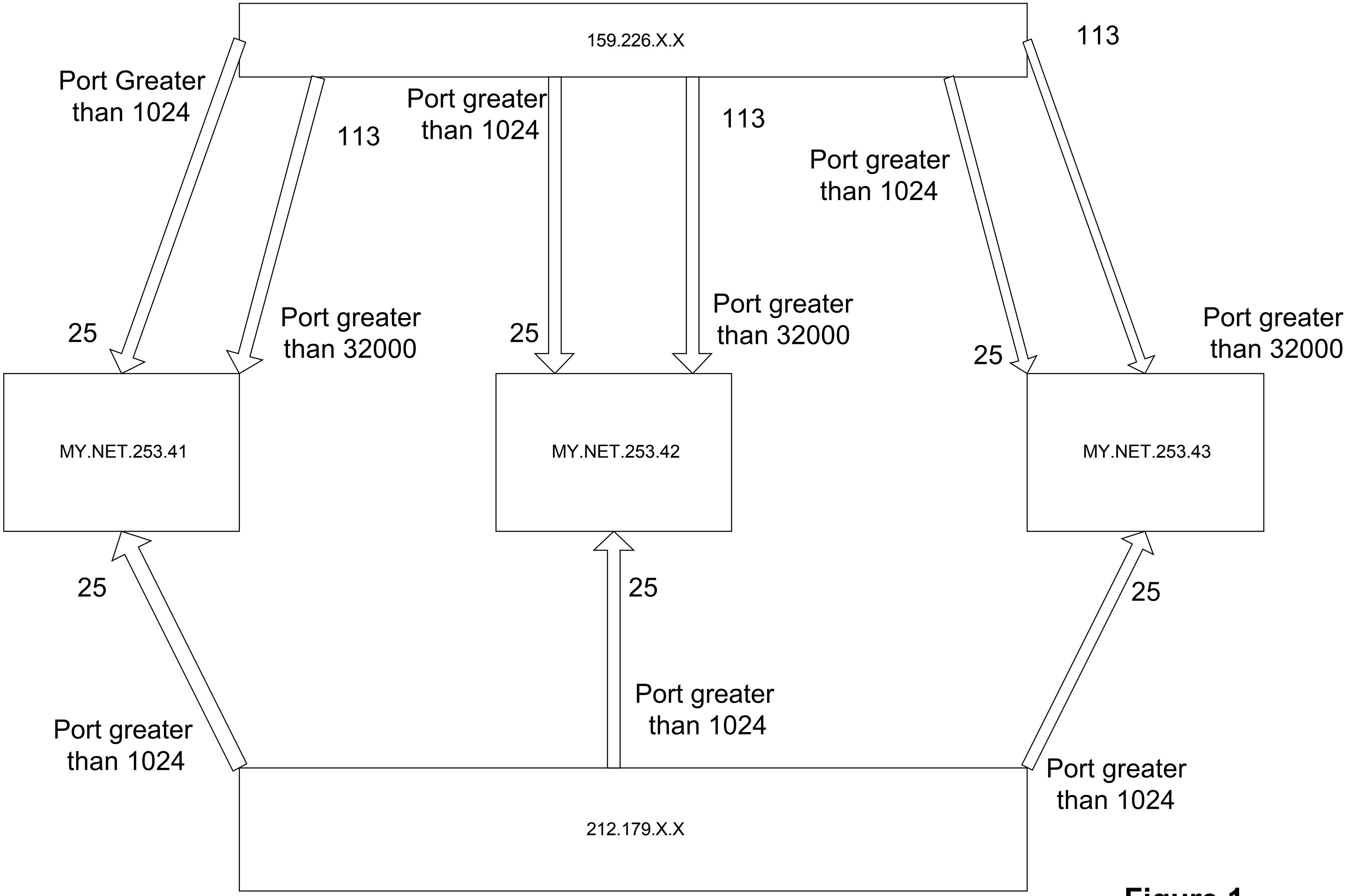


Figure 1
E-mail Systems

3.2.1.2 Watchlist 000220 IL-ISDNNET-990517

Snort Rule

Custom - This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from a version other than 1.6.

possibly, an earlier version of Snort. It has been set up to capture any traffic from 212.79.X.X to MY.NET.X.X It will be similar to the following, although the actual rule may have more options specified.

```
alert TCP 212.179.0.0/16 any -> $HOME_NET any (msg:" Watchlist 000220 IL-ISDNNET-990517 ";)
```

Description: This rule has been generated to log any TCP traffic with a source address belonging to NV-PICTUREVISION in Israel. GIAC Enterprises obviously has a special interest in keeping track of traffic from them.

Known Issues: Stacheldraht and Trinity distributed denial of service (DDoS), "ILOVEYOU" virus, EvilFTP, phAse Zero, ExploreZip.worm, and SubSeven. Are all known to be distributed by, take part in, and communicate within an Internet Relay Chat (IRC) channel. There is more information required from GIAC Enterprises, in order to better understand their specific reason for leaving this address block open. Several machines at GIAC Enterprises appear to be participating in IRC, sessions with this address block, NV-PICTUREVISION.

CERT® Co-Ordination Center: Results of the Distributed-Systems Intruder Tools Workshop http://www.cert.org/reports/dsit_workshop-final.html

CERT® Advisory CA-1994-14 Trojan Horse in IRC Client for UNIX http://www.cert.org/advisories/CA-1994-14.html

CAN-2000-0138 A system has a distributed denial of service (DDOS) attack master, agent, or zombie installed, such as (1) Trinoo, (2) Tribe Flood Network (TFN), (3) Tribe Flood Network 2000 (TFN2K), (4) stacheldraht, (5) mstream, or (6) shaft. http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Stacheldraht

SANS Institute, Help Defeat Denial of Service Attacks: Step-by-Step http://www.sans.org/dosstep/index.htm

SANS Institute, NAPSTER - Should You Be Worried About It? http://www.sans.org/infosecFAQ/napster.htm

SANS Institute, Gnutella defeats many perimeter defenses http://www.sans.org/infosecFAQ/gnutella.htm

SANS Institute, The "stacheldraht" Distributed Denial of Service Attack Tool http://www.sans.org/newlook/resources/IDFAQ/stacheldraht.htm

SANS Institute, Distributed Denial of Service Attack Tools: trinoo and wintrinoo http://www.sans.org/newlook/resources/IDFAQ/trinoo.htm

Detection:

08/11-11:35:46.194548 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.2:23 -> MY.NET.98.168:1026 to **08/11-11:51:55** Telnet from MY.NET.98.168

08/15-11:24:13.774593 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.32.2:7070 -> MY.NET.10.77:1357 Real Audio

08/16-09:06:07.879318 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.61.247:2052 -> MY.NET.5.29:443 to **08/16-09:08:27** single machine single port HTTPS

08/17-00:51:52.776518 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.44.62:30246 -> MY.NET.15.41:6690

08/17-12:45:31.229873 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.66.2:4807 -> MY.NET.181.87:6699 to **08/17-12:50:46** single machine single port (IRC, Napster)

08/18-02:58:42.142904 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.32.2:7070 -> MY.NET.98.164:1745 to **08/18-03:02:17** single machine single port possibly Real Audio

08/18-07:09:45.681474 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.61.252:1875 -> MY.NET.5.29:443 to **08/18-07:18:44** single machine single port
possibly HTTPS

08/20-09:05:41.982563 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.29.150:1098 -> MY.NET.53.28:4407 to **08/20-09:08:38** single machine single port

08/20-10:22:34.092107 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.61.244:2350 -> MY.NET.5.29:443 to **08/20-10:27:29** single machine single port
possibly HTTPS

09/03-03:37:20.882995 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.27.111:1526 -> MY.NET.206.154:6700 to **09/03-03:51:05** single machine single port

09/03-07:30:23.843182 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.62.74:1984 -> MY.NET.224.78:6346

09/03-13:22:15.772022 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.27.6:1948 -> MY.NET.253.105:26411

09/03-13:22:16.947056 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.27.6:1947 -> MY.NET.253.105:21

09/03-13:22:17.317752 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.27.6:1948 -> MY.NET.253.105:26411

```
09/03-13:22:37.089030  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.27.6:1948 -> MY.NET.253.105:26411
09/06-08:41:41.245944  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.61.5:21263 -> MY.NET.220.42:2367
09/06-18:47:21.629523  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.47.30:6346 -> MY.NET.223.62:2995 to 09/06-22:59:21 single machine various ports
09/07-03:27:11.855673  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.4:42790 -> MY.NET.253.42:25 to 09/07-03:27:35 possible e-mail
09/07-03:27:14.459009  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.4:42795 -> MY.NET.253.41:25 to 09/07-03:28:01 possible e-mail
09/09-10:45:10.461542  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.66.2:22756 -> MY.NET.221.94:6699 to 09/09-10:57:36 single machine single port
                                                                                                                   (IRC, Napster)
09/12-10:20:43.120997  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.127.45:1063 -> MY.NET.202.58:6688 to 09/12-10:30:50. single machine single
port(IRC)
09/12-13:14:42.722174  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.7.36:1462 -> MY.NET.253.42:25 to 09/13-04:35:26 possible e-mail
09/13-12:10:54.004320  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.204:1430 -> MY.NET.205.254:6699 (IRC, Napster)
09/13-15:09:12.472237  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.61.5:21263 -> MY.NET.204.150:2669 to 09/13-15:15:23 single machine single port
09/14-07:41:19.868471  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.58.174:2172 -> MY.NET.157.200:6699 to 09/14-07:45:26 single machine single port
                                                                                                                                  (IRC Napster)
09/14-10:44:51.813279  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.7.36:1192 -> MY.NET.253.43:25 to 09/14-10:46:18 possible e-mail
09/14-22:43:16.170099  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.67.195:6699 -> MY.NET.208.178:2575 to 09/14-22:43:47 single machine
                                                                                                                                  single port (IRC, Napster)
```

```
inetnum:     212.179.58.0 - 212.179.58.255
netname:      NV-PICTUREVISION
person:      Nati Pinko
address:      Bezeq International
address:      Petach Tikvah  Israel
phone:       +972 3 9257761
```

3.2.1.3 WinGate 1080 Attempt

Snort Rule

```
alert TCP !$HOME_NET !53 -> $HOME_NET 1080 (msg:"MISC-WinGate-1080-Attempt"; flags: S; )
```

Description: This Snort rule has been created to capture any traffic coming from any source address, not using port 53, destined for MY.NET port 1080. This port has many well known vulnerabilities some of which are mentioned below.

Known Issues: This activity is associated with individuals probing for Windows machines that they can then "Nuke", or for crashing a Proxy Service on UNIX Machines

ISS Notification: NukeNabber connection timeout denial of service http://xforce.iss.net/static/1540.php
CERT® Vulnerability Note VN-98.03 Topic: WinGate IP Laundering http://www.cert.org/vul_notes/VN-98.03.WinGate.html
CERT® Windows 95/98 Computer Security Information TechTip http://www.cert.org/tech_tips/win-95-info.html
CERT® Incident Note IN-99-01 "sscan" Scanning Tool http://www.cert.org/incident_notes/IN-99-01.html
CVE-1999-0290 The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0290
CVE-1999-0291 The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0291
CVE-1999-0441 Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0441
CVE-1999-0494 Denial of service in WinGate proxy through a buffer overflow in POP3.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0494

Detection:

There were various detects, from many subnets, to many of MY.NET.X.X. These entries require further investigation once there is a network map provided showing which nodes on MY.NET provide which services.

The following detects are important, as this is a node mapping the MY.NET network and starts at
09/11-18:40:36.435240 [**] WinGate 1080 Attempt [**] 168.187.26.157:1518 -> MY.NET.1.9:1080
scans the following subnets 1,2,4,5,7,9,10,11,12,13,NO 14.2,15,17,18,20,21,25,26,53,54and ends with
09/11-19:16:00.783231 [**] WinGate 1080 Attempt [**] 168.187.26.157:1270 -> MY.NET.54.234:1080. Obviously an automated scan, hitting approximately 5-10 nodes per second. There were some addresses duplicated, and also out of sequence, so it is likely that the script on 168.187.26.157 may have been having problems, or that there were routing issues from 168.187.26.157 from/to MY.NET.X.X.

Supporting data from the Type 2 Files
Sep 11 18:41:16 168.187.26.157:1854 -> MY.NET.1.205:1080 SYN **S*****
Scanned the following subnets @ 5-10/sec 1,2,4,5,7,9,10,11,12,13,NO 14.2,15,17,18,20,21,25,26,53,54

Kuwait Ministry of Communications (NET-MOC-KW)
Netname: MOC-KW
  Netnumber: 168.187.0.0
  Coordinator: Sharif, Majeed (MS695-ARIN) msharif@KEMS.NET
        (965) 2443808

3.2.1.4 SYN-FIN scan!

Snort Rule

```
alert TCP !$HOME_NET any -> $HOME_NET any (msg:"SCAN-SYN FIN"; flags: SF; )
```

Description: This rule has been created to detect illegal combinations of flags in a TCP session. A node (client) wishing to communicate with another node (server) establishes the TCP connection with a TCP packet containing a SYN, the initial sequence number it wishes to use, to the well known port it wishes to connect to. If the server is willing and able to establish a connection, it responds with a packet, which contains a SYN-ACK, increments the client's initial sequence number by 1 and supplies the server's initial sequence number. If that port is not active on the server, a reset is sent instead to the client node. The client responds to the server's SYN with an ACK, and increments the server's initial sequence number by 1. The initial sequence numbers are incremented as appropriate to each node, with each transfer of data. Communication then continues until the session is complete. The start of session tear down is initiated with a FIN from, which ever node has completed first. The other node responds with an ACK, and if it has completed its portion of the session, as well, it responds with a FIN, if it has not completed its side of the communication it continues until it is complete. The node that has sent its FIN will continue to respond even though it may already have sent a FIN, until both ends of the conversation have sent a FIN, to which the other node responds with an ACK.. As can be observed, the combination of SYN and FIN do not naturally occur together at the same time, in a normal session. This combination of TCP flags is generally used to evade filtering routers or Firewalls which do not maintain state, and thereby pass traffic into the network being protected, for the purpose of mapping it out.

Known Issues:

CERT® Incident Note IN-99-01 "sscan" Scanning Tool http://www.cert.org/incident_notes/IN-99-01.html
CERT Incident Note IN-98.02 New Tools Used For Widespread Scans http://www.cert.org/incident_notes/IN-98.02.html

Detection:

```
08/17-09:39:01.814788  [**] SYN-FIN scan! [**] 130.149.41.70:1242 -> MY.NET.217.46:994
08/17-15:40:13.987546  [**] SYN-FIN scan! [**] 130.149.41.70:1063 -> MY.NET.217.46:994
08/17-15:40:31.391622  [**] SYN-FIN scan! [**] 130.149.41.70:1063 -> MY.NET.217.46:994
08/17-15:41:19.562367  [**] SYN-FIN scan! [**] 130.149.41.70:1063 -> MY.NET.217.46:994
130.149.41.70 Canonical name: bessy.physik.TU-Berlin.DE
Technische Universitaet Berlin (NET-TUB)
  Netnumber: 130.149.0.0
Coordinator: Kasielke, Dieter  (DK116-ARIN)  Kasielke@ZRZ.TU-BERLIN.DE
            +49 30 314 23733

08/18-06:10:13.466733  [**] SYN-FIN scan! [**] 18.116.0.75:111 -> MY.NET.6.15:111
08/18-06:10:13.848468  [**] SYN-FIN scan! [**] 18.116.0.75:111 -> MY.NET.15.127:111
08/18-06:10:17.256725  [**] SYN-FIN scan! [**] 18.116.0.75:111 -> MY.NET.100.130:111
18.116.0.75 Canonical name: FLUTTER.MIT.EDU
Massachusetts Institute of Technology (NET-MIT-TEMP)
  Netname: MIT
  Netblock: 18.0.0.0 - 18.255.255.255
  Coordinator:Schiller, Jeffrey I  (JIS-ARIN)  jis@MIT.EDU
                 +1 617 253-8400 (FAX) +1 617 258-8736

09/02-00:28:03.467564  [**] SYN-FIN scan! [**] 210.101.101.110:111 -> MY.NET.6.15:111 and
09/02-10:14:17.559327  [**] SYN-FIN scan! [**] 210.101.101.110:23 -> MY.NET.6.15:23
inetnum:     210.101.64.0 - 210.101.127.255
```

```
netname:     KORNET
descr:       Korea Telecom
person:      Gisu Choi  e-mail:      mgr@ns.kornet.nm.kr
phone:       +82 2 766 1407  fax-no:      +82 2 766 6008
country:     KR
```

```
09/02-20:12:37.581548  [**] SYN-FIN scan! [**] 24.201.209.192:6688 -> MY.NET.202.254:2547
Videotron Ltee (NETBLK-VL-2BL)
   Netname: VL-2BL
   Netblock: 24.200.0.0 - 24.202.255.255
   Maintainer: VLCA
   Coordinator: Roy, Pierre  (PR163-ARIN)  pierre_roy@VIDEOTRON.COM  for abuse, E-mail abuse@videotron.ca
                                    (514) 985-8656
```

09/07-21:33:23.187413 [**] SYN-FIN scan! [**] 213.25.136.60:9704 -> MY.NET.1.4:9704

213.25.136.60 then maps the following subnets, 1,2,4,5,6,7,9,10,11,12,13, tests the one node MY.NET.14.2, continues mapping the following subnets 15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85 subnets in a random fashion using port 9704 as the source and destination until **09/07-21:33:30.514289**.

Supporting data from the Type 2 Files

```
Sep  7 21:33:23 213.25.136.60:9704 -> MY.NET.1.4:9704 SYNFIN **SF****
1,2,4,5,6,7,9,10,11,12,13,MY.NET.14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85
Sep  7 21:40:36 213.25.136.60:9704 -> MY.NET.85.254:9704 SYNFIN **SF****
```

Supporting data from the Type 3 files

```
09/07-21:33:30.514289 213.25.136.60:9704 -> MY.NET.1.4:9704
starts and maps the following subnets 1,2,4,5,6,7,9,10,11,12,13,MY.NET.14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85
09/07-21:40:44.000483 213.25.136.60:9704 -> MY.NET.85.254:9704
```

```
inetnum:     213.25.136.0 - 213.25.136.15
netname:     E-SOLUTIONS
descr:       e-SOLUTIONS.com Poland Sp. z o.o. PL
person:   Wieslaw Kosidlak e-mail:      wkosidlak@mfrelas.com
phone:       +48 81 7453340
fax-no:      +48 81 7453315
```

09/11-06:45:13.077482 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.1.3:21

starts and maps 1,2,4,5,7,9,10,11,12,13,NO 14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100, 104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151,152,153,154,155,156,157,158,159,160,161,162,163,178,179180,181,182,183,184,185,186,188,190,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225226,227,228,229,230,231,232,253,254

09/11-07:06:48.842343 [**] SYN-FIN scan! [**] 210.61.144.125:21 -> MY.NET.254.250:21. Port 21 was used as the source and destination ports in both cases

Supporting data from the Type 2 Files

Sep 11 06:45:13 **210.61.144.125**:21 -> MY.NET.1.3:21 SYNFIN **SF****
mapping 1,2,4,5,7,9,10,11,12,13,NO 14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100, 104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151, 152,153,154,155,156,157,158,159,160,161,162,163,178,179,180,181,182,183,184,185,186,188,190, 198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,253,254
Sep 11 07:06:48 210.61.144.125:21 -> MY.NET.254.250:21 SYNFIN **SF

Supporting data from the Type 3 Files

210.61.144.125
09/11-06:45:14.854416 210.61.144.125:21 -> MY.NET.1.3:21
starts and maps the following subnets 1,2,4,5,6,7,9,10,11,12,13,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100, 104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151,152,153,154,155,156,157,158,159,160,161,162,163,178,179180,181,182,183,184,185,186,188,190,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225226,227,228,229,230,231,232,253,254
09/11-07:06:50.833774 210.61.144.125:21 -> MY.NET.254.254:21

inetnum: **210.61.144.0 - 210.61.144.255**
netname: HINET8-144-TW
descr: Abnet Information Co., Ltd
descr Taipei, Taiwan TW
person: Wen-Lon Li
address: Abnet Information Co., Ltd
phone: +886-2-558-2115
fax-no: +886-2-558-2116
e-mail: abnet@ms15.hinet.net

3.2.1.5 Attempted Sun RPC high port access

Snort Rule

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

Description: This rule has been created to detect the individuals that are attempting to compromise the Remote Procedure Call (RPC) programs directly. The normal port for the portmapper service is port 111. When the portmapper service is quizzed, on port 111, it would identify what RPC services are running on that machine, those services are generally located in the 32700 port area on a machine implementing a SUN operating System. In this case the individual is going directly for the RPC services without accessing the portmapper. This method of accessing the services directly, is considered more “stealthy” than asking the portmapper, and then accessing the ports, after being informed by the RPC service where they are. In the Sun implementation of UNIX, it is generally recognized that the RPC ports start at 32771, hence the attack (and Snort rule) is specifically targeting (identifying) Sun machines. The ports found open, will represent services running which will then provide information on which services to attack.

Known Issues:

CVE-1999-0008 Buffer overflow in NIS+, in Sun's rpc.nisd program http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0008

CVE-1999-0212 Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0212

CVE-1999-0320 SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0320

CVE-1999-0974 Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad service.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0974

CAN-1999-0195 Denial of service in RPC portmapper allows attackers to register or unregister RPC services or spoof RPC services using a spoofed source IP address such as 127.0.0.1. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0195

CAN-1999-0568 rpc.admind in Solaris is not running in a secure mode. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0568

CAN-1999-0795 The NIS+ rpc.nisd server allows remote attackers to execute certain RPC calls without authentication to obtain system information, disable logging, or modify caches. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0795

CERT® Advisory CA-2000-17 Input Validation http://www.cert.org/advisories/CA-2000-17.html

CERT® Incident Note IN-2000-10 http://www.cert.org/incident_notes/IN-2000-10.html

CERT® Summary CS-2000-03 http://www.cert.org/summaries/CS-2000-03.html

CERT® Incident Note IN-99-04 Similar Attacks Using Various RPC Services http://www.cert.org/incident_notes/IN-99-04.html

CA-99-08 - Buffer Overflow Vulnerability in rpc.cmsd http://www.cert.org/advisories/CA-99-08-cmsd.html

CA-99-05 - Vulnerability in statd exposes vulnerability in automountd http://www.cert.org/advisories/CA-99-05-statd-automountd.html

CA-98.11 - Vulnerability in ToolTalk RPC Service http://www.cert.org/advisories/CA-98.11.tooltalk.html

ID FAQ - The trouble with RPCs http://www.sans.org/newlook/resources/IDFAQ/trouble_RPCs.htm

RFC for RPC Binding Protocols for ONC RPC Version 2 http://www.ietf.org/rfc/rfc1833.txt

Detection:

```
09/02-09:39:11.608534  [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
09/07-05:37:39.663140  [**] SUNRPC highport access! [**] 212.204.196.241:665 -> MY.NET.6.15:32771
```

inetnum: **212.204.196.241 - 212.204.196.250**
netname: S193
descr: Widexs BV
notify: hostmaster@widexs.nl
person: M Mace e-mail: mace@s072.widexs.nl
address: Hoofddorp NL

```
phone:       +31 23 5698073

09/06-23:10:10.012419  [**] SUNRPC highport access! [**] 193.64.205.17:56880 -> MY.NET.211.2:32771 to
09/06-23:13:18.324468  [**] SUNRPC highport access! [**] 193.64.205.17:56880 -> MY.NET.211.2:32771
inetnum:     193.64.205.0 - 193.64.205.15
netname:     TARVEASUNNOT-FI-2
descr:       KPNQwest Finland 193.64/15 superblock
notify:      hostmaster@FI.KPNQwest.net
person:      Janne Jaaskelainen phone:       +358 9 54919390
address:     ESPOO, FINLAND

09/07-21:10:18.892811  [**] SUNRPC highport access! [**] 207.29.195.22:2646 -> MY.NET.211.2:32771
NetReach, Inc. (NETBLK-NRCH-NETREACH-NET)
   Ambler, PA 19002 AUS
   Netname: NRCH-NETREACH-NET
   Netblock: 207.29.192.0 - 207.29.207.255
   Coordinator: NetReach, Inc.  (IN17-ARIN)  admin@netreach.net
                     215.283.2300

09/08-16:34:54.280910  [**] SUNRPC highport access! [**] 205.188.4.42:5190 -> MY.NET.210.2:32771
America Online, Inc (NETBLK-AOL-DTC)
   Sterling, VA 20166 AUS
   Netname: AOL-DTC
   Netblock: 205.188.0.0 - 205.188.255.255
   Coordinator: America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
                      703-265-4670

09/11-21:24:53.037663  [**] SUNRPC highport access! [**] 209.10.41.242:21 -> MY.NET.211.2:32771
Globix Corporation (NETBLK-GLOBIXBLK3)
   NY, NY 10012
   Netname: GLOBIXBLK3
   Netblock: 209.10.0.0 - 209.11.159.255
   Coordinator: Hostmaster, Globix Corporation  (GCH2-ARIN)  arin-admin@GLOBIX.NET
                      212.334.8500 (FAX) 212.334.8615
```

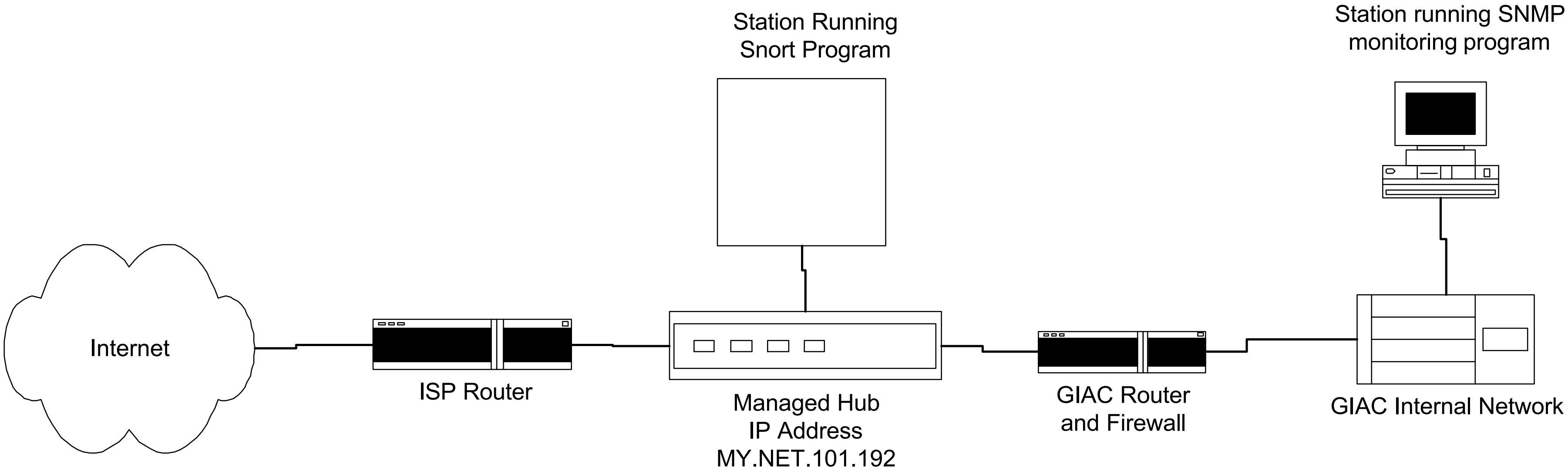
3.2.1.6 SNMP public access

Snort Rule Custom - This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from an earlier version of Snort. It has been set up to capture SNMP traffic to port 161 on MY.NET.X.X It will be similar to the following, although the actual rule may have more options.

```
alert UDP any any -> $HOME_NET 161 (msg:"SNMP public access ";)
```

Description:

The Simple Network Monitoring Protocol (SNMP) is a network management protocol used to monitor the status of network equipment. It can provide information such as whether a device is functioning normally, how busy it is, and in some cases SNMP can be used to adjust the operating parameters. The vulnerability here is that the device at MY.NET.101.192 has the standard default community name (SNMP equivalent of a password) as "public". It is impossible to tell whether this is true for just the "read" community or the "read and write" community. At a minimum, anyone else running a program that can perform SNMP "gets" on the device at MY.NET.101.192 will be able to download the settings in that device. In a worst case the settings could be changed, possibly locking out the network monitoring device, and making the device in-operable. The monitoring device probably has its IP address assigned by Dynamic Host Control Protocol (DHCP), it is booted every morning, the address lease is renewed at boot time, therefore the IP address changes on a daily period. The attached drawing presents a plausible location for this device. Default SNMP community strings is noted as item 10 on the SANS list of Top Ten Security Issues http://www.sans.org/topten.htm



Known Issues:

For vulnerability information see the following:

CAN-2000-0885 Buffer overflows in Microsoft Network Monitor (Netmon) allow remote attackers to execute arbitrary commands via a long Browser Name in a CIFS Browse Frame, a long SNMP community name, or a long username or filename in an SMB session, aka the "Netmon Protocol Parsing" vulnerability. NOTE: It is highly likely that this candidate will be split into multiple candidates.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0885

CVE-1999-0294 All records in a WINS database can be deleted through SNMP for a denial of service.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0294

CVE-1999-0472 The SNMP default community name "public" is not properly removed in NetApps C630 Netcache, even if the administrator tries to disable it.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0472
CVE-2000-0221 The Nautica Marlin bridge allows remote attackers to cause a denial of service via a zero length UDP packet to the SNMP port.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0221
CVE-2000-0379 The Netopia R9100 router does not prevent authenticated users from modifying SNMP tables, even if the administrator has configured it to do so.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0379
CVE-2000-0515 The snmpd.conf configuration file for the SNMP daemon (snmpd) in HP-UX 11.0 is world writable, which allows local users to modify SNMP configuration or gain privileges.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0515
CAN-1999-0186 In Solaris, an SNMP subagent has a default community string that allows remote attackers to execute arbitrary commands as root, or modify system parameters.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0186
CAN-1999-0254 A hidden SNMP community string in HP OpenView allows remote attackers to modify MIB tables and obtain sensitive information.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0254
CAN-1999-0499 NETBIOS share information may be published through SNMP registry keys in NT.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0499
CAN-1999-0516 An SNMP community name is guessable.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0516
CAN-1999-0517 An SNMP community name is the default (e.g. public), null, or missing.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0517
CAN-1999-0792 ROUTERmate has a default SNMP community name which allows remote attackers to modify its configuration.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0792
The Phrack article - Network Management Protocol Insecurity: SNMPv1 http://www.2600.net/phrack/p50-07.html
Network Node Manager V6.1 Buffer Overflow Insecurity http://www.delphisplc.com/thinking/whitepapers/security/DST2K0012.txt

Detection:

```
08/15-19:59:52.855020  [**] SNMP public access [**] MY.NET.97.154:1049 -> MY.NET.101.192:161 to 08/15-20:18:19
08/16-19:38:33.208653  [**] SNMP public access [**] MY.NET.97.244:1042 -> MY.NET.101.192:161 to 08/16-19:47:31 Source Address changes
08/17-16:13:10.575484  [**] SNMP public access [**] MY.NET.98.177:1047 -> MY.NET.101.192:161 to 08/17-16:13:30 Source Address changes
08/19-11:50:50.750779  [**] SNMP public access [**] MY.NET.98.148:1039 -> MY.NET.101.192:161 to 08/19-11:55:30 Source Address changes
08/20-16:30:47.670389  [**] SNMP public access [**] MY.NET.97.246:1057 -> MY.NET.101.192:161 to 08/20-16:32:56 Source Address changes
08/20-19:40:10.476871  [**] SNMP public access [**] MY.NET.98.191:1046 -> MY.NET.101.192:161 to 08/20-19:52:25 Source Address changes
09/02-07:54:01.499976  [**] SNMP public access [**] MY.NET.98.181:1051 -> MY.NET.101.192:161 to 09/02-08:46:52 Source Address changes
09/03-12:42:35.351628  [**] SNMP public access [**] MY.NET.98.109:1045 -> MY.NET.101.192:161 to 09/03-17:39:19 Source Address changes
09/03-19:23:50.069324  [**] SNMP public access [**] MY.NET.98.114:1063 -> MY.NET.101.192:161 to 09/03-20:39:56 Source Address changes
09/07-18:42:57.351709  [**] SNMP public access [**] MY.NET.98.190:1071 -> MY.NET.101.192:161 to 09/07-18:45:07 Source Address changes
09/09-15:50:00.735356  [**] SNMP public access [**] MY.NET.97.206:1052 -> MY.NET.101.192:161 to 09/09-15:51:10 Source Address changes
09/10-15:27:45.937458  [**] SNMP public access [**] MY.NET.98.172:1042 -> MY.NET.101.192:161 to 09/10-20:10:16 Source Address changes
09/11-18:31:00.333609  [**] SNMP public access [**] MY.NET.97.217:1066 -> MY.NET.101.192:161 to 09/11-21:00:02 Source Address changes
09/12-17:48:35.407166  [**] SNMP public access [**] MY.NET.98.201:1048 -> MY.NET.101.192:161 to 09/12-18:31:24 Source Address changes
09/13-18:34:12.286961  [**] SNMP public access [**] MY.NET.98.171:1059 -> MY.NET.101.192:161 to 09/13-19:35:04 Source Address changes
```

3.2.1.7 SMB Name Wildcard

Snort Rule:

Custom - This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from an earlier version of Snort. It has been set up to capture SMB traffic to port 137 on MY.NET.X.X It will be similar to the following, although the actual rule may have more options

```
alert TCP !$HOME_NET any -> $HOME_NET 137 (msg:"SMB Name Wildcard ";)
```

Description: This detection rule has been implemented to capture NetBIOS Server Message Block (SMB) network traffic on port 137. Ports 135, 136, 137,138,and 139 are used by Microsoft NT, all versions, as well as Windows 95, and 98 for communication. There are various commands available such as , NBTSTAT -A (Target IP Address) which the educated intruder can use to obtain desirable information about a target host, in addition this port can be used to introduce various Virus infections and worms. There are a large number of exposures, from this group of ports. Global File Sharing, a feature enabled by having these ports open, is item 7 on the SANS list of Top Ten Security Issues http://www.sans.org/topten.htm

Known Issues:

CERT® Coordination Center Windows NT Configuration Guidelines Tech Tip http://www.cert.org/tech_tips/win_configuration_guidelines.html
CERT® Incident Note IN-2000-02 Exploitation of Unprotected Windows Networking Shares http://www.cert.org/incident_notes/IN-2000-02.html
CERT® Vulnerability Note VN-2000-03 Topic: Denial of Service Attack in NetBIOS Services http://www.cert.org/vul_notes/VN-2000-03.html
CAN-2000-0885 Buffer overflows in Microsoft Network Monitor (Netmon) allow remote attackers to execute arbitrary commands via a long Browser Name in a CIFS Browse Frame, a long SNMP community name, or a long username or filename in an SMB session, aka the "Netmon Protocol Parsing" vulnerability. NOTE: It is highly likely that this candidate will be split into multiple candidates. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0885
CAN-2000-0544 Windows NT and Windows 2000 hosts allow a remote attacker to cause a denial of service via malformed DCE/RPC SMBwriteX requests that contain an invalid
data length. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0544
CAN-1999-0520 A system-critical NETBIOS/SMB share has inappropriate access control. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0520
CAN-1999-0519 A NETBIOS/SMB share password is the default, null, or missing. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0519
CAN-1999-0518 A NETBIOS/SMB share password is guessable. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0518
CAN-1999-0495 A remote attacker can gain access to a file system using .. (dot dot) when accessing SMB shares. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0495
CVE-1999-0391 The cryptographic challenge of SMB authentication in Windows 95 and Windows 98 can be reused, allowing an attacker to replay the response and impersonate a user. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0391
CVE-1999-0225 Windows NT 4.0 allows remote attackers to cause a denial of service via a malformed SMB logon request in which the actual data size does not match the specified size. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0225
IN-2000-02 Exploitation of Unprotected Windows Networking Shares http://www.cert.org/incident_notes/IN-2000-02.html
SANS Intrusion Detection FAQ, Port 137 Scan http://www.sans.org/newlook/resources/IDFAQ/port_137.htm

Detection:

```
08/11-16:13:19.788046  [**] SMB Name Wildcard [**] 205.229.90.194:63733 -> MY.NET.181.37:137
08/11-16:15:33.790499  [**] SMB Name Wildcard [**] 131.118.254.222:137 -> MY.NET.6.7:137
08/11-16:16:04.289653  [**] SMB Name Wildcard [**] 168.143.29.9:137 -> MY.NET.60.17:137
08/11-16:16:05.792099  [**] SMB Name Wildcard [**] 168.143.29.9:137 -> MY.NET.60.17:137
08/11-16:26:04.299200  [**] SMB Name Wildcard [**] 62.136.168.18:1124 -> MY.NET.70.121:137
08/11-16:28:02.117131  [**] SMB Name Wildcard [**] 166.72.86.217:137 -> MY.NET.100.230:137
08/11-16:28:46.525879  [**] SMB Name Wildcard [**] 207.79.66.3:614 -> MY.NET.253.53:137
```

```
08/11-16:28:46.525941  [**] SMB Name Wildcard [**] 207.79.66.3:137 -> MY.NET.253.53:137
08/11-16:30:34.358284  [**] SMB Name Wildcard [**] 131.118.254.222:137 -> MY.NET.6.7:137
08/11-16:30:37.362453  [**] SMB Name Wildcard [**] 131.118.254.222:137 -> MY.NET.6.7:137
08/11-16:34:17.629689  [**] SMB Name Wildcard [**] 162.33.184.239:137 -> MY.NET.60.11:137
08/11-16:35:13.544620  [**] SMB Name Wildcard [**] 166.72.86.217:137 -> MY.NET.100.165:137 to 08/11-16:38:42
08/11-16:36:08.602748  [**] SMB Name Wildcard [**] 168.143.29.9:137 -> MY.NET.60.17:137
08/11-16:39:35.261511  [**] SMB Name Wildcard [**] 166.72.86.217:137 -> MY.NET.100.230:137
08/11-16:41:20.567074  [**] SMB Name Wildcard [**] 216.164.133.254:137 -> MY.NET.60.8:137
08/11-16:42:10.039852  [**] SMB Name Wildcard [**] 206.171.108.1:724 -> MY.NET.6.7:137
08/11-16:42:11.495774  [**] SMB Name Wildcard [**] 206.171.108.1:724 -> MY.NET.6.7:137
08/11-16:45:37.056832  [**] SMB Name Wildcard [**] 131.118.254.222:137 -> MY.NET.6.7:137
08/11-16:46:06.899175  [**] SMB Name Wildcard [**] 168.143.29.9:137 -> MY.NET.60.17:137
08/11-16:47:52.458244  [**] SMB Name Wildcard [**] 168.167.8.12:137 -> MY.NET.253.24:137
08/11-16:48:00.142286  [**] SMB Name Wildcard [**] 166.72.86.217:137 -> MY.NET.100.230:137
08/11-16:51:16.439948  [**] SMB Name Wildcard [**] 64.7.58.194:137 -> MY.NET.20.10:137
08/11-16:51:17.369342  [**] SMB Name Wildcard [**] 24.28.62.226:1975 -> MY.NET.70.121:137
08/11-16:52:02.576485  [**] SMB Name Wildcard [**] 209.150.98.231:137 -> MY.NET.130.91:137
08/11-16:52:02.576547  [**] SMB Name Wildcard [**] 209.150.98.231:137 -> MY.NET.130.91:137
08/11-16:56:11.373867  [**] SMB Name Wildcard [**] 168.143.29.9:137 -> MY.NET.60.17:137
08/15-20:00:00.271636  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137 to 08/20-19:50:11
09/02-07:54:03.629167  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137 to 09/13-19:29:55
08/19-22:57:41.121096  [**] SMB Name Wildcard [**] 129.37.161.200:137 -> MY.NET.100.130:137 to 08/19-22:58:42
09/09-10:52:16.881113  [**] SMB Name Wildcard [**] 129.37.160.81:137 -> MY.NET.100.130:137
08/18-12:13:28.981943  [**] SMB Name Wildcard [**] 4.17.88.66:137 -> MY.NET.6.15:137
```

3.2.1.8 Null scan!

Snort Rule

```
alert TCP !$HOME_NET any -> $HOME_NET any (msg:"IDS4 - SCAN-NULL Scan"; flags: 0; seq: 0; ack: 0; )
```

Description: This rule has been created to detect packets that have a number of normal packet features set to zero values. The packet has no normal TCP flags set, the flags which could be set are U (Urgent), ACK (Acknowledge),PSH (Push), RST (Reset), SYN (Synchronize), FIN (Finish), and the sequence number is set to zero. A TCP session starts with a three way handshake, it functions as follows. A node (client) wishing to communicate with another node (server) establishes the TCP connection with a TCP packet containing a SYN, the initial sequence number it wishes to use, to the well known port it wishes to connect to, the Maximum Segment Size (MSS) and Maximum Transmission Unit (MTU) size. If the server is willing and able to establish a connection, it responds with a packet, which contains a SYN-ACK, increments the client's initial sequence number by 1, supplies the server's initial sequence number, MSS and MTU. If that port is not active on the server, a reset is sent instead to the client node. The client responds to the server's SYN with an ACK, and increments the servers initial sequence number by 1. The initial sequence numbers are incremented as appropriate to each node, with each transfer of data. Communication then continues until the session is complete. The start of session tear down is initiated with a FIN from, which ever node has completed first. The other node responds with an ACK, and if it has completed its portion of the session, as well, it responds with a FIN, if it has not completed it's side of the communication it continues until it is complete. The node that has sent its FIN will continue to respond even though it may already have sent a FIN until both ends of the conversation have sent a FIN to which the other node responds with an ACK. The session is then completely torn down. As can be observed from the discussion this packet cannot exist in normal TCP communication. At least one of the items being tracked must be set, generally 2, most often being a sequence number and ACK, or Sequence number and Fin.

Known Issues: The Null Scan is used to map Operating Systems (O/S), as different O/S's will respond in a different manner to illegal combinations of flags and sequence numbers in TCP packets. The responses have already been tabulated, and are implemented in a program named NMAP. See http://www.insecure.org/nmap/ for the program. There is an article called "*Remote OS detection via TCP/IP Stack Fingerprinting*" by Fyodor, at, http://www.insecure.org/nmap/nmap-fingerprinting-article.html , the section *TCP Options* describes how the Null Scan was implemented in NMAP.
A paper by Ofir Arkin, *Network Scanning Techniques*, section 3.1.3.1.4 NULL, gives another description. It is available at http://www.sys-security.com/html/papers.html

Detection:

```
09/09-04:36:24.118542  [**] Null scan! [**] 24.6.140.249:1336 ->    MY.NET.130.190:20 @Home Network (NETBLK-RDC1-MD-1) Redwood City, CA 94063
08/19-15:06:39.505723  [**] Null scan! [**] 24.8.241.127:1049 ->    MY.NET.70.217:6699 @Home Network (NETBLK-RDC1-TX -23) Redwood City, CA 94063
08/15-16:25:18.236825  [**] Null scan! [**] 24.13.104.131:4190 ->   MY.NET.253.112:443 @Home Network (NETBLK-RDC1-MD-6) Redwood City, CA 94063
09/05-17:58:59.801817  [**] Null scan! [**] 24.19.101.91:1226 ->    MY.NET.222.46:6699 @Home Network (NETBLK-ATHOME) Redwood City, CA 94063
09/09-19:28:57.104613  [**] Null scan! [**] 24.22.125.94:3224 ->    MY.NET.223.38:6699 @Home Network (NETBLK-INDNPLS1-IN-4) Redwood City, CA 94063
08/17-00:07:03.380164  [**] Null scan! [**] 24.23.198.174:2523 ->   MY.NET.217.46:1396 @Home Network (NETBLK-BB1-SVNNHA-2) Redwood City, CA 94063
09/13-10:09:42.525112  [**] Null scan! [**] 24.28.33.193:6699 ->    MY.NET.224.134:3087 TimeWarnerCable-Road-Runner-TAMPA-MCR24A-24-28-33-0-to-33-
0
09/09-23:29:25.490480  [**] Null scan! [**] 24.29.7.199:1083 ->     MY.NET.218.94:6699 TimeWarnerCable-Road-Runner-AUS-RroundRock-MCR2-24-29-6-0-to-7-0
09/06-16:33:05.624173  [**] Null scan! [**] 24.72.8.78:2276 ->      MY.NET.213.58:4279 Cable Regina (NETBLK-CABLER) Regina, SK, Canada
09/06-05:06:11.002604  [**] Null scan! [**] 24.72.23.136:3254 ->    MY.NET.213.58:4407 Cable Regina (NETBLK-CABLER) Regina, SK, Canada
09/14-12:44:05.485501  [**] Null scan! [**] 24.91.58.197:100 ->     MY.NET.221.230:1152 Continental Cablevision (NETBLK-CVSN-CCNE-2BL) Boston, MA
09/11-11:25:19.046208  [**] Null scan! [**] 24.92.174.232:1032 ->   MY.NET.217.130:6688 TimeWarnerCable-RoadRunner-TampaBay-C5Fortune (NETBLK-RRTAM-
C5F)
09/14-16:18:53.799118  [**] Null scan! [**] 24.92.188.4:4269 -> MY.NET.106.164:6699 TimeWarnerCable-RoadRunner-TampaBay-S8EastBay (NETBLK-RRTAM-S8EB)
```

```
08/20-17:22:40.162264  [**] Null scan! [**] 24.112.241.246:1866 ->  MY.NET.201.58:6699 Rogers@Home Bloor (NETBLK-ON-ROG-BLOOR-2)
09/06-13:58:22.763407  [**] Null scan! [**] 24.113.80.28:1993 ->    MY.NET.203.110:1464 < See type 3 analysis at the end of this section
09/09-01:04:12.702304  [**] Null scan! [**] 24.115.96.111:1584 ->   MY.NET.222.198:20 Rogers@Home Surrey (NETBLK-BC-ROG-1-1SURREY-5)
09/06-19:35:28.946592  [**] Null scan! [**] 24.160.189.151:2093 ->  MY.NET.220.206:6699 Road Runner-Columbusnrm2mcr-24-160-188-0-24-160-191-255
08/18-02:22:16.211087  [**] Null scan! [**] 24.164.181.31:6699 ->   MY.NET.217.26:1159 ServiceCo LLC - Road Runner (NET-ROAD-RUNNER-5)
09/11-04:49:39.789570  [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.13:23 to 09/11-05:16:34 all port 23
09/11-05:05:15.617311  [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.114:135 @Home Network (NETBLK-BLTMMD1-MD-1) Redwood City, CA 94063
09/11-05:08:27.725009  [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.146:77 @Home Network (NETBLK-BLTMMD1-MD-1) Redwood City, CA 94063
09/11-05:17:08.347707  [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.225:23 @Home Network (NETBLK-BLTMMD1-MD-1) Redwood City, CA 94063
09/11-05:17:36.813698  [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.226:2000 @Home Network (NETBLK-BLTMMD1-MD-1) Redwood City, CA 94063
09/11-05:18:51.565916  [**] Null scan! [**] 24.180.134.156:50110 -> MY.NET.208.241:23 @Home Network (NETBLK-BLTMMD1-MD-1) Redwood City, CA 94063
09/06-16:50:56.129540  [**] Null scan! [**] 24.180.196.93:6699 ->   MY.NET.217.250:2493 @Home Network (NETBLK-BLTMMD1-MD-3) Redwood City, CA 94063
08/16-19:17:36.848025  [**] Null scan! [**] 24.200.201.223:1635 ->  MY.NET.162.183:6346 Videotron Ltee (NETBLK-VL-D-MF-18C8C900) Montreal, QC
Canada
09/12-18:56:17.389378  [**] Null scan! [**] 24.201.116.208:1242 ->  MY.NET.209.210:6000 Videotron Ltee (NETBLK-VL-D-MD-18C97400) Montreal, QC
Canada
09/06-10:45:09.693242  [**] Null scan! [**] 24.232.79.188:1042 ->   MY.NET.206.114:6699 Cablevision S.A. (NETBLK-DIALUP-4) Buenos Aires, AR
08/17-05:54:29.219132  [**] Null scan! [**] 128.61.59.79:1338 ->    MY.NET.217.34:6699 Georgia Institute of Technology (NET-GATECH-EDU) Atlanta, GA
09/08-20:18:44.900710  [**] Null scan! [**] 128.61.105.106:6699 ->  MY.NET.218.202:49455 Georgia Institute of Technology (NET-GATECH-EDU) Atlanta, GA
09/03-21:20:23.811989  [**] Null scan! [**] 128.153.151.115:1410 -> MY.NET.205.50:832 Clarkson University (NET-CLARKSON) Potsdam, NY
09/13-13:27:15.448853  [**] Null scan! [**] 128.138.14.148:1417 ->  MY.NET.179.52:6699 University of Colorado (NET-COLORADO) Boulder, CO
09/06-11:59:43.392789  [**] Null scan! [**] 128.194.51.187:3223 ->  MY.NET.210.114:6688 Texas A&M University (NET-TAMU-NET) College Station, Texas
09/08-14:00:26.231456  [**] Null scan! [**] 128.226.152.34:1584 ->MY.NET.206.114:6699 State University of New York(NET-BINGHAMTON) Binghamton, NY
09/06-12:40:15.686069  [**] Null scan! [**] 129.93.214.47:1168 ->   MY.NET.223.186:6699 University of Nebraska-Lincoln (NET-HUSKERNET) Lincoln NE
09/03-11:44:47.774295  [**] Null scan! [**] 129.59.24.21:1540 ->    MY.NET.204.126:6699 Vanderbilt University (NET-VANDERBILT) Nashville, TN
09/12-01:08:16.376861  [**] Null scan! [**] 130.49.220.26:1248 ->   MY.NET.226.6:6699 University of Pittsburgh (NET-U-PITT) Pittsburgh, PA
09/12-01:27:10.803183  [**] Null scan! [**] 130.239.142.167:1180 -> MY.NET.223.58:6699 Umea University (NET-UMUNET) Umea, SE
08/17-09:04:37.256707  [**] Null scan! [**] 130.149.41.70:1110 ->   MY.NET.217.46:994 Technische Universitaet Berlin (NET-TUB) Berlin GERMANY
08/17-12:41:10.144753  [**] Null scan! [**] 130.149.41.70:0 ->      MY.NET.217.46:1062 Technische Universitaet Berlin (NET-TUB) Berlin GERMANY
08/17-08:43:32.411061  [**] Null scan! [**] 130.239.11.230:6699 ->  MY.NET.181.173:4554 Umea University (NET-UMUNET) Umea, SE
09/05-17:11:57.165483  [**] Null scan! [**] 131.155.192.220:52782 ->MY.NET.5.7:21 Eindhoven University of Technology (NET-TUEINDHOVEN)
NETHERLANDS
09/05-11:14:09.669789  [**] Null scan! [**] 132.199.220.223:2675 -> MY.NET.205.26:6699 University of Regensburg (NET-UNIR-LAN) Regensburg, DE
08/15-00:49:41.935984  [**] Null scan! [**] 137.82.136.39:6699 ->   MY.NET.97.150:1340 University of British Columbia (NET-UBC) Vancouver, Canada
09/12-15:03:22.834735  [**] Null scan! [**] 139.91.171.50:1460 -> MY.NET.211.234:6699 Foundation of Research and Technology Hellas (NET-FORTH) Crete,
GREECE
09/06-17:17:53.195699  [**] Null scan! [**] 141.40.205.133:6699 ->  MY.NET.224.34:1693 Leibniz Rechenzentrum (LRZ) (NET-LRZ-WEIHSTEPH) Muenchen,
DE
09/11-17:49:05.140901  [**] Null scan! [**] 150.216.127.179:6699 -> MY.NET.206.66:1865 East Carolina University (NET-ECUNET) Greenville, NC 27834
09/08-18:56:39.751099  [**] Null scan! [**] 151.196.73.119:37196 -> MY.NET.253.112:22 Windermer Information Systems Technology (NETBLK-BA-151-196-73-
                                                                                                              64-128) Annapolis, MD
09/10-10:44:47.461045  [**] Null scan! [**] 153.19.25.156:1079 ->   MY.NET.223.42:6346 POLIP (NET-TASKPOLIP) Uniwersytet Gdanski Gdansk, PL
```

```
09/09-21:11:13.643976  [**] Null scan! [**] 62.2.64.86:1070 ->      MY.NET.218.10:6699 CABLECOM-MAIN-NET Zuerich, CH
09/13-00:05:21.928777  [**] Null scan! [**] 62.10.136.40:1159 ->    MY.NET.212.134:2 TISCALINET Cagliari, Italy
09/08-12:59:46.211994  [**] Null scan! [**] 63.144.227.21:3115 ->   MY.NET.208.190:5501 Sterling University Grove (NETBLK-QWEST-63-144-227-0) Tallahassee, Fl
09/02-16:25:42.155352  [**] Null scan! [**] 63.226.208.41:28517 ->  MY.NET.253.41:22 U S WEST Communications Svcs, Inc. (NETBLK-USW-INTERACT99)
                                                                                                        Minneapolis, MN
08/17-09:46:44.698226  [**] Null scan! [**] 193.251.71.243:6699 ->  MY.NET.146.68:1387 France Telecom IP2000 ADSL BAS Rennes, France
09/06-04:15:37.743959  [**] Null scan! [**] 194.94.18.43:1546 ->    MY.NET.220.42:6346 ROGGENKAMP Network for student appartments Bielefeld, Bielefeld, Germany
09/09-21:22:02.605745  [**] Null scan! [**] 194.237.99.150:1511 ->  MY.NET.223.38:6699 MOTALA-KOMUN IT-Enheten/Data Motala, Sweden
09/10-09:23:42.094521  [**] Null scan! [**] 195.132.204.48:1737 ->  MY.NET.220.154:6699  CYBERCABLE, Lyonnaise Communication, PARIS, FRANCE
09/07-09:57:12.506948  [**] Null scan! [**] 195.150.132.211:1117 -> MY.NET.202.158:6346  PETROINFORM, Krakow, PL
09/03-01:56:22.365955  [**] Null scan! [**] 200.145.151.163:2195 -> MY.NET.221.114:6699 RNP (Brazilian Research Network) (NETBLK-BRAZIL-BLK2) Sao Paulo, BR
08/11-20:18:48.417718  [**] Null scan! [**] 200.52.201.4:1409 ->MY.NET.217.222:6699 MEGA CABLE S.A. DE C.V. (NETBLK-MEGACABLE-RED-1)
GUADALAJARA,MX
08/16-01:42:30.220172  [**] Null scan! [**] 207.151.147.201:58190 ->MY.NET.60.8:21 Los Nettos (NETBLK-LOS-NETTOS-BLK3) USC Info Sciences Marina del Rey,
CA
09/14-04:42:35.562188  [**] Null scan! [**] 207.230.248.254:6699 -> MY.NET.208.18:4617 In2net Network Inc. (NETBLK-IN2NETT-BLK-1) Vancouver, BC Canada
09/10-23:14:41.993432  [**] Null scan! [**] 211.111.108.136:6346 -> MY.NET.224.34:3034 KRNIC-KR-25 Korea Network Information Center, Seoul, Republic of Korea
09/03-16:06:17.788917  [**] Null scan! [**] 212.33.70.83:1190 ->    MY.NET.206.66:6688 Prizmanet Elektronik Yayincilik  Hizm. San. ve Tic. A.S., Istanbul-Turkey
09/11-19:23:44.441830  [**] Null scan! [**] 213.6.43.74:2148 ->     MY.NET.208.114:6399  MOBILCOM-CITYLINE-NET Dialpool, Duesseldorf, Germany
09/02-10:47:37.939613  [**] Null scan! [**] 213.56.48.243:1800 ->   MY.NET.201.198:4704 FR-OLEANE-BLOCK-7, Oleane SA, Gennevilliers, France
09/11-03:09:17.614037  [**] Null scan! [**] 216.161.190.169:4995 -> MY.NET.226.54:4913 U S WEST Interact Services (NETBLK-USW-INTERACT98) Minneapolis, MN
09/05-22:03:52.540729  [**] Null scan! [**] 216.63.200.250:1174->MY.NET.203.106:6699 WCHTKS BASIC DSL RBACK1 216.63.200.0 (NETBLK-SBCIS30284) Plano,
TX
09/09-20:42:00.111596  [**] Null scan! [**] 216.123.60.71:1130 ->   MY.NET.202.134:3105 NETCOM Canada Inc. (NETBLK-NETCOM-CA-BLK4) Toronto, ON, Canada
```

24.180.134.156 Also recorded as an Nmap TCP Ping scan of MY.NET.208.X and Probable NMAP fingerprint attempt

@Home Network (NETBLK-BLTMMD1-MD-1)
Redwood City, CA 94063 USA
Netname: BLTMMD1-MD-1
Netblock: **24.180.128.0 - 24.180.143.255**
Coordinator: Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

The sources of the vast majority, of the scans are from dialup or cable modem users. It is impossible to tell if the scan was initiated by the owner of that machine or by a 3rd party that "owns" that machine. In any case, it is quite likely that the machines that belong to dialup or cable modem users, are poorly secured, and therefore likely to be used in a Distributed Denial of Service attack. At this point little effective action can be taken, other than identifying that @ Home Network was a major player, and therefore needs to be more closely watched. A longer term solution will require building a database of identified frequent problem sites and addresses.

3.2.1.9 NMAP TCP ping!

Snort Rule

```
alert TCP !$HOME_NET any -> $HOME_NET any (msg:"IDS28 - PING NMAP TCP"; flags: A; ack: 0; )
```

Description: The detection rule has been created to detect the specific ping footprint of the Operating System fingerprinting tool named Nmap. The following is taken from "*Nmap network security scanner man page*" available at http://www.insecure.org/nmap/nmap_manpage.html
"*Nmap is designed to allow system administrators and curious individuals to scan large networks to determine which hosts are up and what services they are offering. Nmap supports a large number of scanning techniques such as: UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), Reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep, and Null scan."* The tool Nmap is available at http://www.insecure.org/nmap/
There is an article called "*Remote OS detection via TCP/IP Stack Fingerprinting*" by Fyodor, at, http://www.insecure.org/nmap/nmap-fingerprinting-article.html , the paper describes how Nmap was implemented.
A paper by Ofir Arkin, *Network Scanning Techniques*, in section *4.3 TCP/IP Stack Fingerprinting* gives another description of the tricks that Nmap uses. It is available at http://www.sys-security.com/html/papers.html
ID FAQ - What is nmap and what can it do? http://www.sans.org/newlook/resources/IDFAQ/What_is_nmap.htm
Information Security Reading Room - Nmap - The Tool, It's Author and It's Implications http://www.sans.org/infosecFAQ/nmap.htm

Known Issues:

CAN-2000-0324 pcAnywhere 8.x and 9.x allows remote attackers to cause a denial of service via a TCP SYN scan, e.g. by nmap.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0324

Once an intruder can determine what operating system is running, then the exploits for that operating system can be put into play.

Cisco Advisory Notices: http://www.cisco.com/warp/public/707/advisory.html
Compaq (Digital UNIX) http://www.compaq.com/support
FreeBSD http://www.freebsd.org/security/
HP HP-UX For the US, Canada, Asia-Pacific, & Latin America: http://us-support.external.hp.com
For Europe:http://europe-support.external.hp.com
To Retrieve a Security Patch Matrix: ftp://us-ffs.external.hp.com/export/patches/hp-ux_patch_matrix/
IBM AIX http://techsupport.services.ibm.com/support/rs6000.support/downloads
http://techsupport.services.ibm.com/rs6k/fixes.html
Microsoft http://www.microsoft.com/technet/security/current.asp
Nortel Networks (Bay Networks): http://www12.nortelnetworks.com/ (you will need to know your equipment type, and may need an account)
SCO (OpenServer and Unixware) http://www.sco.com/security/ (Security Bulletins and Patches)
http://www.sco.com/support/ftplists/index.html (General OS patches)
Sun Solaris http://sunsolve.sun.com (Recommended & Security Patches)
SGI http://support.sgi.com
Linux
Caldera http://www.caldera.com/support/security/
Debian http://www.debian.org/security/index.en.html
Mandrake: http://www.linux-mandrake.com/en/fupdates.php
Red Hat: http://www.redhat.com/support/updates.html
SuSe http://www.suse.com/support/download/updates/index.html

http://www.suse.de/en/support/security/index.html

Detection:

```
08/11-12:57:26.064901  [**] NMAP TCP ping! [**] 205.128.11.157:80 ->      MY.NET.1.8:53
08/11-12:57:26.064946  [**] NMAP TCP ping! [**] 205.128.11.157:53 ->      MY.NET.1.8:53
08/11-22:52:09.700179  [**] NMAP TCP ping! [**] 205.128.11.157:80 ->      MY.NET.1.9:53
08/11-22:52:09.700228  [**] NMAP TCP ping! [**] 205.128.11.157:53 ->      MY.NET.1.9:53
08/20-00:42:09.533868  [**] NMAP TCP ping! [**] 205.128.11.157:80 ->      MY.NET.1.10:53
08/20-00:42:09.534100  [**] NMAP TCP ping! [**] 205.128.11.157:53 ->      MY.NET.1.10:53
HeadHunter.net (NETBLK-INFLOW-HED2)
   Atlanta, GA 30309 USA
   Netname: INFLOW-HED2
   Netblock: 205.128.11.0 - 205.128.11.127
   Coordinator: Powers, Adam  (AP336-ARIN)  apowers@inflow.com
                        404.873.8397
```

```
08/11-23:42:06.552435  [**] NMAP TCP ping! [**] 209.218.228.201:80 ->    MY.NET.1.8:53
08/11-23:42:06.552484  [**] NMAP TCP ping! [**] 209.218.228.201:53 ->    MY.NET.1.8:53
RND Networks (NETBLK-ATWORK-RND2)
   Mahwah, NJ 07430 AUS
   Netname: ATWORK-RND2
   Netblock: 209.218.228.128 - 209.218.228.255
   Coordinator:Leung, Stephen  (SL109-ARIN)  stephen@RNDNETWORKS.COM
                           201-512-9771 x 228
```

```
08/16-01:42:30.231975  [**] NMAP TCP ping! [**] 207.151.147.201:58192 -> MY.NET.60.8:21
08/16-01:42:30.235007  [**] NMAP TCP ping! [**] 207.151.147.201:58194 -> MY.NET.60.8:31898
08/16-01:42:51.248359  [**] NMAP TCP ping! [**] 207.151.147.201:58194 -> MY.NET.60.8:40833
USC Information Sciences Institute PO 11565
   Marina del Rey, CA 90295 USA
   Netname: LOS-NETTOS-BLK3
   Netblock: 207.151.0.0 - 207.151.255.255
   Coordinator:  LosNettos Hostmaster  (LH-ORG-ARIN)  hostmaster@LN.NET
                         310.822.1511 x198
```

```
08/17-12:15:42.997099  [**] NMAP TCP ping! [**] 213.8.52.189:80 ->         MY.NET.60.14:80
inetnum:     213.8.0.0 - 213.8.7.255
netname:     EURONET
descr:       Ramat-Gan pop.
descr:       send SPAM and ABUSE complaints to abuse@inter.net.il
```

```
person:      Dudi Davidesko
address:     Internet Gold - Euronet Golden Lines LTD.
address:     Park Sibel, Rosh Haain
address:     Israel
phone:       +972-3-9020020
fax-no:      +972-3-9024222
e-mail:      dudi@xchange.wan.inter.net.il
```

```
08/19-08:17:02.783410  [**] NMAP TCP ping! [**] 192.55.91.36:62449 ->    MY.NET.5.111:42185
08/20-15:12:12.562098  [**] NMAP TCP ping! [**] 192.55.91.36:58331 ->    MY.NET.5.29:40045
NASA Lewis Network Control Center (NET-LERC-CRAYNET2)
   Cleveland, OH 44135 US
   Netname: LERC-CNET2
   Netnumber: 192.55.91.0
   Coordinator: NASA - John H. Glenn Research Centerat Lewis Field  (ZN14-ARIN)  gnoc@grc.nasa.gov
                          (216) 433-9850  pager-(216) 549-0650
```

```
09/02-13:01:18.139069  [**] NMAP TCP ping! [**] 2.2.2.2:80 -> MY.NET.60.14:80 The source address is a Reserved address, obviously spoofed.
```

```
09/02-16:25:40.522963  [**] NMAP TCP ping! [**] 63.226.208.41:28521 ->  MY.NET.253.41:1
U S WEST Communications Svcs, Inc. (NETBLK-USW-INTERACT99)
   Minneapolis, MN 55413 USA
   Netname: USW-INTERACT99
   Netblock: 63.224.0.0 - 63.231.255.255
 Coordinator: U S WEST ISOps  (ZU24-ARIN)  abuse@uswest.net
                         612-664-4689
```

```
09/02-07:26:07.823169  [**] NMAP TCP ping! [**] 202.187.24.3:80 ->          MY.NET.60.14:80
09/07-03:52:20.386042  [**] NMAP TCP ping! [**] 202.187.24.3:80 ->          MY.NET.179.77:80
09/12-22:36:20.070878  [**] NMAP TCP ping! [**] 202.187.24.3:80 ->          MY.NET.1.3:53
inetnum              202.187.24.0 - 202.187.24.255
         netname              JARING-UNITAR2
         descr               Universiti Tun Abdul Razak
         descr               47300 Petaling Jaya Selangor MY
         notify              dbmon@apnic.net, inverse
         notify              ip-request@jaring.my, inverse
```

```
09/08-18:56:36.842935  [**] NMAP TCP ping! [**] 151.196.73.119:37198 -> MY.NET.253.112:22
09/08-18:56:39.753348  [**] NMAP TCP ping! [**] 151.196.73.119:37200 -> MY.NET.253.112:30462
09/08-18:57:06.505923  [**] NMAP TCP ping! [**] 151.196.73.119:48518 -> MY.NET.253.112:32999
Dixie Printing & Packaging (NETBLK-DIXIE-196-73)
```

```
   7358 Baltimore-Annapolis Blvd
   Glen Burnie, MD 21061 USA
   Netname: DIXIE-196-73
   Netblock: 151.196.73.0 - 151.196.73.63
   Coordinator: Ongoing Business Support Services  (OBS-ORG-ARIN)  business-support@MERCURY.BALINK.COM
      800-475-7840 Fax- 703-453-6770

09/11-04:48:14.514481  [**] NMAP TCP ping! [**] 24.180.134.156:50114 -> MY.NET.208.1:35829 to
09/11-05:18:51.575120  [**] NMAP TCP ping! [**] 24.180.134.156:50112 -> MY.NET.208.241:23  <<Scan of 208, subnet various ports
@Home Network (NETBLK-BLTMMD1-MD-1)
   Redwood City, CA 94063 US
   Netname: BLTMMD1-MD-1
   Netblock: 24.180.128.0 - 24.180.143.255
   Coordinator: Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
                        (650) 556-5599
```

3.2.1.10 SUNRPC highport access!

Snort Rule

Custom - This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from an earlier version of Snort. It has been set up to capture what appears to be successful access to the ports identified as SUNRPC Highports. The rule would be similar to the one below. This rule would need to be supplied by GIAC Enterprises, in to better understand what differentiates it from the *Attempted SUN RPC highport access* above.

```
alert TCP !$HOME_NET any -> $HOME_NET 32771:32780 (msg:"SUNRPC highport access!"; ack: 1; [options here to
discriminate between the two])
```

Description: In this discussion this Alert will be treated the same as Attempted SUN RPC highport access, above, except one has to assume that there was success, and therefore this is a more significant issue. This rule has been created to detect the individuals that are successful in compromising the Remote Procedure Call (RPC) programs directly. The normal port for the portmapper service is port 111. When the portmapper service is quizzed, on port 111, it would identify what RPC services are running on that machine, those services are generally located in the 32700 port area. In this case the individual is successful in going directly for the RPC services without accessing the portmapper. This method of accessing the services more directly, is considered more “stealthy” than asking the portmapper, and then accessing the ports, after being informed by the RPC service where they are. In the Sun implementation of UNIX, it is generally recognized that the RPC ports start at 32771, hence the attack (and Snort rule) is specifically targeting Sun machines. The ports which are open, and successfully attached to, will represent services running which will then be the service to attack.

Known Issues:

CVE-1999-0008 Buffer overflow in NIS+, in Sun's rpc.nisd program http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0008
CVE-1999-0212 Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0212
CVE-1999-0320 SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0320
CVE-1999-0974 Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad service. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0974
CAN-1999-0195 Denial of service in RPC portmapper allows attackers to register or unregister RPC services or spoof RPC services using a spoofed source IP address such as 127.0.0.1. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0195
CAN-1999-0568 rpc.admind in Solaris is not running in a secure mode. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0568
CAN-1999-0795 The NIS+ rpc.nisd server allows remote attackers to execute certain RPC calls without authentication to obtain system information, disable logging, or modify caches. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0795
CERT® Advisory CA-2000-17 Input Validation http://www.cert.org/advisories/CA-2000-17.html
CERT® Incident Note IN-2000-10 http://www.cert.org/incident_notes/IN-2000-10.html
CERT® Summary CS-2000-03 http://www.cert.org/summaries/CS-2000-03.html
CERT® Incident Note IN-99-04 Similar Attacks Using Various RPC Services http://www.cert.org/incident_notes/IN-99-04.html
CA-99-08 - Buffer Overflow Vulnerability in rpc.cmsd http://www.cert.org/advisories/CA-99-08-cmsd.html
CA-99-05 - Vulnerability in statd exposes vulnerability in automountd http://www.cert.org/advisories/CA-99-05-statd-automountd.html
CA-98.11 - Vulnerability in ToolTalk RPC Service http://www.cert.org/advisories/CA-98.11.tooltalk.html
ID FAQ - The trouble with RPCs http://www.sans.org/newlook/resources/IDFAQ/trouble_RPCs.htm
RFC for RPC Binding Protocols for ONC RPC Version 2 http://www.ietf.org/rfc/rfc1833.txt

Detection

```
09/02-09:39:11.608534  [**] SUNRPC highport access! [**] 212.204.196.241:857 -> MY.NET.6.15:32771
inetnum:     212.204.196.241 - 212.204.196.250
```

```
descr:      WeBHold
person:      Luc Willemars
address:     Zaandam
address:     NL
phone:       +31 75 6141212
e-mail:     Luc.Willemars@WeBHold.nl
```

```
09/06-23:10:10.012419  [**] SUNRPC highport access! [**] 193.64.205.17:56880 -> MY.NET.211.2:32771
netname:     TARVEASUNNOT-FI-2
descr:      Tarveasunnot Oy
descr:       FI-02630 ESPOO FI
route:      193.64.0.0/15
descr:      KPNQwest Finland 193.64/15 superblock
person:      Janne Jaaskelainen
address:     Tarveasunnot Oy
phone:       +358 9 54919390
notify:     hostmaster@kpnqwest.fi
```

```
09/07-21:10:18.892811  [**] SUNRPC highport access! [**] 207.29.195.22:2646 -> MY.NET.211.2:32771
NetReach, Inc. (NETBLK-NRCH-NETREACH-NET)
Ambler, PA 19002 US
Netname: NRCH-NETREACH-NET
Netblock: 207.29.192.0 - 207.29.207.255
Coordinator: NetReach, Inc.  (IN17-ARIN)  admin@netreach.net
                   215.283.2300
```

```
09/08-16:34:54.280910  [**] SUNRPC highport access! [**] 205.188.4.42:5190 -> MY.NET.210.2:32771
America Online, Inc (NETBLK-AOL-DTC)
Sterling, VA 20166US
Netname: AOL-DTC
Netblock: 205.188.0.0 - 205.188.255.255
Coordinator: America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
                     703-265-4670
09/11-21:24:53.037663  [**] SUNRPC highport access! [**] 209.10.41.242:21 -> MY.NET.211.2:32771
Globix Corporation (NETBLK-GLOBIXBLK3)
NY, NY 10012
Netname: GLOBIXBLK3
Netblock: 209.10.0.0 - 209.11.159.255
Coordinator: Hostmaster, Globix Corporation  (GCH2-ARIN)  arin-admin@GLOBIX.NET
                    212.334.8500 (FAX) 212.334.8615
```

3.2.1.11 Queso fingerprint

Snort Rule

```
alert TCP !$HOME_NET any -> $HOME_NET any (msg:"IDS29 - SCAN-Possible Queso Fingerprint attempt"; flags: S12; )
```

Description: The Queso program is an older version of an Operating System (O/S) finger printing program. It identifies remote systems by looking at the response received by sending packets with different characteristics and TCP flags to an open port and comparing the responses to an internal table. It needs to have an open port to talk to. To prevent Queso from finger printing the O/S, limit the number of open services, and place filters on the Intrusion Detection System watching those services. Documentation can be found at http://www.apostols.org/projectz/queso/
There is an article called "*Remote OS detection via TCP/IP Stack Fingerprinting*" by Fyodor, at, http://www.insecure.org/nmap/nmap-fingerprinting-article.html , the paper describes how Nmap was implemented, a tool similar to Queso, but a newer implementation. The same issues are still true for Queso.
A paper by Ofir Arkin, *Network Scanning Techniques*, in section *4.3 TCP/IP Stack Fingerprinting* gives another description of the tricks that Nmap uses. It is available at http://www.sys-security.com/html/papers.html

Known Issues:

CAN-2000-0324 pcAnywhere 8.x and 9.x allows remote attackers to cause a denial of service via a TCP SYN scan, e.g. by nmap.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0324
Once an intruder can determine what operating system is running, then the exploits for that operating system can be put into play.

Cisco Advisory Notices: http://www.cisco.com/warp/public/707/advisory.html
Compaq (Digital UNIX) http://www.compaq.com/support
FreeBSD http://www.freebsd.org/security/
HP HP-UX For the US, Canada, Asia-Pacific, & Latin America: http://us-support.external.hp.com
For Europe:http://europe-support.external.hp.com
To Retrieve a Security Patch Matrix: ftp://us-ffs.external.hp.com/export/patches/hp-ux_patch_matrix/
IBM AIX http://techsupport.services.ibm.com/support/rs6000.support/downloads
http://techsupport.services.ibm.com/rs6k/fixes.html
Microsoft http://www.microsoft.com/technet/security/current.asp
Nortel Networks (Bay Networks): http://www12.nortelnetworks.com/ (you will need to know your equipment type, and may need an account)
SCO (OpenServer and Unixware) http://www.sco.com/security/ (Security Bulletins and Patches)
http://www.sco.com/support/ftplists/index.html (General OS patches)
Sun Solaris http://sunsolve.sun.com (Recommended & Security Patches)
SGI http://support.sgi.com
Linux
Caldera http://www.caldera.com/support/security/
Debian http://www.debian.org/security/index.en.html
Mandrake: http://www.linux-mandrake.com/en/fupdates.php
Red Hat: http://www.redhat.com/support/updates.html
SuSe http://www.suse.com/support/download/updates/index.html
http://www.suse.de/en/support/security/index.html

Detection:

```
08/15-15:30:15.258499  [**] Queso fingerprint [**] 216.123.63.13:4232 ->   MY.NET.75.106:1488
```

```
08/17-15:49:40.548646  [**] Queso fingerprint [**] 130.149.41.70:1068 ->   MY.NET.217.46:994
09/03-02:17:47.893294  [**] Queso fingerprint [**] 24.19.244.80:6699 ->    MY.NET.162.200:3889
09/03-20:49:29.075604  [**] Queso fingerprint [**] 24.24.137.232:1342 ->    MY.NET.219.194:6355
09/05-09:00:54.631991  [**] Queso fingerprint [**] 24.3.161.193:32814 ->    MY.NET.145.9:110
09/06-09:31:55.767609  [**] Queso fingerprint [**] 147.126.59.89:37262 -> MY.NET.253.24:113
09/07-19:27:42.236314  [**] Queso fingerprint [**] 64.80.63.121:4114 ->      MY.NET.204.214:6355
09/07-22:17:10.280425  [**] Queso fingerprint [**] 64.80.63.121:2436 ->      MY.NET.201.86:6355
09/07-22:45:53.172847  [**] Queso fingerprint [**] 64.80.63.121:3475 ->      MY.NET.209.130:6355
09/08-17:49:55.816019  [**] Queso fingerprint [**] 64.80.63.121:4683 ->      MY.NET.217.182:113
09/08-19:24:14.509505  [**] Queso fingerprint [**] 64.80.63.121:1535 ->      MY.NET.209.130:6355
09/08-20:14:18.371606  [**] Queso fingerprint [**] 64.80.63.121:3240 ->      MY.NET.210.194:113
09/08-20:14:56.799825  [**] Queso fingerprint [**] 128.61.105.106:6699 ->  MY.NET.218.202:49452
09/08-20:14:57.789482  [**] Queso fingerprint [**] 128.61.105.106:0 ->        MY.NET.218.202:6699
09/09-03:18:36.048804  [**] Queso fingerprint [**] 216.15.191.130:33869 -> MY.NET.253.43:25
09/09-15:01:09.888516  [**] Queso fingerprint [**] 216.15.191.130:56815 -> MY.NET.6.35:25
09/10-10:29:43.642064  [**] Queso fingerprint [**] 64.80.63.121:3360 ->       MY.NET.223.42:113
09/10-15:16:55.765640  [**] Queso fingerprint [**] 64.80.63.121:4279 ->       MY.NET.217.26:6346
09/11-02:39:51.343881  [**] Queso fingerprint [**] 64.80.63.121:3644 ->       MY.NET.208.26:6355
09/11-09:13:54.380467  [**] Queso fingerprint [**] 64.80.63.121:1524 ->       MY.NET.224.34:6346
09/11-15:54:44.962789  [**] Queso fingerprint [**] 216.15.191.130:39926 -> MY.NET.6.34:25
09/13-03:16:38.464594  [**] Queso fingerprint [**] 216.15.191.130:40335 -> MY.NET.253.42:25
09/13-05:23:19.543649  [**] Queso fingerprint [**] 216.15.191.130:32770 -> MY.NET.253.41:25
09/14-01:05:01.718720  [**] Queso fingerprint [**] 129.2.146.48:6699 ->      MY.NET.201.146:1184
09/14-10:22:54.817324  [**] Queso fingerprint [**] 213.228.1.13:1581 ->      MY.NET.219.0:6346
```

```
64.80.63.121
CollegePark/LexingtonCrossing (NETBLK-PAET-MI-CPRK-LEX)
   Gainesville, FL 32608 US
   Netname: PAET-MI-CPRK-LEX
   Netblock: 64.80.63.0 - 64.80.63.255
   Coordinator: Darby, Brian  (BD114-ARIN)  bdarby@campuslink.com
                                  734-975-8075
```

3.2.1.12 Probable NMAP fingerprint attempt

Snort Rule

```
alert TCP !$HOME_NET any -> $HOME_NET any (msg:"IDS5 - SCAN-Possible NMAP Fingerprint attempt"; flags: SFPU; )
```

Description:

The detection rule has been created to detect the footprint of the Operating System fingerprinting tool named Nmap. The following is taken from "*Nmap network security scanner man page*" available at http://www.insecure.org/nmap/nmap_manpage.html
"*Nmap is designed to allow system administrators and curious individuals to scan large networks to determine which hosts are up and what services they are offering. Nmap supports a large number of scanning techniques such as: UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), Reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep, and Null scan."* The tool Nmap is available at http://www.insecure.org/nmap/
There is an article called "*Remote OS detection via TCP/IP Stack Fingerprinting*" by Fyodor, at, http://www.insecure.org/nmap/nmap-fingerprinting-article.html , the paper describes how Nmap was implemented.
A paper by Ofir Arkin, *Network Scanning Techniques*, in section *4.3 TCP/IP Stack Fingerprinting* gives another description of the tricks that Nmap uses. It is available at http://www.sys-security.com/html/papers.html
ID FAQ - *What is nmap and what can it do?* http://www.sans.org/newlook/resources/IDFAQ/What_is_nmap.htm
Information Security Reading Room - *Nmap - The Tool, It's Author and It's Implications* http://www.sans.org/infosecFAQ/nmap.htm

Known Issues:.

CAN-2000-0324 pcAnywhere 8.x and 9.x allows remote attackers to cause a denial of service via a TCP SYN scan, e.g. by nmap.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0324

Once an intruder can determine what operating system is running, then the exploits for that operating system can be put into play.

Cisco Advisory Notices: http://www.cisco.com/warp/public/707/advisory.html
Compaq (Digital UNIX) http://www.compaq.com/support
FreeBSD http://www.freebsd.org/security/
HP HP-UX For the US, Canada, Asia-Pacific, & Latin America: http://us-support.external.hp.com
For Europe:http://europe-support.external.hp.com
To Retrieve a Security Patch Matrix: ftp://us-ffs.external.hp.com/export/patches/hp-ux_patch_matrix/
IBM AIX http://techsupport.services.ibm.com/support/rs6000.support/downloads
http://techsupport.services.ibm.com/rs6k/fixes.html
Microsoft http://www.microsoft.com/technet/security/current.asp
Nortel Networks (Bay Networks): http://www12.nortelnetworks.com/ (you will need to know your equipment type, and may need an account)
SCO (OpenServer and Unixware) http://www.sco.com/security/ (Security Bulletins and Patches)
http://www.sco.com/support/ftplists/index.html (General OS patches)
Sun Solaris http://sunsolve.sun.com (Recommended & Security Patches)
SGI http://support.sgi.com
Linux
Caldera http://www.caldera.com/support/security/
Debian http://www.debian.org/security/index.en.html
Mandrake: http://www.linux-mandrake.com/en/fupdates.php
Red Hat: http://www.redhat.com/support/updates.html

SuSe http://www.suse.com/support/download/updates/index.html
http://www.suse.de/en/support/security/index.html

Detection:

```
08/15-10:23:16.502216  [**] Probable NMAP fingerprint attempt [**] 216.181.188.154:1951 -> MY.NET.6.44:110
08/16-01:42:40.602586  [**] Probable NMAP fingerprint attempt [**] 207.151.147.201:58191-> MY.NET.60.8:21
08/17-10:08:26.121492  [**] Probable NMAP fingerprint attempt [**] 130.149.41.70:1050   -> MY.NET.217.46:994
08/18-09:01:49.485959  [**] Probable NMAP fingerprint attempt [**] 24.23.198.174:1467   -> MY.NET.217.46:2928
09/02-16:25:42.155404  [**] Probable NMAP fingerprint attempt [**] 63.226.208.41:28518  -> MY.NET.253.41:22
09/08-18:56:39.751302  [**] Probable NMAP fingerprint attempt [**] 151.196.73.119:37197 -> MY.NET.253.112:22
09/11-04:48:56.731170  [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.5:23 + various nodes on MY.NET.208 .X all port 23
09/11-05:00:29.685702  [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.74:21
09/11-05:05:15.620812  [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.114:135
09/11-05:13:59.133815  [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.190:1025
09/11-05:17:25.440117  [**] Probable NMAP fingerprint attempt [**] 24.180.134.156:50111 -> MY.NET.208.226:2000
```

```
24.180.134.156
@Home Network (NETBLK-BLTMMD1-MD-1)
  Redwood City, CA 94063 US
  Netname: BLTMMD1-MD-1
  Netblock: 24.180.128.0 - 24.180.143.255
  Coordinator:
    Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
    (650) 556-5599
```

3.2.1.13 External RPC call

Snort Rule:

Custom -This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from an earlier version of Snort. It has been set up to capture traffic to port 111 on MY.NET.X.X It will be similar to the following, although the actual rule may have more features:

```
alert TCP !$HOME_NET any -> $HOME_NET 111 (msg:" External RPC call "; offset: 40; depth: 8; )
```

Description: This is item number 3 on the SANS list of Top Ten Security Issues http://www.sans.org/topten.htm
The individuals are attempting to access the Portmapper service on port 111. Issuing a request such as "rpcinfo –p" , once a connection is established will dump a list of currently registered Remote Procedure Call services available on that machine. This information can then be used to formulate an attack. This approach to information gathering is efficient, but noisy, and readily detected, and is generally not used by a sophisticated operator, unless they are aware of a specific, new, exploit, and are going for a kill.

Known Issues:

CVE-1999-0008 Buffer overflow in NIS+, in Sun's rpc.nisd program http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0008
CVE-1999-0212 Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0212
CVE-1999-0320 SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0320
CVE-1999-0974 Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad service.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0974
CAN-1999-0195 Denial of service in RPC portmapper allows attackers to register or unregister RPC services or spoof RPC services using a spoofed source IP address such as 127.0.0.1.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0195
CAN-1999-0568 rpc.admind in Solaris is not running in a secure mode. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0568
CAN-1999-0795The NIS+ rpc.nisd server allows remote attackers to execute certain RPC calls without authentication to obtain system information, disable logging, or modify caches. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0795
CERT® Advisory CA-2000-17 Input Validation http://www.cert.org/advisories/CA-2000-17.html
CERT® Incident Note IN-2000-10 http://www.cert.org/incident_notes/IN-2000-10.html
CERT® Summary CS-2000-03 http://www.cert.org/summaries/CS-2000-03.html
CERT® Incident Note IN-99-04 Similar Attacks Using Various RPC Services http://www.cert.org/incident_notes/IN-99-04.html
CA-99-08 - Buffer Overflow Vulnerability in rpc.cmsd http://www.cert.org/advisories/CA-99-08-cmsd.html
CA-99-05 - Vulnerability in statd exposes vulnerability in automountd http://www.cert.org/advisories/CA-99-05-statd-automountd.html
CA-98.11 - Vulnerability in ToolTalk RPC Service http://www.cert.org/advisories/CA-98.11.tooltalk.html
CAN-2000-0800 String parsing error in rpc.kstatd in the linuxnfs or knfsd packages in SuSE and possibly other Linux systems allows remote attackers to gain root privileges. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0800

ID FAQ - *The trouble with RPCs* http://www.sans.org/newlook/resources/IDFAQ/trouble_RPCs.htm

Detection:

```
08/18-06:10:13.484691  [**] External RPC call [**] 18.116.0.75:111 -> MY.NET.6.15:111
08/19-01:39:20.501009  [**] External RPC call [**] 141.223.124.31:2796 -> MY.NET.6.15:111
08/19-01:41:50.539748  [**] External RPC call [**] 141.223.124.31:3033 -> MY.NET.100.130:111
08/19-10:11:34.529565  [**] External RPC call [**] 209.160.238.215:2572 -> MY.NET.6.15:111
08/19-10:11:47.587415  [**] External RPC call [**] 209.160.238.215:4980 -> MY.NET.15.127:111
08/19-10:13:58.565702  [**] External RPC call [**] 209.160.238.215:2815 -> MY.NET.100.130:111
09/02-00:28:06.989407  [**] External RPC call [**] 210.101.101.110:861 -> MY.NET.6.15:111
09/03-11:47:57.195160  [**] External RPC call [**] 210.100.199.219:3478 -> MY.NET.100.130:111
09/03-11:42:36.076230  [**] External RPC call [**] 210.100.199.219:2378 -> MY.NET.6.15:111
09/10-03:15:33.932802  [**] External RPC call [**] 161.31.208.237:874 -> MY.NET.6.15:111
```

3.2.1.14 TCP SMTP Source Port traffic

Snort Rule:

Custom -This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from an earlier version of Snort. It has been set up to capture traffic from any machine, port 25, to any machine on MY.NET.X.X. It will be similar to the following, although the actual rule may have more features:

```
alert TCP !a 25 -> a any (msg:" TCP SMTP Source Port traffic ";)
```

Description: Sendmail is the program most often used on UNIX mailservers. It uses SMTP to communicate. Sendmail vulnerabilities are listed as item 5 on the SANS list of Top Ten Security Issues http://www.sans.org/topten.htm

Known Issues:

CAN-2000-0738 WebShield SMTP 4.5 allows remote attackers to cause a denial of service by sending e-mail with a From: address that has a . (period) at the end,
which causes WebShield to continuously send itself copies of the e-mail.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0738

CAN-2000-0657 Buffer overflow in AnalogX proxy server 4.04 and earlier allows remote attackers to cause a denial of service via a long HELO command in the SMTP protocol.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0657

CVE-2000-0582 Check Point FireWall-1 4.0 and 4.1 allows remote attackers to cause a denial of service by sending a stream of invalid commands (such as binary zeros) to the SMTP Security Server proxy.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0582

CVE-1999-0203 In Sendmail, attackers can gain root privileges via SMTP by specifying an improper “mail from” address and an invalid “rcpt to” address that would cause the mail to bounce to a program.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0203

CVE-1999-0047 MIME conversion buffer overflow in Sendmail versions 8.8.3 and 8.8.4. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0047

CVE-1999-0130 Local users can start Sendmail in daemon mode and gain root privileges. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0130

CVE-1999-0131 Buffer overflow and denial of service in Sendmail 8.7.5 and earlier through GECOS field gives root access to local users.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0131

CVE-1999-0203 In Sendmail, attackers can gain root privileges via SMTP by specifying an improper "mail from" address and an invalid "rcpt to" address that would cause the mail to bounce to a program. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0203

CVE-1999-0204 Sendmail 8.6.9 allows remote attackers to execute root commands, using ident. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0204

CVE-1999-0206 MIME buffer overflow in Sendmail 8.8.0 and 8.8.1 gives root access. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0206

Advisory CA-1997-05 MIME Conversion Buffer Overflow in Sendmail Versions 8.8.3 and 8.8.4 http://www.cert.org/advisories/CA-1997-05.html

SANS CVE Entries http://www.sans.org/y2k/CVE.htm

Detection:

08/17-00:06:16.011962 [**] TCP SMTP Source Port traffic [**] 206.46.170.21:25 -> MY.NET.97.181:25,
5 occurrences during a 30 minute period, probably normal SMTP mail traffic between SMTP Mail Servers.

09/10-15:36:32.348040 [**] TCP SMTP Source Port traffic [**] 156.40.66.2:25 -> MY.NET.253.53:757
3 occurrences during an hour, the last two close together. This is symptomatic of a user directly picking up mail from 156.40.66.2, an external mail server. This could be a possible policy violation.

3.2.1.15 Possible wu-ftpd exploit - GIAC000623

Snort Rule:

Custom This rule has been custom designed by the installer/maintainer of the Snort Intrusion Detection system at this site, or is from an earlier version of Snort. It has been set up to capture traffic from any machine, to any machine on MY.NET.X.X. port 21 It will be similar to the following, although the actual rule may have more features:

```
alert TCP !a any -> a 21 (msg:" Possible wu-ftpd exploit - GIAC000623 ";[options here will be specific for WU-ftpd as
determined by the rule designer])
```

Description:

24.17.189.83 has scanned MY.NET.99.104, MY.NET.150.24, MY.NET.202.190, and MY.NET.202.202 looking for wu-ftpd exploits. The specific MY.NET machines should checked to see if they are at the latest version of the wu-FTP server and if not then upgraded, or ftp disabled if possible. This is a targeted attack, therefore 24.17.189.83 has knowledge that those machines may be FTP servers.

Known Issues:

CVE-1999-0080 wu-ftp FTP server allows root access via
"site exec" command. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0080
CVE-1999-0081 wu-ftp allows files to be overwritten via the rnfr command. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0081
CVE-1999-0368 Buffer overflows in wuarchive ftpd (wu-ftpd) and ProFTPD lead to remote root access, a.k.a.palmetto.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0368
CVE-1999-0878 Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges
via MAPPING_CHDIR. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0878
CVE-1999-0879 Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges
via macro variables in a message file. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0879
CVE-1999-0880 Denial of service in WU-FTPD via the SITE NEWER command, which does not free memory properly
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0880
CVE-1999-0997 wu-ftp with FTP conversion enabled allows an attacker to execute commands via a malformed file
name that is interpreted as an argument to the program that does the conversion, e.g. tar or
uncompress. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0997
CAN-1999-0076 Buffer overflow in wu-ftp from PASV command causes a core
dump. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0076
CAN-1999-0156 wu-ftpd FTP daemon allows any user and password combination.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0156
CAN-2000-0573 The lreply function in wu-ftpd 2.6.0 and earlier does not properly
cleanse an untrusted format string, which allows remote attackers to execute arbitrary commands
via the SITE EXEC command. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0573

Detection:

```
09/08-04:53:17.038845  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:3446 -> MY.NET.99.104:21
09/08-05:25:41.092146  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:4640 -> MY.NET.150.24:21
09/08-05:25:41.167678  [**]             Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:4640 -> MY.NET.150.24:21
09/08-05:59:01.961301  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:2362 -> MY.NET.202.202:21
09/08-05:59:02.084974  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:2363 -> MY.NET.202.190:21
09/08-05:59:04.101862  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:2362 -> MY.NET.202.202:21
09/08-05:59:04.191384  [**]             Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:2362 -> MY.NET.202.202:21
```

```
09/08-05:59:04.271433  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.17.189.83:2363 -> MY.NET.202.190:21
@Home Network (NETBLK-BB1-RDC1-TX-10)
   Redwood City, CA 94063 US
   Netname: BB1-RDC1-TX-10
   Netblock: 24.17.176.0 - 24.17.191.255
   Coordinator: Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
                        (650) 556-5599
```

3.2.1.16 Happy 99 Virus

Snort Rule

```
alert TCP any any -> $HOME_NET 25(msg:"Happy99 Virus"; content: "X-Spanska\:Yes"; )
```

Description: A mail message with the features of the Happy 99 Virus has been sent to MY.NET .6.35, and MY.NET.179.80 Those computers are likely infected, and may have infected other machines unless an anti Virus campaign is in place. Recommendation: verify presence of infection and remove institute anti-virus campaign.

Known Issue: "Happy 99" virus information can be found at http://www.symantec.com/avcenter/venc/data/happy99.worm.html

Detection:

```
08/16-14:36:46.954418  [**] Happy 99 Virus [**] 128.8.198.101:12805 -> MY.NET.6.35:25
128.8.198.101Canonical name: wmuc.umd.edu
University of Maryland (NET-UMDNET)
   College Park, MD 20742  US
   Netname: UMDNET
   Netnumber: 128.8.0.0
   Coordinator: University of Maryland DNS Administration  (UM-ORG-ARIN)  dnsadmin@NOC.UMD.EDU
                                    (301) 405-3003

   08/20-15:41:12.157972  [**] Happy 99 Virus [**] 24.2.2.66:58102 -> MY.NET.179.80:25
   No match for "24.2.2.66:".
```

3.2 Analysis of Type 2 Data

3.2.1 First generic type:

This generic type consists of various scans, generated by various tools, all have the following in common. There is one source node, it is sending packets at MY.NET.A.X, where A is a subnet number. And X is a node on that subnet. This type of scanning is generally referred to as a horizontal scan.

3.2.1.1 Destination Port 21

FTP has been designed to use two operating ports, 20 (FTP-Data) and 21 (FTP-Control) . Port 21 is used to setup and control the FTP session. It has a number of vulnerabilities. Further information can be found at:

CA-99-13 Multiple Vulnerabilities in WU-FTPD http://www.cert.org/advisories/CA-99-13-wuftpd.html
CA-2000-13 Two Input Validation Problems In FTPD http://www.cert.org/advisories/CA-2000-13-wuftpd.html
CA-97.27 FTP Bounce and the following Trojans http://www.cert.org/advisories/CA-97.27.FTP_bounce.html
Back Construction, Blade Runner, Dolly Trojan, Fore, FTP Trojan, Invisible FTP, Juggernaut 42,MotIv FTP, Larva, Net Administrator, Traitor 21, WebEx, WinCrash
CERT: CA-99-03-FTP-Buffer-Overflows http://www.cert.org/advisories/CA-99-03-FTP-Buffer-Overflows.html
SITE STATS-Getting "STATS" from a web server gives good information on how to exploit other weaknesses, such as the PASV exploit. http://www.infowar.com/iwftp/iw_sec/iw_sec_01_followup.shtml
Advisory: IIS FTP Exploit/DoS Attack-Buffer overflow in the NLIST command can crash or break into the FTP server. http://www.securityfocus.com/templates/archive.pike?list=1&msg=007b01be47b2$0fd4d5f0$abd40018@CORE
FTP PASV "Pizza Thief" Exploit-Predicting possible connections in order to redirect output.
Vulnerability in Broker FTP Server v. 3.0 Build 1-LIST http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind9906&L=ntbugtraq&F=P&S=&P=629
BugtraqID: 599 http://www.securityfocus.com/bid/599.html
BugtraqID: 572-This client has a vulnerability in the code that processes the response to a CWD command. http://www.securityfocus.com/bid/572.html
BugtraqID: 269-CWD or LS commands with strings longer than 155 overflow the buffer http://www.securityfocus.com/bid/269.html
BugtraqID: 301 http://www.securityfocus.com/bid/301.html
BugtraqID: 442-A buffer overflow exists in the authentication code, so that long hostnames or usernames can be used to break into the system. http://www.securityfocus.com/bid/442.html
BugtraqID: 650 http://www.securityfocus.com/bid/650.html
BugtraqID: 658 http://www.securityfocus.com/bid/658.html
BugtraqID: 401 http://www.securityfocus.com/bid/401.html
Malformed RPC Packet DoS Vulnerability in Windows 2000, http://www.microsoft.com/technet/security/bulletin/fq00-066.asp
Faststream FTP++ 2.0,W2k,DOS http://www.delphisplc.com/thinking/whitepapers/
Exploit code released for Firewall-1 FTP PASV security vulnerability http://www.securiteam.com/exploits/Exploit_code_released_for_Firewall-1_FTP_PASV_security_vulnerability.html
EFTP vulnerable to two DoS attacks http://archives.neohapsis.com/archives/bugtraq/2000-09/0089.html

210.61.144.125

```
Sep 11 06:45:13 210.61.144.125:21 -> MY.NET.1.3:21 SYNFIN **SF****
```

mapping 1,2,4,5,7,9,10,11,12,13,NO 14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100,

104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151,
152,153,154,155,156,157,158,159,160,161,162,163,178,179,180,181,182,183,184,185,186,188,190,
198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,253,254
Sep 11 07:06:48 210.61.144.125:21 -> MY.NET.254.250:21 SYNFIN **SF
inetnum: **210.61.144.0 - 210.61.144.255**
netname: HINET8-144-TW
descr: Abnet Information Co., Ltd
descr Taipei, Taiwan TW
person: Wen-Lon Li
address: Abnet Information Co., Ltd
phone: +886-2-558-2115
fax-no: +886-2-558-2116
e-mail: abnet@ms15.hinet.net

195.114.226.41 Scanned the various subnets below , using various source ports all going to destination port 21
Aug 15 00:46:11 **195.114.226.41**:2244 -> MY.NET.1.2**:21** SYN **S*****
1,2,4,5,6,7,10,11,12,13,MY.NET.14.2:21,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100,104,105,106,107,108,109,110,111,112,115,120,
130,
139,140,141,142,143,144,145,146,150,151,152,153,155,156,157,158,159,160,161,162,163,178,179,180,181,182,183,184,185,186,188,190,198,199,200,201,202203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,232,253,254,
Aug 15 02:35:53 195.114.226.41:4249 -> MY.NET.254.252:21 SYN **S*****
@15-30 hosts per sec, out of sequence, low speed network connection??
inetnum: 195.114.226.0 - 195.114.226.255
netname: MW02
descr: MultiWeb
person: Duncan Schoen
address: Multiweb BV
address: Netherlands
phone: +31 72 5140626
fax-no: +31 72 5140621
e-mail: beheer@multiweb.nl

213.188.8.45 Scanned various subnets, using various source ports all going to destination port 21
Aug 16 04:13:27 213.188.8.45:2415 -> MY.NET.208.42:21 SYN **S*****
Aug 16 04:13:28 213.188.8.45:2404 -> MY.NET.201.246:21 SYN **S*****
to
Aug 16 04:14:17 213.188.8.45:2420 -> MY.NET.221.78:21 SYN **S*****
inetnum: 213.188.8.0 - 213.188.9.255
netname: NO-ELTELE-OST-FAST
descr: Fast Search & Transfere ASA Norway

role: ElTele Rogaland Contact Role
address: ElTele Rogaland As
address: Professor Olav Hansensvei 13
address: N-4003 Stavanger Norway
phone: +47 51 87 44 00
fax-no: +47 51 87 44 01
e-mail: ragnhild.aass@etr.no

24.94.176.113 Scanned only MY.NET.100.X, using various source ports all going to destination port 21
Aug 18 21:00:03 24.94.176.113:1476 -> MY.NET.100.0:**21** SYN **S*****
Aug 18 21:00:03 24.94.176.113:1482 -> MY.NET.100.6:21 SYN **S*****
Aug 18 21:00:03 24.94.176.113:1483 -> MY.NET.100.7:21 SYN **S*****
Aug 18 21:00:03 24.94.176.113:1484 -> MY.NET.100.8:21 SYN **S*****
Aug 18 21:00:03 24.94.176.113:1485 -> MY.NET.100.9:21 SYN **S*****
RoadRunner-KansasCity-Leavenworth-DHUB (NETBLK-KC-RR-176)
Kansas City, MO 64133 US
Netblock: 24.94.176.0 - 24.94.176.255
Coordinator: Channell, Bruce (BC87-ARIN) abuse@rr.com
877-777-2263 (FAX) 703-345-3504

4.54.37.160 Scanned subnets indicated below, using various source ports all going to destination port 21
Aug 18 02:41:41 4.54.37.160:3618 -> MY.NET.6.12:21 SYN **S*****
fast, 3 subnets in 9 seconds, went .6,.60,.1
Aug 18 02:50:27 4.54.37.160:4239 -> MY.NET.1.204:21 SYN **S******
BBN Planet (NET-SATNET)
Cambridge, MA 02138
US
Netblock: 4.0.0.0 - 4.255.255.255
Coordinator: Soulia, Cindy (CS15-ARIN) csoulia@genuity.net
800-632-7638

210.100.192.254 Scanned various subnets, using various source ports all going to destination port 21
Sep 5 18:33:24 210.100.192.254:37885 -> MY.NET.18.154**:21** SYN **S*****
inetnum: 210.100.128.0 - 210.100.255.255
netname: PUBNET
descr: Korea Telecom
descr: Seoul KR
remarks: ISP in Korea
person: Gisu Choi
address: Korea Telecom
phone: +82 2 766 1407

```
fax-no:     +82 2 766 6008
e-mail:     mgr@ns.pubnet.nm.kr
```

212.143.237.22 Scanned various subnets, using various source ports all going to destination port 21

```
Sep  5 18:05:53 212.143.237.22:3545 -> MY.NET.10.50:21 SYN **S*****
inetnum:     212.143.0.0 - 212.143.0.255
netname:     NV-BB
descr:       NetVision Ltd. IL
route:       212.143.0.0/16
descr:       Netvision Ltd.
descr:       Internet Service Provider
descr:       Haifa 31605 Israel
phone:       +972 48 560 600
fax-no:      +972 48 551 132
e-mail:      noc-team@netvision.net.il
trouble:     Send abuse and spam reports to abuse@netvision.net.il
```

195.130.128.202 Scanned various subnets, using various source ports all going to destination port 21

```
Sep  8 16:19:19 195.130.128.202:21035 -> MY.NET.5.37:21 SYN **S*****
inetnum:     195.130.128.0 - 195.130.149.255
netname:     TELENET
descr:       Telenet Operaties N.V.  BE
route:       195.130.128.0/19
changed:     Piet.Spiessens@telenet.be
address:     B-2800 Mechelen Belgium
e-mail:      tech@telenet-ops.be
trouble:     IMPORTANT: To report intrusion attempts, hacking,
trouble:     IMPORTANT: spamming, or other unaccepted behavior
trouble:     IMPORTANT: by a Telenet/Pandora customer, please
trouble:     IMPORTANT: send a message to abuse@pandorC.De
```

24.17.189.83

```
Sep  8 03:48:41 24.17.189.83:2041 -> MY.NET.1.60:21 SYN **S*****
```

Scanned subnets indicated below, using various source ports all going to destination port 21
1,2,4,5,7,9,10,11,12,13,NO 14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100,
104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151,152,153,154,155,156,157,158,159,160,161,162,163,178,179180,181,182,183,184,185,186,188,190,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225226,227,228,229,230,231,232,253,254 @ 10 nodes per sec, some out of sequence

```
Sep  8 06:27:48 24.17.189.83:3915 -> MY.NET.254.254:21 SYN **S*****
@Home Network (NETBLK-BB1-RDC1-TX-10)
   Redwood City, CA 94063 US
   Netname: BB1-RDC1-TX-10
   Netblock: 24.17.176.0 - 24.17.191.255
   Coordinator:  Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
      (650) 556-5599
```

213.188.8.45 Scanned various subnets, using various source ports all going to destination port 21

```
Sep  9 05:24:37 213.188.8.45:4253 -> MY.NET.201.222:21 SYN **S*****
inetnum:     213.188.8.0 - 213.188.9.255
netname:     NO-ELTELE-OST-FAST
descr:       Fast Search & Transfere ASA
role:        ElTele Rogaland Contact Role
address:     Professor Olav Hansensvei 13
address:     N-4003 Stavanger Norway
phone:       +47 51 87 44 00
fax-no:      +47 51 87 44 01
e-mail:      ragnhild.aass@etr.no
```

212.170.19.199 Scanned subnets indicated below, using various source ports all going to destination port 21

```
Sep 11 11:42:05 212.170.19.199:1784 -> MY.NET.5.1:21 SYN **S*****
5,6,7,9,10,11,12,13,14...nets @ 5-10/sec
Sep 11 11:49:41 212.170.19.199:4112 -> MY.NET.13.249:21 SYN **S*****
inetnum:     212.170.0.0 - 212.170.15.255
netname:     TTDNET
descr:       Telefonica Data Espana (NCC#1999085999 )
address:     Telefonica Data Espana
address:     28040 Madrid Spain
phone:       +34 902 230 210
fax-no:      +34 91 4567825
e-mail:      secure@telefonica-data.com
trouble:     For security related problems contact:
trouble:     -    security@ttd.net
trouble:     For problems relating electronic mail abuse contact:
trouble:     -    spam@ttd.net
trouble:     For problems relating dns servers  contact:
trouble:     -    redip_servicios@tsai.es
trouble:     - Port scanning related problems:
trouble:     -    scan@ttd.net
remarks:     Information http://www.telefonica-data.com
notify:      david.ortega@telefonica-data.com
```

64.1.198.164 Scanned various subnets, using various source ports all going to destination port 21

```
Sep 10 21:30:37 64.1.198.164:4684 -> MY.NET.222.7:21 SYN **S*****
Concentric Network Corporation (NETBLK-CONCENTRIC-BLK5)
```

San Jose, CA 95126-3429
Netname: CONCENTRIC-BLK5
Netblock: 64.0.0.0 - 64.3.255.255
Coordinator: DNS and IP ADMIN (DIA-ORG-ARIN) hostmaster@CONCENTRIC.NET
(408) 817-2800 Fax- - - (408) 817-2630

206.18.105.224 Scanned subnets indicated below, using various source ports all going to destination port 21

```
Sep 13 20:02:15 206.18.105.224:2610 -> MY.NET.5.141:21 SYN **S*****
```

scanning nets5,6,7

```
Sep 13 20:07:00 206.18.105.224:3216 -> MY.NET.7.213:21 SYN **S*****
```

The Internet Group (NETBLK-CERF-TIG-A)
Nuevo, CA 92567 USA
Netname: CERF-TIG-A
Netblock: 206.18.96.0 - 206.18.111.255
Coordinator: DNS Administrator (CERF-HM-ARIN) dns@CERF.NET
(619) 812-5000 Fax- - - - (408) 522-9911

3.2.1.2 Destination Port 23

This port is generally used for Telnet. The Telnet service provides a Virtual Terminal service on the remote machine. In can ultimately provide ROOT access to a machine. If the Telnet port is open for access, then the intruder can Telnet in to the machine and start trying Username and Password combinations. Success will be quick and easy if all of the well known standard passwords have not been changed from the defaults. If the default passwords have been changed then this becomes a slightly more challenging way to obtain access. Further information can be found at:

BugtraqID: 459 http://www.securityfocus.com/bid/459.html
BugtraqID: 594-Possible to set the TERM environmental variable before connecting. http://www.securityfocus.com/bid/594.html
Nt4.0 Telnet to port 53 vulnerability http://support.microsoft.com/support/kb/articles/Q169/4/61.ASP
Win2k Telnet.exe malicious server vulnerability http://www.insecure.org/sploits/NT.NTLM.auto-authentication.html
Windows 2000 Telnet Client NTLM Authentication" Vulnerability http://www.insecure.org/sploits/NT.NTLM.auto-authentication.html
NetStructure 7180 backdoor vulnerability http://www.securiteam.com/exploits/NetStructure_7180_backdoor_vulnerability.html
3Com's HiPer ARC vulnerable to a Denial of Service attack. http://www.securiteam.com/exploits/3Com_s_HiPer_ARC_vulnerable_to_a_Denial_of_Service_attack.html

129.186.93.133 Scanned subnets indicated below, using various source ports all going to destination port 23

```
Sep  6 21:24:04 129.186.93.133:1544 -> MY.NET.1.12:23 SYN **S*****
1,2,4,5,6,7,9,10,11,12,13,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100,104,105,106,107,108,109,110,111,112,115,120,130,138,139,14
0,141,142,143,144,145,146,150,151,152,153,154,155,156,157,158,159,160,161,162,163,178,179,180,181,182,183,184,185,186,188,190,198,199,200
Sep  6 22:29:25 129.186.93.133:4725 -> MY.NET.200.72:23 SYN **S*****
```

Iowa State University (NET-CYCLONENET)
Ames, IA 50011 US
Netname: CYCLONENET
Netblock: 129.186.0.0 - 129.186.255.255
Coordinator: Contact, Technical (TC42-ARIN) tech-contact@IASTATE.EDU
515-294-2256

128.171.57.194 Scanned subnets indicated below, using various source ports all going to destination port 23

```
Sep  6 12:52:36 128.171.57.194:1061 -> MY.NET.1.7:23 SYN **S*****
2,4,5,6,7,9,10,11,12,13,15,17,18,20,21,25,26
```

University of Hawaii (NET-HAWAII)
University of Hawaii
Honolulu, HI 96822 US
Netname: HAWAII
Netnumber: 128.171.0.0
Coordinator: University of Hawaii Keller Hall202 (ZU32-ARIN) netcontact@HAWAII.EDU
808 521-2879

205.238.205.3 Scanned various subnets, using various source ports all going to destination port 23

```
Sep 11 13:18:20 205.238.205.3:9358 -> MY.NET.201.36:23 SYN **S*****
```

KINGS COLLEGE (NET-KINGS-EPIX)
WILKES-BARRE, PA 18711 US

```
Netname: KINGS-EPIX
Netnumber: 205.238.205.0
Coordinator:   Blanck, William R.  (WB385-ARIN)  bblanck@EPIX.NET
                (717)-674-4135
```

3.2.1.3 Destination Port 53

This port is generally used for Domain Name System (DNS) transfers. The DNS programs use UDP transfers in normal operation. TCP is only used in situations where there is a large amount of information to be transferred, and most implementations uses a source port of 53, both UDP and TCP. In the situations below, the systems performing the scan are using destination port 53, in order to evade filtering routers or firewalls in order to map a network.

For vulnerabilities further information can be found at:
IN-2000-04 Denial of Service Attacks using Nameservers http://www.cert.org/incident_notes/IN-2000-04.html
CA-2000-03 Continuing Compromises of Nameservers http://www.cert.org/advisories/CA-2000-03.html
CA-99-14 Multiple Vulnerabilities in BIND http://www.cert.org/advisories/CA-99-14-bind.html
CA-98.05 Multiple Vulnerabilities in BIND http://www.cert.org/advisories/CA-98.05.bind_problems.html
DoS in Windows NT DNS servers by flooding port 53 with too many characters. http://support.microsoft.com/support/kb/articles/Q162/9/27.asp

193.120.216.2 Scanned various subnets, using various source ports all going to destination port 53

```
Sep 11 07:23:42 193.120.216.2:2666 -> MY.NET.60.39:53 SYN **S*****
route:      193.120.0.0/16
descr:      IEUNET-AGG-ROUTE-1
descr:      Principal address block of EUnet Ireland
remarks:    Aggregated route covering multiple EUnet Ireland networks Esat Net is the trading name for EUnet Ireland Ltd.
notify:     noc@esat.net
role:       Esat.Net NOC
address:    Dublin 2, Ireland
phone:      +353 1 6790832
fax-no:     +353 1 6708118
e-mail:     noc@esat.net
```

206.186.79.9 Scanned subnets indicated below, using various source ports all going to destination port 53 @ approx. 15 nodes per sec

```
Sep  9 22:35:21 206.186.79.9:2351 -> MY.NET.1.4:53 SYN **S*****
Sep  9 22:35:21 206.186.79.9:2352 -> MY.NET.1.5:53 SYN **S*****
1,2,4,5,7,9,10,11,12,13,14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100,
104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151,152,153,154,155,156,157,158,159,160,161,162,163,17
8,179181,182,183,184,185,186,188,190,198,199,200,201,201,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217.218,219,220,221,222,223,2
24,225,226227,228,229,230,231,232,253,254,1,2,4,5,6,7,9,10,11,12,13,NO14,15,17,18,20,21,25,26
Sep 10 02:13:08 206.186.79.9:2450 -> MY.NET.254.204:53 SYN **S*****
Sprint Canada Inc. (NETBLK-INSINC-BLK2)
   Vancouver, BC  V6A 4E6 CA
   Netname: INSINC-BLK2
   Netblock: 206.186.0.0 - 206.186.255.0
   Coordinator: Network Operations Contact  (NOC71-ORG-ARIN)  noc@SPRINT-CANADA.NET
                       800-665-3633  Fax- 800-555-5641
```

3.2.1.4 Destination Port 27374

Scanning for this port is a **scan for Bad Blood, SubSeven, SubSeven 2.1 Gold, SubSeven 2.1.4 Defcon8, Trojans**.
Further information can be found at http://www.commodon.com/threat/threat-sub7.htm
and http://xforce.iss.net/static/2245.php

35.10.82.111

```
Aug 16 04:35:21 35.10.82.111:2814 -> MY.NET.1.6:27374 SYN **S*****
```

Scanned subnets, using various source ports all going to destination port 27374
1,2,4,5,6,7,9,10,11,12,13,MY.NET.14.2, 15,17,18,20 ,21,25, 26, high speed all in sequence, more than 50 per sec, starts skipping nodes, possibly has a map,

```
Aug 16 04:37:08 35.10.82.111:1443 -> MY.NET.26.237:27374 SYN **S*****
Aug 16 04:58:52 35.10.82.111:2246 -> MY.NET.1.2:27374 SYN **S*****
```

pattern as above, but continues with 26,53,54,60,68,69,70,71,messed with 70.221,70.223 then continues with 71.31,75,85,94,97,98,99,100,104,105,106,107,108,109,110,111,112,115,120,130,139,140,141,142,143,144,145,146,150,151,152,153,155,156,157,158,159,160,161,162,163,178,179,180,181,182,183,184,185,186,188,190,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,253,254,

```
Aug 16 05:16:28 35.10.82.111:3144 -> MY.NET.254.254:27374 SYN **S*****
```

Michigan State University (NETBLK-MICH-618)
Computer Laboratory
East Lansing, MI 48824 AUS
Netname: MICH-618
Netblock: **35.8.0.0 - 35.10.255.255**
Coordinator: Nelson, Doug (DEN4-ARIN) nelson@msu.edu
517-353-2980

207.19.142.78 Scanned subnets indicated below, using various source ports all going to destination port 27374

```
Sep  5 16:20:18 207.19.142.78:1093 -> MY.NET.223.159:27374 SYN **S*****
```

scanned 223,224,225,226,227,228,230,220,221,223,224,225,227,228,229,230,231,232,200,201,202,203,204,206,208,209,210

```
Sep  5 16:33:13 207.19.142.78:3924 -> MY.NET.210.205:27374 SYN **S*****
```

Baltimore County Public Library (NETBLK-UU-207-19-140-143)
Towson, MD 21204 USA
Netname: UU-207-19-140-143
Netblock: **207.19.140.0 - 207.19.143.255**
Coordinator: Old, Chip (CO53-ARIN) fold@BCPL.NET
410-887-6180 (FAX) 410-887-2091

3.2.1.5 Destination Port 12346

This is a **scan for GabianBus, NetBus 1.0 X-bill Trojans**
Further information can be found at BugtraqID: 1013 http://www.securityfocus.com/bid/1013.html

```
Sep  9 06:56:04 210.55.227.138:3519 -> MY.NET.200.4:12346 SYN **S*****
200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,
218,219,220,221,222,223,224,225,226,227,228,229,230,231,232, some out of sequence, 5 nodes per sec
inetnum:    210.55.227.0 - 210.55.227.255
netname:    WORLD-NET
descr:      Word-Net Ltd.
descr:      6-8 Nugent St. Auckland NZ
rev-srv:    home.world-net.co.nz
rev-srv:    shell.world-net.co.nz
person:     Thomas Lee
address:    P.O. Box 8591 Symonds Street Auckland
phone:      +64-9-3099004 fax-no:     +64-9-3099811
e-mail:     thomas@world-net.co.nz
```

3.2.1.6 Destination Port 1243

This is a **scan for BackDoor-g, Sub 7.2, Tiles virii**
Further information can be found at http://www.commodon.com/threat/threat-sub7.htm and http://xforce.iss.net/static/2245.php

```
Sep  8 23:27:41 62.136.41.111:2715 -> MY.NET.1.151:1243 SYN **S*****
Sep  8 23:27:41 62.136.41.111:2720 -> MY.NET.1.156:1243 SYN **S*****
Sep  8 23:27:41 62.136.41.111:2723 -> MY.NET.1.159:1243 SYN **S*****
Sep  8 23:27:41 62.136.41.111:2727 -> MY.NET.1.163:1243 SYN **S*****
inetnum:    62.136.0.0 - 62.136.255.255
netname:    POL-CAG1
descr:      CAG Block 1
descr:      Planet Online Limited
descr:      In case of problems please call +44 113 234 6068
descr:      Please do not send abuse reports to tech or admin contacts
descr:      Planet Online Limited
descr:      The White House
descr:      Melbourne St.
person:     Darren Marshall
address:    Planet Online LTD
address:    The Whitehouse
address:    Melbourne St Leeds LS2 7PS
address:    Great Britain
phone:      +44 1132345566
```

3.2.1.7 Destination Ports 27374 and 12346

The scanning activity for port **27374 is a scan for Bad Blood, SubSeven, SubSeven 2.1 Gold, SubSeven 2.1.4 Defcon8, Trojans. 12346 is a scan for GabianBus, NetBus 1.0, X-bill Trojans**
Further information on port 27374 issues can be found at http://www.commodon.com/threat/threat-sub7.htm and http://xforce.iss.net/static/2245.php
Further information on port 12346 issues can be found at BugtraqID: 1013 http://www.securityfocus.com/bid/1013.html

```
Sep  9 07:00:32 210.55.227.138:4507 -> MY.NET.232.247:27374 SYN **S*****
Sep  9 07:00:33 210.55.227.138:4518 -> MY.NET.232.252:12346 SYN **S*****
Sep  9 07:00:33 210.55.227.138:4519 -> MY.NET.232.253:27374 SYN **S*****
inetnum:     210.55.227.0 - 210.55.227.255
netname:     WORLD-NET
descr:       Word-Net Ltd.
descr:       6-8 Nugent St. Auckland NZ
rev-srv:     home.world-net.co.nz
rev-srv:     shell.world-net.co.nz
notify:      nic@netgate.net.nz
person:      Thomas Lee
address:     P.O. Box 8591 Symonds Street Auckland NZ
phone:       +64-9-3099004
fax-no:      +64-9-3099811
country: e-mail:      thomas@world-net.co.nz

Aug 15 17:20:34 195.57.243.171:64525 -> MY.NET.60.8:22 SYN **S*****
Aug 15 17:20:34 195.57.243.171:64526 -> MY.NET.60.8:383 SYN **S*****
inetnum:     195.57.243.0 - 195.57.243.255
netname:     BITELNET
descr:       Internet Service Provider
person:      Jose A. Mejias
address:     BITel S.A.
address:     Palma de Mallorca 07014 (Baleares) SPAIN
phone:       +34 971 225700
fax-no:      +34 971 225701
```

3.2.2 Second Generic Type

3.2.2.1 Destination Port 7000

The following are scans for X-Windows Font Server (if safe) and Remote Grab, Kazimas, Exploit Translation Server, SubSeven 2.1 Gold Trojans (if unsafe)
threat-sub7 http://www.commodon.com/threat/threat-sub7.htm
Xforce SubSeven Info http://xforce.iss.net/static/2245.php
The Simovits Consulting Trojan listing identifies this as **potentially, SubSeven 2.1 Gold, Remote Access / ICQ Trojan**
http://www.simovits.com/nyheter9902.html

```
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.6.42:7000 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.70.142:7000 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.6.48:7000 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.60.43:7000 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.6.45:7000 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.6.33:7003 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.60.12:7003 UDP
Aug 15 05:27:19 24.3.39.44:7001 -> MY.NET.1.13:7003 UDP
Aug 15 06:27:19 24.3.39.44:7001 -> MY.NET.70.142:7000 UDP
Aug 15 06:27:20 24.3.39.44:7001 -> MY.NET.6.48:7000 UDP
Aug 15 06:27:20 24.3.39.44:7001 -> MY.NET.60.43:7000 UDP
Aug 15 06:27:19 24.3.39.44:7001 -> MY.NET.6.45:7000 UDP
Aug 15 06:27:20 24.3.39.44:7001 -> MY.NET.6.33:7003 UDP
Aug 15 06:27:20 24.3.39.44:7001 -> MY.NET.60.12:7003 UDP
Aug 15 06:27:20 24.3.39.44:7001 -> MY.NET.1.13:7003 UDP
```

continues in bursts until the end of supplied information

@Home Network (NETBLK-MD-COMCAST-OWML-1)
425 Broadway
Redwood City, CA 94063 US
Netname: MD-COMCAST-OWML-1
Netblock**: 24.3.32.0 - 24.3.39.255**
Coordinator:Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

3.2.2.2 Source and Destination Port 9704

This address has already been mentioned in the Type 1 (SYN-FIN scan)

Exposure

CVE-1999-0048 Talkd, when given corrupt DNS information, can be used to execute arbitrary commands with root privileges.
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0048

```
Sep  7 21:33:23 213.25.136.60:9704 -> MY.NET.1.4:9704 SYNFIN **SF****
1,2,4,5,6,7,9,10,11,12,13,MY.NET.14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85
Sep  7 21:40:36 213.25.136.60:9704 -> MY.NET.85.254:9704 SYNFIN **SF****
09/07-21:33:30.514289 213.25.136.60:9704 -> MY.NET.1.4:9704
inetnum:    213.25.136.0 - 213.25.136.15
netname:    E-SOLUTIONS
descr:      e-SOLUTIONS.com Poland Sp. z o.o. PL
person:   Wieslaw Kosidlak e-mail:      wkosidlak@mfrelas.com
phone:      +48 81 7453340
fax-no:     +48 81 7453315
```

3.2.2.3 Source Port 7777

This traffic is more than likely Napster, block incoming TCP port 7777 to any port greater than 1024 It also uses port 6699, 8875,8888, 9009.
The Napster site (http://www.napigator.com/) the audio and video exchange network.

```
Sep  6 20:24:49 209.123.198.156:7777 -> MY.NET.213.10:1071 UDP   to
Sep  6 20:29:58 209.123.198.156:7777 -> MY.NET.213.10:1068 UDP

Sep  7 22:40:35 209.123.198.156:7777 -> MY.NET.204.126:2317 UDP  to
Sep  7 22:53:40 209.123.198.156:7777 -> MY.NET.204.126:2432 UDP
Net Access Corporation (NETBLK-NAC-NETBLK02)
   Newton, NJ 07860
   Netname: NAC-NETBLK02
   Netblock: 209.123.0.0 - 209.123.255.255

   Coordinator: Pavely, Ryan  (RP2938-ARIN)  paradox@NAC.NET
                       201-983-0725 (FAX) 201-983-0453


Sep  9 20:59:53 63.248.55.245:7777 -> MY.NET.204.126:4855 UDP to
Sep 14 17:00:01 63.248.55.245:7777 -> MY.NET.204.126:2682 UDP

Sep  9 20:43:01 63.248.55.245:7777 ->     MY.NET.204.166:1519 UDP to
Sep 10 23:58:13 63.248.55.245:7777 -> MY.NET.204.166:1200 UDP
Sep  9 20:43:01 63.248.55.245:7777 ->     MY.NET.213.10:3969 UDP to
Sep 10 00:40:24 63.248.55.245:7777 ->     MY.NET.208.238:1227 UDP
```

```
Sep 11 20:51:59 63.248.55.245:7777 -> MY.NET.208.58:1055 UDP to
Sep 14 16:18:53 63.248.55.245:7777 -> MY.NET.208.58:1428 UDP

Sep 13 22:02:14 63.248.55.245:7777 -> MY.NET.213.78:1068 UDP to
Sep 14 17:00:01 63.248.55.245:7777 -> MY.NET.213.78:2526 UDP
```

```
Flashcom, Inc. (NETBLK-NETBLK-FLASHCOM-2)
   Huntington Beach, CA 92649 US
   Netname: NETBLK-FLASHCOM-2
   Netblock: 63.248.0.0 - 63.248.255.255
   Coordinator: Benton, Curtis  (CB373-ARIN)  curtisb@flashcom.com
                      (714) 891-7891
```

3.2.2.4 MY.NET.X.X Source Port 53,

Probably Normal, Local DNS Server traffic, more information required to know if these machines, MY.NET.1.3, MY.NET.1.4, & MY.NET.1.5, are DNS Servers but at this point it would appear that they are.

```
Aug 15 09:37:02 MY.NET.1.4:53 -> MY.NET.101.99:1088 UDP  To various hosts and ports greater than 1024 in MY.NET.X.Y
Sep  3 09:03:25 MY.NET.1.4:53 -> MY.NET.120.32:1302 UDP
Aug 16 15:44:15 MY.NET.1.3:53 -> MY.NET.101.89:53051 UDP To various hosts and ports greater than 1024 in MY.NET.X.Y
Sep  3 09:03:21 MY.NET.1.3:53 -> MY.NET.152.15:1986 UDP
Sep  3 09:03:18 MY.NET.1.5:53 -> MY.NET.179.78:2082 UDP To various hosts and ports greater than 1024 in MY.NET.X.Y
Sep  3 09:03:22 MY.NET.1.5:53 -> MY.NET.111.169:1932 UDP
```

3.2.2.5 MY.NET.X.X Source and Destination Port 123,

Probably Normal, Network Time Protocol traffic, more information required to know if these machines, MY.NET.1.4, & MY.NET.1.5, are NTP Servers.

```
Sep  3 09:03:20 MY.NET.1.5:123 -> MY.NET.179.54:123 UDP
Sep  3 09:03:22 MY.NET.1.4:123 -> MY.NET.100.96:123 UDP
```

3.2.2.6 MY.NET.1.13 Source Port 7003, & 40531

MY.NET.1.13 is probably a compromised host not an X-Windows font Server.

The following are scans for X-Windows Font Server (if safe) and Remote Grab, Kazimas, Exploit Translation Server, SubSeven 2.1 Gold Trojans (if unsafe)
   threat-sub7 http://www.commodon.com/threat/threat-sub7.htm
   Xforce SubSeven Info http://xforce.iss.net/static/2245.php

The Simovits Consulting Trojan listing identifies this as potentially, SubSeven 2.1 Gold, Remote Access / ICQ Trojan http://www.simovits.com/nyheter9902.html

```
Sep  3 09:03:22 MY.NET.1.13:7003 -> MY.NET.53.207:7001 UDP
Sep  3 09:03:19 MY.NET.1.13:40577 -> MY.NET.6.20:111 UDP
Sep  3 09:03:19 MY.NET.1.13:7003 -> MY.NET.60.164:7001 UDP
Sep  3 09:03:19 MY.NET.1.13:7003 -> MY.NET.100.83:7001 UDP
Sep  3 09:03:19 MY.NET.1.13:7003 -> MY.NET.110.82:7001 UDP
Sep  3 09:03:22 MY.NET.1.13:7003 -> MY.NET.60.170:7001 UDP
Sep  3 09:03:20 MY.NET.1.13:7003 -> MY.NET.53.110:7001 UDP
Sep  3 09:03:21 MY.NET.1.13:7003 -> MY.NET.53.76:7001 UDP
Sep  3 09:03:21 MY.NET.1.13:7003 -> MY.NET.53.149:7001 UDP
Sep  3 09:03:21 MY.NET.1.13:7003 -> MY.NET.60.182:7001 UDP
Sep  3 09:03:21 MY.NET.1.13:7003 -> MY.NET.60.175:7001 UDP
Sep  3 09:03:21 MY.NET.1.13:7003 -> MY.NET.60.167:7001 UDP
Sep  3 09:03:22 MY.NET.1.13:7003 -> MY.NET.60.12:7003 UDP
Sep  3 09:03:22 MY.NET.1.13:7003 -> MY.NET.53.106:7001 UDP
Sep  3 09:03:22 MY.NET.1.13:40531 -> MY.NET.6.33:7008 UDP
```

3.2.3 Third Generic Type

The following, are scans, of a single machine looking for open ports. The destination IP address stays the same the ports change, also referred to a “Vertical Scan” .This is a dangerous situation, as generally these machines may have already been targeted, as having a specific set of features, that can be exploited. The intruder is now checking which ports are available to be exploited, or a specific port to be used for a known exploit.

```
Sep  8 15:09:35 159.226.185.4:41023 -> MY.NET.97.199:10242 UDP to
Sep  8 15:10:48 159.226.185.4:64959 -> MY.NET.97.199:52607 UDP
```

```
The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
   Institute of Computing Technology Chinese Academy of Sciences
   Beijing 100080, China
   Netname: NCFC
   Netnumber: 159.226.0.0
   Coordinator: Qian, Haulin  (QH3-ARIN)  hlqian@NS.CNC.AC.CN
                      +86 1 2569960
```

```
Sep  8 15:10:33 210.125.174.11:60053 -> MY.NET.97.199:28167 UDP  to
Sep  8 15:19:56 210.125.174.11:53952 -> MY.NET.97.199:12489 UDP
Various ports, both source & destination, just this host
inetnum:     210.125.128.0 - 210.125.255.255
descr:       Korean Education Network
descr:       San 56-1, Shilrim-dong, Kwanak-gu, Seoul, Korea
person:      Eunkyung Kim
phone:       +82 2 880 5364
fax-no:      +82 2 887 0130
e-mail:      mgr@kren.nm.kr
```

```
Sep  4 11:42:13 216.99.200.242:16589 -> MY.NET.97.216:953 SYN **S*****
Aracnet Internet Services (NETBLK-ARACNET-COM-1)
Beaverton, OR 97005-2241 US
Netname: ARACNET-COM-1
Netblock: 216.99.192.0 - 216.99.223.255
Coordinator: aracnet.com  (AN44-ORG-ARIN)  noc@ARACNET.COM
                      (503) 626-7696
```

```
Sep  4 20:07:27 24.180.174.167:4523 -> MY.NET.60.11:1545 SYN **S******
Sep  4 20:50:18 24.180.174.167:2751 -> MY.NET.253.42:9876 SYN **S*****
@Home Network (NETBLK-BLTMMD1-MD-2)
   Redwood City, CA 94063 US
   Netname: BLTMMD1-MD-2
   Netblock: 24.180.160.0 - 24.180.175.255
   Coordinator: Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
                    (650) 556-5599

Sep  4 20:25:26 209.123.109.175:1575 -> MY.NET.219.118:24 SYN **S*****
Sep  5 19:35:37 209.123.109.175:2975 -> MY.NET.207.74:528 SYN **S*****
Net Access Corporation (NETBLK-NAC-NETBLK02)
Newton, NJ 07860
Netname: NAC-NETBLK02
Netblock: 209.123.0.0 - 209.123.255.255
Coordinator: Pavely, Ryan  (RP2938-ARIN)  paradox@NAC.NET
                     201-983-0725 (FAX) 201-983-0453

Sep  5 17:11:15 131.155.192.220:2510 -> MY.NET.5.7:257 SYN **S*****
Eindhoven University of Technology (NET-TUEINDHOVEN)
Eindhoven NETHERLANDS
Netname: TUENET1
Netnumber: 131.155.0.0
Coordinator: Schillemans, Joop F.A.  (JFAS-ARIN)  rcjoop@URC.TUE.NL
                    +31 40-472147


Sep  5 17:11:49 147.208.171.139:2682 -> MY.NET.98.160:2005 SYN **S*****
Intel Corporation (NET-INTEL-FSO)
Santa Clara, CA 95052-8119 US
Netname: INTEL-FSO
Netnumber: 147.208.0.0
Coordinator: Sedayao, Jeff  (JS751-ARIN)  sedayao@ORPHEUS.SC.INTEL.COM
                     (408) 765-2935 (FAX) (408) 653-0449

Sep  8 12:50:51 207.123.169.54:5173 -> MY.NET.220.190:593 SYN **S*****
Sep 11 09:22:09 207.123.169.54:22706 -> MY.NET.217.206:339 SYN **S*****
Sep 11 09:22:30 207.123.169.54:13490 -> MY.NET.202.150:988 SYN **S*****
BBN Planet (NETBLK-BBN-PLANET)
Cambridge, MA 02173 US
Netname: BBN-PLANET
Netblock: 207.120.0.0 - 207.123.255.255
 Coordinator:BBN Network Operations Center  (BNOC-ARIN)  ops@BBNPLANET.COM
```

800-632-7638 617-873-8730 fax: 617-873-6315

Sep 8 18:56:33 **151.196.73.119**:8360 -> **MY.NET.253.112**:118 SYN **S*****
Dixie Printing & Packaging (NETBLK-DIXIE-196-73)
Glen Burnie, MD 21061 USA
Netname: DIXIE-196-73
Netblock: **151.196.73.0 - 151.196.73.63**
Coordinator: Ongoing Business Support Services (OBS-ORG-ARIN) business-support@MERCURY.BALINK.COM
800-475-7840 Fax- 703-453-6770

Sep 9 17:34:21 **147.208.171.139**:2739 -> **MY.NET.97.230**:31792 SYN **S*****
Intel Corporation (NET-INTEL-FSO)
Santa Clara, CA 95052-8119 US
Netname: INTEL-FSO
Netnumber: **147.208.0.0**
Coordinator: Sedayao, Jeff (JS751-ARIN) sedayao@ORPHEUS.SC.INTEL.COM
(408) 765-2935 (FAX) (408) 653-0449

Sep 10 18:23:59 **216.234.161.76**:3660 -> **MY.NET.218.34**:393 SYN **S*****
Tera-Byte Online Services (NETBLK-TERA-BYTE-1)
Edmonton, AB T5J0K1 CA
Netname: TERA-BYTE-1
Netblock: 216.234.160.0 - 216.234.191.255
Coordinator: Network Operations Centre (NO58-ORG-ARIN) noc@TERA-BYTE.COM
+1-780-413-1868 Fax- +1-780-413-1869

Sep 13 16:52:25 **216.99.200.242**:28883 -> MY.NET.98.188:648 SYN **S*****
starts out TCP, changes to UDP
Sep 13 17:06:25 **216.99.200.242**:56815 -> MY.NET.98.188:5002 UDP
Aracnet Internet Services (NETBLK-ARACNET-COM-1)
Beaverton, OR 97005-2241 US
Netname: ARACNET-COM-1
Netblock: **216.99.192.0 - 216.99.223.255**
Coordinator: aracnet.com (AN44-ORG-ARIN) noc@ARACNET.COM
(503) 626-7696

Sep 14 04:42:35 **207.230.248.254**:6699 -> **MY.NET.208.18:**4617 NULL ********
In2net Network Inc. (NETBLK-IN2NETT-BLK-1)
Vancouver, BC V6Z 1N9 CA
Netname: IN2NETT-BLK-1
Netblock: **207.230.248.0 - 207.230.248.255**
Coordinator: Lai, David (DL552-ARIN) david@LYNX.BC.CA

604-331-2511

3.2.4 Fourth Generic Type

This category is a special case where it appears that MY.NET.217 is communicating out to a group of nodes at 198.62.155.x, possibly a cluster of nodes serving a specific need or MY.NET.217.10 is compromised.

```
Aug 28 15:38:59 198.62.155.109:40490 -> MY.NET.217.10:1490 SYN **S*****
Aug 28 15:38:57 198.62.155.10:39850 -> MY.NET.217.10:2032 SYN **S*****
Aug 28 15:38:57 198.62.155.11:39851 -> MY.NET.217.10:868 SYN **S*****
Aug 28 15:38:57 198.62.155.104:39868 -> MY.NET.217.10:665 SYN **S*****
Aug 28 15:38:57 198.62.155.102:39899 -> MY.NET.217.10:471 SYN **S*****
Aug 28 15:39:00 198.62.155.105:40595 -> MY.NET.217.10:984 SYN **S*****
Aug 28 15:39:00 198.62.155.111:40546 -> MY.NET.217.10:1482 SYN **S*****
Aug 28 15:39:00 198.62.155.101:40547 -> MY.NET.217.10:1992 SYN **S*****
Aug 28 15:39:00 198.62.155.106:40541 -> MY.NET.217.10:1013 SYN **S*****
```

The source address changed among the nodes 103,109,10,11,104,102,105,111,101,106,107,103,109,10,11,109,111,101,
the destination address always stayed the same ....

```
Aug 28 15:40:48 198.62.155.106:42978 -> MY.NET.217.10:392 SYN **S*****
Aug 28 15:40:48 198.62.155.102:42974 -> MY.NET.217.10:906 SYN **S*****
Aug 28 15:40:48 198.62.155.104:42976 -> MY.NET.217.10:781 SYN **S*****
Aug 28 15:40:48 198.62.155.103:42975 -> MY.NET.217.10:818 SYN **S*****
```

Smart Consulting (NET-SMART-NET)
Fredrick, MD 21702 US
Netname: SMART-NET
Netnumber: 198.62.155.0
Coordinator: Smart, Robert (RS622-ARIN) bsmart@GREBYN.COM
301-662-0374

3.2.5 Fifth Generic Type –Tools

3.2.5.1 Nmap

24.180.134.156 Scanned subnets indicated below, using various source ports, caught as an NMAP scan
Sep 11 04:48:03 24.180.134.156:3089 -> MY.NET.208.1:757 SYN **S*****
Sep 11 04:48:56 24.180.134.156:50111 -> MY.NET.208.5:23 **NMAPID** **SF*P*U
1,2,5,6,9,13,17,18,21,25,29,33,34,37,38,41,45,49,50,53,54,57,61,65,66,69,70,73,74,77,78,81,85,89,93,94,97,98,101,102,105,106,109,110,113,114,117,121,122,125,129,133,137,138,141,145,146,149,153,154,157,161,165,166,169,173,177,178,181,182,185,186,189,190,193,197,201,205206,209,213,214,217,221,225,226,229,230,233,237,241,245
Sep 11 05:19:13 24.180.134.156:50108 -> MY.NET.208.245:23 SYN **S*****
@Home Network (NETBLK-BLTMMD1-MD-1)
Redwood City, CA 94063 US
Netname: BLTMMD1-MD-1
Netblock: **24.180.128.0 - 24.180.143.255**
Coordinator: Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

3.2.5.2 QUESO

Sep 11 09:13:54 64.80.63.121:1524 -> MY.NET.224.34:6346 SYN 21S***** RESERVEDBITS
The following is supporting evidence from the Analysis of Type 1 data
09/11-09:13:54.380467 [**] **Queso fingerprint** [**] 64.80.63.121:1524 -> MY.NET.224.34:6346
64.80.63.121
CollegePark/LexingtonCrossing (NETBLK-PAET-MI-CPRK-LEX)
Gainesville, FL 32608 US
Netname: PAET-MI-CPRK-LEX
Netblock: 64.80.63.0 - 64.80.63.255
Coordinator: Darby, Brian (BD114-ARIN) bdarby@campuslink.com
734-975-8075

Sep 13 19:21:29 24.3.161.193:33044 -> MY.NET.145.9:110 SYN 21S***** RESERVEDBITS
The following is supporting evidence from the Analysis of Type 1 data
09/05-09:00:54.631991 [**] **Queso fingerprint** [**] 24.3.161.193:32814 -> MY.NET.145.9:110
@Home Network (NETBLK-NJ-COMCAST-UNION-1)
Redwood City, CA 94063 US
Netname: NJ-COMCAST-UNION-1
Netblock: 24.3.160.0 - 24.3.175.255
Coordinator: Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

3.3 Analysis of Type 3 Data

This section of data contains packets that have various illegal bit combinations set. It has a number of nodes communicating with MY.NET.X.X, but more importantly some **Out Of Band traffic headed out** from MY.NET.X.X This information can be found under the heading **Suspicious Traffic**, below

The information appears to have been collected with a filter that detects the SYN-FIN flags together at the same time.

210.61.144.125 This address has already been mentioned in the Type 1 (SYN-FIN scan) and Type 2 (Scans to port 21) analysis
Standard FTP issues mentioned earlier

```
09/11-06:45:14.854416 210.61.144.125:21 -> MY.NET.1.3:21
starts and maps the following subnets 1,2,4,5,6,7,9,10,11,12,13,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85,94,97,98,99,100, 104,105,106,107,108,109,110,111,112,115,120,130,138,139,140,141,142,143,144,145,146,150,151,152,153,154,155,156,157,158,159,160,161,162,163,178,179180,181,182,183,184,185,186,188,190,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225226,227,228,229,230,231,232,253,254
09/11-07:06:50.833774 210.61.144.125:21 -> MY.NET.254.254:21
inetnum:     210.61.144.0 - 210.61.144.255
netname:     HINET8-144-TW
descr:       Abnet Information Co., Ltd
descr        Taipei, Taiwan TW
person:      Wen-Lon Li
address:     Abnet Information Co., Ltd
phone:       +886-2-558-2115
fax-no:      +886-2-558-2116
e-mail:      abnet@ms15.hinet.net
```

213.25.136.60 This address has already been mentioned in the Type 1 (SYN-FIN scan) and Type 2 (Source and Destination Port 9704) analysis
Exposure

CVE-1999-0048 Talkd, when given corrupt DNS information, can be used to execute arbitrary commands with root privileges**.**
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0048

```
09/07-21:33:30.514289 213.25.136.60:9704 -> MY.NET.1.4:9704
starts and maps the following subnets 1,2,4,5,6,7,9,10,11,12,13,MY.NET.14.2,15,17,18,20,21,25,26,53,54,60,68,69,70,71,75,85
09/07-21:40:44.000483 213.25.136.60:9704 -> MY.NET.85.254:9704
inetnum:     213.25.136.0 - 213.25.136.15
netname:     E-SOLUTIONS
descr:       e-SOLUTIONS.com Poland Sp. z o.o. PL
person:   Wieslaw Kosidlak e-mail:     wkosidlak@mfrelas.com
phone:       +48 81 7453340
fax-no:      +48 81 7453315
```

```
09/01-00:07:56.201741 24.108.139.90:191 -> MY.NET.221.66:1438
09/01-00:11:19.168802 24.108.139.90:1438 -> MY.NET.221.66:6355 one machine, one port
Videon CableSystems Alberta Inc (NET-VIDEONCABALB)
   Edmonton, Alberta T5S 1S2 CA
   Netname: VIDEONCABALB
   Netblock: 24.108.0.0 - 24.109.15.255
   Coordinator: Kevin, Patzer  (PK110-ARIN)  k.patzer@videon.ca
                       780.486.6892

09/06-13:58:22.763407  [**] Null scan! [**] 24.113.80.28:1993 ->    MY.NET.203.110:1464 from Type 1 analysis
09/01-07:44:41.858576 24.113.80.28:2439 -> MY.NET.207.34:2272 to
09/01-08:49:10.942426 24.113.80.28:0 -> MY.NET.207.34:2439 using the two ports noted
Rogers@Home Lngly (NETBLK-BC-ROG-LNGL-2)
Toronto, ON M4Y 2Y5 CA
   Netname: BC-ROG-LNGL-2
   Netblock: 24.113.80.0 - 24.113.81.255
   Coordinator: Network Security, Fraud  (AD30-ARIN)  abuse@rogers.home.net
                       (416) 935-4729

09/11-04:48:57.783879 24.180.134.156:50111 -> MY.NET.208.5:23 to
09/11-05:19:11.134298 24.180.134.156:50111 -> MY.NET.208.245:23,  208 subnet, port 23
@Home Network (NETBLK-BLTMMD1-MD-1)
   Redwood City, CA 94063 US
   Netname: BLTMMD1-MD-1
   Netblock: 24.180.128.0 - 24.180.143.255
   Coordinator: Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
                       (650) 556-5599
```

Suspicious traffic

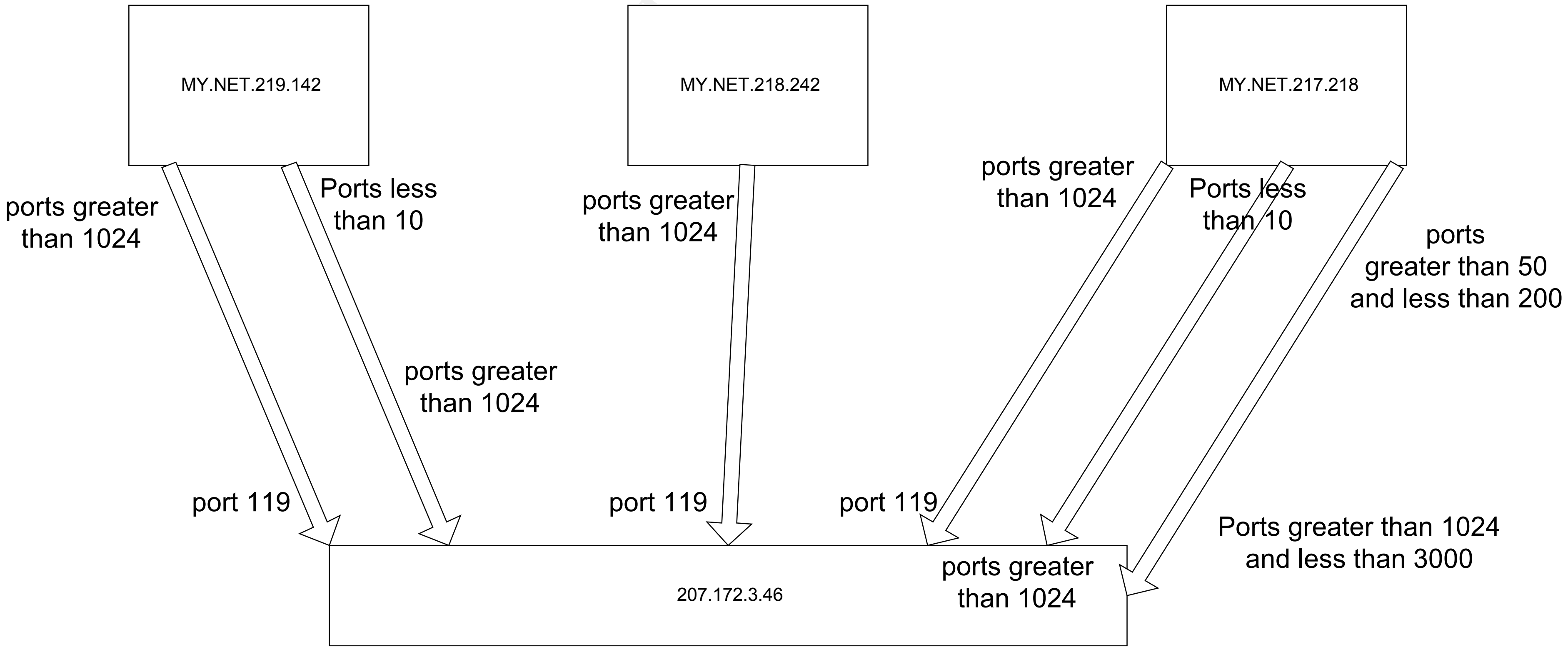


Figure 2
Suspicious Traffic
that had SYN and Fin set
as well as combinations of
other flags

Description

The following traffic has been detected outgoing from MY.NET.219.142, MY.NET.218.242, and MY.NET.217.218 it is all going to 207.172.3.46. The server 207.172.3.46 claims to be a News server, the traffic data that has been collected by the probe would indicate otherwise. There are many illegal flag combinations set in the data. The overall basis for the detect is discussed below with a description of the three way handshake.

The Three Way Handshake works as follows, a node (client) wishing to communicate with another node (server) establishes the TCP connection with a TCP packet containing a SYN, the initial sequence number it wishes to use, to the well known port it wishes to connect to, the Maximum Segment Size (MSS) and Maximum Transmission Unit (MTU) size. If the server is willing and able to establish a connection, it responds with a packet, which contains a SYN-ACK, increments the client's initial sequence number by 1, supplies the server's initial sequence number, MSS and MTU. If that port is not active on the server, a reset is sent instead to the client node. The client responds to the server's SYN with an ACK, and increments the server's initial sequence number by 1. The initial sequence numbers are incremented as appropriate to each node, with each transfer of data. Communication then continues until the session is complete. The start of session tear down is initiated with a FIN from, which ever node has completed first. The other node responds with an ACK, and if it has completed its portion of the session, as well, it responds with a FIN, if it has not completed its side of the communication it continues until it is complete. The node that has sent its FIN will continue to respond even though it may already have sent a FIN, until both ends of the conversation have sent a FIN, to which the other node responds with an ACK.. As can be observed, the combination of SYN and FIN do not naturally occur together at the same time, in a normal session. Therefore one can conclude that the packets having this combination of flags, as a minimum must be crafted.

```
08/29-14:38:43.705179 MY.NET.219.142:1052 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:31489  DF
*1SF**AU Seq: 0xA   Ack: 0x5F4EE3F8   Win: 0x5010
04 1C 00 77 00 00 00 0A 5F 4E E3 F8 01 B3 50 10  ...w...._N....P.
05 B4 2F CB 20 20 20 20 20 00                    ../.     .

08/29-14:39:11.535016 MY.NET.219.142:1052 -> 207.172.3.46:119 to
08/29-17:08:30.879081 MY.NET.219.142:1137 -> 207.172.3.46:119

09/01-03:30:24.867890 MY.NET.218.242:1075 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:16403  DF
21SF**** Seq: 0xC1660   Ack: 0x19ECC   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 6515 CA33 82A3 0014 0000 0000 0000 0000 0000 0000 0000 0000 0000 EOL EOL EOL
EOL EOL EOL EOL EOL

09/01-03:30:24.867890 MY.NET.218.242:1075 -> 207.172.3.46:119 to
09/01-04:57:08.155219 MY.NET.218.242:1102 -> 207.172.3.46:119

09/01-17:06:17.833362 MY.NET.217.218:1099 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:55949  DF
**SF*P*U Seq: 0xE2B0AB   Ack: 0xFC13   Win: 0x5010
00 00 FC 13 2D 2B 50 10 22 38 7F 03 20 20 20 20  ....-+P."8..
20 00                                             .

09/01-17:06:17.833362 MY.NET.217.218:1099 -> 207.172.3.46:119 to
09/14-02:29:29.560104 MY.NET.217.218:1337 -> 207.172.3.46:119
```

```
 207.172.3.46 Canonical name: reader4.news.rcn.net

Erol's Internet Services (NETBLK-NETBLK-EROLS-BLK-3)
   Springfield, VA 22151
   Netname: NETBLK-EROLS-BLK-3
   Netblock: 207.172.0.0 - 207.172.255.255
   Coordinator: Network Operations Center  (EROLS-NOC-ARIN)  noc@RCN.COM
      703-321-8000 Fax- 703-321-8316
```

Example 1

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/04-23:09:17.423574 MY.NET.217.218:0 -> 207.172.3.46:1156
TCP TTL:126 TOS:0x0 ID:62050  DF
**SF*PAU Seq: 0x77000A   Ack: 0x1A024DB0   Win: 0x5010
1A 02 4D B0 21 3B 50 10 22 38 D0 9F 20 20 20 20  ..M.!;P."8..
20 00                                             .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/04-23:10:36.591951 MY.NET.217.218:0 -> 207.172.3.46:1156
TCP TTL:126 TOS:0x0 ID:56176  DF
2*SFRPA* Seq: 0x77000A   Ack: 0x1B014E1C   Win: 0x5010
20 20 20 20 20 00                                     .

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

Example 2

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/05-13:02:54.730088 MY.NET.217.218:0 -> 207.172.3.46:1074
TCP TTL:126 TOS:0x0 ID:13082  DF
**SFRPAU Seq: 0x77007C   Ack: 0xD89835A7   Win: 0x5010
D8 98 35 A7 20 3F 50 10 22 38 2A EE 20 20 20 20  ..5. ?P."8*.
20 00                                             .

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

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/05-13:15:28.575665 MY.NET.217.218:0 -> 207.172.3.46:1074
TCP TTL:126 TOS:0x0 ID:36189  DF
**SF*P*U Seq: 0x77007C   Ack: 0xE8483EC9   Win: 0x5010

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

Example 3

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/05-16:28:43.201535 MY.NET.217.218:2328 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:18313  DF
**SF*P** Seq: 0x12F   Ack: 0xEA7D8CA8   Win: 0x5010
00 00 01 2F EA 7D 8C A8 10 0B 50 10 1C 84 D2 56  .../.}....P....V
20 20 20 20 20 00                                     .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/05-16:33:10.619588 MY.NET.217.218:2328 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:13511  DF
**SFR*AU Seq: 0x12F   Ack: 0xEDFD8E89   Win: 0x5010
20 00                                             .

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

Example 1, 2 and 3 demonstrate the existence of packets with the same Sequence number, over a very short period of time. The sequence number field is a 4 byte field. It does not wrap in a normal TCP conversation, unless a very large number of bytes of information are transferred, certainly not within 1 minute as in Example 1, or less than 13 minutes as in Example 2 or in less than 5 minutes as in Example 3. If the packets were TCP retry, the remainder of the contents of the packet would be the same, they are not. Therefore one has more evidence that these are crafted packets being sent out from MY.NET.217.218.

The probe is collecting data from only one view point, that of its position in the network, and the filter that has been loaded into it. In order to determine with some certainty, exactly what the compromise is, would require capturing the data flow in both directions from the 3 compromised hosts.

Another set of issues is due to the NAPSTER/InternetRelayChat/Trojan set of ports in use as part of this activity, examples below.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/08-02:31:35.313799 MY.NET.217.218:6699 -> 128.118.215.123:1823
```

```
TCP TTL:126 TOS:0x0 ID:4697  DF
21SFRPA* Seq: 0x2B0056   Ack: 0x217B08B8   Win: 0x5010
21 7B 08 B8 22 DF 50 10 22 38 6A C8 20 20 20 20  !{..".P."8j.
20 00                                             .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/08-02:32:05.728602 MY.NET.217.218:6699 -> 128.118.215.123:1823
TCP TTL:126 TOS:0x0 ID:5739  DF
21SFRPA* Seq: 0x550056   Ack: 0x217B0906   Win: 0x5010
1A 2B 07 1F 00 55 00 56 21 7B 09 06 0A DF 50 10  .+...U.V!{....P.
22 38 82 7A 20 20 20 20 20 00                    "8.z      .

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

08/28-09:55:13.260265 MY.NET.202.202:1694 -> 128.61.68.140:6699 **Georgia Institute of Technology** (NET-GATECH-EDU)

08/29-01:10:11.515712 MY.NET.201.82:2021 -> 141.161.120.190:6699 **Georgetown University** (NET-GEORGETOWN-NET)

09/04-03:11:49.883185 MY.NET.218.74:1169 -> 198.164.220.55:6699 **University of New Brunswick** (NETBLK-NEWBRUN) NBNET001

09/04-14:09:22.979271 MY.NET.222.110:1325 -> 193.129.5.70:6699 UUNET UK

09/06-21:39:04.979855 MY.NET.208.102:1607 -> 216.161.236.5:6699 U S WEST Interact Services (NETBLK-USW-INTERACT98)

09/08-14:39:40.758761 MY.NET.210.206:1092 -> 171.64.115.13:6699 **Stanford University** Network (NETBLK-NETBLK-SUNET)

09/12-11:39:21.498008 MY.NET.209.94:1065 -> 169.229.90.83:6699 **University of California**, Office of the President

09/12-11:49:34.238617 MY.NET.209.94:1071 -> 152.17.106.86:6699 **Wake Forest University** (NET-WAKE-FOREST)

09/13-12:45:19.296837 MY.NET.222.82:0 -> 169.229.117.60:6699 **University of California**, Office of the President

The following nodes have all transmitted packets with illegal combinations of flags. They should be evaluated for battle damage. The severity of the compromise on these hosts is to a large degree dependent on site policy, as the activity show signs of Napster/IRC activity. These nodes should be checked for known virus infections, and checked that they are running the latest versions of software. If Napster/IRC activity is not allowed on this site then stronger action should be taken as per site policy.

MY.NET.201.82, MY.NET.201.110,MY.NET.201.146,
MY.NET.202.10,MY.NET.202.26,MY.NET.202.50,MY.NET.202.102,MY.NET.202.202,
MY.NET.203.98,
MY.NET.204.74,MY.NET.204.78,
MY.NET.205.190,MY.NET.205.226,
MY.NET.206.26,MY.NET.206.134,MY.NET.206.162,MY.NET.206.182,
MY.NET.208.6,MY.NET.208.102,MY.NET.208.178,MY.NET.208.162,
MY.NET.209.94,
MY.NET.210.150,MY.NET.210.206,MY.NET.210.218,
MY.NET.211.182,
MY.NET.212.6,
MY.NET.217.54,MY.NET.217.154,MY.NET.217.206,MY.NET.217.218,MY.NET.217.222,MY.NET.217.242,
MY.NET.218.14,MY.NET.218.74,MY.NET.218.82,MY.NET.218.154,MY.NET.218.158,MY.NET.218.242,
MY.NET.219.30,MY.NET.219.130,MY.NET.219.142,MY.NET.219.178,MY.NET.219.230,
MY.NET.220.10,MY.NET.220.18,MY.NET.220.82,MY.NET.220.114,MY.NET.220.134,MY.NET.220.142,MY.NET.220.190,
MY.NET.221.218,
MY.NET.222.82,MY.NET.222.110,MY.NET.222.198,MY.NET.222.210,MY.NET.222.218,MY.NET.222.250,
MY.NET.223.14,MY.NET.223.26,MY.NET.223.54,
MY.NET.226.22,MY.NET.226.234

Known Issues

It is difficult with this information to know exactly what the compromise is. It has the features of a remote control program (Trojan) of some nature. It is impossible to separate the Napster/IRC traffic from the Trojan traffic, without more information directly from the compromised nodes.
CIAC-2318 IRC On Your Dime? http://ciac.llnl.gov/ciac/documents/CIAC-2318_IRC_On_Your_Dime.pdf
RFC1459, IRC Protocol http://www.irchelp.org/irchelp/rfc/
The above links describe some of the issues.

Exposure

One has to conclude that 207.172.3.46 is the server in this situation and the MY.NET.X.Y hosts are the clients. There are likely other external hosts participating as well. The hosts MY.NET.219.142, MY.NET.218.242, and MY.NET.217.218 are certainly compromised. The full packets have to be captured to attempt to decode the activity. This may not be possible depending on the type of communication, as some covert channels are encrypted. The hosts larger group of hosts listed above are participating in IRC type activities , and if they are not compromised now, without strong precautions, soon will be. They are transmitting packets with illegal combinations of flags, that alone suggests a problem.

Assignment 4 – Analysis Process

A decision was made after Stephen Northcott's presentation on Friday morning October 20th that the Practical Exam needed a process to be followed in order to have the opportunity to be successful with it. It was calculated that there was roughly 4 weeks to complete the Practical. It became obvious to me after some careful consideration that:

The first week should be spent gathering and analyzing all available information, getting the general format of the presentation down, having a look at the data to be analyzed, and running it through a few scripts (if any were to be found).

The second week would be spent in completing the "Analyze This" section. This included the analysis and writing it up.

About 2/3'rd's of the third week would be spent completing the "Evaluate an Attack".

Four days would be spent on the analysis of the 4 Network Detects.

Any remaining time would be used in cleanup of the whole Practical Assignment.

The first Monday back after the course (I had additional classes on Saturday and Sunday), I downloaded the information and was still surprised at (scared by) the volume of information. This almost made me drop the plan, instead I decided to PANIC for the first 15 minutes and get that out of the way, so that the PANIC, wouldn't have to be done at "crunch time" at the end , then I could get back to the plan.

I looked through the Practicals that other students had submitted for DC and Parlament Hill, downloaded the ones marked as Honours, and a few others that had what I thought to be good information and started reading. By the end of the first week I had read the 3 recent Practicals that had attained an Honors status, scanned about 6 or so others and written up 2 of the Network detects.

By the end of the second week I had some of the "Analyze This" section complete. The method that I chose to look at the data, was to put all of the Type 1 files together into 1 massive file, run a Perl script by Wynn Fenwick, to break out the threat vectors. Then sort the file into smaller pieces using the Unix command "grep", with appropriate modifiers. The smaller files were manually scanned through, using Microsoft Word, Excel, and Notepad as appropriate. Any one of the Microsoft Editors or "grep", were used as appropriate, for further filtering of the data. A similar manual process was used to sort through the other 2 types of datafiles.

I started the third week by completing the "Evaluate an Attack" section. This took a small portion of that week, then back to the "Analyze This" section.

The fourth week was spent completing the 2 remaining sections of 4 Network Detects, the remaining time was again put into "Analyze This" and cleanup.

It became obvious after returning to the "Analyze This " section so many times, that there could be a more effective way to look at this data. It also became obvious that each solution could very easily end up being problem specific. An answer could be to load the data into a database, it would have to be organized to be able to find all common items. Examples are, common attacks, common source addresses, common destination addresses, common tools, and quite possibly also provide a list of items not common to any. This area requires more thought, and possibly more
analysis, before a workable solution becomes obvious.

T hanks go to the following people for their assiatance Kevin Comis, Rick Dallow, and Glenn Davis.

Attached there is an html file (port.html) that I use to keep track of known Viruii, Trojans, and reference material sites.