Global Information Assurance Certification Paper

SANS GIAC Certification Program

GCIA Practical Assignment

Intrusion Detection and Analysis

Adrian Brindley
SANS On-Line
GIAC GCIA Practical Submission (v3.4 Sep 2003)

Table of Contents

Introduction

There have been many SANS articles and papers presented on the use and ethics of honeypots and honeynets within the SANS Reading Room so the intention of Section 1 of this paper is to give a very brief overview of these type of systems, and then to expand upon this further with an explanation and run-through of a currently available commercial honeypot system called KFSensor. Section 2 then details two network detects that have been captured by the honeypot in a home environment and one network detect from the incidents.org website. A detailed 5-day analysis is then carried out for the University based scenario in the final Section.

Assignment #1: Describe the State of Intrusion Detection

KFSensor – a Windows based Low-Interaction honeypot

Honeypots may be implemented on live production systems or used as "research tools" within controlled environments, these can also be further categorized as either low-interaction or high-interaction types of system. The benefit of a honeypot within a production environment is that it allows intruders intentions to possibly be drawn to particular systems (honeypots) allowing valuable time for support personnel to secure the live production systems if required. The benefit of a research honeypot is that it allows capture and analysis of a more diverse range of attacks e.g. Day 0 type exploits and to allow full analysis and monitoring of new and emerging hacker techniques. A honeynet is a more complex arrangement of honeypot systems e.g. Windows / Linux servers grouped together behind a honeywall (a honeywall is similar to a normal firewall and is used to tightly control the traffic direction and rates within the honeynet environment).

The purpose of using the KFSensor software (shortened to KFS for the remainder of this paper) was to increase my understanding of the workings and benefits of a software "emulated" honeypot and to hopefully obtain a number of interesting attacks for analysis within the Network Detects assignment section. KFS can be downloaded from www.kfsensor.com and is provided with a 14-day evaluation licence (following the on-line registration process), alternatively you may request an educational licence to provide an extended evaluation period. The version used for this particular paper was "Educational KFSensor v2.0.2".

KFS Overview

KFS runs on Microsoft Windows platforms (NT, W2K, XP, Server 2003) and is based upon a central program (daemon) that is responsible for providing the necessary "service emulation" and connection management to and from the honeypot. The KFS software does not install any additional device drivers or modify the existing IP stack and relies entirely on

the existing Windows environment for passing parameters via normal application interface calls (user space mode). This allows the honeypot in addition to it's own inherent security functions to be further protected by the additional layer of the host OS system e.g. file access rights / process control. The KFS software can fully emulate many services such as WWW, SMTP, FTP, POP3 and also has an additional version for high security environments. This is a "High Integrity" coded version that has certain services compiled out for high security applications; this ensures that even if a compromise was achieved by some means the honeypot could not be used to launch further attacks via that particular service e.g. WWW or SMTP. Although the system is "based upon a single running program i.e. no direct attacker access to the Operating System services there is always the requirement to apply best practice and apply the latest service packs and patches especially if used as a "research" honeypot that is directly connected to the Internet.

The KFS honeypot is simple and fast to set-up (via use of the Set Up Wizard) but flexible in its approach and uses "multiple scenario" configurations to provide differing port and service responses. Only one scenario may be active at any one time with the honeypot being simply switched to any other scenario as and when required e.g. finger, chargen, LinuxFTP type services for basic Unix/Linux type emulation.

Service Emulation

The honeypot can be configured to respond to stimulus using any combination of the following methods:

1) Simple port open and listening (referred to as a listener) e.g. port scanning – the honeypot responds providing full 3-way handshake for TCP type connections and accepts normal UDP type requests.

2) Banner responses to stimulation e.g. Telnet Login – the honeypot responds with a pre-configured message prompt or error message and this is further enhanced by use of a range of dynamic variables to provide real-time feedback to the attacker e.g. current date / IP addresses in use / port numbers.

3) Full service simulation, there are 11 servers currently defined as in-built "Sim Standard Servers", examples that follow show the output from the clients perspective (where possible):

1. HTTP: This is a fully working web server that emulates Microsoft's IIS web server (IIS) and the default IIS web pages can easily be replaced by your own definitions, example GET output via telnet:

```
Request sent to server: GET / HTTP/999.99

HTTP/1.1 400 Bad Request
Content-Type: text/html
Server: Microsoft-IIS/6.0
Date: Sun, 14 Nov 2003 09:42:54 GMT
Connection: close
Content-Length: 20
```

```
<h1>Bad Request</h1>
```

2. SMTP: The Simple Mail Transfer Protocol emulation is capable of acting as an open relay for SMTP, there are additional safeguards to control interaction with this service (MS-SMTP) and does not allow relaying by default. Example SMTP service simulation from KFS, this has queuing enabled to allow mails to be received but not forwarded, example output:

```
>>>>220 networksforu.com Microsoft ESMTP MAIL Service, Version: 6.0.2600.1106 ready at
Mon, 10 Jun 2003 17:26:21 +0000
HELO qqq-6j4vecjhtdb
>>>>250 networksforu.com Hello [xx.xx.xx.xx]
MAIL FROM:
>>>>250 2.1.0 SuperMan2173912016@hotmail.com....Sender OK
RCPT TO:
>>>>250 2.1.5 ch69v5@hotmail.com
DATA
>>>>354 Start mail input; end with .
From:
To: ch69v5@hotmail.com
Subject: SuperMan – xx.xx.xx.xx
X-Mailer: SuperMail v1.1
Mime-Version: 1.0
Content-Type: text/plain;%09charset=us-ascii

Server Output Test to IP – xx.xx.xx.xx
.
>>>>250 2.6.0 Queued mail for delivery
QUIT
```

3-6. Window networking / NetBIOS / SMB / CIFS (Common Internet File System): KFS can emulate all four of Microsoft's NetBIOS and SMB/CIFS services. NBT Datagram Service (138), NBT Name Service (137), NBT Session Service (139) and NBT SMB (445). *The honeypot does not actively broadcast any NBT datagrams or participate in NBT announcements – it purely receives and transmits directed datagrams via the honeypot interface.

7. FTP: File Transfer Protocol emulation (GuildFTPd), example output:

```
220-anydomain.com
220 Please enter your name:
    USER someone
331 User name okay, Need password.
    PASS (hidden)
530 Password not accepted.
    Cannot login waiting to retry (30s)...
```

8.POP3:Post Office Protocol emulation (MS-POP3)

9.Telnet: Telnet server emulation. (MS-Telnet), example configured output:

```
**Warning** : Un-Authorized Access Prohibited - All communication to this computer system
is monitored continuously.
```

10.Terminal Server: Terminal Server is a Microsoft application that allows remote users to log on to a server (MS-Tserver)

11.VNC: VNC is a cross platform application that is used for remote control access to host(s). The VNC emulation allows hackers to attempt to log on to the service, but rejects all passwords sent. The emulator returns the VNC 003.003 version number to any requests.

A special server that allows full customisation:

External: This a special type server allows you to fully customize the honeypot, it provides this flexibility for service simulation by allowing an “External Console Application” to be configured under it’s “Simulation Server” settings. This allows executables (.exe), PERL scripts (.pl), Python scripts or batch files (.bat) to be invoked when a particular service is probed or attacked. Perl scripts written for Honeyd systems will also run when the environment is correctly configured (e.g. via Perl, Cygwin set-up).

Main KFS Window

The system uses multiple-drop down menus to allow quick configuration and monitoring of the areas required.

Figure 1. Screenshot of the Main KFS Window:

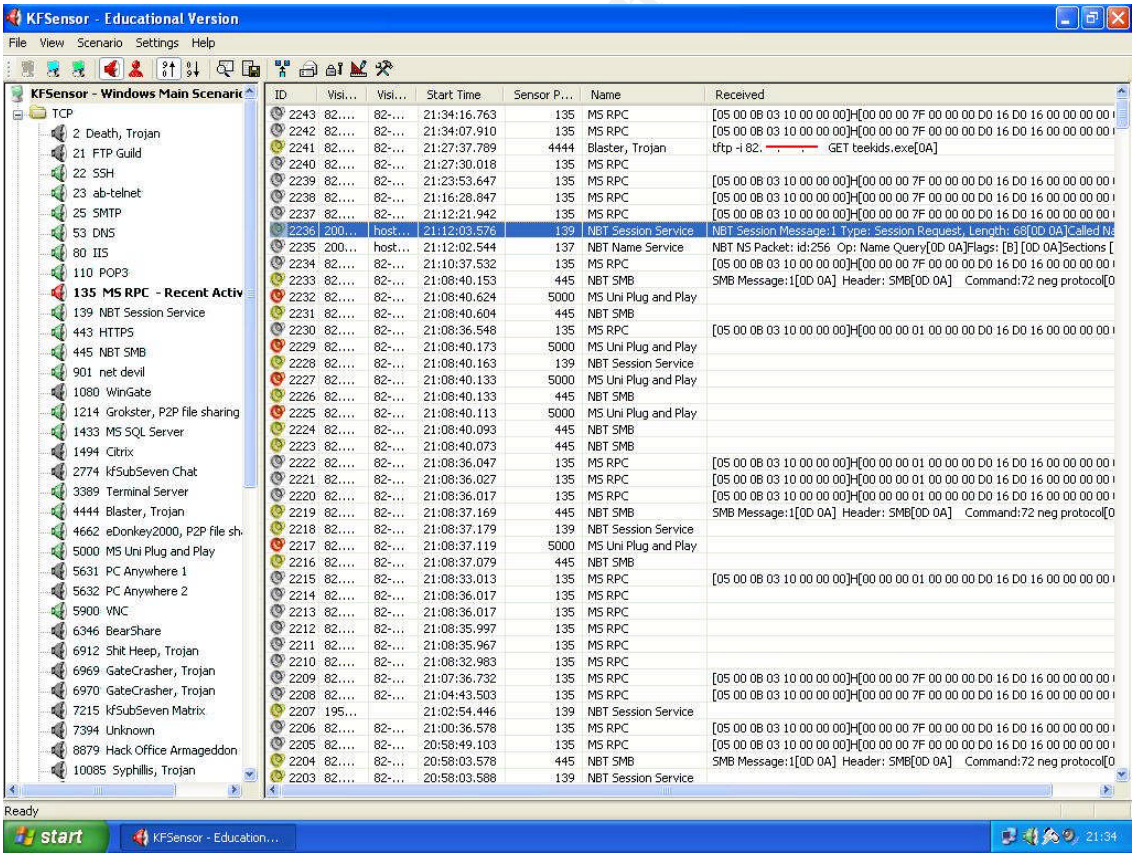


Figure 1 shows the Active Scenario / Listener ports that are currently open, unique Alert ID, Visitor source IP address (obscured), Visitor Domain resolved by KFS (obscured), Start time of Attack, Sensor Port number being attacked, Name of Port and a summary of the Received data. There are further TCP / UDP "listeners" available via the scroll-down bar on the left-hand pane.

Scenario's are used to create default listener ports and services and the default set-up generally enables all services apart from auto-forwarding ports e.g. SMTP.

Figure 2. Edit Scenario window:

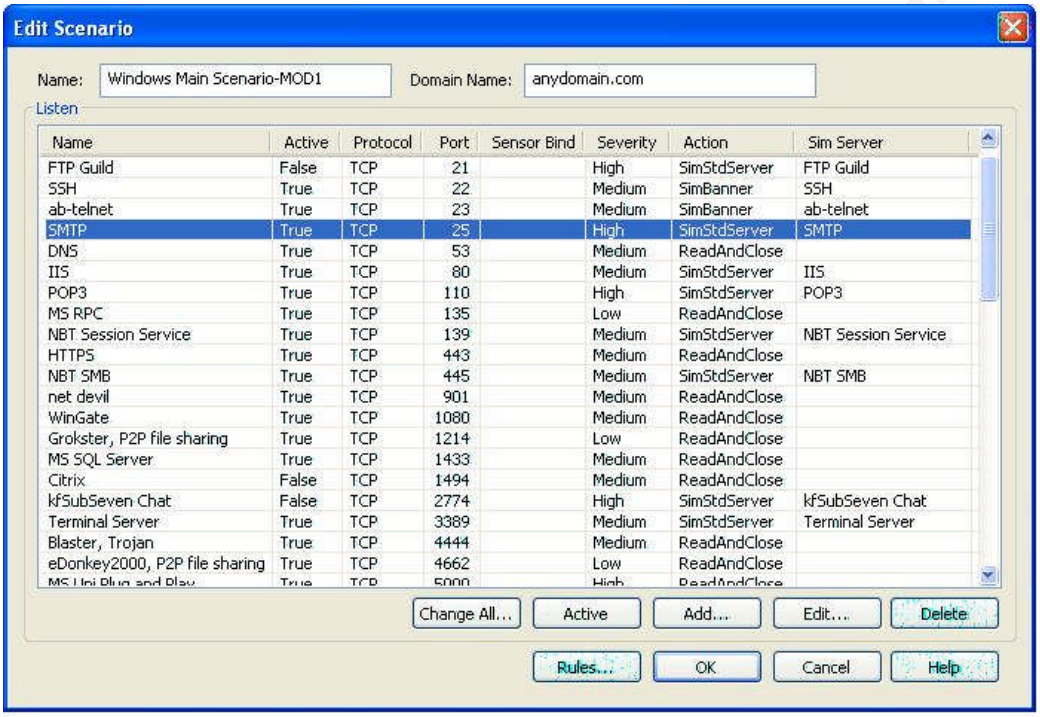


Figure 2 shows the Active Scenario in greater detail with the Windows Main Scenario duplicated to "Windows Main Scenario-MOD1" and a new telnet service that has been added manually (ab-telnet). This window also allows Listener ports to be enabled / disabled, individual event Severity levels to be set and the honeypot domain name to be configured.

Figure 3. Edit Sim Servers window (Std):

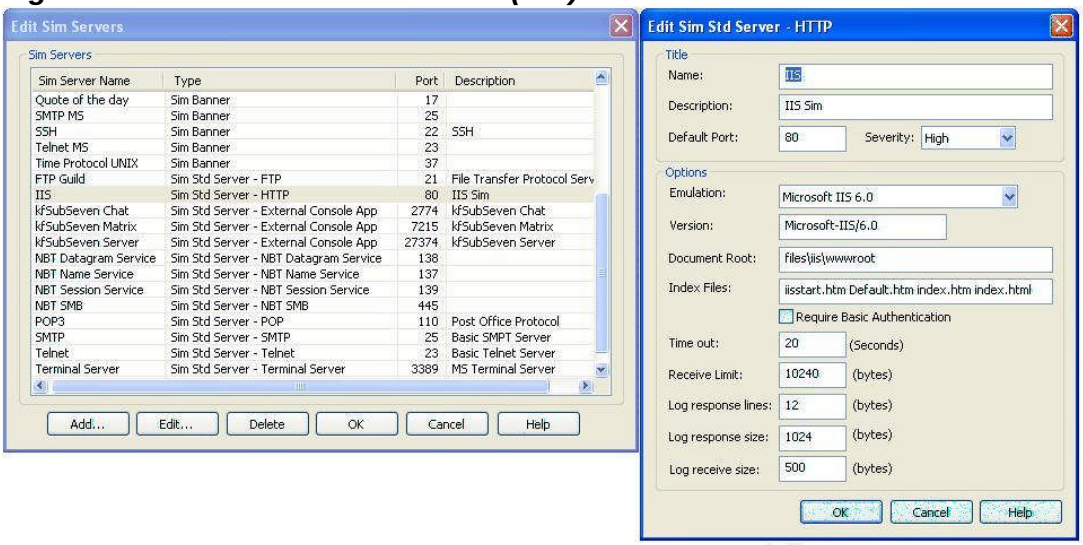


Figure 3 Shows the Edit Sim Servers (Std) window and detailed HTTP simulated server configuration, this allows the IIS parameters to be configured such as document root, index files and time-outs. Further options exist common to all Sim Servers that control the amount of time the server is active for or setting of a maximum byte transfer count. Also shown are the add-on services for SubSeven emulation (kfSubSeven Chat, kfSubSeven Matrix and kfSubSeven Server) that were manually added.

The External Console option allows the flexibility of the honeypot to be increased by providing a method for an unlimited number of additional “services” to be implemented. The window below shows the SubSeven emulator being set-up and tested.

Figure 4. Edit Sim Servers window (External Console):

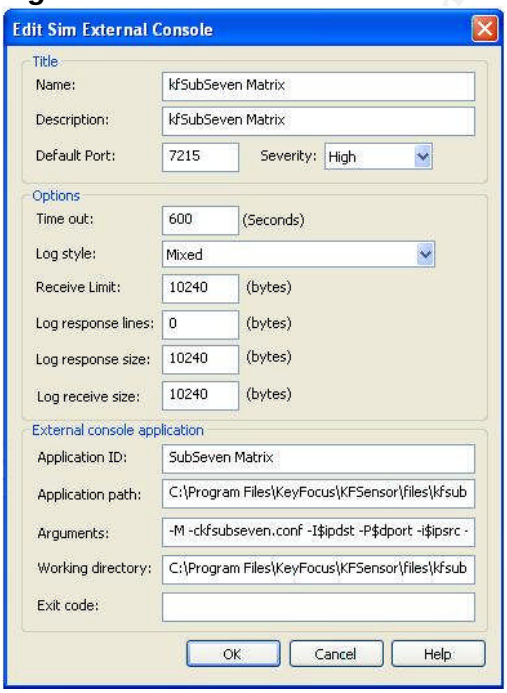


Figure 4. Shows the Edit Sim Servers window (External Console) and an emulation of the SubSeven Matrix function, to allow the new service application to interact correctly parameters need to be passed via the environment variables or command line parameters:

Examples:

Parameter Line Arg	Command	Env. Variable	Example
Sensor IP Address	$ipdst	KFSENSOR_ADDR and HONEYD_IP_DST	192.168.1.10
Sensor Port	$dport	KFSENSOR_PORT and HONEYD_DST_PORT	80
Visitor IP Address	$ipsrc	VISITOR_ADDR and HONEYD_IP_SRC	192.168.1.10
Visitor Port	$sport	VISITOR_PORT and HONEYD_SRC_PORT	3205
Application ID	$appid	KFSENSOR_APPID	Echo
Protocol	$proto	KFSENSOR_PROTOCOL	TCP
Domain Name	$domain	KFSENSOR_DOMAIN	anydomain.com
Number of connections made by visitor	$numcons	KFSENSOR_NUMCONS	5

Example Perl Quote of the Day script provided with KFS:

```
# !/usr/bin/perl
# Honeypot PERL script to emulate the Microsoft "Quote of the day" service
# Released under the BSD license.
# Copyright (c) 2003, KeyFocus Ltd., All rights reserved.
#
@quotes = (
"\"We want a few mad people now. See where the sane ones have landed us!\"\n George
Bernard Shaw (1856-1950)\n",
```

** quotes 2-10 removed for brevity **

```
"\"When a stupid man is doing something he is ashamed of, he always declares\n that it is
his duty.\" George Bernard Shaw (1856-1950)\n"
);
## Windows only has 11 quotes to choose from so pick one from the array at random
srand;
$quote = rand 11;
$quote = int $quote;
print STDOUT $quotes[$quote];
```

MS NBT file transfer

The honeypot is capable of receiving and sending data via the MS NBT simulators and all data is "contained" within 2 directories, these by default are: **`C:\kfsensor\nbtuploads`** for data sent to the honeypot and **`C:\Program Files\KeyFocus\KFSensor\files\nbtdownloads`** for data sent from the honeypot.

How the download works:
If an attacker requests a file to be opened for reading then KFS looks in the Download Path to see if the requested file is exists in that directory, a request made using a sub-directory path is always ignored. If the file is found then it is transferred to the attacker using subsequent read requests.

Two example system files are placed in the default download path as part of the KFS installation. The WIN.INI and SYSTEM.INI files emulate a typical Windows 98 machine and are common points of attack for a worm.

How the upload works
If a request is made for a file to be opened for writing then the honeypot will accept and "write" data to the file in memory (so that the transfer completes). When the file is closed KFS will then create a unique filename in the Upload Path using the following name format:

<attackers ip address><attackers port><requested file name>.bin

Examples:
80_34_127_201_1674_natal_scr.bin
61_221_119_175_1431_scrsvr_exe.bin

All non-standard characters and periods in the file name are converted to underscores and the files are always given a **.bin** file extension instead of the requested file name extension.

Events

All data sent to and from the honeypot is recorded in the central honeypot log C:\kfsenslog_yyyymmdd with all events being written in .XML format by default (this can be changed to other types if required), an example entry follows:

```
<event sensorid="kfsensor" id="9037" type="Connection" action="SimStdServer" name="SMTP"
simname="SMTP" protocol="TCP" severity="High">
  <start>2003-12-02 21:39:03:862</start>
  <end>2003-12-02 21:39:56:528</end>
  <client domain="anydomain.com" ip="127.0.0.1" port="1034" />
  <host ip="127.0.0.1" bindip="" port="25" />
  <connection closedby="Server" />
  <recBytes>61</recBytes>
  <received size="61" coding="kf">
    <![CDATA[helo someone.com%0D%0A
rcpt%0D%0A
abcdefghijklm%0D%0A
123456789%0D%0A
]]>
```

the event log format is as follows:

.XML field def	Value	Description
Sensorid	Kfsensor	Default sensor name
Id	9037	Unique event ID
Type	Connection	TCP Connection active
Action	SimStdServer	Sim server used

Name	SMTP	User defined name
Simname	SMTP	Configured (in-built) name
Protocol	TCP	TCP / UDP
Severity	High	Event severity level
Start	2003-12-02 21:39:03:862	Start time
End	2003-12-02 21:39:56:528	End Time
Client domain	anydomain.com	Visitors domain
IP	127.0.0.1	Visitor IP address
Port	1034	Visitor source IP address
Host ip	127.0.0.1	Honeypot IP
Bindip		Used with multiple interfaces
Connection closed by	Server	Server (e.g. timeout) or visitor closed session
recBytes	50	Bytes received
Received size	50	Hybrid Bytes received
Coding	KF	Type of coding, hex, text, Hybrid
CDATA	helo someone.com%0D%0A rcpt%0D%0A abcdefghijklm%0D%0A 123456789%0D%0A	Received data

Custom reports can be defined and the log can be filtered to show just those from a certain port, protocol or source IP address.

Any files transferred to the honeypot (e.g. via SMB) also have an MD5 checksum created and this is automatically recorded in the event log.

Event details: All network traffic that is designated as a single connection is logged under a single event. As well as recording items such as the start and end time of an attack, the visitor's IP and port addresses, and all data transferred both to and from the honeypot is recorded. Events can also be assigned different colour coded severity levels (Low/Med/High) to allow user identification of particular events of interest.

Configurable display columns: Allows a selection from 30 column types e.g. Sensor IP:Port / Received Bytes / Sensor Port.

View by port: The Explorer type interface includes a port tree structure that colour codes the ports depending on how recently they have been attacked. Selecting a port automatically filters the events to show only those targeted at that particular port.

View by visitor: The port view can be exchanged to an IP address tree. This allows the events to be filtered to just show those events from a particular IP address and can be forward or reverse sorted as required.

Figure. Event details window:

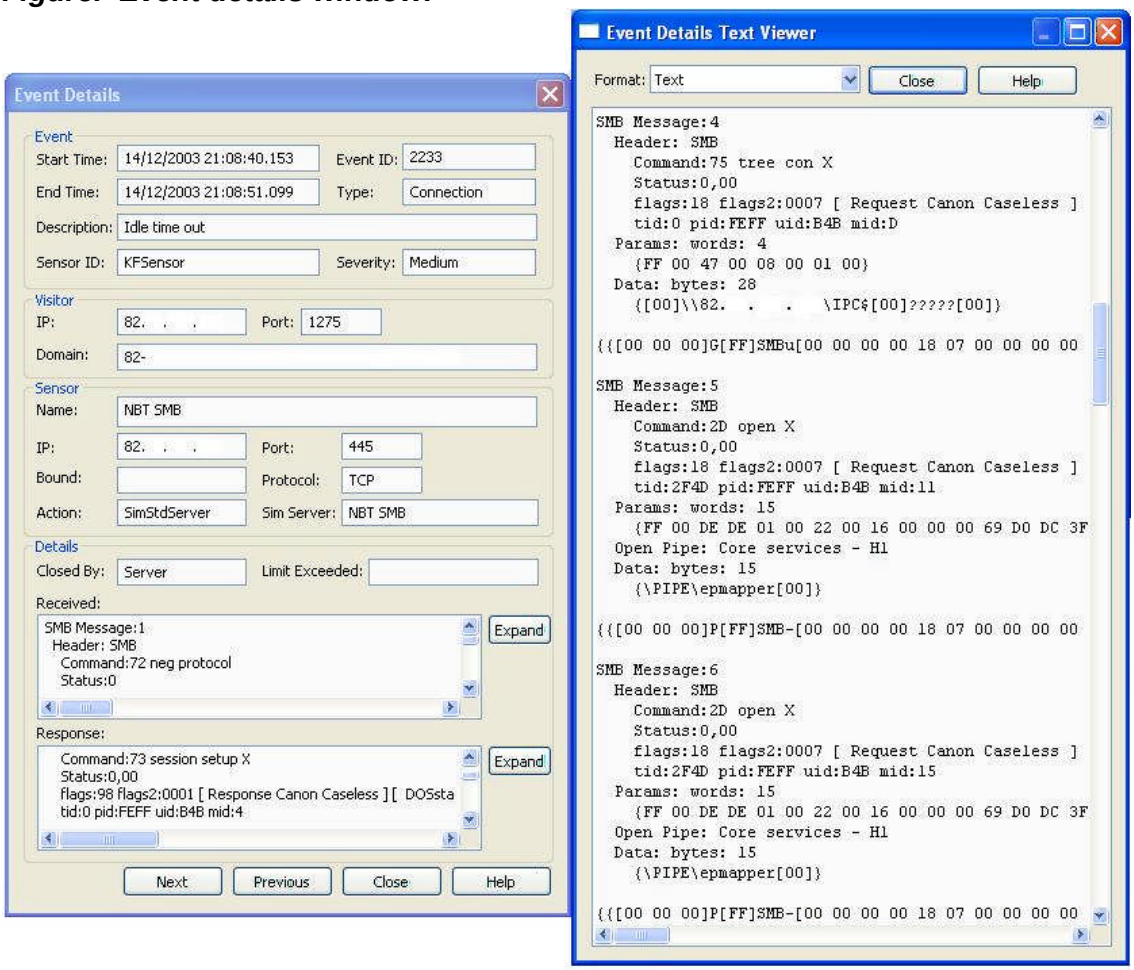


Figure 5. Event details window showing an NBT attack, this is shown by a double-click on the event of interest.

Alerts

KFS provides a range of alert options:

Visual Alert: Displays an alarm icon in the system tray

Audio alerts: Customisable alert sound generated when an event occurs.

Email alerts: Two different formats are available for email alerting. The short format provides minimal information on an event and is suitable for sending to a portable device. The long format provides much more detailed information and is suitable for a normal email client. Long format example:

```
Sent: Tuesday, December 05, 2003 5:24 PM
```

```
Subject: KFSensor (Educational Version):id:1228, visitor:127.0.0.1:3167, Severity: High

 KFSensor Event id: 1228
 ======================
 Start:09/12/2003 17:24:15.354
 End:  09/12/2003 17:24:18.508

 Type: Connection
 Severity: High
 Protocol: TCP

 Host: 127.0.0.1:23
 Visitor: localhost, 127.0.0.1:3167

 Telnet
 Action: SimStdServer
 Sim Server: Telnet
 Connection closed by Visitor

 Received: 534 bytes
 ------------------------------------------------------------
  {IAC DO Echo}{IAC DO SuppressGoAhead}{IAC WILL NewEnvironmentOption}{IAC WILL
  NegotiateAboutWindowSize}{IAC SB NegotiateAboutWindowSize 00 50 00 19}....
 ------------------------------------------------------------

 Response: 571 bytes
 ------------------------------------------------------------
   {IAC WILL Echo}{IAC WILL SuppressGoAhead}{IAC DO NewEnvironmentOption}{IAC DO
  NegotiateAboutWindowSize}{IAC DO BinaryTransmission}{IAC WILL BinaryTransmission}
 {IAC DO Echo}{IAC WILL Echo}Welcome to Microsoft Telnet Service

 login:
 {IAC DO SuppressGoAhead}
 {IAC WILL SuppressGoAhead}
 {IAC WILL NewEnvironmentOption}....
```

SysLog alerts: Alerts can be sent to a separate UNIX SysLog server via the normal 514/udp port e.g:

```
Jul 7 13:34:35 192.168.2.9 kfsensor id: 11510, sensor: TCP
127.0.0.1:110, visitor: localhost, 127.0.0.1:4484, recbytes: 25
```

Event log alerts: KFS can send alerts to the local machine's Event Log, enabling it to be detected by third party event monitoring software; parameters to trim event data are available to ensure that the system log does not become too large.

External alerts: KFS also provides the ability to invoke an external application (.exe / Perl or Python script) to handle an alert event. This flexible feature can have many different uses such as:

1. Creating your own custom event log file
2. Invoking an automated response to the probe or attack
3. Automated whois lookup (via user scripting)
4. Automated Dshield / MyNetwatchman lookup (via user scripting)

Other Features

Denial Of Service (DOS) attack protection: To protect the honeypot from an excessive amount of attacker generated events it incorporates a number of controls to limit the amount of traffic it will accept. Global limits for the honeypot as well as individual connection rate limits can be configured via the DOS settings window (Figure 6), individual rates can be varied for TCP and UDP allowing for example TCP connections to be accepted even if the UDP threshold has been exceeded (e.g. spoofed UDP source connections).

Figure 6. DOS Attack Settings window:

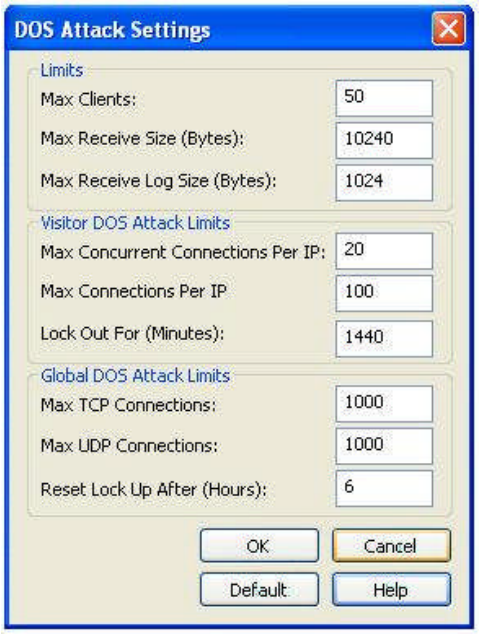


Figure 6 shows the DOS Attack Settings window that is used to prevent attackers from overwhelming the honeypot with events and log data.

Scenario rules: The honeypot can react differently depending on the IP address of an attacker by using “scenario rules” to override the normal honeypot behaviour e.g. specific rules can be defined which cause the server to ignore requests from a certain IP source or network range or to automatically change the severity level of an alert.

Database integration: KFSensor can optionally store events into an ODBC SQL based database. As well as improving the system's performance, it also has the advantage that you can create your own custom reports using any database tool.

Export logs in multiple formats: Events can be exported in XML, HTML, Tab Separated and Comma Separated CSV type formats.

Systems service: KFS can be run as a systems service, allowing it to start before a user has logged on.

Secure configuration: KFS does not need Admin or root privileges to operate. By taking advantage of Window's native security mechanisms the host machine can be effectively secured against compromise.

High integrity version: The high-risk honeypot features are removed in this version. This makes it suitable for use in security sensitive areas of an organisation.

Unfortunately I was limited on the amount of testing that could be completed before running the honeypot in the home environment but managed to run “httprint” (HTTP web fingerprinting tool) against the WWW emulation service to determine it’s capability (top 6 matches shown):

```
C:\httprint\httprint_200\win32>httprint -h 192.168.0.1 -s signatures.txt -P0
httprint v0.200 (beta) - web server fingerprinting tool
(c) 2003, net-square solutions pvt. ltd. - see readme.txt
http://net-square.com/httprint/
httprint@net-square.com
---------------------------------------------------
Finger Printing on http://192.168.0.1:80/
Derived Signature: Microsoft-IIS/6.0
Banner Reported:   Microsoft-IIS/6.0
Banner Deduced:    Microsoft-IIS/6.0
                           Scores:      Confidence:
Microsoft-IIS/6.0:         141          84.94%
Netscape-Enterprise/6.0:   93           18.54%
Netscape-Enterprise/4.1:   88           14.88%
Microsoft-IIS/5.0:         88           14.88%
Microsoft-IIS/5.1:         83           11.71%
Apache/1.3.26:             83           11.71%
```

Conclusions

There is no doubt that a honeypot can be another useful asset in the fight against hacking and malicious activity (either Internet or Internally based), the systems they are designed to emulate can employ a wide range of sophistication from simple port open responses to a fully “emulated” service where large amounts of data can be transferred (as in the case of a running web server). Against the varying levels of sophistication though must be balanced the resources and expenditure that is required to keep the honeypot functioning effectively, there is no point in purchasing a system, setting it up and then leaving it un-monitored for it’s remaining days. The KFS honeypot is an ideal candidate for low-interaction environments where fast and relatively easy deployment is required and at the same time having the flexibility to cope with varying demands, the system I used for 3 weeks ran flawlessly and provided me with an immense insight into the Internet activities of hacker, crackers and script-kiddies all across the globe. The event alert and logging systems ensured that nothing was missed and I was amazed that the total virus (or malicious) file upload count to the honeypot had reached nearly 530 by the end of the evaluation - averaging just over 25 per day – a vast amount for a home connected system. The KFS honeypot in my opinion provides a very fast set-up / fully configured system and starts to provide instant and substantial volumes of data on all types of attack as soon as it is connected to either the Internet or Internal network.

References

1) http://www.honeypots.net/
2) http://www.honeyd.org/
3) http://www.keyfocus.net/kfsensor/help/index.php
4) http://freshmeat.net/projects/thp/?topic_id=43
5) http://project.honeynet.org/misc/project.html
6) http://www.net-square.com

Assignment #2: Network Detects

Network Detect # 1 of 3 – *Multiple connections to Proxy ports 8080, 3128 and 1080*

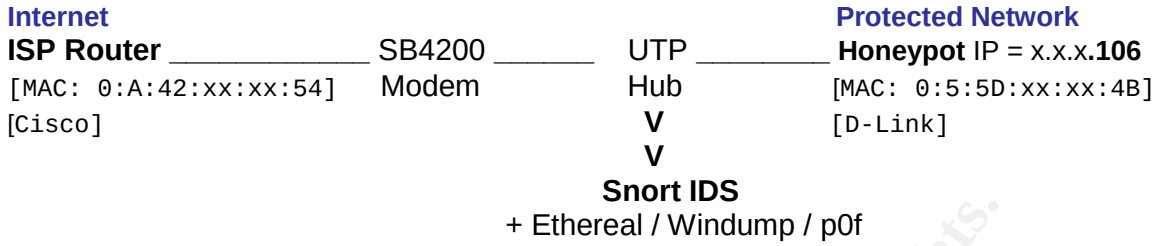
Snort IDS detect summary:

```
12/25-07:46:23.444971  [**] [1:620:3] SCAN Proxy (8080) attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:2765 -> x.x.x.106:8080
12/25-07:46:36.249572  [**] [1:620:3] SCAN Proxy (8080) attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:47602 -> x.x.x.106:8080
12/25-07:46:40.921233  [**] [1:620:3] SCAN Proxy (8080) attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:47603 -> x.x.x.106:8080
12/25-07:46:40.972558  [**] [1:620:3] SCAN Proxy (8080) attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:47604 -> x.x.x.106:8080

12/25-07:49:49.461472  [**] [1:618:4] SCAN Squid Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:2765 -> x.x.x.106:3128
12/25-07:49:57.267173  [**] [1:618:4] SCAN Squid Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:52872 -> x.x.x.106:3128
12/25-07:49:57.275090  [**] [1:618:4] SCAN Squid Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:52873 -> x.x.x.106:3128
12/25-07:49:57.285405  [**] [1:618:4] SCAN Squid Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:52874 -> x.x.x.106:3128

12/25-08:03:43.249061  [**] [1:615:4] SCAN SOCKS Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:2765 -> x.x.x.106:1080
12/25-08:03:55.286595  [**] [1:615:4] SCAN SOCKS Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:46210 -> x.x.x.106:1080
12/25-08:03:56.413483  [**] [1:615:4] SCAN SOCKS Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:46211 -> x.x.x.106:1080
12/25-08:04:03.224243  [**] [1:615:4] SCAN SOCKS Proxy attempt [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 168.226.149.217:46206 -> x.x.x.106:1080
```

Source of Trace:

This detect was captured on my home honeypot which has a single Class C address. The honeypot consisted of the following components: Broadband Internet connection to an SB4200 cable modem, this was then connected to a 10Mbps UTP unmanaged hub. Two connections were then made from the hub - 1 connection to the honeypot system itself (Windows XP based and also running ICF – Internet Connection Firewall) and 1 connection to a separate stand-alone “analyser system” that I used for monitoring the honeypot. The analysis system LAN cable was unidirectional (created from the www.snort.org FAQ, see Appendix A) and only allowed data to be received from the hub, this was to ensure that the analyser system itself could in no way become compromised. The LAN interface on the analyser system was also placed into stealth mode via OS registry modifications (see

Appendix B). Note: the full honeypot IP address has been obfuscated from all of the log entries and packet trace outputs that follow.



Detect was generated by:

Windows XP system running the **EagleX IDS** Snort package downloaded from www.enagesecurity.com. This suite of programs comprised of: Snort 2.0.1-ODBC-MySQL-WIN32 (Build 88**), IDSCentre v1.1 RC4 (GUI front-end for Snort), Apache v1.2.28, MySQL v3.23.55, ACID v0.9.6b23, JPGraph v1.9.1 and WinPCAP v3.0 (Politecnico di Torino). The analyser system was running Windows XP with the latest Service Packs and patches applied. The Snort binary environment and ruleset were also updated to a later version that was available at the time of this analysis - Snort 2.0.2-ODBC-MySQL-WIN32 (Build 92)** downloaded from www.Snort.org.

Other versions of tools used to aid analysis that were running on the analyser system were: **ethereal** v0.9.16, **windump** (v current-cvs.tcpdump.org) and **p0f** v2.0.4-beta1, windump was mainly used to provide a constant full packet capture into daily analysis files (e.g. 081103.eth, 091103.eth) to ensure all data could be fully examined following IDS alerts. Windump command line example: windump -n -i 1 –s 0 -w 121103.eth.

The standard Snort ruleset was implemented with all rules and logging enabled to the central alert.ids file and log directories. The summary output at the beginning of this particular detect was produced with the following command line:

C:\EagleX\Snort\bin\Snort -r 251203.eth -c ..\etc\Snort.conf -l ..\logs -qk none –A console src host 168.226.149.217

After analysing the alerts file I decided to investigate the “proxy” scans further and the following data was extracted from the Snort 168.226.149.217 log sub-directory:

```
[**] [1:620:3] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:46:23.444966 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:2765 -> x.x.x.106:8080 TCP TTL:41 TOS:0x0 ID:63777 IpLen:20 DgmLen:60 DF
******S* Seq: 0x18553BCC  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47170285 0 NOP WS: 0

[**] [1:620:3] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:46:36.249562 0:A:42:6B:F4:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:47602 -> x.x.x.106:8080 TCP TTL:41 TOS:0x0 ID:64266 IpLen:20 DgmLen:60 DF
******S* Seq: 0x188937E4  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47172846 0 NOP WS: 0
```

```
[**] [1:620:3] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:46:40.921228 0:A:42:6B:F4:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:47603 -> x.x.x.106:8080 TCP TTL:41 TOS:0x0 ID:51016 IpLen:20 DgmLen:60 DF
******S* Seq: 0x19BD17DD  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47173781 0 NOP WS: 0

[**] [1:620:3] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:46:40.972549 0:A:42:6B:F4:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:47604 -> x.x.x.106:8080 TCP TTL:41 TOS:0x0 ID:38216 IpLen:20 DgmLen:60 DF
******S* Seq: 0x197F45D7  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47173791 0 NOP WS: 0

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

[**] [1:618:4] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:49:49.461467 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:2765 -> x.x.x.106:3128 TCP TTL:41 TOS:0x0 ID:33029 IpLen:20 DgmLen:60 DF
******S* Seq: 0x243EE2ED  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47211496 0 NOP WS: 0

[**] [1:618:4] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:49:57.267168 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:52872 -> x.x.x.106:3128 TCP TTL:41 TOS:0x0 ID:5038 IpLen:20 DgmLen:60 DF
******S* Seq: 0x256779AF  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47213057 0 NOP WS: 0

[**] [1:618:4] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:49:57.275085 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:52873 -> x.x.x.106:3128 TCP TTL:41 TOS:0x0 ID:19767 IpLen:20 DgmLen:60 DF
******S* Seq: 0x25142734  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47213059 0 NOP WS: 0

[**] [1:618:4] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-07:49:57.285400 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:52874 -> x.x.x.106:3128 TCP TTL:41 TOS:0x0 ID:23536 IpLen:20 DgmLen:60 DF
******S* Seq: 0x248A31F5  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47213061 0 NOP WS: 0

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

[**] [1:615:4] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/27-08:03:43.249051 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:2765 -> x.x.x.106:1080 TCP TTL:41 TOS:0x0 ID:58482 IpLen:20 DgmLen:60 DF
******S* Seq: 0x593B0B1C  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47378286 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]

[**] [1:615:4] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/25-08:03:55.286587 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:46210 -> x.x.x.106:1080 TCP TTL:41 TOS:0x0 ID:17827 IpLen:20 DgmLen:60 DF
******S* Seq: 0x5A36BD76  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47380693 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]

[**] [1:615:4] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/25-08:03:56.413473 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
```

```
168.226.149.217:46211 -> x.x.x.106:1080 TCP TTL:41 TOS:0x0 ID:10528 IpLen:20 DgmLen:60 DF
******S* Seq: 0x59C1D097  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47380919 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]

[**] [1:615:4] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/25-08:04:03.224237 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x4A
168.226.149.217:46206 -> x.x.x.106:1080 TCP TTL:41 TOS:0x0 ID:29476 IpLen:20 DgmLen:60 DF
******S* Seq: 0x5A7D8C2D  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 47382281 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]
```

The above 12 alerts originated from a single IP address and analysis indicated that the same attack was being applied across 3 separate destination ports which I therefore grouped into a general “open proxy scan” attack, the corresponding Snort rules that generated these alerts were:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy \(8080\)
attempt"; flags:S,12; classtype:attempted-recon; sid:620; rev:3;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid Proxy
attempt"; flags:S,12; classtype:attempted-recon; sid:618; rev:4;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN SOCKS Proxy
attempt"; flags:S,12; reference:url,help.undernet.org/proxyscan/;
classtype:attempted-recon; sid:615; rev:4;)
```

The alerts triggered when Snort detected a SYN connection from any source IP address / source port to TCP target port 8080 (Proxy), 3128 (Squid) or 1080 (SOCKS) respectively. In addition the “flags:” field indicates S for match on SYN bit set and “12” to ignore reserved high-order bits 1 and 2 which are the ECN (Explicit Congestion Notification) and CWR (Congestion Window Reduced) bits which are used for congestion control (RFC3168). The ECN/CWR bits are available (can be enabled) under Linux distro’s and certain network devices. Disabling the 3 individual scan rules to allow the Snort engine to detect any other attacks from this particular IP address did not produce any additional alerts or data.

Probability the source address was spoofed:

Highly unlikely as the source is trying to detect open public proxy servers requiring a full TCP connection response, it is also likely that a valid proxy response is required before the source identifies the destination address (in this case the honeypot) as a valid “open” proxy server i.e. not just a normal SYN/ACK response to an open port scan. Also, analysis of the ID, Sequence numbering and Acknowledgement fields do not indicate any attempt at packets being manipulated.

Description of attack:

This attack begins as a normal recon / port scan and then progresses to multiple connect attempts to TCP ports 8080, 3128 and 1080 the purpose of which is to locate the presence of publicly accessible proxy servers on the Internet. The Snort database classifies these alerts as attempted-recons and a link for the SOCKS alert provides further details at

http://help.undernet.org/proxyscan relating to their Undernet IRC servers and validation of users.

A proxy server can be used (for example) to forward HTTP requests on behalf of a client browser e.g. Opera, Internet Explorer, Mozilla etc., the proxy server therefore protects or "hides" internal networks as the only source address seen on the Internet is the IP address of the proxy server itself. Proxy servers should be configured to allow internal network connections only, however, there are many Proxy's that are mis-configured and allow an external Internet connection to be made via the Proxy. The use of the open (listening) ports on the honeypot allowed this attack to progress slightly further as the initial probe detected open ports and the attack then continued further by sending full proxy connection requests.

Attack mechanism:

The attack began at 07:46:24 and lasted until 08:04:19, a total of 114 TCP packets were transmitted to and from the honeypot and the process of the attack is broken down as follows:

1. Source connects from port 2765 to destination port 8080.
2. Source then send requests to the honeypot on port 8080 and attempts to request outbound connections from the honeypot to 200.16.144.250 using the following connect methods: HTTP CONNECT, SOCKS v4 and SOCKS v5 (the honeypot does not allow the outbound connections – it just records the requests that have been sent).
3.The source then repeats the above steps using target ports 3128,1080 and uses the same initial scan source port of 2765.

Output from windump showing a request to 200.16.144.250 using the HTTP CONNECT method:

```
07:46:36.571856 IP (tos 0x0, ttl 41, id 64268, len 113) 168.226.149.217.47602 >
x.x.x.106.8080: P [tcp sum ok] 1:62(61) ack 1 win 5808 <nop,nop,timestamp 47172911 0>(DF)
0x0000   4500 0071 fb0c 4000 2906 e09e a8e2 95d9        E..q..@.).......
0x0010   xxxx xxxx b9f2 1f90 1889 37e5 a3c4 5841        R%.j......7...XA
0x0020   8018 16b0 8358 0000 0101 080a 02cf cd2f        .....X........./
0x0030   0000 0000 434f 4e4e 4543 5420 3230 302e        ....CONNECT.200.
0x0040   3136 2e31 3434 2e32 3530 3a32 3520 4854        16.144.250:25.HT
0x0050   5450 2f31 2e30 0d0a 5573 6572 2d41 6765        TP/1.0..User-Age
0x0060   6e74 3a20 7961 7068 2d30 2e39 310d 0a0d        nt:.yaph-0.91...
0x0070   0a                                            .
```

Output from windump showing a request to 200.16.144.250 using the Socks v4 connect method:

```
07:46:41.388247 IP (tos 0x0, ttl 41, id 51018, len 61) 168.226.149.217.47603 >
x.x.x.106.8080: P [tcp sum ok] 1:10(9) ack 1 win 5808 <nop,nop,timestamp 47173874 0> (DF)
0x0000   4500 003d c74a 4000 2906 2125 a8e2 95d9        E..=.J@.).!%....
0x0010   xxxx xxxx b9f3 1f90 19bd 17de a3d7 1742        R%.j...........B
0x0020   8018 16b0 ff90 0000 0101 080a 02cf d0f2        ................
0x0030   0000 0000 0401 0019 c810 90fa 00               .............
```

Output from windump showing a request to 200.16.144.250 using the Socks v5 connect method:

```
07:46:41.414273 IP (tos 0x0, ttl 41, id 38218, len 55) 168.226.149.217.47604 >
x.x.x.106.8080: P [tcp sum ok] 1:4(3) ack 1 win 5808 <nop,nop,timestamp 47173879 0> (DF)
```

```
0x0000   4500 0037 954a 4000 2906 532b a8e2 95d9        E..7.J@.).S+....
0x0010   xxxx xxxx b9f4 1f90 197f 45d8 a3d8 6af5        R%.j......E...j.
0x0020   8018 16b0 d644 0000 0101 080a 02cf d0f7        .....D..........
0x0030   0000 0000 0501 00                              .......
```

The connection requests above are then repeated on ports 3128 and 1080.

The passive finger-printing tool p0f also identifies the attacking system as a Linux 2.4/2.6 system at an estimated distance of 23 hops from the honeypot:

```
C:\EagleX\Snort\bin>p0f -s 251203.eth -l
p0f - passive os fingerprinting utility, version 2.0.4-beta1
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
WIN32 port (C) M. Davis <mike@datanerds.net>, K. Kuehl <kkuehl@cisco.com>
p0f: listening (SYN) on '251203.eth', 207 sigs (12 generic), rule: 'all'.

168.226.149.217:2765 - Linux 2.4/2.6 (up: 131 hrs) -> x.x.x.106:8080 (distance 23, link:
pppoe (DSL))
```

11 further p0f entries removed for brevity – identical output

```
[+] End of input file.
```

The HTTP CONNECT trace has interesting data in that it details in the User-Agent: field "yaph-0.91", a google for this provides numerous references mainly to sourceforge.net indicating that the attacker is likely to be using an automated program called **YAPH - Yet Another Proxy Hunter.** This is an Open Source program that utilises the capabilities of nmap and ProxyChains to detect and validate open proxy servers (ProxyChains allows TCP tunnelling through proxies and to allow a user-defined list of proxies to be chained together for further obfuscation).

Extracts from YAPH readme downloaded from http://prdownloads.sourceforge.net/yaph/yaph-0.91.tar.gz?download:

```
=================================================================
YAPH - Yet Another Proxy Hunter ver 0.91 README

To get full power of yaph you must have the following programs:
Nmap                    http://www.insecure.org/nmap
ProxyChains             http://proxychains.sourceforge.net

General.
Yaph provides ability to reveal public proxy servers. It can search & validate socks v4,
socks v5 and http (connect method) proxy servers. HTTP proxy servers are checked for
CONNECT method only. Validated proxy server is public proxy that can be used for tcp
tunneling. While using tcp tunneling your IP address stays private. The best tools
(proxifiers) for tcp tunneling via proxies are:

ProxyChains (unix)      http://proxychains.sourceforge.net
ProxyCap (win)          http://www.proxylabs.com

List checking.
Yaph knows to check list of proxy servers to reveal working proxies.
Supported input formats are:

Proxy Chains format  ( as it appears in proxychains.conf) like:
http            192.168.1.2         8080
socks4          192.168.1.3         1080
socks5          192.168.1.4         1080

Network scan.
Yaph knows to use Nmap to find new proxy servers.
```

```
Some examples for stealth scans:

proxychains yaph -D10.1.1.2,10.2.5.1,10.1.3.2,10.2.5.4 10.0.0.*

In this example, nmap will scan hosts 10.0.0.1 to 10.0.0.255 looking for open ports
1080,8080,3128 with decoys 10.1.1.2 10.2.5.1 10.1.3.2 10.2.5.4 and then validation of
open ports will be performed via proxy server defined in proxychains.conf

proxychains yaph -sT -P0 10.0.0.*

In this example EVERYTHING will go via proxyserver. The nmap scan will go through proxy
server!!! Validation of open ports as well scanned hosts are 10.0.0.1 - 10.0.0.255 (this
is slow, but your IP is never appears in the logs of scanned hosts)
=====================================================================
```

As can be seen from the text above the additional use of ProxyChains can also be used further to increase the difficulty level in tracing back to the true source of the attacker by using a “chain” of public proxy servers – it is therefore likely that the true attacker for this particular event is not at the IP address recorded in the Snort alerts. The return address for the proxy validation request also indicates a dial-up account.

Correlations:

www.mynetwatchman.com reports for the IP source address:

```
Incident Detail
Incident ID:  67700486
Source IP:  168.226.149.217
Provider Domain:  speedy.com.ar
DNS Name:  168-226-149-217.speedy.com.ar

Total Event Count :  1276
Total Distinct Agent:143/162900
Response :           No Response
Status Description:  Escalated
Exclusion Reason :   None

Network Name/NextNIC Start IP - End IP
LACNIC-ERX-168-226-0-0/DUMMY 168.226.0.0 - 168.226.255.255

Latin American and Caribbean IP address Regional Registry
OrgID:      LACNIC
Address:    Potosi 1517
City:       Montevideo
PostalCode: 11500
Country:    UY
ReferralServer: whois://whois.lacnic.net

whois.lacnic.net query output for 168.226.149.217:

inetnum:     168.226/16
status:      assigned
owner:       Telefonica de Argentina
ownerid:     AR-TEAR7-LACNIC
responsible: Marcelo A. Muñoz
address:     Defensa, 390, Piso 5
address:     1065 - Buenos Aires - CF
country:     AR
```

```
phone:          +54 11 4-3335509 []
```

myNetWatchman Total Event Count : 1276

Most Recent Event											
Date/Time (UTC)	Agent Alias	Agent Type	Log Type	Target IP	# of IPs Targeted	IP Protocol	Target Port	Port/ Issue Description	Source Port	Explanation	Event Count
25 Dec 2003 08:14:59	jankemi	Perl	Cisco PIX	199.17.x.x	167	6	1080	SOCKS Proxy	2765	mNW Info	170
25 Dec 2003 08:14:21	jankemi	Perl	Cisco PIX	199.17.x.x	172	6	3128	RingZero Probe	2765	mNW Info	173
25 Dec 2003 08:12:38	jankemi	Perl	Cisco PIX	199.17.x.x	169	6	8080	RingZero RingZero	2765	mNW Info	173

A further 860 event counts have been removed for brevity – events go back to 24 Dec 2003 15:32:55.

www.dshield.org reports for the IP source address:

```
IP Address: 168.226.149.217
HostName:   168-226-149-217.speedy.com.ar

DShield Profile:
Country:                        AR
Contact E-mail:                 tasamail@telefonica.com.ar
AS Number:                      0
Total Records against IP:       5603
Number of targets:              1335
Date Range:                     2003-12-23 to 2003-12-28

Top 10 Ports hit by this source:

Port      Attacks      Start       End
1080      1996         2003-12-23  2003-12-26
8080      1824         2003-12-23  2003-12-26
3128      1782         2003-12-23  2003-12-26
137       1            2003-12-28  2003-12-28

Last Fightback Sent:  sent to tasamail@telefonica.com.ar on 2003-12-24 10:41:47
```

www.dshield.org reports for the IP CONNECT target address (which appears to be a dial-up connection):

```
IP Address: 200.16.144.250
HostName: ppp58-r5300-cap2.via-net-works.net.ar
DShield Profile: Country: AR
Contact E-mail: noc@telintar.net.ar
AS Number: 0
Total Records against IP:  1
Number of targets:  1
Date Range: 2004-01-03 to 2004-01-03
Summary was recently updated.

Top 10 Ports hit by this source:
Port      Attacks      Start       End
11616     1            2004-01-03  2004-01-03
```

```
Last Fightback Sent:  not sent

Whois:

inetnum:     200.16.144/22
status:      reassigned
owner:       S&M International S.A.
ownerid:     AR-SISA3-LACNIC
address:     Av Roque Saenz Pe#a 971 4 Piso
address:     Buenos Aires, BA 1035
country:     AR
owner-c:     MC90-ARIN
inetrev:     200.16.144/22
nserver:     NS1.SMINTER.COM.AR
```

KFS honeypot log entries shown below for the port 8080 connect:

```
<event sensorid="KFSensor" id="9783" type="Connection" action="ReadAndClose" name="proxy"
protocol="TCP" severity="Medium">
  <start>2003-12-25 07:46:37:865</start>
  <end>2003-12-25 07:46:57:093</end>
  <client domain="168-226-149-217.speedy.com.ar" ip="168.226.149.217" port="47602" />
  <host ip="x.x.x.106" bindip="" port="8080" />
  <connection closedby="Client" />
  <recBytes>61</recBytes>
  <received size="61" coding="kf">
    <![CDATA[CONNECT 200.16.144.250:25 HTTP/1.0%0D%0A
User-Agent: yaph-0.91%0D%0A
%0D%0A
]]>
  </received>
</event>

<event sensorid="KFSensor" id="9784" type="Connection" action="ReadAndClose" name="proxy"
protocol="TCP" severity="Medium">
  <start>2003-12-25 07:46:42:522</start>
  <end>2003-12-25 07:46:57:093</end>
  <client domain="168-226-149-217.speedy.com.ar" ip="168.226.149.217" port="47603" />
  <host ip="x.x.x.106" bindip="" port="8080" />
  <connection closedby="Client" />
  <recBytes>9</recBytes>
  <received size="9" coding="kf">
    <![CDATA[%04%01%00%19%C8%10%90%FA%00]]>
  </received>
</event>

<event sensorid="KFSensor" id="9785" type="Connection" action="ReadAndClose" name="proxy"
protocol="TCP" severity="Medium">
  <start>2003-12-25 07:46:42:562</start>
  <end>2003-12-25 07:46:57:093</end>
  <client domain="168-226-149-217.speedy.com.ar" ip="168.226.149.217" port="47604" />
  <host ip="x.x.x.106" bindip="" port="8080" />
  <connection closedby="Client" />
  <recBytes>3</recBytes>
  <received size="3" coding="kf">
    <![CDATA[%05%01%00]]>
  </received>
</event>
```

References:

1) http://yaph.sourceforge.net/

2) http://www.multiproxy.org/
3) http://www.squid-cache.org/Doc/FAQ/FAQ.html
4) http://www.socks.permeo.com/
5) HTTP Tunnels Through Proxies, Daniel Alman July 30 2003, SANS GSEC
6) http://www.kb.cert.org/vuls/id/868219
7) http://www.kb.cert.org/vuls/id/150227

Evidence of active targeting:

The honeypot appears to have been targeted as I could not locate any previous scanning or network mapping activities via the windump or ICF (Internet Connection Firewall) logs.

Severity:

The following ratings are marked against scale: **1** (lowest) to **5** (highest) and for the purpose of the Severity Rating I have assumed that the honeypot would be providing a Proxy service for it's internal networks and outbound Internet access requirements for internal users.

Criticality = 4 (measure of target(s) value / critical nature)
Proxy servers provide critical systems access, loss of this system would potentially cause a range of effects from total site access loss to inconvenience loss for users of a particular service.

Lethality = 1 (if the attack(s) succeeded damage estimate)
If this was a scan for open proxies and the system had correctly responded then this server would have been added to a list of open proxy servers causing possibly resource exhaustion / degraded service / responses etc.

System Countermeasures =1 (host(s) defence posture)
The honeypot had no additional defences apart from the in-built XP Internet Connection Firewall which was configured to allow the same port access as that being offered by the honeypot (this was to allow log correlation), in addition only ICMP echo request and replies were being allowed to and from the honeypot.

Network Countermeasures = 2 (network (s) defensive systems/configs)
No additional filtering devices or router ACL's were present apart from that configured by my ISP – direct access from Internet, however, the system was being continuously monitored by the Snort IDS.

Severity Rating = 2
(criticality + lethality) – (system countermeasures + network countermeasures)

Defensive recommendation:

Where possible apply router ACL's to prevent Internet access to the proxy ports.

Ensure Firewall's are configured to drop inbound connections from the Internet to proxy servers and allow local network access only – this could be further improved by allowing ranges or an approved list of internal devices allowed to use the proxy services.

Multiple choice test question:

Do the major Socks protocol versions support UDP transfer?

a. In Socks v4 implementations only
b. In Socks v5 implementations only
c. In both versions
d. Neither, UDP via Socks is still under development

Answer : b

Socks v5 supports UDP, a number of more advanced Socks implementations can also support secure bi-directional UDP channels, streaming, multimedia, real-time applications and other complex H.323 applications.

Network Detect # 2 of 3 – NETBIOS SMB SMB_COM_TRANSACTION Max Parameter and Max Count of 0 DOS

Snort IDS detect:

```
[**] [1:2101:4] NETBIOS SMB SMB_COM_TRANSACTION Max Parameter and Max Count of 0 DOS
Attempt [**]
[Classification: Detection of a Denial of Service Attack] [Priority: 2]
12/29-11:12:07.437849 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x88
211.215.52.143:4535->x.x.x.106:139 TCP TTL:110 TOS:0x0 ID:25668 IpLen:20 DgmLen:122 DF
***AP*** Seq: 0x2AB67D60  Ack: 0x7D43DE38  Win: 0x437A  TcpLen: 20
[Xref => http://www.corest.com/common/showdoc.php?idx=262][Xref =>
http://www.microsoft.com/technet/security/bulletin/MS02-045.asp][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0724]
```

Source of Trace:

This source of this detect was identical to detect #1 of 3 and used the same network topology and honeypot set-up. The ttl of the packet indicates it was likely to have an initial ttl of 128 and a window size 0x437A (17274) suggesting that the attacking machine is a Windows 2000 system, the DF flag is also set and TOS is set to 0.

Detect was generated by:

Snort IDS version and set-up as per detect #1 of 3. The command used to extract the data was “snort -r 291203.eth -c ..\etc\Snort.conf -l ..\logs -qk none –A console src host 211.215.52.143”.

Snort signature:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS SMB
SMB_COM_TRANSACTION Max Parameter and Max Count of 0 DOS Attempt";
flow:to_server,established; content:"|00|"; offset:0; depth:1;
content:"|FF|SMB|25|"; offset:4; depth:5; content:"|00 00 00 00|"; offset:43;
depth:4; reference:cve,CAN-2002-0724;
reference:url,www.microsoft.com/technet/security/bulletin/MS02-045.asp;
reference:url,www.corest.com/common/showdoc.php?idx=262; classtype:denial-of-
service; sid:2101; rev:4;)
```

This signature is alerting to a possible Denial Of Service (DOS) attack against the SMB (Server Message Block) Protocol implementation in Windows NT, W2K and XP. A vulnerable system may be open to DOS via a crafted packet containing an enumeration request when both the Max Parameter Count and Max Data Count fields are set to 0.

In addition to the normal destination port and flow “to_server” triggers this rule is also set to match a total of 4 fields and is further explained below:

`1) content:"|00|"; offset:0; depth:1;`
Match content 00 starting at offset 0 within the 1st byte into the payload (payload begins at offset 0x37). This match checks that this transaction is a “Session Message” type.

2) content:"|FF|SMB|25|"; offset:4; depth:5;
Match content FF 53 4D 42 25 starting at offset 4 for a further 5 bytes into the payload. This match checks that the Server component = SMB

3) content:"|00 00 00 00|"; offset:43; depth:4
Match content 00 00 00 00 starting at offset 43 for a further 4 bytes into the payload.

*This match actually contains the 2 16-bit fields for Max Parameter Count and Max Data Count that trigger the buffer overflow.

Snort alert:

```
[**] NETBIOS SMB SMB_COM_TRANSACTION Max Parameter and Max Count of 0 DOS Attempt [**]
12/29-11:12:07.437849 0:A:42:xx:xx:54 -> 0:5:5D:xx:xx:4B type:0x800 len:0x88
211.215.52.143:4535 -> x.x.x.106:139 TCP TTL:110 TOS:0x0 ID:25668 IpLen:20 DgmLen:122 DF
***AP*** Seq: 0x2AB67D60  Ack: 0x7D43DE38  Win: 0x437A  TcpLen: 20
0x0000: 00 05 5D 07 1C 4B 00 0A 42 6B F4 54 08 00 45 00  ..]..K..Bk.T..E.
0x0010: 00 7A 64 44 40 00 6E 06 75 43 D3 D7 34 8F xx xx  .zdD@.n.uC..4.R%
0x0020: xx xx 11 B7 00 8B 2A B6 7D 60 7D 43 DE 38 50 18  .j....*.}`}C.8P.
0x0030: 43 7A 9D 5C 00 00 00 00 00 4E FF 53 4D 42 25 00  Cz.\.....N.SMB%.
0x0040: 00 00 00 18 07 00 00 00 00 00 00 00 00 00 00 00  ................
0x0050: 00 00 BD 1B FF FE D4 E5 10 00 10 02 00 00 00 00  ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 02 00 4C  ..............L
0x0070: 00 00 00 00 00 02 00 01 00 91 5D 0B 00 5C 50 49  ..........]..\PI
0x0080: 50 45 5C 00 00 00 00 01                          PE\.....
```

Probability the source address was spoofed:

Highly unlikely as this requires a normal TCP connection to have been established, if the attacker had wanted to remain anonymous for the DOS or exploit attempt then he/she would have opened a backdoor or remote control mechanism on the 211.215.52.143 system – a lot of work to apply a DOS attack and I would consider it is more likely that this is a false positive by the IDS

Description of attack:

SMB stands for "Server Message Block" and is also known as CIFS (Common Internet File System) This protocol is intended to provide an open cross-platform mechanism for client systems to request file services from server system over a network. Current CIFS implementation under Windows runs over port 139/tcp and/or port 445/tcp (Direct Host), depending whether NetBIOS over TCP/IP is enabled or not.

The Windows Operating System is normally shipped with anonymous access enabled by default and will therefore be vulnerable to a denial of service attack. A successful attack will trigger an operating system halt (Blue Screen).

Attack mechanism:

The DOS attack can be applied via publicly available tools such as SMBdie and requires a full SMB exchange via TCP to have been completed which has occurred on the honeypot, this attack however is a false positive as the windump and Ethereal trace analysis indicate that a SMB C$ share access was requested prior to the alert (these traces have been sent

to www.snort.org for feedback). The honeypot had also been uploaded with a malicious file via it's SMB open share access from this particular attacker address.

Correlations:

Using the online utilities at www.hexillion.com provided the following "Domain Dossier" information for the **211.215.52.143** address:

Domain Dossier Investigate domains and IP addresses

Network Whois record:

```
Querying whois.arin.net with "211.215.52.143"...
Querying whois.apnic.net with "211.215.52.143"...

   % [whois.apnic.net node-1]
   % Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html

   inetnum:      211.212.0.0 - 211.215.255.255
   netname:      HANANET
   descr:        Hanaro Telecom, Inc.
   country:      KR
   admin-c:      IS37-AP
   tech-c:       SH243-AP
   remarks:      ****************************************
   remarks:      Allocated to KRNIC Member.
   remarks:      If you would like to find assignment
   remarks:      information in detail please refer to
   remarks:      the KRNIC Whois Database at:
   remarks:      http://whois.nic.or.kr/english/index.html
   remarks:      ****************************************
   mnt-by:       MNT-KRNIC-AP
   mnt-lower:    MNT-KRNIC-AP
   changed:      hostmaster@apnic.net 20010615
   changed:      hostmaster@apnic.net 20010730
   status:       ALLOCATED PORTABLE
   source:       APNIC

http://whois.nic.or.kr/english/index.html query:

   IP Address       : 211.215.52.0-211.215.52.255
   Network Name     : HANANET-INFRA
   Connect ISP Name : HANANET
   Connect Date     : 20011016
   Registration Date  : 20031105

   [ Organization Information ]
   Orgnization ID     : ORG3930
   Org Name           : Hanaro Telecom Inc.
   State              : KYONGGI
   Address            : 726-1 Janghang 2(i)-dong , Goyang-si Ilsan-gu
   Zip Code           : 411-837
```

There are no reports on www.dshield.org or www.mynetwatchman.com against IP address 211.215.52.143.

References:

http://www.snort.org/snort-db/sid.html?sid=2101
http://www1.corest.com/common/showdoc.php?idx=262
http://www.snort.org/docs/snort_manual/
http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/MS02-045.asp
http://www.kb.cert.org/vuls/id/311619
http://www.kb.cert.org/vuls/id/342243
http://www.kb.cert.org/vuls/id/250635

Evidence of active targeting:

The honeypot attack appears to have been part of a larger shares type scan / connect session as multiple repeat requests were being made to access C$ drive shares.

Severity:

The following ratings are marked against scale: **1** (lowest) to **5** (highest) and for the purpose of the Severity Rating I have assumed that the honeypot was providing a critical unrestricted file / print sharing service for a medium sized department (50-100 users).

Criticality = 3 (measure of target(s) value / critical nature)
The honeypot is providing critical file and print sharing services for the department, I would rate this a 3.

Lethality = 5 (if the attack(s) succeeded damage estimate)
If this DOS attack had succeeded users would have lost all access whilst the system was being re-booted, there would also have been the requirement to locate the source and method of the attack so it is likely that the system would have been down for many hours to allow further investigation.

System Countermeasures = 4 (host(s) defence posture)
I would like to assume that the IT department had been applying updates and patches in a timely manner and was aware of Windows security issues and available SMB exploits.

Network Countermeasures = 3 (network (s) defensive systems/configs)
Assumed that this is an internal system and that no Internet or untrusted LAN access was permitted, router ACL's and internal Firewalls should be denying 139/tcp and 445/tcp type traffic where at all possible.

Severity Rating = 1
(criticality + lethality) – (system countermeasures + network countermeasures)

Defensive recommendation:

Disable all anonymous access created through SMB null sessions

Block access to the SMB ports from all internally un-trusted networks.

Deny all Internet access to ports 139/tcp and 445/tcp at the network perimeter / Firewall

Apply vendors recommended patch updates.

Multiple choice test question:

Samba is used in *NIX environments to provide SMB functionality, can a Samba enabled host participate in a domain as a PDC (Primary Domain Controller)?

a. Yes
b. No – a PDC must be a Microsoft based system
c. No – a Samba host can only be a BDC (Backup Domain Controller)
d. None of the above – Samba and Microsoft implementations are not compatible

Answer : a

Samba implements the NT LM 0.12 protocol. Samba can participate in a domain (both as a PDC and a Member of a domain) and can also participate as a master browser.

Network Detect # 3 of 3 – TCP Zone Transfer attempt (AXFR)

Source of Trace:

This detect was obtained from the http://www.incidents.org/logs/raw listing.

Fileref: **2002.10.1**
Size/Date**:** **16,574,758 Mon Dec 2 15:44:57 2002.**

The file reference is dated 10.1 however Snort and windump output indicates actual date and timestamps for the alerts on 11.1 between 00:00-23:59 hours.

The first priority was to determine the topology of the network and using Ethereal (v0.9.16) to read in the binary file 2 new columns were added to the "default" window to display the source and destination MAC addresses, these were "Hw src addr (resolved)" and "Hw dest (addr resolved)". This enabled me to determine what MAC addresses were in use by using the column sort option to confirm that only 2 unique MAC addresses were present in the trace file – 00:03:e3:d9:26:c0 and 00:00:0c:04:b2:33. The "Hw resolved" options also indicated **Cisco**_d9:26:c0 and **Cisco**_04:b2:33 as the equipment vendor. A double-check against http://standards.ieee.org/regauth/oui/index.shtml confirmed Ethereal's "Cisco" identification.

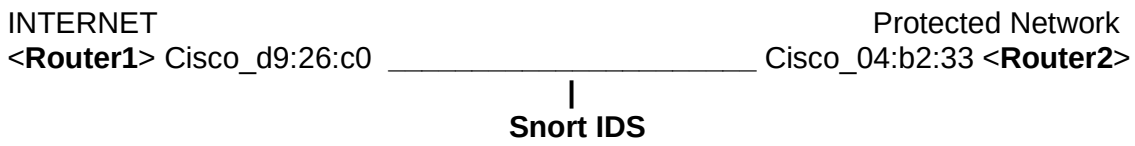
The analysis at this point also indicated that the "protected" network range was the 207.166.0.0/16 network with an address target range of 207.166.0.32 to 207.166.253.205. The Snort data indicated alerts originating from the protected source IP address 207.166.87.157 and a number of packets originating from source IP 255.255.255.255. There were no broadcast packets to 255.255.255.255.

A further capability of Ethereal is it's function to display "protocol hierarchy statistics", this helps to give an indication on the type of traffic being transferred and although the trace file only contains Snort generated alert traffic it is a good way to obtain an overview of protocols in use.

The protocol statistics gave a total of 14,597 packets and total data size of 26.4MB, this was further broken down as:

Packets		Packets	
14,597	**TCP**	**0**	**UDP**
11,808	HTTP (25.6MB)		
2,121	DATA (0.8MB)		
18	MSN Messenger (4.6KB)		
10	NetBIOS / SMB		
7	DNS		
3	Unreassembled Fragments		
2	FTP CONTROL / DATA		
1	X.11		

The previous data indicates that it was highly likely that the Snort sensor was placed between the Internet and the protected network as depicted below:



Detect was generated by:

Windows XP OS running the **EagleX IDS** package downloaded from www.enagesecurity.com. This suite of programs comprised of: Snort 2.0.1-ODBC-MySQL-WIN32 (Build 88**)**, IDSCentre v1.1 RC4 (GUI front-end for Snort), Apache v1.2.28, MySQL v3.23.55, ACID v0.9.6b23, JPGraph v1.9.1 and WinPCAP v3.0 (Politecnico di Torino).

******The Snort binary environment and ruleset were also updated to a later version that was available at the time of this analysis - Snort 2.0.2-ODBC-MySQL-WIN32 (Build 92)** downloaded from www.Snort.org.

The versions of other tools used to aid analysis were: **ethereal** v0.9.16, **windump** (v current-cvs.tcpdump.org) and **p0f** v2.0.4-beta1.

The standard Snort ruleset was used with all rules enabled.

Command used to obtain alert output to the central alert.ids file and log sub-directory:

Snort -r 2002.10.1.eth -c ..\etc\Snort.conf -l ..\logs -qveXk none

Overview of command-line options:

-r read and process the specified tcpdump file (2002.10.1)
-c specify location of the Snort configuration/rules file
-l specify location of the alert log directory
-q start quiet – no Snort banner / summary statistics
-v verbose output
-e display the data-link layer header
-k <mode> checksum mode (all,noip,notcp,noudp,noicmp,none)
-A set alert mode to fast, full, console, or none
-X dump raw data starting at the Link Layer (the –d option could have been used to dump only the application layer but I wanted the output to provide a hex offset for explanation of the DNS packet structure).

The IP header checksum and TCP header checksum were invalid in all packets within the 2002.10.1.eth file and only produced a single alert when Snort was run without the "–k none" parameter, this was due to the packet mangling process and data obfuscation carried out by SANS on the source data.

```
Snort processed 14597 packets.
Breakdown by protocol:                Action Stats:

    TCP: 14597      (100.000%)        ALERTS: 3084
    UDP: 0          (0.000%)          LOGGED: 3084
   ICMP: 0          (0.000%)          PASSED: 0
    ARP: 0          (0.000%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
    IPX: 0          (0.000%)
  OTHER: 0          (0.000%)
```

The Snort alerts that I decided to analyse further were 3 “DNS Zone Transfer TCP” attempts:

The Snort signature that triggered these particular alerts was:

```
# $Id: dns.rules,v 1.29 2003/05/14 18:07:56 cazz Exp $

   alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS zone transfer TCP";
   flow:to_server,established; content: "|00 00 FC|"; offset:15;
   reference:cve,CAN-1999-0532; reference:arachnids,212; classtype:attempted-
   recon; sid:255; rev:8;)
```

Summary of Snort Alerts triggered:

The following console alert summary was produced with the command line: C:\EagleX\Snort\bin\Snort -r 2002.10.1.eth -c ..\etc\Snort.conf -l ..\logs -qk none –A console tcp and dst port 53 and dst host 207.166.87.159

```
11/01-16:14:05.656507  [**] [1:255:8] DNS zone transfer TCP [**]
167.206.112.181:55501 -> 207.166.87.159:53
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
11/01-20:01:04.886507  [**] [1:255:8] DNS zone transfer TCP [**]
167.206.112.181:56530 -> 207.166.87.159:53
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
11/01-23:45:46.636507  [**] [1:255:8] DNS zone transfer TCP [**]
167.206.112.181:57551 -> 207.166.87.159:53
```

Snort Alerts triggered (full):

The console output below was produced with the following command line :

C:\EagleX\Snort\bin>Snort -r 2002.10.1.eth -c ..\etc\Snort.conf -l ..\logs -qveXk none –A console tcp and dst port 53 and dst host 207.166.87.159

```
[**] DNS zone transfer TCP [**]
11/01-16:14:05.656507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x50
167.206.112.181:55501 -> 207.166.87.159:53
TCP TTL:249 TOS:0x0 ID:35551 IpLen:20 DgmLen:66 DF
***AP*** Seq: 0x97ACD3F9  Ack: 0x483766DF  Win: 0x8052  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 42 8A DF 40 00 F9 06 01 57 A7 CE 70 B5 CF A6  .B..@....W..p...
0x0020: 57 9F D8 CD 00 35 97 AC D3 F9 48 37 66 DF 50 18  W....5....H7f.P.
0x0030: 80 52 5A DA 00 00 44 EF 00 00 00 01 00 00 00 00  .RZ...D.........
0x0040: 00 00 04 58 58 58 58 03 63 6F 6D 00 00 FC 00 01  ...XXXX.com.....

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

```
[**] DNS zone transfer TCP [**]
11/01-20:01:04.886507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x50
167.206.112.181:56530 -> 207.166.87.159:53
TCP TTL:249 TOS:0x0 ID:24033 IpLen:20 DgmLen:66 DF
***AP*** Seq: 0x6B472F9A  Ack: 0xA166B801  Win: 0x8052  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 42 5D E1 40 00 F9 06 2E 55 A7 CE 70 B5 CF A6  .B].@....U..p...
0x0020: 57 9F DC D2 00 35 6B 47 2F 9A A1 66 B8 01 50 18  W....5kG/..f..P.
0x0030: 80 52 77 6C 00 00 4A CB 00 00 00 01 00 00 00 00  .Rwl..J.........
0x0040: 00 00 04 58 58 58 58 03 63 6F 6D 00 00 FC 00 01  ...XXXX.com.....

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

[**] DNS zone transfer TCP [**]
11/01-23:45:46.636507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x50
167.206.112.181:57551 -> 207.166.87.159:53
TCP TTL:249 TOS:0x0 ID:5587 IpLen:20 DgmLen:66 DF
***AP*** Seq: 0x4B3F04A  Ack: 0xF152BF10  Win: 0x8052  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 42 15 D3 40 00 F9 06 76 63 A7 CE 70 B5 CF A6  .B..@...vc..p...
0x0020: 57 9F E0 CF 00 35 04 B3 F0 4A F1 52 BF 10 50 18  W....5...J.R..P.
0x0030: 80 52 AF 26 00 00 5D FC 00 00 00 01 00 00 00 00  .R.&..].........
0x0040: 00 00 04 58 58 58 58 03 63 6F 6D 00 00 FC 00 01  ...XXXX.com.....

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

Taking the above alert (11/01-23:45:46.636507) as an example - the DNS payload (marked in bold and underlined) can be broken down as follows:

Offset	Hex	Bytes	Description
0x0036-37	5DFC	2 bytes	Length of DNS payload=24,060 (dec)
0x0038-39	0000	2 bytes	DNS Transaction ID=0
0x003a-3b	0001	2 bytes	Flags (standard query)=001
0x003c-3d	0000	2 bytes	Number of Questions=0
0x003e-3f	0000	2 bytes	Answer Records=0
0x0040-41	0000	2 bytes	Authority Records=0
0x0042-43	0458	2 bytes	Additional Records=1,112 (dec)
0x0044-4d	[04]XXXX.com**[0000FC]**		Query
0x004e-4f	0001	2 bytes	Class=0001

The Snort rule triggered on the basis that this was a TCP(6) packet to destination port 53 and matched content "**0000FC**" (AXFR – full Zone transfer request) at payload offset 15.

When analysing these 3 packets further via Ethereal it indicated that the 3 packets were "Unreassembled Packets", which I believe it reported because the "decoder" interpreted the DNS length field with a value of 24,060 bytes and / or the "Additional Records" field being set to 1,112. It appears therefore to Ethereal that further TCP segments are expected.

Probability the source address was spoofed:

This is highly unlikely as the alert indicates a full 3-way TCP handshake has already been completed and the source is now supposedly sending DNS “formed” packets to the target.

Description of attack:

The Domain Name Service (DNS) is a critical component of the Internet and is used for resolution of names to IP addresses and vice versa, it is based upon either “lookups” requested by clients (UDP protocol based) or “zone” transfers (TCP protocol based) request packets forwarded by secondary DNS servers. A “zone” is an independently administered domain within the DNS hierarchy e.g. ebay.com, zoom.co.uk and theses may be further sub-dived into smaller “zones” such as “sales.ebay.com” and “support.zoom.co.uk”.

A Zone (name domain) transfer is the process by which the primary DNS database allows any secondary authoritive DNS server(s) to remain synchronized with itself by the secondary server(s) verifying the serial number field in the SOA resource record for each zone.

DNS servers should normally be configured to only allow zone transfers between primary and secondary DNS servers and no other system as the record fields contain various types of sensitive data that will be of use to attacker(s), such as the IP addresses of critical hosts and possible hardware information of internal machines (hinfo). IDS or Firewall logs that indicate that port 53/tcp is being used are generally an early pointer that a network is being actively probed, a caveat to the use of TCP however is that normal client lookups can also make use of 53/tcp when the requested (queried) data will not fit within the maximum 512 byte upper limit for normal UDP client responses.

Attack mechanism:

Packet breakdown:

```
(#1) 11/01-16:14:05.656507  167.206.112.181:55501 -> 207.166.87.159:53     IP ID=35551
                            |
                   src port increment= +1029
                   time increment= +13499s
                   avg source port increment= 4.5736/min
                   win=32850
                   IP ID increment=  -11518
                            |
(#2) 11/01-20:01:04.886507  167.206.112.181:56530 -> 207.166.87.159:53     IP ID=24033
                            |
                   src port increment= +1021
                   time increment= +13482s
                   avg source port increment= 4.543/min
                   win=32850
                   IP ID increment=  -18446
                            |
 (#3) 11/01-23:45:46.636507  167.206.112.181:57551 -> 207.166.87.159:53    IP ID=5587
                            |
                   win=32850
```

Although the packets appeared as normal “DNS” requests there are a number of anomalies with all 3 packets that indicate packet crafting:

1) Length field indicates an average 20KB DNS payload
2) DNS Transaction ID is always set to 0 (this is used to track responses to requests that are sent and should be different for each request)
3) Question field is set to 0 (this should be 1 as the AXFR record has been set)
4) Additional Records field is constantly set to 1,112 (dec)

Additional data: IP Window size = 32850, IP ttl = 249 (hop count assumed to be 6)

The alert data was also piped through p0f to attempt to identify the source operating system. P0f in this situation did not produce any additional information as to the attacking OS version as it utilises SYN packets for it’s primary analysis against it’s “p0f.fp” fingerprint database. The alternative modes of –A (SYN+ACK) and –R (RST/RST+ACK) did not produce any results for the addresses either as all SYN and SYN+ACK packets had been purged from the alert file.

A manual check through the p0f.pf text file matched a window size of 32850 against NetApp Data OnTap OS however the ttl of 64 did not match the original 255 being assumed for the packet source, an Internet search also found a small number of references to NetApp.

OS signature (p0f.txt):
32850:64:1:64:N,W1,N,N,T,N,N,S,M*: NetApp:5.x::**NetApp Data OnTap** 5.x

Correlations:

Using the online utilities at www.hexillion.com provided the following “Domain Dossier” information for the **167.206.112.181** address:

Domain Dossier Investigate domains and IP addresses

canonical name **olympus.srv.hcvlny.cv.net.**

Network Whois record:

```
   Querying whois.arin.net with "167.206.112.181"...

   Cablevision Systems Corp. CVNET (NET-167-206-0-0-1)
                                 167.206.0.0 - 167.206.255.255
   Cablevision Systems Corp CVNET-SERVERS (NET-167-206-112-0-1)
                                 167.206.112.0 - 167.206.112.255

   Querying whois.arin.net with "!NET-167-206-112-0-1"...

   CustName:   Cablevision Systems Corp
   Address:    111 New South Road
   City:       Hicksville
   StateProv:  NY
```

```
PostalCode: 11801
Country:    US
RegDate:    1999-05-08
Updated:    1999-05-08

NetRange:   167.206.112.0 - 167.206.112.255
CIDR:       167.206.112.0/24
NetName:    CVNET-SERVERS
NetHandle:  NET-167-206-112-0-1
Parent:     NET-167-206-0-0-1
NetType:    Reassigned
Comment:
RegDate:    1999-05-08
Updated:    1999-05-08
```

Similarly for the **207.166.87.159** address:

Domain Dossier Investigate domains and IP addresses

canonical name **host-207-166-87-159.ucn.net**

Network Whois record:

```
Querying whois.arin.net with "207.166.87.159"...

OrgName:    I-Link Worldwide Inc
OrgID:      ILKW
Address:    13751 S Wadsworth Park Dr, Suite 200
City:       Draper
StateProv:  UT
PostalCode: 84020
Country:    US

NetRange:   207.166.64.0 - 207.166.111.255
CIDR:       207.166.64.0/19, 207.166.96.0/20
NetName:    I-LINK3
NetHandle:  NET-207-166-64-0-1
Parent:     NET-207-0-0-0-0
NetType:    Direct Allocation
NameServer: NS.I-LINK.NET
NameServer: NS1.I-LINK.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    1996-07-10
Updated:    2001-09-28

TechHandle: CN214-ARIN
TechName:   Nielson, Casey
TechPhone:  +1-801-576-5000
```

There are no reports on www.dshield.org or www.mynetwatchman.com against IP address 167.206.112.181.

References:

1) http://www.ethereal.com
2) http://www.whitehats.com/ids/

3) http://www.cert.org/advisories/
4) http://www.isc.org/products/BIND/
5) http://www.stearns.org/p0f/
6) http://www.freesoft.org/CIE/Course/Section4/9.htm
7)http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/techref/w2k3tr_dns_how.asp
8) http://razor.bindview.com/publish/papers/tcpseq.html

Evidence of active targeting:

This attack is originating from the same source address and to the same destination with no indications from the alert file of any probes to other targets within the protected network range, it also does not appear to have created any response from the target. However, the host may have been probed via earlier reconnaissance or via whois and DNS server lookups.

Severity:

The following ratings are marked against scale: **1** (lowest) to **5** (highest)

Criticality = 5 (measure of target(s) value / critical nature)
The DNS server is a vital component for normal site operation and would incapacitate any site if it were compromised.

Lethality = 4 (if the attack(s) succeeded / damage estimate)
This trace was part of an established connection and may either have been an attempt at a new compromise via a buffer overflow type attack or other type of DNS software instability. CERT currently has 33 advisories related to DNS issues/ vulnerabilities.

System Countermeasures = 3 (host(s) defence posture)
This is difficult to calculate without access to full network traces, the number of DNS alerts detected appears to indicate a regular interval attack against the intended target at an approximate rate of 3.5 – 3.75 hours indicating that these types of requests are failing.

Network Countermeasures = 3 (network (s) defensive systems / configuration)
Again without access to full network flows and the manipulation of data by SANS the full network countermeasures rating is difficult to estimate. The network appears to be allowing outbound P2P file access (Gnutella) and MSN Messenger access, although the range of "well-known" ports inbound does appear to controlled to a degree by router access lists. Source port NAT'ing also appears not to be in use.

Severity Rating = 3
(criticality + lethality) – (system countermeasures + network countermeasures)

Defensive recommendation:

Ensure that the Primary DNS server only allows TCP zone transfers from authoritive secondary DNS servers, BIND v8/9 can be configured via the “allow-transfer” directive and Windows 2000 via Allow Zone transfer tab option.

If zone transfers cannot be restricted on the DNS server itself then border routers/ firewalls should be configured with specific rules for IP addresses allowed to carry-out do zone transfers.

Consider implementation of split-DNS that would allow resolution via external and internal DNS servers i.e. use an external DNS server for all inbound requests from the Internet and a separate DNS server inside the firewall perimeter for resolution of all internal based services.

Multiple choice test question:

How many types of Zone transfer are there?

a. Two - AXFR and IXFR types
b. One - AXFR is currently the only supported type
c. Four – AXFR, IXFR, SRV and YXDOMAIN types
d. Three – AXFR, IXFR and NXDOMAIN types

Answer : a

There are two types of zone transfer (or duplication transfer). Full zone transfer (AXFR) replicates the entire zone, whereas incremental zone transfer (IXFR) replicates only zone records that have changed. SRV = Service location (a common DNS resource record), YXDOMAIN = DNS update error code (a name that should not exist does exist) and NXDOMAIN = DNS update error code (a name that should exist does not exist).

This detect was posted to the incidents.org mailing list on Tue 23/12/2003 at 15:46 for comment however no feedback was received.

Assignment # 3: Analyze This (scenario-based)

Executive Summary

SANS University have requested a security audit of their campus and have provided unrestricted access to the University's Intrusion Detection System logs to enable this to be accomplished. Emphasis was to be placed upon the detection of any University systems that may have:

1. become compromised via external hacking, Trojan, Worm or rootkit activity
2. been engaged in illegal or unauthorised network activities (against the University's acceptable usage policy)
3. possibly been targeted by external systems (non-MY.NET hosts)

The audit took place between the 19th and 23rd Dec 2003 and a number of areas have been identified that require further attention, these are NIMDA, Code Red infected hosts and Trojan activity from MY.NET.42.1 and 42.3 hosts. Further data packet captures are required for a number of events that indicate host compromise. There also appears to be a number of servers infected with the SubSeven Trojan communicating to external addresses.

Files Analyzed

The following files were downloaded from the GIAC website http://www.incidents.org/logs for the University analysis scenario and cover the 5 day period from Dec 19th to Dec 23rd:

Alert Logs	Scan Logs	Out of Spec Logs
alert.031219.gz	scans.031219.gz	n/a – contains OOS data for 27/10
alert.031220.gz	scans.031220.gz	oos_report_031216.txt
alert.031221.gz	scans.031221.gz	oos_report_031217.txt
alert.031222.gz	scans.031222.gz	oos_report_031218.txt
alert.031223.gz	scans.031223.gz	oos_report_031219.txt

There appeared to be a problem with the oos_report_031219.txt file as this contained information for 27/10 so I was unable to use this data within the analysis. Also, the OOS report log data for 031220 through 031223 was actually contained in the GIAC website files 031216 through 031219 and this is recorded above.

The log files consists of 3 types of information – Snort generated alerts, Scan output and Out of Specification (abnormal) packet data which is used for further collaboration against any alert or scan events. There were no raw "binary" logs available for this particular analysis.

The Snort scan logs contain events for each IP address or TCP / UDP port scanned so this created an incredibly large scan total of 17.6 Million entries over the 5 day analysis period.

Scan data is also duplicated to a degree in the Alert file which uses the Snort Portscan Pre-Processor (SPP) to collate multiple scan entries. OOS logs contain data relating to packets with illegal IP or header flag combinations being enabled.

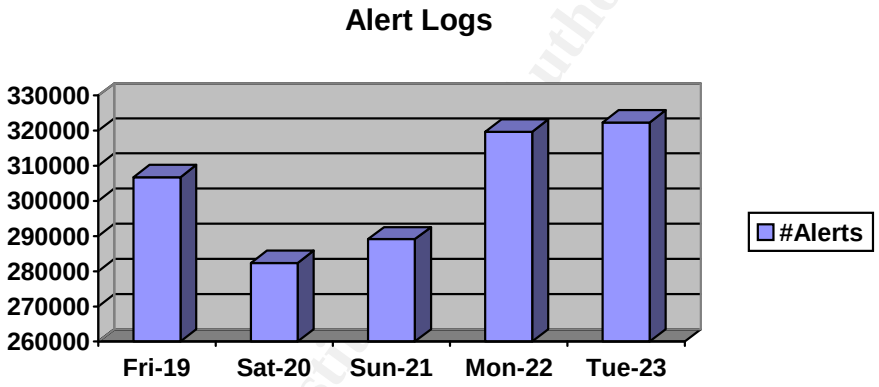
The number of events per day for each particular log type is summarized and graphed below:

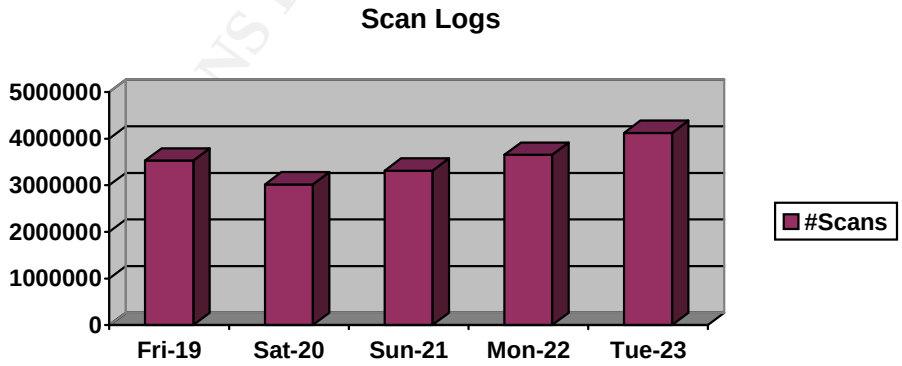
Summary data:

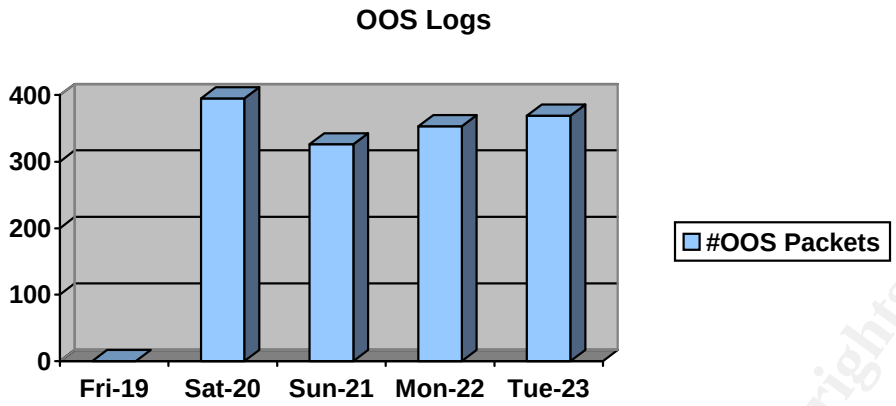
Date	Alert Logs	Scan Logs	OOS Logs
12/19/03	306758 entries	3529881 entries	n/a
12/20/03	282394 entries	3015283 entries	395 entries
12/21/03	289127 entries	3312726 entries	326 entries
12/22/03	319659 entries	3649436 entries	353 entries
12/23/03	322326 entries	4120037 entries	369 entries
Total	**1520264**	**17627363**	**1443**

Using the above Alert Log total there was an average of 211.15 alerts per minute over the 5-day period, which indicates either a very high level of suspicious activity or un-tuned IDS.

Graph of Alerts, Scans and OOS data:







Analysis

I placed the main focus of the analysis on the University's Snort Alert logs with the Scan and OOS files being used to correlate events of concern. Although scan data is important it must be balanced against the overall picture so that real attacks can be identified. Due to the duplication to a large extent between the Alert logs and Scan logs and the availability of Perl analysis scripts it was possible to remove all of the "spp_portscan" events from the alert logs to allow easier manipulation of the recorded data, the following table therefore indicates the "Events Of Interest" that were collated following processing by various Perl scripts made available by previous SANS GCIA analysts (further details of these are provided in the Analysis Process section) -

Total Events Of Interest (post csv.pl Perl script run i.e. scans removed):

Date	Alert Logs (total entries)
12/19/03	18172
12/20/03	18695
12/21/03	14826
12/22/03	22832
12/23/03	15212
Total	**89,619**

These alerts will be the basis of the analysis.

Detects Listing

The following table details the total count output for each of the unique Events of Interest (EOI) that were produced. The Top 10 Alerts (in bold) account for 88212 events, which represents 98.3% of the overall alert total.

Table 1. Summary of Snort Alerts

Alert Description	Total	INT	EXT	Unique Source	Unique Dst
MY.NET.30.3 activity	23810	0	23810	126	1
MY.NET.30.4 activity	21855	0	21855	243	1
Incomplete Packet Fragments Discarded	13663	126	13537	141	165
TFTP - Internal TCP connection to external tftp server	7865	4606	3259	18	23
EXPLOIT x86 NOOP	4713	0	4713	177	151
SMB Name Wildcard	4343	4713	0	95	88
connect to 515 from inside	3556	3556	0	4	4
High port 65535 udp - possible Red Worm - traffic	3238	1402	1836	188	249
ICMP SRC and DST outside network	1713	0	1713	71	112
NMAP TCP ping!	1696	0	1696	137	59
High port 65535 tcp - possible Red Worm - traffic	984	661	323	73	132
Null scan!	662	0	662	55	46
Possible trojan server activity	323	152	171	46	45
TCP SRC and DST outside network	272	0	272	19	55
[UMBC NIDS IRC Alert] IRC user /kill detected	172	0	172	47	34
SUNRPC highport access!	171	0	171	4	4
FTP passwd attempt	118	0	118	90	2
SMB C access	107	0	107	45	3
[UMBC NIDS] External MiMail alert	79	0	79	42	1
EXPLOIT x86 setuid 0	45	0	45	39	28
EXPLOIT x86 setgid 0	33	0	33	21	25
RFB - Possible WinVNC - 010708-1	30	21	9	15	10
FTP DoS ftpd globbing	29	0	29	8	1
TFTP - External TCP connection to internal tftp server	16	6	10	5	5
EXPLOIT NTPDX buffer overflow	11	0	11	8	6
Tiny Fragments – Possible Hostile Activity	10	9	1	4	4
EXPLOIT x86 NOPS	9	0	9	1	1
Attempted Sun RPC high port access	8	0	8	4	4
EXPLOIT x86 stealth noop	8	0	8	6	6
IRC evil - running XDCC	8	8	0	1	2
Probable NMAP fingerprint attempt	8	0	8	4	5
TFTP - Internal UDP connection to external tftp server	7	1	6	4	5
TFTP - External UDP connection to internal tftp server	7	0	7	2	2
DDOS mstream client to handler	5	0	5	3	3
External FTP to HelpDesk MY.NET.70.50	5	0	5	5	1
External FTP to HelpDesk MY.NET.53.29	5	0	5	3	1
External FTP to HelpDesk MY.NET.70.49	5	0	5	5	1
[UMBC NIDS IRC Alert] K\:line'd user detected	5	0	5	2	2
DDOS shaft client to handler	5	0	5	1	1
NIMDA - Attempt to execute cmd from campus host	4	4	0	3	2
[UMBC NIDS IRC Alert] User joining Warez channel detect	4	0	4	4	3
[UMBC NIDS IRC Alert] User joining XDCC channel detect	3	0	4	3	2
EXPLOIT identd overflow	2	0	2	2	2
Traffic from port 53 to port 123	1	1	1	1	1
[UMBC NIDS IRC Alert] Possible sdbot floodnet detected .	1	1	0	1	1
Bugbear@MM virus in SMTP	1	0	1	1	1
TCP SMTP Source Port traffic	1	0	1	1	1

Happy 99 Virus	1	0	1	1	1
Possible wu-ftpd exploit - GIAC000623	1	0	1	1	1
PHF attempt	1	0	1	1	1

The Top 10 alerts in terms of event occurrences were:

Alert #1 (23810 alerts)**:** **MY.NET.30.3 activity**

Example: 12/19-00:03:10.872391 [**] MY.NET.30.3 activity [**] 68.55.113.28:1031 -> MY.NET.30.3:xx

SID: None – custom rule
(SID = Snort ID – unique reference)

Brief description:
This is a custom rule that has been added by the University to alert on any activity to the internal host MY.NET.30.3, the logs indicate activity to the destination ports indicated below:

21	23	80	524	1023	1123	1257	1258	1337	1489	1730	1930
2036	3019	3389	3810	4899	5900	6129	20168				

Unique IP sources: 126
All alerts were generated from external IP addresses and there were no OOS log references to host MY.NET.30.3.

Correlation:
None available as this is a custom rule.

Defensive Recommendations:
It is difficult to identify the purpose of this rule apart from a heightened level of awareness on any activity targeting this system as a wide range of ports are triggering the IDS. If the University is concerned with activity to this host then it would improve the IDS's effectiveness if more specific rules were also implemented e.g. for buffer overflow, rootkit or Trojan activity.

Alert #2 (21855 alerts)**:** **MY.NET.30.4 activity**

Example: 12/19-00:01:42.136009 [**] MY.NET.30.4 activity [**] 66.196.72.58:40771 -> MY.NET.30.4:xx

SID: None – custom rule

Brief description:
This is another custom rule that has been added by the University to alert on any activity to the internal host MY.NET.30.4, logs indicate activity to destination ports indicated below:

21	23	80	524	1022	1038	1064	1117	1163	1175	1257	1290
1477	1536	1604	1703	1771	1803	1861	2036	3389	3800	3810	4899

5900	6129	1808	20168	51443	52080						

Unique IP sources: 243
All alerts were generated from external IP addresses and there were no OOS log references to host MY.NET.30.4. The University should consider more specific rules as commented above.

Correlation:
None available as this is a custom rule.

Defensive Recommendations:
It is also difficult to identify the purpose of this rule apart from a heightened level of awareness on any activity targeting this system as a wide range of ports are being triggered upon (23-52080).

Alert #3 (13667 alerts)**:** **Incomplete Packet Fragments Discarded**

12/19-01:12:29.449663 [**] Incomplete Packet Fragments Discarded [**] 203.113.233.203:0 -> MY.NET.82.117:0

SID: None

Brief description:

The Snort fragmentation processor is responsible for IP fragments re-assembly and prior to Snort v1.8 was based upon the “defrag” preproccesor, this (according to snort.org FAQ’s and mailing lists) produced erroneous alerts and was replaced by the “frag2” engine.

Correlation:
http://www.snort.org/docs/snort_manual/node17.html#SECTION00383000000000000000
http://www.mcabee.org/lists/snort-users/Nov-01/msg00820.html

Defensive Recommendations:
This indicates that the University is likely to be using an old Snort version / configuration, the latest stable release should be used with a current ruleset, also these alerts should be investigated further to confirm these are definitely false positives from the defrag pre-processor.

Alert #4 (7869 alerts)**:** **TFTP - Internal TCP connection to external tftp server**

12/19-08:31:06.864717 [**] TFTP - Internal TCP connection to external tftp server [**] MY.NET.70.225:1736 -> 68.61.18.36:69

SID: None – custom rule

Brief description:

TFTP (Trivial File Transfer Protocol) is a simpler form of FTP (File Transfer Protocol) and normally runs over UDP, it has no user authentication or security methods and is normally used for high throughput transfers (e.g. used by diskless workstation boot-up or router firmware upgrades). This alert however is alerting to TCP protocol port 69 usage.

Correlation:
None available as this is a custom rule.

Defensive Recommendations:
TFTP is used for valid transfer such as IOS router uploads and workstation boot-up but this is normally via UDP – this alert again requires further packet analysis to determine the type of connections being made Internally and externally via this port.

Alert #5 (4713 alerts)**:** **EXPLOIT x86 NOOP**

12/19-01:07:32.708769 [**] EXPLOIT x86 NOOP [**] 65.203.33.194:13978 -> MY.NET.190.95:135

SID: None

Brief description:
NOOP is a machine code 0x90 which means a "No-Operation instruction" and following various web searches appears to have caused many false positives by the simple matching of 24 contiguous 0x90 byte sequence within a packet payload.

Correlation:
A previously posted practical by David Oborn, GCIA details this alert further and the EXPLOIT x86 NOOP rule appears to have been re-named SHELLCODE-X86-NOPS according to the arachNIDS database at
http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids181&view=signatures

Defensive Recommendations:
This alert has a high false positive level because this sequence can occur quite normally in binary file download streams, to ensure this is a benign threat a full packet capture should be taken to analyse this alert in greater detail.

Alert #6 (4343 alerts)**:** **SMB Name Wildcard**

12/19-00:08:33.610855 [**] SMB Name Wildcard [**] MY.NET.11.6:137 -> 169.254.0.0:137

SID: None

Brief description:
SMB (Server Message Block) is part of the Microsoft Network filesharing system and is used to provide a number of integrated services, the NetBIOS name query that generated this alert was caused by a UDP request to destination port 137. The Name Wilcard events indicate that a an IP address NAME resolution has been requested (this is part of the normal Windows architecture but can also be used for attempted enumeration for a

workstation name, domain name or user name). The 4343 alerts were generated from internal MY.NET devices as broadcasts or requests to external hosts.

Correlation:
http://www.microsoft.com
http://www.whitehats.com/info/IDS177

Defensive Recommendations:
SMB / NetBIOS should not be allowed to egress from an internal network and should always be blocked either inbound or outbound at the network perimeter, where this is not possible then protocol tunnelling techniques should be considered.

Alert #7 (3556 alerts)**: connect to 515 from inside**

12/19-00:08:35.090261 [**] connect to 515 from inside [**] MY.NET.162.41:721 -> 128.183.110.242:515

SID: None – custom rule

Brief description:
This rule has been added by the University to trigger whenever a connection is made to an external IP address on destination port 515 from any MY.NET IP address range. The destination port is normally associated with the Unix line printer service (lpr daemon) and all connections were attempted from source port 721. Out of the 3556 alerts that were generated the host MY.NET.162.41 produced 3544 events and was repeatedly connecting to 128.183.110.242 which according to an ARIN whois lookup is registered to NASA. 3 connects from MY.NET.60.16 were made to 66.160.63.18 which resolves to Windermere Technologies, US. The remaining alerts were to 192.168.0 and 192.168.2 targets (possible mis-configuration)

Correlation:
Further information on known issues with the line printer service can be found at http://www.kb.cert.org/vuls/byid

Defensive Recommendations:
Ensure that firewall and router ACL's are correctly configured for outbound port 515 if remote printing is required consider revising Snort rules to monitor for more specific lpr/lpd vulnerabilities.

Alert #8 (3242 alerts)**: High port 65535 udp - possible Red Worm - traffic**

12/19-00:18:32.372769 [**] High port 65535 udp - possible Red Worm - traffic [**] MY.NET.163.76:6257 -> 218.102.85.203:65535

SID: None – custom rule

Brief description:

This is custom rule designed to trigger on likely worm activity; there are 3242 events by Alert Message recorded and this likely to be associated with Code Red and Code Red II type worms.

Correlation:
http://www.cert.org/advisories/CA-2001-19.html
http://www.cert.org/incident_notes/IN-2001-09.html

Alert #9 (1713 alerts)**:** **ICMP SRC and DST outside network**

12/19-00:04:42.158743 [**] ICMP SRC and DST outside network [**] 172.166.97.125 -> 172.168.67.165

SID: None – custom rule.

Brief description:
This IDS rule has been enabled by the University to trigger whenever an ICMP packet is seen that contains "external" IP addresses for both the source and destination fields i.e. do not equal HOME_NET variable. This is likely to be spoofed address attacks to either intentionally generate "spurious" IDS events to increase noise or a possible attempt at overloading an IDS / Firewall / router logging system prior to an attack.

Correlation:
None available as this is a custom rule.

Defensive Recommendations:
Run a full packet trace to ensure ICMP is not being re-directed by some means at the network perimeters, ensure that network perimeter router ACL's and Firewall rulebases contain anti-spoof rules (RFC1918 addresses plus University Internal address ranges).

Alert #10 (1696 alerts)**:** **NMAP TCP ping!**

12/19-00:14:57.676803 [**] NMAP TCP ping! [**] 61.30.119.193:80 -> MY.NET.1.4:53

SID: None – custom rule.

Brief description:
NMAP (Network Mapper) is a powerful Open Source network utility used for exploration and security audit mapping. It runs across a wide range of platforms and has numerous options for scanning and fingerprinting target systems or networks. This rule appears to match the Snort rule "SCAN nmap TCP" SID628 and is using the NMAP TCP ping option that sends a TCP ACK with an ACK number set to 0. Out of the 1696 alerts there were 137 unique sources and 59 unique destinations detected, all alerts were inbound to MY.NET addresses.

Correlation:
http://www.insecure.org/nmap/
http://www.snort.org/snort-db/sid.html?sid=628

http://members.dodo.net.au/~ps2man/Nmap/nmap.html

Defensive Recommendations:
This particular scan type is being used for reconnaissance purposes and generally uses TCP ACK packets on Port 80, this allows hosts to be identified that are be behind Firewalls or systems that block ICMP Ping packets.

The following list details total counts for each of the Snort Events of Interest (EOI)

Top 10 Talkers

Table 2. Summary of Top 10 Internal and External IP addresses

Events of Interest by Top 10 Internal Source Adresses

Total	Internal Address
3546	MY.NET.162.41
3115	MY.NET.21.67
2838	MY.NET.21.92
2648	MY.NET.21.68
2330	MY.NET.21.79
2295	MY.NET.42.1
1990	MY.NET.42.3
1845	MY.NET.11.6
1391	MY.NET.21.89
1346	MY.NET.163.76

Top 10 from total uniques of 247.

Events of Interest by Top 10 External Source Adresses

Total	External Address	DNS Reverse Lookup
1101	67.20.173.236	md-wmnsmd-cuda1-c6c-236.chvlva.adelphia.net
988	68.32.122.89	pcp01840932pcs.owngsm01.md.comcast.net
796	66.168.239.240	Unresolved
658	219.48.176.27	Unresolved
653	68.55.27.157	pcp02560368pcs.owngsm01.md.comcast.net
587	67.20.160.15	md-wmnsmd-cuda2-c2d-15.chvlva.adelphia.net
534	131.92.177.18	aeclt-cfdoa4.apgea.army.mil
502	151.196.116.233	pool-151-196-116-233.balt.east.verizon.net
499	151.196.10.167	pool-151-196-10-167.balt.east.verizon.net
459	141.157.68.234	pool-141-157-68-234.balt.east.verizon.net

Top 10 from total uniques of 1453.

External IP reference (x5)

5 Hosts indicating internal TFTP port 69 TCP connection activity to external addresses:

IP address #1:

```
Address lookup
canonical name host109-186.pool21759.interbusiness.it.
aliases
addresses 217.59.186.109

Domain Whois record
it = Italy

Querying whois.nic.it with "interbusiness.it"...

domain:      interbusiness.it
x400-domain: c=it; admd=0; prmd=interbusiness;
org:         Telecom Italia S.p.A.
descr:       InterBusiness
descr:       Network Service Provider
admin-c:     CD2-ITNIC
tech-c:      FG82-ITNIC
tech-c:      GLM2-ITNIC
postmaster:  FG82-ITNIC
zone-c:      DRS9-ITNIC
nserver:     151.99.125.2 dns.interbusiness.it
nserver:     193.205.245.66 dns3.nic.it
nserver:     151.99.250.2 server-b.cs.interbusiness.it
nserver:     151.99.125.138 dns.opb.interbusiness.it
remarks:     Fully Managed
remarks:     Please report Spam/Abuse only to abuse@interbusiness.it
mnt-by:      INTERBUSINESS-MNT
created:     before 19960129
expire:      20040129
changed:     domain@cgi.interbusiness.it 20020426
source:      IT-NIC

person:      Camillo Di Vincenzo
address:     Telecom Italia S.P.A.
address:     Via Paolo Di Dono, 44
address:     I-00143 Roma
address:     Italy
phone:       +39 06 36871
fax-no:      +39 06 36871
nic-hdl:     CD2-ITNIC
changed:     domain@cgi.interbusiness.it 20001115
changed:     hostmaster@nic.it 20030424
changed:     hostmaster@nic.it 20030428
source:      IT-NIC


Network Whois record

Querying whois.ripe.net with "217.59.186.109"...

% This is the RIPE Whois server.
% The objects are in RPSL format.

inetnum:      217.59.186.0 - 217.59.186.127
netname:      AFA-SYSTEMS
descr:        AFA SYSTEMS S.R.L.
```

```
country:      IT
admin-c:      ADN84-RIPE
tech-c:       ADN84-RIPE
status:       ASSIGNED PA
notify:       network@cgi.interbusiness.it
mnt-by:       INTERB-MNT
changed:      network@cgi.interbusiness.it 20010326
source:       RIPE

route:        217.56.0.0/14
descr:        INTERBUSINESS
origin:       AS3269
remarks:      Send report of network abuse/spam
remarks:      only to: abuse@interbusiness.it .
remarks:      If you report abuse to any other address
remarks:      you will get no response.
notify:       network@cgi.interbusiness.it
mnt-by:       INTERB-MNT
changed:      mattu@cgi.interbusiness.it 20011009
source:       RIPE

person:       Antonio Di Nonno
address:      AFA SYSTEMS S.R.L.
address:      Loc. greppe di pantano
address:      I- 86039 Termoli CB
address:      Italy
phone:        +39 0875724104
fax-no:       +39 0875724104
nic-hdl:      ADN84-RIPE
changed:      domain@cgi.interbusiness.it 20010322
source:       RIPE
```

IP address #2:

```
Address lookup
canonical name pcp03529920pcs.pthurn01.mi.comcast.net.
aliases
addresses 68.61.18.36


Domain Whois record

Querying whois.internic.net with "dom comcast.net"...

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: COMCAST.NET
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: DNS01.JDC01.PA.COMCAST.NET
   Name Server: DNS02.JDC01.PA.COMCAST.NET
   Status: ACTIVE
   Updated Date: 17-nov-2003
```

```
   Creation Date: 25-sep-1997
   Expiration Date: 24-sep-2008

>>> Last update of whois database: Thu, 15 Jan 2004 06:36:07 EST <<<

Network Whois record
Querying whois.arin.net with "68.61.18.36"...

Comcast Cable Communications, Inc. JUMPSTART-1 (NET-68-32-0-0-1)
                                  68.32.0.0 - 68.63.255.255
Comcast Cable Communications, Inc. MICHIGAN-B-6 (NET-68-61-0-0-1)
                                  68.61.0.0 - 68.61.255.255

# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.

Querying whois.arin.net with "!NET-68-61-0-0-1"...

CustName:   Comcast Cable Communications, Inc.
Address:    3 Executive Campus
Address:    5th Floor
City:       Cherry Hill
StateProv:  NJ
PostalCode: 08002
Country:    US
RegDate:    2003-03-19
Updated:    2003-03-19

NetRange:   68.61.0.0 - 68.61.255.255
CIDR:       68.61.0.0/16
NetName:    MICHIGAN-B-6
NetHandle:  NET-68-61-0-0-1
Parent:     NET-68-32-0-0-1
NetType:    Reassigned
Comment:    NONE
RegDate:    2003-03-19
Updated:    2003-03-19

TechHandle: IC161-ARIN
TechName:   Comcast Cable Communications Inc
TechPhone:  +1-856-317-7200
TechEmail:  cips_ip-registration@cable.comcast.com

OrgAbuseHandle: NAPO-ARIN
OrgAbuseName:   Network Abuse and Policy Observance
OrgAbusePhone:  +1-856-317-7272
OrgAbuseEmail:  abuse@comcast.net

OrgTechHandle: IC161-ARIN
OrgTechName:   Comcast Cable Communications Inc
OrgTechPhone:  +1-856-317-7200
OrgTechEmail:  cips_ip-registration@cable.comcast.com

# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database
```

IP address #3:

```
Address lookup
canonical name martiab1.miniserver.com.
aliases
addresses 69.10.132.121


Domain Whois record
Querying whois.internic.net with "dom miniserver.com"...

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: MINISERVER.COM
   Registrar: TUCOWS, INC.
   Whois Server: whois.opensrs.net
   Referral URL: http://www.opensrs.org
   Name Server: NS1.MEMSET.COM
   Name Server: NS2.MEMSET.COM
   Name Server: NS3.MEMSET.COM
   Status: ACTIVE
   Updated Date: 09-apr-2003
   Creation Date: 02-apr-2002
   Expiration Date: 02-apr-2004


>>> Last update of whois database: Thu, 15 Jan 2004 06:36:07 EST <<<

Network Whois record
Querying whois.arin.net with "69.10.132.121"...

RackForce Hosting Inc. RACKFORCE-1 (NET-69-10-128-0-1)
                              69.10.128.0 - 69.10.159.255
Memset Ltd. MEMSET-MAINNET (NET-69-10-132-0-1)
                                69.10.132.0 - 69.10.132.255

# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.

Querying whois.arin.net with "!NET-69-10-132-0-1"...

OrgName:    Memset Ltd.
OrgID:      MEMSE
Address:    5 Wincanton Close
City:       Alton
StateProv:  Hants
PostalCode: GU342TQ
Country:    GB

NetRange:   69.10.132.0 - 69.10.132.255
CIDR:       69.10.132.0/24
NetName:    MEMSET-MAINNET
NetHandle:  NET-69-10-132-0-1
Parent:     NET-69-10-128-0-1
NetType:    Reassigned
NameServer: NS1.MEMSET.NET
NameServer: NS2.MEMSET.NET
Comment:    NONE
```

```
RegDate:    2003-05-09
Updated:    2003-05-09

OrgTechHandle: RCR12-ARIN
OrgTechName:   Craig-Wood, Robert
OrgTechPhone:  +44 1420 83999
OrgTechEmail:  tech@memset.com

# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

IP address #4:

```
Address lookup
canonical name corp.windermeregroup.com.
aliases
addresses 66.160.63.18


Domain Whois record
Querying whois.internic.net with "dom windermeregroup.com"...

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: WINDERMEREGROUP.COM
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: ANNAPOLIS-NS1.WINDERMEREGROUP.COM
   Name Server: ANNAPOLIS-NS2.WINDERMEREGROUP.COM
   Status: ACTIVE
   Updated Date: 25-nov-2002
   Creation Date: 27-jan-1998
   Expiration Date: 26-jan-2006


>>> Last update of whois database: Thu, 15 Jan 2004 06:36:07 EST <<<

Network Whois record

Querying whois.arin.net with "66.160.63.18"...

Cavalier Telephone CAVTEL-BLK-2 (NET-66-160-0-0-1)
                                  66.160.0.0 - 66.160.127.255
Windermere Technologies WINDERMERETECH46416 (NET-66-160-63-0-1)
                                  66.160.63.0 - 66.160.63.255

# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.

Querying whois.arin.net with "!NET-66-160-63-0-1"...

OrgName:    Windermere Technologies
OrgID:      WINDER-4
Address:    401 Defense Hwy
```

```
City:       Annapolis
StateProv:  MD
PostalCode: 21401
Country:    US

NetRange:   66.160.63.0 - 66.160.63.255
CIDR:       66.160.63.0/24
NetName:    WINDERMERETECH46416
NetHandle:  NET-66-160-63-0-1
Parent:     NET-66-160-0-0-1
NetType:    Reassigned
Comment:
RegDate:    2002-02-16
Updated:    2002-02-16
```

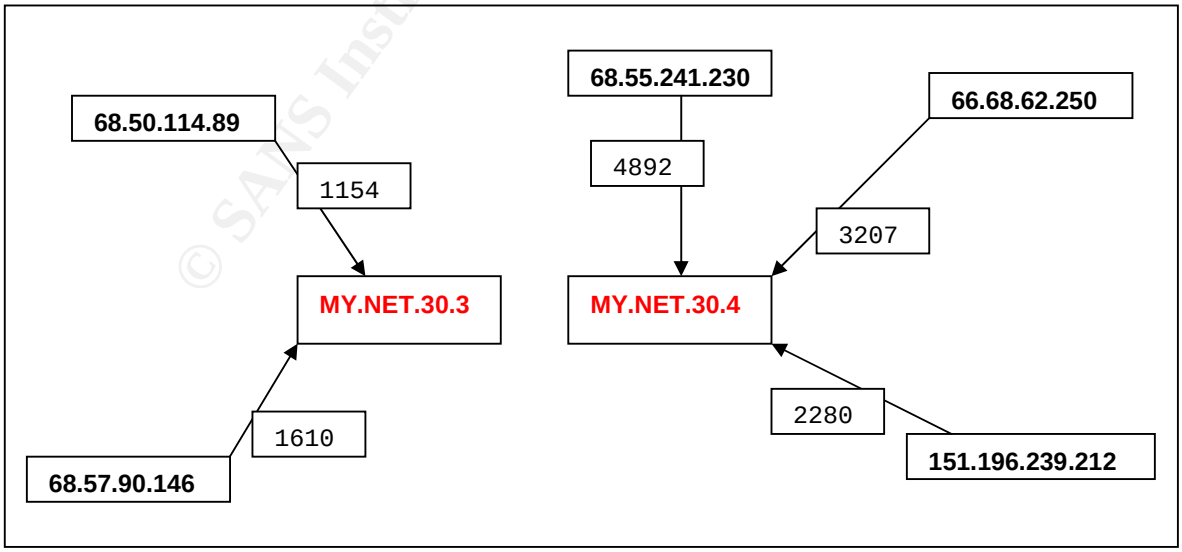
IP address #5:

```
Address lookup
canonical name stormy.membrain.com.
aliases
addresses 66.93.118.125


Domain Whois record
Querying whois.internic.net with "dom membrain.com"...

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: MEMBRAIN.COM
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NS1.MEMBRAIN.COM
   Name Server: NS3.MEMBRAIN.COM
   Name Server: NS2.MEMBRAIN.COM
   Status: ACTIVE
   Updated Date: 03-mar-2003
   Creation Date: 16-apr-1997
   Expiration Date: 17-apr-2008


>>> Last update of whois database: Thu, 15 Jan 2004 06:36:07 EST <<<

Querying whois.networksolutions.com with "membrain.com"...

Welcome to the Network Solutions Registrar WHOIS Server.

Network Whois record

Querying whois.arin.net with "66.93.118.125"...

Speakeasy Network SPEAKEASY-5 (NET-66-92-0-0-1)
                                  66.92.0.0 - 66.93.255.255
285562 SPEK-285562-0 (NET-66-93-118-112-1)
                                  66.93.118.112 - 66.93.118.127
```

```
# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.

Querying whois.arin.net with "!NET-66-93-118-112-1"...

CustName:   285562
Address:    21711 Filigree Court
City:       Ashburn
StateProv:  VA
PostalCode: 20147
Country:    US
RegDate:    2003-05-22
Updated:    2003-05-22

NetRange:   66.93.118.112 - 66.93.118.127
CIDR:       66.93.118.112/28
NetName:    SPEK-285562-0
NetHandle:  NET-66-93-118-112-1
Parent:     NET-66-92-0-0-1
NetType:    Reassigned
Comment:
RegDate:    2003-05-22
Updated:    2003-05-22

TechHandle: AS3414-ARIN
TechName:   Stollar, Andreas
TechPhone:  +1-206-728-9770
TechEmail:  abuse@speakeasy.net

# ARIN WHOIS database, last updated 2004-01-14 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```


Link Graph

Events of Interest by relationship (External -> Internal):



Defensive Recommendations

The University needs to take a number of steps to improve it's current security posture. Apart from the Code Red and Nimda infections there are multiple reconnaissance scans occurring on the external perimeter and to a lesser extent internally.

1. Implement a server / desktop Anti-Virus policy and install a major vendor's product – this will help ensure central updates and management can be implemented easily and effectively.
2. Locate and immunize infected hosts
3. Apply vendor recommended service patches and security updates
4. Ensure that network perimeters are secured via Access Control Lists and that Firewall policy is enforced in a more "deny" oriented fashion where possible i.e. consider the impact of open ports and compromise vectors against the requirements of un-restricted traffic that the University is normally used to.
5. Carry-out scheduled internal and external Penetration Tests (PEN) where and wherever possible to continuously confirm the condition of the network perimeter.

Description of Analysis Process

The logs were first downloaded and sanity checked, there were 10-15 entry corruptions in each of the alert logs and these were manually removed to allow correct output from a number of Perl scripts that were going to be run. The log files were also trimmed at midnight to ensure data did not contain any overlap data, the worst-case error rate produced a maximum of 0.1% of total alerts per day that could not be used in the analysis. The alert files were then run through csv.pl that created a comma separated daily alert file (e.g. alert.031219.csv). As mentioned earlier the normal alert file also contains Scan entries marked "spp_portscan" – the Perl script then also purged any references to this type of entry. This process was repeated for each of the alert log files and then all 5 .csv files were concatenated into a single .csv file containing the 89,619 entries. This file was then manipulated into 2 Excel spreadsheets to allow me to sort and analyse the alerts, finally this was reduced to a single spreadsheet following removal of the MY.NET.30.3 / .4 alerts and Incomplete Packet Fragments Discarded (these high number of alerts were analysed separately).

The main .csv file was then run through another Perl script called summarize.pl, this created a single .txt file that was used to collate alerts into "Events of Interest"; the EOI data contained the following group break-downs:

1) EOI's by Alert Message
2) EOI's by Source IP (External Only)
3) EOI's by Source Port (External Only)
4) EOIs by Relationship (External->Internal Only)
5) EOIs by Relationship (External->External Only)
6) EOIs by Source IP (Internal Only)
7) EOIs by Source Port (Internal Only)

8) EOIs by Relationship (Internal->Internal Only)
9) EOIs by Relationship (Internal->External Only)
10) EOIs by Destination IP (Internal Only)
11) EOIs by Destination Port (Internal Only
12) EOIs by Destination IP (External Only)
13) EOIs by Destination Port (External Only

I also used 2 further Perl scripts called scanalyze.pl and scancount.pl to analyze the Scan log files. The OOS logs were analyzed using Windows GVIM (a fast and powerful GUI text editor with equivalent power to the Unix vi program) and Microsoft Excel to collate the OOS data.

The scripts were created by 2 previous SANS analysts:

1) Tod A. Beardsley, GIAC GCIA Practical - csv.pl and analyze.pl
2) Chris Kuethe, GIAC GCIA Practical - scanalyze.pl and scancount.pl

Reference

1. SANS/FBI. "Top 20 most critical Internet Security Vulnerabilities." SANS. Oct. 17 2002
2. http://www.sans.org/top20/#index (Nov. 7 2002)
3. http://www.dshield.org/
4. http://isc.incidents.org
5. http://www.whitehats.com
6. http://www.snort.org
7. http://www.ietf.org/rfc/
8. http://www.cert.org

Appendix A

http://www.snort.org/docs/FAQ.txt

```
3.1 How do I setup snort on a 'stealth' interface?

*BSD and Linux:  ifconfig eth1 up
Solaris:         ifconfig eth1 plumb; ifconfig eth1 up

NT/W2K/XP:
NOTE: You are at your own risk if you follow these instructions. Editing your
registry is DANGEROUS and should be done with extreme caution. Follow these
steps at your OWN risk.

1.  Get your device's hex value. ('snort -W' works for this)
2.  open Regedt32
3.  Navigate out to: HKEY_LOCAL_MACHINE\( \backslash \)SYSTEM\( \backslash \)
    CurrentControlSet\( \backslash \)Services\( \backslash \)Tcpip\( \backslash
    \)Parameters\( \backslash \)Interfaces\( \backslash \)
    {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}
4.  Select the network card you wish to setup as the monitoring interface (this
    will be the {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX} value).
5.  Set IPAddress:REG_MULTI_SZ: to null (Double click on the string, delete
    data in the Multi-String Editor, then click OK)
6.  Set SubnetMask:REG_MULTI_SZ: to null (Double click on the string, delete
    data in the Multi-String Editor, then click OK)
7.  Set DefaultGateway:REG_MULTI_SZ: to null (Double click on the string,
    delete data in the Multi-String Editor, then click OK)
8.  Close the Registry Editor, your changes will be saved automatically.
9.  In a command prompt, run 'ipconfig' to verify the interface does not have
    an IP bound to it.

If you do not receive an IP address listing from the interface you modified,
you are good to go. To run snort with the specified interface, use the -i flag
such as 'snort -v -d -p -i1'


3.2 How do I setup a receive-only ethernet cable?

The UTP Y-Cable specified by Joe Lyman:
A less noisy option: it involves a couple of cat 5 cables and a single speed
hub. The idea is to use the rcv cables for the wire going to the sniffer box
and use the xmit cables from another hub port. This will give you a link light
and allow your sniffer to rcv only. Cannot xmit because the xmit cables are not
connected. This has been successfully used on netgear single speed hubs. It
wont work on dual speed hubs due to the negotiation of speed.

Pin outs. They are reversed in the picture in order to prevent lines from
crossing, and I only included the pins used.

  * HUB PORT 1     HUB PORT 2
       -------     -------
       x x r r     r r x x
       6 3 2 1     1 2 3 6
       | | | |         | |
       | | | ----------- |
       | | -------------
       |
       | |

       6 3 2 1
       r r x x                   x = xmit, r = rcv
       -------
       SNIFFER
```

Appendix B

Extract from KFS v2.0.2 Educational.pdf document:

KFSensor supports the four Windows networking services using four Sim Standard Servers:

NetBIOS Name Service (NBNS)
UDP 137
Sim Std Server : NBT Name Service

NetBIOS Datagram
UDP 138
Sim Std Server : NBT Datagram Service

NetBIOS Session Service
TCP 139
Sim Std Server : NBT Session Service

SMB Direct
TCP 445
Sim Std Server : NBT SMB

Emulation Type:
Currently there is only one type of emulation, called "Anything Goes". This emulation does not enforce any rules or restrictions, such as insisting a session is set up before a file is opened. It always returns a positive response to any request it receives.

Configuring Windows networking for KFSensor
In order for KFS to act as a honeypot for Windows Networking it needs to listen to the standard ports used by the Windows system. It is not possible to run KFSensor on these ports and let Windows use them at the same time so Windows Networking must be disabled.
Disabling NBT/SMB.
The following instructions are for Windows XP. Windows 2000 and 2003 are nearly identical. Disable NBT:
1. Go to the Control Panel and select Network Connections.
2. Double click on a network icon.
3. Select the Properties button on the General tab.
4. Go to the Networking tab.
5. Uncheck the following two items in the list box.
File and Printer Sharing for Microsoft Networks
Client For Microsoft Networks
6. Click on the "Internet Protocol (TCP/IP)" item and select the Properties button.
7. On the Properties sheet select the Advanced button.
8. Select the WINS tab.
9. In the "NetBIOS setting" box select "Disable NetBIOS over TCP/IP".
10. Press OK three times.
11. Repeat steps 2-10 for each network interface.
12. The NetBIOS Helper Service will record an error in the event log when it attempts to start.
This service can be disabled in the Services windows accessed from the Control Panel.

Disabling SMB Direct

SMB works in a different way to NBT. It binds to all available IP addresses on TCP 445. There is no way of configuring it to work on some network adaptors and not others. It is possible to leave it running while disabling NBT, however to run KFSensor on the complete set of Windows Networking ports it must be disabled.

1. From the Start menu select Run.
2. Enter "regedt32" and click on OK.
3. Expand the tree and select the key: **HKLM\System\CurrentControlSet\Services\NetBT\Parameters**
4. Rename the value "TransportBindName" to "xTransportBindName"
5. Exit regedt32 and re-boot the machine.