



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Advanced Incident Response, Threat Hunting, and Digital Forensics (Forensics
at <http://www.giac.org/registration/gcfa>

If it quacks like a duck, is it really a duck?

SANS GCFA

Practical Assignment v 1.2

26 May 2003

Andrew Hall

© SANS Institute 2003, Author retains full rights.

Abstract

This document is a submission by Andrew Hall for the GCFA practical certification.

This submission is for Assignment version 1.2.

Questions 1, 2b and 3 have been addressed.

© SANS Institute 2003, Author retains full rights.

Table of Contents

Table of Contents.....	3
Introduction	5
Document Standards	5
Part 1 – Analyze and Unknown Binary.....	6
Introduction.....	6
Analysis Methodology	6
Analysis Environment	7
Test and Watch Environments	8
Analysis.....	11
script	11
mount.....	11
unzip	12
unzip	13
stat.....	14
md5sum	15
file	15
readelf.....	16
strings	20
strace.....	23
ps.....	25
netstat.....	25
ethereal.....	26
gcc	26
Strace (again)	27
md5sum (again).....	29
lokid usage.....	29
Summary.....	29
Legal Implications.....	30
Interview Questions.....	33
Additional Information.....	35
Part 2 – Option 2 – Forensic Tool Validation - Ethereal	36
Introduction.....	36
Scope	36
Tool Description	37
History	37
Core Functionality	37
Availability / Usage	41
System Footprint.....	42
Test Apparatus	46
Analysis Objectives.....	47
Criteria for Approval	47
Machine Specifications	48
Environmental Considerations	52
Description of the procedures.....	53

Standards Used	53
Integrity Validation	54
Host Preparation	54
Data Generation	64
Results	66
Analysis	70
Analysis Objective 1	73
Analysis Objective 4	74
Analysis Objective 2	77
Analysis Objective 3	82
Conclusion.....	87
Presentation	88
Part 3 – Legal Issues of Incident Handling.....	93
Question Setting.....	93
Question 1	93
Question 2	99
Question 3	101
Question 4	103
Question 5	104
References	105
Online Documents.....	105
Legislation	106
Other	107

Introduction

This document is a submission by Andrew Hall for the GCFA practical certification.

This submission is for Assignment version 1.2.

Document Standards

There are a variety of standards used in this document; the following is a summary of these standards used:

Commands that have been run on a system are displayed in a small blue font. The outputs of the command are displayed in a small black font. For instance;

```
[root@FIRE] root> mount /mnt/floppy/  
mount: block device /dev/fd0 is write-protected, mounting read-only
```

In order to present the output information in a readable manner, some small formatting to the output display has been undertaken. This may include the consolidation of output, which is indicated by a <snip> sign. It may also include the bolding of output to highlight a certain feature of the output.

No substantive alterations are made to the outputs.

Finally, quotes from outside sources are presented in italics, and referenced via footnotes.

Part 1 – Analyze and Unknown Binary

Introduction

For this element of the practical assignment, we have been given an unknown binary to analyze. The binary has been presented to us in the format of a single ZIP¹ file.

The objectives of this part of the assignment is to analyze this binary, and try to determine information about this file, such as what it does, what it is used for, where it might have come from, and what issues arise from the running of this binary in a managed network environment.

Analysis Methodology

As stated in the assignment specification, the binary was provided without any details, such as its purpose or capabilities. The binary *could* be malicious. This analysis methodology will briefly discuss how to approach such an analysis situation, and to ensure that the following key elements of forensic analysis are maintained:

- the analysis is performed on a trusted and known environment
- the original analysis target (ie the file) is not tainted throughout the investigation
- the analysis target does not taint the analysis environment
- information gathered from the analysis is verified and validated
- the analysis is documented in full, and the procedures taken could be used again to reproduce the same results

Firstly, a sterile analysis environment will be created. This environment will be dedicated solely to the analysis of this unknown binary. The analysis environment will include the following system;

- *Analysis environment* – where the poking and probing of the file occurs
- *Test environment* – where the binary can be safely executed
- *Watch environment* – where we can gather information about the binary as it runs

More information of these environments is detailed below.

¹ <http://www.info-zip.org> (26 May 2003)

The binary will initially be introduced to the analysis environment. The objective here will be to gather as much information about the file, its original permissions, how it was built, where it was built, what it is likely to do, etc.

Based on the information gathered from the analysis environment, the test and watch environment will be specifically tuned to watch and verify predicted and expected information. The objective here will be to verify the initial analysis, and to verify these predictions as to what the binary will do when executed. Finally, the Test and Watch environment test is expected to yield additional information, which was not found in the initial analysis.

As discussed in the assignment specification, correlation with online resources will be made in both the analysis and test phase of the study. It is hoped that source code might be found, which can give a greater chance of mapping exactly what the binary does, and importantly where it came from and why an attacker may use this binary.

Analysis Environment

The analysis environment used will be a purpose built forensics environment called FIRE² - *Forensic Incident Response Environment*.

*"FIRE is a portable bootable cdrom based distribution with the goal of providing an immediate environment to perform forensic analysis, incident response, data recovery, virus scanning and vulnerability assessment."*³

The FIRE CD, version 0.3.5b, provides over 177 different packages for forensic analysis, virus scanning, penetration testing, data recovery and live system analysis. The FIRE CD is bootable, and provides a full X session, so the investigator can use it on most Intel compatible machines and boot into a full forensic environment. See Figure 1 - FIRE boot screen.

FIRE will not mount or touch the local hard drives unless instructed. In this case we will be loading FIRE onto a machine with hard disks installed, but they will not be mounted.

A major advantage of FIRE is that it provides an analysis environment that is the same every time. You can be assured that the binary files which you use and the environment in which you executed those files will be exactly the same next time you boot the CD.

Since it is a fully functional Linux environment you can also enable networking, download and add additional tools etc as required.

² <http://biatchux.dmzs.com/?section=main> (26 May 2003)

³ <http://biatchux.dmzs.com/?section=main> (26 May 2003)

Additional nice features of FIRE are that all xterminal consoles have a built-in timestamp feature on every command that is run. When the script command is also used, a complete history of commands and the date and time run is captured.

A major disadvantage of FIRE is that there are no *man* (manual) pages installed. So, you have to rely on --help options from the utilities themselves. Furthermore, a number of the tools on the CD are not the latest versions, with some tools becoming quite out of date.

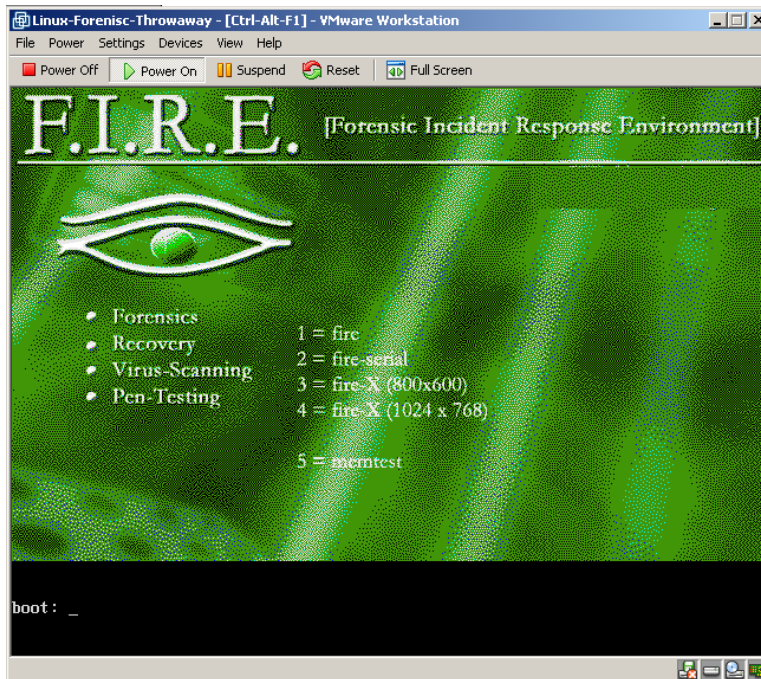


Figure 1 - FIRE boot screen⁴

Test and Watch Environments

The Test environment will make use of VMware⁵. VMware will be installed within the Watch environment.

*"VMware Workstation works by letting multiple operating systems run on physical computers in virtual machines. Operating systems and software applications are isolated in these secure virtual machines that co-exist on a single piece of hardware."*⁶

⁴ Note – FIRE will not present a graphical interface in VMware at present.

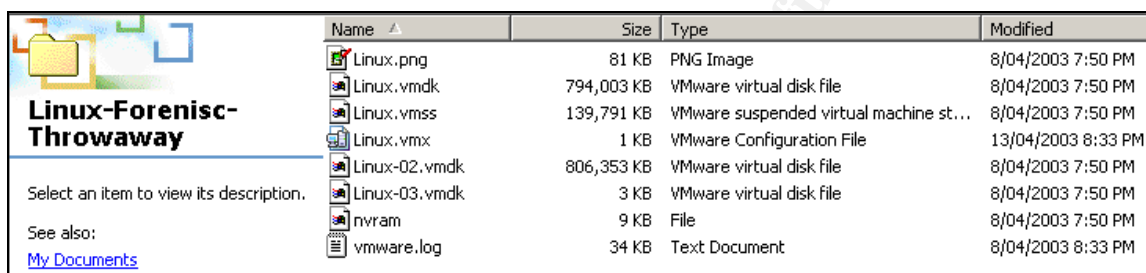
⁵ <http://www.vmware.com> (26 May 2003)

⁶ http://www.vmware.com/products/desktop/ws_features.html (26 May 2003)

When VMware installs an Operating System, the whole install image is stored in a single data file. The advantage of using a VMware installation is that once the install image is created, you can baseline it, and make a copy of the original data files. If your working image is destroyed or becomes too contaminated, you can just copy the original data file back and there is no need to rebuild a whole machine.

If you boot a copy of your original image, you will be guaranteed to have the exact same image every time, you can perform a range of tests and be confident that you have always started with the same baseline.

Figure 2 - VMware *install footprint* shows a sample VMware install, illustrating the files that can be copied for later comparison.



Name	Size	Type	Modified
Linux.png	81 KB	PNG Image	8/04/2003 7:50 PM
Linux.vmdk	794,003 KB	VMware virtual disk file	8/04/2003 7:50 PM
Linux.vmsx	139,791 KB	VMware suspended virtual machine st...	8/04/2003 7:50 PM
Linux.vmx	1 KB	VMware Configuration File	13/04/2003 8:33 PM
Linux-02.vmdk	806,353 KB	VMware virtual disk file	8/04/2003 7:50 PM
Linux-03.vmdk	3 KB	VMware virtual disk file	8/04/2003 7:50 PM
nvram	9 KB	File	8/04/2003 7:50 PM
vmware.log	34 KB	Text Document	8/04/2003 8:33 PM

Figure 2 - VMware install footprint

In this test environment, the guest operating system will be a Linux Red Hat 6.1 and a Linux Red Hat 7.2 installation, and the host operating system will be a Windows 2000 Professional host.

The nature of the VMware install means that neither the guest nor the host operating systems have direct access to the others file systems. As far as each host is concerned, the other is just another host on the local network.

The host operating system will not be connected to a live network.⁷

The *grave_robber*⁸ utility will be run over each test environment, so that full binary md5s are created and stored, as well as outputs of all core processes. This data can then be collected once the binary has been executed, thus allowing for an easy comparison of changed files and system state. The command to execute is the following;

```
[root@localhost bin]# ./grave_robber -E
```

⁷ VMware has the concept of 'Host' and 'Guest' operating systems. The Host operating system is the operating system running VMware. In this case, it is a Windows 2000 Professional machine. The Guest operating system is the system running within VMware, ie the Linux installs

⁸ From The Coroners Tool Kit version 1.09

The `-E` option will collect everything, it will baseline all the files, and obtains the output of more processes than by default. This is especially useful for later comparison when the binary is actually being run. While the `-E` option is comprehensive, it should be noted that this will take a reasonable period of time to run within a VMware session. An example of the extensive output is shown below. The `command_out` directory shows the extensive list of command outputs which are baselined.

```
[root@localhost localhost.localdomain]# ls
```

```
body body.S command_out conf_vault icat MD5_all MD5_all.md5 pcat proc
removed_but_running trust user_vault
```

```
[root@localhost localhost.localdomain]# cd command_out/
```

```
[root@localhost command_out]# ls (formatting changed for display purposes)
```

```
arp                arp.md5
lastcomm           lastcomm.md5
netstat-a          netstat-a.md5
netstat-in         netstat-in.md5
netstat-rn         netstat-rn.md5
netstat-na         netstat-na.md5
rpcinfo            rpcinfo.md5
uptime            uptime.md5
last               last.md5
rpm                rpm.md5
df                 df.md5
lsmod              lsmod.md5
w                  w.md5
ifconfig           ifconfig.md5
who                who.md5
dmesg              dmesg.md5
lsof               lsof.md5
showmount-a        showmount-a.md5
ipcs               ipcs.md5
lsof0              lsof0.md5
showmount-e        showmount-e.md5
finger             finger.md5

nfsstat            nfsstat.md5
ksyms              ksyms.md5
top                top.md5
major_minor        major_minor.md5
ps                 ps.md5
uname              uname.md5
free_inode_info._dev_sda1  free_inode_info._dev_sda1.md5
free_inode_info._dev_sda2  free_inode_info._dev_sda2.md5
free_inode_info._dev_cdrom  free_inode_info._dev_cdrom.md5
```

During the analysis, the `script` command will also be used. This is a standard unix utility which will capture all user input and output within a shell. This allows for full input/output captures of the commands run on the Test system.

Finally, it should be noted that backups of the original ZIP file have been made and have been stored on permanent media (ie a CD), and this will be used as a baseline file for later comparisons.

Analysis

script

As mentioned above, the script command will be used to keep track of commands and outputs during analysis.

```
[root@FIRE] root> script
Script started, file is transcript
[Sun Mar 9 11:22:03]
```

This command will cause all the input and output within this shell to be placed in a file called typescript. The data will populate into the file when the script shell is exited, i.e. CTRL-C or exit.

A word of warning ... Script will capture everything typed into the shell, including typos, tabs etc. This can make it quite difficult to read in MS Word! So be careful with typing and using bash tab competes. For the sake of easier reading, white space characters and typos have been removed / fixed in this document.

Notice that after the execution of each command, there is a date and time stamp. This is part of the standard shells within FIRE.

mount

```
[root@FIRE] root> mount /mnt/floppy/
mount: block device /dev/fd0 is write-protected, mounting read-only
[Sun Mar 9 11:23:40]
```

```
[root@FIRE] root> ls /mnt/floppy/
binary_v1.2.zip
[Sun Mar 9 11:23:44]
```

The *binary_v1.2.zip* has been previously downloaded onto a floppy disk, and this disk has been inserted into the analysis machine. The first thing we need to do is mount the floppy.

The physical read-only tab has been set on the floppy disk, so the disk has been automatically mounted read-only.

The listing command just makes sure that the mount is successful and that we have the right disk in the drive.

unzip

The next thing to do is to inspect the zip file to see what its contents are. To do this use the `unzip -Zv` command. The `-Z` option is equivalent to running `zipinfo`. This gives such information as names, dates/times, attributes, size, compression method etc about files contained in the zip archive.

The `-v` option gives a verbose output.

The relevant information will be bolded in this output.

```
[root@FIRE] root> unzip -Zv /mnt/floppy/binary_v1.2.zip
```

```
Archive: /mnt/floppy/binary_v1.2.zip 7309 bytes 2 files
```

```
End-of-central-directory record:
```

```
-----
```

```
Actual offset of end-of-central-dir record: 7287 (00001C77h)
Expected offset of end-of-central-dir record: 7287 (00001C77h)
(based on the length of the central directory and its expected offset)
```

This zipfile constitutes the sole disk of a single-part archive; its central directory contains 2 entries. The central directory is 102 (00000066h) bytes long, and its (expected) offset in bytes from the beginning of the zipfile is 7185 (00001C11h).

There is no zipfile comment.

```
Central directory entry #1:
```

```
-----
```

atd.md5

```
offset of local header from start of archive: 0 (00000000h) bytes
file system or operating system of origin: MS-DOS, OS/2 or NT FAT
version of encoding software: 2.0
minimum file system compatibility required: MS-DOS, OS/2 or NT FAT
minimum software version required to extract: 2.0
compression method: deflated
compression sub-type (deflation): normal
file security status: not encrypted
extended local header: no
file last modified on (DOS date/time): 2002 Aug 22 14:58:08
32-bit CRC value (hex): e5376cb4
compressed size: 38 bytes
uncompressed size: 39 bytes
length of filename: 7 characters
length of extra field: 0 bytes
length of file comment: 0 characters
disk number on which file begins: disk 1
apparent file type: text
```

non-MSDOS external file attributes: 81B600 hex
MS-DOS file attributes (20 hex): arc

There is no file comment.

Central directory entry #2:

atd

offset of local header from start of archive: 75 (0000004Bh) bytes
file system or operating system of origin: MS-DOS, OS/2 or NT FAT
version of encoding software: 2.0
minimum file system compatibility required: MS-DOS, OS/2 or NT FAT
minimum software version required to extract: 2.0
compression method: deflated
compression sub-type (deflation): normal
file security status: not encrypted
extended local header: no
file last modified on (DOS date/time): 2002 Aug 22 14:57:54
32-bit CRC value (hex): d0ee3072
compressed size: 7077 bytes
uncompressed size: 15348 bytes
length of filename: 3 characters
length of extra field: 0 bytes
length of file comment: 0 characters
disk number on which file begins: disk 1
apparent file type: binary
non-MSDOS external file attributes: 81B600 hex
MS-DOS file attributes (20 hex): arc

There is no file comment.

[Sun Mar 9 11:23:56]

So from this command we know now that the zip file contains two files, atd and atd.md5. We will assume that the file for analysis is the atd file.

The atd file appears to have been zipped on a FAT filesystem (mostly used by the Microsoft Windows operating system), with version 2.0 of zip. The file was deflated with no encryption.

The last modification time of the atd file was the 22nd August 2002 at 14:57:54, and the atd file is probably a binary. The uncompressed size of the binary is 15348 bytes.

unzip

Now that we know a little more about the files in the zip, we can extract the files with the unzip utility.

```
[root@FIRE] root> unzip -X /mnt/floppy/binary_v1.2.zip
Archive: /mnt/floppy/binary_v1.2.zip
  inflating: atd.md5
  inflating: atd
[Sun Mar 9 11:24:09]
```

The `-X` options restores owner information (i.e. UID and GIDs) under Unix systems. This tool can also be used to restore permissions under NT, VMS and OS/2 systems.

stat

Before we list (`ls`) the extracted files we will run the `stat` command to find out more specific information about the `atd` file. If we were to run an `ls` now, we would lose the access time MAC attribute.

```
[root@FIRE] root> stat atd
File: "atd"
Size: 15348          Blocks: 32      IO Block: 4612079248453144576 Regular File
Device: 100h/256d    Inode: 7245     Links: 1
Access: (0666/-rw-rw-rw-) Uid: ( 0/  root)  Gid: ( 0/  root)
Access: Thu Aug 22 14:57:54 2002
Modify: Thu Aug 22 14:57:54 2002
Change: Sun Mar 9 11:24:09 2003
```

```
[Sun Mar 9 11:24:21]
```

This output gives the following information about the `atd` file;

The file size is confirmed to be 15348 bytes, and had permissions on read and write to all when it was compressed. The owner of the file indicates root as the owner (i.e. uid = gid = 0). Given that the filesystem which this file was compressed on was the FAT filesystem, the owner of the file is not relevant. FAT file system properties only allow for read only, archive, system, and hidden properties⁹, so the root owner here is that of the unzipper – me!

Of more interest are the MAC times. This indicates that the file was last accessed and modified on the *22nd August 2002 at 14:57:54*, and changed on *9th March 2003 at 11:24:09*. The August date indicates the date when the file was zipped up by the first analyst¹⁰, while the March date shows the time when it was unzipped to my analysis machine.

⁹ The Computer Technology Documentation Project "Windows NT Workstation Reference Version 0.6.0 1 Dec 2000 URL:

<http://www.comptechdoc.org/os/windows/ntwsguide/ntwsppermissions.html> (26 May 2003)

¹⁰ It is assumed that the original analyst who collected and zipped this file had the correct time on their machine.

Although these dates do not help in identifying the last time a hacker may have run the file, it does show us how old the atd file is. If we are to look for possible sources for this file, we know that we have to look to files that are at least 8 months old.

md5sum

Now that we have got fragile information from the file, we can run other commands that will destroy some of this original information. By running the *md5sum* command to validate the binary with the supplied checksum, we will change the Access time on the file.

```
[root@FIRE] root> md5sum atd
48e8e8ed3052cbf637e638fa82bdc566 atd
[Sun Mar 9 11:24:28]
```

```
[root@FIRE] root> cat atd.md5
48e8e8ed3052cbf637e638fa82bdc566 atd
[Sun Mar 9 11:24:35]
```

We can see that the md5sums do match, and we can be confident that this is the correct file to analyze.

file

Previously, the *zipinfo* (*unzip -Z*) output indicated that the atd file was a binary. We can use the file command to elaborate on this.

```
[root@FIRE] root> file atd
atd: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared
libs), stripped
[Sun Mar 9 11:24:46]
```

The output can be interrupted as follows;

The atd binary is a Linux 32-bit executable, designed for Intel 80386 architecture (ie a standard PC).

It uses dynamically linked libraries, such that it requires specific libraries to be installed on the machine were it is executed. This could make it more difficult to run, since it cannot operate in a standalone manner without these library dependencies.

The stripped attribute indicates that the `strip`¹¹ command has probably been run over this file, which removes symbols from the object file. The removal of symbols makes it more difficult to analyze the binary, as there is less information to discover. The removal of symbols also has the effect of making the executable much smaller¹².

readelf

Given that the binary is dynamically compiled, we can look further into the binary to see what it links with (ie the libraries it makes calls to). A useful utility to achieve this is the `readelf` tool. Unfortunately this tool is not on the FIRE forensics CD. Instead this command will be run in the Red Hat 7.3 Test environment on the binary.

```
[root@localhost analysis]# readelf -a atd
```

ELF Header:

```
Magic: 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
Class: ELF32
Data: 2's complement, little endian
Version: 1 (current)
OS/ABI: UNIX - System V
ABI Version: 0
Type: EXEC (Executable file)
Machine: Intel 80386
Version: 0x1
Entry point address: 0x8048db0
Start of program headers: 52 (bytes into file)
Start of section headers: 14508 (bytes into file)
Flags: 0x0
Size of this header: 52 (bytes)
Size of program headers: 32 (bytes)
Number of program headers: 5
Size of section headers: 40 (bytes)
Number of section headers: 21
Section header string table index: 20
```

<snip>

Dynamic segment at offset 0x3644 contains 17 entries:

Tag	Type	Name/Value
0x00000001	(NEEDED)	Shared library: [libc.so.5]
0x0000000c	(INIT)	0x8048a70
0x0000000d	(FINI)	0x804a8e0
0x00000004	(HASH)	0x80480e8
0x00000005	(STRTAB)	0x80486ac
0x00000006	(SYMTAB)	0x804828c

¹¹ "GNU Binary Utilities" URL: http://www.gnu.org/manual/binutils-2.12/html_mono/binutils.html#SEC10 (26 May 2003)

¹² "Forensics FAQ"

URL: <http://www.deaddrop.org/security/Presentations/2ndqtr/ForensicsFaq.html> (26 May 2003)

0x0000000a (STRSZ)	528 (bytes)
0x0000000b (SYMENT)	16 (bytes)
0x00000015 (DEBUG)	0x0
0x00000003 (PLTGOT)	0x804c570
0x00000002 (PLTRELSZ)	400 (bytes)
0x00000014 (PLTREL)	REL
0x00000017 (JMPREL)	0x80488dc
0x00000011 (REL)	0x80488bc
0x00000012 (RELSZ)	32 (bytes)
0x00000013 (RELENT)	8 (bytes)
0x00000000 (NULL)	0x0

<snip>

Relocation section '.rel.plt' at offset 0x8dc contains 50 entries:

Offset	Info	Type	Symbol's Value	Symbol's Name
0804c57c	00000107	R_386_JUMP_SLOT	08048a88	longjmp
0804c580	00000207	R_386_JUMP_SLOT	08048a98	strcpy
0804c584	00000307	R_386_JUMP_SLOT	08048aa8	ioctl
0804c588	00000407	R_386_JUMP_SLOT	08048ab8	popen
0804c58c	00000507	R_386_JUMP_SLOT	08048ac8	shmctl
0804c590	00000607	R_386_JUMP_SLOT	08048ad8	geteuid
0804c594	00000807	R_386_JUMP_SLOT	08048ae8	getprotobyndnumber
0804c598	00000a07	R_386_JUMP_SLOT	08048af8	__strtoul_internal
0804c59c	00000b07	R_386_JUMP_SLOT	08048b08	usleep
0804c5a0	00000c07	R_386_JUMP_SLOT	08048b18	semget
0804c5a4	00000d07	R_386_JUMP_SLOT	08048b28	getpid
0804c5a8	00000e07	R_386_JUMP_SLOT	08048b38	fgets
0804c5ac	00000f07	R_386_JUMP_SLOT	08048b48	shmat
0804c5b0	00001107	R_386_JUMP_SLOT	08048b58	perror
0804c5b4	00001207	R_386_JUMP_SLOT	08048b68	getuid
0804c5b8	00001307	R_386_JUMP_SLOT	08048b78	semctl
0804c5bc	00001507	R_386_JUMP_SLOT	08048b88	socket
0804c5c0	00001707	R_386_JUMP_SLOT	08048b98	bzero
0804c5c4	00001907	R_386_JUMP_SLOT	08048ba8	alarm
0804c5c8	00001a07	R_386_JUMP_SLOT	08048bb8	__libc_init
0804c5cc	00001c07	R_386_JUMP_SLOT	08048bc8	fprintf
0804c5d0	00001d07	R_386_JUMP_SLOT	08048bd8	kill
0804c5d4	00001e07	R_386_JUMP_SLOT	08048be8	inet_addr
0804c5d8	00001f07	R_386_JUMP_SLOT	08048bf8	chdir
0804c5dc	00002007	R_386_JUMP_SLOT	08048c08	shmdt
0804c5e0	00002107	R_386_JUMP_SLOT	08048c18	setsockopt
0804c5e4	00002307	R_386_JUMP_SLOT	08048c28	shmget
0804c5e8	00002407	R_386_JUMP_SLOT	08048c38	wait
0804c5ec	00002507	R_386_JUMP_SLOT	08048c48	umask
0804c5f0	00002607	R_386_JUMP_SLOT	08048c58	signal
0804c5f4	00002707	R_386_JUMP_SLOT	08048c68	read
0804c5f8	00002807	R_386_JUMP_SLOT	08048c78	strncmp
0804c5fc	00002907	R_386_JUMP_SLOT	08048c88	sendto
0804c600	00002a07	R_386_JUMP_SLOT	08048c98	bcopy
0804c604	00002b07	R_386_JUMP_SLOT	08048ca8	fork
0804c608	00002c07	R_386_JUMP_SLOT	08048cb8	strdup
0804c60c	00002d07	R_386_JUMP_SLOT	08048cc8	getopt
0804c610	00002e07	R_386_JUMP_SLOT	08048cd8	inet_ntoa
0804c614	00002f07	R_386_JUMP_SLOT	08048ce8	getppid
0804c618	00003007	R_386_JUMP_SLOT	08048cf8	time

0804c61c	00003107	R_386_JUMP_SLOT	08048d08	gethostbyname
0804c620	00003307	R_386_JUMP_SLOT	08048d18	sprintf
0804c624	00003407	R_386_JUMP_SLOT	08048d28	difftime
0804c628	00003507	R_386_JUMP_SLOT	08048d38	atexit
0804c62c	00003707	R_386_JUMP_SLOT	08048d48	semop
0804c630	00003807	R_386_JUMP_SLOT	08048d58	exit
0804c634	00003907	R_386_JUMP_SLOT	08048d68	__setfpucw
0804c638	00003a07	R_386_JUMP_SLOT	08048d78	open
0804c63c	00003b07	R_386_JUMP_SLOT	08048d88	setsid
0804c640	00003c07	R_386_JUMP_SLOT	08048d98	close

There are no unwind sections in this file.

Symbol table '.dynsym' contains 66 entries:

Num:	Value	Size	Type	Bind	Vis	Ndx	Name
0:	00000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	08048a88	0	FUNC	GLOBAL	DEFAULT	UND	longjmp
2:	08048a98	30	FUNC	GLOBAL	DEFAULT	UND	strcpy
3:	08048aa8	0	FUNC	WEAK	DEFAULT	UND	ioctl
4:	08048ab8	0	FUNC	WEAK	DEFAULT	UND	popen
5:	08048ac8	42	FUNC	GLOBAL	DEFAULT	UND	shmctl
6:	08048ad8	0	FUNC	WEAK	DEFAULT	UND	geteuid
7:	0804c644	0	OBJECT	GLOBAL	DEFAULT	ABS	_DYNAMIC
8:	08048ae8	292	FUNC	GLOBAL	DEFAULT	UND	getprotobyname
9:	0804c6d0	4	NOTYPE	WEAK	DEFAULT	17	errno
10:	08048af8	1132	FUNC	GLOBAL	DEFAULT	UND	__strtoul_internal
11:	08048b08	99	FUNC	GLOBAL	DEFAULT	UND	usleep
12:	08048b18	42	FUNC	GLOBAL	DEFAULT	UND	semget
13:	08048b28	0	FUNC	WEAK	DEFAULT	UND	getpid
14:	08048b38	0	FUNC	WEAK	DEFAULT	UND	fgets
15:	08048b48	59	FUNC	GLOBAL	DEFAULT	UND	shmat
16:	0804c6d8	84	OBJECT	GLOBAL	DEFAULT	17	_IO_stderr_
17:	08048b58	0	FUNC	WEAK	DEFAULT	UND	perror
18:	08048b68	0	FUNC	WEAK	DEFAULT	UND	getuid
19:	08048b78	47	FUNC	GLOBAL	DEFAULT	UND	semctl
20:	0804c72c	4	OBJECT	GLOBAL	DEFAULT	17	optarg
21:	08048b88	94	FUNC	WEAK	DEFAULT	UND	socket
22:	0804c528	4	OBJECT	GLOBAL	DEFAULT	12	__environ
23:	08048b98	54	FUNC	GLOBAL	DEFAULT	UND	bzero
24:	08048a70	0	FUNC	GLOBAL	DEFAULT	7	_init
25:	08048ba8	0	FUNC	WEAK	DEFAULT	UND	alarm
26:	08048bb8	70	FUNC	GLOBAL	DEFAULT	UND	__libc_init
27:	0804c528	4	NOTYPE	WEAK	DEFAULT	12	environ
28:	08048bc8	0	FUNC	WEAK	DEFAULT	UND	fprintf
29:	08048bd8	0	FUNC	WEAK	DEFAULT	UND	kill
30:	08048be8	57	FUNC	GLOBAL	DEFAULT	UND	inet_addr
31:	08048bf8	0	FUNC	WEAK	DEFAULT	UND	chdir
32:	08048c08	36	FUNC	GLOBAL	DEFAULT	UND	shmdt
33:	08048c18	111	FUNC	WEAK	DEFAULT	UND	setsockopt
34:	0804c730	2	OBJECT	GLOBAL	DEFAULT	17	__fpu_control
35:	08048c28	42	FUNC	GLOBAL	DEFAULT	UND	shmget
36:	08048c38	0	FUNC	WEAK	DEFAULT	UND	wait
37:	08048c48	0	FUNC	WEAK	DEFAULT	UND	umask
38:	08048c58	84	FUNC	GLOBAL	DEFAULT	UND	signal
39:	08048c68	0	FUNC	WEAK	DEFAULT	UND	read

```

40: 08048c78 38 FUNC GLOBAL DEFAULT UND strncmp
41: 08048c88 124 FUNC WEAK DEFAULT UND sendto

42: 08048c98 146 FUNC GLOBAL DEFAULT UND bcopy
43: 08048ca8 0 FUNC WEAK DEFAULT UND fork
44: 08048cb8 79 FUNC GLOBAL DEFAULT UND strdup
45: 08048cc8 44 FUNC GLOBAL DEFAULT UND getopt
46: 08048cd8 67 FUNC GLOBAL DEFAULT UND inet_ntoa
47: 08048ce8 0 FUNC WEAK DEFAULT UND getppid
48: 08048cf8 0 FUNC WEAK DEFAULT UND time
49: 08048d08 292 FUNC GLOBAL DEFAULT UND gethostbyname
50: 08048ae0 0 FUNC GLOBAL DEFAULT 10_fini
51: 08048d18 38 FUNC WEAK DEFAULT UND sprintf
52: 08048d28 16 FUNC GLOBAL DEFAULT UND difftime
53: 08048d38 52 FUNC GLOBAL DEFAULT UND atexit
54: 0804c570 0 OBJECT GLOBAL DEFAULT ABS _GLOBAL_OFFSET_TABLE_
55: 08048d48 42 FUNC GLOBAL DEFAULT UND semop
56: 08048d58 128 FUNC GLOBAL DEFAULT UND exit
57: 08048d68 62 FUNC GLOBAL DEFAULT UND __setfpucw
58: 08048d78 0 FUNC WEAK DEFAULT UND open
59: 08048d88 0 FUNC WEAK DEFAULT UND setsid
60: 08048d98 0 FUNC WEAK DEFAULT UND close
61: 0804c6d0 4 OBJECT GLOBAL DEFAULT 17_errno
62: 0804a8d8 0 OBJECT GLOBAL DEFAULT ABS _etext
63: 0804c6cc 0 OBJECT GLOBAL DEFAULT ABS _edata
64: 0804c6cc 0 OBJECT GLOBAL DEFAULT ABS __bss_start
65: 0804c7f8 0 OBJECT GLOBAL DEFAULT ABS _end

```

Histogram for bucket list length (total of 37 buckets):

Length	Number	% of total	Coverage
0	9	(24.3%)	
1	8	(21.6%)	12.3%
2	10	(27.0%)	43.1%
3	4	(10.8%)	61.5%
4	5	(13.5%)	92.3%
5	1	(2.7%)	100.0%

No version information found in this file.

From this output we can gather the following information;

The entry point of the file is normal, ie around 0x804, and the shared library is libc.so.5. We should have no problems linking to this library if we want to test run the binary, since it is a common library.

Also of interest are the symbols listed. The symbols indicate that this binary must do something to a network, or make use of a network. For instance the symbols of *gethostbyname*, *setsockopt*, *socket*, *inet_addr* and *getprotobyname* are common networking functions. The use of *open*, and *close* calls indicate that sockets are been created, and that data packets are being crafted with the help of *strcpy*.

strings

To find out further information about the internal of the binary we can use the *strings* command. This command outputs strings of four (4) or more printable characters from a file. When running strings over a binary, it is hoped that information will be found which indicates the purpose of the file, such as help or usage information, and maybe even find information about its origins.

```
[root@FIRE] root> strings -a atd > atd.strings.out  
[Sun Mar 9 11:24:58]
```

The following is a list of the more useful elements of the strings output;

```
/lib/ld-linux.so.1  
libc.so.5  
...  
getprotobyname  
...  
socket  
...  
setsockopt  
...  
inet_ntoa  
...  
gethostbyname  
...  
lokid: Client database full  
DEBUG: stat_client nono  
lokid version:      %s  
remote interface:  %s  
active transport: %s  
active cryptography: %s  
server uptime:     %.02f minutes  
client ID:         %d  
packets written: %ld  
bytes written:     %ld  
requests:         %d  
N@[fatal] cannot catch SIGALRM  
lokid: inactive client <%d> expired from list [%d]  
...  
/dev/tty  
[fatal] cannot detach from controlling terminal  
/tmp  
[fatal] invalid user identification value  
...  
Unknown transport  
lokid -p (i|u) [ -v (0|1) ]  
[fatal] socket allocation error  
[fatal] cannot catch SIGUSR1  
Cannot set IP_HDRINCL socket option  
[fatal] cannot register with atexit(2)  
LOKI2 route [(c) 1997 guild corporation worldwide]  
[fatal] cannot catch SIGALRM
```

```

[fatal] cannot catch SIGCHLD
[SUPER fatal] control should NEVER fall here
[fatal] forking error
lokid: server is currently at capacity. Try again later
lokid: Cannot add key
lokid: popen
[non fatal] truncated write
/quit all
lokid: client <%d> requested an all kill
      sending L_QUIT: <%d> %s
lokid: clean exit (killed at client request)
[fatal] could not signal process group
/quit
lokid: cannot locate client entry in database
lokid: client <%d> freed from list [%d]
/stat
/swapt
[fatal] could not signal parent
lokid: unsupported or unknown command string
lokid: client <%d> requested a protocol swap
      sending protocol update: <%d> %s [%d]
lokid: transport protocol changed to %s
GCC: (GNU) 2.7.2.1
...

```

From this output we can establish the following about the file;

- we again validate that the binary links with libc.so.5, and that this is binary utilizes the network with functions of *socket*, *getprotobyname*, *inet_ntoa*, *gethostbyname*.
- we also get a name to the binary – *lokid*.

Lokid is a well known covert channel utility, published in the online publication *Phrack*¹³. In the words of Phrack,

“The concept of the Loki Project is simple: arbitrary information tunneling in the data portion of ICMP_ECHO and ICMP_ECHOREPLY packets. Loki exploits the covert channel that exists inside of ICMP_ECHO traffic. This channel exists because network devices do not filter the contents of ICMP_ECHO traffic. They simply pass them, drop them, or return them. The trojan packets themselves are masqueraded as common ICMP_ECHO traffic. We can encapsulate (tunnel) any information we want. (Astute readers will note that Loki is simply a form of steganography).”

Given how well known and recognized the Loki concept and Loki2¹⁴ program are, there are many network signatures¹⁵ for their activity and source code exists. It

¹³ Phrack Magazine, Volume Seven, Issue Forty-Nine, “Project Loki” URL: <http://www.phrack.org/show.php?p=49&a=6> (26 May 2003)

should be relative simple to validate that this binaries activity (when executed) matches that of the known Loki implementations.

To continue the analysis of the strings output, we should consider the following information;

- *lokid: inactive client <%d> expired from list [%d]* – Looks like this binary does connect to clients
- *lokid -p (i|u) [-v (0|1)]* – Looks like a usage command, indicating multiple versions are supported
- *LOKI2 route [(c) 1997 guild corporation worldwide]* – This may assist in determining the source of the code
- *[SUPER fatal] control should NEVER fall here* – Looks like the author has a sense of humour. It also gives us something unique to look for in a source code base.
- *lokid: Cannot add key* – Perhaps this client / server implementation supports key based authentication
- *lokid: client <%d> requested a protocol swap* – Perhaps this implementation allows for the underlying protocol to be changed on the fly.
- *GCC: (GNU) 2.7.2.1* – The version of GCC used to compile the code. This will become useful later when trying to match up a compiled version of the code with the given binary. Each version of GCC will result in changes in the compiled code binary.

The strings analysis has been quite useful, yielding much information. The information gathered here will allow for some investigation to take place as to the expected behaviour of the program before we actually run and debug it. This will allow for the configuration of the test environment to gather the most information about the running binary.

The two obvious references on the Loki and Loki2 projects are from *Phrack* itself, <http://www.phrack.org/show.php?p=49&a=6> and <http://www.phrack.org/show.php?p=51&a=6>

From the source code provided on *Phrack* for Loki2, we find a large number of pattern matches from the strings output and with the source code. Some examples are listed below;

Phrack Source Code	atd strings Ouput
#define L_MSG_BANNER "\nLOKI2\troute [(c) 1997 guild corporation worldwide]\n"	LOKI2 route [(c) 1997 guild corporation worldwide]
#define S_MSG_USAGE "\nlokid -p (i u) [-v (0 1)]\n"	lokid -p (i u) [-v (0 1)]

¹⁴ Phrack Magazine, Volume Seven, Issue Fifty-One, "Loki2 - the implementation", 01 Sept 1997
URL: <http://www.phrack.org/show.php?p=51&a=6> (26 May 2003)

¹⁵ Internet Security Systems, X-Force Database, "loki (1452)" URL:
http://www.iss.net/security_center/static/1452.php (26 May 2003)

<code>err_exit(1, 0, verbose, "\nlokid: Cannot add key\n");</code>	lokid: Cannot add key
<code>#define S_MSG_PACKED "\nlokid: server is currently at capacity. Try again later\n"</code>	lokid: server is currently at capacity. Try again later
<code>#define WORKING_ROOT "/tmp"</code>	/tmp
<code>#define L_MSG_NOPRIV "\n[fatal] invalid user identification value"</code>	[fatal] invalid user identification value
<code>#define L_MSG_WIERDERR "\n[SUPER fatal] control should NEVER fall here\n"</code>	[SUPER fatal] control should NEVER fall here
<code>if (verbose) fprintf(stderr, "lokid: client <%d> requested an all kill\n", c_id);</code>	lokid: client <%d> requested an all kill
<code>#define S_MSG_UNKNOWN "\nlokid: cannot locate client entry in database\n"</code>	lokid: cannot locate client entry in database
<code>if (verbose) fprintf(stderr, "\nloki: Transport protocol changed to %s.\n", pprot -> p_name);</code>	lokid: transport protocol changed to %s
<code>if (verbose) fprintf(stderr, "\nlokid: client <%d> requested a protocol swap\n", c_id);</code>	lokid: client <%d> requested a protocol swap

Given the similarities, we can be reasonably confident that the atd binary is closely related to the Loki2 *Phrack* code. It is highly likely that the atd binary will behave similar to the Loki2 program. With this in mind we will proceed with a test of the binary. However, it should be noted that the atd binary may have other Trojan elements in it, or it might be a modified version of the Loki2 implementation, so we can not assume that it is just the *Phrack* Loki2 program.

strace

We will run the binary on the Red Hat 6.1 Test Installation. This old version of Red Hat already contains libc5 libraries, which is what the binary will require to run. The atd binary has been transferred to the test host via floppy disk.

Note that running the binary will change the state of the Test system. However, the original VMware data files for the install have been copied elsewhere, so the original state can be restored. Also, the *grave_robber* tool has also been used to baseline the system, so any changes made the execution of the binary will be easily found.

```
[root@localhost /tmp]# uname -a
```

```
Linux localhost.localdomain 2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000 i686 unknown
```

In order to run the binary, we will need to make it executable.

```
[root@localhost /tmp]# chmod u+x atd
```

The *strace* program is now used to trace the execution of the program. *strace* will output to standard error all the system calls which a running program makes. Using this it is possible to see exactly what the running process is doing. The output has been clipped to show the more relevant output.


```
[root@localhost /tmp]# strace ./atd
execve("./atd", ["/atd"], [/* 19 vars */]) = 0
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x40007000
mprotect(0x40000000, 21772, PROT_READ|PROT_WRITE|PROT_EXEC) = 0
mprotect(0x8048000, 13604, PROT_READ|PROT_WRITE|PROT_EXEC) = 0
stat("/etc/ld.so.cache", {st_mode=S_IFREG|0644, st_size=10936, ...}) = 0
open("/etc/ld.so.cache", O_RDONLY) = 3
```

<snip>

```
socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 3
sigaction(SIGUSR1, {0x804a6b0, [], SA_INTERRUPT|SA_NOMASK|SA_ONESHOT},
{SIG_DFL}, 0x4003ac68) = 0
socket(PF_INET, SOCK_RAW, IPPROTO_RAW) = 4
setsockopt(4, SOL_IP, IP_HDRINCL, [1], 4) = 0
getpid() = 628
getpid() = 628
shmget(870, 240, IPC_CREAT|0) = 6
semget(1052, 1, IPC_CREAT|0x180|0600) = 6
shmat(6, 0, 0) = 0x40008000
write(2, "\nLOKI2\troute [(c) 1997 guild cor"..., 52
LOKI2 route [(c) 1997 guild corporation worldwide]
) = 52
time([1050598690]) = 1050598690
close(0) = 0
sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}, 0x4003ac68) = 0
sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}, 0x4003ac68) = 0
sigaction(SIGTSTP, {SIG_IGN}, {SIG_DFL}, 0x4003ac68) = 0
fork() = 629
close(4) = 0
close(3) = 0
semop(6, 0xbffffb0c, 2) = 0
shmdt(0x40008000) = 0
semop(6, 0xbffffb0c, 1) = 0
_exit(0) = ?
```

The output appears consistent with what the *Phrack* article and the binary code we have so far reviewed. The *strace* shows the following;

That an *ICMP* and *RAW* socket were created, with the program running under pid¹⁶ number 628. The “LOKI2 route [(c) 1997 guild corporation worldwide]” string is printed to standard out. The process is then forked, with the child process being under pid 629. The first process then finishes.

The effect of this is that the machine now has two new listening sockets for which a connection could be made to.

¹⁶ Process ID number

ps

We can confirm that the forked process is still running under the above mentioned pid.

```
[root@localhost /tmp]# ps -aux
```

```
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
<snip>
```

```
root      629  0.0  0.2   868   308 ?        S    02:45   0:00 ./atd
```

```
<snip>
```

netstat

Earlier research¹⁷ showed a known way of detecting an installed Loki server;

"To determine if LOKI is running, look for programs that have an ICMP raw socket open. This can be done from a root shell on Linux with a command similar to: "netstat -a -n -w" If you see something like this:

Active Internet connections (including servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

*raw 0 0 0.0.0.0:1 0.0.0.0:**

*raw 0 0 0.0.0.0:1 0.0.0.0:**

*raw 0 0 0.0.0.0:255 0.0.0.0:**

*Some process has an ICMP raw socket open on the system, which might be indicative of a LOKI daemon. Also look for 0.0.0.0:17, which might indicate a loki daemon running in UDP mode. For Solaris, the command would be netstat -a -n -P icmp."*¹⁸

```
[root@localhost /tmp]# netstat -an
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:25	0.0.0.0:*	LISTEN
raw	0	0	0.0.0.0:255	0.0.0.0:*	7
raw	0	0	0.0.0.0:1	0.0.0.0:*	7
raw	0	0	0.0.0.0:1	0.0.0.0:*	7
raw	0	0	0.0.0.0:6	0.0.0.0:*	7

```
<snip>
```

¹⁷ Internet Security Systems, X-Force Database, "loki (1452)" URL: http://www.iss.net/security_center/static/1452.php (26 May 2003)

¹⁸ Internet Security Systems, X-Force Database, "loki (1452)" URL: http://www.iss.net/security_center/static/1452.php (26 May 2003)

Clearly we have a match of the two new raw sockets. The baseline of *netstat* from the original install is shown below;

tcp	0	0 0.0.0.0:25	0.0.0.0:*	LISTEN
raw	0	0 0.0.0.0:1	0.0.0.0:*	7
raw	0	0 0.0.0.0:6	0.0.0.0:*	7

The 0.0.0.0:1 line indicates that an ICMP listener has been created, and the 0.0.0.0:255 indicates that ip protocol 255 also has a listener. This protocol number is listed by the IANA is reserved.

Note that it appears that by default Red Hat Linux 6.1 has an ICMP listener displayed in netstat, hence the two lines of ICMP listeners.

ethereal

Since the VMware guest operating system is running within the host or watch operating system, it is possible to capture network traffic that is travelling to and from the guest operating system. A useful utility for capturing this traffic is *ethereal*.

Ethereal is a libpcap based utility which presents a graphical interface to network traffic, similar to *tcpdump*.

During the time which atd was running, no network traffic was witnessed coming from the VMware host. The atd process does not appear to have attempted to “phone home” or initiate any network activity at the time or running the binary.

gcc

Since the source code for Loki is still provided by *Phrack*, it is possible to compile the code and run it. By running the code, the various system changes and library calls can be tracked to see if the original Loki program behaves the same as the atd binary.

Although it is highly unlikely that the MD5 hash on our Loki compilation will match that of the atd md5sum, elements such as the *strace* output and system changes (such as reported by *netstat*) should give a solid indication as to whether these binaries are essentially the same code.

To compile the Phrack code, the following commands were used with the RedHat 7.3 installation.

The `extract.c`¹⁹ file was downloaded and compiled.

```
[root@localhost analysis]# gcc -Wall extract.c -o extract.o
```

The Loki program was then compiled, it should be noted that a few minor changes were required to compile the code²⁰, namely the following;

In `loki.h`, added `#include<linux/types.h>`

In `loki.h`, removed `#include<linux/signal.h>`

The Makefile was modified so that the `NO_CRYPT` was enabled. This was done so that there are fewer libraries to link against, which may increase the chances of a successful compilation.

```
[root@localhost analysis]# make linux
```

The state of the directory looked as follows after compilation.

```
[root@localhost loki]# ls
```

<code>client_db.c</code>	<code>crypt.h</code>	<code>lokid</code>	<code>loki.h</code>	<code>pty.c</code>	<code>shm.o</code>	
<code>client_db.h</code>	<code>crypt.o</code>	<code>lokid.c</code>	<code>loki.o</code>	<code>pty.o</code>	<code>surplus.c</code>	
<code>client_db.o</code>	<code>loki</code>	<code>lokid.o</code>	<code>Makefile</code>	<code>shm.c</code>	<code>surplus.o</code>	<code>crypt.c</code>
<code>loki.c</code>	<code>md5</code>	<code>shm.h</code>				

This compilation results in two binaries being created, `lokid` and `loki`.

Strace (again)

In comparing the output of the newly compiled binary and the `atd` binary, once the memory and library calls are made, the execution is all but identical.

<pre>[root@localhost analysis]# strace ./lokid <snip> socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 3 rt_sigaction(SIGUSR1, {0x804a744, [USR1], SA_RESTART 0x4000000}, {SIG_DFL}, 8) = 0 socket(PF_INET, SOCK_RAW, IPPROTO_RAW) = 4 setsockopt(4, SOL_IP, IP_HDRINCL, [1], 4) = 0</pre>	<pre>[root@localhost /tmp]# strace ./atd <snip> socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 3 sigaction(SIGUSR1, {0x804a6b0, [], SA_INTERRUPT SA_NOMASK SA_ONESHOT}, {SIG_DFL}, 0x4003ac68) = 0 socket(PF_INET, SOCK_RAW, IPPROTO_RAW) = 4</pre>
---	--

¹⁹ Phrack Magazine, Volume Seven, Issue Fifty-One “Phrack Magazine Extraction Utility” Sept 1997 URL: <http://www.phrack.org/phrack/51/P51-17> (26 May 2003)

²⁰ Thanks to Matt Pratt for his assistance with this compilation. mattpratt@yahoo.com

getpid()	= 2148	setsockopt(4, SOL_IP, IP_HDRINCL, [1], 4) = 0	
getpid()	= 2148	getpid()	= 628
shmget(2390, 240, IPC_CREAT 0)	=	getpid()	= 628
6914057		shmget(870, 240, IPC_CREAT 0)	= 6
semget(2572, 1, IPC_CREAT 0x180 0600)	=	semget(1052, 1, IPC_CREAT 0x180 0600)	= 6
65538		shmat(6, 0, 0)	= 0x40008000
shmat(6914057, 0, 0)	=	write(2, "\nLOKI2\troute [(c) 1997 guild cor"..., 52	
0x40014000		LOKI2 route [(c) 1997 guild corporation worldwide]	
write(2, "\nLOKI2\troute [(c) 1997 guild cor"..., 52) = 52	
LOKI2 route [(c) 1997 guild corporation worldwide]		time([1050598690])	=
) = 52		1050598690	
time([1050832306])	=	close(0)	= 0
1050832306		sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}, 0x4003ac68)	= 0
close(0)	= 0	sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}, 0x4003ac68)	= 0
rt_sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}, 8) = 0		sigaction(SIGTSTP, {SIG_IGN}, {SIG_DFL}, 0x4003ac68)	= 0
rt_sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}, 8) = 0		fork()	= 629
rt_sigaction(SIGTSTP, {SIG_IGN}, {SIG_DFL}, 8) = 0		close(4)	= 0
fork()	= 2149	close(3)	= 0
close(4)	= 0	semop(6, 0xbffffb0c, 2)	= 0
close(3)	= 0	shmdt(0x40008000)	= 0
semop(65538, 0xbffff880, 2)	= 0	semop(6, 0xbffffb0c, 1)	= 0
shmdt(0x40014000)	= 0	_exit(0)	= ?
semop(65538, 0xbffff870, 1)	= 0		
_exit(0)	= ?		

The strings of the Lokid compiled file also matched very closely with that of atd. Not surprisingly, the symbols at the start of the strings output did not match as well, but this is due to different compilers, linked libraries etc. However, the actual string components matched all but identically, with the most notable difference being the order which strings displayed the output. Again, this is attributed to difference in compilers, libraries etc.

The effect on the system when Lokid was executed was also very similar, such as the *netstat* output and *ps* showing the spawned Lokid process running.

`[root@localhost loki]# netstat -an`

<snip>

```
raw    0    0 0.0.0.0:1        0.0.0.0:*        7
raw    0    0 0.0.0.0:255     0.0.0.0:*        7
```

These netstat lines were not present in the original grave robber netstat capture.

md5sum (again)

As mentioned above, the md5sums of the Lokid compiled binary will not be the same as the atd binary. But for completeness sake, the md5 and file size of the lokid binary is listed below. You can see that the file size of 15,540 bytes is very similar to that of the atd binary, which was 15,348 bytes in size.

```
[root@localhost loki]# md5sum lokid
b000abaf9af5bfa03b15a650c9ce87    lokid
```

```
[root@localhost loki]# ls -l lokid
-rwxr-xr-x    1 root  root           15540 Apr  20 20:46 lokid
```

lokid usage

As mentioned previously, Loki provides a covert channel. In particular, it provides a cracker the ability to remotely control and monitor a compromised machine via a covert channel. Specifically, an attacker issues commands and obtains their results through an ICMP covert channel.

By sending its control and actual data through ICMP and UDP protocols, normal packet filtering may not detect the covert command. For instance, many firewalls will allow ICMP Echo Request and ICMP Echo Replies to pass through.

The effect is that Loki will just look like ordinary ICMP traffic to the unwary administrator.

An example of how this works shown below. It relies on the compromised machine to be running the Lokid binary (ie atd), and for the “hacker” to use the loki client binary.

The client connects to the compromised machine with the following command (obviously, 127.0.0.1 is the IP address of the target machine);

```
[root@localhost loki] ./loki -d 127.0.0.1
```

```
LOKI2 route [(c) 1997 guild corporation worldwide]
loki>
```

A Loki prompt is returned to the user. They now can view and execute arbitrary command and files on the target machine. Since the server binary runs as root (in order to open a socket), it has full control of the system.

Summary

It would appear that the atd binary is a renamed Lokid server. This piece of malware can be used to create a covert channel, with the purpose to tunnel data and administrative control through ICMP and UDP protocols. Many firewalls would allow such traffic through them, thus allowing a user to transmit data in and out without security policy enforcement.

The file is probably named atd in an attempt to hide the binary in a normal install binary list. A normal system binary called atd can be found on many Linux systems, including Red Hat 6.1 and Red Hat 7.3. In these installations atd is used for task scheduling. On a Red Hat 6.1 system, the atd process runs as a daemon from /usr/sbin. It is quite likely that the malware version of atd would be placed in /usr/sbin, and would appear to be a normal system process if a process listing (*ps*) was performed.

On a Red Hat host, the normal atd binary also runs as a daemon, so a normal administrator would not think the atd process running as a daemon would be unusual.

The behaviour and attributes of the atd binary match those of the compiled lokid server, so closely that this code found is probably the source of the atd binary. A summary of the behaviour is as follows;

- The binary forks, creating a new process under the atd name
- Two raw sockets are created, which listen on IP protocol 1 (ICMP) and 255 (reserved). These sockets are in addition to existing system and application sockets.
- Client software, loki, can connect to the server and execute arbitrary commands and view any files

Although the actual removal of the lokid, or atd binary is trivial, and also trivial to stop the process and socket listener, the larger concern is to what data has been transferred or commands issued through the covert channel. My strong recommendation is that the rest of the compromised machine should be analysed to see if any other tampering has occurred.

Legal Implications

In determining whether the installation and usage of the Loki server was a criminal offence under Australian Commonwealth Law, the first piece of legislation to consider would be the recently passed *Cybercrime Act (Cwth)* 2001²¹.

²¹ *Cybercrime Act (Cwth)* 2001 URL:
<http://scaleplus.law.gov.au/html/pasteact/3/3486/pdf/161of2001.pdf> (26 May 2003)

"The Cybercrime Act 2001 deals with computer crimes such as hacking, denial of service attacks, spreading of computer viruses and web site vandalism (Ellison, 2001). Other offences covered by the Act include unauthorised use of a computer with intent to commit a serious offence such as stalking, fraud or sabotage" (Ellison, 2001). This Act also gives greater powers to law enforcement agencies in investigating computer crimes. Under this Act, the police can duplicate computer data and examine computer equipment and disks at another location. (Scaleplus, 2001). With a court order, the Act makes it compulsory for computer owners to provide assistance to the police in the course of their investigations. (Scaleplus, 2001). Previously, these powers were nonexistent which made it hard for investigation into complex systems protected by passwords and encryption. Moreover, the maximum penalty for computer offences has been increased from two years to 10 years (Taylor, 2001)"²²

In particular, sections 477 – 478 of the Act define the computer offences which would be relevant to a person found using a Loki server in a Commonwealth network. They are as follows;

s477.1 Unauthorized access, modification or impairment with intent to commit a serious offence

To prosecute under this section, you would need to show that the person who using the Loki server was not authorized to access the data on that machine, and that the person intends to commit a serious offence²³ against the Commonwealth (or State or Territory), by the access to the data.

It should be noted that under s477(1)(6), the person can still be found guilty even if the committing of the serious offence is impossible.

s477.2 Unauthorized modification of data to cause impairment

If the person using the Loki server was modifying data through the server, then this section could apply. Again, you would need to show that the modification was unauthorized and that it occurred in a certain manner (see s477(2)(d)).

Similar to the previous section, s477(2)(3) still makes it an offence even if there is no actual impairment to the data held on the computer, or the "reliability, security or operation" of the data.

This section contains a 10 year imprisonment penalty.

s477(3) Unauthorized impairment of electronic communication

²² Yam, Jason "Hacking and Cybercrimes" URL: <http://home.vicnet.net.au/~kengsn/Hacking.pdf> (26 May 2003)

²³ See section 477(1)(9) – Serious Offence is to be read as any offence that is punishable by imprisonment for life or a period of 5 or more years

This is an alternative section to s477(2).

s478.1 Unauthorized access to, or modifications of, restricted²⁴ data

In order to prosecute under this section, you would need to show that the person intentionally and knowingly gained unauthorized access to restricted data. In the Loki server example, you would need to discover exactly what the server was used for.

A 2 year imprisonment penalty would apply for conviction under this section.

s478.2 Unauthorized impairment of data held on a computer disk etc

Similar to the above example, in order to prosecute you would need to show an unauthorized impairment of the reliability, security or operation of data, with intent to cause the impairment, and the knowledge that the impairment is unauthorized.

Again, a 2 year imprisonment penalty would apply.

s478.3 Possession or control of data with intent to commit a computer offence

This section would be a particularly good prosecution to use for Loki. S478(3)(4)(c) defines possession or control of data as being able to control data held in a computer that is in the possession of another person. So if you could show that the offender was controlling the Loki server, and that there was an intention to commit an offence under s478, then a conviction would be possible.

A 3 year imprisonment penalty would apply for this section.

Other legislation which would be worth looking into would include the *Telecommunications Act 1997 (Cwth)*, and *Telecommunications (Interception) 1979 Act (Cwth)*. This legislation governs roles and responsibilities of network carriers, and how their roles with regard to assisting police investigations and preventing crimes against the Commonwealth.

If a conviction under Commonwealth Law was not possible, an argument in Civil law, such as Tort Law, may also be possible. Such an action may be feasible to recover pecuniary costs for damage caused by the Loki server running, and the damage caused by the loss of data or control of a machine. It may even be possible to recover the costs of the investigation and cleanup.

²⁴ Restricted is defined for this section in s278(1)(3) – held on a computer, and to which access is restricted by an access control system associated with the function of the computer

If you were unable to prove that the Loki program was executed by the offender, then it may be difficult to link the offender to actual access to the data or machines involved.

It would also limit how you could use the legislation with respect to the offences being committed by means of a telecommunications service, a requirement in the above mentioned sections.

If you were unable to prosecute in Law, then the use of the Loki server inside the Australian Parliament House DPRS network would be in clear breach of the *DPRS Conditions for Internet Access*. In particular there would be a breach for the installation of unauthorized software. This breach could result a loss of salary, reassignment, or termination of employment.

Other breaches could be found under the *Parliamentary Services Act 1999* where an employee breaches the Parliamentary Service Code of Conduct.

Interview Questions

If the opportunity was to arise for the interview of the user of the Loki server, important preparation would need to be completed.

Preparation for the interview is of key importance, with preparation of the subject matter which will be discussed, the subject who will be interviewed and a readiness for obstacles or issues which will arise during the interview.

The preparation will also reflect back to a variety of purposes for the interview. For instance, the interview might serve more than just to elicit a confession, or to gain further information, such as technical information, information about other parties involved, other activity done by the accused etc.

The KUBARK Counterintelligence Interrogation manual offers the following important qualifications or skills for an interviewer to have, which again can be achieved with good interview preparation;

*"Perhaps the four qualifications of chief importance to the interrogator are (1) enough operational training and experience to permit quick recognition of leads; (2) real familiarity with the language to be used; (3) extensive background knowledge about the interrogatee's native country (and intelligence service, if employed by one); and (4) a genuine understanding of the source as a person."*²⁵

With this in mind, the following questions are proposed;

²⁵ KUBARK "Counter Intelligence Interrogation", July 1963 URL:
<http://www.hiddenmysteries.com/freebook/neuro/k1.html#l> (26 May 2003)

Question – “John, a number of my IDS sensors have been picking up some strange traffic lately, including a bunch of ICMP alerts. Basically, it is clogged up my IDS and annoying all the analysts. I was wondering if you could help me out with that this might be?”

Reason – This question allows a quick way for the suspect to confess, without feeling like they are in too much trouble. The interview is not there to punish, just to gather further facts which a later law enforcement body can utilize.

The question also gives just enough technical information regarding Loki, ie ICMP events, to trigger in the mind of the interviewee that we know what they have been up to.

Question – “John, I have seen all sorts of IDS events in my time with this company, but never ones like this. My analysts all agree that it looks like there was a Loki server running in the network. I remember playing with that back in uni. That is pretty old stuff ... why would anyone use old tools like that?”

Reason - This question gives away some more information to the interviewee. It lets them know that you and your team are knowledgeable and know what goes on in the network. It gives the interviewee an opportunity to defend their technical prowess, and perhaps even elicit some information as to what the offender was trying to exploit / achieve.

Question – “John, my boss knows about this. He really wants to get this resolved quickly. You should see how much paper work is meant to go with this sort of activity! Shall we get this all resolved so we can both move on?”

Reason - This question starts to strike a bit more fear into the interviewee, with the revelation that higher powers are starting to be involved. Hopefully this will get the interviewee to talk a bit more

Question – “John, I can help you out here. That software is pretty powerful stuff. You can understand why our security policy does not allow such code to run in the network. Tell me what you were trying to achieve”

Reason - This question is playing on the empathy approach again. It is attempting to get the interviewee on side, and to disclose what they were trying to achieve.

Question - “John, while you were being interviewed we have impounded your desktop PC and laptop. Both machines will be completely analyzed for non compliance with the Company’s IT Security Policy. Are you aware of the IT Security Policy and the penalties associated? Are there other items on these machines which you would like to talk to me about now?”

Reason – This is the getting tough question to really scare the pants off the interviewee. This is where you may just pick up on some other things that this user was up to, which will make your later analysis job easier.

Additional Information

The following references are recommended for further information;

- Phrack Magazine, Volume Seven, Issue Forty-Nine, “Project Loki” URL: <http://www.phrack.org/show.php?p=49&a=6> (26 May 2003)
- Phrack Magazine, Volume Seven, Issue Fifty-One, “Loki2 - the implementation”, 01 Sept 1997 URL: <http://www.phrack.org/show.php?p=51&a=6> (26 May 2003)
- Australian Legislation - <http://scaleplus.law.gov.au/> (26 May 2003)
- Bootable Forensic Environments - <http://fire.dmzs.com/> (26 May 2003)
- VMWare virtual operating system – <http://www.vmware.com/> (26 May 2003)

Part 2 – Option 2 – Forensic Tool Validation - Ethereal

Introduction

For this assignment option a file *sn.zip* has been provided. This zip file contains a binary file, *sn.dat* and the binary file's corresponding md5 checksum. A quick check of the *sn.md5* file and a checksum of the *sn.dat* file itself show that it is the same *sn.dat* binary which has been previously analyzed by SANS GCFA students. A good analysis of this binary can be found in the paper of James Fung²⁶.

Table 2.2.1 – Baseline cryptographic hashes

holmes:/tmp/binary# md5sum sn.dat 0e954f43fd73f56e812a7285f32e41d3 sn.dat holmes:/tmp/binary# openssl sha1 sn.dat SHA1(sn.dat)= 2314f1a3eadaef2f40c0afb60e53647871ece222

Figure 3 - md5sum of sn.dat as taken by James Fung

It was determined that the *sn.dat* file is actually a compiled version of ADMsniff²⁷. This tool is a packet sniffer which can be compiled for SunOS or Linux. It captures plain text connections, probably in an attempt to capture usernames, passwords and other sensitive information submitted over plain text.

*"The script revealed that eight ports that normally run clear text services were being logged. The following ports: 21 (ftp), 23 (telnet), 109 (pop2), 110 (pop3), 143 (imap), 512 (exec), 513 (login), and 514 (shell) were found to take requests.... After examining the source code ... it was discovered that an additional port, port 31337, the "eleet" port, would also be captured"*²⁸

Since there has already been good analysis of the *sn.dat* binary, we can use this information to verify and test out a particular forensic tool. For this paper the tool for analysis will be the packet capturing tool of *Ethereal*²⁹.

Scope

As described on the Ethereal Homepage,

²⁶ Fung, James "Dead linux machines do tell tales – GCFA Practical Assignment" URL: http://www.giac.org/practical/GCFA/James_Fung_GCFA.pdf (26 May 2003)

²⁷ <http://www.securityfocus.com/tools/215>
<http://packetstormsecurity.nl/groups/ADM/indexsize.shtml> (26 May 2003)

²⁸ Strubinger, Ray "Exercises in the art and science of computer forensics - GCFA Practical Assignment" URL: http://www.giac.org/practical/GCFA/Ray_Strubinger_GCFA.pdf (26 May 2003)

²⁹ <http://www.ethereal.com> (26 May 2003)

“Ethereal is a free network protocol analyzer for Unix and Windows. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the capture data, viewing summary and detail information for each packet. Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.”³⁰

What this means for a forensic analysis, or any network analyst, is that raw packets can be captured from a network, and graphically displayed. The protocol analysis functionality of Ethereal allows an analyst to see exactly is happening or has happened on a network. An analyst does not need to be an expert in the network protocol being viewed in order to gather useful information from Ethereal.

The captures can be stored in a file for later analysis, and loaded on other machines running Ethereal.

Given that the sn.dat, or ADMsniffer captures network traffic, we can compare the output of sn.dat with what Ethereal captures and verify that Ethereal is presenting useful and accurate data back to the analyst.

Tool Description

History

As the earlier description explains, Ethereal is a network protocol analyzer. This functionality, combined with the graphical user interface, distinguishes it from other tools, such as tcpdump³¹. Like many network capturing and sniffing tools, Ethereal is build upon the Libpcap libraries, so there are many similarities between Ethereal and other packet sniffers.

Ethereal entered development in late 1997, being written by Gerald Combs who was trying to track down network problems. Over time, the tool evolved from just a packet sniffer to become a protocol analyser, with many individuals contributing code to dissect and interpret various Internet protocols.

Core Functionality

At Ethereal version 0.9.3 there were 366 protocols which Ethereal could analyze, or dissect for the user. This is where Ethereal has the ability to provide vast amounts of usable information to an analyst. As most network analysts are

³⁰ <http://www.ethereal.com> (26 May 2003)

³¹ <http://www.tcpdump.org> (26 May 2003)

aware, the current Internet protocols are loosely based on the OSI resource model, which is comprised of the following components;

Physical Network – This layer is concerned with raw data bits being transmitted over a physical circuit.

Protocols in this layer include RS-232, RS-449, X.21, and IEEE 802

Data Link Layer – This layer is tasked with providing a reliable, point to point or multi-point connection to the higher network layer.

Protocols in this layer include ISDN, IEEE 802, and HDLC

Network Layer – This layer is responsible for providing an interface between the network device and the delivery system. It is also responsible for choosing which data link service to use.

The most well known protocol in this layer is IP.

Transport Layer – This layer takes data from the above session layer and splits it into components for the network layer. The layer provides an abstraction of the actual networking from a user application, and adds additional networking functionality, such as a reliable connection orientated protocol in the case of TCP.

The most well known protocol in this layer is TCP

Session Layer – This layer aims to provide connections between users of the presentation layer, and includes various options for communication, authorization etc.

Examples of protocols in this layer include RPC and NFS

Presentation and Application Layer – The higher the layers get the more blurred their precise function, hence these two are combined. These are the layers where actual applications interface with the user and then the lower network.

Examples of protocols in this layer include HTTP, FTP, SMTP etc

Each higher layer is encapsulated in the lower layer, ie a TCP packet is encapsulated into the payload of an IP packet. So a raw “frame” on the “wire” is actually comprised of control and data components for each layer. This is illustrated Figure 4 - Encapsulation of Network Protocols

Figure 4 - Encapsulation of Network Protocols

Since Ethereal can analyze so many protocols, the effect is that on most networks, the protocols can be fully dissected and analyzed from the Physical / Data Link Protocols all the way through to the application protocols. Ie from Frame and ATM all the way through to Quake, or HTTP, and everything else in between from routing protocols to specific vendor protocols.

When it is said that Ethereal allows for the dissection of a protocol, what this means is that it can determine the components of each protocol. For instance, in the Figure 4 - Encapsulation of Network Protocols, each layer (ie Ethernet, IP, TCP, HTTP) has its own control information and data relevant to the state of the connection. The Ethernet Frame will hold information of the source and destination MAC addresses, while the IP Datagram will hold information such as source and destination IP addresses IP flags, fragmentation fields, etc etc.

It is these individual elements which Ethereal can break out from the raw packet and display in a human readable output. This allows for relatively easy protocol analysis, even if the analyst is not an expert in the raw protocol itself.

Consider the following screenshots from Ethereal (version 0.9.7 on Windows 2000 Professional), Figure 5 - Illustration of Ethereal dissection (part 1), Figure 6 - Illustration of Ethereal dissection (part 2), and Figure 7 - Illustration of fine granularity dissection

The first two figures show how from the raw hex packet format, Ethereal has dissected and highlighted just the HTTP protocol GET request. From the same

overall packet, the third figure shows the fine granularity of Ethereal, displaying the IP TTL value imbedded in the same raw packet. Not only has Ethereal dissected and highlighted the TTL field, it has displayed it in a human readable format. Hex value 80 = decimal value 128, which is displayed in the blue highlighted portion of the illustration.

```

Frame 8 (421 bytes on wire, 421 bytes captured)
Ethernet II, Src: 00:20:e0:6a:4f:64, Dst: 00:50:8b:6b:9f:15
Internet Protocol, Src Addr: 192.168.50.233 (192.168.50.233), Dst Addr: 192.168.50.22 (192.168.50.22)
Transmission Control Protocol, Src Port: 3103 (3103), Dst Port: http (80), Seq: 3708104545, Ack: 4154989759
Hypertext Transfer Protocol
GET /index.html HTTP/1.1\r\n
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-ppt\r\n
Accept-Language: en-au\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; windows NT 5.0)\r\n
Host: 192.168.50.22\r\n
Connection: Keep-Alive\r\n
\r\n

```

Figure 5 - Illustration of Ethereal dissection (part 1)

```

0000 00 50 8b 6b 9f 15 00 20 e0 6a 4f 64 08 00 45 00 .P.k... .jod..E.
0010 01 97 fc 5a 40 00 80 06 16 b6 c0 a8 32 e9 c0 a8 ...Z@... ....2...
0020 32 16 0c 1f 00 50 dd 05 2f 61 f7 a8 1c bf 50 18 2....P.. /a....P.
0030 44 70 5d c7 00 00 47 45 54 20 2f 69 6e 64 65 78 Dp]...GE T /index
0040 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a .html HT TP/1.1..
0050 41 63 63 65 70 74 3a 20 69 6d 61 67 65 2f 67 69 Accept: image/gi

```

Figure 6 - Illustration of Ethereal dissection (part 2)

The screenshot shows the Ethereal (Wireshark) interface. The top pane displays a list of packets, with packet 8 selected. The middle pane shows the detailed view of packet 8, highlighting the IP header and the Time to Live (TTL) field, which is set to 128. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.50.23	192.168.50.22	TCP	3389 > 60473 [PSH, ACK] Seq=1270953635 Ack=3708104545
2	0.000097	192.168.50.22	192.168.50.23	TCP	60473 > 3389 [ACK] Seq=306655844 Ack=1270954000
3	0.250176	PREMAX_6a:4f:64	Broadcast	ARP	who has 192.168.50.22? Tell 192.168.50.233
4	0.250339	PREMAX_6a:4f:64	PREMAX_6a:4f:64	ARP	192.168.50.22 is at 00:50:8b:6b:9f:15
5	0.250347	192.168.50.233	192.168.50.22	TCP	3103 > http [SYN] Seq=3708104544 Ack=0 win=1460
6	0.250545	192.168.50.22	192.168.50.233	TCP	http > 3103 [SYN, ACK] Seq=4154989758 Ack=3708104545
7	0.250568	192.168.50.233	192.168.50.22	TCP	3103 > http [ACK] Seq=3708104545 Ack=4154989759
8	0.250725	192.168.50.233	192.168.50.22	HTTP	GET /index.html HTTP/1.1
9	0.250907	192.168.50.22	192.168.50.233	TCP	http > 3103 [ACK] Seq=4154989759 Ack=3708104545
10	0.251903	192.168.50.22	192.168.50.233	HTTP	HTTP/1.1 200 OK
11	0.251931	192.168.50.22	192.168.50.233	HTTP	Continuation
12	0.251941	192.168.50.22	192.168.50.233	TCP	http > 3103 [FIN, ACK] Seq=4154992301 Ack=3708104545
13	0.251982	192.168.50.233	192.168.50.22	TCP	3103 > http [ACK] Seq=3708104912 Ack=4154992301
14	0.292366	192.168.50.233	192.168.50.22	TCP	3103 > http [FIN, ACK] Seq=3708104912 Ack=4154992301
15	0.292551	192.168.50.22	192.168.50.233	TCP	http > 3103 [ACK] Seq=4154992302 Ack=3708104912

Frame 8 (421 bytes on wire, 421 bytes captured)

Ethernet II, Src: 00:20:e0:6a:4f:64, Dst: 00:50:8b:6b:9f:15

Internet Protocol, Src Addr: 192.168.50.233 (192.168.50.233), Dst Addr: 192.168.50.22 (192.168.50.22)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)

Total Length: 407

Identification: 0xfc5a

Flags: 0x04

Fragment offset: 0

Time to live: 128

Protocol: TCP (0x06)

Header checksum: 0x16b6 (correct)

Source: 192.168.50.233 (192.168.50.233)

Destination: 192.168.50.22 (192.168.50.22)

Transmission Control Protocol, Src Port: 3103 (3103), Dst Port: http (80), Seq: 3708104545, Ack: 4154989759

Hypertext Transfer Protocol

GET /index.html HTTP/1.1\r\n

0010 01 97 fc 5a 40 00 80 06 16 b6 c0 a8 32 e9 c0 a8 ...Z@...2...

0020 32 16 0c 1f 00 50 dd 05 2f 61 f7 a8 1c bf 50 18 2....P.. /a....P.

0030 44 70 5d c7 00 00 47 45 54 20 2f 69 6e 64 65 78 Dp]...GE T /index

0040 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a .html HT TP/1.1..

0050 41 63 63 65 70 74 3a 20 69 6d 61 67 65 2f 67 69 Accept: image/gi

Figure 7 - Illustration of fine granularity dissection

Given this power of being able to have a human readable view into raw packets or frames, Ethereal adds to this by giving analysts the ability to filter on any of these fields. This allows for the extension beyond the standard PCAP style filters of just filtering on fields in the IP or TCP headers, and includes the ability to dissect on application layer fields. A standard PCAP style filter would be – *src host 1.1.1.1 and dst net 2.2 and dst port 80* – This filter would show all TCP and UDP traffic which had a source IP address of 1.1.1.1 and was destined to port 80 in the 2.2 B class network.

An example of the extension beyond PCAP would be if you were analyzing a *syslog* stream, perhaps looking for all the times an interface went into promiscuous mode, you could apply a filter of - *syslog.msg >="promiscuous"*.

Obviously, all these filters can be concatenated together to give some very powerful filter expressions, including the ability to follow a full session stream of data. We will return to some of this functionality in the analysis of the sn.dat binary.

Availability / Usage

At the time of writing this paper, the most recent version of Ethereal was 0.9.11, with binaries being available for pretty much all the major platforms including; AIX, BeOS, FreeBSD, HP-UX, FreeBSD, Solaris, Windows, MacOS X and of course Linux in it's many forms (from Debian to RedHat to LinuxPPC).

Overall, if you have an X-Windows style environment, and a few basic dependencies, Ethereal is available for you. The important dependencies for running Ethereal are GTK+³² version 1.2 or later, and Libpcap³³.

Other optional dependencies include perl for documentation and Zlib for reading gzip compressed files on the fly, and NET-SNMP for the translation of SNMP OIDs to names. I will not discuss these in any further detail, as they are not relevant for the testing of the sn.dat binary.

Because of these dependencies, and the need to run Ethereal in a X-Windows / Windows style environment, the chances of getting a portable statically compiled binary for a Linux based system is small.

A simple alternative though is to use a bootable Linux distribution, such as FIRE³⁴ or Knoppix³⁵. Both of these bootable Linux X86 CD images allow an

³² <http://www.gtk.org/> (26 May 2003)

³³ <http://www.gtk.org> (26 May 2003)

³⁴ <http://fire.dmzs.com> (26 May 2003)

³⁵ <http://www.knopper.net/knoppix/> (26 May 2003)

analyst to boot from a CD in a full X-Windows environment, and have full access to a range of Linux tools, including Ethereal.

The recent version of the Knoppix bootable CD, version 3.2 ships with Ethereal 0.9.5 with GTK+ 1.2.10, Glib 1.2.10 and libpcap 0.6.

Such bootable CD's give an analyst a working environment which is fully functional and exactly the same every time it is used. Data can be permanently stored onto local hard disks, removable drives, or even USB based memory sticks.

This is a simple and sound way that an analyst can be sure that the tools are being used in an evidentiary sound way. An MD5 or other baseline can be taken of the CD ISO and its contents and because of the read only nature of CD's, the analyst can be guaranteed of the integrity of the tools on the CD. By default, local machine hard disks are not even mounted, so there is minimal risk of contamination from these.

Knoppix includes full instructions and a build environment for creating your own binaries and operating environment. It would therefore be possible to build all your analyst tools, including Ethereal, from source, which would further give a guarantee of tool integrity, because of the option of source code review if required.

System Footprint

On a Windows 2000 Professional host, an Ethereal install has a total footprint of 23.3 MB (including documentation), with an 8.78 MB ethereal.exe binary. The required WinPcap libraries only have an install footprint of 231 KB.

On the above mentioned Knoppix system, the Ethereal binary is 3.8 MB in size, and there is another 388 KB of additional man pages and plugins.

On a Red Hat 7.3 installation with Ethereal version 0.9.3, the following files are opened by the program during a normal session. If Ethereal is started, a real time capture is taken, the resulting capture is saved to a file called test-save, and Ethereal is exited.

This list of files opened was captured with *strace* and the open lines extracted with *grep*.

```
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
open("/usr/lib/libsnmp.so.0", O_RDONLY) = 3
open("/lib/libcrypto.so.2", O_RDONLY) = 3
open("/usr/lib/libpcap.so.0.6.2", O_RDONLY) = 3
open("/usr/lib/libgtk-1.2.so.0", O_RDONLY) = 3
```

```

open("/usr/lib/libgdk-1.2.so.0", O_RDONLY) = 3
open("/usr/lib/libgmodule-1.2.so.0", O_RDONLY) = 3
open("/usr/lib/libglib-1.2.so.0", O_RDONLY) = 3
open("/lib/libddl.so.2", O_RDONLY) = 3
open("/usr/X11R6/lib/libXi.so.6", O_RDONLY) = 3
open("/usr/X11R6/lib/libXext.so.6", O_RDONLY) = 3
open("/usr/X11R6/lib/libX11.so.6", O_RDONLY) = 3
open("/lib/i686/libm.so.6", O_RDONLY) = 3
open("/usr/lib/libz.so.1", O_RDONLY) = 3
open("/lib/i686/libc.so.6", O_RDONLY) = 3
open("/etc/snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/share/snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/share/snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/lib/snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/lib/snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/root/.snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/var/ucd-snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/var/ucd-snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.snmp/mibs", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = -1
ENOENT (No such file or directory)
open("/usr/share/snmp/mibs/.index", O_RDONLY) = 3
open("/usr/share/snmp/mibs/IP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMPv2-SMI.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/SNMPv2-TC.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/IF-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMPv2-MIB.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/IANAifType-MIB.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/TCP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/UDP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/RFC1213-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/RFC1155-SMI.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/UCD-SNMP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/UCD-DEMO-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/HOST-RESOURCES-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/HOST-RESOURCES-TYPES.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/IPV6-ICMP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/IPV6-MIB.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/IPV6-TC.txt", O_RDONLY) = 5
open("/usr/share/snmp/mibs/IPV6-TCP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/IPV6-UDP-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMP-VIEW-BASED-ACM-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMP-FRAMEWORK-MIB.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/SNMP-COMMUNITY-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMP-TARGET-MIB.txt", O_RDONLY) = 4
open("/usr/share/snmp/mibs/UCD-DLMOD-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMP-MPD-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMP-USER-BASED-SM-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMP-NOTIFICATION-MIB.txt", O_RDONLY) = 3
open("/usr/share/snmp/mibs/SNMPv2-TM.txt", O_RDONLY) = 3
open("/etc/snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)

```

```

open("/usr/share/snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/share/snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/lib/snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/lib/snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/root/.snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/var/ucd-snmp/snmp.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/var/ucd-snmp/snmp.local.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/lib/ethereal/plugins/0.9.3",
O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
open("/usr/lib/ethereal/plugins/0.9.3/coseventcomm.so", O_RDONLY) = 4
open("/usr/lib/ethereal/plugins/0.9.3/cosnaming.so", O_RDONLY) = 4
open("/usr/lib/ethereal/plugins/0.9.3/gryphon.so", O_RDONLY) = 4
open("/usr/lib/ethereal/plugins/0.9.3/mgcp.so", O_RDONLY) = 4
open("/usr/local/lib/ethereal/plugins/0.9.3",
O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/plugins", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) =
-1 ENOENT (No such file or directory)
open("/usr/share/locale/locale.alias", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_IDENTIFICATION", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_MEASUREMENT", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_TELEPHONE", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_ADDRESS", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_NAME", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_PAPER", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_MESSAGES", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_MESSAGES/SYS_LC_MESSAGES", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_MONETARY", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_COLLATE", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_TIME", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_NUMERIC", O_RDONLY) = 3
open("/usr/lib/locale/en_AU/LC_CTYPE", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/locale.alias", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/locale.dir", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/iso8859-1/XI18N_OBJS", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/common/xlcDef.so.2", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/locale.alias", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/locale.dir", O_RDONLY) = 3
open("/usr/X11R6/lib/X11/locale/iso8859-1/XLC_LOCALE", O_RDONLY) = 3
open("/root/.Xauthority", O_RDONLY) = 4
open("/usr/X11R6/lib/X11/locale/locale.alias", O_RDONLY) = 4
open("/usr/X11R6/lib/X11/locale/locale.dir", O_RDONLY) = 4
open("/usr/X11R6/lib/X11/locale/common/ximcp.so.2", O_RDONLY) = 4
open("/usr/X11R6/lib/X11/locale/compose.dir", O_RDONLY) = 4
open("/usr/X11R6/lib/X11/locale/iso8859-1/Compose", O_RDONLY) = 4
open("/usr/lib/gconv/gconv-modules.cache", O_RDONLY) = 5
open("/usr/lib/gconv/ISO8859-1.so", O_RDONLY) = 5
open("/etc/nsswitch.conf", O_RDONLY) = 4
open("/etc/ld.so.cache", O_RDONLY) = 4
open("/lib/libnss_files.so.2", O_RDONLY) = 4
open("/etc/passwd", O_RDONLY) = 4
open("/etc/gtk/gtkrc", O_RDONLY) = 4
open("/root/.gtkrc", O_RDONLY) = 4
open("/usr/share/themes/Raleigh/gtk/gtkrc", O_RDONLY) = 5
open("/usr/lib/gtk/themes/engines/libraleigh.so", O_RDONLY) = 6

```

```

open("/usr/lib/gtk/themes/engines/libraleigh.so", O_RDONLY) = 6
open("/usr/share/locale/en_AU/LC_MESSAGES/libc.mo", O_RDONLY) = -1 ENOENT (No
such file or directory)
open("/usr/share/locale/en/LC_MESSAGES/libc.mo", O_RDONLY) = -1 ENOENT (No such
file or directory)
open("/etc/ethereal/ethereal.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/preferences", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/cfilters", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/filters", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/dfilters", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/filters", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/usr/share/locale/en_AU/LC_MESSAGES/gtk+.mo", O_RDONLY) = -1 ENOENT (No
such file or directory)
open("/usr/share/locale/en/LC_MESSAGES/gtk+.mo", O_RDONLY) = -1 ENOENT (No such
file or directory)
open("/root/.ethereal/colorfilters", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/tmp/etherXXXXARS115", O_RDWR|O_CREAT|O_EXCL, 0600) = 6
open("/tmp/etherXXXXARS115", O_RDONLY) = 6
open("/tmp/etherXXXXARS115", O_RDONLY) = 8
open("IOR.txt", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/root", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 6
open("/root/test-save", O_RDONLY) = 6
open("/root/test-save", O_RDONLY) = 9
open("IOR.txt", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ethereal/ethereal.conf", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/preferences", O_RDONLY) = -1 ENOENT (No such file or
directory)
open("/root/.ethereal/preferences", O_WRONLY|O_CREAT|O_TRUNC, 0666) = -1 ENOENT
(No such file or directory)

```

Of interest in this file list, are the libraries which are opened. These are displayed in the first 16 lines. There is a lot of SNMP related, and locale related files opened following the libraries which are loaded.

The generated file which is saved, ie *test-save*, is initially written to a temp file, and this is then renamed to the final save name.

```

rename("/tmp/etherXXXXARS115", "/root/test-save") = 0

```

An additional *strace* was taken where ethereal was opened and the *test-save* file was opened for analysis. Ethereal only opened the file in a Read Only format, ie

```

open("/root/test-save", O_RDONLY) = 6
open("/root/test-save", O_RDONLY) = 7

```

Ethereal can be asserted as being safe for analyzing an existing capture file because it will not alter the contents of the file.

Test Apparatus

Since the object of this test is to validate the Ethereal tool, a test environment has been constructed which will make use of three independent and different Ethereal installs.

As the behaviour of the sn.dat file is already known, ie that it will log certain clear text network activity, the object of the test environment will be to validate that the traffic which sn.dat captures is equally captured and interpreted by the various Ethereal instances running.

The test network will consist of two physical machines, and three logical operating systems. The first machine will be running the bootable Linux distribution of Knoppix. The second machine will be running Windows 2000 Professional with a Red Hat 7.3 installation running in VMware. See Figure 8 - Physical Test Network Setup. The Knoppix machine will run the sn.dat binary from its /tmp directory. Meanwhile, a number of connections will be made from the Knoppix machine to services (FTP and Telnet) to the Red Hat virtual machine.

Figure 8 - Physical Test Network Setup

Figure 9 - Logical Test Network Data Flow

Figure 9 - Logical Test Network Data Flow illustrates this logical flow of information. A connection will be made out from the Knoppix host, where both sn.dat and the Knoppix Ethereal instance will capture the traffic. The traffic will traverse through the Windows host, again with an Ethereal instance capturing the traffic, and then through VMware to the Red Hat installation – which again will be running Ethereal and capturing the network traffic.

Analysis Objectives

As a result of the FTP and Telnet sessions between the Knoppix host and the Red Hat host, there will four packet capture created. One from each of the three Ethereal instances, and one capture from the sn.dat file.

The aim will be to review and compare the following;

- Validate that the FTP and Telnet connections were captured by sn.dat
- Validate that the FTP and Telnet connections were captured by Ethereal
- Validate that the Ethereal instances accurately represent what the sn.dat file captured
- Validate that the various Ethereal instances match each other with respect to the data captured

Criteria for Approval

The expected results are that the data captured by the sn.dat binary will match that collected by the three Ethereal captures. We do not expect exact character matches to occur because of the representation of white space by the various tools and operating systems.

It is expected that the three Ethereal instances will capture the same data, and these should be identical.

Machine Specifications

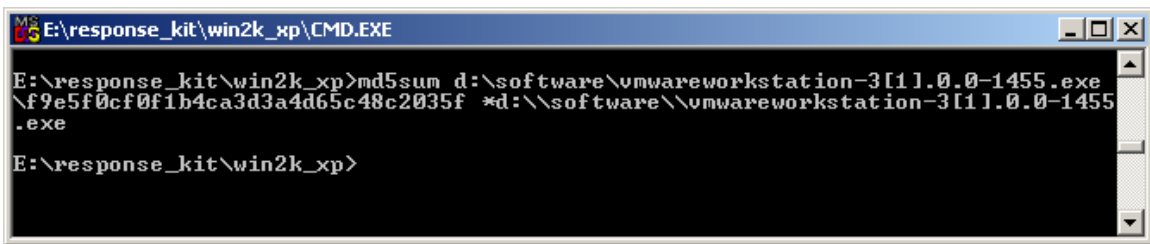
The specifics of each of the machines involved are as follows;

Knoppix Test Machine

- Knoppix version KNOPPIX_V3.2-2003-03-28-EN.iso
- Booted on Generic Intel PC – No relevant serial number – Labelled Fred
- CD-Rom installed and set to boot first in BIOS
- Hard-disks in machine, but will not be mounted by Knoppix
- ISO Downloaded from http://public.planetmirror.com/pub/knoppix/KNOPPIX_V3.2-2003-03-21b-EN.iso
- MD5 Validation from http://public.planetmirror.com/pub/knoppix/KNOPPIX_V3.2-2003-03-21b-EN.iso.md5 22e2619e08828930c01a9b5dfdef3e99 KNOPPIX_V3.2-2003-03-28-EN.iso
- MD5Sum validated on Windows host with md5sum utility provided on GCFA SANS CD version 1.6.

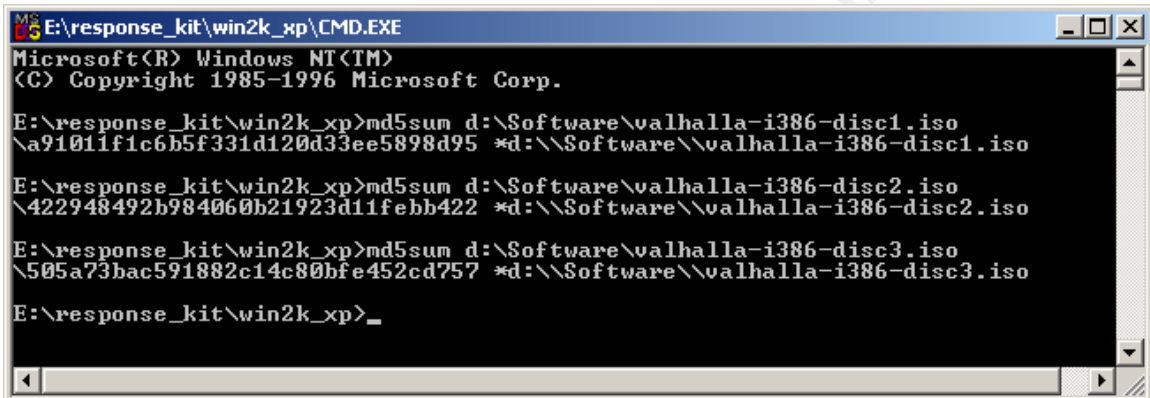
Red Hat Test Machine

- Red Hat version 7.3
- Generic Workstation install plus the following RPM packages;
 - ethereal-0.9.3-3
 - gnome-ethereal-0.9.3-3
 - ltrace-0.3.10-7
 - strace-4.4-4
 - telnet-server-0.17-20
 - wu-ftpd-2.6.2-5
- Two ISO's downloaded from <ftp://ftp.mirror.aarnet.edu.au/pub/redhat/redhat-7.3/en/iso/i386/>
 - Valhalla-i386-disk1.iso
 - Valhalla-i386-disk2.iso
- Corresponding MD5SUM file
 - a91011f1c6b5f331d120d33ee5898d95 valhalla-i386-disc1.iso
 - 422948492b984060b21923d11febb422 valhalla-i386-disc2.iso
 - Validation taken at Figure 11 - Red Hat ISO Validation
- MD5Sum validated on Windows host with md5sum utility provided on GCFA SANS CD version 1.6.
- Installed in VMware Workstation, version 3.0.0 build-1455, running on Windows 2000 Test machine



```
E:\response_kit\win2k_xp\CMD.EXE
E:\response_kit\win2k_xp>md5sum d:\software\vmwareworkstation-3[1].0.0-1455.exe
\f9e5f0cf0f1b4ca3d3a4d65c48c2035f *d:\software\vmwareworkstation-3[1].0.0-1455.exe
E:\response_kit\win2k_xp>
```

Figure 10 - Checksum of VMware install binary

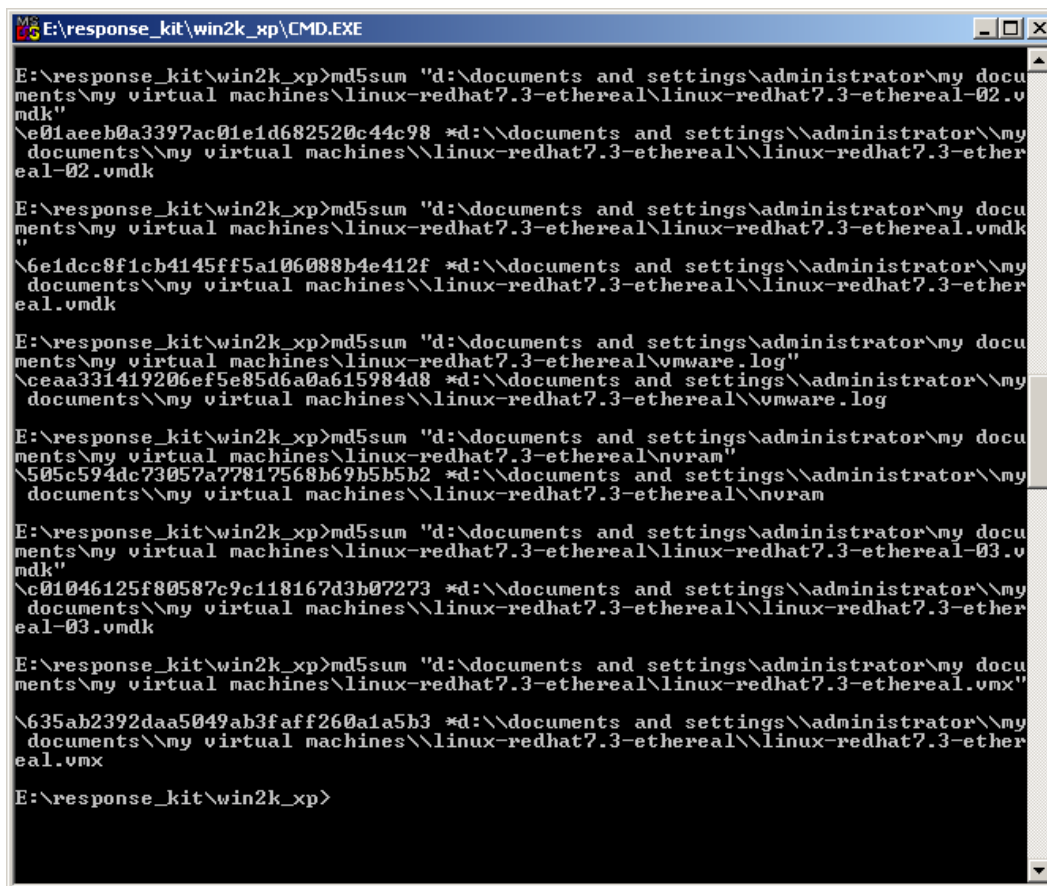


```
E:\response_kit\win2k_xp\CMD.EXE
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
E:\response_kit\win2k_xp>md5sum d:\Software\valhalla-i386-disc1.iso
\xa91011f1c6b5f331d120d33ee5898d95 *d:\Software\valhalla-i386-disc1.iso
E:\response_kit\win2k_xp>md5sum d:\Software\valhalla-i386-disc2.iso
\422948492b984060b21923d11febb422 *d:\Software\valhalla-i386-disc2.iso
E:\response_kit\win2k_xp>md5sum d:\Software\valhalla-i386-disc3.iso
\505a73bac591882c14c80bfe452cd757 *d:\Software\valhalla-i386-disc3.iso
E:\response_kit\win2k_xp>_
```

Figure 11 - Red Hat ISO Validation

- VMware data and configuration files of the install image – see Figure 12 - VMware data files md5sums

Although it would not be possible to generate the exact same checksums from another Red Hat install in VMware, these checksums can be used to validate the original build test environment. It would therefore be possible to fully replicate the test with the exact same system state as when the test was completed.



```
E:\response_kit\win2k_xp>md5sum "d:\documents and settings\administrator\my documents\my virtual machines\linux-redhat7.3-ethereal\linux-redhat7.3-ethereal-02.vmdk"
\xe01aeeb0a3397ac01e1d682520c44c98 *d:\\documents and settings\\administrator\\my documents\\my virtual machines\\linux-redhat7.3-ethereal\\linux-redhat7.3-ethereal-02.vmdk

E:\response_kit\win2k_xp>md5sum "d:\documents and settings\administrator\my documents\my virtual machines\linux-redhat7.3-ethereal\linux-redhat7.3-ethereal.vmdk"
\6e1dcc8f1cb4145ff5a106088b4e412f *d:\\documents and settings\\administrator\\my documents\\my virtual machines\\linux-redhat7.3-ethereal\\linux-redhat7.3-ethereal.vmdk

E:\response_kit\win2k_xp>md5sum "d:\documents and settings\administrator\my documents\my virtual machines\linux-redhat7.3-ethereal\vmware.log"
\ceaa331419206ef5e85d6a0a615984d8 *d:\\documents and settings\\administrator\\my documents\\my virtual machines\\linux-redhat7.3-ethereal\\vmware.log

E:\response_kit\win2k_xp>md5sum "d:\documents and settings\administrator\my documents\my virtual machines\linux-redhat7.3-ethereal\nvram"
\505c594dc73057a77817568b69b5b5b2 *d:\\documents and settings\\administrator\\my documents\\my virtual machines\\linux-redhat7.3-ethereal\\nvram

E:\response_kit\win2k_xp>md5sum "d:\documents and settings\administrator\my documents\my virtual machines\linux-redhat7.3-ethereal\linux-redhat7.3-ethereal-03.vmdk"
\c01046125f80587c9c118167d3b07273 *d:\\documents and settings\\administrator\\my documents\\my virtual machines\\linux-redhat7.3-ethereal\\linux-redhat7.3-ethereal-03.vmdk

E:\response_kit\win2k_xp>md5sum "d:\documents and settings\administrator\my documents\my virtual machines\linux-redhat7.3-ethereal\linux-redhat7.3-ethereal.vmx"
\635ab2392daa5049ab3faff260a1a5b3 *d:\\documents and settings\\administrator\\my documents\\my virtual machines\\linux-redhat7.3-ethereal\\linux-redhat7.3-ethereal.vmx

E:\response_kit\win2k_xp>
```

Figure 12 - VMware data files md5sums

Windows Test Machine

- Windows 2000 Professional
- See Figure 13 - Windows Test Environment - Build Version
- Additional software installed;
 - Ethereal 0.9.11, see Figure 15 - Checksum on Ethereal and WinPcap Installers for checksum of installer
Downloaded from <http://www.ethereal.com/distribution/win32/>
 - WinPcap 3.0, see Figure 15 - Checksum on Ethereal and WinPcap Installers for checksum of installer
Downloaded from <http://winpcap.mirror.ethereal.com/install/default.htm>
 - VMware workstation 3.0.0 build 1455, See Figure 10 - Checksum of VMware install binary for checksum of installer
 - See Figure 14 - Windows Test Environment - Installed Programs
- Installed on Dell Inspiron 8100 Laptop – Service tag – GLD-XXXX

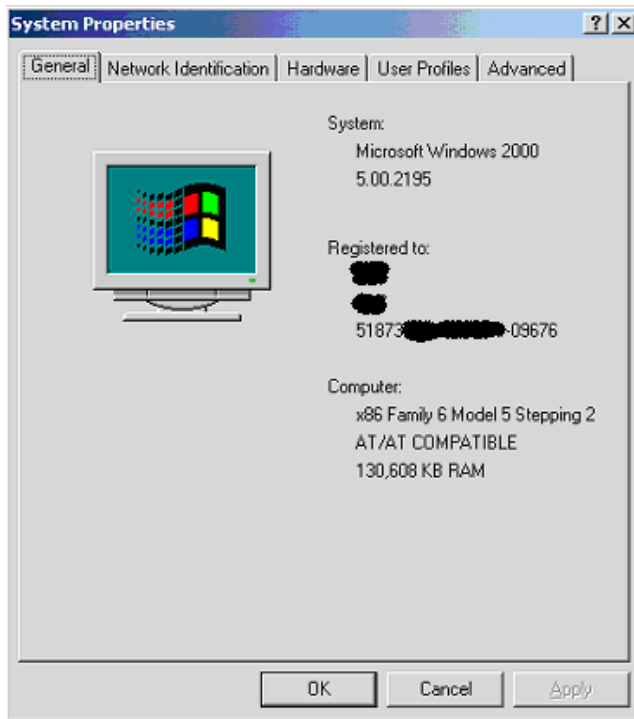


Figure 13 - Windows Test Environment - Build Version

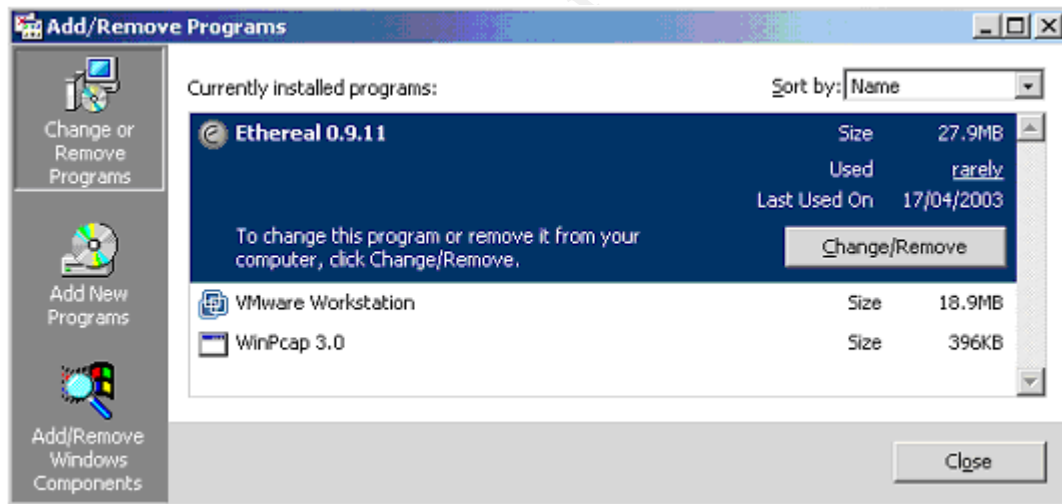
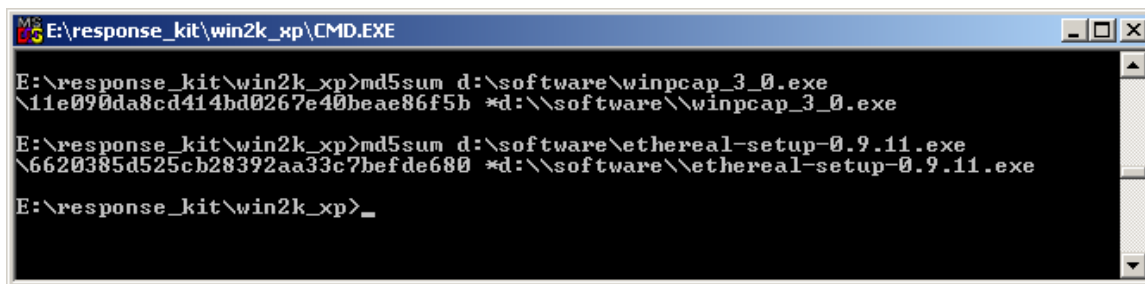


Figure 14 - Windows Test Environment - Installed Programs



```
E:\response_kit\win2k_xp\CMD.EXE
E:\response_kit\win2k_xp>md5sum d:\software\winpcap_3_0.exe
\11e090da8cd414bd0267e40beae86f5b *d:\software\winpcap_3_0.exe
E:\response_kit\win2k_xp>md5sum d:\software\ethereal-setup-0.9.11.exe
\6620385d525cb28392aa33c7befde680 *d:\software\ethereal-setup-0.9.11.exe
E:\response_kit\win2k_xp>
```

Figure 15 - Checksum on Ethereal and WinPcap Installers

Environmental Considerations

The two physical test machines are connected via a 10/100 Netgear switch. The interfaces from the Generic PC and the Dell Laptop are both 10/100 compatible. There are no other networks attached to this test network.

The Windows Installation has been connected to the Internet to download the installed software. This connection was protected by a Mandrake Linux based firewall call Mandrake Multi-Network Firewall³⁶, and a PIX firewall. No inbound connections from the Internet were possible to the Windows host whilst connected through the Firewall. It is highly unlikely that this host has been tainted by the Internet connection.

The md5sum.exe file used to validate the Knoppix and Red Hat ISO md5sums was that provided on the SANS GCFA CD version 1.6.

The Red Hat installation has not been active whilst the host operating system (ie the Windows host) has been connected to any other network.

The Knoppix install is fresh each time it is booted, and has not connected to any other network.

Although Ethereal does rely on a number of libraries and system files, both the Red Hat and Windows installations are fresh, with no alterations from their original install. The ISO images which have been used for both the Red Hat and Knoppix installs have been validated with the distributing web site, and these are believed to be valid.

The ISO images used have been kept and their details recorded such that further validation of these could be performed if required.

It is not believed that there are any other outside forces which will impact on this test.

³⁶ <http://www.mandrakesecure.net/en/mnf.php> (26 May 2003)

Description of the procedures

Standards Used

Data/output files will be stored in the following directories;

Knoppix – /tmp
Red Hat – /opt/analysis
Windows - d:\opt\analysis

The format of saved data files will have the following format;

<machine_name>-<generating process>-<identifier>-<YYYYMMDDHHMM-X>

eg knoppix-ethereal-port_80_filter-2003032212902

The 'script' command will be used on the Knoppix and Red Hat systems. The command will be executed in the respective data directory. This will ensure a complete history is maintained of commands executed during the tests.

Since both the Knoppix and Red Hat environments are X-Windows based, if multiple xterminals are used, the name of the script output file should be set to the following format – '<machine_name>-typescript-YYYYMMDDHHMM-X', where X is a unique, incrementing integer.

Users – the following users will be used for the test;

Knoppix – root
Red Hat – root
Windows - administrator

Data transfer between the three logical machines will be via floppy disk. The floppy disk will be initialized on the Red Hat host with the following command;

```
dd if=/dev/zero of=/dev/fd0 bs=512 count=2880
```

Any gathered data to be permanently stored for evidentiary purposes will have their md5sums created from the SANS GCFA CD version 1.6 of md5sum.exe, and then be burnt to CD. These md5sums of each individual data file will be stored in the CD ISO.

Once a CD ISO has been created, an MD5sum of this iso will be taken, and written on the CD label. The CDs will be labelled in the following format;

Title - SANS GCFA Assignment Q2b
Validation of Ethereal tool
Created – YYYYMMDDHHMM
Author – Andrew Hall
MD5Sum- xxxxxx

The CD will then be signed by the author. The CD will be stored in a locked safe, and the md5sum of the CD iso stored in a separate secure location.

Integrity Validation

To ensure the integrity of the tests, the following precautions will be taken;

- The tests will be performed on the closed network (as previously discussed)
- Both the Red Hat and Knoppix md5sums will be validated before testing begins – ie their known base environment
- Any permanent data kept will be burnt to CD as described above
- The floppy disk used to transfer files between the test machines will be initialized before use
- The Knoppix host will not mount any local drives
- The script command will be used on the Linux machines to maintain an accurate log of information
- The md5sum.exe binary will be run from the SANS GCFA CD
- Complete instructions / steps of the test will be recorded so that reproduction of results will be possible.
- Data will be read live from the network

Host Preparation

1. Red Hat Host
 - a. Validate the checksums of the VMware data files to be used
 - b. Start the Red Hat test install within VMware
 - c. Login as the root user
 - d. Edit the Red Hat servers /etc/ntp.conf file
 - i. Add the following line – this will setup the Red Hat host as a timeserver. Although this won't be synced to an external source, at least the three test servers will have the same time

restrict 192.168.0.0 mask 255.255.255.0

- ii. Restart ntp with the command `/etc/init.d/ntpd restart`

```
[root@localhost tmp]# /etc/init.d/ntpd restart
Shutting down ntpd: OK
Starting ntpd: OK
```

- e. Ensure that the timezone is correct using the *tzselect* command
 - i. Set the timezone to Australia → New South Wales – most locations
- f. Edit the `/etc/xinetd.d/telnet` file
 - i. Edit the following line – this will allow users to log in via telnet on this machine
disable = no
- g. Edit the `/etc/xinetd.d/wu-ftpd` file
 - i. Edit the following line – this will allow users to log in via ftp on this machine
disable = no
- h. Restart xinetd to make these services start

```
[root@localhost tmp]# /etc/init.d/xinetd restart
Stopping xinetd: OK
Starting xinetd: OK
```

- i. Add a new user to the machine – this way another user other than root is available for testing

```
[root@localhost tmp]# useradd andrew
[root@localhost tmp] # passwd andrew
Changing password for user andrew.
New password:
BAD PASSWORD: it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
```

- j. Edit the file `/etc/init.d/syslog`
 - i. Edit the following line – this will allow other hosts to syslog to this machine – add a `-r` option to the command line
SYSLOGD_OPTIONS
 - ii. Restart the syslog daemon

- k. Ensure that the time and date of the machine are reasonable

```
[root@localhost root]# date
Thu Apr 17 15:07:24 EST 2003
```


I. Set the IP address of the host

```
[root@localhost root]# ifconfig eth0 192.168.0.1 netmask 255.255.255.0
broadcast 192.168.0.255
```

m. Validate the following system information

```
[root@localhost root]# uname -a
Linux localhost.localdomain 2.4.18-3 #1 Thu Apr 18 07:37:53 EDT 2002 i686
unknown
```

```
[root@localhost root]# mount
/dev/sda2 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

```
[root@localhost root]# ifconfig -a
eth0    Link encap:Ethernet  HWaddr 00:50:56:4C:B9:0C
        inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
        UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500
        Metric:1
        RX packets:1020 errors:0 dropped:0 overruns:0 frame:0
        TX packets:529 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:145625 (142.2 Kb)  TX bytes:86324 (84.3 Kb)
        Interrupt:10 Base address:0x10a0

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:70 errors:0 dropped:0 overruns:0 frame:0
        TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:4582 (4.4 Kb)  TX bytes:4582 (4.4 Kb)
```

```
[root@localhost root]# netstat -an37
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:1024            0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:1025          0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:111            0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:21             0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:23             0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:25           0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:1024           0.0.0.0:*
```

³⁷ As can be seen, the syslog daemon is not listening! That is because it was never restarted. This was easily verified in the script file taken of the commands run. It is of no consequence for the remainder of the test.

```
udp    0    0 0.0.0.0:68      0.0.0.0:*
udp    0    0 0.0.0.0:111     0.0.0.0:*
udp    0    0 192.168.50.229:123 0.0.0.0:*
udp    0    0 127.0.0.1:123    0.0.0.0:*
udp    0    0 0.0.0.0:123     0.0.0.0:*
<snip>
```

```
[root@localhost root]# ethereal -v
```

```
ethereal 0.9.3, with GTK+ 1.2.10, with GLib 1.2.10, with libpcap 0.6, with libz
1.1.3, with UCD SNMP 4.2.4
```

Script done on Thu Apr 17 15:08:31 2003

n. Start Ethereal - Figure 16 - Red Hat Ethereal Screen

```
[root@localhost root]# ethereal
```

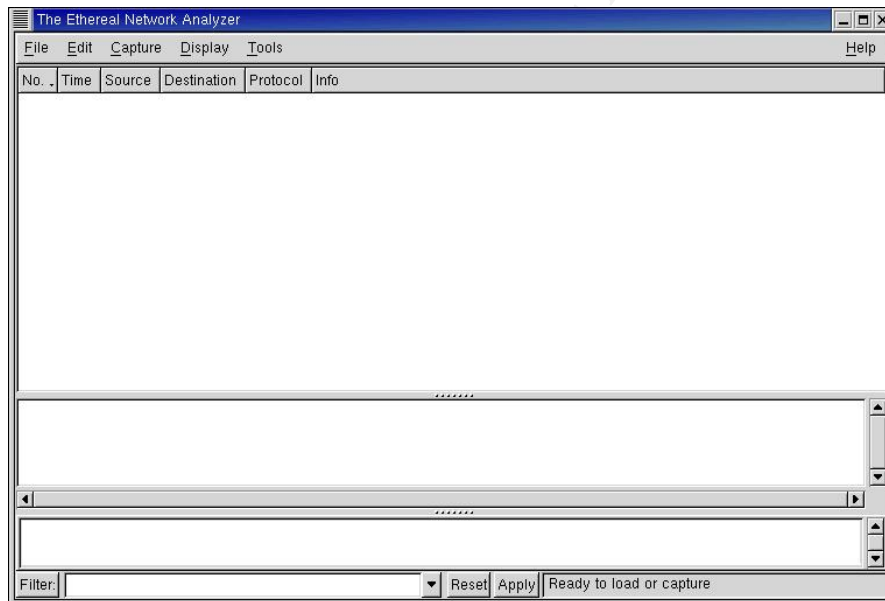


Figure 16 - Red Hat Ethereal Screen

- i. Select the Capture → Start Menu Option
- ii. Configure as illustrated in Figure 17 - Red Hat Capture Options, and select OK

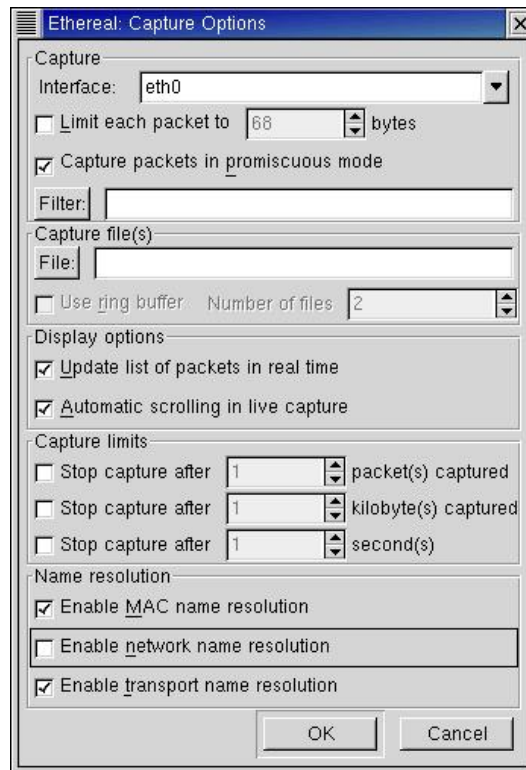


Figure 17 - Red Hat Capture Options

2. Windows Host

- a. Login with the Administrator account
- b. Ensure that the Windows host is in the correct timezone (GMT+10)
- c. Configure the Windows host to synchronize time with the Red Hat host - Figure 18 - Windows Time Synchronization

```
H:\response_kit\win2k_xp>net time /SETSNTP:192.168.0.1
The command completed successfully.
```

Figure 18 - Windows Time Synchronization

- d. Validate the following system information - Figure 19 - Windows System Information, Figure 20 - Windows System Information (Cont)

```

H:\response_kit\win2k_xp>uname -a
CYGWIN_NT-5.0 SAS-ITL-ISS02 1.3.3(0.46/3/2) 2001-09-12 23:54 i686 unknown

H:\response_kit\win2k_xp>mount
c: on /cygdrive/c type user (textmode,nomount)
d: on /cygdrive/d type user (textmode,nomount)
h: on /cygdrive/h type user (textmode,nomount)

H:\response_kit\win2k_xp>ipconfig /all

Windows 2000 IP Configuration

        Host Name . . . . . : sas-itl-iss02
        Primary DNS Suffix . . . . . :
        Node Type . . . . . : Broadcast
        IP Routing Enabled. . . . . : No
        WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix . :
        Description . . . . . : Intel EtherExpress(TM) PRO/10 <ISA M
ode>
        Physical Address. . . . . : 00-A0-C9-67-5D-8C
        DHCP Enabled. . . . . : No
        IP Address. . . . . : 192.168.0.2
        Subnet Mask . . . . . : 255.255.255.0
        Default Gateway . . . . . :
        DNS Servers . . . . . :

```

Figure 19 - Windows System Information

Note – Additional VMware NAT and host only network information is not displayed. In this test environment VMware is using the network bridging mode.

```

H:\response_kit\win2k_xp>netstat -an

Active Connections

    Proto Local Address          Foreign Address         State
    TCP    0.0.0.0:135             0.0.0.0:0               LISTENING
    TCP    0.0.0.0:445             0.0.0.0:0               LISTENING
    TCP    0.0.0.0:1025            0.0.0.0:0               LISTENING
    TCP    0.0.0.0:1027            0.0.0.0:0               LISTENING
    TCP    192.168.0.2:139         0.0.0.0:0               LISTENING
    TCP    192.168.182.1:139      0.0.0.0:0               LISTENING
    TCP    192.168.207.1:139      0.0.0.0:0               LISTENING
    UDP    0.0.0.0:135             *:*:
    UDP    0.0.0.0:445             *:*:
    UDP    0.0.0.0:1026            *:*:
    UDP    192.168.0.2:137         *:*:
    UDP    192.168.0.2:138         *:*:
    UDP    192.168.0.2:500        *:*:
    UDP    192.168.182.1:137      *:*:
    UDP    192.168.182.1:138      *:*:
    UDP    192.168.182.1:500      *:*:
    UDP    192.168.207.1:137      *:*:
    UDP    192.168.207.1:138      *:*:
    UDP    192.168.207.1:500      *:*:

```

Figure 20 - Windows System Information (Cont)

e. Start Ethereal

© SANS

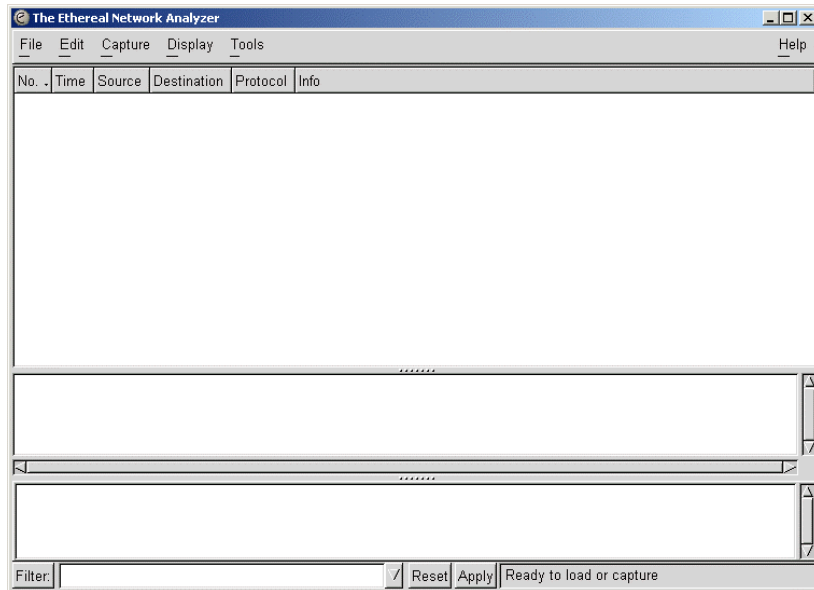


Figure 21 - Windows Ethereal

- i. Select the Capture → Start Menu Option

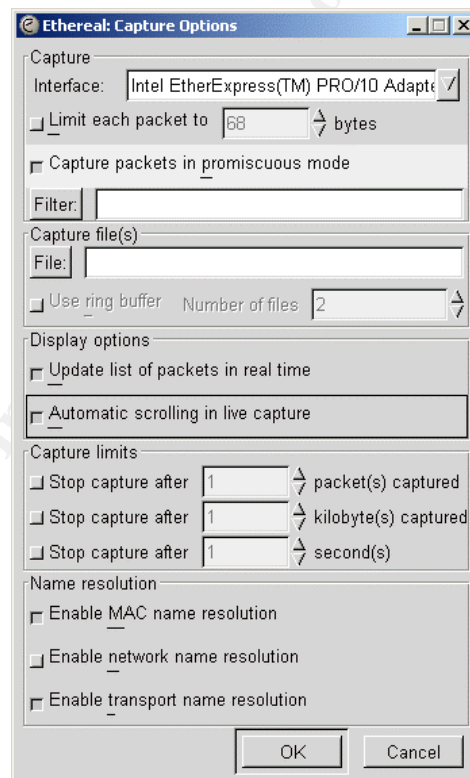


Figure 22 - Windows Ethereal Capture Options

- ii. Configure as illustrated in Figure 22 - Windows Ethereal Capture Options, and select OK

3. Knoppix Host

- a. Boot the Knoppix host from CD
- b. Once the X interface has loaded, open a Terminal window
- c. Set the root password

```
[knoppix@tty1[knoppix]$ sudo su
[root@tty1[knoppix]# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

- d. Set the timezone of the host – set timezone to Australia → New South Wales – most locations

```
[root@tty1[knoppix]# cd /tmp
[root@tty1[tmp]# tzselect
```

- e. Set the IP address for the Knoppix host

```
[root@tty1[tmp]# ifconfig eth0 192.168.0.3 netmask 255.255.255.0 broadcast
192.168.0.255
```

- f. Set the Knoppix host to synchronize its time from the Red Hat host

```
[root@tty1[tmp]# ntpdate -v 192.168.0.1
18 Apr 07:23:24 ntpdate[6430]: ntpdate 4.1.0 Mon Mar 25 23:39:50 UTC 2002
(2)
17 Apr 15:19:15 ntpdate[6430]: step time server 192.168.0.1 offset -
57849.404316 sec
```

- g. Validate the date synchronization

```
[root@tty1[tmp]# date
Thu Apr 17 15:19:18 EST 2003
```

- h. Edit the /etc/syslog.conf file

- i. Add the following line – This will enable the Knoppix host to send syslog messages to the Red Hat host

```
*.* @192.168.0.1
```

- ii. Restart the syslog daemon

```
[root@tty1[tmp]# /etc/init.d/syslogd restart
Stopping system log daemon: syslogd.
Starting system log daemon: syslogd.
```

- i. Validate the following system information

```
[root@tty1[tmp]# uname -a
Linux Knoppix 2.4.20-xfs #1 SMP Mit Mär 26 15:37:36 CET 2003 i686 Intel(R)
Pentium(R) 4 CPU 1.50GHz GenuineIntel GNU/Linux
```

```
[root@tty1[tmp]# mount
/dev/root on / type ext2 (rw)
/dev/cdrom on /cdrom type iso9660 (ro)
/dev/cloop on /KNOPPIX type iso9660 (ro)
/dev/shm on /ramdisk type tmpfs (rw,size=201376k)
usb on /proc/bus/usb type usbdevfs (rw,devmode=0666)
automount(pid296) on /mnt/auto type autofs
(rw,fd=6,pgrp=296,minproto=2,maxproto=4)
```

```
[root@tty1[tmp]# ifconfig -a
eth0  Link encap:Ethernet HWaddr 00:02:A5:CB:59:FE
      inet addr:192.168.0.3 Bcast:192.168.0.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:3051 errors:0 dropped:0 overruns:0 frame:0
      TX packets:2998 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:2077821 (1.9 MiB) TX bytes:349238 (341.0 KiB)
      Interrupt:20 Base address:0x1000

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:13 errors:0 dropped:0 overruns:0 frame:0
      TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:884 (884.0 b) TX bytes:884 (884.0 b)
```

```
[root@tty1[tmp]# netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address      Foreign Address    State
tcp    0      0 0.0.0.0:68         0.0.0.0:*          LISTEN
tcp    0      0 0.0.0.0:6000       0.0.0.0:*          LISTEN
udp    0      0 0.0.0.0:514       0.0.0.0:*
<snip>
```

```
[root@tty1[tmp]# ethereal -v
ethereal 0.9.5, with GTK+ 1.2.10, with GLib 1.2.10, with libpcap 0.6, with libz
1.1.4, without UCD SNMP
```

j. Mount the local floppy drive – double click the Floppy disk icon on the desktop

k. Copy the sn.zip file into the working directory and validate the files

```
[root@tty1[tmp]# cp /mnt/floppy/sn.zip .
[root@tty1[tmp]# unzip sn.zip
Archive: sn.zip
inflating: sn.dat
extracting: sn.md5
```

```
[root@ttyp1[tmp]# md5sum sn.dat
0e954f43fd73f56e812a7285f32e41d3 sn.dat
```

```
[root@ttyp1[tmp]# cat sn.md5
0e954f43fd73f56e812a7285f32e41d3 sn.dat
```

l. Prepare the sn.dat file for execution

```
[root@ttyp1[tmp]# chmod a+x sn.dat
```

m. Start Ethereal

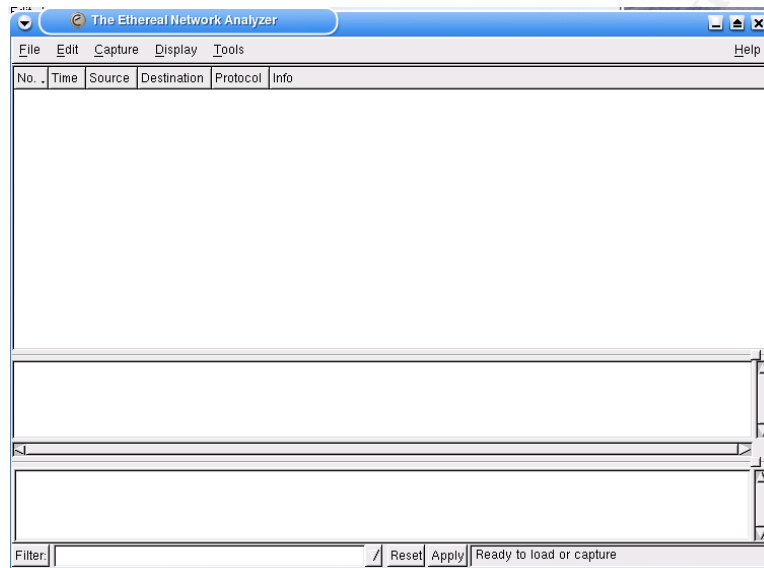


Figure 23 - Knoppix Ethereal

- i. Select the Capture → Start Menu Option
- ii. Configure the capture options as illustrated in Figure 24 - Knoppix Ethereal Capture Options, and select OK

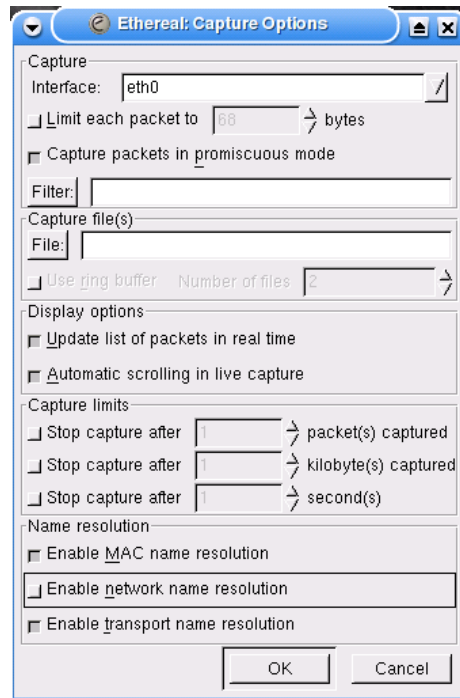


Figure 24 - Knoppix Ethereal Capture Options

- n. Start sn.dat

```
[root@tty1[tmp]# ./sn.dat eth0
ADMSniff priv 1.0 in libpcap we trust !
credits: ADM, mel , ^pretty^ for the mail she sent me
```

Data Generation

- i. FTP Connection

- a. From the Knoppix host, open another Terminal window and create an FTP session to the Red Hat host as the user andrew.
- b. Initiate the following commands;
 - ls
 - ls -a
 - get bash_history
 - get .bash_history
 - bye

- ii. Telnet Connection

- a. From the Knoppix host, create a telnet session to the Red Hat host, again as the user andrew
- b. Initiate the following commands;

- la -s
- ls -a
- cat /etc/passwd
- this is not a commend
- exit

- iii. Stop all the Ethereal Instances – i.e. Figure 25 - Button to stop Ethereal capture

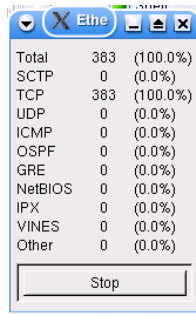


Figure 25 - Button to stop Ethereal capture

- iv. Save the captured Ethereal data to local files, Select File → Save
 - a. Ensure that the file is labelled, and placed in the appropriate directory as specified *Standards Used* section of this document.

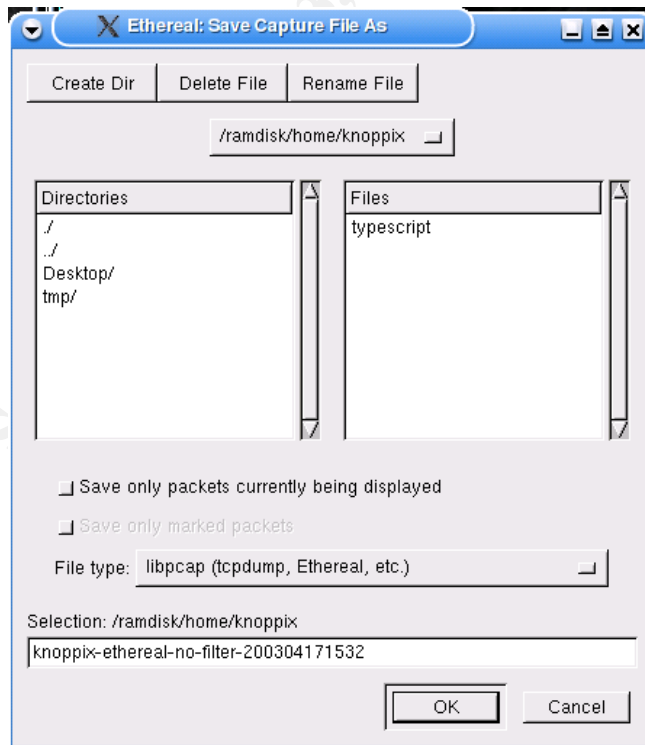


Figure 26 - Saving an Ethereal capture file

- v. Stop the sn.dat instance, i.e. CTRL-C the process
- vi. Archive results
 - a. All the Ethereal captures, sn.dat output, and script typescript files should have md5sums calculated for the files, and then the files should be collected from each of the hosts to be backed up and burnt to CD.
 - b. It is important to backup these files before analysis begins on them, so that the originals are not tainted.

Results

i. FTP Command

Script started on Thu Apr 17 15:29:39 2003

```
[knoppix@ttyp4[tmp]$ ftp 192.168.0.1
Connected to 192.168.0.1.
220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready.
Name (192.168.0.1:knoppix): andrew
331 Password required for andrew.
Password:
230 User andrew logged in. Access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for directory listing.
total 0
226 Transfer complete.
ftp> ls -a
200 PORT command successful.
150 Opening ASCII mode data connection for directory listing.
total 64
drwx----- 2 500 500 4096 Mar 27 19:40 .
drwx----- 2 500 500 4096 Mar 27 19:40 ..
-rw----- 1 500 500 67 Mar 29 09:09 .bash_history
-rw-r--r-- 1 500 500 24 Mar 27 19:06 .bash_logout
-rw-r--r-- 1 500 500 191 Mar 27 19:06 .bash_profile
-rw-r--r-- 1 500 500 124 Mar 27 19:06 .bashrc
-rw-r--r-- 1 500 500 854 Mar 27 19:06 .emacs
-rw-r--r-- 1 500 500 118 Mar 27 19:06 .gtkr
226 Transfer complete.
ftp> get bash_history
local: bash_history remote: bash_history
200 PORT command successful.
550 bash_history: No such file or directory.
ftp> get .bash_history
local: .bash_history remote: .bash_history
```

200 PORT command successful.
150 Opening BINARY mode data connection for .bash_history (67 bytes).
226 Transfer complete.
67 bytes received in 0.04 secs (1.7 kB/s)
ftp> bye
221-You have transferred 67 bytes in 1 files.
221-Total traffic for this session was 917 bytes in 1 transfers.
221-Thank you for using the FTP service on localhost.localdomain.
221 Goodbye.

ii. Telnet Command

[knoppix@ttyp4[tmp]\$ telnet 192.168.0.1
Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^['.
[SSL not available]
Password:
Login incorrect

login: andrew
Password:
Last login: Thu Apr 17 15:29:56 from 192.168.0.3
[andrew@localhost andrew]\$ la -s
bash: la: command not found
[andrew@localhost andrew]\$ ls -a
.bash_history .bash_logout .bash_profile .bashrc .emacs .gtkr
andrew@localhost andrew]\$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
mailnull:x:47:47:/var/spool/mqueue:/dev/null
rpm:x:37:37:/var/lib/rpm:/bin/bash
ntp:x:38:38:/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
gdm:x:42:42:/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/bin/false
ident:x:98:98:ident user:/sbin/nologin

Sn.dat Output - The_l0gz

- 68 -

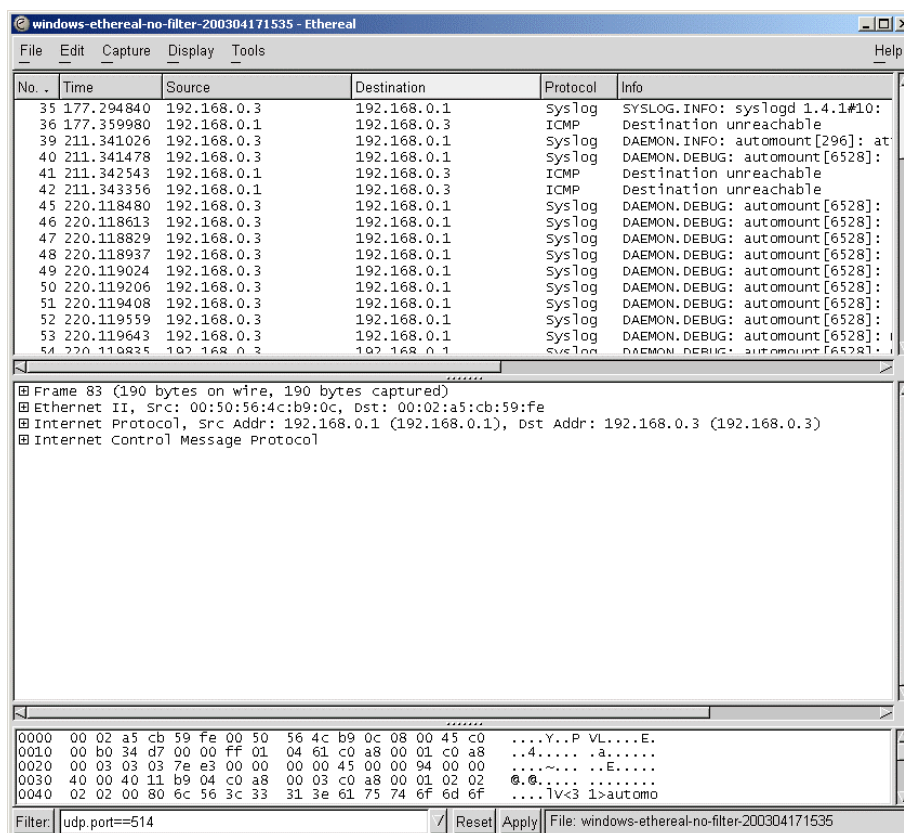


Figure 27 - Ethereal screen with captured data

Analysis

Each of the Ethereal screens will have a large amount of traffic on the screen, such as in Figure 27 - Ethereal screen with captured data, and each screen will show different data. The next task is to dissect this data into more meaningful elements. However, before any further breakdowns are down, there should be a brief explanation of this screen.

As can be seen, there are six columns across the screen (six columns is the default display). These can all be changed / moved around if required. The No. column shows the packet number allocated by ethereal during a session. If the first packet captured is allocated 1, the next is allocated 2 etc.

The Time column by default shows the seconds since the beginning of the capture. This can be adjusted to a more meaningful time by selecting the Display → Options menu, and selecting Date and time of day. This will display the time in the format of YYYY-MM-DD hh:mm:ss.uuuu. See Figure 28 - Ethereal Time and Date Display Options

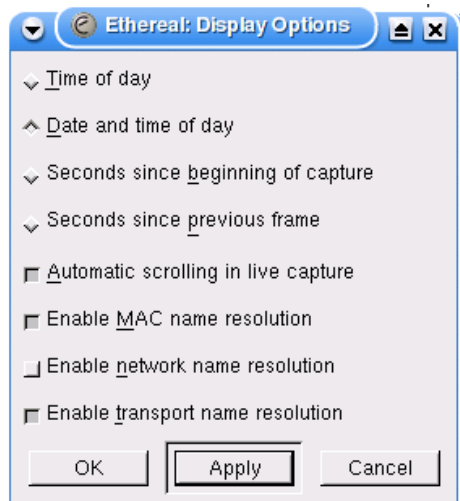


Figure 28 - Ethereal Time and Date Display Options

The source and destination columns are the IP addresses involved in the communications, and the protocol column is the transport or application layer protocol used. You can see that Ethereal only shows FTP in the protocol field when specific FTP commands are being executed / returned.

Finally, the info field shows further details of the raw protocol. In this case, if a specific FTP command is executed, then the raw FTP command will be displayed (i.e. USER Andrew), if there is just data being transferred in the FTP session, then the raw TCP data will be displayed.

In comparing all of the data from each Ethereal capture, it is immediately apparent that each screen has captured different data. See Figure 27 - Ethereal screen with captured data, Figure 29 - Knoppix Capture and Figure 30 - Red Hat Capture. In particular, the Knoppix host shows only FTP and Telnet connections, the Windows host shows syslog and ICMP unreachable messages, while the Red Hat host shows Windows broadcasts.

The syslog messages are from the Knoppix host, and the ICMP messages are Destination Unreachable (Port Unreachable) messages which are sent from the Red Hat host since it is not listening on the syslog port.

The reason for the difference in the display of the Ethereal captures is because each capture was started at a different time and because of the physical network setup. For instance, the Windows host would see all traffic directed to and from both it and the Red Hat host, while the Knoppix host would only see traffic directed directly to or from itself.

As can be seen from the captures, there is a lot of unwanted information. To address the analysis objectives, filtering of this information is required.

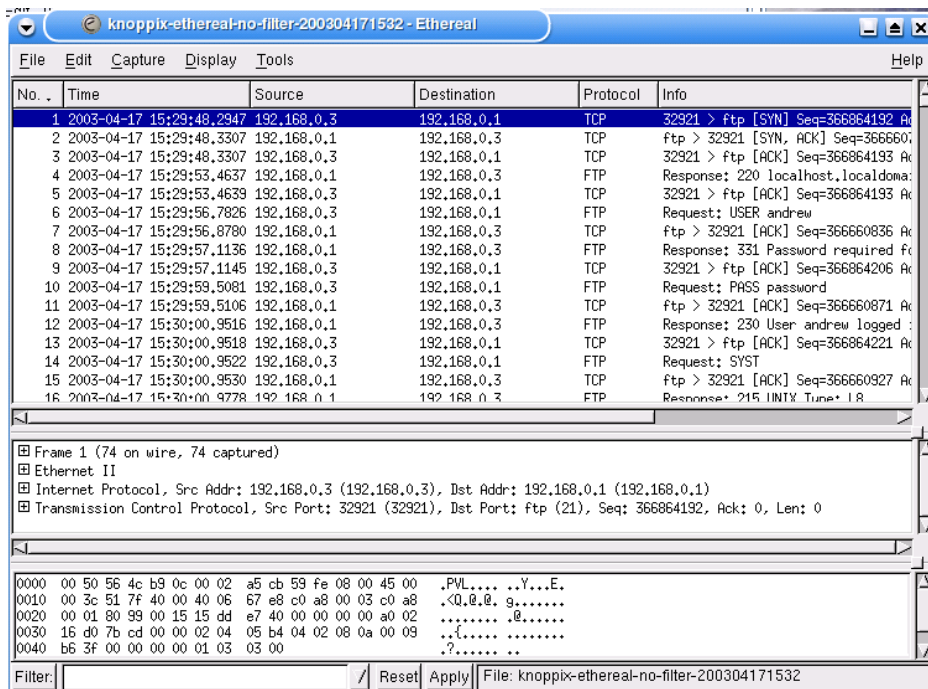


Figure 29 - Knoppix Capture

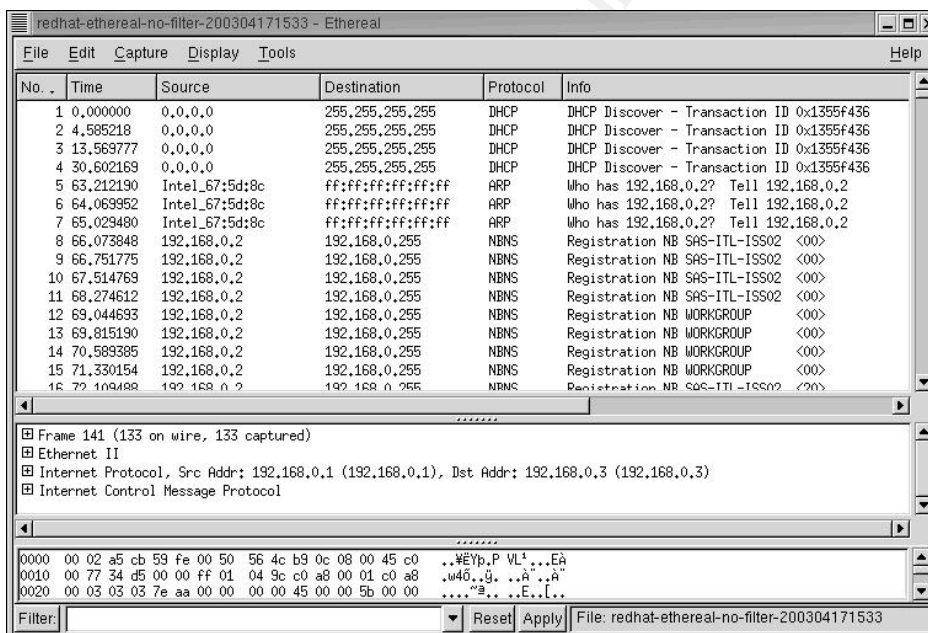


Figure 30 - Red Hat Capture

Analysis Objective 1

Validate that the FTP and Telnet connections were captured by sn.dat

You can see from the output that sn.dat (above) breaks out the two flows in the TCP communication. I.e. the data which traveled from 192.168.0.3 to 192.168.0.1 and the reverse of this. This is represented in the sn.dat file by the line similar to `--=[192.168.0.3:32921 --> 192.168.0.1:21]=--`.

This allows for an easy breakdown of the data flows, and the following comparisons can be made with the actual data sent and received - Table 1 - Validation of sn.dat capture

As can be seen from the table, an accurate match up between the commands executed and the sn.log capture file can be made. With only a minimal understanding of the actual FTP commands, ie LIST for ls³⁸, it can be concluded that the sn.dat file has accurately captured the FTP traffic.

Of interest is that the sn.dat file has not captured the FTP data connection on 20/tcp. This is why the output of the directory listing and the contents of the file retrieved file are not reported by sn.dat.

A similar match up can be completed with the Telnet traffic stream passing back to the client. The traffic travelling to the server was not captured in a human readable format. This is analyzed under Analysis Objective 3.

Table 1 - Validation of sn.dat capture

Actual Commands	Sn.dat output file capture	
<code>[knoppix@ttyp4[tmp]\$ ftp 192.168.0.1</code>	<code>--=[192.168.0.3:32921 --> 192.168.0.1:21]=--</code>	<code>--=[192.168.0.1:21 --> 192.168.0.3:32921]=--</code>
Connected to 192.168.0.1. 220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready. Name (192.168.0.1:knoppix): andrew 331 Password required for andrew. Password: 230 User andrew logged in. Access restrictions apply. Remote system type is UNIX. Using binary mode to transfer files.?.....B.^.....D.._k..... .._k USER andrew.....`.....` PASS password1..a.....1..a. SYST7..a.....C..a. PORT 192,168,0,3,128,154..... ...C..c.....C..c. LIST.....K..c.....Q..c.....c. PORT 192,168,0,3,128,155.....cc.^.....?....._k..B 220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready.....`.....` 331 Password required for andrew.....a.....a..... 230 User andrew logged in. Access restrictions apply.....a....1.....a....1 215 UNIX Type: L8.....c....C 200 PORT command successful.....c....C 150 Opening ASCII mode data connection for directory listing.....c....K

³⁸ A full set of commands can be found at Network Working Group RFC: 959, "File Transfer Protocol (FTP)", Oct 1985 URL: <http://war.jgaa.com/ftp/rfc/rfc959.txt> (26 May 2003)

Once you have found part of the FTP stream to investigate, Ethereal has a powerful feature called *Follow TCP Stream*. This automatically creates a filter that will show a single, complete TCP session.

In order to compare each of the Ethereal captures, this filter functionality will be used to extract out the single FTP connection in each of the capture. To use the *Follow TCP Stream* function, right mouse click on a packet from the FTP stream to trace, and select *Follow TCP Stream* - Figure 31 - Follow TCP Stream

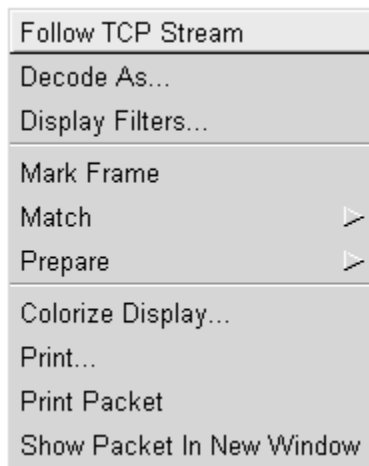
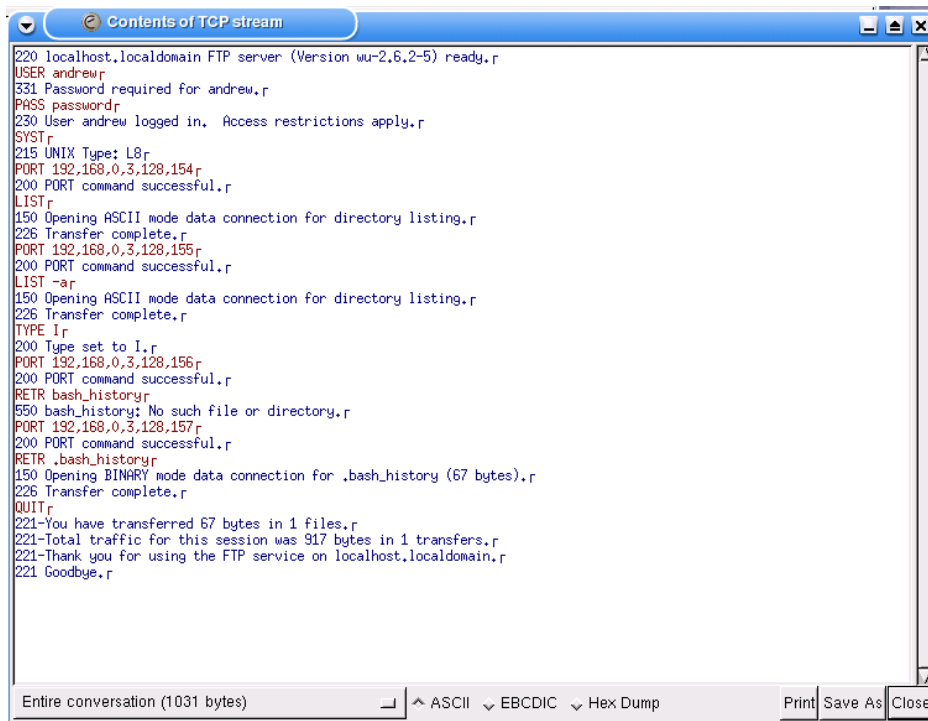


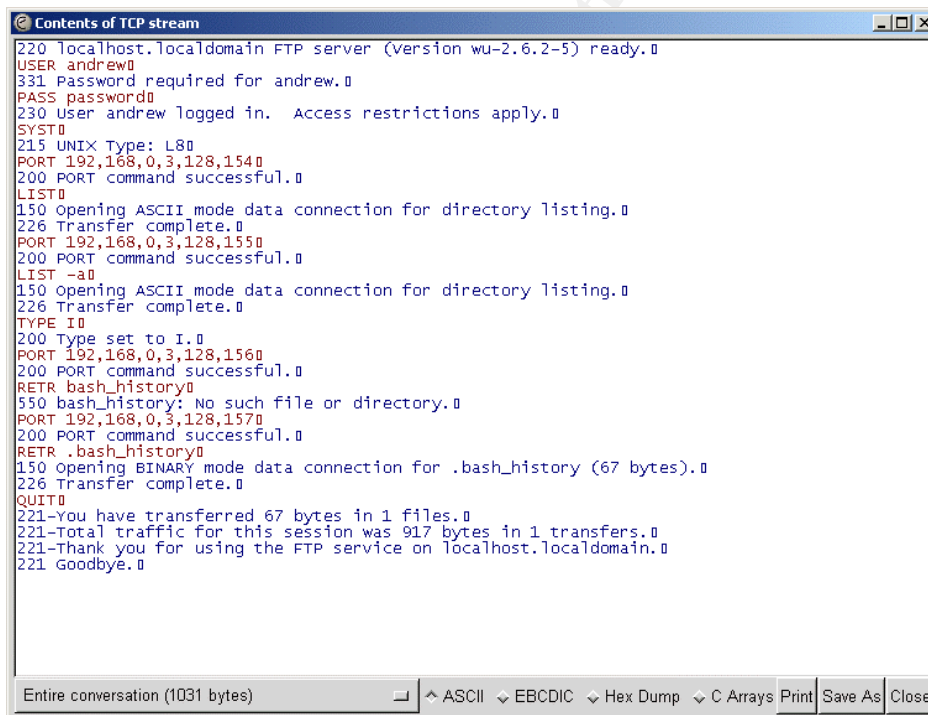
Figure 31 - Follow TCP Stream

The result on each of the Ethereal captures is as follows in Figure 32 - Knoppix capture of FTP connection, Figure 33 - Windows capture of FTP connection and Figure 34 - Red Hat capture of FTP connection

A screenshot of a Knoppix terminal window titled "Contents of TCP stream". The window displays a text-based FTP session log. The log shows a user named 'andrew' logging in with a password, followed by directory listings and file transfers. The session ends with a 'Goodbye' message. The bottom of the window has a status bar with "Entire conversation (1031 bytes)" and buttons for "Print", "Save As", and "Close".

```
220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready.r
USER andrew.r
331 Password required for andrew.r
PASS password.r
230 User andrew logged in. Access restrictions apply.r
SYST.r
215 UNIX Type: L8.r
PORT 192,168,0,3,128,154.r
200 PORT command successful.r
LIST.r
150 Opening ASCII mode data connection for directory listing.r
226 Transfer complete.r
PORT 192,168,0,3,128,155.r
200 PORT command successful.r
LIST -a.r
150 Opening ASCII mode data connection for directory listing.r
226 Transfer complete.r
TYPE I.r
200 Type set to I.r
PORT 192,168,0,3,128,156.r
200 PORT command successful.r
RETR bash_history.r
550 bash_history: No such file or directory.r
PORT 192,168,0,3,128,157.r
200 PORT command successful.r
RETR .bash_history.r
150 Opening BINARY mode data connection for .bash_history (67 bytes).r
226 Transfer complete.r
QUIT.r
221-You have transferred 67 bytes in 1 files.r
221-Total traffic for this session was 917 bytes in 1 transfers.r
221-Thank you for using the FTP service on localhost.localdomain.r
221 Goodbye.r
```

Figure 32 - Knoppix capture of FTP connection

A screenshot of a Windows terminal window titled "Contents of TCP stream". The window displays a text-based FTP session log, identical to the one in Figure 32. The session shows a user named 'andrew' logging in, directory listings, and file transfers. The bottom of the window has a status bar with "Entire conversation (1031 bytes)" and buttons for "Print", "Save As", and "Close".

```
220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready.
USER andrew
331 Password required for andrew.
PASS password
230 User andrew logged in. Access restrictions apply.
SYST
215 UNIX Type: L8
PORT 192,168,0,3,128,154
200 PORT command successful.
LIST
150 Opening ASCII mode data connection for directory listing.
226 Transfer complete.
PORT 192,168,0,3,128,155
200 PORT command successful.
LIST -a
150 Opening ASCII mode data connection for directory listing.
226 Transfer complete.
TYPE I
200 Type set to I.
PORT 192,168,0,3,128,156
200 PORT command successful.
RETR bash_history
550 bash_history: No such file or directory.
PORT 192,168,0,3,128,157
200 PORT command successful.
RETR .bash_history
150 Opening BINARY mode data connection for .bash_history (67 bytes).
226 Transfer complete.
QUIT
221-You have transferred 67 bytes in 1 files.
221-Total traffic for this session was 917 bytes in 1 transfers.
221-Thank you for using the FTP service on localhost.localdomain.
221 Goodbye.
```

Figure 33 - Windows capture of FTP connection

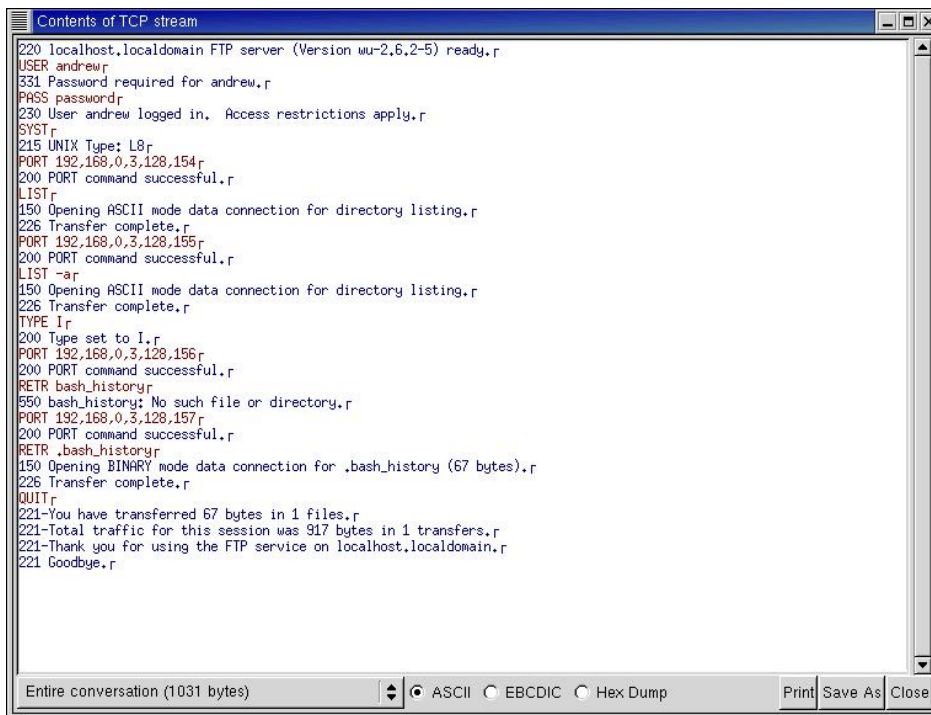


Figure 34 - Red Hat capture of FTP connection

As can be seen from the captures, each is identical except for the representation of the end of line character.

From these it can be concluded that the three versions of Ethereal have all captured the same FTP data.

A similar investigation of the Telnet data stream also shows an exact match.

Analysis Objective 2

Validate that the FTP and Telnet connections were captured by Ethereal

In comparing the FTP streams with the actual commands entered, a similar tabular comparison can be created. Again, it is fairly simple to see how Ethereal matches up with the commands actually executed from the FTP client.

Interestingly, Ethereal has represented the data in a slightly different order where the *200 port command successful* command is issued from the Red Hat (FTP Server) host. The FTP client displays the command message prior to the *150 Opening ASCII mode data*, while Ethereal shows these commands as arriving from the server after the *150 Opening ASCII mode data* and other status information.

The *200 port command successful* is a notification from the server that the FTP-Data connection on 20/tcp was successfully. It is this data channel where the

results of the client query will be found. For instance, when the `ls -a` (or `LIST -a`) was issued, the separate data channel will be where the actual file list will be found.

The FTP client appears to displays this data in a different order from how it arrives across the network. The Ethereal capture is illustrating that the FTP connection is performing correctly as specified in the FTP RFC³⁹.

Despite this interesting oddity, it still can be seen that the data does match what was actually sent across the network.

In addition to capturing the actual commands and command parameters for the FTP connection, Ethereal has also captured the FTP data channel. From this the results of the commands, such as the `LIST -a` can be found. Figure 35- FTP Data Ethereal Capture shows a *Follow TCP Stream* capture for the FTP-Data channel when the `LIST -a` command was issued. As can be seen from the figure, this matches exactly the directory listing displayed in the FTP client.

The conclusion is that Ethereal has accurately captured the TCP stream.

Table 2 - Validation of Ethereal Capture

Actual Commands	Ethereal Capture
<pre>[knoppix@tty4[tmp]\$ ftp 192.168.0.1</pre>	
<pre>Connected to 192.168.0.1. 220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready. Name (192.168.0.1:knoppix): andrew 331 Password required for andrew. Password: 230 User andrew logged in. Access restrictions apply. Remote system type is UNIX. Using binary mode to transfer files.</pre>	<pre>220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready. USER andrew 331 Password required for andrew. PASS password 230 User andrew logged in. Access restrictions apply. SYST 215 UNIX Type: L8 PORT 192,168,0,3,128,154 200 PORT command successful.</pre>
<pre>ftp> ls 200 PORT command successful. 150 Opening ASCII mode data connection for directory listing. total 0 226 Transfer complete.</pre>	<pre>LIST 150 Opening ASCII mode data connection for directory listing. 226 Transfer complete. PORT 192,168,0,3,128,154 200 PORT command successful.</pre>
<pre>ftp> ls -a 200 PORT command successful. 150 Opening ASCII mode data connection for directory listing. total 64 drwx----- 2 500 500 4096 Mar 27 19:40 . drwx----- 2 500 500 4096 Mar 27 19:40 ..</pre>	<pre>LIST -a 150 Opening ASCII mode data connection for directory listing. 226 Transfer complete. TYPE I 200 Type set to I. PORT 192,168,0,3,128,154 200 PORT command successful.</pre> <p>See Figure 35- FTP Data Ethereal Capture for the capture of this output</p>

³⁹ Network Working Group RFC: 959, "File Transfer Protocol (FTP)", Oct 1985 URL: <http://www.w3.org/Protocols/rfc959/> (26 May 2003)

<pre> -rw----- 1 500 500 67 Mar 29 09:09 .bash_history -rw-r--r-- 1 500 500 24 Mar 27 19:06 .bash_logout -rw-r--r-- 1 500 500 191 Mar 27 19:06 .bash_profile -rw-r--r-- 1 500 500 124 Mar 27 19:06 .bashrc -rw-r--r-- 1 500 500 854 Mar 27 19:06 .emacs -rw-r--r-- 1 500 500 118 Mar 27 19:06 .gtkr 226 Transfer complete. ftp> get bash_history local: bash_history remote: bash_history 200 PORT command successful. 550 bash_history: No such file or directory. ftp> get .bash_history local: .bash_history remote: .bash_history 200 PORT command successful. 150 Opening BINARY mode data connection for .bash_history (67 bytes). 226 Transfer complete. 67 bytes received in 0.04 secs (1.7 kB/s) ftp> bye 221-You have transferred 67 bytes in 1 files. 221-Total traffic for this session was 917 bytes in 1 transfers. 221-Thank you for using the FTP service on localhost.localdomain. 221 Goodbye. </pre>	<p>capture of this output</p> <pre> RETR bash_history 550 bash_history: No such file or directory. PORT 192,168,0,3,128,157 200 PORT command successful. RETR .bash_history 150 Opening BINARY mode data connection for .bash_history (67 bytes). 226 Transfer complete. QUIT 221-You have transferred 67 bytes in 1 files. 221-Total traffic for this session was 917 bytes in 1 transfers. 221-Thank you for using the FTP service on localhost.localdomain. 221 Goodbye. </pre>
--	--

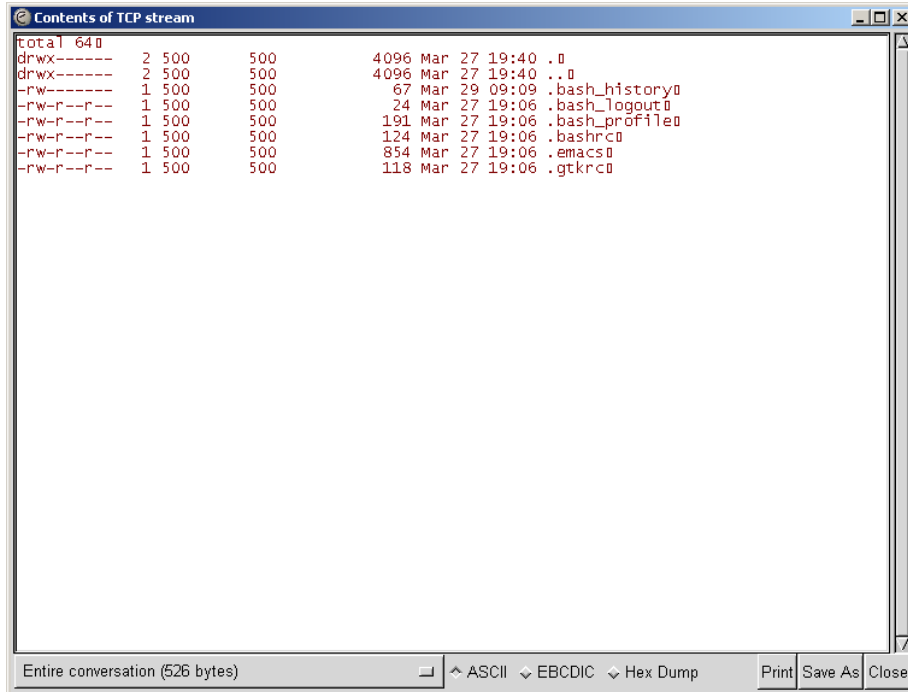


Figure 35- FTP Data Ethereal Capture

In comparing the Telnet stream, Figure 36 - Ethereal capture of Telnet session, it again can be seen that the capture from Ethereal matches that of the actual Telnet session. Unlike the FTP capture, the Telnet capture has more information passed over the network which the Telnet client does not show.

For instance, there is X display information, ie *USER.knoppix.DISPLAY.Knoppix:0.0*, and extra shell information which the Telnet client does not show the user. The ethereal display also shows the characters in the command as being repeated, ie echoed, which again is not seen by a user.

The representation of white space by Ethereal also is different by that displayed by the Telnet client in the command shell. This is to be expected.

Overall, it is still clear that Ethereal has captured all the data sent and received by the user using the Telnet client.

```
Contents of TCP stream
..%.#.....!.....#.....P.....38400,3840
0...#.Knoppix:0.0...USER.knoppix.DISPLAY.Knoppix:0.0....xterm.....Password: password.
Login incorrect
login: aannddrreeww.
Password: password.
Last login: Thu Apr 17 15:29:56 from 192.168.0.3
.]0;andrew@localhost:~.[andrew@localhost andrew]$ llaa --ss.
bash: lla: command not found
.]0;andrew@localhost:~.[andrew@localhost andrew]$ llss --ad.a
.[00m.[01;34m..[00m..[01;34m...[00m..[00m.bash_history.[00m..[00m.bash_logout.[00m..[00m.bash_profile.[
00m..[00m.bashrc.[00m..[00m.emacs.[00m..[00m.gtkrc.[00m
.[m.]0;andrew@localhost:~.[andrew@localhost andrew]$ ccaatt //eett.c/ppaass..swd.
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
mailnull:x:47:47:/var/spool/mqueue:/dev/null
rpm:x:37:37:/var/lib/rpm:/bin/bash
ntp:x:38:38:/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/x11/fs:/bin/false
gdm:x:42:42:/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nsd:x:28:28:NSD Daemon:/bin/false
ident:x:98:98:ident user:/sbin/nologin
radvd:x:75:75:radvd user:/bin/false
andrew:x:500:500:/home/andrew:/bin/bash
.]0;andrew@localhost:~.[andrew@localhost andrew]$ ttthhiiss iiss nnoott aa cocommmmeennttd..d.
bash: this: command not found
.]0;andrew@localhost:~.[andrew@localhost andrew]$ eexxiitt.
logout
.[H.[2]
```

Figure 36 - Ethereal capture of Telnet session

Analysis Objective 3

Validate that the Ethereal instances accurately represent what the sn.dat file captured

Like the sn.dat output, Ethereal can break the network flows of information into two. By selecting the *Entire Conversation* pull down menu from the *Follow TCP Stream* window, an individual network flow can be selected. Ie *192.168.0.1.ftp à 192.168.0.3:32921 (825 bytes)*. The output of selecting this is shown in Figure 37 - Single Ethereal Flow.

By retrieving each flow, it will be easy to compare the Ethereal captures with those from sn.dat.

As can be seen in Table 3 - sn.dat and Ethereal capture comparison, aside from white space representation and other stray characters in the sn.dat output, the substance is identical for data traveling in both directions.

Table 3 - sn.dat and Ethereal capture comparison

sn.dat Output	Ethereal Capture
--[192.168.0.1:21 --> 192.168.0.3:32921]==--	192.168.0.1.ftp → 192.168.0.3:32921
.....^.....?....._k...B 220 localhost.localdomain FTP server (Version uu-2.6.2-5) ready.....`.....`.....	220 localhost.localdomain FTP server (Version uu-2.6.2-5) ready.r
331 Password required for andrew.....a.....a.....	331 Password required for andrew.r
230 User andrew logged in. Access restrictions apply.....a.....1.....a.....1	230 User andrew logged in. Access restrictions apply.r
215 UNIX Type: L8.....c.....C	215 UNIX Type: L8r
200 PORT command successful.....c.....C	200 PORT command successful.r
150 Opening ASCII mode data connection for directory listing.....c.....K	150 Opening ASCII mode data connection for directory listing.r
226 Transfer complete.....c.....	226 Transfer complete.r
200 PORT command successful.....c.....	200 PORT command successful.r
150 Opening ASCII mode data connection for directory listing.....c.....	150 Opening ASCII mode data connection for directory listing.r
226 Transfer complete.....f.....	226 Transfer complete.r
200 Type set to I.....f.....	200 Type set to I.r
200 PORT command successful.....f.....	200 PORT command successful.r
550 bash_history: No such file or directory.....j.....	550 bash_history: No such file or directory.r
200 PORT command successful.....j.....	200 PORT command successful.r
150 Opening BINARY mode data connection for .bash_history (67 bytes).....j.....!	150 Opening BINARY mode data connection for .bash_history (67 bytes).r

226 Transfer complete.....l....Z 221-You have transferred 67 bytes in 1 files.....l....Z 221-Total traffic for this session was 917 bytes in 1 transfers.....l....Z 221-Thank you for using the FTP service on localhost.localdomain.....l....Z 221 Goodbye.....l....Z.....l....Z.	226 Transfer complete.r 221-You have transferred 67 bytes in 1 files.r 221-Total traffic for this session was 917 bytes in 1 transfers.r 221-Thank you for using the FTP service on localhost.localdomain.r 221 Goodbye.r
--=[192.168.0.3:32921 --> 192.168.0.1:21]==	192.168.0.3:32921 → 192.168.0.1:ftp
.....?.....B.^.....D..k....._k USER andrew..... PASS password1..a.....1..a. SYST7..a.....C..a. PORT 192,168,0,3,128,154.....C..c.....C..c. LIST.....K..c.....Q..c.....c. PORT 192,168,0,3,128,155.....c.....c. LIST -a.....c.....c.....c. TYPE I.....f.....f. PORT 192,168,0,3,128,156.....f.....f. RETR bash_history.....f.....f. PORT 192,168,0,3,128,157.....j.....j. RETR .bash_history!..j.....!..j.....Z..j. QUIT.....Z..l.....Z..l.....Z..l.....Z..l.....Z..l..	USER andrewr PASS passwordr SYST r PORT 192,168,0,3,128,154r LIST r PORT 192,168,0,3,128,155r LIST -ar TYPE I r PORT 192,168,0,3,128,156r RETR bash_historyr PORT 192,168,0,3,128,157r RETR .bash_historyr QUIT r



```
--=[ 192.168.0.1:23 --> 192.168.0.3:32926 ]--  
.....n....s...o...s...#...'..o.....%.....!.."..  
.....#..'.....o.....o.....o.....o.....o.....o.....Password: .....o~..  
.....o...4.....o...H.....o...W.....o...d.....o...t.....o...|.....o.....o.....o.....o.....p...  
Login incorrect...login:  
.....q@...a....qO...n.....qX...d.....qk...r.....q{...+e.....q...w.....q...J.....q...J  
Password: .....q.....q.....r.....r).....r5.....r9.....rJ.....rZ.....r_.....  
Last login: Thu Apr 17 15:29:56 from 192.168.0.3.....s1..  
.j0;andrew@localhost:~.....s:...}[andrew@localhostandrew]$ .....s.....s.....l.....s...  
a.....s.....7.....s.....7.....s.....C.....s.....C-.....s.....Os.....s.....[.....s.....\bash: la: command not  
found.....s....._]0;andrew@localhost:~.....s.....`[andrew@localhost andrew]$  
.....tT.....l.....tp...s.....t|.....t.....t.....t.....t.....t.....a.....t.....t.....[00m.[01;34m..[00m  
.[01;34m..[00m .[00m.bash_history.[00m .[00m.bash_logout.[00m .[00m.bash_profile.[00m  
.[00m.bashrc.[00m .[00m.emacs.[00m  
.[00m.gtkrc.[00m.....t.....*. [m.....t.....+]0;andrew@localhost:~.....t.....+[andrew@localhost andrew]$  
.....uw.....c.....u.....a.....u.....t.....u.....  
.....u...?.....u.../?l.....u...S.....u...Se.....u...ft.....u...rc/.v3...p.....v;...a.....vE.....s.....vV.....sw  
d.....vs.....vx.....root:x:0:0:root:/root:/bin/bash..bin:x:1:1:bin:/bin:/sbin/nologin..daemon:x:2:2:dae  
mon:/sbin:/sbin/nologin..adm:x:3:4:adm:/var/adm:/sbin/nologin..lp:x:4:7:lp:/var/spool/lpd:/sbin/nologi  
n..sync:x:5:0:sync:/sbin/bin/sync.shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown..halt:x:7:0:halt:/sb
```

```

in:/sbin/halt..mail:x:8:12:mail:/var/spool/mail:/sbin/nologin..news:x:9:13:news:/var/spool/news:..uucp
:x:10:14:uucp:/var/spool/uucp:/sbin/nologin..operator:x:11:0:operator:/root:/sbin/nologin..games:x:1
2:100:games:/usr/games:/sbin/nologin..gopher:x:13:30:gopher:/var/gopher:/sbin/nologin..ftp:x:14:50
:FTP User:/var/ftp:/sbin/nologin..nobody:x:99:99:Nobody:/sbin/nologin..vcsa:x:69:69:virtual
console memory
owner:/dev:/sbin/nologin..mailnull:x:47:47:/var/spool/mqueue:/dev/null..rpm:x:37:37:/var/lib/rpm:/bi
n/bash..ntp:x:38:38:/etc/ntp:/sbin/nologin..rpc:x:32:32:Portmapper RPC
user:/sbin/nologin..xfs:x:43:43:X Font
Server:/etc/X11/fs:/bin/false..gdm:x:42:42:/var/gdm:/sbin/nologin..rpcuser:x:29:29:RPC Service
User:/var/lib/nfs:/sbin/nologin..nfsnobody:x:65534:65534:Anonymous NFS
User:/var/lib/nfs:/sbin/nologin..nscd:x:28:28:NSCD Daemon:/bin/false..ident:x:98:98:pident
user:/sbin/nologin..radvd:x:75:75:radvd
user:/bin/false..andrew:x:500:500:/home/andrew:/bin/bash.....vz.....]0;andrew@localhost:~.....v
z.....[andrew@localhost andrew]$ .....x.....nt.....x.....h.....x.....x.....i.....x.....x.....s.....y.....
.....y2.....i.....yC.....yM.....s.....yN.....yP.....y.....n.....yk.....o.....yr.....t.....y....+
.....y.....a.....y.....F
.....y.....R.....y.....[c.....y.....eo.....y.....jm.....y.....wm.....y.....e.....y.....n.....y.....t.....z.....z.....
.....z.....z.....d.....zC.....zE.....bash: this: command not
found.....zl.....]0;andrew@localhost:~.....zJ.....[andrew@localhost andrew]$
.....{k.....9e.....{.....Lx.....{.....Xi.....{.....dt.....{.....{.....logout.....{.....[H.[2J.....{.....

```

```

.....#.%.!..".....#......Password: 
Login incorrect
Login: andrew
Password: 
Last login: Thu Apr 17 15:29:56 from 192.168.0.3
]0;andrew@localhost:~.[andrew@localhost andrew]$ la -s0
bash: la: command not found
]0;andrew@localhost:~.[andrew@localhost andrew]$ ls -a0
.[00m.[01;34m..[00m..[01;34m...[00m..[00m.bash_history.[00m..[00m.bash_logout.[00m..[
00m.bash_profile.[00m..[00m.bashrc.[00m..[00m.emacs.[00m..[00m.gtkrc.[00m
.[m.]0;andrew@localhost:~.[andrew@localhost andrew]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
mailnull:x:47:47:/var/spool/mqueue:/dev/null
rpm:x:37:37:/var/lib/rpm:/bin/bash
ntp:x:38:38:/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
gdm:x:42:42:/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS user:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/bin/false
ident:x:98:98:pident user:/sbin/nologin
radvd:x:75:75:radvd user:/bin/false
andrew:x:500:500:/home/andrew:/bin/bash
]0;andrew@localhost:~.[andrew@localhost andrew]$ this is not a comment. .d0
bash: this: command not found
]0;andrew@localhost:~.[andrew@localhost andrew]$ exit0
Logout
.[H.[2J

```

Figure 38 - Ethereal capture of Telnet flow to 192.168.0.3

The reason that the `/etc/passwd` is not fully displayed is because the TAB key was used in the telnet client to get a command / filename completion. This Ethereal example is quite illustrative for showing that Ethereal has captured all the white space characters entered during the session. The rectangle after each command is the end of line character, the characters after the `/et` and `pas` are tab characters, and the rectangle in the *command* word shows a backspace was entered.

From the previous objective, Analysis Objective 2 - Validate that the FTP and Telnet connections were captured by Ethereal, it was shown that Ethereal accurately captured the Telnet connection. It is therefore concluded that sn.dat has not accurately captured (or displayed) this Telnet flow.

```
--=[ 192.168.0.3:32926 --> 192.168.0.1:23 ]==  
.....\n.....s.n.%.....!.".'.#.....o.....o.....o...P....  
.38400,38400...#.Knoppix:0.0...'USER.knoppix.DISPLAY.Knoppix:0.0.....xterm.....o.....  
o.....o.....  
..o.p.....4.o~a.....H.o.s.....W.o.s.....d.o.w.....t.o.o.....|.o.r.....o.d.....o.....  
p.....p.a.....q@.....q@n.....qO.....qOd.....qX.....qXr.....qk.....+.qke.....+.{  
.....:q(w.....q.....J.q.....J.q.....K.q.....q.p.....q.a.....q.s.....r.s.....f.w.....f  
)o.....r5r.....r9d.....rJ.....r_.....}.s1.....s:.....s:l.....s.....s.a.....s.....7..s.  
.....B.s.....C.s.-  
.....l.s.....O.s.s.....P.s.....[.s.....\.s....._.s.....`s.....`s.....s.l.....tT.....tTs..  
.....tp.....tp.....tl.....tl-  
.....t.....ta.....t.....t.....t.....*.t.....+.t.....+.t.....,t.....tc.....uw.....uw  
a.....u.....ut.....u.....u.  
.....u.....?..u/. J.u.....S.u.e.....S.u.....f.u.t.....g.u.....r.u.....y.u.....u.p.....v3  
.....v3a.....v;.....v;s.....vE.....vE.....vV.....vV.....vs.....vx.....vZ.....vZ  
.....n.vzt.....n.x.....x.h.....x.....xi.....x.....xs.....x.....x.  
.....y.....yi.....y2.....y2s.....yM.....yM  
.....yP.....yPn.....y_.....y_o.....yk.....ykt.....yr.....+.yr  
.....+.y......y.a.....;.y.....F.y.  
.....G.y.....R.y.c.....[.y.o.....e.y.....e.y.....j.y.m.....k.y.....w.y.m.....w.y.....y.e.....  
.y.....yn.....y.....yt.....y.....y.....z.d.....z.....z.....zC.....zE..  
.....zl.....zJ.....9.zJe.....9.{k.....L.{kx.....N.{.....X.{i.....Y.{.....d.{t.....e.{.....{  
.....{.....{.....{.....{.....{
```

```

..%......!..".'.....#....P..... 38400,38400....#.knoppix:0.0....USER.knoppix
x.DISPLAY.knoppix:0.0.....xterm.....password0.andrew0.password0.la -s0.ls -a0.cat /et
.pas.0.this is not a comment000.exit0.

```

Figure 39 = Ethereal capture of Telnet flow to 192.168.0.1

Conclusion

The tests conducted verified the following;

- Validation that the FTP and Telnet connections were captured by sn.dat, although there was difficulty reading the Telnet connection, and the FTP data connection was not captured by sn.dat
- Validation that the FTP and Telnet connections were captured by Ethereal
- Validation that the Ethereal instances accurately represent what the sn.dat file captured
- Validation that the various Ethereal instances match each other with respect to the data captured

The tests were a success, and the results obtained were mostly as expected.

The tests and the surrounding preparation indicate that the Ethereal tool could be used in a forensic investigation, and would make a useful tool for the presentation of network traffic to a court.

Ethereal could be used in an incident response or incident analysis situation, since it can read in live network traffic, or read previously created libpcap log files. Ethereal would be used to dissect and present the network traffic in a human readable form. The tool successfully translates the network protocols into a format, which someone who is not a network expert can interpret and understand.

Given that Ethereal does not alter data files, the integrity of the files analysed is maintained.

In order to make Ethereal forensically sound, the following recommendations are made;

- compile Ethereal from source code
- maintain md5 check sums of both the original source code and the binaries produced
- consider the use of Ethereal in a known environment, such as the Knoppix environment. Build your own Knoppix environment from known source
- Ensure that machine clocks are synchronized before using Ethereal for live captures

- ensure that data (libpcap) files saved from Ethereum are handled to maintain their integrity
- record all procedures for how data was extracted from Ethereum, such information must include filters applied, display options set, and time zones used.
- document procedures and findings on paper, preferably on bound, numbered pages with ink. Entries should be dated and initialed

Presentation

The presentation of any computer generated information to a court is a delicate, yet essential element in computer crimes prosecutions. Because of the nature of computer logs, traditional evidence laws have had to accommodate the nature of both the raw (human unfriendly) log evidence, as well as the experts required to interpret the information.

In 1900, Judge Leonard Hand observed that *"no one will deny that the law should in some way effectively use expert knowledge wherever it will aid in settling disputes. The only question is as to how to do it best."*⁴⁰

In Australian Law, there are two main goals for submitting evidence, that being to gain admissibility for the evidence, and then to give the evidence weight in the court. Courts rely on evidence to identify facts that establish the truth about what has happened. Legal representatives of each party attempt to show the court that their facts are correct.

When presenting evidence to the court, there are three main issues that must be addressed⁴¹;

- how to adduce (that is, put to the court) evidence of the fact
- whether the court will permit the evidence to be given (i.e. admissibility)
- weighting of the evidence - how much importance the court will give to it in reaching its decision.

Under Australian Commonwealth and State evidence Acts⁴², computer logs (such as those produced by Ethereum) would be admissible under the definition of a document⁴³. The term document is defined in the evidence acts to include any

⁴⁰ Wood, Justice James "Expert Witnesses – The new era" June 2001 URL: http://www.agd.nsw.gov.au/sc%5Csc.nsf/pages/Wood_June2001 (26 May 2003)

⁴¹ "IT Laws in Australia" URL: <http://www-staff.it.uts.edu.au/~jim/cit2/site/law/LawAUMainFrame.htm> (26 May 2003)

⁴² Evidence Act (Cwth) 1995 URL: <http://scaleplus.law.gov.au/html/pasteact/2/1182/top.htm> (26 May 2003)

⁴³ National Archives of Australia "Records in Evidence", 1998 URL: http://www.naa.gov.au/recordkeeping/overview/evidence/records_in_evidence.htm (26 May 2003)

record of information, and includes anything on which there are marks, figures, symbols or perforations having a meaning for persons qualified to interpret them; or anything from which sounds, images or writings can be reproduced with or without the aid of anything else.

Obviously, the legislation only permits the acceptance of such computer logs, and certainly does not guarantee acceptance. The most important element after the evidence acceptance is weighting of the evidence.

“The admissibility/weight distinction sometimes becomes quite unclear and particularly so in relation to novel scientific evidence. Most of the evidence that comes from modern computers is likely to fall into that category. In England the assessment of this is largely a matter of weight and cases are lost if an expert and a judge appear to be telling a jury that they must accept a finding.”⁴⁴

This statement illustrates that the presentation of information from a computer tool, such as Ethereal, is going to have to ensure that the tools output can be accepted and validated by the court, and not with sole reliance on a technical specialist telling the court what an output means. It is thus important for a technical expert to provide context, explanations and interpretations on the information presented, and to let the court decide on the facts which it supports.

Some important elements to consider for the presentation of such evidence include the following;

- authenticity - that is, specifically linked to the circumstances and persons alleged
- accuracy - free from any reasonable doubt about the quality of procedures used to collect the material,
- analysed - the material is appropriate and necessary and can be produced by someone who can explain the evidence.
- completeness - tells within its own terms a complete story of particular set of circumstances or events
- concise – the complex and complete evidence can be expressed in an accurate and concise executive style summary, such that a court can understand the what is being proved by the evidence presented

In addition to these, there are a number of other particular elements related to the presentation of computer generated output which will increase the weight of a piece of evidence. These include the following;

⁴⁴ Sommer, Peter “Intrusion Detection as Evidence”, March 2000 URL: <http://www.bcs.org.uk/lac/ids.htm> (26 May 2003)

- Chain of Custody – a complete history of who has had access to the evidence, who has been the custodian of the evidence, where, why and how the evidence has been transported or interfered with.
- Transparency of forensic method – the methods of forensic analysis have been exposed so that a third party expert could re-create the results from the given evidence
- Correlated Evidence – a single stream of log file evidence is unlikely to be highly regarded. Multiple independent streams of evidence should be used to corroborate each other. The context of pieces of data is also important.
- Output generated from a tool requires a full description of the tool, how it was configured and how it was operated. Again, the context of the tool in a network or computer system is going to be important
- In accordance with the best evidence rule, the best evidence is going to be the original log files created on a computer. Even if these require another process to perform analysis and make them easier to read, the raw log should be maintained and presented.
- Expert opinions – It should be ensured that the person presenting the evidence be an expert in the area. The evidence is more likely to have a higher weight in the court if the expert has a high degree of industry experience with the tool / data being presented, and that the expert is highly regarded in the industry

The requirement to present Ethereal evidence to the court may occur for a number of reasons.

- To capture network traffic (specific or non specific to an individual or investigation) and these captures contain evidence as to the accused network activities
- To interpret raw network traffic files, such as libpcap captured by another tool such as tcpdump
- To illustrate the effect of behavior of a network tool, such as this example with sn.dat

In any of the above, the following would need to be presented to the court.

- Raw logs – preferably the entire disk which contains the logs would be presented to the court. It would be essential that the raw (binary) libpcap files which Ethereal displays be provided.

- Original checksums of all logs / images and how these checksums were originally obtained, and how they can still be validated
- Chain of evidence of disks and log files, including checksums of files and binaries used
- Ethereal version, configuration, network details in which the capture or test was performed. If the version of Ethereal used was built from source code, it would be an advantage to submit this as well. The source code to a tool can give more weight to the credibility and accuracy of the tool.
- Detailed descriptions of the procedures on how the capture was taken, or the test was performed. In addition, the submission of any notes taken during the analysis would be important. Original hand written notes taken with pen and paper are going to be highly regarded
- Details of how well regarded and accepted Ethereal is in the computer networks industry

As for interpreting the Ethereal output to the court, given that a central function of Ethereal is to display network traffic in a human readable format, the presentation of its results in a forensic investigation would not be too difficult.

Although there is the above mentioned requirement to submit the complete raw libpcap log files, these are not going to be conducive to presenting to a court. More effective would be the presentation of specific screens from Ethereal which directly relate the demonstration of the existence of a certain fact.

In the case of the sn.dat verification, the most useful screens would be the Follow Stream interfaces which were presented before. These show in almost plain text the application layer communications on the network. With only minimal explanations of the FTP protocol, Ethereal could show the flows between machines and a human readable format for FTP communications.

Of course, the presentation of specific screens is of little weight unless the complete procedures of how that screen was extracted were also presented to the court, and that these procedures did actually recreate the data shown.

It would be also essential to show the date and time elements of the Ethereal screen, and to explain any discrepancy in the time displayed and times recorded in other evidence. Similarly, it would be important to highlight the IP addresses involved and to discuss how these matched the network in which the capture was occurring.

All the evidence must be correlated together to illustrate the facts of which the evidence show.

It is important to remember that in presenting forensic evidence to the court, the aim is to produce a fact, and to build a story line of how the facts fit together. It is therefore important to draw specific attention to the fact which is being illustrated by the particular Ethereal screen, and to also draw specific attention to the timings of events.

Finally, a word of warning for court presentation of Ethereal evidence. Under the *Telecommunications (Interceptions) Act 1979*, the use of Ethereal to capture data may only be allowed in specific circumstances. This act prohibits the interception of any form of communication except for broadcast radio, so it would be worth while to ensure that the evidence collected has been gained in a legal manner before it is presented to the court.

© SANS Institute 2003, Author retains full rights

Part 3 – Legal Issues of Incident Handling

Question Setting

You are a system administrator for an Internet Service Provider that provides Interact access for paying customers. You receive a telephone call from a law enforcement officer who informs you that an account on the system was used to hack into a government computer. He asks you to verify the activity by reviewing your logs and determine if your logs reflect whether or not the activity was initiated there or from another upstream provider. You review your logs and can only determine a valid user account logged in via a dialup account during the period of the suspicious activity.

Question 1

What, if any, information can you provide to the law enforcement officer over the phone during the initial contact?

In determining Internet Service Provider obligations to law enforcement bodies, there are two key pieces of Australian Commonwealth legislation which need to be referenced. This legislation being the *Telecommunications Act 1997* and the *Telecommunications (Interception) Act 1979*.

Both Acts can be found at <http://scaleplus.law.gov.au>

It is assumed that the ISP mentioned in the above question will satisfy the service provider definition under s86(a) *Telecommunications Act 1997* that a service provider is a carriage service provider. A carriage service provider is defined in s87.

It is also assumed that the hack into the Government computer is a breach of a Commonwealth criminal offence, such as those found in the recent *Cybercrime Act 2001*⁴⁵ or *Crimes Act 1914*⁴⁶.

The legislation sets out two basic premises which need to be balanced for the disclosure of information.

On one side, Part 13 of the *Telecommunications Act 1997* makes it an offence for an ISP and its employees to disclose any information which comes into its possession in the course of its ISP business, where the information relates to:

⁴⁵ *Cybercrime Act (Cwth) 2001* URL: <http://scaleplus.law.gov.au/cgi-bin/download.pl?/scale/data/pasteact/3/3486> (26 May 2003)

⁴⁶ *Crimes Act (Cwth) 1914* URL: <http://scaleplus.law.gov.au/cgi-bin/download.pl?/scale/data/pasteact/0/28> (26 May 2003)

- The content or substance of a communication carried by the ISP
- Carriage services supplied or intended to be supplied by the ISP
- The affairs or personal particulars of another person

The relevant exceptions (Part 13 Division 3) to these rules include;

- where the disclosure is reasonably necessary for the enforcement of the criminal law
- where the disclosure is required or is otherwise authorized under a warrant or under law

In contrast, Part 14 of the *Telecommunications Act 1997* obligates ISPs to give officers and authorities of the Commonwealth, States and Territories such help which is reasonably necessary for the enforcement of the criminal law, or the enforcement of laws imposing pecuniary penalties s313(3)(c).

The Australian Communications Authority (ACA)⁴⁷ has interpreted this to include the request for log files, such as back up tapes showing details of a subscribers Internet session⁴⁸.

It should be noted that the two telecommunications Acts make a specific difference between requests for information regarding the content of communications such as the content of emails.

Access by agencies to the content of an Internet communication in transit will amount to an interception and can only be authorized under the *Telecommunications (Interception) Act 1997*. Subsequently, a law enforcement agency will have to apply for a specific interception warrant to access either stored content of communications, or to intercept the content of communications whilst in transit.

For requests outside those requiring a warrant, Part 13 of the *Telecommunications Act 1997* lists the requirement for making certified and uncertified requests for information.

The main requirements for a *certified* request are listed in s282;

- S282(3) - an authorized officer of a criminal law-enforcement agency has certified that the disclosure is reasonably necessary for the enforcement of the criminal law.

⁴⁷ <http://www.aca.gov.au>

⁴⁸ The ACA have produced a series of guideline documents and fact sheets specifically for ISPs. An example fact sheet can be found at Australian Communications Authority "Internet Service Providers and Law Enforcement and National Security" Nov 2000 URL: http://www.aca.gov.au/consumer_info/fact_sheets/industry_fact_sheets/fsi13.pdf (26 May 2003)

- S282(6)(a),(b) - the disclosure does not relate to contents or substance carried by a service provider

Additional requirements are also placed on the *certified* request;

- S282(7) - A certificate under subsection (3), (4) or (5) must comply with such requirements as are determined in writing by the ACA. These can be found at http://www.aca.gov.au/aca_home/legislation/radcomm/determinations/telecom/282-2.htm
- S282(8) - Before making a determination under subsection (7), the ACA must consult the Privacy Commissioner.
- S282(9) - A certificate under subsection (3), (4) or (5) may be: in written form; or in electronic form (for example, electronic mail).
- S282(10) Defines some of the terms used in the above sections, such as Authorised Officer which is a *Senior officer who is specifically authorized to issue such certificates*, Criminal law enforcement agencies who this applies to, such as the *Australian Federal Police, Australian Crime Commission* etc

For an *uncertified* request, s282(1) can be used where by ISP must be satisfied that the disclosure of the information is reasonably necessary for the enforcement of criminal law, protection of public revenue or enforcement of a law imposing a pecuniary penalty. In contrast to the certified request, the onus is on the ISP to make the determination as to whether the information can be given out.

The *Australian Communications Authority* has indicated in their *Telecommunications and Law Enforcement manual Chapter 3 - The concept of reasonably necessary assistance*, para 3.12⁴⁹ that a set of procedures and safeguards are needed to be followed to ensure the request has passed through known procedures within the requesting agency, and the service provider can be comfortable that it was for reasonable necessity. Detailed requirements for the written request as listed under para 3.12.

In summary;

- where there is a request for information from a law enforcement agency (which is not ASIO - S283),
- and this information is not for the content of communications,
- and this is not a situation of serious or an imminent threat to life (s287)

⁴⁹ Australian Communications Authority "Telecommunications and Law Enforcement" July 1998
URL: http://www.aca.gov.au/aca_home/licensing/radcomm/about_radcomms_licensing/leac.pdf
(26 May 2003)

there will be two basic ways in which an agency can request information to assist in criminal investigations.

1. Under s282(1) of the *Telecommunications Act* 1997, an agency must submit a written request which sets out the offence being investigated, and the onus is on the service provider to make a decision as to whether this is reasonably necessary.
2. Under s282(3) of the *Telecommunications Act* 1997, an agency can obtain a certificate from one of its senior officials that this disclosure is reasonably necessary. The onus is on the senior official to determine if the action is reasonably necessary.

A useful flow chart of an ISPs obligations can be found on page 30 of the ACIF Industry Code for the Provision of assistance to national security, enforcement and government agencies⁵⁰. See Figure 40 - ACIF flow chart of ISP obligations

For an ISP to determine reasonably necessary the *Australian Communications Authority* has correlated some resources which will help define this, and these can be found in *Chapter 3 - The concept of reasonable necessity assistance, Internet Service Providers Interception Obligations*⁵¹.

There is also commentary from the *Human Rights and Equal Opportunity Commission Privacy Commissioner* regarding Advice from the *Federal Privacy Commissioner* about Reasonably Necessary Uses or Disclosures. In this notice, the recommendations are that

“Organizations are not obliged to disclose information under the sections of the TA if they are not satisfied that the use or disclosure is reasonably necessary ... Organizations are urged to adopt a cautious and accountable approach to the obligation to assess reasonable necessity”

Finally, the ACA also note in *Chapter 5 - Privacy and record keeping requirements* that in some cases a verbal request will be made for information pending a certificate to be issued under s282(3). The recommendation made in para 5.7 is that the current practice is for the service provider to gather the information in anticipation of the certificate, but not to release the information until the certificate is supplied. This of course assumes that the information will not cease to exist or degrade because of the delay.

⁵⁰ Australian Communications Industry Forum, “Industry Code – Provision of Assistance to National Security, Enforcement and Government Agencies” 2002 URL: http://www.aca.gov.au/telcomm/industry_codes/codes/c537b.pdf (26 May 2003)

⁵¹ Australian Communications Authority “Telecommunications and Law Enforcement” July 1998 URL: http://www.aca.gov.au/aca_home/licensing/radcomm/about_radcomms_licensing/leac.pdf (26 May 2003)

In conclusion, given the above question of giving a law enforcement officer information over the phone, it would appear that this would not be possible. Even if the officer had obtained a certificate, the current practice is not to disclose the information until the certificate arrives. If there is no certificate, then a formal letter will need to be issued to the ISP in accordance with the details discussed above.

The risk of providing information would be to breach the privacy components of Part 13 of the *Telecommunications Act 1997*.

Further information discussing all the above mentioned processes can be found in the *Australian Communications Industry Forum – Industry Code – Provision of assistance to National Security, Enforcement, and Government Agencies; ACIF C537:2002* which can be found at

http://www.aca.gov.au/telcomm/industry_codes/codes/c537b.pdf

OBLIGATION TO PROVIDE ASSISTANCE TO LAW ENFORCEMENT AGENCIES

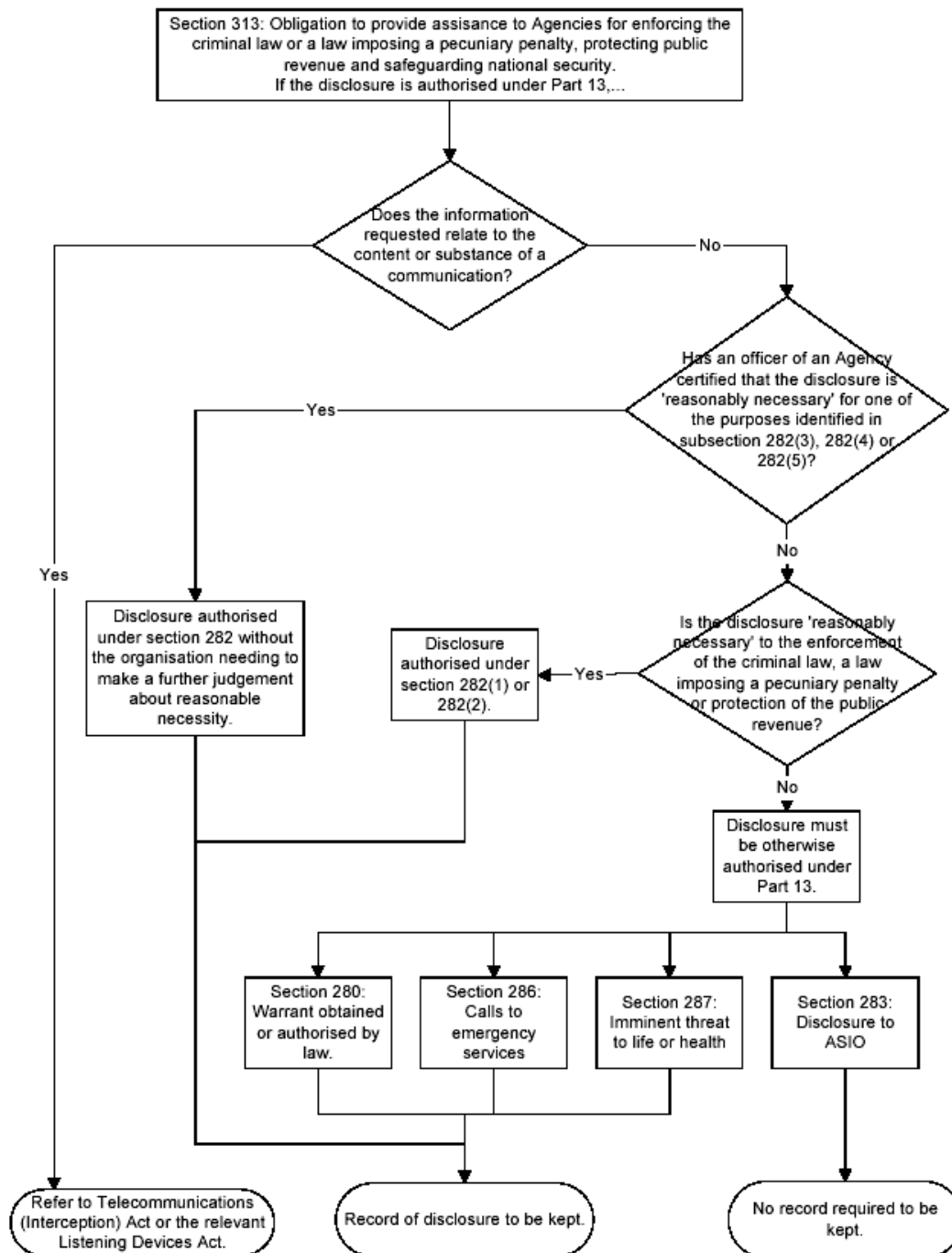


Figure 40 - ACIF flow chart of ISP obligations⁵²

⁵² Australian Communications Industry Forum, "Industry Code – Provision of Assistance to National Security, Enforcement and Government Agencies" 2002 URL: http://www.aca.gov.au/telcomm/industry_codes/codes/c537b.pdf (26 May 2003)

Question 2

What must the law enforcement officer do to ensure you preserve this evidence if there is a delay in obtaining any required legal authority?

None of the legislation or explanatory materials gave a definitive explanation for this situation. However, given that the law enforcement officer will have no access to the ISP systems until the certificate or written notice is received, the onus on preserving the evidence is on the ISP.

It is clear that the information is not allowed to degrade during this period, so the ISP will need to ensure that any information, such as the authentication logs, are maintained and accessible. In the ACA document previously referenced, Chapter 2 - *An overview of the obligations of telecommunications organizations to law enforcement and national security* para 2.17 stresses that the information relating to the request must be treated with a high level of confidentiality and sensitivity. It recommends that access to the information be limited to only those in the ISP which are providing assistance to the investigation. It is strongly recommended that one or two people within the service provider be empowered to receive request instructions and to act on them.

Reference could be made to international references, such as the recent *Digital Evidence in the Courtroom: A guide for preparing Digital Evidence for Courtroom Presentation – Master Draft Document*, Revised March 12, 2003, The National Center for Forensic Science. Section B of this document discusses suggested procedures and requirements for the Integrity, Discovery and Disclosure of Electronic Evidence⁵³.

Other external references for the preservation of data can be found in *The World Business Organizations Storage of traffic data for law enforcement Purposes*⁵⁴ document. It aims to provide

“A harmonized international approach which balances government, business and user interests is needed to ensure that storage of and access to traffic data for LEA purposes is adequate, effective and fair”

The article focuses on Traffic data, but this could easily extend to other log files. Under this document, the following pointers are given;

⁵³ The National Centre for Forensic Science “Digital Evidence in the Courtroom: A guide for preparing digital evidence for courtroom presentation – Master Draft Document” March 2003 URL: http://www.ncfs.org/DE_courtroomdraft.pdf (26 May 2003)

⁵⁴ Commission on E-Business, IT and Telecoms “Policy Statement – Storage of traffic for law enforcement purposes” Nov 2002 URL: http://www.iccwbo.org/home/news_archives/2002/stories/traffic%20data.pdf (26 May 2003)

- Data preservation should be favoured over data retention as less burdensome and costly to business and less harmful to public confidence.
- Traffic data must be defined explicitly and narrowly to exclude, for example, content data, the data created when individuals make financial transactions using communications devices, and other types of related data such as decryption keys.
- Data retention must be justified, proportionate and necessary for the purposes of investigating and prosecuting terrorism and other criminal activity only. The types and time periods of data to be retained should be kept to an absolute minimum.
- Access to traffic data should be limited to law enforcement agencies on production of a warrant or similar instrument, and for the purposes of investigating and prosecuting terrorism and other criminal activity.
- Private CSPs, e.g. those serving closed corporate user group customers and not the general public, should not be required to retain traffic data. Traffic data retention requirements for private CSPs would impose unnecessary obligations on organizations such as universities and non-profit institutions which do not offer services to the public

A recent publication from the *Australasian Centre for Policy Research - The Virtual Horizon: Meeting the Law Enforcement Challenges - Developing an Australasian law enforcement strategy for dealing with electronic crime*, Police Commissioners' Conference Electronic Crime Working Party 2000⁵⁵ had the following comment regarding the preservation of evidence by ISPs for law enforcement agencies;

“retaining and preserving evidence electronic is volatile and thus easily destroyed. Whilst legislation might be enacted to permit police to order the retention and preservation of evidence by third parties such as telecommunications and ISP operators, knowing exactly who should be issued with the order is likely to produce significant difficulties”

Given this comment, and the absence of other more recent commentary, it would appear that there is little formal advice regarding an ISPs obligations.

It should be noted that under S313(5) of the *Telecommunications Act 1997* a service provider (s313(6) and its employees and agents) will not be liable to an action of damages for acts done or omitted in good faith in performing their duties under s313, and s313(1)(b) places an obligation on service providers to do the providers best to prevent telecommunications networks and facilities from being used in, or in relation to the commission of offences against the laws of the Commonwealth or the States or Territories

⁵⁵ Australasian Centre for Policing Research “ The virtual horizon: Meeting the law enforcement challenges” 2000 URL: http://www.acpr.gov.au/pdf/ACPR134_1.pdf (26 May 2003)

Given the original *Telecommunications Act 1997* requirements that an ISP must give support to law enforcement agencies, and the above comment on liability and best effort and that law enforcement agencies are probably going to have to use this information for evidentiary purposes, it would assumed that the ISP must do all that is reasonable in the circumstances to ensure the data is preserved, secured and that its integrity is verifiable.

In this case, the following would be recommended;

- the relevant logs should be isolated to a secured location, where access controls are in place. A secondary copy of these should also be made and secured in a separate environment.
- a chain of custody register should be put into place for the information gathered
- the procedures for this extraction should be recorded
- the original log files should also be isolated, but not necessarily given to law enforcement. They should be kept separate in case additional information is requested. This will ensure that the privacy of other users is not compromised before the appropriate order is made for this information
- the relevant logs should have md5 checksums generated
- any analysis performed by the ISP should be done on copies of the logs, and not those logs which will be given to the law enforcement agency

The overall aim will be to ensure that the log files are not tainted such that they will lose their evidentiary value, or lose their original content, during the wait for law enforcement collection

Question 3

What legal authority, if any, does the law enforcement officer need to provide to you in order for you to send him you logs?

As discussed in the first question, under most circumstances, there are two ways in which information can be requested for the purposes of assisting authorities in the enforcement of criminal laws;

1. Under s282(1) of the *Telecommunications Act 1997*, an agency must submit a written request which sets out the offence being investigated, and the onus is on the service provider to make a decision as to whether this is reasonably necessary.
2. Under s282(3) of the *Telecommunications Act 1997*, an agency can obtain a certificate from one of its senior officials that this disclosure is reasonably necessary. The onus is on the senior official to determine if the action is reasonably necessary.

In the context of the s282(1) *uncertified* request, the ACA recommended practices specifies that the written request should contain the following features;

- be in writing and dated
- either on specifically designed forms containing the agency's logo or on letterhead
- individually signed/authorized by an officer or staff member of the agency
- priority is indicated
- specific identification of the service or services inquired about
- citing the offence being investigated (*the information is reasonably necessary for the investigation of an offence contrary to section ____ of the ____ Act of 19__*)
- a signed assurance that the information will be used only for the purpose for which it was sought and that it will be secured against unauthorized disclosure
- the agency's return fax number

Further information on these requirements can be found in paragraph 3.12, 3.14⁵⁶.

In the context of the s282(3) *certified* request, the ACA recommend that as long as the certificate appears authentic to the meet the requirements of the act, then the request should be acted on. The *LEAC Sub Committee* has the following recommendations regarding the identification of a valid certificate. The certificate should;

- identify the person making the certification and specify that the person is authorized
- identify the agency
- include the date of the issue of the certificate
- specify where the information is to be sent
- specify which provisions of the Act are being relied upon
- allude to the prohibition of secondary disclosure
- certify that each disclosure on the form is reasonably necessary, and
- specify the addressee (ie the CSP)

Further information on these requirements can be found in paragraph 3.6 - 3.11⁵⁷.

⁵⁶ Australian Communications Authority "Telecommunications and Law Enforcement" July 1998
URL: http://www.aca.gov.au/aca_home/licensing/radcomm/about_radcomms_licensing/leac.pdf
(26 May 2003)

⁵⁷ Australian Communications Authority "Telecommunications and Law Enforcement" July 1998
URL: http://www.aca.gov.au/aca_home/licensing/radcomm/about_radcomms_licensing/leac.pdf
(26 May 2003)

As for sending the logs to the officer, s314A *Telecommunications Act 1997* specified the requirements for the agreement of the delivery of the logs to the law enforcement body. It also addresses the situation for resolving disputes on how the logs can be delivered.

Question 4

What other investigative activity are you permitted to conduct at this time?

Like the reminder of the law on this area, there is little clarification. However, it would be assumed that under s313(1)(b), where a service provider is obliged to do their best to prevent telecommunications networks from being used in the commissioning of offences, that some further investigation would be permitted.

However, there is still the further limitations placed on service providers in the *Telecommunications (Interceptions) Act 1979*. In this act s7(1) prohibits the listening to or recording a communication without the knowledge of both parties.

For instance, on the face of this provision, it would prevent an ISP from running traffic capture utilities, or recording all an offenders Internet activities.

The exceptions provided where interception is permitted, are defined in s7(2). These include where it is reasonably necessary for the employee to do that act or thing in order to perform their duties effectively, including;

- (ii) the operation or maintenance of a telecommunications system
- (iii) the identifying or tracing of any person who has contravened or is suspected of having contravened or being likely to contravene a provision of the Crime
- s Act 1914

The courts may also take into account such matters as are specified by the regulations - s7(2A). This may include company policies and guidelines, as well as any relevant industry guidelines or industry codes.

In the given circumstances, it would appear that extra investigative activity will be permitted, as long as it can be shown that is reasonable in the circumstances to maintain the ISP and to meet the obligations to assist under s313. It would be important to stay within the bounds of any company policies regarding this sort of investigation, so as to not over step the reasonably necessity test, and be liable under s7(1) of the *Telecommunications (Interception) Act 1979*.

If further investigation is required, but is not conferred by the local security policy, and will require the interception of traffic, it would be strongly recommended to contact the relevant law enforcement agency, and negotiate with them to issue a

request for that information. At least that way, the liability is transferred from the ISP to the law enforcement agency.

Question 5

How would your actions change if your logs disclosed a hacker had gained unauthorized access to your system and at some point, created an account for him/her to use, and used that account to hack into the government system?

Overall, the same rules are going to apply. Under s313(b), the ISP would be obliged to notify and assist the law enforcement agency to prevent the commissioning of offences. However, because of the privacy restrictions under Part 13 of the Telecommunications Act 1997 direct information disclosure would not be possible until requests under s232 or a warrant was obtained for the disclosure of the information.

The obvious action to be taken in this circumstance is to follow the local ISPs Incident Response Plan / Policy or reference the local IT Security Policy and Guidelines. It is these documents which will allow for the suspension of accounts, the disconnection of a user etc.

It would certainly be recommended to take these actions to comply with s313(b), so as to prevent the further affect of the hacker on the Government systems. Under a liberal interpretation of the s232 requirements for a certified request, immediate actions could be taken to disclose information to a law enforcement agency, as long as there was an undertaking made by the law enforcement agent that the formal certified request would be forthcoming.

Because of the lack of case law deciding on the interpretation of the *reasonably necessity* and *do the providers best* tests, it is unclear how far a court would let a service provide intercept and investigate one of their users, and how much information could be disclosed without the formal request mechanisms being used.

Overall, in this circumstance, it would be recommended to follow the local security policies, incident response plans and other internal policy, with the obvious first action being to disable the account used by the hacker and to block their access. The above mentioned legislation would still apply, and if there is a risk that personal information could be disclosed to a law enforcement agency, a formal request would need to be received.

References

Online Documents

Australasian Centre for Policing Research “The virtual horizon: Meeting the law enforcement challenges” 2000 URL: http://www.acpr.gov.au/pdf/ACPR134_1.pdf (26 May 2003)

Australian Communications Authority “Internet Service Providers and Law Enforcement and National Security” Nov 2000 URL: http://www.aca.gov.au/consumer_info/fact_sheets/industry_fact_sheets/fsi13.pdf (26 May 2003)

Australian Communications Authority “Telecommunications and Law Enforcement” July 1998 URL: http://www.aca.gov.au/aca_home/licensing/radcomm/about_radcomms_licensing/leac.pdf (26 May 2003)

Australian Communications Industry Forum, “Industry Code – Provision of Assistance to National Security, Enforcement and Government Agencies” 2002 URL: http://www.aca.gov.au/telcomm/industry_codes/codes/c537b.pdf (26 May 2003)

Commission on E-Business, IT and Telecoms “Policy Statement – Storage of traffic for law enforcement purposes” Nov 2002 URL: http://www.iccwbo.org/home/news_archives/2002/stories/traffic%20data.pdf (26 May 2003)

The Computer Technology Documentation Project “Windows NT Workstation Reference Version 0.6.0 1 Dec 2000 URL: <http://www.comptechdoc.org/os/windows/ntwsguide/ntwspmissions.html> (26 May 2003)

“GNU Binary Utilities” URL: http://www.gnu.org/manual/binutils-2.12/html_mono/binutils.html#SEC10 (26 May 2003)

“Forensics FAQ”
URL: <http://www.deaddrop.org/security/Presentations/2ndqtr/ForensicsFaq.html> (26 May 2003)

Fung, James “Dead linux machines do tell tales – GCFA Practical Assignment” URL: http://www.giac.org/practical/GCFA/James_Fung_GCFA.pdf (26 May 2003)

Internet Security Systems, X-Force Database, “loki (1452)” URL: http://www.iss.net/security_center/static/1452.php (26 May 2003)

“IT Laws in Australia” URL: <http://www-staff.it.uts.edu.au/~jim/cit2/site/law/LawAUMainFrame.htm> (26 May 2003)

KUBARK “Counter Intelligence Interrogation”, July 1963 URL: <http://www.hiddenmysteries.com/freebook/neuro/k1.html#I> (26 May 2003)

National Archives of Australia “Records in Evidence”, 1998 URL: http://www.naa.gov.au/recordkeeping/overview/evidence/records_in_evidence.htm (26 May 2003)

The National Centre for Forensic Science “Digital Evidence in the Courtroom: A guide for preparing digital evidence for courtroom presentation – Master Draft Document” March 2003 URL: http://www.ncfs.org/DE_courtroomdraft.pdf (26 May 2003)

Network Working Group RFC: 959, “File Transfer Protocol (FTP)”, Oct 1985 URL: <http://war.jgaa.com/ftp/rfc/rfc959.txt> (26 May 2003)

Phrack Magazine, Volume Seven, Issue Forty-Nine, “Project Loki” URL: <http://www.phrack.org/show.php?p=49&a=6> (26 May 2003)

Phrack Magazine, Volume Seven, Issue Fifty-One, “Loki2 - the implementation”, 01 Sept 1997 URL: <http://www.phrack.org/show.php?p=51&a=6> (26 May 2003)

Phrack Magazine, Volume Seven, Issue Fifty-One “Phrack Magazine Extraction Utility” Sept 1997 URL: <http://www.phrack.org/phrack/51/P51-17> (26 May 2003)

Sommer, Peter “Intrusion Detection as Evidence”, March 2000 URL: <http://www.bcs.org.uk/lac/ids.htm> (26 May 2003)

Strubinger, Ray “Exercises in the art and science of computer forensics - GCFA Practical Assignment” URL: http://www.giac.org/practical/GCFA/Ray_Strubinger_GCFA.pdf (26 May 2003)

Wood, Justice James “Expert Witnesses – The new era” June 2001 URL: http://www.agd.nsw.gov.au/sc%5Csc.nsf/pages/Wood_June2001 (26 May 2003)

Yam, Jason “Hacking and Cybercrimes” URL: <http://home.vicnet.net.au/~kengsn/Hacking.pdf> (26 May 2003)

Legislation

Cybercrime Act (Cwth) 2001 URL: <http://scaleplus.law.gov.au/html/pasteact/3/3486/pdf/161of2001.pdf> (26 May 2003)

Evidence Act (Cwth) 1995 URL:

<http://scaleplus.law.gov.au/html/pasteact/2/1182/top.htm> (26 May 2003)

Telecommunications Act (Cwth) 1997

Telecommunications (Interception) Act (Cwth) 1979

Other

- <http://www.info-zip.org> (26 May 2003)
- <http://biatchux.dmzs.com/?section=main> (26 May 2003)
- <http://www.vmware.com> (26 May 2003)
- <http://www.securityfocus.com/tools/215> (26 May 2003)
- <http://packetstormsecurity.nl/groups/ADM/indexsize.shtml> (26 May 2003)
- <http://www.ethereal.com> (26 May 2003)
- <http://www.tcpdump.org> (26 May 2003)
- <http://www.gtk.org/> (26 May 2003)
- <http://fire.dmzs.com> (26 May 2003)
- <http://www.knopper.net/knoppix/> (26 May 2003)
- <http://www.mandrakesecure.net/en/mnf.php> (26 May 2003)
- <http://www.aca.gov.au> (26 May 2003)