Fight crime.
Unravel incidents... one byte at a time.

WHEN IS A SERVICE NOT A SERVICE?

GCFA PRACTICAL ASSIGNMENT
V.1.3

Adrian Hammill

PART 1
Analysis of an Unknown Binary

Summary

Part 1 of the assignment is to provide a forensic analysis of a binary file to determine its use and capabilities. The file for the purposes of this assignment had been seized from a computer and requires analysis.

The machine on which I will be carrying out my investigation is not connected to a network and therefore any execution of the binary during my analysis is in a contained environment.

The file will be analysed on 1 standalone computer providing two forensic platforms.

1. RedHat Linux 8.0 running VMware v.4.0 (Machine name SCOOBY)
2. Windows 2000 Professional running on Vmware v.3.2.0 (Machine name MUTLEY)

Other virtual machines will be created as necessary.

The file binary_v1.3.zip has been downloaded from the GIAC download link.

For ease of reading where a command is used from a prompt (DOS or LINUX), or an application is used in conjunction with this investigation it will be shown in ***bold Italics***.

Binary Details

The binary_v1.3.zip file was unzipped on Scooby using "***unzip –X*** binary_v1.3.zip"

Filename

The filename of the extracted file from the zip archive is target2.exe; no other files were present in the zip archive.

File/MACTime

By using ***ls –i*** to provide a directory listing of target2.exe I was able to determine the inode for the file. This in turn will allow me to use ***debugfs*** to find out the MAC times.

The Modification Date was "Thu Feb 20 12:24:48 2003 "
The Accessed date was "Thu Feb 20 12:24:48 2003"
The Creation date was "Sun Jul 06 11:01:13 2003"

The Creation date is more recent than the modification date as this is the time that the file was created on my analysis system not the time that the file was actually created.

File owner

Depending on the method used to zip a file, using ***unzip -X*** to unzip it will often reveal the user and group ownership of the file within.
On this occasion however the ***- X*** parameter was not successful and so the user and group ownership were inherited from the local user.

File Size

Target2.exe is 26793 bytes in size.

MD5 hash

The MD5 hash value of the file is 848903a92843895f3ba7fb77f02f9bf1 and a screenshot is shown in Figure 1

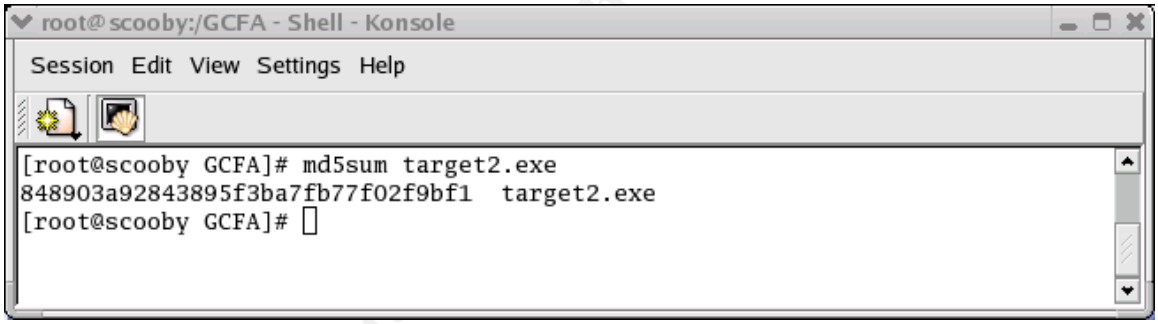


Figure 1

Keywords

The "***strings***" command was used to pull out of the binary any printable strings which existed in the file. ***Strings*** is a command which searches through any file for four or more printable characters.

Some of the interesting strings that were recovered are as follows:

MFC42.DLL
impossibile creare raw ICMP socket
RAW ICMP SendTo:
Icmp BackDoor V0.1
Code by Spoof. Enjoy Yourself!
Your PassWord:

loki
cmd.exe
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe

The references to dll's, services and cmd.exe in the output from ***strings*** above would indicate that the file is a MS Windows executable.

Having run strings I thought it worthwhile to run the file through ***BinText***. ***BinText*** is a windows tool that at its most basic is graphical version of strings. It does however have extended functionality that permits the user to define which characters make up part of the string. It also recognises null characters. The results with ***BinText*** varied slightly from those received from ***strings*** in that more literal character strings were visible. An example of this is below (the full output from ***BinText*** and ***strings*** are included in annex A)

!This program cannot be run in DOS mode.
\Winnt\system32\smsses.exe
\\199.107.97.191\C$
\winnt\system32
\Winnt\system32\reg.exe

None of the above were visible using ***strings*** but were with ***BinText.***
The output from both 'strings' and 'BinText' were interesting in that they both indicated that the binary was targeted at an MS Windows platform.

Program Description

Analysis Method

In an effort to find out what type of file has been distributed for analysis I carried out my investigation in the following way.
The file "binary_v1.3.zip" was downloaded from the GIAC download site in a zipped format. No MD5 hashes were provided to verify that the download file was the intended file and had not been corrupted.
The zipped file was unzipped on SCOOBY using ***unzip -X*** binary_v1.3.zip the –X argument was used as this restores any UID\GID information which may have been stored when the zip archive was created. However as described earlier the UID and GUID inherited by the file were that of the local user carrying out the investigation (me).
The extracted file was "target2.exe". Steps were then taken to establish the Modification Access and Creation (MAC) times of target2.exe. By using ***ls -i*** target2.exe I was able to identify the particular inodes where the file was extracted.

The inode information could then be fed in as an argument to ***debugfs*** which would in turn provide the MAC times of target2.exe. The draw back with analysing a file from a zipped archive is that invariably the creation time becomes the current time of unzipping.A screen shot of the debugfs output is shown in Figure 2

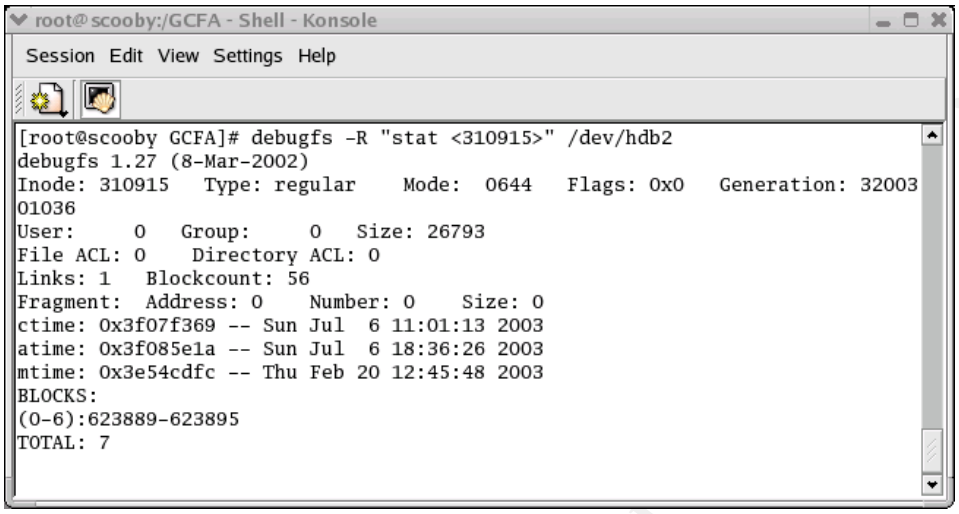


Figure 2

The next phase in analysing the binary was to run the command ***file*** target2.exe. ***file*** compares the signature of the file to a known source list and from this can determine the type of file we are dealing with.
The screen shot of this command being run is shown in Figure 3

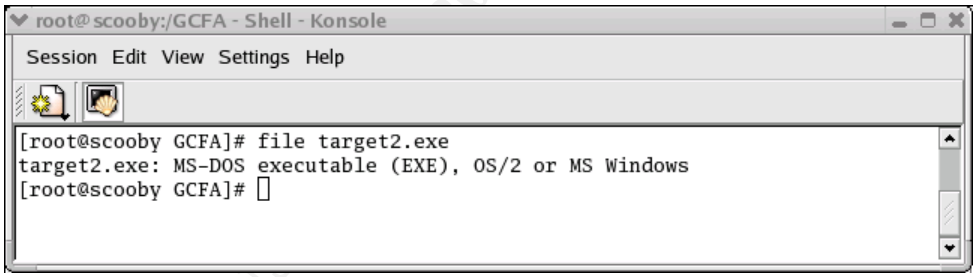


Figure 3

The file was identified as being an MS-DOS executable (EXE), OS/2 or MS Windows. As described previously I used ***strings*** to retrieve more information, which could be useful to my investigation.
I looked at the output from strings, which were provided in the keyword section more closely, and in particular :
"impossibile creare raw ICMP socket
RAW ICMP SendTo:
Icmp BackDoor V0.1
Code by Spoof. Enjoy Yourself!
Your PassWord:
loki

cmd.exe"
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe

I felt that some large hints as to what the program maybe intended for were visible, it would appear at face value to be an ICMP Backdoor helpfully shown with a version number v.01. It provides access to the command prompt on the compromised system. It was allegedly "coded by spoof "who cheerily tells us to "Enjoy Yourself!". The relevance of the word LOKI is that LOKI is an ICMP backdoor attack that has been around for some time now. LOKI was targeted at *NIX systems and therefore the implication is that spoof is acknowledging the creators of LOKI in their windows version of the LOKI attack tool.
The phrase "impossibile creare raw icmp socket" will also have value as I can search the internet for it and further occurrences of this spelling mistake (assuming of course that it is a spelling mistake and not just a foreign language!)
The strings referencing
"Open service failed!, Service %s Already exists, Local Printer Manager Service, smsses.exe" are also of great interest as these imply that the binary is trying to create and start a service, and has its own built in error code should the service already exist.

Having carried out as much analysis as I could on SCOOBY I transferred my analysis to MUTLEY. I again unzipped the binary_v.1.3.zip but this time used ***winzip*** to see if the MAC details were reported any differently. The created date was now being reported to be the same as the last modified date (20 Feb 2003 12:45:48) but the last accessed time was the time that the file was extracted***.***

I once again took an MD5 hash of target2.exe using a windows compatible md5sum calculator to ensure that I was still examining the same file. The MD5SUM is shown in Figure 4 for completeness.

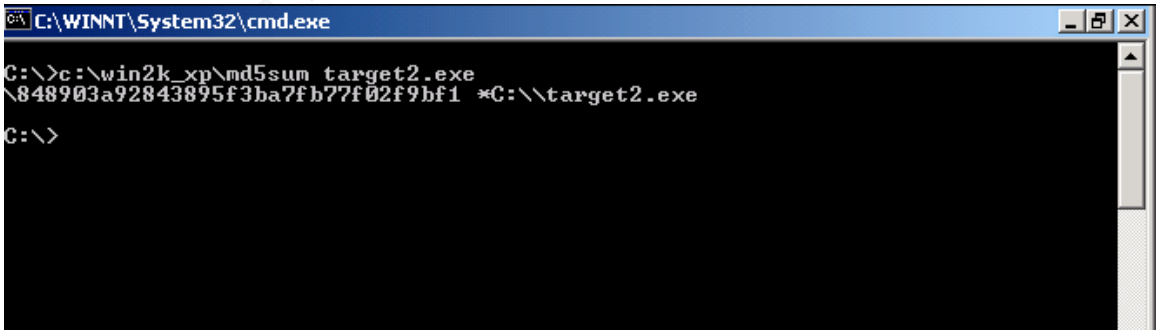


Figure 4

Having established I was analysing the same file I continued with my analysis. I used ***BinText*** to extract from the executable any strings of text that were available.

BinText has an advantage over strings in that it can be tailored to include or exclude particular punctuation and null characters. By using ***BinText*** I was able to extract more useful information.
The main interesting sections from ***BinText*** were as follows

This program cannot be run in DOS mode,
\Winnt\system32\smsses.exe,
\\199.107.97.191\C$,
Winnt\system32\reg.exe,
Local Partners Access

To determine what the program was to be used for I looked at the output from both ***strings*** and ***BinText*** . I believe that target2.exe creates an NT service namely " Local Printer Manager Service". The service is intended to provide an ICMP backdoor based on LOKI but built for a Microsoft Windows environment.

The BinText output showing \Winnt\system32 is relevant in with regard to the platform that the executable runs under. The default installation directory for Microsoft Windows NT Workstation 4 and Windows 2000 Professional is \WINNT. The default installation directory for Windows 98,ME and XP is \WINDOWS, thus I have made the assumption that the exploit was not being directed at these operating systems. To narrow down even further the operating system which the exploit is targeted against I created a virtual Windows NT 4 workstation machine in a Vmware session. I applied the latest service packs and then copied the target2.exe file onto the machine. I executed target2.exe by firstly running it from a command prompt and then secondly by double clicking on it. I received an error with regards to msvcrt.dll not having the *xp*locate functionality within it. By looking at which versions of msvcrt.dll would have the correct functionality it became apparent that none were available for Windows NT4. Having already discounted Windows 98,ME and XP on the location of the default installation directories I could now discount Windows NT4 on functionality. So a process of elimination now leaves me with the Windows 2000 family as being the target for this exploit.

From what I have determined the parts which were particularly relevant were targeted at creating a service " Local Printer Manager Service" to implement the ICMP backdoor. A registry entry is created for the service called " Local Partners Access ". Several references exist with regards to querying whether a service is running or not, stopping the service, making changes and then restarting the service. I once again created a virtual test environment this time with Windows 2000 professional. I ran the executable in the controlled environment and tried several parameters with it.
By a process of trial and error it became apparent that the parameters -i and -d followed by an IP address would install and delete respectively the "Local Printer Manager Service" on to the local machine.

However the service fails to start as it is dependent on smsses.exe being present in the c:\winnt\system32 directory.
No default installation of Windows 2000 has smsses.exe installed. I was unable to locate anywhere on the Internet a copy of smsses.exe or any reference to it. I can only assume that it is the second part of the exploit, which remains on the compromised system.
What my research on the Internet did find however was a Microsoft Security Bulletin MS00-053[1], which describes a Service Control Manager Named Pipe Impersonation Vulnerability. This described how it was possible for a "non-privileged user to elevate their existing security context to that of a service started by the Service Control Manager." In short "a malicious user could use a named pipe connection to instruct a windows 2000 based computer to start a predefined process that has security permissions higher than the permission that is assigned to the user." In the context of the binary target2.exe I believe that this is the vulnerability that is being exploited and that the service which is gaining higher security permissions is "Local Printer Manager Service" which is dependent on the executable smsses.exe. By renaming another executable to smsses.exe (In my test lab I picked tlntsvr.exe) I was able to get the telnet service running and open to attack. If the real smsses.exe is indeed LOKI then an ICMP backdoor will be permanently open.

Last Time of Use

In my test environment I kept a note of the MAC times and watched how they changed each time the binary was executed. The Created and Accessed times changed to the local time of the environment that I placed it into, and the accesses time changed whenever I ran the executable.
The modification time did not change throughout my analysis. With this in mind, taking another look at the original extraction MAC times show that the Modification and Accessed time were identical "Thu Feb 20 12:24:48 2003 ". I have therefore made the assumption that the file has not been executed since the file was last modified. Looking at the properties of the file through MS Windows Figure 5 we can see that the created date is "Thu Feb 20 12:24:48 2003 ". From this we can deduce that this version of the file was not executed on the compromised system and has not been run since it was compiled.

[1] http://www.microsoft.com/technet/security/bulletin/MS00-053.asp

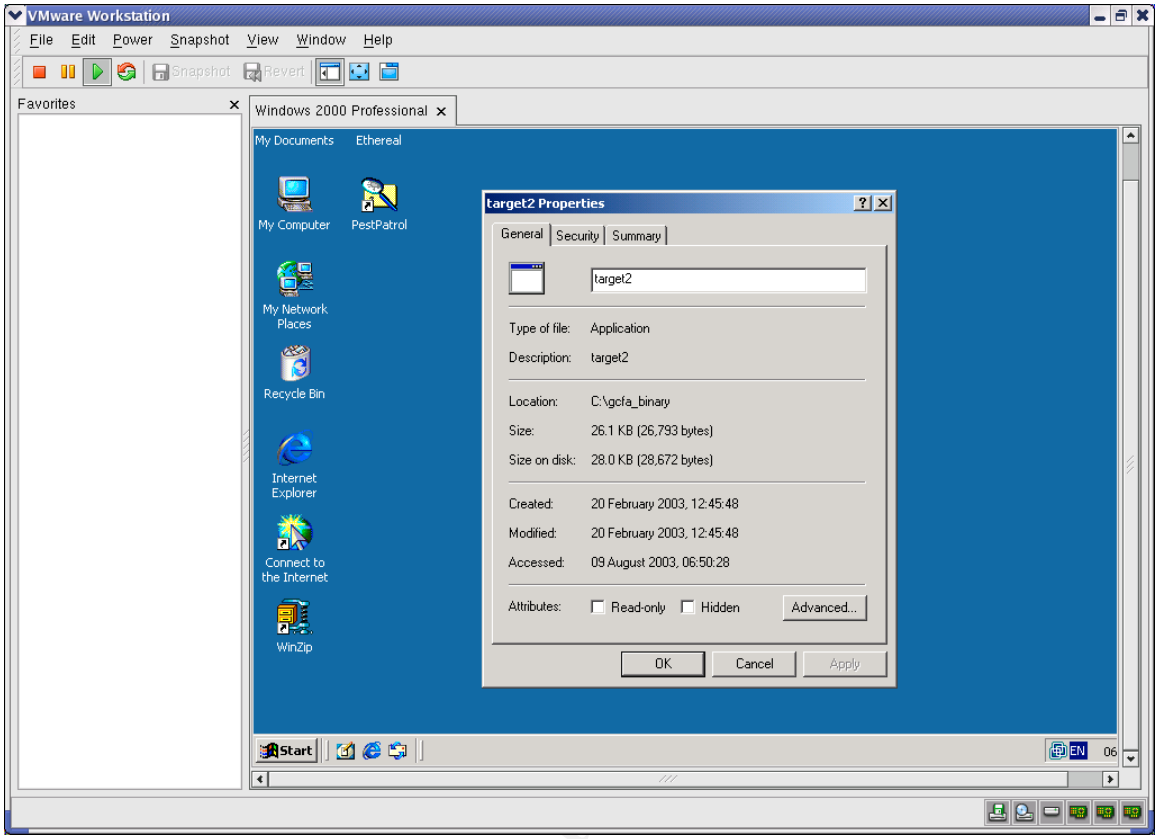


Figure 5

Forensic Detail

The trail left by the installation of target2.exe is as follows.

Registry Entry

By monitoring the registry while installing the service the following entries were added under Hkey_Local _Machine

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Local Partners Access]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):value
"DisplayName"="Local Printer Manager Service"
"ObjectName"="LocalSystem"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Local Partners Access\Security]
"Security"=hex:value
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Local Partners Access\Enum]
"0"="Root\\LEGACY_LOCAL_PARTNERS_ACCESS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

The following service is created

Local Printer Manager Service
It is set to run under the system account and to start automatically. The executable which it is dependent on is smsses.exe.
The Service fails to start and so an error message in the event log will be found which will show the service failed to start because the executable smsses.exe is not present

Files Used

The files that are used when the target2.exe is executed are as follows.

Kernel32.dll,
advapi32.dll,
gdi32.dll
mfc42.dll,
msvcp60.dll,
msvcrt.dll,
ntdll.dll,
rpcrt4.dll,
user32.dll,
ws2help.dll,
wsa_32.dll.
By watching the process running through ***procexp.exe*** I was able to identify the files which were used.
I also loaded the file into Dependency walker from the Windows 2000 resource kit and it was also able to validate the files which were required for the target to run. A screen shot of dependency walker is given in Figure 6

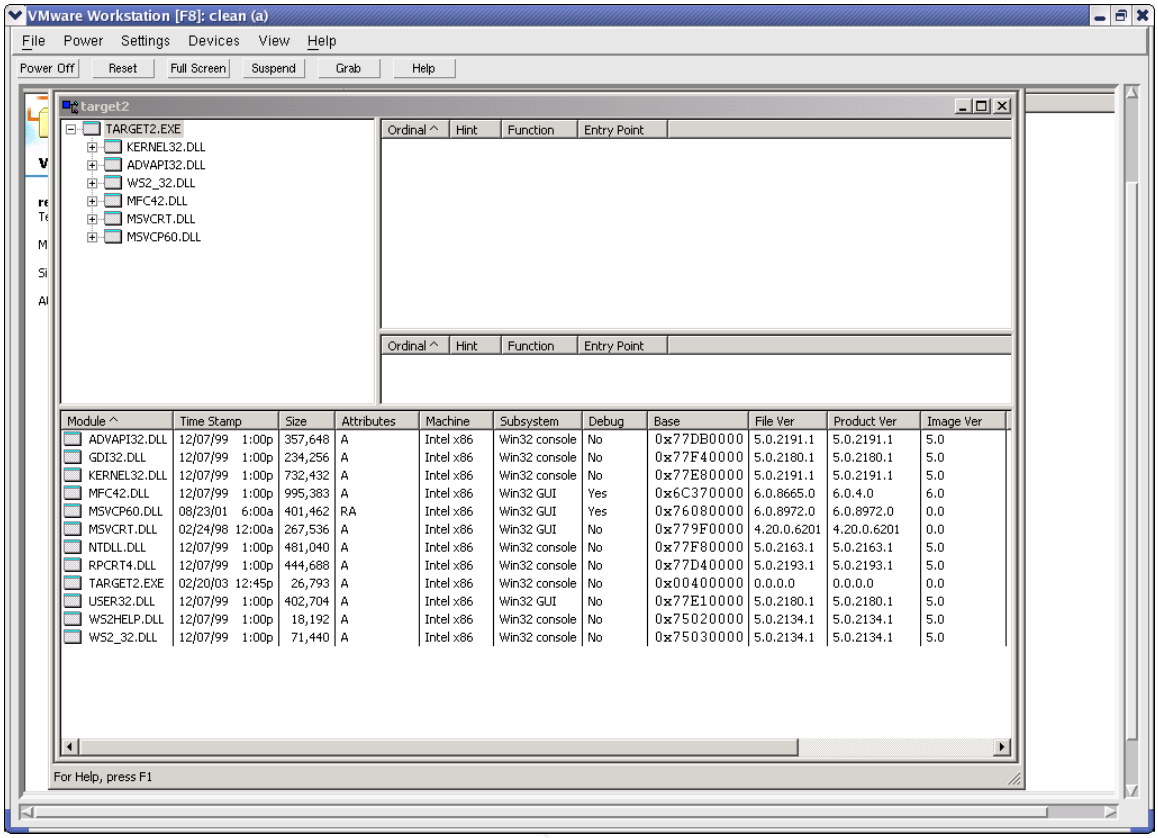


Figure 6

How is file system affected by installation of this program

From the tests I have conducted I can find no changes to the File System itself. Changes only occur to the services which are running (i.e. a new one is created)

Useful information

Several strings taken from BinText were identified as being interesting I shall explain their meaning here.

a) \winnt\system32\smsses.exe - The relevance of this line of text is with regard to the executable smsses.exe which is not found on any installation of Windows NT or 2000. This implies that the attack is made up of two executables.

b) \\199.107.97.191\c$ - This IP address was found and by using SAMSPADE I was able to identify the owner of the IP address. It belongs to the Azusa Pacific University in California that is participant in the Honeynet Project. Therefore I suspect one of two things that this exploit directs an attack at the Honeynet project or the source of this copy of the exploit was Azusa Pacific University.

Figure 7 shows the screen shot of SAMSPADE displaying the relevant information.

I considered contacting Azusa Pacific University with regards to one of their IP addresses being present in some malicious code but felt that was not within the spirit of the exercise. In a real world example it is an action I would have no worries about doing.

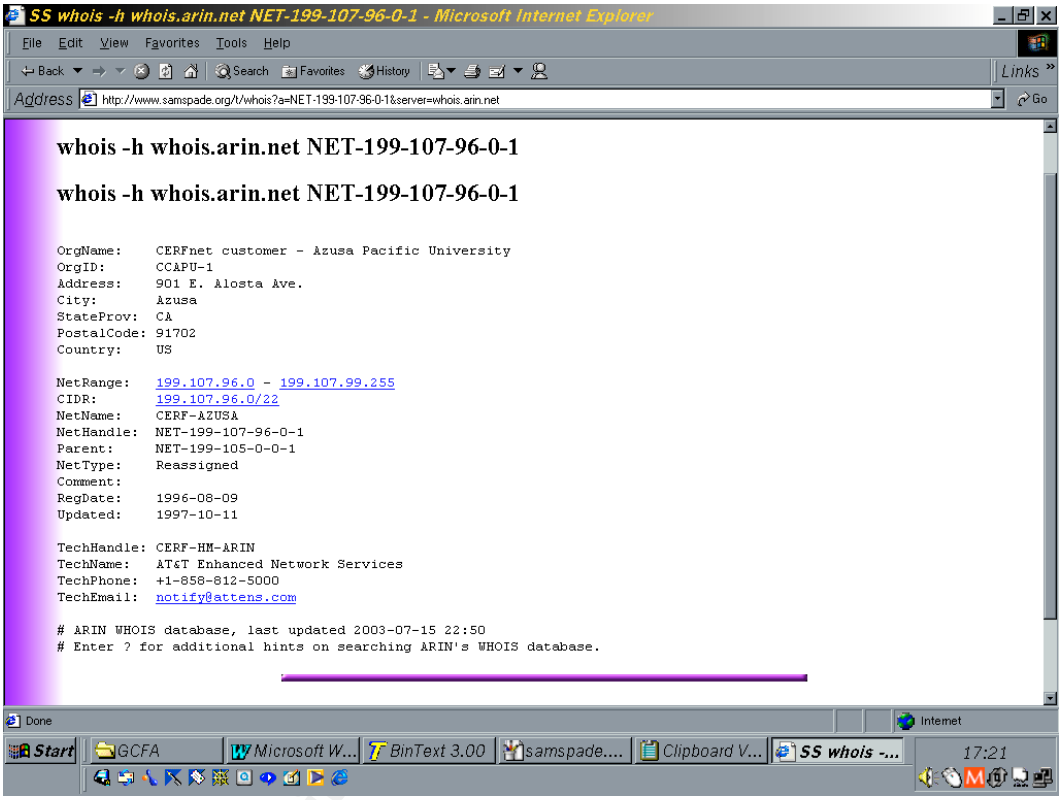


Figure 7

Program Identification

After extensive research of the Internet using all the usual culprits and more (www.google.com, www.dogpile.com, www.excite.com etc) looking for any "code by spoof, any windows ICMP backdoors v.0.1 and smsses.exe I had drawn a complete blank. I once again took a look at the output of BinText and identified " impossibile creare raw icmp socket" as another target to search for. I believed that the word creare and impossibile were mistakes in spelling and should they crop up anywhere else I may have found the code I was looking for (in fact they are italian spellings). The search results from Google, Dogpile and Excite all returned with one particular piece of code by Dark Schneider called icmp_tunnel.h which had brought up a match for the whole phrase " impossibile creare raw icmp socket".

A copy of this code was downloaded from www.s0ftpj.org[2]. I searched through the source code for other bits of information and found other remnants in there. The source code for ICMP_tunnel.h containing the relevant text is included in Annex B. Using Microsoft Visual C++ I compiled the code for ICMP_tunnel and carried out an MD5 hash on the resultant executable. The output from MD5SUM is shown in Figure 8 and can be seen to be different from that received for target2.exe.
I do not however believe that this is the code I was looking for so in the quest for finding it I failed.

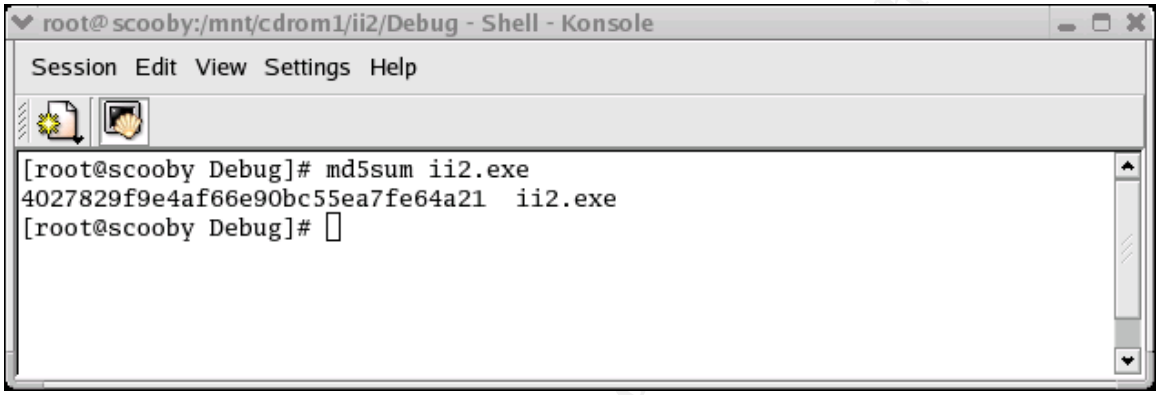


Figure 8

Legal Implications

By analysing the compromised system it would be possible to determine whether or not the program had been executed.
The symptoms of execution would show a new service had been created with the name Local Printer Manager Service, and by investigating the registry it would be possible to see the new registry key under services for Local Partner Access.

With these conditions in mind the laws for the UK which have been broken come under the section 1 and 3 of the Computer Misuse Act 1990.[3]

Section 1 states that -

> "a person is guilty of an offence under section 1-
> - if he causes a computer to perform any function with intent to secure access to any program or
> - data held in any computer, the access he intends to secure is unauthorised and

[2] www.s0ftpj.org/tools/icmplibv1.h

[3] www.hmso.gov.uk/bgi-bin2/hmso_hl?db=hmso-new&ste…/ukpga_19900018_en_2.ht

- he knows at the time that he causes the computer to perform the function that that is the case".

I believe that section 1 covers the implementation of the ICMP backdoor part of the attack.

Section 3 states that -

"a person is guilty of an offence

- if he does any act which causes an unauthorised modification of the contents of any computer and
- at the time he does the act he has the requisite intent and the requisite knowledge".

I believe that by installing a service and exploiting the named pipes impersonation vulnerability the person is knowingly guilty of unauthorised modifications to a computer.

Sentencing

If the person is found guilty under section 1 they are liable on conviction to a maximum prison sentence of 6 months or a maximum fine of £2000 or both.
If the person is found guilty under section 3 they are liable on conviction to a maximum prison sentence of 5 years or an unlimited fine or both.
A recent example of an offender being successfully prosecuted on three counts under Section 3 of the Computer Misuse Act 1990 is that of Simon Vallor. [4]A web designer from North Wales, Vallor was responsible for the creation and distribution of three destructive viruses (Gokar, Admirer and Redesi) which spread to 42 countries. Vallor was jailed in January 2003 for two years.
Vallor appealed[5] in July 2003 claiming not to realise the extent of the damage his viruses would cause, however the presiding judge dismissed the appeal saying that Vallor's crimes were calculated and disruptive.

[4] www.legalday.co.uk/lexnex/evershed03/e80220103a.htm

[5] www.sophos.com/virusinfo/articles/vallorappeal.html

Interview Questions

I would begin the interview by explaining that our organisation is currently going through a process for ISO 17799 accreditation. As such we are required to interview members of staff about security awareness.

The line of questioning I would follow when interviewing the person suspected of running the exploit is as follows:

- Under the current system operating procedures which it was necessary for you to sign to get an account on the network it is disciplinary offence to share your password, have you ever shared your password with someone?

- Do you think that it is possible that someone has discovered your password and used your account?
- How secure do you think our systems are, do you feel that it is within your capabilities to break into them?

- Do you feel that we should be more vigilant in the ways that we patch our systems?

- Have you any interest in Norse gods, for instance do you know who the Norse God of Mischief is?

- If I was to tell you his name was LOKI would that ring any bells?

- Do you know another use of the word LOKI.

- Have you ever tried running unauthorised code on any of our systems?

- On such a day your account was seen to run some unauthorised code in the light of your previous answers can you explain how this is.

Hopefully the questions will draw out the fact that they have either

- Shared their account with another member of staff (a disciplinary offence)

- Left a machine logged on but unattended (a minor offence)

- Compromised their password (a minor offence but if they have and realised it and done nothing about it more serious) or

- Own up to the whole thing. The jump to questioning about Norse Gods is just to distract them momentarily.

Part 2
Option 1: Perform Forensic Analysis on a system

Summary

Web sweeper logs had alerted a system administrator monitoring the network under his control that a user account was attempting to download unauthorised code. It was brought to the attention of the Security team who believed the user in question had no valid reason for attempting such a download.
The Hard Drive for the machine was removed and brought to me for examination. Our main concern is that the user has been able to circumvent the Firewalls and Web Sweepers. The user in question had local administration access on to the workstation and had they been successful in their downloads they would have had the ability to install the code.
A forensic examination of this hard drive will establish whether or not any unauthorised code has been successfully installed on the machine and executed. The Machine was confiscated within two hours of the alert but had already been powered down, therefore it was not possible to capture a live machine or the contents of its memory.

System Description

The system is a Windows NT4 workstation working on an NT domain. The network is Ethernet based with all clients having a 10/100mb network card.
It is a tightly configured machine in an attempt to prevent possible infection by malicious code. Users are required to sign an acceptable use policy before using the network, which states within it that the installation of any unauthorised software is prohibited. It is connected to the Internet through a Proxy server and firewall which. The Internet network is not part of the corporate LAN and so has no access to production data. The network is provided purely for research purposes for development and engineering staff.
The workstation itself is built on Windows NT4 service pack 6a, with Office 2000 service pack1 as the Office Automation suite. A PDF reader is also provided. Outlook Express is the Mail client.

Hardware

TAG	DESCRIPTION
GCFA290703_01	Dell Optiplex GX110 Tag Pxxxx Ser No 123456789
GCFA290703_02	Maxtor Hard Drive 15gb Model 91531u9 Cylinders 16383 Heads 16 Sectors 63 S/n g31xxxxx

The computer system consisted of a Pentium III 1 Ghz processor, 128MB SDRam, a single removable Maxtor 15GB IDE hard drive, an Internal Zip 250 ide drive, an internal 3.5" floppy drive and a 10/100MB Allied Telesyn ethernet card. The motherboard had a built in sound card with no speakers attached and no sound drivers loaded.
Items GCFA290703_01 and GCFA290703_02 were the only items that were seized from the Engineering Department. The equipment had been linked up to a standard issue keyboard and mouse (PS2 connection) and an LCD Monitor however I did not feel it was necessary to remove these items as it would add no extra value to the investigation.

Image Media

I took item GCFA290703_02 (removable hard disk) from the suspect machine and placed it into a disk imaging machine to image.
The disk imaging machine is a tower PC with 3 removable hard disk bays(2 IDE 1 SCSI), a CD Writer (SCSI), DVD(IDE), Internal ZIP 250(IDE), 1.4MB Internal Floppy Disk Drive, 1.4ghz processor and 1GB RAM. The operating system on this machine is RedHat 8.0.
The imaging environment is booted from a floppy boot disk and is on /dev/hdb (not an ideal situation but one that will be rectified when time permits) and as such the suspect drive will be seen as ***/dev/hda***.

I placed GCFA290703_02 into the spare IDE bay and a blank SCSI disk into the SCSI bay. I started up the imaging machine and once I had logged on ran ***fdisk -l /dev/hda*** to list the disk device and its partitions.GCFA290703_02 had a single NTFS partition.
I then ran ***MD5SUM /dev/hda*** to establish the hash algorithm of the hard disk so that I can prove once I have imaged the disk that I am working with an identical copy of the suspect drive. Screen shot Figure 9 shows the output from that command.

Figure 9

My next step was to ensure that the device I was imaging to was completely blank. To do this is I firstly check that there is nothing on the disk that I required.

I then wrote /dev/zero to the device using ***dd if=/dev/zero of=/dev/sda*** this overwrote each bit with a zero character to ensure that there was no remnant data on the device.I then recreated the partition using ***fdisk*** and remade a file system using ***mkfs***.I mounted the newly cleaned device at /mnt/firewire and then proceeded to make a dd image of GCFA290703_02 using. Figure 10 below shows the result of an MD5SUM check on the imaged media and by comparing it with the number displayed in Figure 9 it can clearly be seen that they are the same.

Figure 10

I then transferred the newly created image to my forensics machine Scooby.

Media Analysis of System

System Description

As described in Part 1 of this assignment my Forensic machine consists of a RedHat 8.0 Linux machine called Scooby in which resides a VMware version 4.0 Windows 2000 virtual environment called Mutley.
The Workstation itself is a 1.4ghz Athlon Processor, 500MB RAM, CD Writer, DVD Player, 1.4 MB floppy disk. Two removable Hard disk drive bays.
I have Autopsy 1.7 pre-configured on the Scooby so that a symbolic link is all that is required to attach to the data. This way there is less movement of the evidence and no chance of corruption by inheriting slack space from another medium.
On Mutley I have Encase, Ilook Investigator and Netanalysis along with all the other tools supplied during the course.
The reason I use various tools is that I have not found one tool that does all of the jobs well, they all have their strengths and weaknesses and essentially compliment each other. It is also an ideal way to verify ones evidence has not changed and that the results found are accurate and consistent.

Precautions Taken

I took the disk and mounted it into my environment using the ***mount -o ro*** so that the disk was mounted read only.
I have it configured so that the virtual machine can see the evidence disk in read only mode through being mounted as a secondary disk in the virtual session.

Modification of Operating System

To check whether or not the system operating system had been modified I loaded the image hd_dell.dd into ***iLook investigator***[6]. ***iLook investigator*** is a widely known and well respected Forensic Analysts tool. With this tool I loaded the image of the hard drive and mapped its file system. When an image is mounted with iLook it automatically creates categories of found items, the investigator can create more for himself but the default ones are a great starting point. ILook's automatic categorisation has a section for undeleted files.
The next step I took was to run an MD5 hash analysis on each individual file. By then using a process called negative hash analysis I can reduce the number of files to be investigated from the image by comparing the hash values with the HashKeeper database.
By comparing the contents of GCFA290703_02 with a known good version of Windows NT4 sp6 I was able to determine that the operating system had not been modified.

Backdoor and Sniffer checks

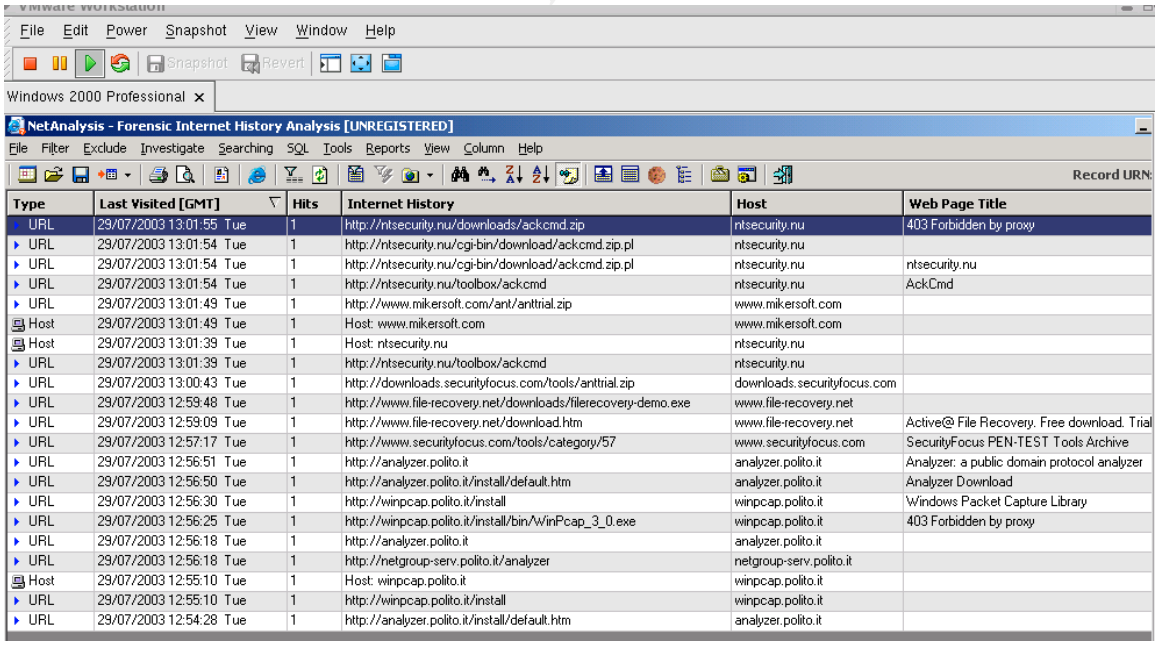
I then went on to look for backdoors which may have been installed on the system for this I used a product called Pest Patrol, which can identify and eliminate up to 11,000 known backdoors, Trojans and sniffers.
To enable me to do this I mounted the image hd_dell.dd as a read only loop device on Scooby. This then enabled me to look at the file system in detail.
It was possible using functionality in VMware 4 to have this file system mounted as a read only shared folder to my MS Windows 2000 forensic virtual machine Mutley.
I then started ***Pest Patrol*** and scanned the shared folder for any Trojans, Backdoors and Sniffers. After scanning 10,500 files Pest Patrol reported that no Trojans, Sniffers or Backdoors were present on the disk.

[6] www.ilook-forensics.org - For more information on iLook

IE history

Initial searches were fruitless when looking for Internet History files, this is due to the machine being configured not to cache profiles locally. I returned to ***iLook*** and the undeleted files which it had found. Under the profile of the user I was investigating it was possible to recover his Temporary Internet folder. ***iLook*** allows you to copy the contents of the folder into your local filesystem. So it was with this utility that I extracted the user's internet history folder onto my forensic machine into an area that had been prepared so as not to contaminate the retrieved data.
These internet history files were then opened using ***NetAnalysis*** [7]– A Forensic Internet History Analysis tool.
NetAnalysis is a powerful tool which allowed me to open all of the user's internet history files into one window. I was then able to sort by date and time and effectively relive the users internet past from this viewer.
NetAnalysis has some great functionality which allows you the investigator to filter events, to group by websites, retrieve searches, and produce reports on how often a user has visited particular sites. Identify which sites are of possible pornographic nature and which sites where visited because of a redirect.

When I first loaded the Internet History folder into NetAnalysis the records were sorted into date order with the most recent events at the top. I was pleasantly surprised to see the following in Figure 11 as it vindicated the interest that had been taken in the user.



Type	Last Visited [GMT]	Hits	Internet History	Host	Web Page Title
URL	29/07/2003 13:01:55 Tue	1	http://ntsecurity.nu/downloads/ackcmd.zip	ntsecurity.nu	403 Forbidden by proxy
URL	29/07/2003 13:01:54 Tue	1	http://ntsecurity.nu/cgi-bin/download/ackcmd.zip.pl	ntsecurity.nu	
URL	29/07/2003 13:01:54 Tue	1	http://ntsecurity.nu/cgi-bin/download/ackcmd.zip.pl	ntsecurity.nu	ntsecurity.nu
URL	29/07/2003 13:01:54 Tue	1	http://ntsecurity.nu/toolbox/ackcmd	ntsecurity.nu	AckCmd
URL	29/07/2003 13:01:49 Tue	1	http://www.mikersoft.com/ant/anttrial.zip	www.mikersoft.com	
Host	29/07/2003 13:01:49 Tue	1	Host: www.mikersoft.com	www.mikersoft.com	
Host	29/07/2003 13:01:39 Tue	1	Host: ntsecurity.nu	ntsecurity.nu	
URL	29/07/2003 13:01:39 Tue	1	http://ntsecurity.nu/toolbox/ackcmd	ntsecurity.nu	
URL	29/07/2003 13:00:43 Tue	1	http://downloads.securityfocus.com/tools/anttrial.zip	downloads.securityfocus.com	
URL	29/07/2003 12:59:48 Tue	1	http://www.file-recovery.net/downloads/filerecovery-demo.exe	www.file-recovery.net	
URL	29/07/2003 12:59:09 Tue	1	http://www.file-recovery.net/download.htm	www.file-recovery.net	Active@ File Recovery. Free download. Trial
URL	29/07/2003 12:57:17 Tue	1	http://www.securityfocus.com/tools/category/57	www.securityfocus.com	SecurityFocus PEN-TEST Tools Archive
URL	29/07/2003 12:56:51 Tue	1	http://analyzer.polito.it	analyzer.polito.it	Analyzer: a public domain protocol analyzer
URL	29/07/2003 12:56:50 Tue	1	http://analyzer.polito.it/install/default.htm	analyzer.polito.it	Analyzer Download
URL	29/07/2003 12:56:30 Tue	1	http://winpcap.polito.it/install	winpcap.polito.it	Windows Packet Capture Library
URL	29/07/2003 12:56:25 Tue	1	http://winpcap.polito.it/install/bin/WinPcap_3_0.exe	winpcap.polito.it	403 Forbidden by proxy
URL	29/07/2003 12:56:18 Tue	1	http://analyzer.polito.it	analyzer.polito.it	
URL	29/07/2003 12:56:18 Tue	1	http://netgroup-serv.polito.it/analyzer	netgroup-serv.polito.it	
Host	29/07/2003 12:55:10 Tue	1	Host: winpcap.polito.it	winpcap.polito.it	
URL	29/07/2003 12:55:10 Tue	1	http://winpcap.polito.it/install	winpcap.polito.it	
URL	29/07/2003 12:54:28 Tue	1	http://analyzer.polito.it/install/default.htm	analyzer.polito.it	

Figure 11

[7] A demo version of NetAnalysis was downloaded from www.digitaldetective.co.uk

The internet history files showed general normal internet usage until the 29th July when a concerted effort was made to download ackcmd.exe (a Trojan), WinPcap (a packet sniffer), anttrial (Advanced Net tools scanning the network for network shares) and a file recovery demonstrator. All of these downloads were blocked at the Firewall.

Timeline Analysis

To create a timeline analysis of the disk I used Encase, this provides the output into a text format which was easily imported into Excel for sorting. The complete timeline (all 42,000 lines of it are added as an extra file attachment to the practical submission)

Operating System Installation Date.

By looking at the Application and System event logs on the evidence I was able to determine that the first installation date for the operating system was November 1st 2000. Looking at the timeline for this period I was able to observe the following in Figure 12, this shows that MAC times for the records in the Master File Table (MFT) confirm the installation date to be November 1st 2000. The MFT is created when an NTFS volume is first formatted. The timeline Indicates that the installation was carried out on a piecemeal basis, with service packs, Office Automation suite and device drivers added over the first month.

Wed Nov 01 2000 09:52:31	0	mac	-/-r-xr-xr-x	$Volume
	8192	mac	-/-r-xr-xr-x	$Boot
	3751176	mac	-/-r-xr-xr-x	$Bitmap
	0	mac	-/-r-xr-xr-x	$BadClus
	36000	mac	-/-r-xr-xr-x	$AttrDef
	0	mac	-/-r-xr-xr-x	$Quota
	24798883	84 m	ac -/-r-xr-xr	/$BadClus:$Bad
	15260672	mac	-/-r-xr-xr-x	$MFT
	4194304	mac	-/-r-xr-xr-x	$LogFile
	131072	mac	-/-r-xr-xr-x	$UpCase
	4096	mac	-/-r-xr-xr-x	$MFTMirr

Figure 12

Figure 13 shows the NT service pack being installed on the 30th November illustrating how this machine was not built in one day.

Thu Nov 30 2000 11:01:45	6656	.a.	-/-rwxrwxrwx	/WINNT/system32/spmsg.dll
Thu Nov 30 2000 11:01:54	56	m.c	d/dr-xr-xr-x	/WINNT/$NtServicePackUninstall$
Thu Nov 30 2000 11:01:55	429840	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/autochk.exe
	136976	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/acledit.dll
	40208	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/smss.exe
	1301200	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/win32k.sys (deleted realloc)

	1301200	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/win32k.sys
	363792	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/ntdll.dll
	247056	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/advapi32.dll
	21264	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/arp.exe
Thu Nov 30 2000 11:01:56	449296	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/autoconv.exe
	65808	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/cacls.exe
	22800	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/atsvc.exe
	28432	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/bootok.exe
	34064	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/basesrv.dll
	20752	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/bootvrfy.exe
Thu Nov 30 2000 11:01:57	53008	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/compact.exe
	29456	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/csrsrv.dll
	28432	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/ddhelp.exe
	139024	.a.	-/-rwxrwxrwx	/WINNT/$NtServicePackUninstall$/ddraw.dll

Figure 13

Figure 14 below shows a printer driver being installed on Friday 16th May 2003.

Fri May 16 2003 11:13:32	144	m.c	d/drwxrwxrwx	/Program Files/Hewlett-Packard/LaserJet All-in-one/Uninstall
	272	m.c	d/drwxrwxrwx	/Program Files/Hewlett-Packard
Fri May 16 2003 11:13:38	470	ma.	-/-rwxrwxrwx	/Program Files/Hewlett-Packard/LaserJet All-in-one/Uninstall/2200/ComMsg.huf
Fri May 16 2003 11:13:39	166	ma.	-/-rwxrwxrwx	/Program Files/Hewlett-Packard/LaserJet All-in-one/Uninstall/2200/Testpage.huf
Fri May 16 2003 11:13:40	144	m.c	d/drwxrwxrwx	/Program Files/Hewlett-Packard/LaserJet All-in-one/help
	259	ma.	-/-rwxrwxrwx	/Program Files/Hewlett-Packard/LaserJet All-in-one/Uninstall/2200/Help.huf
	333	ma.	-/-rwxrwxrwx	/Program Files/Hewlett-Packard/LaserJet All-in-one/Uninstall/2200/Config_Silent.huf

Figure 14

And finally Figure 15 shows the dates an Anti-Virus product was installed on the workstation.

Thu May 22 2003 10:32:31	774217	m..	-/-rwxrwxrwx	/Program Files/Common Files/Symantec Shared/VirusDefs/20030723.016/NAVEX15.VXD
	774217	m..	-/-rwxrwxrwx	4 /Program Files/Commo

	804736 m..	-/-rwxrwxrwx	n Files/Symant ec Shared/Virus Defs/200307 16.005/NAVE X15.VXD 4 /Program Files/Commo n Files/Symant ec Shared/Virus Defs/200307 16.005/NAVE X15.EXP
	804736 m..	-/-rwxrwxrwx	/Program Files/Commo n Files/Symant ec Shared/Virus Defs/200307 23.016/NAVE X15.EXP
Thu May 22 2003 10:32:33	12568 ma.	-/-rwxrwxrwx	/PkgClnup.lo g

Figure 15

Recover Deleted files

As described in the Internet History section above the users profiles are not cached locally, they are effectively deleted. So to enable me to recover the Temporary Internet files I used ***iLook***. On initial loading and mapping of the filesystem iLook recovers any files it believes to be undeleted (i.e. have not been deleted securely\properly). It then makes them available for extraction and examination.
I verified that the extracted file was identical to the one inside the image by comparing MD5 hashes.

The files had been deleted at user logoff at 14:05 29/07/03 and it is the index.dat files within which are most pertinent as these contain the Internet History and temporary Internet files.

String Search

I conducted a string search on the media using the words ***ackcmd, winpcap, packet.sys, anttrial, cheops and Lophtcrack***. I chose these phrases because they are indicative of things the user was trying to download and the interests which they now seem to have. ***Packet.sys*** is the driver which is installed by ***WinPcap***, ***cheops*** is a network mapping tool and I would expect anyone who is trying to use tools such as ***anttrial*** to enumerate network shares to also be using ***cheops*** to discover the extents of the network. Lophtcrack is on my search list as

it was one of the first tools I came across in my first excursion into Network Security and an old favourite of many.

The string search recovered only the temporary internet files which I had already found. With one exception ***Lophtcrack***, the search results found an entry for this in the pagefile.sys. Investigating further using Encase I was able to extract a zip file which contained the executable for Lophtcrack. Searching through the timelines and file listings proved unsuccessful and it would appear that Lophtcrack has not been installed on this machine.

Conclusions

Based on the analysis I have carried out I believe that the use of the machine is generally for research purposes and for fault resolution. The machine can be seen to be used by two people when looking through the timeline, but for the purposes of this investigation I was only interested in John Doe. A large percentage of the Internet traffic was with regards to service pack downloads, network card information and Microsoft Knowledge base lookups. The Internet connection was also used for personal lookups with several instances of browsing Estate Agent websites, and computer games websites and Football all of which cause me no concern and are not against the acceptable use policy which we have in place. We can gather from his internet searches that the user is considering a move (the locations searched for are within their current neighborhood) and that they are an avid games player who resorts to cheating when the game defeats him (several visits to games crack sites gives this away). However what is concerning is the attempts on one particular day to download a Packet Sniffer, Trojan and Advanced Net Tools this is against the acceptable use policy and the culprit will be invited for an interview with a view to establishing the facts. From looking at the hard disk in great detail it can be seen that although attempts were made to download prohibited material the user concerned was unsuccessful. With the exception of Lophtcrack which has widened the spread of the search. All networks which this person has had access to will need to be checked and the person interviewed. I believe this could become a long drawn out investigation, outside the remit of this assignment.

Part 3
Legal Issues of Incident Handling

Brief summary of the scenario

I am a system administrator working for an Internet Service Provider (ISP).
I take a call from a Law Enforcement Officer who informs me that a Government computer was hacked into from an account on my system.
The law enforcement officer has been able to prove to me over the phone that he is who he says he is.
The law enforcement officer requests that I verify the activity by reviewing my logs and to determine whether or not the attack was initiated from my system or from another provider.
My review shows that only a valid user account logged in via dial-up during the time of suspicious activity.

The answers provided in this section are based on United Kingdom (UK) law.
I have made the assumption that for the purposes of this assignment that Law Enforcement equates to the Police Force, it does not include Custom and Excise or any Intelligence Agency as in my opinion they would not be partaking in an investigation of this nature.
I have also made the assumption that the ISP is UK based.

A. Any details, which are held on a computer with reference to an account or to a user, are protected under the Data Protection Act 1998. This Act provides for the protection of information held about an individual by an organisation. The responsibilities of an organisation not to divulge that information to another or to publish it without consent. Thus in the first instance I can not legally provide any information about the account which they are enquiring about to a Law Enforcement Officer over the telephone. I can however confirm whether or not any activity took place at the time the suspicious activity took place.
B. The Law Enforcement Officer can present me with a certificate under section 29 of the Data Protection Act requesting that I maintain and provide access to the information he needs to continue with his investigation. Section 29 provides an exemption when the information is being used for "the prevention or detection of crime, and the apprehension or prosecution of offenders".The Law Enforcement Officer would be able to apply for a certificate under Section 29, this would require him going to a senior officer (Inspector) within his Constabulary who would provide a certificate through a single point of contact with the ISP. At all times the officer must be able to provide a clear audit trail of his actions.

C. If the Law Enforcement Officer serves a Data Protection Form onto me requesting the Logs from the time of the event and any account details about the account involved including ownership, I can still refuse to hand over any information. How wise this would be I am not sure but the Data Protection Act would allow me to say no. The Law Enforcement Officer would then have to

present a solid case to a circuit court judge to obtain a special production order under schedule 1 of the Police and Criminal Evidence Act 1984 (PACE). Schedule 1 paragraph 2 of PACE sets out the conditions that must be fulfilled in order for a circuit judge to provide a special production order. The relevant parts of this paragraph are:

"2.

A) (iii) that the material is likely to be of substantial value(whether by itself or together with other material) to the investigation in connection with which the application is made.

(iv) that the material is likely to be relevant evidence.

B) other methods of obtaining the material -

(i) have been tried without success."

Paragraph 5 of schedule 1 goes on to dictate that

"where the material consists of information contained in a computer.

(b) ……..give a constable access to the material in a form which it is visible and legible."

If I were served with one of these I can once again refuse to provide the information but in doing so would be in contempt of court and would be in deep trouble.

D. The investigative activity that I can perform is based around checking system functionality and integrity, verifying that the system is in good working order. There is a provision under the Data Protection Act, which requires me to verify user details. With this in mind I would be able to check the validity of the user account which is linked to the attack. However I have no specific investigative remit.

E. If it became apparent that a hacker had hacked my system to create an account, which was then used to hack a government system, I would feed the information into the Incident Response Team. After the initial contact from the Law Enforcement Officer the Incident Response co-ordinator would have been contacted to make them aware of a potential compromise. The IR team would have already been carrying out work on the periphery and would now take a stronger stance in maintaining the integrity of our system. Images of the affected systems should be taken to be used for evidence should the case ever come to court. All vulnerabilities, systems and procedures, which have been exploited, will have to be tightened down to make the system more secure. All passwords should be changed and the validity of all accounts checked. The main aim now is to secure the system by following basic incident handling procedures. The compromised systems will need to be investigated. A criminal offence has taken place and it is our responsibility to ensure that Law Enforcement has as much evidence as they require in their pursuit of the offender.

We should at minimum be following the Incident Handling guidelines to ensure that our system can be restored to a production environment without

infection or outstanding vulnerabilities. We should also be ensuring that none of our other systems have been compromised and are suitably patched.

Preperation
Identification
Containment
Eradication
Recovery
Lessons learned

Are the words that should be at the foremost of our mind

REFERENCES

Books and Papers

Anti Hacker Toolkit - Jones, Schema and Johnson
Hacking Exposed 3rd Edition - McClure, Scambray & Kurtz
Reverse Engineering Hostile Code - Joe Stewart
The Police and Criminal Evidence Act 1984 3rd edition Michael Zander
Microsoft Windows NT Resource Kit - Microsoft Press

Websites

http://www.vogon-computer-evidence.com/forensic_services-03.htm
www.homeoffice.gov.uk
www.securityfocus.com/infocus/1637
http://www.microsoft.com/technet/security/bulletin/MS00-053.asp
http://www.legislation.hmso.gov.uk/acts/acts1998/19980029.htm
http://www.legislation.hmso.gov.uk/acts/acts1990/Ukpga_19900018_en_1.htm
www.google.com
www.dogpile.com
www.excite.com
www.packetstormsecurity.nl
www.samspade.org
http://home.apu.edu/~bmccarty/honeynet.html
http://www.honeynet.org/alliance/index.html
www.digitaldetective.co.uk
http://www.totse.com/en/zines/crh/crh002.html
www.security-corporation.com
www.guardent.com
www.s0ftpj.org/tools/icmplibv1.h
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0737
www.legalday.co.uk/lexnex/evershed03/e80220103a.htm
www.sophos.com/virusinfo/articles/vallorappeal.html

Other resources

K McNulty Home Office
I J Sansome Hampshire Police

Annex A - Output from BinText

File pos	Mem pos	ID	Text
0000004D	0040004D	0	!This program cannot be run in DOS mode.
000001D0	004001D0	0	.text
000001F8	004001F8	0	.rdata
0000021F	0040021F	0	@.data
00000248	00400248	0	.rsrc
000011D0	004011D0	0	D$,QPR
000011FC	004011FC	0	D$ j'P
0000121E	0040121E	0	T$,j'RP
000012FE	004012FE	0	T$,VRS
00001327	00401327	0	D$ j'P
00001349	00401349	0	T$,j'RP
00001408	00401408	0	L$ j'Q
0000142B	0040142B	0	D$,j'PQ
00001540	00401540	0	D$0QPR
0000156E	0040156E	0	D$$j'P
00001590	00401590	0	T$0j'RP
00001678	00401678	0	T$0URV
000016A1	004016A1	0	D$$j'P
000016C3	004016C3	0	T$0j'RP
00001803	00401803	0	D$ j'PQ
000019AF	004019AF	0	T$$QRj
000019CE	004019CE	0	D$$PW
00001BD6	00401BD6	0	h0A@
00001CEA	00401CEA	0	SPhxD@
00001D10	00401D10	0	SQhpD@
00001D65	00401D65	0	D$@SPS
00001E16	00401E16	0	T$|RP
00001E77	00401E77	0	USSSP3
00001F25	00401F25	0	D$(PQ
00002050	00402050	0	x!xu\
00002056	00402056	0	x"iuV
0000205C	0040205C	0	x#tuP
0000207A	0040207A	0	IQh@A@
00002270	00402270	0	t1h@D@
000022B4	004022B4	0	Ht Ht
0000243E	0040243E	0	Ph<B@
00002460	00402460	0	T$(QR
0000249D	0040249D	0	L$0PQ
00002528	00402528	0	Ph0C@
000032EA	004032EA	0	Sleep
000032F2	004032F2	0	HeapAlloc
000032FE	004032FE	0	GetProcessHeap
00003310	00403310	0	TerminateProcess
00003324	00403324	0	ReadFile
00003330	00403330	0	PeekNamedPipe
00003340	00403340	0	CloseHandle
0000334E	0040334E	0	CreateProcessA
00003360	00403360	0	CreatePipe
0000336E	0040336E	0	WriteFile
0000337A	0040337A	0	GetLastError

```
0000338A   0040338A    0   LocalAlloc
00003396   00403396    0   KERNEL32.dll
000033A6   004033A6    0   StartServiceCtrlDispatcherA
000033C4   004033C4    0   SetServiceStatus
000033D8   004033D8    0   RegisterServiceCtrlHandlerA
000033F6   004033F6    0   CloseServiceHandle
0000340C   0040340C    0   ControlService
0000341E   0040341E    0   QueryServiceStatus
00003434   00403434    0   OpenServiceA
00003444   00403444    0   CreateServiceA
00003456   00403456    0   OpenSCManagerA
00003468   00403468    0   DeleteService
00003478   00403478    0   StartServiceA
00003488   00403488    0   ChangeServiceConfigA
000034A0   004034A0    0   QueryServiceConfigA
000034B4   004034B4    0   ADVAPI32.dll
000034C4   004034C4    0   WSAIoctl
000034D0   004034D0    0   WSASocketA
000034DC   004034DC    0   WS2_32.dll
000034E8   004034E8    0   MFC42.DLL
000034F4   004034F4    0   memmove
00003506   00403506    0   fprintf
00003518   00403518    0   sprintf
00003522   00403522    0   perror
0000352C   0040352C    0   strstr
0000353E   0040353E    0   printf
00003546   00403546    0   MSVCRT.dll
00003554   00403554    0   __dllonexit
00003562   00403562    0   _onexit
0000356C   0040356C    0   _exit
00003574   00403574    0   _XcptFilter
00003582   00403582    0   __p___initenv
00003592   00403592    0   __getmainargs
000035A2   004035A2    0   _initterm
000035AE   004035AE    0   __setusermatherr
000035C2   004035C2    0   _adjust_fdiv
000035D2   004035D2    0   __p__commode
000035E2   004035E2    0   __p__fmode
000035F0   004035F0    0   __set_app_type
00003602   00403602    0   _except_handler3
00003616   00403616    0   _controlfp
00003624   00403624    0   ??0Init@ios_base@std@@QAE@XZ
00003644   00403644    0   ??1Init@ios_base@std@@QAE@XZ
00003664   00403664    0   ??0_Winit@std@@QAE@XZ
0000367C   0040367C    0   ??1_Winit@std@@QAE@XZ
00003692   00403692    0   MSVCP60.dll
00004049   00404049    0   ERROR 3
00004055   00404055    0   ERROR 2
00004061   00404061    0   ERROR 1
0000406C   0040406C    0   impossibile creare raw ICMP socket
00004098   00404098    0   RAW ICMP SendTo:
000040AE   004040AE    0   ========================= Icmp BackDoor V0.1
=======================
000040F4   004040F4    0   ========= Code by Spoof. Enjoy Yourself!
0000411E   0040411E    0   Your PassWord:
00004138   00404138    0   cmd.exe
```

```
00004142   00404142     0    Exit OK!
00004150   00404150     0   Local Partners Access
0000416A   0040416A     0   Error UnInstalling Service
0000418A   0040418A     0   Service UnInstalled Sucessfully
000041B2   004041B2     0   Error Installing Service
000041CE   004041CE     0   Service Installed Sucessfully
000041F5   004041F5     0   Create Service %s ok!
0000420D   0040420D     0   CreateService failed:%d
00004229   00404229     0   Service Stopped
0000423D   0040423D     0   Force Service Stopped Failed%d
00004260   00404260     0   The service is running or starting!
00004288   00404288     0   Query service status failed!
000042A8   004042A8     0   Open service failed!
000042C1   004042C1     0   Service %s Already exists
000042DC   004042DC     0   Local Printer Manager Service
000042FC   004042FC     0   smsses.exe
00004309   00404309     0   Open Service Control Manage failed:%d
00004338   00404338     0   Start service successfully!
00004358   00404358     0   Starting the service failed!
00004378   00404378     0   starting the service <%s>...
00004398   00404398     0   Successfully!
000043A8   004043A8     0   Failed!
000043B4   004043B4     0   Try to change the service's start type...
000043E0   004043E0     0   The service is disabled!
000043FC   004043FC     0   Query service config failed!
000062DB   004062DB     0   ?????
00005064   00405064     0   Hello from MFC!
000060F3   004060F3     0   \winnt\system32\smsses.exe
00006181   00406181     0   \winnt\system32\smsses.exe
000062B3   004062B3     0   \\199.107.97.191\C$
0000632F   0040632F     0   \winnt\system32
000063A7   004063A7     0   \winnt\system32\reg.exe
0000642F   0040642F     0   \winnt\system32\reg.exe
000064B7   004064B7     0   \winnt\system32\reg.exe
0000653F   0040653F     0   \winnt\system32\reg.exe
000065BD   004065BD     0   \winnt\system32\reg.exe
00006645   00406645     0   \winnt\system32\reg.exe
000066CD   004066CD     0   \winnt\system32\reg.exe
00006755   00406755     0   \winnt\system32\reg.exe
000067DD   004067DD     0   \winnt\system32\reg.exe
00005062   00405062     1   Hello from MFC!
```

Annex B - Source Code of ICMPlib_v1.h

```
/*
########## ICMPLIB_V1.h ##################################################
######################### ICMP Tunneling Library ##########################
########################################################## by FuSyS #########

                V.1 - NO (C)1998 FuSyS - TCP/IP Tools Unlimited

************************************************************************
*  COSA:  Una libreria in standard C per sfruttare la possibilita' *
*                       offerta dal protocollo ICMP di inserire dati all'interno *
*                       del datagramma.                                          *
*                                                                                *
*  CHI:                 individui dotati di una conoscenza base di C e TCP/IP    *
*                       che siano abbastanza fantasiosi da trovare un uso per    *
*                       questo tipo di codice. Se non avete questi requisiti,    *
*                       per favore impadronitevene prima di tornare a questa     *
*                       lib.                                                     *
*                                                                                *
*  OS:                  Linux 1.3.x e seguenti (raw sockets)                     *
*                                                                                *
*  TNX:                 Daemon9 e THC per i loro lavori                          *
*                                                                                *
*  LETTURE:             TCP/IP Illustrated Vol.1 di R.W.Stevens,                 *
*                       Project LOKI di Daemon9,                                 *
*                       /usr/include/*.h                                         *
************************************************************************

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <signal.h>
#include <errno.h>
extern int errno;

#include <sys/types.h>
#include <sys/time.h>
#include <sys/param.h>
#include <sys/socket.h>
#include <sys/file.h>
#include <netinet/in_systm.h>
#include <netinet/in.h>

#ifdef linux
 #include "linux_ip_icmp.h"
#else
 #include <netinet/ip_icmp.h>
 #include <netinet/ip.h>
#endif

#include <arpa/inet.h>
#include <netdb.h>

#define ECHO_TAG      0xF001
#define ECHO_LAST     0xF002
#define REPLY         1
#define LAST          1
#define YEAH                    1
#define NOPE                    0

#define ICMP_HDR                8               /* 8-byte ICMP header */
#define IP_HDR                                  20              /* 20-byte IP header */
#define MAXMESG                 4096    /* dati max*/
#define MAXPACKET               5004    /* dimensioni max del pacchetto */
                                        /* ICMP_HDR + MAXMESG */

        int     sockfd ;
        int             ip_spoof ;
```

```
        u_long      spoof_addr ;
        u_int       icmp_init = 1 ;
        struct      sockaddr_in  clisrc;

/*************************************************************************
* Funzioni per DNS e checksum - sempre le solite :) niente di nuovo qui         *
*************************************************************************/

u_long  nameResolve(char *hostname);
char    *hostLookup(u_long in);
u_short in_chksum(u_short *ptr, int nbytes);

u_long nameResolve(char *hostname)
{
 struct in_addr addr;
 struct hostent *hostEnt;

 if((addr.s_addr=inet_addr(hostname)) == -1)
 {
  if(!(hostEnt=gethostbyname(hostname)))
  {
        fprintf(stderr,"Errore nella risoluzione del nome:`%s`\n",hostname);
        exit(0);
  }
  bcopy(hostEnt->h_addr,(char *)&addr.s_addr,hostEnt->h_length);
 }
 return addr.s_addr;
}

char *hostLookup(u_long in)
{
 char hostname[1024];
 struct in_addr addr;
 struct hostent *hostEnt;


 bzero(&hostname,sizeof(hostname));
 addr.s_addr = in;
 hostEnt = gethostbyaddr((char *)&addr, sizeof(struct in_addr),AF_INET);

 if(!hostEnt)
  strcpy(hostname,inet_ntoa(addr));
 else
  strcpy(hostname,hostEnt->h_name);

 return(strdup(hostname));
}

u_short in_chksum(u_short *ptr, int nbytes)
{
 register long          sum;        /* assumes long == 32 bits */
 u_short               oddbyte;
 register u_short       answer;     /* assumes u_short == 16 bits */

 /*
  * Our algorithm is simple, using a 32-bit accumulator (sum),
  * we add sequential 16-bit words to it, and at the end, fold back
  * all the carry bits from the top 16 bits into the lower 16 bits.
  */

 sum = 0;
 while (nbytes > 1)
 {
  sum += *ptr++;
  nbytes -= 2;
 }

        /* mop up an odd byte, if necessary */
 if (nbytes == 1)
 {
```

```
  oddbyte = 0;           /* make sure top half is zero */
  *((u_char *) &oddbyte) = *(u_char *)ptr;   /* one byte only */
  sum += oddbyte;
 }

 /*
  * Add back carry outs from top 16 bits to low 16 bits.
  */

 sum  = (sum >> 16) + (sum & 0xffff);    /* add high-16 to low-16 */
 sum += (sum >> 16);                     /* add carry */
 answer = ~sum;          /* ones-complement, then truncate to 16 bits */

 return((u_short) answer);
}

/**************************************************************************
********************** Ed ora .... s_C_iotaim =;) ***********************
**************************************************************************/

void ICMP_init(void)
{
        int spoof_opt = 1;

 if(icmp_init)
 {
  if (ip_spoof == NOPE) {
        if((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) < 0 ) {
                fprintf(stderr, "Impossibile creare raw ICMP socket ");
                exit(0);
        }
  }
  if (ip_spoof == YEAH) {
        if((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) < 0 ) {
       fprintf(stderr, "Impossibile creare raw socket ");
       exit(0);
        }
        if(setsockopt(sockfd, IPPROTO_IP, IP_HDRINCL, &spoof_opt,
                sizeof(spoof_opt)) < 0 ) {
                fprintf(stderr,"Impossibile creare IP Header ");
                exit(0);
        }
  }
  icmp_init = 0;
 }
}

void ICMP_reset(void)
{
 close(sockfd);
 icmp_init = 1;
}

int ICMP_send
(char *send_mesg, size_t mesglen, u_long dest_ip, int echo, int last)
{
 int             sparato;
 struct tunnel {
     struct icmp    icmp;
     u_char         data[MAXMESG];
 } icmp_pk;
 int             icmplen = sizeof(struct icmp);
 int             pach_dim;
 struct sockaddr_in   dest;
 int             destlen;

 if(mesglen > MAXMESG)
  return(-1);

 if(icmp_init)
```

```
  ICMP_init();

 destlen = sizeof(dest);
 bzero((char *) &dest, destlen);
 dest.sin_family      = AF_INET;
 dest.sin_addr.s_addr  = dest_ip;

 pach_dim = mesglen + sizeof(struct icmp);
 memset(&icmp_pk, 0, pach_dim);
 icmp_pk.icmp.icmp_type = ICMP_ECHOREPLY;
 bcopy(send_mesg, icmp_pk.icmp.icmp_data, mesglen);
 icmp_pk.icmp.icmp_cksum = in_chksum((u_short *) &icmp_pk.icmp,
                                     (sizeof(struct icmp)+mesglen));
 if(echo) icmp_pk.icmp.icmp_seq = ECHO_TAG;
 if(last) icmp_pk.icmp.icmp_seq = ECHO_LAST;

 if( (sparato = sendto(sockfd, &icmp_pk, pach_dim, 0, (struct sockaddr *)
          &dest, destlen)) < 0 ) {
                perror("RAW ICMP SendTo: ");
                return(-1);
 }
 else if(sparato != pach_dim) {
          perror("Dimensioni pacchetto IP errate: ");
          return(-1);
 }
 return(sparato);
}

int ICMP_sp_send(char *send_mesg, size_t mesglen, u_long dest_ip, u_long sp_ip)
{
 int              sparato;
 struct spoof {
          struct ip     ip;
          struct icmp             icmp;
          u_char          data[MAXMESG];
 } sp_pk;
 int                              iplen = sizeof(struct ip);
 int                              icmplen = sizeof(struct icmp);
 int              pach_dim;
 struct sockaddr_in    dest;
 int              destlen;

 if(mesglen > MAXMESG)
  return(-1);

 if(icmp_init)
  ICMP_init();

 destlen = sizeof(dest);
 bzero((char *) &dest, destlen);
 dest.sin_family      = AF_INET;
 dest.sin_addr.s_addr  = dest_ip;

 pach_dim = mesglen + sizeof(struct ip) + sizeof(struct icmp);
 memset(&sp_pk, 0, pach_dim);

 sp_pk.ip.ip_v = 4;
 sp_pk.ip.ip_hl = 5;
 sp_pk.ip.ip_len = htons(iplen + icmplen + mesglen);
 sp_pk.ip.ip_ttl = 255;
 sp_pk.ip.ip_p = IPPROTO_ICMP;
 sp_pk.ip.ip_src.s_addr = sp_ip;
 sp_pk.ip.ip_dst.s_addr = dest_ip;

 sp_pk.icmp.icmp_type = ICMP_ECHOREPLY;
 bcopy(send_mesg, sp_pk.icmp.icmp_data, mesglen);
 sp_pk.icmp.icmp_cksum = in_chksum((u_short *) &sp_pk.icmp,
                                     (sizeof(struct icmp)+mesglen));

 if((sparato = sendto(sockfd, &sp_pk, pach_dim, 0, (struct sockaddr *)
```

```
                        &dest, destlen)) < 0 ) {
        perror("RAW ICMP SendTo: ");
        return(-1);
    }
    if(sparato != pach_dim) {
        perror("Dimensioni pacchetto IP errate: ");
        return(-1);
    }
    return(sparato);
}

int ICMP_recv(char *recv_mesg, size_t mesglen, int echo)
{
    struct recv {
            struct ip     ip;
            struct icmpicmp;
            char                    data[MAXMESG];
    } rcv_pk;
    int         pach_dim;
    int         accolto;
    int   iphdrlen;
    int   clilen = sizeof(clisrc);

    if(icmp_init)
      ICMP_init();

    while(1)
    {
      pach_dim = mesglen + sizeof(struct ip) + sizeof(struct icmp);
      memset(&rcv_pk, 0, pach_dim);
      if( (accolto = recvfrom(sockfd, &rcv_pk, pach_dim, 0, (struct
            sockaddr *) &clisrc, &clilen)) < 0 )
        continue;

      iphdrlen = rcv_pk.ip.ip_hl << 2;
      if(accolto < (iphdrlen + ICMP_MINLEN))
        continue;
      accolto -= iphdrlen;

     if(!echo){
      if(!rcv_pk.icmp.icmp_id && !rcv_pk.icmp.icmp_code &&
            rcv_pk.icmp.icmp_type == ICMP_ECHOREPLY && rcv_pk.icmp.icmp_seq !=
            ECHO_TAG && rcv_pk.icmp.icmp_seq != ECHO_LAST)
        break;
     }
     if(echo){
      if(!rcv_pk.icmp.icmp_id && !rcv_pk.icmp.icmp_code &&
            rcv_pk.icmp.icmp_type == ICMP_ECHOREPLY
         && (rcv_pk.icmp.icmp_seq == ECHO_TAG || rcv_pk.icmp.icmp_seq ==
            ECHO_LAST) )
        break;
     }
     }
       if(!echo){
            accolto -= ICMP_HDR;
            bcopy(rcv_pk.icmp.icmp_data, recv_mesg, accolto);
            return(accolto);
       }
       if(echo){
            if(rcv_pk.icmp.icmp_seq == ECHO_TAG) {
            accolto -= ICMP_HDR;
            bzero(recv_mesg, sizeof(recv_mesg));
            bcopy(rcv_pk.icmp.icmp_data, recv_mesg, accolto);
            return(accolto);
            }
         return(-666);
       }
```

Last Updated: July 18th, 2018

Upcoming SANS Forensics Training

Community SANS Columbia FOR500	Columbia, MD	Jul 23, 2018 - Jul 28, 2018	Community SANS
SANS Riyadh July 2018	Riyadh, Kingdom Of Saudi Arabia	Jul 28, 2018 - Aug 02, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LA	Jul 30, 2018 - Aug 06, 2018	Live Event
San Antonio 2018 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	vLive
SANS August Sydney 2018	Sydney, Australia	Aug 06, 2018 - Aug 25, 2018	Live Event
SANS Boston Summer 2018	Boston, MA	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS New York City Summer 2018	New York City, NY	Aug 13, 2018 - Aug 18, 2018	Live Event
Mentor Session - AW FOR508	Phoenix, AZ	Aug 14, 2018 - Sep 13, 2018	Mentor
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
Data Breach Summit & Training 2018	New York City, NY	Aug 20, 2018 - Aug 27, 2018	Live Event
SANS Chicago 2018	Chicago, IL	Aug 20, 2018 - Aug 25, 2018	Live Event
Community SANS Columbia FOR610	Columbia, MD	Aug 20, 2018 - Aug 25, 2018	Community SANS
Mentor Session - FOR508	Copenhagen, Denmark	Aug 22, 2018 - Oct 06, 2018	Mentor
SANS San Francisco Summer 2018	San Francisco, CA	Aug 26, 2018 - Aug 31, 2018	Live Event
SANS Copenhagen August 2018	Copenhagen, Denmark	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, Japan	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Tampa-Clearwater 2018	Tampa, FL	Sep 04, 2018 - Sep 09, 2018	Live Event
SANS vLive - FOR585: Advanced Smartphone Forensics	FOR585 - 201809,	Sep 04, 2018 - Oct 11, 2018	vLive
Threat Hunting & Incident Response Summit & Training 2018	New Orleans, LA	Sep 06, 2018 - Sep 13, 2018	Live Event
Threat Hunting & IR Summit - FOR572: Advanced Network Forensics and Analysis	New Orleans, LA	Sep 08, 2018 - Sep 13, 2018	vLive
Threat Hunting & IR Summit - FOR526: Memory Forensics In-Depth	New Orleans, LA	Sep 08, 2018 - Sep 13, 2018	vLive
Threat Hunting & IR Summit - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	New Orleans, LA	Sep 08, 2018 - Sep 13, 2018	vLive
SANS Baltimore Fall 2018	Baltimore, MD	Sep 08, 2018 - Sep 15, 2018	Live Event
SANS Munich September 2018	Munich, Germany	Sep 16, 2018 - Sep 22, 2018	Live Event
Community SANS Columbia FOR508	Columbia, MD	Sep 17, 2018 - Sep 22, 2018	Community SANS
Community SANS Madrid FOR508 (in Spanish)	Madrid, Spain	Sep 17, 2018 - Sep 22, 2018	Community SANS
SANS London September 2018	London, United Kingdom	Sep 17, 2018 - Sep 22, 2018	Live Event
Community SANS Toronto FOR508	Toronto, ON	Sep 17, 2018 - Sep 22, 2018	Community SANS
Network Security 2018 - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	Las Vegas, NV	Sep 23, 2018 - Sep 28, 2018	vLive