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

EXECUTIVE SUMMARY
GIAC Certified Forensic Analyst
Practical Assignment
Version 1.2 (December, 2002)
Helen Psaila – 19th May, 2003

Overview

The assignment consists of three parts:

Part One – Analyse an unknown binary,
Part Two – Option 1 -Perform Forensic Analysis on a suspected compromised system, and
Part Three – Legal Issues of Computer Incident Handling.

Part One – Analyse an unknown binary

This involved doing a forensic analysis on a file. The aim was to analyse the file and determine what it was and what it was used for.

It was analysed on a system that was not connected to the network in a strictly controlled manner that would ensure that if it had a destructive payload, would not put any other systems at risk.

To ensure that the file was not alte red during the investigation, it was analysed in read-only mode at all times. I was able to determine that the file was a Unix-based executable, which was only able to run on systems with older versions of library files.

I discovered that this binary was a file called Lokid, which had been renamed to hide its true identity. It had no destructive capabilities and is used as an ICMP tunnelling program. Running the program is not illegal and as such no criminal action can be taken against the person using and possessing this program.

Part Two -Option 1 - Perform Forensic Analysis on a suspected compromised system

For the forensic analysis section, I built a Windows XP system that was connected to the Internet via a high -speed cable modem connection. It was connected to the Internet intermittently to determine if anyone was able to compromise the system.

Eventually, the system started to display unusual activity so it was shutdown and a forensic analysis was done.

By going through a file timeline and relevant logs, intruder type activity was found. The final determination was that numerous people probed the system for vulnerabilities. Most of the attempts were either from virus activity or "script kiddies" but not all. The final conclusion was that although the system was probed, it was not compromised.

Part Three – Legal Issues of Computer Incident Handling

The final part of the assignment covers the Australian legal responsibilities of an Internet Service Provider (ISP) conducting their business in Australia and their obligations when dealing with local Law Enforcement officers.

In Australia, ISPs are under no obligation to provide Law Enforcement with any information what so ever unless the officer has a warrant or court order that covers the information that is required by the officer.

However, although the ISP is not obliged to provide the information they can willingly provide a limited amount of information to Law Enforcement providing they do not breach information privacy acts.

Conclusion

All three parts deal with information handling as it relates to computer investigations in one form or another. There are many issues that incident handlers need to consider and when responding to incidents and requests for more information. Some of these issues are procedural, some are legal and others are Organisational. Regardless of which category they fall under an Incident Responder must ensure that they maintain the highest standards possible, ensuring that the rights and obligations of people and companies are always handled in a responsible and professional manner free from any bias or misrepresentation.

SANS

GIAC Certified Forensic Analyst

Practical Assignment

Version 1.2 (December, 2002)

Helen Psaila
19th May, 2003

Table of Contents

Part One – Analyse an Unknown Binary

Initial response

Upon arriving at work, I am contacted by our on -call Incident Response Team. The team was called in overnight to analyse a suspected compromised system. They have spent all night probing the system and have discovered a binary, which appears to be suspect. As they are at the end of their shift, they ask if I can analyse the file for them. They have placed the file on one of the servers in a zip format. I am given the name and location of the file and have been given access to the directory so that I can download it and start my analysis immediately.

Before downloading the file I set up my forensic workstation. This involves connecting a second hard drive, which I have locked away, to my laptop. I remove the current hard drive and I replace it with this second hard drive, which consists of Linux Red Hat 7.3 and specific forensic tools that I use. This hard drive is sterilised [1] and rebuilt after each investigation I do, this means that it's ready to go whenever I need it. When undertaking an investigation I try to avoid connecting it to the network, if at all possible, so that I don't have to worry about this system being compromis ed whilst I'm in the process of analysing data. This is especially important seeing that many investigations I conduct point to it being "an inside job". People who have been implicated often take any measures possible to destroy evidence once they become aware that an investigation is under way. They will manipulate and delete data being analysed, attempt to hide their tracks and/or try to compromise the forensic investigation process so that it will be questioned when it goes to court. I try to never un derestimate the measures that people will go to in order to cover things up which is another reason why I will never leave the forensic workstation unattended or unsecured. On booting the forensic workstation I manually verify that it is set to the current time and time zone before commencing work. I would like to verify this time with the time set on the suspected compromised server to ensure that there is no time deviation but firstly, I have no idea which server to look at and, secondly, whether or not the Response Team have altered anything that may have changed the server time.

Commencement of Investigation Process – Binary Details

The next step involves downloading the file to a floppy disk. This is done from another system still connected to the network, which I also use during my investigation when I need to connect to the Internet to search for information. I download the file and run Winzip which show me two files to extract, "atd" and "atd.md5" from the Winzip archive file. I theorise that the unknown binary is the "atd" file and that the other file, "atd.md5", has possibly been created by

[1] The process used to sterilise computer media involves executing the following command "dd if=/dev/zero of=/dev/xxx bs=yyy count=1" where xxx represents the device i.e. floppy/hard drive and yyy represents the size of the media being sterilised.

the Response Team using an MD5 [2] tool when they acquired the binary. However, I need to prove this as fact because it could be a binary hidden to look like an MD5 file or if it really is an actual MD5 file the intruder may have been using it to determine if system administrators alter his "atd" binary in an attempt to clean the system or catch him. Of particular importance to my investigation is the fact that running md5sum on a file changes it's MAC (Modify, Access and Change) time so I need to determine if the last access time on the "atd" file is accurate.

I extract both files to a floppy disk which has been sterilised using the method previously mentioned, I remove it from the drive, and then as a precaution, write protect it using the write -protect tab on the floppy disk itself to ensure that no further data can be written to the disk.

This disk is then placed in the floppy drive of my forensic workstatio n and the drive mounted as read only using the following command:

```
[root@localhost root]# mount  -o ro,loop,nosuid,noexec.nodev,noatime
/mnt/floppy
```

Firstly, I want to determine if the "atd.md5" file is the MD5Sum of the "atd" file. I run strings on atd.md5 and receive the following output:

```
[root@localhost root]# strings /mnt/floppy/atd.md5
48e8e8ed3052cbf637e638fa82bdc566  atd
[root@localhost root]#
```

I then run md5sum on the "atd" file.

```
[root@localhost root]# md5sum /mnt/floppy/atd
48e8e8ed3052cbf6 37e638fa82bdc566  /mnt/floppy/atd
[root@localhost root]#
```

They are both the same so I now know that atd.md5 is the hash for "atd". Now to see if the integrity of the file has been preserved or if the file has has been altered by when the original md5sum was calculated. To do this I run the "stat" command on both files.

[2] MD5 is an algorithm "fingerprint" used to verify file and data integrity. Any alteration to a file or data will change this "fin gerprint" so it is obvious to see when it is no longer the same as the original.

```
[root@localhost root]# stat /mnt/floppy/atd
  File: "/mnt/floppy/atd"
  Size: 15348           Blocks: 30         IO Block:   -
4611693921167212032 Regular File
Device: 700h/1792d      Inode : 5           Links: 1
Access: (0755/ -rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/
root)
Access: Thu Aug 22 13:57:54 2002
Modify: Thu Aug 22 13:57:54 2002
Change: Thu Aug 22 13:57:54 2002

[root@localhost root]# stat /mnt/floppy/atd.md5
  File: "/mnt/floppy/atd.md5"
  Size: 39              Blocks: 1          IO Block:   -
4611693921167212032 Regular File
Device: 700h/1792d      Inode: 6           Links: 1
Access: (0755/ -rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/
root)
Access: Thu Aug 22 13:58:0 8 2002
Modify: Thu Aug 22 13:58:08 2002
Change: Thu Aug 22 13:58:08 2002
```

As can be seen above the MAC times are different but only marginally and this difference may be due to the time it took to cut, paste and save the hash sum to the md5 file. This means that I will have to assume that the last accessed time on "atd" may have been altered by who ever created atd.md5 file and is therefore not an accurate indication of when a server process or the intruder last accessed it. This has an impact on my inv estigation because I won't be able to determine the last time the program was run. This emphasises the need for caution when conducting an investigation because every action undertaken has the potential to alter the crime scene.

Now, I will analyse the binary to determine what it is. So far I know the name of the file (atd), the size of the file (15348 kb as shown by the stat command), the MAC times and also the GID and UID. The GID is the group ID associated with the file and the UID shows the User ID as sociated with the file. I can also see that it is not a SUID root file [3] so this is good to know.

As can bee seen from the stat command, both the UID and GID are associated with "root". However, this doesn't mean much because knowing what has been done so far in collecting the file I am assuming that the investigators could have changed the ownership during the process used to collect the file so I will not make any definitive assumptions on ownership and permissions of the file.

I decide to search the Internet using Google's search engine to see if "atd" is a known executable and I discover that the file "atd" is a Unix based file that starts the "atd" daemon. This daemon runs commands at a specified time as

[3] A SUID root file allows unprivileged users to run it as it relies on the security of the program rather than the user. This is dangerous because it means that anyone can run the file. If a hacker were to replace a real SUID file with one of their own it could provide them with a back door and complete system access.

scheduled by the "at" command. The file I am analysing may be this or it may have been renamed to look like it so that it's real purpose wouldn't be obvious or to hide a command imbedded in it. The main thing I will need to be careful of is that when I run commands during my investigation that the co mmand runs on the binary and not on the system "atd" file. To prevent this from occurring I decide that the safest thing to do is to the rename the system "atd" file to "atd-sys".

Program Description

To see what type of file I am dealing with I run the "file" command on "atd". The output of this shows me the following:

```
[root@localhost root]# file /mnt/floppy/atd
/mnt/floppy/atd: ELF 32 -bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), stripped
[root@localhost roo t]#
```

I now know that it's an ELF (Executable Linking File), which is used on Unix based systems so it's definitely not a Windows or MSDOS file. It's an executable file designed for Intel hardware, and it uses shared libraries present on the system. It has been stripped so it discards symbols from object files, which optimises it for speed and performance.

I also have a look at the renamed "atd" file on my system. Running "file" on it also gives me the same output so I still have no idea of the file's purpose or if the two files serve the same purpose.

Next, I run the "strings -a" command to see what readable text is in the file and I notice text that says "LOKI2 route [© 1997 guild corporation worldwide]". There are also numerous text references to lokid. This is a great clue and searching the Internet using this information with Google I appear to find what I am looking for at http://www.phrack.org/show.php?p=49&a=6 and http://www.phrack.org/show.php?p=51&a=6 . The site also contains the source code for the program so I would now be able to compile the entire program to see it's full capabilities.

Phrack Magazine Vol 7, Issue 49, file 6 of 16 which is on the Phrack site states that:

> 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_ECHOtraffic. They simply pass t hem, drop them, or return them. The trojan packets themselves are masqueraded as common ICMP_ECHO traffic. We can encapsulate (tunnel) any information we want. From here on out, Loki traffic will

refer to ICMP_ECHO traffic that tunnels information. (Astute readers will note that Loki is simply a form of steganography).

Loki is not a compromise tool. It has many uses, none of which are breaking into a machine. It can be used as a backdoor into a system by providing a covert method of getting commands e xecuted on a target machine. It can be used as a way of clandestinely leeching information off of a machine. It can be used as a covert method of user-machine or user-user communication. In essence the channel is simply a way to secretly shuffle data (confidentiality and authenticity can be added by way of cryptography).

So now I suspect that the program I have is an ICMP [4] tunnelling program used to cover a two-way network communication session established between two systems. It relies on two components , a server and a client. Loki is the client portion and Lokid is the server daemon installed on the compromised system, and these two bundled together are known as Loki2. According to Phrack Magazine Vol 7, Issue 51, article 6 of 17, "This is not a clandestine program. You want clandestine? Implement LOKI2 as an lkm, or, even better, write kernel diffs and make it part of the O/S". This could be a clue as to why the program name is "atd" instead of the default "loki" or "lokid" as the perpetrator may have tried to hide the program without having to go to the effort of making it an lkm [5].

As previously mentioned I have no way of finding out when the program was last run on the compromised system but I will see what other information the "strings" command can give me.

The first two lines of the program show me that it uses ld -linux.so.1 and libc.so.5, which haven't been in use for some years so I know the program compilation, is not recent. In fact by doing a "strings –a atd | fgrep GCC" I discover that the GCC compiler version used was "GCC: (GNU) 2.7.2.1" which was in use in 1996. This timeframe ties in with the Loki Project, which was published the following year, 1997.

Next I want to see if I can display any compiler and runtime linker symbol tables. I run the "nm" command and receive a message that says no symbols are found. Then I try the "ldd" command to see if I can identify any dynamic libraries used and receive an error message that says "No such file or directory". Lastly, I run "objdump" and als o receive no useful information.

I decide to run all of these commands on my systems "atd" file to check for similarities and I am able to determine that the binary is not the same type of file. This is great because I now know that the binary I have has nothing to do with "at" Daemon. If the file is Lokid, then it has been renamed to mask its true purpose.

[4] ICMP is Internet Control Message Protocol and is used by machines to negotiate packet delivery.
[5] An lkm is a loadable kernel module. LKMs are generally used by the operating system to load device drivers. A rootkit using a LKM is extremely hard to detect and is therefore most beneficial to a hacker.

Forensic Details

My only option now is to run the file to see what it does. To determine how portable the file is I decided to try running i t on my forensic system. Normally, I would not consider doing this because of the potential havoc it could wreak but I am only investigating one file so if it corrupts the system I can always start again. Given that my forensic system is not on the network the main issue I am concerned about is it altering my system files but as part of my analysis I will monitor what the file does when it runs. I want to know if it opens/uses any system ports, applications, processes or does anything else notable. The tool I am going to use is apptrace, which will track system calls, so I be able to determine if the file tries to access the network, file system, memory or other system functions.

Having set apptrace up, I am now ready to run the file. I try to run it and it doesn't run, and I receive the error message "command not found". Given that I have discovered that the GCC information I found in this file is old it would appear that it won't run with my version of Linux so I need to install and test the binary on an older system. So I now know that the file is not very portable at all as it relies on old system libraries and can only run on systems with these older library versions unless it is updated/rewritten to run on newer versions.

I manage to find an old Linux version, Slackware 3.0 that has the GCC 2.7.2 compiler, which should be the correct version needed to run the binary as it has the correct ld-linux.so.1 and libc.so.5 libraries. To start with I will run the binary whilst this system is disconnected from the network, so I configure it as a stand-alone device. Later when I had more information and, if necessary, I will reconfigure it to be set up on an isolated network using two PCs.

I copy the binary to /usr/local/src, and then run the "ps" command to see what processes are running prior to executing the binary. The output of which can be seen below.

```
PID TTY STAT  TIME COMMAND
    1  ?  S     0:08 init [3]
    2  ?  SW    0:00 (kflushd)
    3  ?  SW<   0:00 (kswapd)
    4  ?  SW    0:00 (nfsiod)
    5  ?  SW    0:00 (nfsiod)
    6  ?  SW    0:00 (nfsiod)
    7  ?  SW    0:00 (nfsiod)
   13  ?  S     0:00 update (bdflush)
   72  ?  S     0:00 /usr/sbin/crond  -l10
   83  ?  S     0:00 /usr/sbin/syslogd
   85  ?  S     0:00 /usr/sbin/klogd
   89  ?  S     0:00 /usr/sbin/inetd
   91  ?  S     0:00 /usr/sbin/lpd
   94  ?  S     0:00 /usr/sbin/rpc.mountd
   96  ?  S     0:00 /usr/sbin/rpc.nfsd
  102  ?  S     0:00 sendmail: accepting connections
  107   1 S     0:00  -bash
  108   2 S     0:00 /sbin/a getty 38400 tty2 linux
  109   3 S     0:00 /sbin/agetty 38400 tty3 linux
  110   4 S     0:00 /sbin/agetty 38400 tty4 linux
  111   5 S     0:00 /sbin/agetty 38400 tty5 linux
  112   6 S     0:00 /sbin/agetty 38400 tty6 linux
  125   1 R     0:00 ps
```

I set up my first tool, Apptrace and then run the binary "atd". I know apptrace is working correctly when it displays the following message "Process 137 attached". Now to see what has occurred, I stop Apptrace and run "ps" again:

```
 PID TTY STAT  TIME  COMMAND
    1  ?  S     0:08 init [3]
    2  ?  SW    0:00 (kflushd)
    3  ?  SW<   0:00 (kswapd)
    4  ?  SW    0:00 (nfsiod)
    5  ?  SW    0:00 (nfsiod)
    6  ?  SW    0:00 (nfsiod)
    7  ?  SW    0:00 (nfsiod)
   13  ?  S     0:00 update (bdflush)
   72  ?  S     0:00 /usr/sbin/crond  -l10
   83  ?  S     0:00 /usr/sbin/syslogd
   85  ?  S     0:00 /usr/sbin/klogd
   89  ?  S     0:00 /usr/sbin/inetd
   91  ?  S     0:00 /usr/sbin/lpd
   94  ?  S     0:00 /usr/sbin/rpc.mountd
   96  ?  S     0:00 /usr/sbin/rpc.nfsd
  102  ?  S     0:00 sendmail: accepting connections
  107   1 S     0:00  -bash
  108   2 S     0:00 /sbin/agetty 38400 tty2 linux
  109   3 S     0:00 /sbin/agetty 38400 tty3 linux
  110   4 S     0:00 /sbin/agetty 38400 tty4 linux
  111   5 S     0:00 /sbin/agetty 38400 tty5 linux
  112   6 S     0:00 /sbin/agetty 38400 tty6 linux
  129   1 T     0:00 bash ./atd
  133   1 T     0:00 strace  -f -o /root/apptrace/atd.129.trace
./atd.orig
  135  ?  S     0:00 ./atd.orig
  137   1 R     0:00 ps
```

I can now see that "atd' is running as a process. The log file apptrace creates is as follows:

```
134   mmap(0, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|0x20,
4294967295, 0) = 0x40006000
134   mprotect(0x8048000, 13604, PROT_READ|PROT_WRITE|PROT_EXEC) = 0
134   stat("/etc/ld.so.cache", {st_mode=S_IFREG|0644, st_size=3475,
...}) = 0
134   open("/etc/ld.so.cache", O_RDONLY) = 4
134   mmap(0, 3475, PROT_READ, MAP_SHARED, 4, 0) = 0x40007000
134   close(4)                          =   0
134   open("/lib/libc.so.5.3.12", O_RDONLY) = 4
134   read(4, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3"..., 4096) = 4096
134   mmap(0, 724992, PROT_NONE, MAP_PRIVATE|0x20, 4294967295, 0) =
0x40008000
134   mmap(0x40008000, 495550, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED, 4, 0) = 0x40008000
134   mmap(0x40081000, 23472, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED, 4, 0x78000) = 0x40081000
134   mmap(0x40087000, 203928, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|0x20, 4294967295, 0) = 0x40087000
134   close(4 )                          = 0
134   mprotect(0x40008000, 495550, PROT_READ|PROT_WRITE|PROT_EXEC) =
0
134   munmap(0x40007000, 3475)          = 0
134   mprotect(0x8048000, 13604, PROT_READ|PROT_EXEC) = 0
134   mprotect(0x40008000, 495550, PROT_READ|PROT_EXEC) = 0
134   SYS_136(0, 0x1, 0x4, 0x40001fb0, 0x8048d38) = 0
134   geteuid()                         = 0
134   getuid()                          = 0
134   brk(0x804c818)                    = 0x804c818
134   brk(0x804d000)                    = 0x804d000
134   open("/usr/share/locale/C/LC_MESSAGES", O_RDONLY) =   -1 ENOENT
(No such file or directory)
134   stat("/etc/locale/C/libc.cat", 0xbffff698) =   -1 ENOENT (No
such file or directory)
134   stat("/usr/lib/locale/C/libc.cat", 0xbffff698) =   -1 ENOENT (No
such file or directory)
134   stat("/usr/lib/locale/libc/C", 0xbffff698) =   -1 ENOENT (No
such file or directory)
134   stat("/usr/share/locale/C/libc.cat", 0xbffff698) =   -1 ENOENT

(No such file or directory)
134   stat("/usr/local/share/locale/C/libc.cat", 0  xbffff698) =  -1
ENOENT (No such file or directory)
134   socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 4
134   sigaction(SIGUSR1, {0x804a6b0, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}) = 0
134   socket(PF_INET, SOCK_RAW, IPPROTO_??? (0xff)) = 5
134   setsockopt(5, IPPROTO_IP3, [1], 4) = 0
134   getpid()                          = 134
134   getpid()                          = 134
134   shmget(376, 240, IPC_CREAT|0)     = 0
134   semget(558, 1, IPC_CREAT|0x180|0600) = 0
134   shmat(0, 0, 0)                     = 0x40007000
134   write(2, " \nLOKI2\troute [(c) 1997 guild c"..., 52) = 52
134   time([1049161628])                = 1049161628
134   close(0)                          = 0
```

```
134   sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}) = 0
134   sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}) = 0
134   sigaction(SIGTSTP, {SIG_IGN}, {SIG_DFL}) = 0
134   fork()                                   = 135
134   close(5)                                 = 0
134   close(4)                                 = 0
134   semop(0, 0x2, 0, 0xbffffb10)             = 0
134   shmdt(0x40007000)                        = 0
134   semop(0, 0x1, 0, 0xbffffb10)             = 0
134   _exit(0)                                 = ?
135   setsid()                                 = 135
135   open("/dev/tty", O_RDWR)                 =   -1 ENXIO (No such device
or address)
135   chdir("/tmp")                            = 0
135   umask(0)                                 = 022
135   sigaction(SIGALRM, {0x8049218, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}) = 0
135   alarm(3600)                              = 0
135   sigaction(SIGCHLD, {0x8049 900, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}) = 0
135   read(4,
```

I can now see what the binary is doing. It allocates memory, tries to open some libraries that aren't present, changes allocated space and opens an ICMP protocol socket. It also tries to open a terminal device that it can't locate. It doesn't alter any system files and doesn't display any Trojan, virus or backdoor characteristics.

All of this information ties in with the information in the Phrack paper. However, I will downlo ad and run lokid to confirm that the two programs are identical.

If the Incident Response Team requires more specific information on what this binary does I will need to get someone who specialises in this area to break down the code in to more detail.

Program Identification

To definitively identify the program I download the source code from http://www.phrack.com/show.php?p=51&a=6 on to the Slackware 3.0 system and unzip it and grab the relevant file needed to compile Loki. I also download the extract.txt file from the same issue so that I can extract the code correctly from the text files into "C" files. I edit out the Perl code so that I can use the "C" script and save the file as extract.c. I am now ready to start.

I run "gcc –o extract extract.c as per the source code instructions and this creates the extract file. The next command is "./extract p51 -06". This creates a directory called L2 that contains all of the files required to compile Loki. The last command I run is "make linux". Now I hav e all of the files needed to run Loki, i.e. Loki and Lokid.

I suspect that the binary I have is Lokid because of the proliferation of this phrase when I previously ran the “strings” command on the file. I decide to run the “strings” command on “Lokid” and compare the output to my original output for “atd”. Apart from a few lines being different it is quite clear that they are both the same file. The differences in the files could be attributed to the fact that they were compiled using different GCC versi ons; this also means there is no point in running md5sum on the files because the outcome would also be different.

Before doing anything else I decide to run “apptrace” on Lokid to compare the results. As can be seen below, the output is the same as the output from “atd” so I now know that the programs are indeed the same.

```
130   mmap(0, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|0x20,
4294967295, 0) = 0x40006000
130   mprotect(0x8048000, 13668, PROT_READ|PROT_WRITE|PROT_EXEC) =
0
130   stat("/etc/ld.so.cach e", {st_mode=S_IFREG|0644,
st_size=3475, ...}) = 0
130   open("/etc/ld.so.cache", O_RDONLY) = 4
130   mmap(0, 3475, PROT_READ, MAP_SHARED, 4, 0) = 0x40007000
130   close(4)                          = 0
130   open("/lib/libc.so.5.3.12", O_RDONLY) = 4
130   read(4, " \177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3"..., 4096) = 4096
130   mmap(0, 724992, PROT_NONE, MAP_PRIVATE|0x20, 4294967295, 0)
= 0x40008000
130   mmap(0x40008000, 495550, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED, 4, 0) = 0x40008000
130   mmap(0x40081000,  23472, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED, 4, 0x78000) = 0x40081000
130   mmap(0x40087000, 203928, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|0x20, 4294967295, 0) = 0x40087000
130   close(4)                          = 0
130   mprotect(0x40008000, 495550, PROT_READ|PROT_WRITE|PROT_EXEC)
= 0
130   munmap(0x40007000, 3475)          = 0
130   mprotect(0x8048000, 13668, PROT_READ|PROT_EXEC) = 0
130   mprotect(0x40008000, 495550, PROT_READ|PROT_EXEC) = 0
130   SYS_136(0, 0x1, 0x4, 0x40001fb0, 0x8048d38  ) = 0
130   geteuid()                         = 0
130   getuid()                          = 0
130   brk(0x804c858)                    = 0x804c858
130   brk(0x804d000)                    = 0x804d000
130   open("/usr/share/locale/C/LC_MESSAGES", O_RDONLY) =   -1
ENOENT (No such file or directory)
130   stat("/etc/locale/C/libc.cat", 0xbffff694) =   -1 ENOENT (No
such file or directory)
130   stat("/usr/lib/locale/C/libc.cat", 0xbffff694) =   -1 ENOENT
(No such file or directory)
130   stat("/usr/lib/locale/libc/C", 0xbffff694) = -1 ENOENT
(No such file or directory)
130   stat("/usr/share/locale/C/libc.cat", 0xbffff694) =   -1 ENOENT
(No such file or directory)
130   stat("/usr/local/share/locale/C/libc.cat", 0xbffff694) =   -1
ENOENT (No such file or directory)

130   socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 4
```

```
130   sigaction(SIGUSR1, {0x804a6ec, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}) = 0
130   socket(PF_INET, SOCK_RAW, IPPROTO_??? (0xff)) = 5
130   setsockopt(5, IPPROTO_IP3, [1], 4) = 0
130   getpid()                           = 130
130   getpid()                          = 130
130   shmget(372, 240, IPC_CREAT|0)     = 0
130   semget(554, 1, IPC_CREAT|0x180|0600) = 0
130   shmat(0, 0, 0)                    = 0x40007000
130   write(2, " \nLOKI2\troute [(c) 1997 guild c"..., 52) = 52
130   time([1049161862])                = 1049161862
130   close(0)                          = 0
130   sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}) = 0
130   sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}) = 0
130   sigaction(SIGTSTP, {SIG_IGN }, {SIG_DFL}) = 0
130   fork()                            = 131
130   close(5)                          = 0
130   close(4)                          = 0
130   semop(0, 0x2, 0, 0xbffffb0c)      = 0
130   shmdt(0x40007000)                 = 0
130   semop(0, 0x1, 0, 0xbffffb0c)      = 0
130   _exit(0)                          = ?
131   setsid()                          = 131
131   open("/dev/tty", O_RDWR)          = -1 ENXIO (No such device
or address)
131   chdir("/tmp")                     = 0
131   umask(0)                           = 022
131   sigaction(SIGALRM, {0x8049220, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}) = 0
131   alarm(3600)                       = 0
131   sigaction(SIGCHLD, {0x804991c, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}) = 0
131   read(4,
```

Legal Implications

The legal implications are interesting because running the program is not illegal. It is merely a tunnelling program that uses the ICMP protocol. It is not used to gain access to a system and in fact access must fi rst be gained prior to installing it. It could be argued that the software was being used to conduct illegal activities but there is no evidence to support this. Its use may be perfectly legitimate, such as a system administrator using it as their preferred remote access tool to administer the server. The fact that it is considered a covert tool doesn't mean it is used to hide illegal activity; it is perfectly valid that someone may want to hide legal activity from being monitored to protect it.

If on the other hand the perpetrator was not authorised to access the server, they could be held legally responsible for the unauthorised access and other computer offences under the Crimes Act 1958 (Victoria) or the Crimes (Computers) Act 1987 (Victoria). Also, if personal data was held on the server they could also be held liable under the Information Privacy Act 2000 (Victoria).

Company Policy

If the perpetrator was a company employee then they may also be in breach of its "Use of Electronic Communication Media Policy" which states that all tools must be used for business purposes only.

If the perpetrator was not authorised to access the server then they could also be liable under the clause that states, "Unauthorised electronic snooping including, but not limited to, network probing or cracking, by any User is prohibited". Also, depending on what type of information was stored on the server they may also be in breach of the company's Privacy policy.

Interview Questions

In this type of scenario I don't want the person to know why he is being interviewed so I will imply that the discussion we are having is a technical strategy meeting focused on improving the infrastructure architecture. I start with my first question:

I've got a paper that covers ICMP traffic, also commonly known as ping traffic. Do know what ICMP traffic is and how it works?

We were thinking of blocking ICMP/ping traffic on the firewalls. Can you see any problems with us doing this?

Do you think blocking this traffic would improve our security and if so how? (This is to see if he mentions the program and/or how Loki can bypass firewall rules by using ping traffic, which is currently allowed through).

Someone mentioned this Loki program to me, and I was told you know about this program. What does it do and how does it work?

Even if he says he doesn't know this program in the previous question, I briefly explain what the program is and then ask for his professional advice on this program and in particular - If we don't block ping traffic do you think it would be a useful tool for our administrators to have as a remote access tool? (This is to see if he comes up with extra information that he shouldn't know if he was unfamiliar with the program – this might prove that he has not been telling the truth).

What other uses do you think the tool may have?

Do you think anyone in the company is currently using it for this purpose/s?

Logs from our IDS show a lot a non-standard ICMP traffic between your system and our server. What do you think is causing this?

Additional Information

Readers seeking additional information may wish to familiarise themselves with ICMP. The RFC can be found at http://www.ietf.org/rfc/rfc792.txt .

The Incident Response book by Kevin Mandia and Chris Prosise as listed in the references on the last page of this paper, is a great book on forensics and shows excellent examples of ICMP traffic monitoring and how Loki traffic differs from normal ping traffic.

Also, Wietse Venema at www.fish.com has written a good hand out on examining an unknown binary called "The Source that came in from the cold".

PART 2 - Option 1 - Perform Forensic Analysis on a suspected compromised system

Background Case Information

On the 9th April, I decide to set up a Windows XP desktop and connect it directly to the Internet via my cable modem, to see if anyone can compromise it. The system that will be used is one that is normally used for testing purposes and hence has removable drive bays installed which makes swapping hard drive easy so I have a spare 20GB hard drive that will be used for this purpose. Further information about the specifications of this system can be found under the Victim System Hardware section.

The drive is sterilised by writing zeros to the entire hard drive, as per Part One, prior to installing the operating system so that during the forensic process no data from previous installs is picked up. The drive is connected as a primary and the Windows XP CD is bootable which makes the install straightforward. I select all of the install options making extra sure that IIS is not one of the options. I don't want IIS on the system as it would add a second level of risk and I only want to test Windows XP. No IDS will be installed because I want the system to represent a basic user's set up as much as possible and in almost all cases where I am asked to do a forensic analysis of a system, this software is not installed. A tool that will be used during the response is IRCR. IRCR stands for Incident Response Collection Report, which is produced by John McLeod. It is the Windows version of Unix's The Coroner's Toolkit by Dan Farmer and Wietse Venema. I have never used this program on Windows XP and I want to make sure that it works correctly, as it is great for automating the data collection process. It's tested and works fine. The program is left on the system but the report directory that I had it create on the local hard drive is deleted. It doesn't apply in this case but normally the report is not saved to the local hard drive as it has the potential to destroy evidence.

The only other setting changed is the security audit functions, as I believe that this is a must for any system that connects to the Internet. It's enabled allowing successful and unsuccessful logon attempts to be recorded in the security log file. The specific settings are as follows:

Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit logon events	Success, Failure
Audit privilege use	Success, Failure

One user account is created called "Me" that automatically logs on to the PC and is a member of the local administrators group. This is the only user administration that is done.

Suspected compromise

The system is now ready to go and I connect it to the Internet intermittently as the opportunity arises. On the 24 th April at approximately 4.50pm I connect it up and decide to leave it running over night. When I get up the following morning I notice that the cable modems receive and transmit lights are flashing rather fast and this is unusual when the system is not being used. Something is not as it should be so I decide that now is the time to consider that it has been "potentially" compromised. I can't look at the machine now and I don't want to keep the system going because no Anti -virus software is installed. If it is infected, it could spread it any further. Another problem with leaving it running is that it gives an intruder time to clean it up to ensure no trace of their visit is left behind. I decide t hat I can do without gathering "on - line" live data such as open ports, current network and memory data. I do a hard power down by switching it off at the power source so that I can do my investigation on it later.

Normally, when I respond to a suspected compromise the very first thing I do is check the Anti-virus software to see if it's installed, up to date, and working correctly. I have found that a lot of virus activity is mistaken for an intrusion and following this methodology has save me a lot of t ime in the past. In this case, it may very well be virus activity that I was seeing and not an intruder or it may just be some other type of innocuous activity. I will need to determine this during my investigation.

Victim System Hardware

I'm now ready to start my investigation so I commence cataloguing the hardware of the victim machine. I treat this as I would any other investigation and complete an evidence tag (a sample is attached in Appendix A) with the system details on it and I take a photocopy of the hard drive, sign it and attach it to Tag # 1.

The specific hardware details are:

Case # 26

Tag # 1

- Seagate U Series 5 hard drive, Model ST320413a, 16,383 Cyl, 16 HDS, 63 Sect, LBA 39,102,336 20 Gbytes, Serial no. 5ED0R038

Tag #2

- Generic brand Tower Desktop;
- Gigabyte PIII motherboard model no.GA -BX200,with no obvious serial number;
- Intel PIII 450 processor;
- 128 MB RAM;
- 1 x MSI IDE CDROM;
- 1 x Panasonic SCSI CD burner;
- 1 x I-Will PCI SCSI card;
- 1x 3.5" floppy drive;
- Geforce II MX400 AGP Video card;
- Sound blaster ISA sound card; and
- 1 x IBM 10/100 Etherjet PCI network card.

I also take digital camera photos where possible because the more evidence that is collected, the better the chances are of solidifying a case. These images are saved along with their MD5 values on my forensic workstation so that they can be included with all of the case files that will later be saved to CDROM.

I would also normally complete an Incident Report form (see Appendix B) and a Contact List (see App endix C). The Incident Report Form contains data such as the person conducting the investigation, who reported the incident, date, time, location as well as details, severity, sensitivity of the incident. I also like to include how widespread the knowledge of the incident is ie. is it public knowledge, known within most of the organisation or is it limited to particular staff only. The Contact List, as the name implies, contains all of the names and contact details of people who are directly involved in t he investigation.

Forensic Workstation Hardware

My forensic workstation for this investigation is as follows:

- IBM Personal Computer 300PL PIII Type 6872 -N2A, S/N 90-3PNLH;
- Company Managed Asset number AAA82695;
- On board S3 Trio video card;
- On-board Lan;
- Onboard sound;
- LG CD reader connected as secondary device to primary controller;
- Primay hard drive is - include all same stuff as evidence tag;
- 2 x USB connections;
- 3.5" inch floppy drive; and
- 128MB RAM

I have just purchased a new 80 GB hard drive for this machine and before installing the operating system on it I sterilise the hard drive using the same

process that I used when setting up my "honeypot". I connect the drive and make sure the system is not connected to the network and start installing the operating system.

Forensic Workstation Software

I am using Red Hat 7.3 with most options installed. No server components such as FTP, DNS, News etc. are installed. The forensic tools I use are:

> The Coroners Toolkit V1.11 (TCT) by Dan Farmer & Wietse Venema;
> Autopsy Forensic Browser V1.71 by Brian Carrier;
> The Sleuth Kit V1.61 by Brian Carrier (prevously called TASK);
> Mac-robber V1.0 by Brian Carrier; and
> Mac_daddy from www.incident -response.org

The Coroners toolkit is used to gather data from the image. I use this because the tools that are used i.e. grave -robber, ils, mactime, icat, pcat, unrm and extra ils2mac are very effective in grabbing data required for a forensic analysis.

Autopsy is a browser based forensic tool. It automates a lot of work for me instead of having to run the command line functions from The Coroners Toolkit. It also has a nice graphical user interface that allows for point and click actions. It uses The Sleuthkit as a backend to run all of it's commands such as dcalc, dls, ffind, fls, hfind, ifind, istat, sorter, dcat, dstat, file fsstat, icat, ils, mactime, sha1 md5. These tools all help gather relevant file, system and set up information and when used together assist in providing a picture of what has occurred.

Mac-robber is a stand-alone file. It does Modified, Accessed and Change (MAC) times on files just like grave -robber but I use this instead because it is faster. Grave-robber is written in Perl and mac -robber is written in C.

Mac_daddy is based on The Coroners Toolkit and is designed to run on floppy disk so that TCT doesn't need to be installed. I use it for the Perl script mac_daddy.pl which correlates the data better than using mactime on it's own.

To ensure that none of these tools alter any of the data whilst they are running I ensure that whenever the image is mounted that it is mounted in read -only mode. This is imperative in ensuring the data integrity.

Before I connect the evidence disk I make sure that the forensic system is in a secure area so only authorised staff can access it. I connect the evidence drive on to the secondary IDE controller to ensure the system won't boot the drive by mistake thereby corrupting the evidence, double check that it's in stand alone mode off the network and boot the forensic workstation. My first task is to check the date and time to ensure the integrity of the time stamps that I will be obtaining during the investigation. This is set correctly so I can now start.

Imaging the evidence

The first command I run is "fdisk –l", and I can see that my evidence disk is /dev/hdc1 and it only has one partition.

```
      Disk /dev/hdc: 20.0 GB, 20020396032 bytes
      255 heads, 63 sectors/track, 2434 cylinders
      Units = cylinders of 16065 * 512 = 8225280 bytes

      Device Boot    Start       End    Blocks   Id  System
      /dev/hdc1   *         1      2434  19551073+   7
HPFS/NTFS

      Disk /dev/hda: 80.0 GB, 80026361856 bytes
      255 heads, 63 sectors/track, 9729 cylinders
      Units = cylinders of 16065 * 512 = 8225280 bytes

         Device Boot    Start       End    Blocks   Id  System
      /dev/hda1   *         1        13    104391   83  Linux
      /dev/hda2            14      9697  77786730   83  Linux
/dev/hda3          9698      9729    257040   82  Linux swap
```

Before doing anything else I want to make sure that if the drive is altered I will know. I do this by running md5sum to obtain the algorithm.

```
[root@localhost root]# md5sum /dev/hdc1
079a4ab1cf137a3c52a3f31b1939ef05  /dev/hdc1
```

I now have all the drive information that I need that will allow me to proceed.

The image I am going to obtain will be a bit copy of the drive rather than just the partition so I run dd. DD is used to duplicate data. It not only duplicates files and directories but it also duplicates "free" space. This is very importa nt because when a hard drive has had files deleted, these files still exist, as the files are not over written; it is just the pointer to the file that is deleted so the data is still there. Even a formatted drive still has data on it which means that the data can still be recovered.

The image has completed successfully as the "in" and "out" records match.

```
[root@localhost root]# dd if=/dev/hdc1 of=/xp.img
39102146+0 records in
39102146+0 records out
```

After it completes I run md5sum on the image to ma ke sure that it matches the algorithm on the original. This is another checkpoint used in ensuring that

I have accurate data. The md5 value is the same so I know that the copy has been successful.

```
[root@localhost root]# md5sum /xp.img
079a4ab1cf137a3c52a3f31b1939ef05  /xp.img
```

Now I know that I have a valid image that has been saved to my forensic system that has maintained it's forensic integrity.

I remove the hard drive and label it with an evidence tag and hand this over to the person who will be storing it in a secure location. We have completed and signed the evidence tag, which ensures that the chain of custody is maintained. I also hand over the rest of the system and follow the same hand over procedure.

Media Analysis & Timeline

The next step is to mount the image so that I can view the file structure. I mount it using the following command:

```
mount –ro, loop,nodev,noexec,noatime  /xp.img /mnt/images
```

The arguments of this command mean:

ro - Mount the file system read -only;
loop – Mount it as a loopback device;
nodev - Do not interpret character or block special devices on the file system;
noexec - Do not allow execution of any binaries on the mounted file system; and

noatime – Do not update access time when reading the file system

It doesn't mount and the error message tells me that the kernel doesn't support NTFS. This is good because I now know that the file system is NTFS. Red Hat doesn't support NTFS so I download the RPM package from http://linux-ntfs.sourceforge.net that will enable NTFS support. I install it and check it is working correctly as per the web site's instructions and then try mounting the image again. This time it is successful.

I can now browse through the mounted directory and see the file and directory structure. Whilst I am browsing through the mounted image I am looking at what directories are present. The normal Windows XP file structure is present. It has Documents and Settings, Program Files, Windows, as well as the IRCR directory. There is nothing unexpected.

Autopsy is then started and I set up my case details as per the program's requirement and it loads the image.

Any investigation should start with determining what has occurred and when. This is called creating a timeline. Autopsy does this by first creating a data file (body file) by running "fls –r –m and ils – m" on the image. It also has options to check allocated files, unallocated files and unallocated meta data. I want all of these and ensure that these items are ticked. When running the next option I don't specify any start or end dates because I want a timeline created on all dates available. The machine has only recently been built so all dates are relevant.

I need to make sure that I have configured Autopsy with the correct time zone because if I haven't then the output of the timeline will not match real -time. To accomplish this I run mac_daddy against where the image is mounted using the following command:

```
./mac_daddy.pl /mnt/image > mactimes.txt
```

Autopsy is set up correctly as the results of both files are the same allowing for the way that Linux handles dates/times on mounted NTFS drives.

From the timeline I can see when the machine was built, the 9th April, and I can also see when it's been connected to the Internet by the files that have been generated. These dates are 10th, 11th, 12th, 13th, 16th and 24th April. Going through the list of files and their last modified date shows nothing unusual.

There is very little activity showing in relation to date/timestamps; that is until the 24th April. The PC was started and connected to the Internet just before 5.00PM. Approximately an hour later I can see some file activity.

```
Apr 24 2003 19:13:08  1118720 .a.  -r-------- root     root
/mnt/image/WINDOWS/system32/msxml3.dll
                      44032 .a.  -r-------- root     root
/mnt/image/WINDOWS/system32/msxml3r.dll
```

On the 24th of April, at 19.13 a dll, msxml3.dll was run long after the PC was started which makes it stand out on it's own. I don't expect any dll to run when a system is meant to be idle. On checking what this file does I discovered that there is a known critical vulnerability with this file.

Microsoft Security Bulletin MS02-008

XMLHTTP Control Can Allow Access to Local Files

> A vulnerability results because an attacker could seek to exploit this flaw and specify a data source that is on the user's local system. The attacker could then use this to return information from the local system to the attacker's web site.

Executing this vulnerabilit y would give a user access to a file of their choosing. The attacker would need to know the exact path to the file but system files reside in default locations so they could then manipulate a known file which may then in turn, elevate their access and their privileges. There are no further file access times until approximately half an hour later at 20.40, where I can see msv1_0.dll has been accessed.

```
Apr 24 2003 20:40:06   108032 .a.  -r-------- root     root
/mnt/image/WINDOWS/system32/msv1_0.dll
                       133632 .a.  -r-------- root     root
/mnt/image/WINDOWS/system32/schannel.dll
```

This dll is not listed under the above vulnerability but I do know that this dll is used to authenticate a users log on. More information about the logon process can be found in the following article:

Ochoa, Hernán. "Modifiying Windows NT Logon Credentials." 25 April 2000. URL: http://www.securityfocus.com/guest/1512

Very interesting. I find more information that tells me that the program, HFNetChck, u ses the file also. This program is used to remotely scan a systems patch status. Considering that the system was meant to be idle I can't see why this file was run unless someone was doing a reconnaissance on the machine. I can only gather that someone has first tried to gain access to the system by exploiting a specific vulnerability and then scanned the system to see what's available. Normally, I would expect these two events to be reversed i.e. scan the system first then execute a vulnerability but it may just be a false indicator and that the perpetrator is scanning for vulnerabilities.

I look for more file changes around this time frame and can only see one other update, the default.log. Looking in to this log with a HEX editor doesn't show me anything in a readable format so I will need to load a compatible reader or read these files in native Windows mode.

Some of the other entries of interest are log files as follows:

Security.log
Secevent.evt
System.log
Nutser.dat.log
Software.log

I can also see two other files that have timestamps worth looking. They are:

Kmixer.sys; and
Gpt.ini.

There is nothing out of the ordinary with these files so I move on.

I can see the.log files but as mentioned previously I need to use a different viewer to look at the secevent.evt file and some of the other log files because they are not in a readable format. I decide that I will view these later when I restore the image to a hard drive.

Autopsy shows deleted files by highlighting them in red and marking them with a /r There are quite a few deleted files under windows \system32. They are DLL files and exe files. The written accessed and changed times are all set to zero as is the file length. I know that files are created as part of the install process of windows. These files are deleted when the process is completed and the files I am seeing show this characteristic so there's no need to restore these.

The c:\rpt directory is also showing as deleted, remember that this is the directory that was created by IRC R which I deleted earlier on. This may come in useful later I will recover this. Doing this with Autopsy is straight forward as you simply need to highlight the file that you want recovered and click on the export button and simply select the location where you want the file saved to.

There are lots of hidden files. These show up with a $ before the file name. Going through the list shows me all of the relevant system files such as the Master Fat Table. No unusual files appear to be present.

I also do a keyword search looking for word such as hack, password, pass, porn, and back. It finds some instances but nothing extraordinary, so I also do a search for any remaining log files. There are no extra log files in the list that I require I haven't already flagged for further investigation.

Restoring the Image

The problem with doing a Windows forensics on a non -Windows system is that Linux doesn't have any native viewers installed that are capable of viewing most of the event/log files such as the Sece vent.evt file that was highlighted during my timeline search. Now that I have my timeline and deleted files, the best thing to do is to restore the image on to a hard drive so I can do a complete operating system analysis. I've got all of the deleted files and timelines I need and will be working on a back -up image anyway so writing files to the hard drive and changing dates/times is irrelevant. I will be looking through event logs, system and registry information for clues on what has occurred.

Working for an IT company, I have no problems locating a drive of the same size. I sterilise it and then move the image on to the drive using dd again.

```
dd of=/evidence/xp.img if=/dev/hdc1
```

Windows XP is similar to Windows 2000 but there are a lot of differences. There are vulnerabilities that are specific to this operating system. I believe that it's a harder operating system for a hacker/intruder to gain control of but it's not impossible. From my initial probing it would appear that someone has already tried executing a known vulnerability so I need to find more evidence.

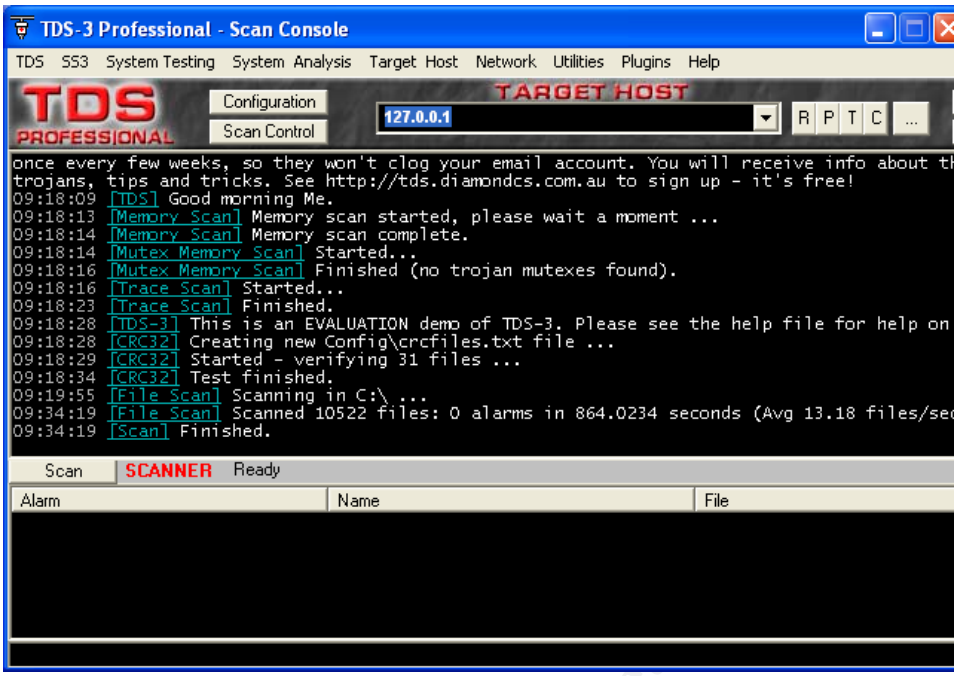
The very first thing I want to do is scan it for viruses. As previously mentioned, I almost always do that as part of my first response and it has certainly paid off. Often, virus/trojan/worm activity gets mistaken for hacker activity and a lot of time and effort can be saved at the start of an investigation by ruling this out. I do this and discover that the system is not infected.

Another item worth checking for which is a reasonably u nknown NTFS feature called, Alternate Data Streams (ADS). Alternate Data Streams are hidden files that are linked to normal files. The normal files can be clearly seen but the linked files can't. By looking at the normal files you would have no idea that they are linked so it is a great technique that can be used to hide files. These are almost impossible to pick up but I use a program called TDS -3 that I have found through testing, works quite well.

The option to turn this scanning on is found under t he "Scan Control" menu and you can set the Scan tasks to include the following options:

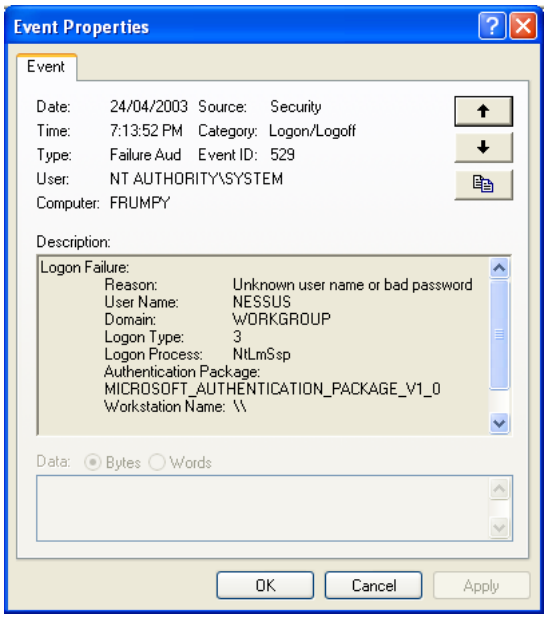
Scan NTFS ADS Hidden Streams; and
Show all NTFS ADS Streams.

When TDS completes its scan it tells me that no issues were found.



Event log evidence

Next, I check the event logs and notice nothing unusual in the Application log. The system log is interesting, as there are a lot of reports of an error on 24/4/2003 between 7.13 PM and 7.14 PM. The error is “The server received an incorrectly formatted request from \\. I’ve never seen this before and certainly I’ve never seen a PC with this name. I check the Security logs for around the same time frame and I can see a lot of Unknown user name or bad passwords from workstation name \\. The user name they are trying to use is Administrator. On first appearance, it looks like someone has been seeing if they can guess the password but did they get in. I can see 16 of these same attempts and the most interesting one is at 7.13.



It seems that someone has used Ness us. The time of this also correlates to the earlier event involving the mscml3.dll file that I found whilst going through the timeline. So the hacker was not trying to execute the vulnerability I found, they were using Nessus to scan for system vulnerabilities as I hinted at earlier. This program is a promoted as a vulnerability -checking program. It is used to gather information about what is on a system. Using Nessus is definitely a hacker reconnaissance technique, but did the person gain access? I look through the rest of the log looking for a "success audit" and can't find one from this system. I can only assume access was not gained but I still need to do more research and will get back to this one later, but for now I want to see what else I can find in the log.

Going through the rest of the security log it shows numerous "Failure Audit" entries. Most of these are trying to use the Administrator password. This is typical of recent virus behaviour that looks for Administrator passwords that are blank or easily guessable. I've already checked for virus infections and know I don't have any so I decide to ignore these entries and there are certainly a lot of them.

However, I cannot ignore entries from a couple other machines. The first one at 8.18.37pm from machine X-YWDCGU0V is trying to use an administrator account sa to get access. This doesn't work so they try administrator, amministratore, forsterkning, Verwalter, user, administrador, default, admin, guest, Administrateur, student, Invit0, uzivat el, test,root, and finally x. The accounts are mostly administrator accounts in different languages as well as some other administrator type of logins ie sa. The person runs it in succession a couple of times and going by the sequence it indicates that they are probably using an automated tool to do this rather than manually.

I then see MAISE1 with sa, administrator, amministratore, forsterkning, Verwalter, user, administrador, default, user, admin, guest, Administrateur,

student, Invitu, uzivatel, test, root, admin, MUSER, Extension. This is almost the same sequence used by one of the first PCs I've listed.

Again, a machine called FERN runs through the same sequence as MAISE1 and X-YWDCGU0V above, so this activity is either another virus trying to gain access or a script that is freely available from the Internet to "script kiddies".

I can also see a lot of attempts from a machine called DOKY3. At first it looks like virus activity because it is constantly trying the Administrator account but then I notice that after numerous attempts they switch to admin, root, test, Owner, Server, then back to Administrator.

I also find USER-93TESL try Administrator, admin, root, test, Owner, Server, then back to Administrator. Again, this is either a virus or the two machines are running the same tool.

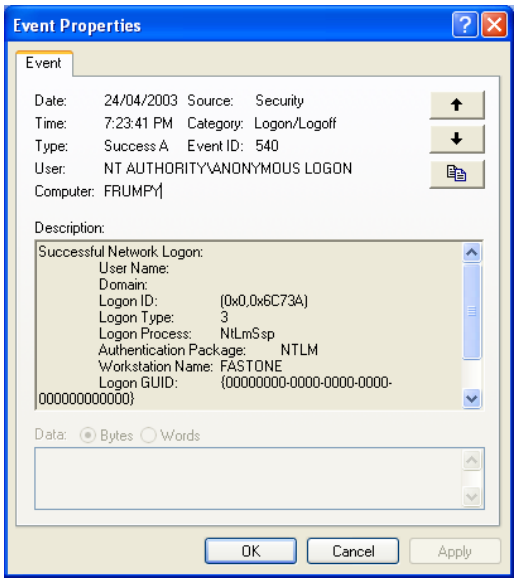
The next PC, FHSOT runs the same logon attempt using usernames Administrator, admin, root, test, Owner, Server, and back to administrator. This is definitely starting to look like virus activity. So I check the we b to see what could cause this behaviour but I am unable to find anything specific.

I see the same activity from PC1 and another unidentifiable machine i.e. it has characters that are not decipherable by my system so most likely in another language.

And that's it for the security event log. So I have one genuine attempt with Nessus and the rest seems to just be "noise". But it's interesting to note that apart from machine X-YWDCGU0V the "noise" activity started at 2.46am and finished at 3.35am so it last ed approximately 50 minutes. Could this be a classroom activity? It would certainly appear that way.

I attempt to ping the machines but the names don't resolve so I have no IP addresses which is as I expected, as they are NETBios names not DNS names. Without having IP addresses it would be almost impossible to trace. In this case there is no point because I am almost certain that none of these users gained access.

I can also see this is the event log:



This definitely shows an attack on the machine so I have two definite attacks one from \\ and one from fastone.

The next log entry at 8.40.06 pm is great because it even shows a user name, in first name middle initial and surname format. Notice that this also is the same time identified by my timeline analysis for when the file msv1_0.dll was accessed. The name in the event log is a fairly common English name so there is no guarantee that it is really the users name but nevertheless it is a great clue to follow if I ever needed more information on wh o was running the attack. In this case I only want to know what's been done to the system and not by whom so I will not pursue this.

Now, I need to find out if either machine gained access. Nessus may have given the first attacker enough information to compromise my system. I can't rely totally on the event logs because if access was gained, they may have altered logs. I could compare the log entries to the one I have restored earlier from IRCR but this snap shot was taken before the compromise and the on ly useful information I will be able to gather is by comparing the file information up to and including when the snapshot was taken. There's not much value in doing this though so I press on.

One of the advantages of using Nessus is that you can find out a lot of information about a system. It has a raft of plug -ins that it can use to find security weaknesses. Once the weaknesses have been identified it is simply a matter of finding the right tools on the Internet that exploit these vulnerabilities.

Next I run IRCR and when it's done I look at the report it's created. It indexes all of the information in a HTML file so I only need to click on the index to get to the items I need.

The main things I am going to check in here are file shares, shared resources, services. As I go through the reports there's nothing that seems out of the

ordinary. I'm beginning to believe that the system has not been compromised and that the attacker running Nessus was either only doing a reconnaissance or failed to gain access.

Having "detailed tracking" turned on also tells me if any processes were run and I can see that non were.

If this was a user machine I would check IE history but in this case it doesn't apply, so I will look at the registry. Regedtd32 is my preferred GUI. It is the 32-bit version and I use this rather than regedit. There are pros and cons for using either. Regedit has a search function but I know which keys I need to look at and don't need this function.

There are 5 hives and the first thing I w ant to look for is start up information. If a hacker has gained access to the system they will often want their tools to activate when the system is restarted so I want to check the start up files and in particular the "run" key to see if this has been modified. The relevant registry key is:

```
HKEY_LOCAL_MACHINE \SOFTWARE \Microsoft \Windows \CurrentVersion \Run
```

I also check the "runonce" key. A hacker will use this key to install packages and/or applications that they need on the system. These packages may either help them gain the root access that they need or may provide extra functionality to them. As the name implies it only needs to run once and then when it is installed they have it available for their use.

```
HKEY_LOCAL_MACHINE \SOFTWARE \Microsoft \Windows \CurrentVersion \RunOnce
```

The other key I want to check here is the uninstall information as not all programs delete the registry key when they uninstall.

HKEY_LOCAL_MACHINE \SOFTWARE \Microsoft \Windows \CurrentVersion \Uninstall

AddressBook
AdobeESD
Branding
Connection Manager
DirectAnimation
DirectDrawEx
Fontcor
ICW
IE40
IE40ata
IE5BAKEX
IEData
Microsoft Netshow Player
MobileOptionPack
Mplayer2
NetMeeting
OutlookExpress
PCHealth
SchedulingAgent

As can be seen there is nothing out of the ordinary here.

Next I look for the SID information. I do this in case a user account has been created and then deleted. A SID is a Security ID. Each user of the system has a unique access token, which is known as a SID. When a user account is created so is a SID and this information is stored in the registry. When a user account is deleted, more often than not this information remains in the registry.

HEKEY_LOCAL_MACHINE \SOFTWARE \Microsoft \WindowsNT \CurrentVersion \ProfileLis
t

S-1-5-18
S-1-5-19
S-1-5-20
S-1-5-21-606747145 -1520

I scroll through the list of four SIDs and they are all valid for the current user accounts. The only user accounts that enabled are Administrator, HelpAssistant and Me (the PC users account) so no new accounts have been created. This is also confirmed by checking under the Documents and Settings, which doesn't contain any new user folders. When a user logs on to a Windows XP system for the first time, a directory with their log on id will automatically be created under the C: \ drive under Documents and Settings.

An item of concern though is the Remote Desktop Users group that I find. This is a standard group created when installing Windows XP. A hacker could add an account here to enable them to log on remotely and they would have

access to the system in the same way as they would if they were sitting in front of the machine. Luckily, this group is empty.

The user account of most concern is the HelpAssistant this account is used to remotely troubleshoot machines. Most users have no need of this and it should be disabled. Given that I've found no evidence to the contrary I am confident that this account hasn't been use.

In case I missed anything I now do a search to picture files. There are numerous picture formats but I am limiting my search to jpg and bmp file extensions. I get a list of 235 files and I scan through all of them all and find nothing extraordinary. The good thing about XP is that it has a built in viewer so when I get the search results back they're readily viewable. If I were doing this on NT or Windows 2000 it would be much more difficult and would need to use a program that can search and view files at the same time such as ACDeeSee. The last file extensions I want to search for a MP3 files. These are music files and are commonly shared over the Internet. One of the aims of some hackers is to find free storage on other people's machines as it's much cheaper to use someone else's resources and music files are very popular as was shown by Napster. The search completes and comes back with no files found.

If a root kit was installed then there's a chance that it may be added as a service. As mentioned, using the "runonce" registry key can help install services during a system reboot. Services are checked using th e computer management console and all looks fine.

Another way used to get files to run is to use the windows scheduler. This program will run commands at specific dates and times as required. To do this I run the "at command. As can be seen below, nothing is scheduled to run. It doesn't mean that this hasn't been used, it simply shows me that nothing is currently set up.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985 -2001 Microsoft Corp.

C:\Documents and Settings \Me>at
There are no entries  in the list.
```

I've now done as much evidence gathering that I can. I collect all the electronic files and images that I have including the ones that I have generated and burn them to a CD. I create a new evidence tag for this as well as one for the hard copies (ie. paper based) files that are relevant and I hand them over to the evidence custodian who places these with the existing evidence that he has. My work is done, for now.

Conclusion

Computer investigation work is counted in days and weeks, not hours. The process is lengthy and time consuming. It requires a fastidious approach that can undergo the toughest scrutiny in a court of law. Not all cases end up in a court of law but the chances are you may not know this when conducting an initial investigation. What may start out as a simple IDS alert could turn in to something more if criminal evidence such as credit card fraud is located. The two incidents may be totally separate; you simply stumbled across an unexpected crime whilst looking for something else.

In this particular incident, the system has not been compromised. However, I did find evidence of two real attempts to compromise it. It seems unlikely that it would be two separate hackers but it's not out of the question. It is possible for a hacker to run more than one attack at the same time when trying to compromise a system. I was unable to find any other evidence of alteration or tampering to the system and there was no trace found of any backdoors or sniffers being installed.

The attack was short lived but I have no doubts that if the hacker persisted they would have eventually gained access. Tools and information on hacking is freely available on the Internet and a lot of these tools don't even require any expertise. I've heard of a lot of "wanna be" hackers locating and running these tools to see if they can easily gain access to systems. Quite often they only want to see what the tools can do and what is possible. Nevertheless this is still wrong but it is the people who w ant to do malicious damage or profit from compromising a system that are the greatest risk.

What really concerned me in this case, is the number of attempts people made at trying to log on to the system. Whether this was by viruses or "script kiddies" is irrelevant, because a compromise regardless of how it occurs is a problem.

Security is the responsibility of all users. Anti -virus software must be kept up to date and people should have at least some basic knowledge of how to secure their PCs. Often then not, ignorance is the greatest enemy. If users really thought about how information accessed on their PCs could be used against them I'm sure that more care would be taken. Social engineering is on the rise all the time.

The security of the computer systems that they own and use is not on most peoples mind. What's scarier still, are the system administrators that don't give it a second thought. I've heard people say that they don't care if someone gets access because they have nothing of value sto red. What they fail to understand is that they may unwittingly be participating in a crime by failing to secure their systems thereby allowing others to use to commit a crime. Is this a crime? This is where responsibility blurs. Maybe if more cases go to court making system administrators and companies legally liable for not sufficiently securing their network we might see a change in this attitude to security. Nevertheless, the hardest hurdle to overcome in relation to securing information is the belief by most people that "it could never happen to me" - it can, and it does, everyday around the world.

Part 3 – Legal Issues of Computer Incident Handling

Outline of Australian Legal Acts that govern computer incident handling

In Australia there are numerous Acts and pieces of legislation that cover Computer related offences. The legislation falls in to two categories:

Commonwealth (Federal); and
State.

In very simplified terms, Federal law tends to govern a crime that affects anything owned or control led by the Commonwealth. All other acts of crime fall under State jurisdiction.

The most relevant acts in relation to computer crime are as follows:

Commonwealth Acts

Crimes Act 1914;
Criminal Code Act 1995;
Cybercrime Act 2001;
Privacy Act 1988;
Telecommunications Act 1997;
Telecommunications (Interception) Act 1997; and
Interpretation Act 1901.

Victorian Acts

Crimes Act 1958;
Information Privacy Act 2000;
Crimes (Computers) Act 1987;
Surveillance Devices Act 1999;
Summary Offences Act 1966; and
Crimes (Property Damage and Computer Offences) Act 2003

In Australia, computer crime has been brought before the courts using traditional criminal offences such as those relating to loss or damage to property or financial loss. This is due to the fact that it is easier to try a case against precedence cases that have already paved the way.

The Model Criminal Code highlighted this discrepancy. It is a federal code that was designed to combine the existing Commonwealth and State Acts and is based on traditional crime elements but also specifically addresses computer crime. It discusses the merits and deficiencies in the existing legislation and presents recommendations to the States on what should be considered in future legislation. Although it is not an Act many States have formalised it's recommendations in to legislative Acts.

The main Commonwealth act relating to computer crime is the "Cybercrime Act 2001". This is "An Act to amend the law relating to computer offences, and for other purposes". Some of the acts that it amends are:

> Australian Security Intelligence Organisation Act 1979;
> Telecommunications (Interception) Act 1997;
> Crimes Act 1914; and
> Customs Act 1901.

Its main purpose is to include computer crime in to traditional criminal law it covers specific computer offences and handling of computer related information.

Some states also have a similar act, which works at State rather than Commonwealth level. One such act that has only recently been passed on the 6th May, 2003 in Victoria is the Crimes (Property Damage and Computer Offences) Act 2003. The act recognises the fact that traditional criminal law failed to adequately cover computer crime and hence was enacted to cover this gap as an amendment to the Crimes Act 1958 (Vic).

Incident Scenario

So how do these Acts govern Incident handling when you are an Internet Service provider? Below is the outline of a particular scenario:

You are the system administrator for an Internet Service Provider that provides Internet access to paying customers. You receive a telephone call from a law enforcement officer who informs you that an account on your 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.

In this scenario the identity of the law enforcement officer has been validated and I am assuming that the government computer is a Federal government computer and not a State government computer because different laws apply for each. There is no mention as to whether or not any other offence other than the break in occurred i.e. why the hacker broke in and whether or not they intended to do anything once access was gained.

Regardless of the hacker's intention I will broadly cover both Summary and Indictable offences.

An Indictable offence is a crime that is serious enough to warrant it being tried in a court of law with a jury present. A Summary offence is still presented in a court of law but is tried by a judge only i.e. no jury is present.

Under the Cybercrime Act 2001 there are two types of computer offences:

Serious computer offences; and
Other computer offences.

As listed in this act under Division 477 serious computer offences that are indictable offences are:

- Unauthorised access, modification or impairment with intent to commit a serious offence;
- Unauthorised, modification of data to cause impairment; and/or
- Unauthorised impairment of electronic communication.

Under Divison 478 – Other computer offences which are summary offences are:

- Unauthorised access to, or modification of, restricted data;
- Unauthorised impairment of data held on a computer disk etc;
- Possession or control of data with intent to commit a computer offence; and/or
- Producing, supplying or obtaining data with intent to commit a computer offence; and

Initial contact from Law Enforcement Officer

The powers governing conduct by Law enforcement officers is covered under Schedule 2 of the Cybercrime Act 2001, under the heading of: "Law enforcement powers relating to electronically stored data. Crimes Act 1914"

An ISP does not need to provide any information to a Law Enforcement officer without a court order in relation to a crime. This is as per section 3LA that states:

3LA Person with knowledge of a computer or computer system to assist access etc

(1) The executing officer may apply to a magistrate for an order requiring a specified person to provide any information or assistance that is reasonable and necessary to allow the officer to do one or more of the following:
  (a) access data held in, or accessible from, a computer that is on warrant premises;
  (b) copy the data to a data storage device;
  (c) convert the data into documentary form.

(2) The magistrate may grant the order if the magistrate is satisfied that:
  (a) there are reasonable ground s for suspecting that evidential material is held in, or is accessible from, the computer; and
  (b) the specified person is:
    (i) reasonably suspected of having committed the offence stated in the relevant warrant; or
    (ii) the owner or lessee of the computer; or
    (iii) an employee of the owner or lessee of the computer; and

(c) the specified person has relevant knowledge of:
  (i) the computer or a computer network of which the computer forms a part; or
  (ii) measures applied to protect data held in, or ac cessible from, the computer.

(3) A person commits an offence if the person fails to comply with the order.

Penalty: 6 months imprisonment.

This means that an ISP is under no obligation to provide the officer with a name, address of the suspect or any other details regarding the activity. This doesn't mean that the ISP will not provide the information. The ISP may have a code of conduct that supports a close interaction with Law Enforcement and they may voluntarily hand over information providing they do not breach any other laws such as the Privacy Act 1998.

Preserving Evidence

An ISP is under no obligation to preserve any evidence pending issue of a warrant. If an officer suspects that the evidence may be destroyed or damaged then they can apply for a warrant by telephone as per section 3R of the Crimes Act 1914:

3R Warrants by telephone or other electronic means

(1) A constable may make an application to an issuing officer for a warrant by telephone, telex, facsimile or other electronic means:
  (a) in an urgent case; or
  (b) if the delay that would occur if an application were made in person would frustrate the effective execution of the warrant.

However, as mentioned in the previous section, if an ISP has a close relationship with Law enforcement and has a methodology in place that supports this mutual co-operation then they may, as a matter or course, decide to ensure that all evidence is preserved pending issue of a warrant or court order.

Providing Log Information

Log information that is held by the ISP can only be released to a Law Enforcement officer under a warrant or court order as mentioned under the Initial contact section.

However, a warrant often states what can and cannot be seized. If it is unclear that the logs relate to the actual crime and may come outside of the scope of the warrant then the officer has the right to examine and seize it as per:

Cybercrime Act 2001 Amendment to Crimes Act 1914

Subsection 3K(2)

(2) A thing found at the premises may be moved to another place for examination or processing in order to determine whether it may be seized under a warrant if:

(a) both of the following apply:

(i) it is significantly more practicable to do so having regard to the timeliness and cost of examining or processing the thing at another place and the availability of expert assistance;

(ii) there are reasonable grounds to believe that the thing contains or constitutes evidential material; or

(b) the occupier of the premises consents in writing.

Investigative Conduct

To provide information to law enforcement or to preserve evidence, the ISP must know what information relates to the incident. To do this, generally an ISP will undertake its own investigative activities. However, the ISP also has guidelines and legal restraints that must be followed when conducting an investigation.

The only investigative activity that an ISP is allowed to undertake to monitor a users activity past/present and future is that which pertains to the ISP being able to ensure the integrity of the s ervices they are providing.

In particular the Telecommunications Act 1997 mentions that the service provider is governed by Industry standards and codes of conduct that ensures the following:

113 Examples of matters that may be dealt with by industry codes and industry standards

f) privacy and, in particular:

(i) the protection of personal information; and

(ii) the intrusive use of telecommunications by carriers or service providers; and

(iii) the monitoring or recording of communications

This is also enforced by the Privacy Act 1988, Division 2 Principle 1 as per below:

Principle 1

Manner and purpose of collection of personal information

1.Personal information shall not be collected by a collector for inclusion in a record or in a generally available publication unless:

(a) the information is collected for a purpose that is a lawful purpose directly related to a function or activity of the collector; and

(b) the collection of the information is necessary for or directly related to that purpose.

2.Personal information shall not be collected by a collector by unlawful or unfair means.

The ISP is not allowed to intercept data outside its operational guidelines and must also comply with the Telecommunications (Interception) Act 1997 which governs "wire tapping".

Unauthorised Access

So how would the ISPs actions change if the logs disclosed a hacker gained unauthorised access to their system at some point, and created an account for him/her to use, and used THAT account to hack into the govern ment system?

To enable me to answer this question I will assume that the hacker who gained unauthorised access did not work for the ISP and the crime was committed in Victoria.

The person has committed a crime against the ISP and as such they are allowed to gather and preserve data for it's own use as it relates directly to its ability to provide continuity of service.

The charge that could be made against the hacker is that of computer trespass. This is covered by the Summary Offences Act 1966, Divisio n 2, 9A:

9A. **Computer trespass**

A person must not gain access to, or enter, a computer system or part of a computer system without lawful authority to do so.

Penalty: 25 penalty units or imprisonment for 6 months.

However, depending on how it is viewed i.e. the ulterior motive of the hacker, it may also be an Indictable offence under the Cybercrime Act 2001, Division 477.1, unauthorised access, modification or impairment to commit a serious offence.

Also, under the Crimes Act 1958 (Vic) the hacker can be charged with:

199. Possessing anything with the intent to destroy or damage property.

A person who has anything in his custody or under his control -

With the purpose of using it, or causing or permitting another to use it, without lawful excuse –

To destroy or damage any property belonging to some other person or to himself, the user or both of them and some other person; or

> To destroy or damage any property in a way which he knows or believes is more likely than not to endanger the life of some other person; or
>
> With the purpose of using it, or causing or permitting another to use it, dishonestly and with a view to gain for himself or another, to destroy or damage property

Ultimately the ISP must ensure that they have guidelines and policies in place that governs the way they handle computer incidents. Given the nature of the data that they handle they not only need to control the way they handle their own information but they need to make sure that they handle customer data in a responsible and legally val id manner.

Appendix A – Evidence Tag

Case No:	Tag No:	Custodian:
Date:	Type of Item:	
Time:		
Detailed Description of Item.		

Date/Time: From Location:	Person receiving evidence (name): Reason:	Signature: To Location:
Date/Time: From Location:	Person receiving evidence (name): Reason:	Signature: To Location:
Date/Time: From Location:	Person receiving evidence (name): Reason:	Signature: To Location:
Date/Time: From Location:	Person receiving evidence (name): Reason:	Signature: To Location:
Date/Time: From Location:	Person receiving evidence (name): Reason:	Signature: To Location:
Date/Time: From Location:	Person receiving evidence (name): Reason:	Signature: To Location:

Appendix B – Incident Report Form

IT Security incident report form

IT SECURITY CONTACT FOR INCIDENT	
Name:	
Phone no	(B)
	(M)

PERSON WHO REPORTED INCIDENT	
Name:	
Business Unit and Department:	
Title:	
Phone no	(B) (F)
	(M)
Address:	
City:	
State:	
Country:	
Email address:	

DATE/TIME OF INCIDEN T DISCOVERED

LOCATION OF INCIDENT

Address:
City:
State:
Country:

TYPE OF INCIDENT

☐ Virus or other malicious code
☐ Intrusion/Probe/Scan
☐ Inappropriate use of system
☐ Website defaced
☐ Denial of Service
☐ Unauthorised Electronic monitoring
☐ Other (specify)……

DETAILS OF INCIDENT

SEVERITY OF INCIDENT

☐ High ☐ Medium ☐ Low ☐ Unknown

SENSITIVITY OF INCIDENT

☐ High ☐ Medium ☐ Low ☐ Unknown

HOW WIDESPREAD IS KNOWLEDGE OF THE INCIDENT

☐ Public Knowledge
☐ Restricted to company employees
☐ Restricted to reporter and Incident Response Team

INCIDENT TEAM COMPOSITION

Names

Appendix C – Incident Contact List

CONTACT LIST – CASE NO.

Name	Dept/Company	Title	Address	City	State	Country	Mobile Ph.	Business Ph

References

AUSTLII Australasian Legal Information Institute
URL: http://www.austlii.edu.au/

Cole, Eric, Hackers Beware. Reading: New Riders, 2002

Fung, James. "Part II: Identifying a Mystery Binary" GCFA Practical Assignment Version 1.0

LeBlanc, David. "Detecting Alternate Data Streams".30 November 2000
URL:
http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=16189&pg=1&show=530

Mandia, Kevin & Prosise, Chris. Incident Response Investigating Computer Crime. Reading: Osborne,/McGraw -Hill, 2001

Microsoft Technet. " XMLHTTP Control Can Allow Access to Local Files" 21 February 2002
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-008.asp

Ochoa, Hernán. "Modifiying Windows NT Logon Credentials." 25 Apri l 2000.
URL: http://www.securityfocus.com/guest/1512

The Open Group. "System Interface & Headers Issue 5 Reference Pages".
The Single UNIX® Specification, Version 2. 1997
URL http://www.opengroup.org/onlinepubs/7908799/xshix.html

Owen, Greg. "Analysis of Unknown Binary". Greg Owen GCFA. September 2002

Alhambra. "Project Loki: ICMP Tunnelling" 11 Aug 1996
URL: http://www.phrack.org/show.php?p=49&a=6

Venema, Wietse. "The Source that came in from the cold"
URL http://www.fish.com/forensics/programs.pdf

Wheeler, David A. "Program Library HOWTO."
URL http://www.dwheeler.com/program -library/

The legal reference I have on my desk ……

Last Updated: February 24th, 2018

Upcoming SANS Forensics Training

SANS London March 2018	London, United Kingdom	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Paris March 2018	Paris, France	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CA	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Secure Singapore 2018	Singapore, Singapore	Mar 12, 2018 - Mar 24, 2018	Live Event
Mentor Session - FOR500	Minneapolis, MN	Mar 13, 2018 - May 01, 2018	Mentor
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, Germany	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TX	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
Mentor Session - FOR610	Milwaukee, WI	Mar 21, 2018 - May 02, 2018	Mentor
SANS Boston Spring 2018	Boston, MA	Mar 25, 2018 - Mar 30, 2018	Live Event
Community SANS Columbia FOR610	Columbia, MD	Mar 26, 2018 - Mar 31, 2018	Community SANS
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, United Arab Emirates	Apr 07, 2018 - Apr 12, 2018	Live Event
Community SANS Virginia Beach FOR508 @ SLAIT	Virginia Beach, VA	Apr 09, 2018 - Apr 14, 2018	Community SANS
SANS vLive - FOR578: Cyber Threat Intelligence	FOR578 - 201804,	Apr 10, 2018 - May 17, 2018	vLive
SANS Zurich 2018	Zurich, Switzerland	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS vLive - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	FOR508 - 201804,	Apr 23, 2018 - May 30, 2018	vLive
SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
SANS Riyadh April 2018	Riyadh, Saudi Arabia	Apr 28, 2018 - May 03, 2018	Live Event
Automotive Cybersecurity Summit & Training 2018	Chicago, IL	May 01, 2018 - May 08, 2018	Live Event
SANS vLive - FOR500: Windows Forensic Analysis	FOR500 - 201805,	May 08, 2018 - Jun 14, 2018	vLive
Security West 2018 - FOR578: Cyber Threat Intelligence	San Diego, CA	May 11, 2018 - May 15, 2018	vLive
Security West 2018 - FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR500: Windows Forensic Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
Security West 2018 - FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR572: Advanced Network Forensics and Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive
Security West 2018 - FOR518: Mac Forensic Analysis	San Diego, CA	May 11, 2018 - May 16, 2018	vLive