Global Information Assurance Certification Paper

GIAC Certified UNIX Security Administrator (GCUX)

Practical Assignment Version 2.0

Online

Submitted by: Tan Koon Yaw

Date: 2 June 2004

Table of Content

Responding and Investigating a Unix Incident with Risk Analysis and Steps to Secure the System

Abstract

Initial response is the stage of preliminary information gathering to determine the probable causes and the next appropriate response to a security incident. Unprepared and improper handling of incidents may result in loss of essential evidence and potentially impede future investigation and even jeopardize the case. More often than not, further investigation into the cause of the incident is required. At times, this may even lead to a legal proceeding. Beside, being ready and equipped to handle such attacks, the organization should also look into performing regular risk analysis and take precautionary steps to secure their systems.

This paper thus aims to discuss the initial response to a security incident and investigation process, in specific to a Unix platform. Risk analysis and steps on how to secure the system to prevent it from future compromise will be discussed as well. The discussions will be divided into three main parts.

In the first part, we discuss the initial response on a Unix platform and what evidence should be gathered and how they should be collected. To carry out the initial response successfully, the responder needs to prepare a set of tools to gather the evidence. We will list out some of the essential tools that a responder should be equipped with.

In the second part, we discuss the steps and tools to carry out the forensic investigation on a Unix system.

Finally, in the third part, we provide a risk analysis on a case study and demonstrate the basic steps in securing the system.

Part One: Initial Response to Unix System

A-1. Introduction

In many cases when a system encounters an incident, the system administrator will jump straight into the system to find out the cause and if possible bring the system back to normal as soon as possible. Such knee-jerk reactions are common, especially for systems supporting critical business operations.

However, in the course of checking and recovering the system, they could unknowingly tamper the evidence and even lead to a lost of information causing potential implications for future investigations. Complication will arise if the recourse actions involve legal proceedings.

Not every incident will lead to a full investigation or legal proceeding. In most cases, the systems cannot afford the downtime to carry out a full investigation to find out the most possible cause behind the incident. However, in the event when a security breach has taken place, proper handling of the system is necessary. Hence it is very important to establish a set of proper and systematic procedures to preserve all evidence during this critical initial response stage. Without proper preparation, it is likely that the evidence may be corrupted.

Initial response is the stage of preliminary information gathering to determine the probable causes and the next appropriate response. Responders should be equipped with the right knowledge on how and what information to collect without disrupting the services. During the initial response, it is also critical to capture the volatile evidence on the live system before they are lost.

In the first part of this paper, we will discuss the initial response on the Unix platform. We will cover what evidence should be gathered and how they should be collected on a Unix system. In order to carry out the initial response successfully, the responder needs to prepare a set of tools to gather the evidence. This paper will also list out some of the essential tools that the responder should be equipped with.

A-2. Initial Response

Initial response is the stage where preliminary information is gathered to determine whether there is any breach of security, and if so, to determine the possible breach and assess the potential impact. This will allow one to determine the next course of action, be it to let the system continues its operation or arrange for immediate isolation for a full investigation.

There should be a well-documented policy and guidelines on how different types of incidents should be handled. It is also important to understand the policies and response posture. The level of success to solve an incident does not depend only on the ability to uncover evidence from the system but also

the ability to follow proper methodology during the incident response and evidence gathering stage.

When users report an incident, all they know is something is not right. Usually they are not sure what is wrong and all the user can tell is that the system had behaved abnormally. Thus, it is up to the responder to ask the right questions to uncover the necessary information.

During the initial response stage, the key questions that the responder should ask are (Who, What, When, Where, How):

- Who found the incident?
- How was the incident discovered?
- When did the incident occur?
- What was the level of damage?
- Where was the attack initiated?
- What techniques were being used to compromise the system?

Many times, when the administrators know that their system has been intruded and compromised, they will usually want to get the system back to normal as soon as possible to avoid much inconvenience to their operations. Hence this will mean that the system will be brought down and subsequently either cleaned up or reinstalled.

The common key questions are whether we should allow the system to disconnect from the network or just leave the system in the original state? If so, can the system be powered off? If the system is brought offline immediately, important evidence could be lost and may even alert the intruder that you now aware of his/her presence. However, one needs to balance that with the consequences of allowing the system to continue to run in its original state. There could be more potential damage to the system especially when one is not aware what the intruder could be doing to your system

Another question is whether one should report the incident to the management immediately or wait until there is a better understanding of the cause? You may not want to create unnecessary alarm in case it turns out to be a false alarm. However, by not reporting the incident immediately, it may also lead you into more serious trouble if it is indeed a compromised system and evidence has been lost or corrupted.

Some of the tell-tail signs of a potentially compromised system are:

- Several abnormal outgoing connections
- Mysterious files found on the system
- Cannot connect to the system for unexplained reason
- High increase of disk usage for unexplained reason
- Mysterious connections
- High usage of system processing for explained reason
- Unauthorized connection

If you encounter any of the above, what should you do immediately? Sit there and cry? Of course not, but very often people are at a lost of what are the correct steps to take when faced with an immediate incident. Thus, proper preparation and how to carry out the initial response is very important. This paper cannot emphasize enough how important it is to carry out the right steps during the initial response stage.

The initial response procedures should be easy to remember and simple to carry out. There are a lot of things to do in a short period of time when an incident occurs such as who to report the incident to, and the roles and responsibilities.

A-3. Preparation for Evidence Gathering

The basic principles to keep in mind when gathering evidence is to perform as little operations on the system as possible and maintain a detailed documentation on every single step performed on the system.

Maintaining a chain of custody is important. Chain of custody establishes a record of who handled the evidence, how the evidence is handled and the integrity of how the evidence is maintained.

When you begin to collect the evidence, record what you have done and the general findings in a forensic notebook together with the respective date and time. Use a tape recorder if necessary. Note that the system that you are working on could be rootkitted.

Avoid performing the following processes on the system:

- Writing to the original media
- Killing any processes
- Meddling the timestamp
- Using untrusted tools
- Meddling the system (reboot, patch, update, reconfigure the system).

A-4. Protecting the Volatile Information

The main objective of the initial response is to gather the volatile information before they are lost when the system is shutdown for further forensic investigation. Volatile information such as running processes, network connection and memory content can be important information that may make or break the case. In some cases, hackers may have tools running in the memory. Gathering such evidence is therefore necessary as part of the initial response procedure. It is therefore essential to capture the volatile information on the live system before they are lost.

The order of volatility is as follows:

- Registers, cache contents
- Memory contents
- State of network connections
- State of running processes
- Contents of file system and hard drives
- Contents of removable and backup media

Information from the first four items can be lost or modified when the system is shutdown or rebooted.

In additional, logs, configuration files, and system files are also useful evidence to check whether an incident has occurred.

Note that Unix allows you to delete a program after it has been executed (running as a process). When the system is powered down, the process will be gone as well. Strictly speaking, the program file is not truly deleted. Rather it is marked for deletion and has been unlinked. It will not be shown on the directory listing. It will only be deleted when the process has terminated or the system is being shutdown. In this case, during the initial response, such volatile evidence should be captured as well. For further details on how Unix deletes a file, you may like to refer to Incident Response: Investigating Computer Crime [1].

A-5. Creating a Response Toolkit

Preserving evidence and ensuring the integrity of the evidence is very important. Therefore it is essential to ensure the programs and tools used to collect the evidence are trusted. The responder should also be equipped with the necessary programs at all time. It is thus useful for the responder to compile all the necessary tools, for example in a convenient media such as the CD-ROM and have it ready when responding to incidents. This will shorten the response time and enable a more successful initial response effort.

Below is a list of tools that you should minimally be equipped with when responding to a Unix system. There could be more depending how much you wish to carry out prior to bit-level imaging of the media. The most important thing is to harvest the volatile information first. Those residing on the media could still be retrieved during the forensic analysis on the media image.

There are many different flavors of Unix. Some of the commands/path may be different. We will therefore use one platform to illustrate how it is being done. The concept demonstrated here can be used to create the response toolkit for other Unix flavors.

You should not use the commands on the victim system. If the system is root-kit, the results from the system command will not be trusted. You will therefore

need to create a trusted toolkit consisting all the trusted binaries and libraries to retrieve the information.

To avoid dependency on the libraries on the compromised system, the trusted tools should be statically complied where possible.

Usually, most of the tools/commands use shared libraries. When creating the trusted tools, you should therefore compile them without using the shared libraries. If this is not possible, the shared libraries should be included as part of your trusted toolkit and set the appropriate environment to use the trusted libraries when running your toolkit. To determine the shared libraries, use command ldd to check on the shared libraries. For example, the command `ldd /bin/ps` will show the shared libraries used by `/bin/ps`.

The linker program for Redhat 8.0 is ld-linux.so.2 [2]. This linker program checks the versions and other services and is hard coded into programs that use it. As such, for this file, you will need to rely on the copy on the victim system. To ensure it is not tampered, you can compute its MD5 and compare it with a trusted version.

Hal Pomeranz has posted a document on compiling statically-linked binaries for Solaris [3].

Set the victim environment (usually LD_LIBRARY_PATH) to use the trusted shared libraries when running your toolkit.

Below is a set of commands/tools that you would need to prepare as part of your toolkit.

Tools/Commands	Description
arp	Check the system arp cache
bash	Bash shell.
cat	Concatenate files and print on the standard output.
date	Display the system date and time.
dd	Convert and copy a file.
df	Display filesystem disk space usage. E.g. df -ah
echo	Display a line of text.
env	Display the current environment.
file	Display file type.
find	Search for files in a directory hierarchy.
finger –ls	User information lookup program.
grep	Display lines matching a pattern.
ifconfig	Display and configure a network interface.
last –aidx	Show listing of last logged in users.

lastlog	Show the last login.
ldd	Display shared library dependencies.
less	File perusal filter. Similar to more but allows backward movement in the file as well as forward movement.
ls	List directory contents. E.g. ls –alR /proc
lsmod	List loaded modules.
lsof	List open files. E.g lsof –d rtd lsof +m –I lsof +L1
md5sum	Compute and check MD5 message digest. This can be used to prove that the evidence remains intact and is not tampered with.
modinfo	Display information about a kernel module
more	File perusal filter.
mount	Mount a file system. E.g. mount /dev/fd0 /mnt/floppy mount /dev/cdrom /mnt/cdrom
nc cryptcat	Netcat. A utility that reads and writes data across network connections. Cryptcat is an equivalent version of netcat but create an encrypted channel of communication.
netstat	Print network connections, routing tables, interface statistic, masquerade connections and multicast memberships. E.g. netstat –anp
ps	Report process status. E.g. ps –auxww
route	Show and manipulate the IP routing table. E.g. route –Cn
script	Make typescript of terminal session.
strace truss (solaris)	Trace system calls and signals.
strings	Display the strings of printable characters in files. Control-D to exit.
tcpdump	Dump traffic on a network.
top top –b –n1	Display top CPU processes.
umount	Unmount file system.
uname	Print system information.
uptime	Display how long the system has been running.

vi	A text editor.
vmstat	Display virtual memory statistic. Helps to know whether is there any abnormalities.
w	Show who is logged on and what are they doing.
who who –Hu	Display who is logged on.

A-6. Gathering the Evidence

Rule of thumb: Limit any operations that will access or modify the file systems as much as possible prior to the disk imaging process.

Step One: Open a Trusted Command Shell

Log on locally at the victim console with root privileges. Mount your trusted toolkit:

```
(victim)# mount /dev/cdrom /mnt/cdrom
```

You can now open a trusted shell from your mounted toolkit.

To ensure all commands are executed from your trusted toolkit, set the PATH environment appropriately to the mounted trusted toolkit. Change env PATH to dot (.). However, this will change the system timestamp.

To ensure that the linker program will get the trusted shared libraries in the trusted toolkit instead from the system, set the LD_LIBRARY_PATH to point to the trusted toolkit. For example, if we have all the trusted tools cut into a CD-ROM with all the shared libraries in the lib directory, we could set the environment to:

```
(victim)# PATH="/mnt/cdrom/bin"
(victim)# LD_LIBRARY_PATH="/mnt/cdrom/lib"
(victim)# export PATH
(victim)# export LD_LIBRARY_PATH
```

Before you set the new PATH and LD_LIBRARY_PATH, record the PATH and LD_LIBRARY_PATH first.

Check that the environment is correct.

```
(victim)# echo $PATH
(victim)# echo $LD_LIBRARY_PATH
```

Step Two: Prepare the Collection System

Remember that you should not write the evidence collected to the original media. A simple way is to write the data to a floppy disk. However, some of the evidence collected may exceed the disk space of the floppy disk. One

simple way is to pipe the data over the network to your responder's system. To do this, we could use the popularly known "TCP/IP Swiss Army Knife" tool, netcat, to perform the job.

The process of setting up the netcat is to first set up the netcat listener on the responder's system.

```
# nc -l -p 55555 >> evidence.txt
```

The above command opens a listening port on the responder's system and redirect anything received to evidence.txt. The switch `-l` indicates listening mode. The listener will close the socket when it receives data. To allow the listener to continue to listen harder after the first data is captured, use the `-L` switch instead. Thus, you can choose whether to create a new file for each command or appending all evidence gathered into one single file by using the appropriate switch. The switch `-p` allows you to select the port for the listener. You could choose any other port.

When the listener is ready, you can start to pipe the evidence to the responder's system by executing the following:

```
# nc <IP address of responder's system> <port> -e <command>
```

OR

```
# <command> | nc <IP address of responder's system> <port>
```

For example, if you want to pipe the directory listing to the responder's system (with IP address 10.1.2.3), you execute:

```
# nc 10.1.2.3 55555 -e ls
```

OR

```
# ls | nc 10.1.2.3 55555
```

Note that the evidence pipe through netcat is in clear. If you prefer to encrypt the channel (for example, you may suspect there is a sniffer on the network), you can use cryptcat. Cryptcat is the standard netcat enhanced with twofish encryption. It is used in the same way as netcat. Note that the secret key is hardcoded to be "metallica" (use the `-k` option to change this key).

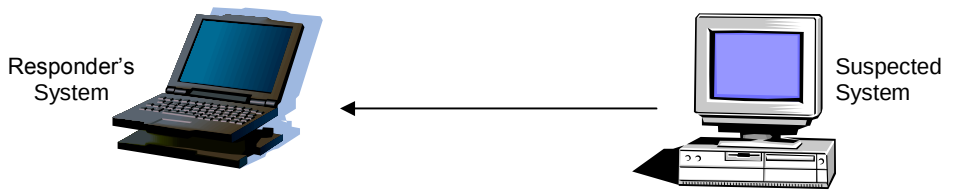

Figure 1: Using netcat to collect evidence

Step Three: Collect Volatile Evidence

The next step is to run the toolkit to collect the volatile evidence.

The system date and time should be recorded before and after collecting the evidence.

Commands	Purpose[1]
# date	* Record the start time of the gathering of evidence. At the same time, check with your current time whether the system time is correct. If not, take note of the difference. This is important when there is a need to correlate with other logs obtained from other systems.
# env	* Record the environment of the system. Note that the PATH and LD_LIBRARY_PATH has changed. You should take note of the old one before setting the new PATH.
# ldd /mnt/cdrom/*	Check and ensure all commands execute will use the trusted toolkit libraries.
# uname –a	* Gather the system information.
# uptime	* Check how long the system has been running. Verify with the system administrator whether the uptime is correct.
# vmstat	* Record the virtual memory statistics to see any abnormality.
# lsmod	* Record all loaded modules.
# df –ah	* Record the file system disk space usage.
# w	* Show who is logged on and what they are doing. * It displays information about the users currently on the system and their processes. The following entries are displayed for each user: login name, tty name, remote host, login time, idle time, JCPU, PCPU and the command line of the current process. * From here, one can collect potential evidence such as any suspicious users or processes running on the system. * Retrieve from utmp file. * Refer to the man pages for more information.
# last –aidx	* Retrieve from wtmp file.
# lastlog	* Retrieve from lastlog file.

[1] Information is mainly extracted from the man pages of the various commands.

# ifconfig –a	* Display the status of all the network interfaces. * From here, one can check whether any interface is running in promiscuous mode. If PROMISC is present, there could be a sniffer running on the system. * Note that Solaris will not display PROMISC. In this case, you will need other means such as lsof and ps to determine.
# netstat –anp # netstat -anr	* Determine the network connections, open ports, routing tables, listening applications. * From here, one can determine any suspicious network connections and applications.
# ps –auxw # ps -eaf (for Sun Solaris platform)	* Report process status. * It gives a snapshot of the current processes. * From here, one can collect evidence on any rogue processes running and their time of execution. * The PID obtained can also be used to investigate further into the /proc directory for the corresponding process. See below for more details on /proc directory.
# lsof –i +M # lsof +L1 # lsof # lsof –D r	* List all open files and the corresponding processes. * -i option will list only those files that have open network sockets. * +M option will list the portmapper associated with the open ports. * +L1 option will list those open files that have been unlinked. This is useful to detect any files used by the attacker that has been deleted (unlinked) but still running or accessible. * The final one without option is to capture everything. * -D r option directs lsof to read the device cache at the default or specific path but prevents it from creating a new device cache file when none exists or the existing one is improperly structured. This is important as we do not want to tamper the system as far as possible.
# ls –alRu (access time) # ls –alRc (modification time) # ls –alR (create time) # ls –alR /proc	* All the timestamp on the files should be captured. This will be helpful when there is a need to correlate one event to another and reconstruct the actions done by the attackers. From the timestamp, you can also gather all files with suspicious timestamp.
# top –b –n1	* Take a snapshot pf the processes

	currently using the most CPU time. This may be helpful to correlate with other evidence gathered.
# date	* Record the end time of the gathering of evidence.

Remember to compute the MD5 value for all the collected evidence to ensure the integrity of the files before carrying out further investigation.

Rootkits

Rootkits can trojanize programs, to lie about and hide information from system administrators. User space tools will not show the true information.

One of the rootkits is Loadable Kernel Modules (LKM). LKM can intercept system commands, hide files, processes and create backdoors. Some examples of LKM rootkits are Knark, Adore and Heroin.

Tools that you can used to detect LKM are Kstat [16] and Chkrootkit [17].

/proc filesystem

From /proc (pseudo filesystem, used as an interface to kernel data structures), we can recover programs that are running in the memory but has already been deleted. Each process has a subdirectory in /proc that corresponds to its process ID (PID).

The following provides significant information for your forensic investigation:

exe link: Allow one to recover deleted files. Use cp to create a copy of the running executables on the system.

fd: Identify all the files a process opens. A file descriptor is being used to reference the associated file or network socket. File descriptor 0, 1, and 2 are predefined for standard input, standard output and standard error respectively.

cmdline: Shows the exact command line arguments used to run an application.

For more details on /proc file system, please refer to Incident Response: Investigating Computer Crime [1].

Information on open ports can also be obtained from:
- /proc/net/tcp
- /proc/net/udp
- /proc/netstat (current connections)

All data are in hex format.

Step Four: Collect other pertinent logs

After gathering the volatile information, the next thing is to gather the pertinent logs. Although this information is not considered to be volatile and could be retrieved during the forensic investigation stage, gathering information from the pertinent logs is very useful in helping the responder to obtain the first hand knowledge of the cause. Note that bit-level image of the media could take a while and during this period, investigation can thus be performed on these logs first.

- /var/adm or /var/log subdirectories
- process accounting logs – lastcomm utility
- web access logs
- xferlog – ftp logs
- history files
- Also check /etc/syslog.conf file to determine any additional logs stored

Files that you should gather and inspect for suspicious event includes:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc hosts
- /etc/hosts.equiv
- ~/.rhosts
- /etc/hosts.allow
- /etc/hosts.deny
- /etc/syslog.conf
- /etc/rc
- /etc/inetd.conf
- crontab files

One may also wish to dump the system RAM and use strings to search for any information that may give you more clues:

- /proc/kmem
- /proc/kcore

Step Five: Perform additional network surveillance

Where possible, it is good to monitor closely any connection to the system subsequently, especially if you suspect the attacker might return. Running a sniffer program on another system to monitor the network activities on that compromised system would be good. Some handy tools to allow you to achieve this are tcpdump, ethereal or even Snort.

A-7. Scripting the Initial Response

The initial response stage will involve much processes and one may not remember all the commands and the appropriate options to execute. Given the time constraint and pressure to undercover the suspicious activities on the system, you may even execute the wrong commands or forget to follow the proper steps, resulting in a loss of important evidence.

A good way to overcome this is to script all those commands and execute them at one go. Put all the trusted tools in a CD-ROM and run the script. By doing this, mistakes will be minimized and it will also provide a standardized procedure. This is especially important when such evidence is required to be presented in court.

An example of the script is provided below. (Please refer to the above table for the complete set of commands.)

```
# Start to collect evidence
date > startdatecollection.txt
env > env.txt
uname –a > uname-a.txt
.
.
.
date > enddatecollection.txt
```

The set of tools to cut into a CD-ROM is described in earlier section.

The root directory of the CD-ROM should contain:

- The required script.
- All the trusted tools described in section A-5.
- A lib directory that contains all the shared libraries required by the trusted tools.

Last but not least, do remember to test all the commands and script to make sure they work.

A-8. What's Next?

After collecting the evidence, the next step is to identify the footprints left by the intruder. For every suspicious event found, the responder should take note of the timestamp and correlate with other logs based on related files, processes, relationship, keywords and timestamp.

Based on initial response finding, one should be able to determine the possible cause of the security breach and decide the next course of action whether to:

- Perform a full bit-level imaging for full investigation;
- Call the law enforcer; or
- Get the system back to normal (reinstall, patch and harden the system).

In all cases, it is best to consult the legal counsel, who is in the best position to advise whether there is a need to notify the relevant customers or partners based on any legislation or contractual agreement.

Part Two: Forensic Investigation on Unix system

B-1. Purpose of Forensic investigation

In the second part of this paper, we will discuss some of the steps in carrying out forensic investigation on a Unix system.

There is really no specific methodology in investigating an incident since every incident could be different. The most important is that the investigator must have a critical mind and a good understanding on the Unix functions so that one is able to respond and analyze the system effectively.

Given that you have already gathered the essential and volatile information earlier. The next step is to investigate the system in details taking both the volatile evidence and the evidence available on the system media. Note that direct investigation on the actual system media will tamper the evidence. Therefore one should create an image of the system media and investigate on the image copy.

B-2. Preparation

For bit-level disk image, there are some available tools that have been recognized to perform an excellent job. Encase and SafeBack are two of the commercial tools that can be considered for image acquisition and restoration, data extraction, and computer forensic analysis. Another tool that you can consider is dd, which is free.

This paper will illustrate some of the ways to image using command dd. One can mount the image using the forensic system with read-only and access the content of the image to investigate the cause.

B-3. Collection and Handling

To image the hard disk without corrupting the evidence, use a separate system to image the victim hard disk. The easiest way to perform imaging is when the hard disk can be removed offline. Otherwise, boot the system from a bootable media and run the operating system from the media. One can then pipe the image over to the investigator's system over the network.

The next step is to connect the victim hard disk to the investigator's system, to create an image for further analysis.

When the victim hard disk is attached onto the investigator's system, check the partition of the victim hard disk first. Simply run `fdisk -l`. This will list the partition tables for the specified device without mounting it. For example,

```
# fdisk -l /dev/hdc
```

After knowing the partition of the victim hard disk, create the image of each partition to the investigator's system. It is better to use a separate hard disk to store the imaged data. The hard disk storage should also be sterilized to ensure it is clean to prevent contaminations of evidence. This step is very important if the case is to be submitted in court in future. The following command can be used to perform the sterilization function:

```
# dd if=/dev/zero of=/dev/hdb (Sterilize the storage media)
```

The commonly used command to create a bit-by-bit image of the media is dd. General command to create an image using dd is

```
# dd if=device of=device bs=blocksize
```

Remember to validate the checksum of the imaged data by comparing the MD5 values.

Mount the victim hard disk in read-only mode. An example to mount in read-only mode:

```
# mount –t ext2 –o ro,loop,noexec,nodev,notime
/victim_images/dev_hdc1.img /mnt/victim_images
```

An example of the command to pipe the victim image to the investigator's system is:

```
(victim system)# dd if=(evidence file to media) | nc x.x.x.x 55555

(responder system)# nc –l –p 55555 | dd of=target.img
```

B-4. Analysis and Investigation

To prevent contamination on the evidence, the investigator should always analyze on the imaged copy.

During the investigation, always look out for suspicious or abnormal events such as the following:

- Check for hidden or unusual files
- Check for unusual processes and open sockets
- Check for unusual application requests
- Examine any jobs running
- Analyze trust relationship
- Check for suspicious accounts
- Determine the patch level of the system

Whenever there is any suspicious observation, take note of the event and timestamp. Correlate the event with other logs based on related files, processes, relationship, keywords and timestamp. The timestamp will also be useful to correlate with external logs such as the logs from firewall and the

intrusion detection system. In short, any suspected events should not be left out.

You may have sieved through the whole system looking for suspicious events. You may have found something to explain the abnormal behavior of the system. However, one may ask: How much is considered enough? How do we know if we have done enough? Have we missed out something?

Unlike hacking, where the hackers just need to find a vulnerability to exploit, an investigator will need to look for all possibilities to uncover the case.

There are no immediate answers to the above questions. However, the minimum requirement is to meet the objective. As every incident is different, always understand the objective clearly before embarking on the investigation.

B-5. Tools to use and their purpose

If you can afford to get a commercial forensics tool to conduct the forensics investigation, consider yourself to be lucky, otherwise, not to worry as there are available freeware tools capable to do the job. The popular ones are The Coroner's Toolkit (TCT), TCTUtils, The Sleuth Kit, and Autopsy Forensic Browser. These tools are often used together for gathering and viewing of the data.

TCT is a collection of programs by Dan Farmer and Wietse Venema for a post-mortem analysis of a UNIX system after break-in. TCT is a collection of tools that can basically organize into four groups:

1. Acquisition. Grave-robber tool is used to capture various types of data and create MD5 hashes to preserve the integrity.

2. Reconnaissance. The set of tools includes pcat, ils, icat and file. They record and analyze processes and inode data.

3. Timelining. Mactime tool is used to create a chronological timeline

4. Recovery. Umrm and lazarus can be used to recover and analyze the disk blocks on the file system.

The article by Derek Cheng gives a very good description on the use of TCT, TCTUtils, and Autopsy Forensic Browser [8].

The Sleuth Kit (previously known as TASK) is a collection of UNIX-based command line file system and media management forensic analysis tools. It can be used to analyze NTFS, FAT, FFS, EXT2FS, and EXT3FS file systems.

The Autopsy Forensic Browser is a graphical interface to the command line digital forensic analysis tools in The Sleuth Kit to allow one to conduct an

investigation easier. Autopsy provides case management, image integrity, keyword searching, and other automated operations [14].

B-6. Recover Deleted Files

Earlier we have discussed that deleted (unlinked) running programs can be retrieved from the /proc filesystem. In the case of deleted files, it may still be possible to recover them.

In Unix, file information is stored in physical disk location known as inode [11]. The inode information includes:

- File access and type information, collectively known as the mode.
- File ownership information.
- Time stamps for last modification, last access and last mode modification.
- Link count.
- File size in bytes.
- Addresses of physical blocks.

When a file is deleted, the link count, file size and data block address are set to zero. However, the data that the inode points to are not deleted. Thus, to recover a deleted file, you will need to know the inode. For a large file, you will obtain a list of inodes.

One can also view a file by referencing its inode number by using a tool, icat (from The, Coroner's Toolkit, TCT [12]).

TCT also contains a list of tools that allow one to identify the inodes that may contain data. You can use ils command to list inode information for every file including those that have already been deleted.

For more information on how to recover a deleted file, see http://www.praeclarus.demon.co.uk/tech/e2-undel/howto.txt.

Part Three: Risk Analysis and Steps to Secure the System

C-1. Introduction

The third and final part of this paper will provide a risk analysis and some basic steps on how to secure the system, to prevent it from being compromised in future.

From my experience, many security incidents result because the systems were not patched promptly or due to misconfiguration. It could be that the system administrators are not familiar with the services or applications or they are simply not aware of the threats and vulnerabilities. In some cases, the system administrators are not equipped with the right knowledge on what need to be configured and how to configure the system securely.

For compromised system, I will recommend reinstalling the operating system and application software from scratch and restore the data from the backup. In this way, you can ensure your system is totally clean. Of course you should ensure that all the data you restored are intact and not contaminated.

We will take an example to demonstrate some of the configurations and hardening steps that need to be taken to secure the system. Our server specifications and requirements are stated below.

C-2. Server Specification

The server is deployed as a web server for extranet consumption. It is hosted at the DMZ where the network segment is meant for both internal staff and business partners to access.

Hardware

Hardware platform	Sun Sparc Ultra-5
CPU	440 MHz
RAM	256 MB
Hard Disk Size	20 GB

Software

Software/Package	Where to get
Sun Solaris 8 Operating System	CD-ROM installation provided by Sun
PERL	http://www.perl.com/download.csp
Apache	http://httpd.apache.org/download.cgi
Mod SSL	http://www.modssl.org/

OpenSSL	http://www.openssl.org/source/
OpenSSH	http://www.openssh.org/
TCP Wrappers	ftp://ftp.porcupine.org/pub/security/
EGD	http://egd.sourceforge.net/

Note that for every software/packages you downloaded, you need to check the MD5 or SIG value to ensure the copy is legitimate and not corrupted.

Services that will be listening on the server are:

- HTTP (Port 80)
- HTTPS (Port 443)
- SSH (Port 22)

C-3. Risk Analysis

Risk assessment is the process of evaluating vulnerabilities and threats to determine the expected loss and establish the degree of acceptability to system operations. At the same time, it has to take into account the business needs and operation constraint.

The basic requirement of the server is to provide web information to the both internal staff and business partners. The network segment that is hosting the server is on a DMZ, protected by a firewall that is only accessible to internal staff and business partners (it is not accessible from the Internet).

As the server will be running web services over port 80 (HTTP), port 443 (HTTPS) and remote access over port 22 (SSH), there is a need to ensure the exposure via these services is minimize where possible and ensure the server is secured against attacks on these services.

Note that all these three applications (Apache, SSL and SSH) are listed as the Top Ten Unix Vulnerabilities in the SANS/FBI Top Twenty Vulnerabilities (http://www.sans.org/top20/). Thus, there is a need to pay extra effort to secure the system from attacks on these applications.

As the web server will be accessible by both internal staff and the business partners. The threats from both ends could be different.

The potential external threats are:

- Exploit the vulnerable component of the application.
- Exploit the bad configuration of the application.
- Exploit the bad coding of the application (e.g. CGI programs).
- Denial of Service.
- Social Engineering.

The potential internal threats, in additional to those mentioned in the external threats are:

- Knowledge of network layout.
- Knowledge of security measures put in place (e.g. firewall configuration, Intrusion Detection System).
- Physical access to the computer room (local attack).

For this case study, we will concentrate on the more serious threats.

We will only specify the steps that are under the administrator's control. We will not describe in details on other security measures put in place that is outside the administrator's control (except for the firewall protection).

- Perimeter protection. Firewall provides a good protection such that it allows only legitimate people to access the legitimate services. In our case, it limits internal staff and business partners to access the server only on port 80 and 443. It also restricts connection to the server on port 22 for specific internal IP address only.

- Proper patch management process. To ensure the server is not vulnerable to the installed services, there should be a proper patch management process to ensure the packages installed are always updated.

- Proper configuration. Even with updated packages, the server should also be properly configured to mitigate the risk that it has been exposed. This includes:

    - At the server end, limit remote access to certain staff only via IP address control and SSH certificate.
    - The server should be hardened such that only necessary services are running on the servers.
    - The server should be hardened to ensure all the access controls are tightened up.
    - The web configuration and application setting should be properly configured. All sample web pages and scripts should be removed.
    - Proper logging on the server should be enabled to monitor any attacks encountered by the servers.
    - Proper backup process.
    - Physical protection.
    - Good password policy.

C-4. Steps to Secure the System (Install, Configure and Harden)

After the risk analysis process, the next step is to install, configure and harden the server in accordance to those risks that we have identified.

This section outlines the steps to install, configure and harden a web server. All steps are done on the console and are categorized into three parts:

- Operating Installation.
- Application Software Installation and Configuration.
- Further Hardening and Securing the System.

The above steps can be referenced independently. For example, if you already have a web server running, you may want to move on to check on the second and third part to ensure your system is configured and secured properly.

There are many guidelines in how to set up a secure web server. We will only list out the minimally essential steps that are easy to follow and configure. This allows you to be able to implement the basic steps in a short time.

Operating System Installation

We will install the system from the SUN issued CD-ROM. We will assume the administrator already has some basic knowledge on Solaris operating system installation, and since the installation is GUI-based and quite straightforward, we will not provide all the detailed installation steps in this paper.

Upon the system start-up, issue a STOP-A to bring the system to OK prompt. At the OK prompt, enter boot cdrom.

OK > boot cdrom

This will restart the system and boot into the CD-ROM to install the operating system.

When prompted for which Solaris Software Group to install, follow the best practices: only install what is needed. Thus, you should select Core System Support.

When ask to partition the hard disk, it will depend on your requirements. One should wipe out everything and perform a fresh installation if there is an existing operating system on the server already.

- The SWAP is usually allocated twice the RAM space.
- Third party software is usually placed and installed in /usr/local/ directory.
- /var directory usually contains all the logs so there should be sufficient space.
- /export/home directory is where users directory are allocated. Since you will be creating a webster user (to be explained in later section), you can place all the web files in this directory. As such, do allocate sufficient space for this partition.

Partition:

```
c0t0d0s0    /                1500
c0t0d0s1    swap             512
c0t0d0s2    whole            all
c0t0d0s3    /usr             3000
c0t0d0s4    /usr/local       5988 (leftover)
c0t0d0s5    /opt             2000
c0t0d0s6    /var             3000
c0t0d0s7    /export/home     4000
```

During installation, you will be prompted to provide a password for root. Remember to assign a good password for root.

After getting the operating system installed, you will need to configure the system so that it can connect to the network. Some of the configurations you need to consider are:

- Configure default gateway in /etc/defaultrouter
  ```
  # vi /etc/defaultrouter
  10.10.1.1
  ```

- Enable DNS in /etc/nsswitch.conf
  ```
  # vi /etc/nsswitch.conf
  hosts: files dns
  ```

- Add nameserver in /etc/resolv.conf
  ```
  # vi /etc/resolv.conf
  domain domain.com
  nameserver 10.10.1.10
  nameserver 10.10.1.11
  ```

- Create the file /etc/notrouter to disable IP Forwarding (since this server is not meant to be a router):
  ```
  # touch /etc/notrouter
  # chown root:sys /etc/notrouter
  # chmod 444 /etc/notrouter
  ```

There are other packages you need to install as they are not installed by default from the Core System Support. For example, you will need:

- GCC compiler
- GNU Zip compression utility
- The Zip compression library

You could obtain the latest version at http://www.sunfreeware.com or http://www.sunfreeware.com/programlistsparc8.html. To install, use the command pkgadd -d <package>

Install Recommended Solaris Patch Cluster

The software package installed from the CD-ROM is usually not the latest version; therefore, always install the Sun Recommended Solaris Patch Cluster to update all the installed software.

To obtain the Patch Cluster, download from ftp://sunsolve.sun.com/pub/patches/8_Recommended.zip or go to http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-license&nav=pub-patches or http://sunsolve.sun.com and download the required patch cluster to /usr/local/ directory.

```
# showrev –p to check the list of patches already installed
```

Run the install_cluster script:

```
# cd /usr/local/8_Recommended (patch cluster directory)
# ./install_cluster
```

If a patch installation fails with:

- return code 8: Patch applies to a package which has not been installed
- return code 2: These patches have already applied to the OS image loaded from CD-ROM

You can view the installation log file at /var/sadm/install_data/.

Note that the server should be rebooted to complete the patching process.

Application Software Installation and Configuration

Next is to install the application software. Most of the third party software will be stored and installed from /usr/local/ directory. In most cases, the software is tar (.tar extension) and gzip (.gz extension).

To unzip, the command is gunzip <software>
To untar, the command is tar xvf <software>

Install PERL

We will need PERL to run some of the applications. You can get PERL at http://www.perl.com/download.csp. Download it to /usr/local/directory.

After unzip and untar, installation is pretty straightforward.

```
# ./configure
# make
# make install
```

Next install the EGD, which is an Entropy Gathering Daemon. This is required for OpenSSH and OpenSSL.

Install EGD

EGD can be obtained from http://egd.sourceforge.net/ or http://www.sunfreeware.com/programlistsparc8.html.

After unzip and untar, go to the egd directory.

```
# perl Makefile.PL
# make
# make install
```

On the same directory, run egd.pl to get the entropy gathering daemon running:

```
# ./egd.pl /tmp/entropy
```

To check it is running:

```
# ps –ef |grep entropy
```

Install OpenSSL

Since we require SSL running on the server, we will install OpenSSL to generate a SSL Certificate. As this is only an extranet application and the purpose of the SSL is to create a secure channel for some of the web services, it is not necessary to obtain a commercial Certificate Authority (CA) to sign the digital certificate. The SSL Certificate will be self-signed.

Download OpenSSL from http://www.openssl.org/source/.

After unzip and untar, installation is as normal:

```
# ./configure
# make
# make test
# make install
```

Edit /usr/local/ssl/openssl.cnf to set the appropriate variables:

- `RANDFILE  = /tmp/entropy`
- `top = /usr/local`
- `dir = $top/OpenSSL_CA`
- `default_days = 730`
- `CountryName_default = <Your Country 2-digit code>`
- `stateOrProvinceName_default = <Your Country>`
- `O.organizationName_default = <Your Organization>`
- `organizationUnitName_default = <Your Division>`
- `#nsComment = "OpenSSL Generated Certificate"`

The files and directories to be created are:

```
/usr/local/OpenSSL_CA/certs/        (where the issued certs are kept)
/usr/local/OpenSSL_CA/crl/          (where the issued crl are kept)
/usr/local/OpenSSL_CA/index.txt     (database index file)
/usr/local/OpenSSL_CA/newcerts/     (default place for new certs)
/usr/local/OpenSSL_CA/cacert.pem    (CA certificate)
/usr/local/OpenSSL_CA/serial        (current serial number)
/usr/local/OpenSSL_CA/crl.pem       (current crl)
/usr/local/OpenSSL_CA/private/cakey.pem   (private key)
```

You will also need to set the appropriate variables for /usr/local/ssl/CA.sh:

- DAYS = "-days 730"
- TOP=/usr/local
- CATOP=$TOP/OpenSSL_CA

The commands to setup the CA, create a certificate and sign the certificate are as follows:

1. Create CA directory and Generate CA Certificate
   In /usr/local/ssl/misc, run **./CA.sh –newca**
   (private key will store in
   /usr/local/OpenSSL_CA/private/cakey.pem)
   (CA Certificate will store in /usr/local/OpenSSL_CA/cacert.pem)

2. Generate Certificate Request (User/Server Certificate)
   In /usr/local/ssl/misc, run **./CA.sh –newreq**
   (newreq.pem (with private key) will store in
   /usr/local/ssl/misc)

3. Sign the Requested Certificate
   In /usr/local/ssl/bin, run **./openssl ca -verbose -in /usr/local/ssl/misc/newreq.pem -out /usr/local/ssl/certs/<filename.pem>**
   (Certificate generated will be stored in
   /usr/local/OpenSSL_CA/newcerts/<serial.pem> and output file stated above)

Generally, we have the commands as:

```
# CA -newca (This will setup the right stuffs such as create
directory hierarchy and generate CA certificate)
# CA –newcert (This will create a new self-signed SSL certificate)
# CA -newreq (This will generate a certificate request (to be signed
by CA later)
# CA -sign (This will sign the generated certificate request)
```

Install Apache and Mod SSL

Now we are ready to install apache with mod ssl. Before we install them, you may want to consider changing the Apache version to something else. Although this is security through obscurity, a small additional step to add to security is always better. Of course we should not rely on this to provide the security. Rather, this offers a protection against some hacker tools or malcodes which may check the version of the server before running the exploit.

Download the Apache (1.3.31, currently the latest version) and Mod SSL (2.8.18) from

Apache: http://httpd.apache.org/download.cgi
ModSSL: http://www.modssl.org/

After unzip and untar the software,

Modify the default web server version

- /usr/local/apache_1.3.31/src/include/httpd.h
  #define SERVER_BASEPRODUCT "Apache" to "<your own name>"
  #define SERVER_REVISION "1.3.31" to "<your own name>"

Install Mod SSl and Apache

Next is to install Mod SSL and Apache. You should unzip and untar both apache and mod ssl package at the same time. Go to Mod SSL directory and comment out the following in the file `/usr/local/mod_ssl-2.8.18-1.3.31/configure` (As you have changed the apache version earlier, comment this out is necessary so that mod ssl will not check the version. Otherwise, the installation will not be successful.)

- gunzip mod_ssl-2.8.18-1.3.31.tar.gz
- tar xvf mod_ssl-2.8.18-1.3.31.tar
- gunzip apache_1.3.31.tar.gz
- tar xvf apache_1.3.31.tar
- cd mod_ssl-2.8.18-1.3.31
- Remark the necessary ifloop in /usr/local/mod_ssl-2.8.18-1.3.31/configure (not to check apache version) (do a search for apache)

```
#if [ .$force != .yes ]; then
#    if [ ".$V_APACHE" != ".$APV" ]; then
#        echo "$0:${T_MD}Error${T_ME}: The mod_ssl/$V_MODSSL can be
used for Apache/$V_APACHE only." 1>&2
#        echo "$0:${T_MD}Error${T_ME}: Your Apache source tree under
$apache is version $APV." 1>&2
#        echo "$0:${T_MD}Hint${T_ME}:  Please use an extracted
apache_$V_APACHE.tar.gz tarball" 1>&2
#        echo "$0:${T_MD}Hint${T_ME}:  with the --with-apache option,
only." 1>&2
#        exit 1
#    fi
#fi
```

Configure MOD SSL:

- ./configure --with-apache=/usr/local/apache_1.3.31 \
  --with-ssl=/usr/local/openssl-0.9.7d \
  --with-crt=/usr/local/ssl/certs/<certname.pem> \
  --with-key=/usr/local/ssl/misc/newreq.pem \
  --prefix=/usr/local/apache

Configure Apache:

- `cd /usr/local/apache_1.3.31`
- `./configure --prefix=/usr/local/apache --enable-module=ssl --enable-module=so`
- `make`
- `make install`

- check /usr/local/apache/conf/httpd.conf (apache configuration file)
- To run without ssl, `# /usr/local/apache/bin/apachectl start`
- To run with ssl, `# /usr/local/apache/bin/apachectl startssl`
- To stop, `# /usr/local/apache/bin/apachectl stop`

For more details on Mod SSL, please refer to http://www.modssl.org/docs/

Now the basic web server should be running using apache. The next step is to configure the apache configuration files (/usr/local/apache/conf/httpd.conf).

Create a user webster and group www. This will be the place where you store all your web pages. Make sure the permissions are set correctly for this directory. When upgrading from one version of apache to another, this will also be easier as there is no need for you to transfer the web pages from one directory to another.

Remove all sample scripts, images apache documentation. The following should be modified for apache configuration file /ust/local/apache/conf/httpd.conf

Point to the appropriate web directory:
- DocumentRoot "/export/home/webster/htdocs"

Turn the Apache signature off:
- ServerSignature Off

Redirect error page to your customize error page:
- ErrorDocument 400 /errorPage.htm
- ErrorDocument 401 /errorPage.htm
- ErrorDocument 402 /errorPage.htm
- ErrorDocument 403 /errorPage.htm
- ErrorDocument 404 /errorPage.htm

Note that Apache is listed one of the SANS/FBI Top Twenty Vulnerabilities. You should check the listed recommendations for further information on how the web server can be secured, http://www.sans.org/top20/#u3.

Install OpenSSH

Note that as the server will be hosted in the computer room, it is unlikely you will access the computer room to make changes to the web pages. One way is to have remote access to the server at your office desk. Using telnet or ftp

is not desirable as information (including User ID and password) transmitted are in clear. OpenSSH is meant to access the server in a more secure manner. Information transmitted over the network should be encrypted. You can also impose more access control on who can have access to the server for this service. For example, instead of just rely on User ID and password, you could configure SSH to use client and server certificate authentication.

Download OpenSSH from http://www.openssh.org/. Steps to install are as follows (current version is openssh-3.8p1.tar.gz):

- `gunzip openssh-3.8p1.tar.gz`
- `tar xvf openssh-3.8p1.tar`
- `Goto the directory <openssh-3.8p1>`
- `./configure --with-prndg-socket=/tmp/entropy --with-tcp-wrappers --with-md5-passwords --with-ssl-dir=/usr/local/openssl-0.9.7d --with-privsep-user=nobody`
- `make`
- `make install`

After installation, we have:

- public/private key pair in /usr/local/etc/
- sshd.pid store in /var/run/sshd.pid
- User binaries: /usr/local/bin
- System binaries: /usr/local/sbin
- Configuration files: /usr/local/etc
- For more information, check /usr/local/openssh-3.8p1/Makefile
- To run ssh daemon, `# /usr/local/sbin/sshd`
- known_hosts file in /.ssh/

Create a startup script for sshd. Install the following script into /et/init.d/sshd:

```
#! /bin/sh
#
# start/stop the secure shell daemon
# Make links to get sshd start/stop at the right time
# /etc/rc0.d/K57sshd
# /etc/rc1.d/K57sshd
# /etc/rc2.d/S99sshd

case "$1" in

'start')
        # Start the audit deamon
        if [ -f /usr/local/sbin/sshd ]; then
                echo "starting SSHD daemon"
                /usr/local/sbin/sshd &
        fi
        ;;

'stop')
        # Stop the audit deamon
```

```
        PID=`/usr/bin/ps -e -u 0|/usr/bin/fgrep sshd|/usr/bin/awk
'{print $1}'`
        if [ ! -z "$PID" ] ;  then
                /usr/bin/kill ${PID} 1>/dev/null 2>&1
        fi
        ;;
*)
        echo "usage: /etc/init.d/sshd {start|stop}"
        ;;
esac
```

- `# chmod 744 sshd (to make it executable)`
- Make hard links:

In the directory /etc/rc0.d, `# ln /etc/init.d/sshd K57sshd`
In the directory /etc/rc0.d, `# ln /etc/init.d/sshd S99sshd`

For more details on installation, configuaration, and operation on the secure shell (SSH), on systems running Solaris 2.x, you may like to refer to CERT: http://www.cert.org/security-improvement/implementations/i062_01.html

Further Hardening and Securing the System

Warning Banner

You can consider display a warning banner when user login into the system. This will be useful to inform users or in fact the intruders of the legal implications for unauthorized access to the system. Thus, In the event of a legal proceeding, the presence of such banner will come in useful.

To add a login banner, create or amend in the file /etc/issue and /etc/motd file.

Although Telnet and FTP will not be running, for added security, you can still consider setting a banner for them.

For Telnet, create a file /etc/default/telnetd with content:

```
BANNER="<Your Banner\n>"
```

For FTP, create a file /etc/default/ftpd with content:

```
BANNER="<Your Banner>"
```

Ensure the proper access rights are set on the files:

```
# chown root:sys /etc/motd
# chown root:root /etc/issue
# chmod 644 /etc/motd /etc/issue
# chown root:sys /etc/default/telnetd /etc/default/ftpd
# chmod 444 /etc/default/telnetd /etc/default/ftpd
```

Install TCP Wrappers

After installing SSH, we now can have remote access to the system via SSH. To restrict who can SSH to this system, we can use TCP Wrapper to control the access.

TCP Wrapper is a filter program where it checks the permitted requested process before passing over to the actual program. In this way, it provides some level of access control based on the source and destination of the connection request and also logging for successful and unsuccessful connections.

It can be obtained from ftp://ftp.porcupine.org/pub/security/. CERT CC has a good documentation on the installation, configuration, and the use of tcp wrapper to log unauthorized connection attempts on systems running Solaris 2.x. You can refer it at http://www.cert.org/security-improvement/implementations/i041.07.html.

After the installation, you will need to create the files /etc/hosts.allow and /etc/hosts.deny.

- Create /etc/hosts.allow with the following content (assuming only the subnet 10.10.10.0/255.255.255.0 needs to access the server only).

```
# vi /etc/hosts.allow
sshd: localhost, 10.10.10.
ALL: ALL: DENY
```

- Create /etc/hosts.deny with:

```
# vi /etc/hosts.deny
ALL: ALL
```

Configure Network Parameters

Create the appropriate setting for the network parameters. This will enhance the system from network attacks.

```
cat <<END_SCRIPT >/etc/init.d/networkconfig
#!/sbin/sh
ndd -set /dev/ip ip_respond_to_timestamp 0
ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0
ndd -set /dev/ip ip_forward_src_routed 0
ndd -set /dev/ip ip_forward_directed_broadcasts 0
ndd -set /dev/ip ip_forwarding 0
ndd -set /dev/ip ip_ignore_redirect 1
ndd -set /dev/ip ip_send_redirects 0
ndd -set /dev/ip ip_strict_dst_multihoming 1
ndd -set /dev/arp arp_cleanup_interval 60000
ndd -set /dev/ip ip_ire_arp_interval 60000
ndd -set /dev/tcp tcp_sack_permitted 2
ndd -set /dev/tcp tcp_conn_req_max_q0 8192
ndd -set /dev/tcp tcp_ip_abort_cinterval 60000
```

```
END_SCRIPT

chown root:root /etc/init.d/networkconfig
chmod 744 /etc/init.d/networkconfig
ln -s /etc/init.d/netconfig /etc/rc2.d/S69networkconfig
```

Randomized the Initial Sequence Number of all TCP Connections

This will provide protection against session hijacking and IP spoofing.

```
# vi /etc/default/inetinit
TCP_STRONG_ISS=2
```

Amend /etc/inetd.conf

There is no need to have ftp and telnet services running anymore. In fact, you should comment out all the services stated in inetd.conf. This includes ftp, telnet, shell, login, exec, tftp, finger, printer, etc.

Restrict remote root login

You should configure to disallow remote root login. Administrator can still login as a normal user and then issue su to escalate into root privileges. This should be the proper way so that there will be proper logging to check who is using root privileges to access the system. The following in the /etc/default/login should not be comment out:

```
# vi /etc/default/login
CONSOLE=/dev/console

# vi /etc/sshd_config
PermitRootLogin no
```

Do check out the other setting in /etc/default/login to suit your operations needs.

Disabled the rlogin

Commented out the following lines in /etc/pam.conf:

```
# rlogin auth sufficient /usr/lib/security/pam_rhosts_auth.so.1
# rlogin auth required /usr/lib/security/pam_unix.so.1
# rsh auth required /usr/lib/security/pam_rhosts_auth.so.1
```

Remove Unnecessary Services

Remove unnecessary startup script/services in /etc/rc2.d and /etc/rc3.d. You can either remove them or rename them from Sxx to NOSxx. For example:

```
# mv S15nfs.server NOS15nfs.server
```

Note that as we have done a base installation, not all services will be available for turning off. However, we will still include them so that you can use them as a checklist to harden other servers that are currently running.

Startup Services	Description
S15nfs.server	NFS Server. If there is no need to share out
S30sysid.net	Use for Auto Configuration
S40llc2	Logical Link Control
S47asppp	Asynchronous PPP
S70uucp	UUCP
S71sysid.sys	Use for Auto Configuration
S71ldap.client	LDAP directory services
S72autoinstall	Use for Auto Configuration
S72slpd	SLPD
S73cachefs.daemon	Cachefs
S73nfs.client	NFS Client.
S74autofs	Auto Mounter
S75flashprom	Flash Prom Update
S76nscd	NSCD
S76snmpdx	SNMP
S77dmi	DMI
S80PRESERVE	Preservation of Files Killed by Vi
S85power	Power Management
S88sendmail	Since this system is not running any mail service, you should disable this. You can still send mail from this system even though the service is disabled. Just configure the appropriate mail relay server in the /etc/mail/sendmail.cf. # vi /etc/mail/sendmail.cf #DSmailhost.$m DShostname.domain.com
S89bdconfig	Buttons n Dials-Setup”
S80mipagent	Mobile IP Agent
S80lp	Printer
S80spc	SPC
S90wbem	Sun Management Center
S92volmgt	Volume Manager. This will disable CD-ROM to be automatically mounted.
S93cacheos.finish	cacheos.finish
S94ncalogd	Network Cache and Accelerator
S95ncad	Network Cache and Accelerator

Lastly, turn off RPC and Dtlogin. Before you turn off these two startup scripts, comment out the Open Windows GUI startup, otherwise you will not be able to get into the system after a reboot.

```
# mv S71rpc NOS71rpc (RPC, CDE GUI will require)
# mv S99dtlogin NOS99dtlogin (Start CDE GUI login)
```

Make comments to /.bash_profile to stop Open Window from starting.

```
# If possible, start the windows system
#
#if [ "`tty`" = "/dev/console" ] ; then
#       if [ "$TERM" = "sun" -o "$TERM" = "sun-color" -o "$TERM" =
"AT386" ]
#       then
#
#               if [ ${OPENWINHOME:-""} = "" ] ; then
#                       OPENWINHOME=/usr/openwin
#                       export OPENWINHOME
#               fi
#
#               echo ""
#               echo "Starting OpenWindows in 5 seconds (type
Control-C to interrupt)"
#               sleep 5
#               echo ""
#               $OPENWINHOME/bin/openwin
#
#               clear           # get rid of annoying cursor
rectangle
#               exit            # logout after leaving windows system
#
#       fi
#fi
```

Enable Logging

In the event of an audit or investigation, proper logging is very important. Thus, there is a need to ensure logging is enabled for both system and applications.

In the file /etc/default/login, set

```
# vi /etc/default/login
RETRIES=3
SYSLOG_FAILED_LOGINS=3
```

This will log all failed login attempts. Note that the file /etc/adm/loginlog should be present. If not, create one:

```
# touch /var/adm/loginlog
# chown root:sys /var/adm/loginlog
# chmod 600 /var/adm/loginlog
```

Add the following in the /etc/syslog.conf:

```
auth.info   /var/log/authlog
```

Ensure the file /var/log/authlog exists with the correct access rights:

```
# touch /var/log/authlog
# chown root:sys /var/log/authlog
# chmod 600 /var/log/authlog
```

Apache by default log to its own log directory.

Enabled Basic Security Module (BSM)

Run the script to enable BSM. You will need to reboot for it to be effective. Configure BSM by editing the files in /etc/security/.

```
# /etc/security/bsmconv
```

Configured the Classes of Events to Log

```
# vi /etc/security/audit_control
dir:/var/audit
flags:lo,ad,pc,fc,fd,fm
naflags:lo,ad
#
# lo - login/logout events
# ad - administrative actions: mount, exportfs, etc.
# pc - process operations: fork, exec, exit, etc.
# fc - file creation
# fd - file deletion
# fm - change of object attributes: chown, flock, etc.
#
```

Fix-modes

Fix-modes is a tool consisting of a set of scripts which modifies file permissions to make things more secure. It is written by Casper Dik. The tool removes group and world write permissions of all files, devices, and directories listed in /var/sadm/install/contents, with the exception of those files listed in exceptions.h, and changes ownership of most files to root. You can download it from http://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/fix-modes/.

Restricted FTP Usage

The following accounts should be contained in the /etc/ftpusers. This will disable them from ftp to the server.

```
# vi /etc/ftpusers
root
daemon
bin
sys
adm
lp
uucp
nuucp
listen
nobody
```

```
noaccess
nobody4
```

Restricted Access to "at" and "crontab" Commands

Restrict the use of "at" and "crontab". Only users listed in these files will be allowed to use "at" and "crontab". These accesses should be given out only on need-to-have basis.

```
# vi /etc/cron.d/cron.allow
# chmod 600 /etc/cron.d/cron.allow
# cp –p /etc/cron.d/cron.allow /etc/cron.d/at.allow
```

Set Good Password

Ensure users have good password and the system account cannot be used to login into the system. Remove or disable the unnecessary accounts:

```
# passwd –l adm
# passwd –l bin
# passwd –l daemon
# passwd –l listen
# passwd –l lp
# passwd –l nobody
# passwd –l noaccess
# passwd –l nuucp
# passwd –l sys
# passwd –l uucp
```

C-5. Design and Implement Ongoing Maintenance

Setting and configuring the system securely is not the end. In fact it is just the beginning. You will need to have ongoing maintenance to ensure the system is running smoothly. Some of the ongoing maintenance you need to consider and implement are discussed below.

Proper Documentation

Proper documentation of the system setup is very essential. However, in many cases, documentation is not done properly.

The following is a list of documentation that you need to establish.

- What are the logs to monitor.
- Where the logs are stored.
- The file system layout.
- What are the critical files to backup: configuration files, keys, data, scripts, etc.
- When the system is startup, what services will be running, how to ensure they are running and how to start running them if not.
- What are the configurations that have been changed, and what are the changes.
- How is the system hardened.

If there is IDS monitoring the network, then it is a good means to detect malicious activities. Otherwise, you may want to consider installing Snort on the system to monitor the activities on the system.

Ongoing Patching

You will need to establish a patch management guidelines and procedures. From time to time, there will be new vulnerabilities and updates on the software and packages that need to be installed on the system. You should get them updated. From the security point of view, such update should be done as soon as possible. However, there is a need to balance the business needs and operations constraint. Whether to update immediately or at the next maintenance schedule time frame, will really depend on the vulnerability. The Rule of Thumb in considering the patch cycle:

- Vulnerability: Does the vulnerability exist? Is there known exploit code? Is the vulnerability vulnerable to local or remote exploit?
- Exposure: Are the vulnerable systems exposed to the attack?
- Impact: What is the impact if systems were compromised?
- Mitigating Controls: Is there any mitigating controls to minimize the system from exposure to attack?

Log Review

It is good to ensure all logging is enabled. But it will be useless if the logs are not reviewed to check and detect anomalies on the system. From the log review, you may detect

- Malicious users trying to intrude into the system.
- Worms that appear on the network.
- System problem.
- System usage.

By having a regular log review, this will help you to identify the problem earlier and take necessary action to prevent or mitigate the risks.

Log review could be tedious. You may want to use tools to help you to do this part of work. An example will be Swatch http://swatch.sourceforge.net/.

Backup

Backup is important to ensure your data are not lost in the event of system failure or system compromised. It will also help you to restore the system back if there is a need. There should be a backup process. This includes incremental backup and full backup. You will need to define the frequency and period of backup.

C-6. Test and Verify the Setup

Having all the proper setting and configuration, you should test and verify that all the setting and configurations are working as intended. Otherwise, it will just give you a false sense of security.

- Reboot the server and check that there is no GUI login.
- Check that root is only allowed to login at console.
- Do a netstat -an to check that only port 80, 443 and 22 are listening. No other services on other ports should be running.
- Check that SSH is only able to login at the specific IP address segment. This is to ensure TCP Wrapper is working fine.
- Check all logging is working fine.
- When login via SSH, check that the warning banner is effective.
- Check that the apache is running correctly (`# ps -ef |grep apache`).

Nessus

Perform a network scan using Nessus. Nessus is a very useful tool to assess the system vulnerabilities. You can download Nessus from http://www.nessus.org/. We will not cover the details of Nessus. However, there are a few good papers written by Harry Anderson on how to use Nessus:

Introduction to Nessus: http://www.securityfocus.com/infocus/1741
Nessus, Part 2: Scanning: http://www.securityfocus.com/infocus/1753
Nessus, Part 3: Analysing Reports: http://www.securityfocus.com/infocus/1759

Benchmarks Tools from The Center for Internet Security

The Center for Internet Security provides a list of benchmarks tools to assist administrators to check their system's configuration and setting. The tools can help you to measure how good your system setup is.

Download the Solaris benchmarks tool and use it to run on your server. This will give you a basic idea on how well your system is configured.

You can get the list of benchmarks tools at http://www.cisecurity.org.

Conclusion

In this paper, we have discussed how to respond and investigate a Unix security incident. It provides you an insight on the proper initial response to incidents, how to conduct an investigation and subsequently how to perform a risk analysis and set up a secure Unix system.

Having good preparation to response to any security incidents will save a lot of time and effort in handling the cases. In fact, ill preparation can impede future investigation and even jeopardize the case. Therefore, planning ahead is necessary to be successful in tackling all security incidents.

As the saying goes, practice makes perfect. Do not wait for a security incident to occur before you start to kick in your established plan, checklist and toolkit.

Last but not least, there are lessons to be learnt from every incident. It is useful to learn from the experience and ensure future errors will not be repeated.

References

[1] Kelvin Mandia and Chris Prosise, “Incident Response: Investigating Computer Crime”, Osborne/McGraw-Hill, July 2001, ISBN: 0-07-213182-9

[2] James Seddon, “Forensic UNIX Initial Response Script and CDROM – Collect the evidence that will be lost by disconnection or shutdown”, URL: http://www.giac.org/practical/GCUX/James_Seddon_GCUX.pdf

[3] Hai Pomeranz, “Static Linking Under Solaris”, URL: http://www.deer-run.com/~hal/sol-static.txt

[4] Mariusz Burdach, “Forensic Analysis of a Live Linux System, Part One”, 22 March 2004, URL: http://www.securityfocus.com/printable/infocus/1769

[5] Mariusz Burdach, “Forensic Analysis of a Live Linux System, Part Two”, 12 April 2004, URL: http://www.securityfocus.com/printable/infocus/1773

[6] Holt Sorenson, “Incident Response Tools For Unix, Part One: System Tools”, 27 March 2003, URL: http://www.securityfocus.com/printable/infocus/1679

[7]: Holt Sorenson, “Incident Response Tools For Unix, Part Two: File-System Tools”, 17 October 2003, URL: http://www.securityfocus.com/printable/infocus/1738

[8]: Derek Cheng, “Freeware Forensics Tools for Unix”, 1 November 2001, URL: http://www.securityfocus.com/infocus/1503

[9] Gideon Rasmussen, “Solaris * Build Document”, URL: http://sysunconfig.net/unixtips/Solaris_build_document.pdf

[10] Koon Yaw Tan, “Windows Responder’s Guide”, URL: http://www.giac.org/practical/GSEC/KoonYaw_Tan_GSEC.pdf

[11] http://www.scit.wlv.ac.uk/~jphb/spos/notes/ufs.inode.html

[12] http://www.porcupine.org/forensics/tct.html

[13] http://www.fish.com/tct/FAQ.html

[14]: http://www.sleuthkit.org/

[15]: http://www.atstake.com/research/tools/forensic/

[16]: http://www.s0ftpj.org/en/tools.html

[17]: http://www.chkrootkit.org/