02230: Computer Security
Lab 5: Auditing

Robin Sharp

Autumn 2005

This lab gives you the opportunity to try out some of the techniques which are available for checking for undesired activity in a computer system. The task requires you to do some detective work in a Linux-based system, in order to determine whether undesired activity appears to have taken place and to find out (as far as possible) what has happened. You are required to hand in a report giving a short description of the results which you obtain.

The lab consists of two parts, each of which relates to a particular aspect of auditing:

1. Logging
2. Forensic examination

1 Logging

Logging is the technique of recording appropriate information about important events which take place in the system. In a typical Linux system, log files are by default kept in the directory `/var/log/`. Of particular interest are the files:

File	Content
`/var/log/messages`	Start and stop of logging, messages about attempts to login and logout.
`/var/log/secure`	Security-related messages, such as attempts to use secure login, possible security failures etc.
`/var/log/httpd/access_log`	Attempts to access web server
`/var/log/httpd/error_log`	Errors in attempts to access web server.

In this first part of the lab, you are required to investigate what was going on in the Apache Web server `galadriel.it.dtu.dk` in the period from 9 March 2003, 04:09:07 to 13 March, 02:29:58. During this period, a lot of attempts were made to access the server, and many of these attempts gave rise to entries in the server error log, `error_log`. Copies of the access and error logs for this period have been saved, and can be found on `galadriel.it.dtu.dk` in:

```
/var/log/dump02230/access_log
/var/log/dump02230/error_log
```

These are – at least in principle – plain text files. The reason it is only "in principle" is that the files may contain unusual control sequences sent off by hackers as part of their attempt to break into the system. The detailed content and format of the files is discussed in Apache's web pages starting at:

```
http://httpd.apache.org/docs-2.0/logs.html
```

To interpret the files, you need to know that the configuration file for the Apache server contains the directive:

```
CustomLog logs/access_log combined
```

which means that the access log uses the format defined as "combined". This is in turn defined by the directives:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{UserAgent}i\"" combined
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{UserAgent}i" agent
```

The error log in Apache servers follows a standard format (see the Web page referred to above), and is not controlled by configuration directives. The level of the error log is set to *warn*.

You should try to explain in as much detail as possible what was going on in the period concerned. Since the log files are rather repetitive, you do not need to explain each individual line in both the logs, but should concentrate on the main types of entry and their significance for the security of the server. You are encouraged to look on the Web and in any other sources which you can get access to in order to discover suitable explanations.

2 Forensic examination

Forensic examination involves studying the system as a whole in order to find evidence of malicious activity. Looking in the log files is a part of this, but malicious activity can in principle *change the content of any file in the system.* Experienced hackers who break into a system usually try to cover their tracks, and typical ways of doing this include:

- Deleting some or all of the log files.
- Temporarily turning off logging while malicious operations are being carried out.
- Setting up unauthorised privileged user accounts, say with root privileges, in order to perform privileged operations without setting off any alarms.
- Substituting standard system programs such as `ls` and `ps` with fake programs which do not list files and processes set up by the hacker, so that the ordinary user or system manager cannot immediately see that anything unusual is going on.
- Hiding files needed for the malicious activity in directories which do not normally contain ordinary files, such as directories rooted in `/tmp`, `/dev` or `/proc`.

In a high-security system implemented to TCSEC level C2 or above, secure auditing, which provides a non-modifiable audit trail of all accesses to objects, is required and makes some of these forms of attack impossible to carry out without detection. However, in a standard Linux system, there is no guarantee that secure auditing is available.

To perform an accurate forensic examination, you need to work on a true copy of your file system made at the time when the malicious activity was discovered. This should be a "bit-for-bit" copy made using the Unix program `dd` or equivalent. In this lab, we assume that this has been done, and that the files which we are looking at are genuine ones, as left behind by the hacker, and that there are no deleted files which once contained important forensic information. (If this were not the case, you would have to use a suitable tool to recover any deleted files first.) You may also assume that the basic system programs (`ls`, `ps`,...) have been checked and found to be working normally. In a real system, readily available tools such as `tripwire` can be used to check regularly that this is the case (see `http://www.tripwire.org`).

The files which have been recovered for use in this investigation are available as follows:

Original files	Copy stored in
/etc/passwd	/var/log/dump02230/passwd
/tmp	/var/log/dump02230/tmp
/dev	/var/log/dump02230/dev

`passwd` is as usual the password file, while `tmp` and `dev` are directories which contain (large numbers of) further files. You should analyse these as well as you can, to see whether you can find any suspicious features. You can find information about the normal layout of the `passwd` file in section 5 of the man pages. Information about what the various parts of the directory listing tell you can be found in the manual page for the `ls` command (in section 1 of the man pages).

3 Vulnerability analysis

Vulnerability analysis is used to check your system for well-known vulnerabilities, i.e. potential threats which are based on known methods of attack. This is usually thought of a part of the system auditing process, since it provides information about things which potentially could change. However, it is not possible to set up a useful vulnerability analysis system in a simple manner in DTU's databars without exposing the databar computers in an unethical manner. There are a number of well-known tools for vulnerability analysis, such as:

- `nmap` for determining open ports and the associated applications (so-called *footprint analysis*),
- `whisker` for analysis of CGI vulnerability,
- `chkrootkit` for checking for the presence of so-called *rootkits* left in your system by hackers in the form of backdoors, Trojan horses or similar malicious programs.
- `Nessus` for general vulnerability analysis.

Most of these analysis programs need the user to have root privileges if they are to give really useful information. If you want to try them out, do so at home, and don't use DTU computers as the targets!

4 Reporting your results

You should present your analysis of what has happened in the system in the two cases considered here in a short report which you hand in **not later than 17:00 on Wednesday 16 November**. Note that the two cases are independent of one another, so you should not expect to be able to use information which you discover by looking in the log files when you try to analyse the other forensic material.

A report is to be handed in by each 2-person group. Please leave your report in one of the "mailboxes" marked 02230 in the entrance to Building 322.

Robin Sharp
October 2005.