UNIX AUDITING - THROUGH ACCOUNTING

James F. Ridgway
3 December 2000

Background

The UNIX Accounting package consists of a standard set of utilities which allow system administrators to perform system performance monitoring, trouble shooting and to maintain systems security. Many SysAdmins do not run “accounting” or are not aware that it exists as on many systems it must be loaded and explicitly activated before it can be used. The utilities can be used to provide daily reports on demand or through “cron” to selectively search for specific users to record and view their activity. Originally, accounting was designed to enable sites to establish a billing scheme based on user connect times.

Types of Accounting

There are several subsets of accounting records provided. The two most important for security auditing are:

1) Connect Accounting - This includes the length of time a user has been logged into the system, the TTY line that was in use, the number of reboots including times, etc.
2) Process Accounting - This includes the user and group Ids that used the process/utility/application, beginning and end times of the process, CPU time and memory used, the commands the user ran and the TTY that was used to control the process.

Accounting uses some of the same files that the UNIX command “last” uses. In fact, it records, formats and makes available in various reports (or dynamic reports through shell command manipulation) information included in the “/etc/wtmp”, “/var/adm/su”, and other logs. Session connects, System state changes, reboots, shutdowns, etc. are retrieved from /etc/wtmp. Other accounting information is stored in a “pacctx” data file which is compiled daily into the Spacctxxxxx file (labeled by date).

Common Usage

When a process ends, a record in the form of “acct.h” is written into the “pacct” file. Then on a scheduled basis you may want to use any or all of the following programs which query this file and create reports.

1) runacct - Typically run nightly from “cron” to process files; /var/adm/pacctxxx, /var/adm/wtmp, /var/adm/fee and /var/adm/acct/nite/disktacct which will produce daily reports summarizing system usage by login

2) prdaily - This program is also run on a daily basis (as the name implies) to process the information gained by “runacct” and build report files by day and month. The combination of these two command allows you to build and maintain system usage reports each day and keep a historical record of them.

Reports include:

Daily Report -

1) Terminal line or port #
2) Minutes that the line was used
3) Percent of time (based on the starting/ending times of the report period) Terminal was used
4) Number of times the port was used in a login session
5) Number of times a user logged out from the given port

Daily Usage Report -

1) User ID that logged in
2) The Login Name
3) CPU Minutes used by processes on behalf of the user
4) Amount of memory (cumulative) the user’s process used
5) Connect Time in minutes
6) Number of disk blocks the user requested
7) Number of processes invoked by the user
8) Number of times the user logged into the system

Total Command Summary - This Report basically provides a list of all Commands/utilities run on the system for the given reporting period as well as the total CPU time, memory required, characters transferred and blocks read for each command. It does not associate user ID’s with the Commands.

Last Login Report - This report provides the date that a particular login was used and as such can keep track of current logins as well as those that haven’t been used in a while and should be deleted. It is not a running total of all logins like you would retrieve with “last” but instead lists each valid system account name once and the associated time that user last logged into the system.

Investigative Accounting

Here’s is where the real power of UNIX accounting lies. Daily reports are nice, but we want to be able to go back to any point in time and look at when John Doe logged into the system, when he logged out and what he did and commands invoked while he was logged in.

To that end, depending on the UNIX version you're using you can use commands like "acctcom" which is included with the standard UNIX (System V) Accounting package or "lastcomm" which is part of the accounting package included in the GNU distribution and also included with the Linux, HP and other Unix Accounting packages. There are a large number of options with each of these commands but using them for user account auditing seems to be most beneficial. A few ways you might do this with the "acctcom" command:

1) acctcom -s 15:00 -u *boss* -l *tty1*
This would retrieve all commands for user 'boss' invoked (listed chronologically) starting at 1500 but limited only to the 'tty1' terminal provided.

```
END AFTER:  Sun Dec 03     15:00:00 2000
COMMAND                                START    END          REAL    CPU       MEM
NAME          USER          TTYNAME    TIME     TIME         (SECS)  (SECS)
(KB
nroff         boss          tty1       14:58:19 15:04:57     398.27   356.83
vi            boss          tty1       15:05:25 15:10:52     328.07    43.93
pwd           boss          tty1       15:12:11 15:12:11       0.77     0.38
cat           boss          tty1       15:12:07 15:12:28      21.45     9.32
lp            boss          tty1       15:12:07 15:12:29      22.40     4.33
```

2) acctcom -s *time* -l *line* -C *seconds* -o *ofile*
This would provide a report in the file "ofile" listing all commands invoked from the specified TTY terminal but only after the time period provided

There are two dozen options to choose from with the "acctcom" command so there are many possibilities in producing the desired output. And, of course you can always use sed, awk, grep or any number of other utilities to customize output. Consult the documentation or manual page on your particular system in regard to options available.

Producing reports by user and Command usage can catch evil users:

You can see exactly what the user was doing from the time he logged in until the time he logged out and infer a good understanding as to whether they were doing legitimate work or engaged in unethical practices such as attempting to find and execute system "SUID" programs, run multiple programs that take most of CPU (resulting in a DOS) or whether in fact that particular user was even on site at the time the programs were executed. Years ago, on an older AT&T System V UNIX we caught a privileged user using 'fsdb' (file system debugger) to disrupt other user's files/directory structures. File corruption seemed to be mysterious until looking though the accounting records and seeing that the fsdb utility was being invoked at time of corruption.

Providing a report by user will not only list commands he explicitly invoked but also commands that were invoked *on his behalf* by the system. These will show up with a "?" under the TTY column of the output. An interesting side to this is that the login process will show all commands invoked by the system from the "login prompt" up to and including the user actually entering commands. The added benefit to this is that the auditor knows *exactly* what the correct sequence of system commands/responses are to achieve a normal login and can therefore easily distinguish any variations that might be indicative of a Trojan being installed and called during a login process.

There are some drawbacks:

Although UNIX Accounting does provide a nice running audit trail by user and command usage it does not provide the command options or files invoked by the user. For example, a user performs the "vi /etc/passwd" command but only "vi" will show up in the accounting files. Additionally, "shell built-in's", i.e. 'cd,' are not included in accounting files. For a full report of exactly all options invoked, one would have to implement a real auditing program obtained from the vendor (usually at charge) or from a third party.

Related Utilities

Two other very useful utilities include "ucomm and wcomm". These utilities are not vendor specific and can be obtained on the Internet and adapted for your system. They work by retrieving information in a distributed environment to create centralized accounting for multiple machines. The first "ucomm", shows the commands run by a user while "wcomm" shows a similar report but listed by command rather than user. Other commands include "sa" (summarizes accounting information), "ac" (summarize login accounting), "dump-utmp" and "dump-acct" (print a utmp file and print accounting files). These commands may or not be available in the accounting package provided with your system but are part of the accounting package with the GNU distribution.

Conclusion

UNIX Accounting packages provide a great deal of information not available through the default system logs. This information can be invaluable in either troubleshooting normal system problems or in determining if a system has been compromised. accounting information can be invaluable in providing a running log of user or system activity in documenting wrongdoing or creating evidence for prosecution.

Sources

The HP-UX Porting and Archive Center
URL "http://hpux.cae.wisc.edu/hppd/hpux/Sysadmin/acct-6.3.2/readme.html and http://hpux.cae.wisc.edu/hppd/hpux/Sysadmin/upacct-1.2/man.html"

Accounting Utilities Manual - lastcomm
URL "http://www.sakurachan.org/pub/GNU/Manuals/acct-6.3.2/html-chapter/accounting-5.html"

How to Enable Process Accounting on Linux
URL "http://ldp.iol.it/HOWTO/mini/Process-Accounting.html

SOLARIS Security, Performance and Accounting Administration - August 1994
by SunSoft

Unix System V System Administrators Guide, 1986 AT&T

Unix Administration Guide for System V, (Prentiss Hall), by
Rebecca Thomas and Rik Farrow