Options for User Auditing on Linux and Solaris Platforms

Cyril Plisko, CTO of Grigale Ltd.
GRIĜALE

January 2012
Executive Summary
A variety of methods exist for auditing user activity in UNIX and Linux environments. Some of them come preinstalled within common distributions, some can be downloaded as freeware, and some are commercially available products.

This whitepaper looks at the most popular methods for auditing. Each method is described, along with tips for how to make the best use of each method. In addition, guidance is provided to show what type of auditing each method is best suited for.

The auditing methods covered include:
- script
- screen
- sudo auditing
- Native audit facility (auditd, auditctl, etc.)
- ObserveIT

Scenario 1: Auditing using script(1)

How script is setup and started:
’script’ is a well-known tool used for basic auditing. In many installations, it comes preinstalled. When not, it can be added to any Linux or Solaris environment with no additional configuration is required.

Using script will start auditing as soon as a user types the command ‘script’.

What script records:
All the typing activity and screen I/O is recorded once the script command is given. Content is saved in a default file named ‘typescript’, or in additional files if desired.

Security and audit implications:
Using script is quick and easy, but script is easy to circumvent, and so is difficult to enforce reliably. For example, one can bypass it simply by deleting the output typescript file, or by exiting out of the script session.

Additionally, script only displays interactive activity. It does not detail what might be taking place underneath the hood in system calls or processes spawned by an interactive command.

When to use script:
The best and most common use for script is to record and audit your own sessions. This is due to the fact that it is cumbersome to configure it as a reliable security auditing tool for other users. Other methods described below are better suited for monitoring other users.
Scenario 2: Auditing Using screen(1)

How screen is setup and started:

‘scren’ is not provided with Linux or Solaris 10 by default, but can be downloaded and installed easily. As with sudo and script above, screen is not a dedicating auditing tool: Its original purpose is to be used as a multiplexing tool to connect various users to one session for collaboration. However, this very concept can be used for auditing as well.

Launching screen after login

Screen can be combined with SSH to force launching of screen immediately upon login, by creating an entry in authorized_keys: (Edit authorized_keys, determine whether there is any screen executable and associate it with the screen script.) In the example below, the script called ‘wrapper’ is used for this purpose.

What screen records:

In the example configuration above, whenever the user enters any command, the command is run through the ‘wrapper’ script as an invariant variable ‘SSH_ORIGINAL_COMMAND’, and is logged by screen.

Security and audit implications:

Screen allows live monitoring of user activity, without buffering. It allows you to connect to the session from another session, with the second window exactly duplicating the original. However, it must be used in read only mode otherwise the second window will reflect typing in the original window and the user will become aware of the auditing activity in real-time.

You can also disconnect from the session without terminating it, and return to it later.

The live monitoring and the disconnect/reconnect abilities make screen an improvement over script. But similar to script, screen does not record internals (spawned processes / commands). Only the interactive session is logged.

An SSH key can be used to login to the session to authenticate that screen is running.
When to use screen:

Screen is best for auditing purposes that require interactive collaboration. For example when the user knows that s/he is being monitored and the auditor provides feedback on actions being done within the session. Screen bridges a gap between a subset of security audit requirements and user training requirements.

Scenario 3: Auditing using sudo

How sudo auditing is setup and started:

As we all know, sudo is not an auditing tool by nature. Its core purpose is for enabling non-root users to run root commands. But sudo can be used for auditing as well, by configuring it to record the session and store it.

‘sudo’ comes preinstalled in Solaris 11 [sudo(1M)] and Linux [sudo(8)], and can be easily downloaded for installation in Solaris 10 via OpenCSW. (ex: `pkgadd -d http://get.opencsw.org/now` and `/opt/csw/bin/pkgutil -i screen sudo`)

What sudo can record:

sudo can record the entire interactive session by logging all the commands used, and can store the session info in a home directory or remote machine, for future viewing/auditing.

Security and audit implications:

The entire session can be replayed later in both normal and fast forward modes, using sudoreplay. This shows exactly what happened in the session and also shows who the specific user was. However, you cannot see the underlying system calls and you also do not get information about whether each command succeeded, unless there is explicit screen output that says so.

More importantly, sudo bash can be used to bypass sudo (or ‘fool’ the audit), so it is not entirely reliable.

When to use sudo:

This method is very effective for answer the question ‘Who is using / abusing sudo?’. In other words, if you have specific issue only surrounding root activity of non-root users, then sudo is effective. If you need a more holistic audit that also covers actions of normal users, you’ll need to merge multiple audit sources.
Scenario 4: Auditing using the native auditd audit facility

How auditd is setup and started:

The native audit facility is preconfigured in Linux [auditd(8)], but must be configured in Solaris [auditd(1M)]. (The Solaris setup of auditd is not rocket science, but does involve some level of complication.) Configuration of the audit facility can be done via the graphical application under X Windows, via the auditctl utility or directly via configuration file modification.

TIP

Audit facility configuration in X Windows

Once installed, it is easy to use and automatically starts with each user session in Solaris 11 or Linux. Solaris 10 systems will require the machine to be rebooted to go into auditing mode, and configuration will require kernel tweaking.

What auditd records:

The audit facility records data from the kernel, included the system calls, user ID and process ID. It captures all the low-level internals with a high level of granular detail. The information can be stored locally or sent to a central server. In Solaris 10, packets are chopped at 1024 bytes. In Solaris 11, the packets can be made larger by a plug-in so that information is not lost in chopping.

Because the audit facility is easy to turn on and runs automatically, you should avoid huge log files by implementing a log-file rotation policy from day 1.

TIP!

Example of information captured using the native audit facility

Security and Audit Implications:

Configuration allows you to determine what events or system calls should be recorded. These are recorded directly from the kernel and are highly-reliable. The events can be tagged so that they are easy to sort through, but the huge amount of data often makes it difficult to locate what you are looking for.
Unlike script and screen, auditd will capture the deep internals. However, it has the opposite problem because it does not capture screen I/O. The audit data is of a very technical nature, and does not necessarily state what the user was doing. (i.e. It will show actions spawned within a script, but does not show the script being run or the interactive command launched.)

When to use auditd:

The audit facility is best applied for monitoring of non-interactive activity such as file changes and system configuration changes.

Scenario 5: Auditing Using ObserveIT

How ObserveIT is setup and started:

ObserveIT is a commercial product that runs as an agent intercepting all interactive session activity. Once installed, it generates log data and delivers that content to an application server. It monitors all activity automatically, starting from the initial login of the user session.

What ObserveIT records:

ObserveIT collects interactive activity (text I/O) along with kernel data (spawned system calls / commands). ObserveIT provides a full audit review web console, which allows for video-style replay of user sessions as well as detailed audit drill-down into the system call kernel data. In the ObserveIT window, any user session can be viewed or replayed, and current sessions can be monitored in real-time.

In the example below, an interactive user ‘brad’ logged in and typed ‘rm –r do*’ to delete the documents directory.

ObserveIT captured the interactive command (rm –r do*) and also the internal commands (the actual filenames deleted, for example.)

Security and audit implications:

ObserveIT links the system calls to the actions in the interactive session, and therefore gives a more bulletproof audit review. ObserveIT generates the video and metadata logs, and also provides the platform for actually searching or navigating within the audit data. This platform includes automated reporting and alerts across any number of Linux or Solaris machines. (The audit database can also include non-Unix/Linux machines such as Windows servers or desktops, if necessary.) Beyond security audits, ObserveIT can also for used for troubleshooting.

When to use ObserveIT:

ObserveIT is best applied when ‘the whole picture’ is necessary.

This more complete picture is applicable for highly-regulated audit requirements where massive log data alone is insufficient. (ex: PCI security regulations, HIPAA, internal security regulations for highly-sensitive data applications, etc.)
Conclusion

As with most matters in Linux and Solaris, there are many ways to achieve user activity audit information. When choosing a desired method, it is important to weigh the various benefits that each method provides. For a simple view into highly-managed processes, `script` and `screen` can be extended to expose user actions. For issues specifically related to privileged use, `sudo` can be hacked to capture audit data. For configuration monitoring, the native `auditd` facility can capture clear file and system changes. ObserveIT provides a more holistic audit that generates logs for any user (root, sudo, named user) which include interactive I/O plus underlying system internals. This more complete picture is applicable for highly-regulated audit requirements where massive log data alone is insufficient. (ex: PCI security regulations, HIPAA, internal security regulations for highly-sensitive data applications, etc.)
About the Author

Cyril is co-founder of Grigale Ltd, a Solaris and Linux consulting firm. Cyril Plisko is a well-known UNIX expert, with vast system administration and kernel development background. He has rich experience in enterprise storage and network solution design as well as performance analysis and tuning for both small and large system configurations.

Cyril is a long-time member of the Solaris community and a technical leader of the Polaris project, as well as Bluetooth stack and Device Mapper projects. He is a Sun-Certified Instructor and is the proud carrier of OpenSolaris Contributor Agreement #OS0001.

During the rare times that he is not hacking, Cyril is most likely to be found at sea, at the helm of a yacht.

Cyril can be reached at cyril.plisko@grigale.com.

About ObserveIT

ObserveIT User Activity Monitoring software meets the complex compliance and security challenges related to user activity auditing, one of the key issues that IT, Security and Compliance officers are facing today.

ObserveIT acts like a security camera on your servers, generating audit logs and video recording of every action the user performs. ObserveIT captures all activity, even for applications that do not produce their own internal logs. Every action performed by remote vendors, developers, sysadmins and business users is tied to a video recording, providing bulletproof forensic evidence.

ObserveIT is the ideal solution for 3rd Party Vendor Monitoring, and PCI/HIPAA/SOX/ISO Compliance Accountability.

Founded in 2006, ObserveIT has a worldwide customer base of Global 2000 companies that spans many industry segments including finance, healthcare, manufacturing, telecom, government and IT services.

For more information, please contact ObserveIT at:
www.observeit.com
sales@observeit.com
US Phone: 1-800-687-0137
Int’l Phone: +972-3-648-0614