Audit framework

The Linux audit framework provides a CAPP-compliant (Controlled Access Protection Profiles) auditing system that reliably collects information about any security-relevant (or non-security-relevant) event on a system. It can help you track actions performed on a system.

Linux audit helps make your system more secure by providing you with a means to analyze what is happening on your system in great detail. It does not, however, provide additional security itself—it does not protect your system from code malfunctions or any kind of exploits. Instead, Audit is useful for tracking these issues and helps you take additional security measures, to prevent them.

The audit framework works by listening to the event reported by the kernel and logging them to a log file.

Note: as of linux 3.12, the audit framework is not yet compatible with the namespace implementation, if you use namespaces, do not use the audit framework.

Note: Depending on your configuration, it may affect the performance of the system.

Contents

Installation

Install a custom kernel with `CONFIG_AUDIT` enabled.

Install **`audit (https://www.archlinux.org/packages/?name=audit)`** and **start**/**enable** `auditd.service`.

Audit framework is composed of the auditd daemon, responsible for writing the audit messages that were generated through the audit kernel interface and triggered by application and system activity.

This daemon can be controlled by several commands and files:

- *auditctl* : to control the behavior of the daemon on the fly, adding rules etc.
- `/etc/audit/audit.rules` : contains the rules and various parameters of the auditd daemon
- *aureport* : generate report of the activity on a system
- *ausearch* : search for various events
- *auditspd* : the daemon which can be used to relay event notifications to other applications instead of writing them to disk in the audit log
- *autrace* : this command can be used to trace a process, in a similar way as strace.
- `/etc/audit/auditd.conf` : configuration file related to the logging.

Adding rules

Before adding rules, you must know that the audit framework can be very verbose and that each rules must be carefully tested before being effectively deployed. Indeed, just one rule can flood all your log within a few minutes.

Audit files and directories access

The most basic use of the audit framework is to log the access to the files you want. To do this, you must use a watch `-w` to a file or a directory The most basic rule to set up is to track accesses to the passwd file :

```
# auditctl -w /etc/passwd -p rwxa
```

You can track access to a folder with :

```
# auditctl -w /etc/security/
```

The first rule keeps track of every read `r` , write `w` , execution `x` , attribute change `a` to the file `/etc/passwd` . The second one keeps track of any access to the `/etc/security/` folder.

You can list all active rules with :

```
# auditctl -l
```

You can delete all rules with :

```
# auditctl -D
```

Once you validate the rules, you can append them to the `/etc/audit/audit.rules` file like that :

```
-w /etc/audit/audit.rules -p rwxa
-w /etc/security/
```

Audit syscalls

The audit framework allows you to audit the syscalls performed with the `-a` option.

A security related rule is to track the `chmod syscall`, to detect file ownership changes :

```
auditctl -a entry,always -S chmod
```

For a list of all syscalls: **syscalls(2) (https://jlk.fjfi.cvut.cz/arch/manpages/man/syscalls.2)**

A lot of rules and posibilities are available, see **auditctl(8) (https://jlk.fjfi.cvut.cz/arch/manpages/man/auditctl.8)** and **audit.rules(7) (https://jlk.fjfi.cvut.cz/arch/manpages/man/audit.rules.7)**.

Search the logs

The audit framework provides some tools to ease the use and the research of events happening on a system.

using pid

You can search events related to a particular pid using `ausearch`:

```
# ausearch -p 1
```

This command will show you all the events logged according to your rules related to PID 1 (i.e. systemd).

using keys

One of the great features of the audit framework is its hability to use `keys` to manage events, such a usage is recommended.

You can use the `-k` option in your rules to be able to find related events easily :

```
# auditctl -w /etc/passwd -p rwxa -k KEY_pwd
```

Then, if you search for events with the key `KEY_pwd`, ausearch will display only event related to the file `/etc/passwd`.

```
# ausearch -k KEY_pwd
```

Look for abnormalies

The `aureport` tool can be used to quickly report any anormal event performed on the system, it includes network interfaces used in promiscous mode, process or thread crashing or exiting with ENOMEM error etc.

The easiest way to use `aureport` is :

```
# aureport -n
```

aureport can be used to generate custom reports, see **aureport(8) (https://jlk.fjfi.cvut.cz/arch/manpages/man/aureport.8)**.

Which files or syscalls are worth-auditing ?

Keep in mind that each audit rule added will generate logs, so you must be ready to treat this amount of information. Basically, each security-related event/file must be monitored, like ids, ips, anti-rootkits etc. On the other side, it's totally useless to track every write syscall, the smallest compilation will fill your logs with this event.

More complex set of rules can be set up, performing auditing on a very fine-grained base. If you want to do so, see **auditctl(8) (https://jlk.fjfi.cvut.cz/arch/manpages/man/auditctl.8)**.

Gather logs from different hosts

The audit framework has a plugin system which provides the possibility to send local logfiles to a remote auditd.

Send logfiles

To send your logfiles to a remote host you need the `audisp-remote` plugin which comes automatically with the **audit (https://www.archlinux.org/packages/?name=audit)** package. Activate the plugin:

```
/etc/audisp/plugins.d/au-remote.conf
```

```
active = yes
direction = out
path = /usr/bin/audisp-remote
type = always
format = string
```

and set the remote host where the logs should be send to:

```
/etc/audisp/audisp-remote.conf
```

```
remote_server = domain.name.or.ip
port = 60
##local_port = optional
transport = tcp
```

Receive logfiles

To make audit listen for remote audispds you just need to set the tcp options:

/etc/audit/auditd.conf

```
tcp_listen_port = 60
tcp_listen_queue = 5
tcp_max_per_addr = 1
##tcp_client_ports = 1024-65535 #optional
tcp_client_max_idle = 0
```

Now you can view the logs of **all** configured hosts in the logfiles of the receiving auditd.

Retrieved from "https://wiki.archlinux.org/index.php?title=Audit_framework&oldid=489758"

- This page was last edited on 9 September 2017, at 19:44.
- Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.