

NAME

auditd – The Linux Audit daemon

SYNOPSIS

auditd [-f] [-l] [-n] [-s **disable|enable|nochange**] [-c <config_dir>]

DESCRIPTION

auditd is the userspace component to the Linux Auditing System. It's responsible for writing audit records to the disk. Viewing the logs is done with the **ausearch** or **aureport** utilities. Configuring the audit system or loading rules is done with the **auditctl** utility. During startup, the rules in */etc/audit/audit.rules* are read by **auditctl** and loaded into the kernel. Alternately, there is also an **augenrules** program that reads rules located in */etc/audit/rules.d/* and compiles them into an *audit.rules* file. The audit daemon itself has some configuration options that the admin may wish to customize. They are found in the **auditd.conf** file.

OPTIONS

- f** leave the audit daemon in the foreground for debugging. Messages also go to stderr rather than the audit log.
- l** allow the audit daemon to follow symlinks for config files.
- n** no fork. This is useful for running off of inittab or systemd.
- s=ENABLE_STATE**
specify when starting if auditd should change the current value for the kernel enabled flag. Valid values for *ENABLE_STATE* are "disable", "enable" or "nochange". The default is to enable (and disable when auditd terminates). The value of the enabled flag may be changed during the lifetime of auditd using 'auditctl -e'.
- c** Specify alternate config file directory. Note that this same directory will be passed to the dispatcher. (default: */etc/audit/*)

SIGNALS**SIGHUP**

causes auditd to reconfigure. This means that auditd re-reads the configuration file. If there are no syntax errors, it will proceed to implement the requested changes. If the reconfigure is successful, a *DAEMON_CONFIG* event is recorded in the logs. If not successful, error handling is controlled by *space_left_action*, *admin_space_left_action*, *disk_full_action*, and *disk_error_action* parameters in *auditd.conf*.

SIGTERM

caused auditd to discontinue processing audit events, write a shutdown audit event, and exit.

SIGUSR1

causes auditd to immediately rotate the logs. It will consult the *max_log_file_action* to see if it should keep the logs or not.

SIGUSR2

causes auditd to attempt to resume logging and passing events to plugins. This is usually needed after logging has been suspended or the internal queue is overflowed. Either of these conditions depends on the applicable configuration settings.

SIGCONT

causes auditd to dump a report of internal state to */run/audit/auditd.state*.

EXIT CODES

- 1** Cannot adjust priority, daemonize, open audit netlink, write the pid file, start up plugins, resolve the machine name, set audit pid, or other initialization tasks.

- 2 Invalid or excessive command line arguments
- 4 The audit daemon doesn't have sufficient privilege
- 6 There is an error in the configuration file

FILES

/etc/audit/auditd.conf - configuration file for audit daemon

/etc/audit/audit.rules - audit rules to be loaded at startup

/etc/audit/rules.d/ - directory holding individual sets of rules to be compiled into one file by augenrules.

/etc/audit/plugins.d/ - directory holding individual plugin configuration files.

/etc/audit/audit-stop.rules - These rules are loaded when the audit daemon stops.

/run/audit/auditd.state - report about internal state.

NOTES

A boot param of audit=1 should be added to ensure that all processes that run before the audit daemon starts is marked as auditable by the kernel. Not doing that will make a few processes impossible to properly audit.

The audit daemon can receive audit events from other audit daemons via the audisp-remote plugin. The audit daemon may be linked with tcp_wrappers to control which machines can connect. If this is the case, you can add an entry to hosts.allow and deny.

SEE ALSO

auditd.conf(5), **auditd-plugins(5)**, **ausearch(8)**, **aureport(8)**, **auditctl(8)**, **augenrules(8)**, **audit.rules(7)**.

AUTHOR

Steve Grubb