## NAME

**audit_control** - audit system parameters

## DESCRIPTION

The **audit_control** file contains several audit system parameters. Each line of this file is of the form:

*parameter*:*value*

The parameters are:

*dir* The directory where audit log files are stored. There may be more than one of these entries. Changes to this entry can only be enacted by restarting the audit system. See audit(8) for a description of how to restart the audit system.

*dist* When set to *on* or *yes*, auditd(8) will be creating hardlinks to all trail files in */var/audit/dist* directory. Those hardlinks will be consumed by the auditdistd(8) daemon.

*flags* Specifies which audit event classes are audited for all users. audit_user(5) describes how to audit events for individual users. See the information below for the format of the audit flags.

*host* Specify the hostname or IP address to be used when setting the local systems's audit host information. This hostname will be converted into an IP or IPv6 address and will be included in the header of each audit record. Due to the possibility of transient errors coupled with the security issues in the DNS protocol itself, the use of DNS should be avoided. Instead, it is strongly recommended that the hostname be specified in the /etc/hosts file. For more information see hosts(5).

*naflags*
Contains the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user.

*minfree*
The minimum free space required on the file system audit logs are being written to. When the free space falls below this limit a warning will be issued. If no value for the minimum free space is set, the default of 20 percent is applied by the kernel.

*policy* A list of global audit policy flags specifying various behaviors, such as fail stop, auditing of paths and arguments, etc.

*filesz* Maximum trail size in bytes; if set to a non-0 value, the audit daemon will rotate the audit trail

file at around this size. Sizes less than the minimum trail size (default of 512K) will be rejected as invalid. If 0, trail files will not be automatically rotated based on file size. For convenience, the trail size may be expressed with suffix letters: B (Bytes), K (Kilobytes), M (Megabytes), or G (Gigabytes). For example, 2M is the same as 2097152.

*expire-after*
Specifies when audit log files will expire and be removed. This may be after a time period has passed since the file was last written to or when the aggregate of all the trail files have reached a specified size or a combination of both. If no expire-after parameter is given then audit log files will not expire and be removed by the audit control system. See the information below for the format of the expiration specification.

*qsize* Specifies the maximum number of outstanding committed audit records that can be in the kernel's post-commit queue pending write to disk. If this number has been reached, user threads performing an auditable event will be suspended until the queue has fallen below the limit. Depending on the underlying kernel implementation, the number of in-flight records can exceed this number, as it does not constrain uncommitted records (e.g., those associated with incomplete auditable system calls), and may also exclude the set of records extracted from the queue and currently being prepared for or undergoing I/O. Other operational limits may be affected by this parameter, such as the minimum free space on disk required to continue system operation, estimated as the maximum number of allowable in-flight records multiplied by the maximum audit record size.

## AUDIT FLAGS

Audit flags are a comma-delimited list of audit classes as defined in the audit_class(5) file. Event classes may be preceded by a prefix which changes their interpretation. The following prefixes may be used for each class:

(none)
Record both successful and failed events.
+ Record successful events.
- Record failed events.
^ Record neither successful nor failed events.
^+ Do not record successful events.
^- Do not record failed events.

## AUDIT POLICY FLAGS

The policy flags field is a comma-delimited list of policy flags from the following list:

**cnt** Allow processes to continue running even though events are not being audited. If not

set, processes will be suspended when the audit store space is exhausted. Currently, this is not a recoverable state.

**ahlt** Fail stop the system if unable to audit an event--this consists of first draining pending records to disk, and then halting the operating system.

**argv** Audit command line arguments to execve(2).

**arge** Audit environmental variable arguments to execve(2).

**seq** Include a unique audit sequence number token in generated audit records (not implemented on FreeBSD or Darwin).

**group** Include supplementary groups list in generated audit records (not implemented on FreeBSD or Darwin; supplementary groups are never included in records on these systems).

**trail** Append a trailer token to each audit record (not implemented on FreeBSD or Darwin; trailers are always included in records on these systems).

**path** Include secondary file paths in audit records (not implemented on FreeBSD or Darwin; secondary paths are never included in records on these systems).

**zonename** Include a zone ID token with each audit record (not implemented on FreeBSD or Darwin; FreeBSD audit records do not currently include the jail ID or name).

**perzone** Enable auditing for each local zone (not implemented on FreeBSD or Darwin; on FreeBSD, audit records are collected from all jails and placed in a single global trail, and only limited audit controls are permitted within a jail).

It is recommended that installations set the **cnt** flag but not **ahlt** flag unless it is intended that audit logs exceeding available disk space halt the system.

## AUDIT LOG EXPIRATION SPECIFICATION

The expiration specification can be one value or two values with the logical conjunction of AND/OR between them. Values for the audit log file age are numbers with the following suffixes:

s Log file age in seconds.
h Log file age in hours.
d Log file age in days.
y Log file age in years.

Values for the disk space used are numbers with the following suffixes:

(space) or
B Disk space used in Bytes.
K Disk space used in Kilobytes.
M Disk space used in Megabytes.
G Disk space used in Gigabytes.

The suffixes on the values are case sensitive. If both an age and disk space value are used they are separated by AND or OR and both values are used to determine when audit log files expire. In the case of AND, both the age and disk space conditions must be met before the log file is removed. In the case of OR, either condition may expire the log file. For example:

```
expire-after: 60d AND 1G
```

will expire files that are older than 60 days but only if 1 gigabyte of disk space total is being used by the audit logs.

## DEFAULT

The following settings appear in the default **audit_control** file:

```
dir:/var/audit
flags:lo,aa
minfree:5
naflags:lo,aa
policy:cnt,argv
filesz:2M
expire-after:10M
```

The *flags* parameter above specifies the system-wide mask corresponding to login/logout as well as authentication and authorization events. The *policy* parameter specifies that the system should neither fail stop nor suspend processes when the audit store fills and that command line arguments should be audited for AUE_EXECVE events. The trail file will be automatically rotated by the audit daemon when the file size reaches approximately 2MB. Trail files will expire when their aggregate size exceeds 10MB.

## FILES

*/etc/security/audit_control*

## SEE ALSO

auditon(2), audit(4), audit_class(5), audit_event(5), audit_user(5), audit(8), auditd(8)

## HISTORY

The OpenBSM implementation was created by McAfee Research, the security division of McAfee Inc., under contract to Apple Computer Inc. in 2004. It was subsequently adopted by the TrustedBSD Project as the foundation for the OpenBSM distribution.

## AUTHORS

This software was created by McAfee Research, the security research division of McAfee, Inc., under contract to Apple Computer Inc. Additional authors include Wayne Salamon, Robert Watson, and SPARTA Inc.

The Basic Security Module (BSM) interface to audit records and audit event stream format were defined by Sun Microsystems.