

NAME

ausearch – a tool to query audit daemon logs

SYNOPSIS

ausearch [*options*]

DESCRIPTION

ausearch is a tool that can query the audit daemon logs for events based on different search criteria. The ausearch utility can also take input from stdin as long as the input is the raw log data. Each commandline option given forms an "and" statement. For example, searching with **-m** and **-ui** means return events that have both the requested type and match the user id given. An exception is the **-m** and **-n** options; multiple record types and nodes are allowed in a search which will return any matching node and record.

It should also be noted that each syscall excursion from user space into the kernel and back into user space has one event ID that is unique. Any auditable event that is triggered during this trip share this ID so that they may be correlated.

Different parts of the kernel may add supplemental records. For example, an audit event on the syscall "open" will also cause the kernel to emit a PATH record with the file name. The ausearch utility will present all records that make up one event together. This could mean that even though you search for a specific kind of record, the resulting events may contain SYSCALL records.

Also be aware that not all record types have the requested information. For example, a PATH record does not have a hostname or a loginuid.

OPTIONS

-a, --event *audit-event-id*

Search for an event based on the given *event ID*. Messages always start with something like msg=audit(1116360555.329:2401771). The event ID is the number after the ':'. All audit events that are recorded from one application's syscall have the same audit event ID. A second syscall made by the same application will have a different event ID. This way they are unique.

--arch *CPU*

Search for events based on a specific CPU architecture. If you do not know the arch of your machine but you want to use the 32 bit syscall table and your machine supports 32 bits, you can also use **b32** for the arch. The same applies to the 64 bit syscall table, you can use **b64**. The arch of your machine can be found by doing 'uname -m'.

-c, --comm *comm-name*

Search for an event based on the given *comm name*. The comm name is the executable's name from the task structure.

--debug

Write malformed events that are skipped to stderr.

--checkpoint *checkpoint-file*

Checkpoint the output between successive invocations of ausearch such that only events not previously output will print in subsequent invocations.

An auditd event is made up of one or more records. When processing events, ausearch defines events as either complete or in-complete. A complete event is either a single record event or one whose event time occurred 2 seconds in the past compared to the event being currently processed.

A checkpoint is achieved by recording the last completed event output along with the device number and inode of the file the last completed event appeared in *checkpoint-file*. On a subsequent invocation, ausearch will load this checkpoint data and as it processes the log files, it will discard all complete events until it matches the checkpointed one. At this point, it will start outputting complete events.

Should the file or the last checkpointed event not be found, one of a number of errors will result and ausearch will terminate. See **EXIT STATUS** for detail.

--eoe-timeout *seconds*

Set the end of event parsing timeout. See **end_of_event_timeout** in *auditd.conf(5)* for details. Note that setting this value will override any configured value found in */etc/auditd/auditd.conf*.

-e, --exit *exit-code-or-errno*

Search for an event based on the given syscall *exit code or errno*.

--escape *option*

This option determines if the output is escaped to make the content safer for certain uses. The options are *raw* , *tty* , *shell* , and *shell_quote*. Each mode includes the characters of the preceding mode and escapes more characters. That is to say *shell* includes all characters escaped by *tty* and adds more. *tty* is the default.

--extra-keys

When the *format* mode is *csv*, this option will add a final column with key information if its exists for the event. This would only occur on SYSCALL records which were the result of triggering an audit rule that defines a key.

--extra-labels

When the *format* mode is *csv*, this option will add columns of information about subject and object labels when they exist.

--extra-obj2

When the *format* mode is *csv*, this option will add columns of information about a second object when it exists. It's rare that a second object is part of a record. Some examples are when a file is renamed from one name to another or when a device is mounted to a path.

--extra-time

When the *format* mode is *csv*, this option will add columns of information about broken down time to make subsetting easier.

-f, --file *file-name*

Search for an event based on the given *filename*. The argument will match normal files as well as *af_unix* sockets.

--format *option*

Events that match the search criteria are formatted using this option. The supported formats are: *raw*, *default*, *interpret*, *csv*, and *text*. The *raw* option is described under the **--raw** command line option. The *default* option is what you get when no formatting options are passed. It includes one line as a visual separator which indicates the time stamp and then the records of the event follow. The *interpret* option is explained under the **-i** command line option. The *csv* option outputs the results of the search as a normalized event in comma separated value (CSV) format suitable for import into analytical programs. The *text* option turns the event into an English sentence that is easier to understand than other options, but it comes at the expense of loss of detail. In most cases this is perfectly fine since the original event still retains all the original information.

-ga, --gid-all *all-group-id*

Search for an event with either effective group ID or group ID matching the given *group ID*.

-ge, --gid-effective *effective-group-id*

Search for an event with the given *effective group ID* or group name.

-gi, --gid *group-id*

Search for an event with the given *group ID* or group name.

-h, --help

Help

-hn, --host *host-name*

Search for an event with the given *host name*. The hostname can be either a hostname, fully qualified domain name, or numeric network address. No attempt is made to resolve numeric addresses to domain names or aliases. This search typically correlates to the *addr* or *host* field of audit events. Also see the `--node` command which searches the *node* field.

-i, --interpret

Interpret numeric entities into text. For example, *uid* is converted to account name. If the audit logs are unenriched, the conversion is done using the current resources of the machine where the search is being run. If you have renamed the accounts, or don't have the same accounts on your machine, you could get misleading results. If the logs are enriched, it uses the supplemental data to do the conversion. This allows accurate log reporting even when run on a different machine than the original logs came from.

-if, --input *file-name* | *directory*

Use the given *file* or *directory* instead of the logs. This is to aid analysis where the logs have been moved to another machine or only part of a log was saved. The path length is limited to 4064 bytes.

--input-logs

Use the log file location from *auditd.conf* as input for searching. This is needed if you are using *ausearch* from a cron job.

--just-one

Stop after emitting the first event that matches the search criteria.

-k, --key *key-string*

Search for an event based on the given *key string*.

-l, --line-buffered

Flush output on every line. Most useful when *stdout* is connected to a pipe and the default block buffering strategy is undesirable. May impose a performance penalty.

-m, --message *message-type* | *comma-sep-message-type-list*

Search for an event matching the given *message type*. (Message types are also known as record types.) You may also enter a *comma separated list of message types* or multiple individual message types each with its own *-m* option. There is an **ALL** message type that doesn't exist in the actual logs. It allows you to get all messages in the system. The list of valid messages types is long. The program will display the list whenever no message type is passed with this parameter. The message type can be either text or numeric. If you enter a list, there can be only commas and no spaces separating the list.

-n, --node *node-name*

Search for events originating from a specific machine. Multiple nodes are allowed, and if any nodes match, the event is matched. This search uses the *node* field in audit events. Also see the `--host` command which search for events related to host information in the audit trail.

-o, --object *SE-Linux-context-string*

Search for event with *tcontext* (object) matching the string.

-p, --pid *process-id*

Search for an event matching the given *process ID*.

-pp, --ppid *parent-process-id*

Search for an event matching the given *parent process ID*.

-r, --raw

Output is completely unformatted. This is useful for extracting records to a file that can still be interpreted by audit tools or when piping to other audit tools.

- sc, --syscall** *syscall-name-or-value*
Search for an event matching the given *syscall*. You may either give the numeric syscall value or the syscall name. If you give the syscall name, it will use the syscall table for the machine that you are using.
- se, --context** *SE-Linux-context-string*
Search for events with either *scontext*/subject or *tcontext*/object matching the given string.
- session** *Login-Session-ID*
Search for events matching the given Login Session ID. This process attribute is set when a user logs in and can tie any process to a particular user login.
- su, --subject** *SE-Linux-context-string*
Search for event with *scontext* (subject) matching the string.
- sv, --success** *success-value*
Search for an event matching the given *success value*. Legal values are **yes** and **no**.
- te, --end** [*end-date*] [*end-time*]
Search for events with time stamps equal to or before the given end time. The format of end time depends on your locale. You can check the format of your locale by running **date '+%x'**. If the date is omitted, **today** is assumed. If the time is omitted, **now** is assumed. Use 24 hour clock time rather than AM or PM to specify time. An example date using the en_US.utf8 locale is 09/03/2009. An example of time is 18:00:00. The date format accepted is influenced by the LC_TIME environmental variable.

You may also use the word: **now**, **recent**, **this-hour**, **boot**, **today**, **yesterday**, **this-week**, **week-ago**, **this-month**, or **this-year**. **Now** means starting now. **Recent** is 10 minutes ago. **Boot** means the time of day to the second when the system last booted. **Today** means now. **Yesterday** is 1 second after midnight the previous day. **This-week** means starting 1 second after midnight on day 0 of the week determined by your locale (see **localtime**). **Week-ago** means 1 second after midnight exactly 7 days ago. **This-month** means 1 second after midnight on day 1 of the month. **This-year** means the 1 second after midnight on the first day of the first month.
- ts, --start** [*start-date*] [*start-time*]
Search for events with time stamps equal to or after the given start time. The format of start time depends on your locale. You can check the format of your locale by running **date '+%x'**. If the date is omitted, **today** is assumed. If the time is omitted, **midnight** is assumed. Use 24 hour clock time rather than AM or PM to specify time. An example date using the en_US.utf8 locale is 09/03/2009. An example of time is 18:00:00. The date format accepted is influenced by the LC_TIME environmental variable.

You may also use the word: **now**, **recent**, **this-hour**, **boot**, **today**, **yesterday**, **this-week**, **week-ago**, **this-month**, **this-year**, or **checkpoint**. **Boot** means the time of day to the second when the system last booted. **Today** means starting at 1 second after midnight. **Recent** is 10 minutes ago. **Yesterday** is 1 second after midnight the previous day. **This-week** means starting 1 second after midnight on day 0 of the week determined by your locale (see **localtime**). **Week-ago** means starting 1 second after midnight exactly 7 days ago. **This-month** means 1 second after midnight on day 1 of the month. **This-year** means the 1 second after midnight on the first day of the first month.

checkpoint means *ausearch* will use the timestamp found within a valid checkpoint file ignoring the recorded inode, device, serial, node and event type also found within a checkpoint file. Essentially, this is the recovery action should an invocation of *ausearch* with a checkpoint option fail with an exit status of 10, 11 or 12. It could be used in a shell script something like:

```
ausearch --checkpoint /etc/audit/auditd_checkpoint.txt -i
_au_status=$?
```

```

        if test ${_au_status} -eq 10 -o ${_au_status} -eq 11 -o ${_au_status} -eq 12
        then
            ausearch --checkpoint /etc/audit/auditd_checkpoint.txt --start checkpoint -i
        fi

```

-tm, --terminal *terminal*

Search for an event matching the given *terminal* value. Some daemons such as cron and atd use the daemon name for the terminal.

-ua, --uid-all *all-user-id*

Search for an event with either user ID, effective user ID, or login user ID (audit) matching the given *user ID*.

-ue, --uid-effective *effective-user-id*

Search for an event with the given *effective user ID*.

-ui, --uid *user-id*

Search for an event with the given *user ID*.

-ul, --loginuid *login-id*

Search for an event with the given *login user ID*. All entry point programs that are PAMified need to be configured with pam_loginuid required for the session for searching on loginuid (audit) to be accurate.

-uu, --uid *guest-uuid*

Search for an event with the given *guest UUID*.

-v, --version

Print the version and exit

-vm, --vm-name *guest-name*

Search for an event with the given *guest name*.

-w, --word

String based matches must match the whole word. This category of matches include: filename, hostname, terminal, keys, and SE Linux context.

-x, --executable *executable*

Search for an event matching the given *executable* name.

EXIT STATUS

- 0 if OK,
- 1 if nothing found, or argument errors or minor file access/read errors,
- 10 invalid checkpoint data found in checkpoint file,
- 11 checkpoint processing error
- 12 checkpoint event not found in matching log file

NOTE

The boot time option is a convenience function and has limitations. The time it calculates is based on time now minus /proc/uptime. If after boot the system clock has been adjusted, perhaps by ntp, then the calculation may be wrong. In that case you'll need to fully specify the time. You can check the time it would use by running:

```
date -d "'cut -f1 -d. /proc/uptime' seconds ago"
```

EXAMPLES

Search for a specific user:

```
# ausearch --start today --loginuid john -i
```

Check the SELinux log for any denials today

```
# ausearch --start today -m avc -i
```

Output any recent SELinux log

```
# ausearch -m avc,user_avc,selinux_err,user_selinux_err -i -ts recent
```

Output logs in text format

```
# ausearch --start today --format text
```

Output TTY events interpreted and shell escaped

```
# ausearch --start today -m TTY -i --escape shell_quote
```

SEE ALSO

auditd(8), **auditd.conf(5)**, **aureport(8)**, **pam_loginuid(8)**.