

**NAME**

**sudoers\_timestamp** - Sudoers Time Stamp Format

**DESCRIPTION**

The **sudoers** plugin uses per-user-ID time stamp files for credential caching. Once a user has been authenticated, they may use **sudo** without a password for a short period of time (5 minutes unless overridden by the *timestamp\_timeout* option). By default, **sudoers** uses a separate record for each terminal, which means that a user's login sessions are authenticated separately. The *timestamp\_type* option can be used to select the type of time stamp record **sudoers** will use.

A multi-record time stamp file format was introduced in **sudo** 1.8.10 that uses a single file per user. Previously, a separate file was used for each user and terminal combination unless tty-based time stamps were disabled. The new format is extensible and records of multiple types and versions may coexist within the same file.

All records, regardless of type or version, begin with a 16-bit version number and a 16-bit record size.

Time stamp records have the following structure:

```
/* Time stamp entry types */
#define TS_GLOBAL      0x01U /* not restricted by tty or ppid */
#define TS_TTY        0x02U /* restricted by tty */
#define TS_PPID       0x03U /* restricted by ppid */
#define TS_LOCKEXCL   0x04U /* special lock record */

/* Time stamp flags */
#define TS_DISABLED   0x01U /* entry disabled */
#define TS_ANYUID     0x02U /* ignore uid, only valid in key */

struct timestamp_entry {
    unsigned short version; /* version number */
    unsigned short size; /* entry size */
    unsigned short type; /* TS_GLOBAL, TS_TTY, TS_PPID */
    unsigned short flags; /* TS_DISABLED, TS_ANYUID */
    uid_t auth_uid; /* uid to authenticate as */
    pid_t sid; /* session ID associated with tty/ppid */
    struct timespec start_time; /* session/ppid start time */
    struct timespec ts; /* time stamp (CLOCK_MONOTONIC) */
    union {
        dev_t ttydev; /* tty device number */
    };
};
```

```

    pid_t ppid;      /* parent pid */
} u;
};

```

The `timestamp_entry` struct fields are as follows:

#### version

The version number of the `timestamp_entry` struct. New entries are created with a version number of 2. Records with different version numbers may coexist in the same file but are not interoperable.

**size** The size of the record in bytes.

**type** The record type, currently `TS_GLOBAL`, `TS_TTY`, or `TS_PPID`.

**flags** Zero or more record flags which can be bit-wise ORed together. Supported flags are `TS_DISABLED`, for records disabled via **sudo -k** and `TS_ANYUID`, which is used only when matching records.

#### auth\_uid

The user-ID that was used for authentication. Depending on the value of the `rootpw`, `runaspw` and `targetpw` options, the user-ID may be that of the invoking user, the root user, the default `runas` user or the target user.

**sid** The ID of the user's terminal session, if present. The session ID is only used when matching records of type `TS_TTY`.

#### start\_time

The start time of the session leader for records of type `TS_TTY` or of the parent process for records of type `TS_PPID`. The `start_time` is used to help prevent reuse of a time stamp record after a user has logged out. Not all systems support a method to easily retrieve a process's start time. The `start_time` field was added in **sudoers** version 1.8.22 for the second revision of the `timestamp_entry` struct.

**ts** The actual time stamp. A monotonic time source (which does not move backward) is used if the system supports it. Where possible, **sudoers** uses a monotonic timer that increments even while the system is suspended. The value of `ts` is updated each time a command is run via **sudo**. If the difference between `ts` and the current time is less than the value of the `timestamp_timeout` option, no password is required.

u.ttydev

The device number of the terminal associated with the session for records of type TS\_TTY.

u.ppid

The ID of the parent process for records of type TS\_PPID.

The **tsdump** utility, included with the sudo source distribution, can be used to display the contents of a *sudoers* time stamp file.

## LOCKING

In **sudoers** versions 1.8.10 through 1.8.14, the entire time stamp file was locked for exclusive access when reading or writing to the file. Starting in **sudoers** 1.8.15, individual records are locked in the time stamp file instead of the entire file and the lock is held for a longer period of time. This scheme is described below.

The first record in the time stamp file is of type TS\_LOCKEXCL and is used as a *lock* record to prevent more than one **sudo** process from adding a new record at the same time. Once the desired time stamp record has been located or created (and locked), the TS\_LOCKEXCL record is unlocked. The lock on the individual time stamp record, however, is held until authentication is complete. This allows **sudoers** to avoid prompting for a password multiple times when it is used more than once in a pipeline.

Records of type TS\_GLOBAL cannot be locked for a long period of time since doing so would interfere with other **sudo** processes. Instead, a separate lock record is used to prevent multiple **sudo** processes using the same terminal (or parent process ID) from prompting for a password at the same time.

## SEE ALSO

sudoers(5), sudo(8)

## HISTORY

Originally, **sudo** used a single zero-length file per user and the file's modification time was used as the time stamp. Later versions of **sudo** added restrictions on the ownership of the time stamp files and directory as well as checks on the validity of the time stamp itself. Notable changes were introduced in the following **sudo** versions:

- 1.4.0 Support for tty-based time stamp file was added by appending the terminal name to the time stamp file name.
- 1.6.2 The time stamp file was replaced by a per-user directory which contained any tty-based time stamp files.

### 1.6.3p2

The target user name was added to the time stamp file name when the *targetpw* option was set.

### 1.7.3

Information about the terminal device was stored in tty-based time stamp files for validity checks. This included the terminal device numbers, inode number and, on systems where it was not updated when the device was written to, the inode change time. This helped prevent reuse of the time stamp file after logout.

### 1.8.6p7

The terminal session ID was added to tty-based time stamp files to prevent reuse of the time stamp by the same user in a different terminal session. It also helped prevent reuse of the time stamp file on systems where the terminal device's inode change time was updated by writing.

### 1.8.10

A new, multi-record time stamp file format was introduced that uses a single file per user. The terminal device's change time was not included since most systems now update the change time after a write is performed as required by POSIX.

### 1.8.15

Individual records are locked in the time stamp file instead of the entire file and the lock is held until authentication is complete.

### 1.8.22

The start time of the terminal session leader or parent process is now stored in non-global time stamp records. This prevents reuse of the time stamp file after logout in most cases.

Support was added for the kernel-based tty time stamps available in OpenBSD which do not use an on-disk time stamp file.

### 1.9.15

Time stamp file path names are now based on the invoking user-ID instead of the user name. This avoids problems with user names that include a path separator character.

## AUTHORS

Many people have worked on **sudo** over the years; this version consists of code written primarily by:

Todd C. Miller

See the CONTRIBUTORS.md file in the **sudo** distribution (<https://www.sudo.ws/about/contributors/>) for an exhaustive list of people who have contributed to **sudo**.

## BUGS

If you believe you have found a bug in **sudoers\_timestamp**, you can either file a bug report in the sudo bug database, <https://bugzilla.sudo.ws/>, or open an issue at <https://github.com/sudo-project/sudo/issues>. If you would prefer to use email, messages may be sent to the sudo-workers mailing list, <https://www.sudo.ws/mailman/listinfo/sudo-workers> (public) or [sudo@sudo.ws](mailto:sudo@sudo.ws) (private).

Please not report security vulnerabilities through public GitHub issues, Bugzilla or mailing lists. Instead, report them via email to [Todd.Miller@sudo.ws](mailto:Todd.Miller@sudo.ws). You may encrypt your message with PGP if you would like, using the key found at <https://www.sudo.ws/dist/PGPKEYS>.

## SUPPORT

Limited free support is available via the sudo-users mailing list, see <https://www.sudo.ws/mailman/listinfo/sudo-users> to subscribe or search the archives.

## DISCLAIMER

**sudo** is provided "AS IS" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. See the LICENSE.md file distributed with **sudo** or <https://www.sudo.ws/about/license/> for complete details.