Abuse Elevation Control Mechanism: Sudo and Sudo Caching, Sub-technique T1548.003 - Enterprise

Archived: 2026-04-02 10:40:10 UTC

Adversaries may perform sudo caching and/or use the sudoers file to elevate privileges. Adversaries may do this to execute commands as other users or spawn processes with higher privileges.

Within Linux and MacOS systems, sudo (sometimes referred to as "superuser do") allows users to perform commands from terminals with elevated privileges and to control who can perform these commands on the system. The `sudo` command "allows a system administrator to delegate authority to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while providing an audit trail of the commands and their arguments."[1] Since sudo was made for the system administrator, it has some useful configuration features such as a `timestamp_timeout`, which is the amount of time in minutes between instances of `sudo` before it will re-prompt for a password. This is because `sudo` has the ability to cache credentials for a period of time. Sudo creates (or touches) a file at `/var/db/sudo` with a timestamp of when sudo was last run to determine this timeout. Additionally, there is a `tty_tickets` variable that treats each new tty (terminal session) in isolation. This means that, for example, the sudo timeout of one tty will not affect another tty (you will have to type the password again).

The sudoers file, `/etc/sudoers`, describes which users can run which commands and from which terminals. This also describes which commands users can run as other users or groups. This provides the principle of least privilege such that users are running in their lowest possible permissions for most of the time and only elevate to other users or permissions as needed, typically by prompting for a password. However, the sudoers file can also specify when to not prompt users for passwords with a line like `user1 ALL=(ALL) NOPASSWD: ALL`.[2] Elevated privileges are required to edit this file though.

Adversaries can also abuse poor configurations of these mechanisms to escalate privileges without needing the user's password. For example, `/var/db/sudo`'s timestamp can be monitored to see if it falls within the `timestamp_timeout` range. If it does, then malware can execute sudo commands without needing to supply the user's password. Additional, if `tty_tickets` is disabled, adversaries can do this from any tty for that user.

In the wild, malware has disabled `tty_tickets` to potentially make scripting easier by issuing `echo \'Defaults !tty_tickets\' >> /etc/sudoers`.[3] In order for this change to be reflected, the malware also issued `killall Terminal`. As of macOS Sierra, the sudoers file has `tty_tickets` enabled by default.
