

Linux - Permission Vulnerabilities

suid programs

💡 The following 'C' program prints out the "effective user name" of the running process as opposed to the user who invoked that process. So if bob runs this program, bob would be printed out. If it is run as sudo, root would be printed out. If the suid bit is set, the owner of the file will be the effective user. So if root owns the file and bob runs it, the effective user will be root.

```
#include <stdlib.h>
#include <pwd.h>
#include <stdio.h>
#include <unistd.h>

/*
SEC335 Illustrate SUID Programs
* based on:
https://stackoverflow.com/questions/8953424/how-to-get-the-username-in-c-c-in-linux
* Make sure run the following
* sudo chown root:root nameofprogram
* sudo chmod u+s nameofprogram
*/

int main(int argc, char *argv[])
{
    struct passwd *pw;
    uid_t uid;

    uid = geteuid ();
    pw = getpwuid (uid);
    if (pw)
    {
        puts (pw->pw_name);
        exit (EXIT_SUCCESS);
    }
    else
    {

```

```
    puts ("Error");  
    exit (EXIT_FAILURE);  
}  
}
```

Deliverable 1. Using the code above, create a file called `effective_user.c` and compile and execute the file as a normal user and using `sudo`. Provide a screenshot similar to the one below.

```
(champuser@kali)-[~/sec335/week10]  
$ ls  
effective_user.c  
  
(champuser@kali)-[~/sec335/week10]  
$ gcc effective_user.c -o effective_user  
  
(champuser@kali)-[~/sec335/week10]  
$ ./effective_user  
champuser  
  
(champuser@kali)-[~/sec335/week10]  
$ sudo ./effective_user  
root  
  
(champuser@kali)-[~/sec335/week10]  
$
```

Deliverable 2. What are the octal (numeric) permissions of the `effective_user` program? Using `ls -l` you should be able to calculate these permissions, you can also use the "stat" program as a shortcut. Remember `r=4`, `w=2`, `x=1`, and `"-"` is a `0`

```
(champuser@kali)-[~/sec335/week10]  
$ ls -l effective_user  
-rwxr-xr-x 1 champuser champuser 16768 Nov  8 08:32 effective_user
```

Repeat the following use of `ls -l` and `stat` on the `passwd` program

```

(champuser@kali)-[~/sec335/week10]
$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 63960 Feb  7 2020 /usr/bin/passwd

(champuser@kali)-[~/sec335/week10]
$ stat /usr/bin/passwd
File: /usr/bin/passwd
Size: 63960      Blocks: 128      IO Block: 4096   regular file
Device: hda1h/2049d  Inode: 2093257   Links: 1
Access: (4755/-rwsr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2021-09-25 13:21:34.835769738 -0400
Modify: 2020-02-07 09:54:14.000000000 -0500
Change: 2021-06-28 10:17:47.847951925 -0400
Birth: 2021-06-28 10:17:47.659951926 -0400

```

💡 The octal code is not intuitive. The `/usr/bin/passwd` program has the `suid` bit set which means that the program runs with the owner's permissions (`root`). This makes sense because when a normal user changes their password the `/etc/passwd` and `/etc/shadow` files must be changed. Note the leading 4 in the octal code. This indicates a `suid` executable (the 'x' is implied).

Deliverable 3. Figure out how to change the ownership of your `c` program executable such that the file is owned by user: `root` and group: `root`. Once you've done that, add the `suid` bit to the program (this is shown in the screenshot) and execute the program as a normal user. Provide a screenshot similar to the one below:

```

(champuser@kali)-[~/sec335/week10]
$ ls -l
total 24
-rwxr-xr-x 1 root root 16768 Nov  8 08:32 effective_user
-rw-r--r-- 1 champuser champuser 563 Nov  8 08:20 effective_user.c

(champuser@kali)-[~/sec335/week10]
$ sudo chmod u+s effective_user

(champuser@kali)-[~/sec335/week10]
$ ./effective_user
root

(champuser@kali)-[~/sec335/week10]
$

```

Updated Nov 14, 2022

Deliverable 4. Hit the internet and find a means to search for suid programs across your kali system. Do so as a normal user as this is a privilege escalation technique you might use. Make sure to document this. You will need to deal with permissions errors by piping those to /dev/null. Provide a screenshot showing your command and listing similar to that below. Your own sudo program should be in the list.

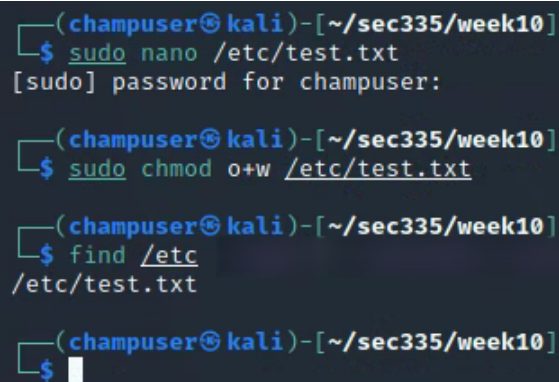
```
(champuser@kali)-[~/sec335/week10]
$ find /
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/snapd/snap-confine
/usr/lib/xorg/Xorg.wrap
/usr/sbin/pppd
/usr/sbin/mount.nfs
/usr/sbin/mount.cifs
/usr/libexec/polkit-agent-helper-1
/usr/bin/kismet_cap_linux_bluetooth
/usr/bin/kismet_cap_rz_killerbee
/usr/bin/su
/usr/bin/fusermount3
/usr/bin/kismet_cap_nrf_51822
/usr/bin/chfn
/usr/bin/umount
/usr/bin/kismet_cap_nrf_mousejack
/usr/bin/sudo
/usr/bin/vmware-user-suid-wrapper
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/kismet_cap_linux_wifi
/usr/bin/kismet_cap_nrf_52840
/usr/bin/mount
/usr/bin/kismet_cap_nxp_kw41z
/usr/bin/kismet_cap_ti_cc_2540
/usr/bin/gpasswd
/usr/bin/kismet_cap_ubertooth_one
/usr/bin/kismet_cap_ti_cc_2531
/usr/bin/ntfs-3g
/home/champuser/sec335/week10/effective_user
/opt/google/chrome/chrome-sandbox
/snap/core/12834/bin/mount
/snap/core/12834/bin/ping
/snap/core/12834/bin/ping6
/snap/core/12834/bin/su
/snap/core/12834/bin/umount
/snap/core/12834/usr/bin/chfn
/snap/core/12834/usr/bin/chsh
/snap/core/12834/usr/bin/gpasswd
```

Updated Nov 14, 2022

Deliverable 5. A suid program has been hidden on rocky (10.0.17.200). Please hunt it down. Provide a screenshot that shows the command and file found. It will be obvious and the name will start with a 'b'.

rwX errors

Deliverable 6. Consider the following screenshot. This user created a file under /etc/ that is world writable. Were this file to be of any security relevance, this could be a problem. Create such a file, and figure out how to find it. Show your command.

A terminal window with a dark background and light blue text. The prompt is (champuser@kali)-[~/sec335/week10]. The user enters \$ sudo nano /etc/test.txt, followed by [sudo] password for champuser:. Then the user enters \$ sudo chmod o+w /etc/test.txt. Next, the user enters \$ find /etc /etc/test.txt. Finally, the user enters \$ and the prompt returns.

```
(champuser@kali)-[~/sec335/week10]
$ sudo nano /etc/test.txt
[sudo] password for champuser:

(champuser@kali)-[~/sec335/week10]
$ sudo chmod o+w /etc/test.txt

(champuser@kali)-[~/sec335/week10]
$ find /etc
/etc/test.txt

(champuser@kali)-[~/sec335/week10]
$
```

Deliverable 7. A world writable file has been hidden on rocky. Please hunt it down. Provide a screenshot that shows the command and file found. It will start with an 's'. (note, the sys and proc directories will give you a lot of false positives)

Deliverable 8. Document your suid and world writable hunting techniques in your tech journal. You may have a need for them later. Provide a link to this technical article. No reflection is required.