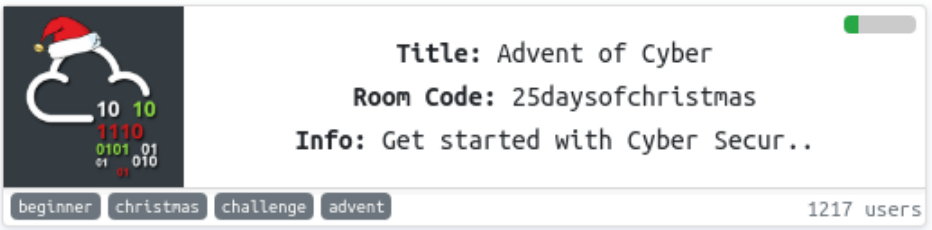
8 DECEMBER 2019 / CHRISTMAS

Linux Privilege Escalation: SUID

Set owner User ID up on execution

Check our Christmas Challenge out! https://tryhackme.com/christmas

This blog post will explain what privilege escalation is and how we can escalate our privileges using SUID permission files. Use these to solve the challenge 8 of the Christmas Advent of Cyber!



Advent of Cyber Room Image

Do this challenge in the Christmas room!
https://tryhackme.com/room/25daysofchristmas

What is Privilege Escalation?

Computer systems are designed to be used by multiple users, and privileges mean what a user is permitted to do. Common privileges include viewing and editing files, or modifying system files.

Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user.

https://payatu.com/guide-linux-privilege-escalation

What is SUID?

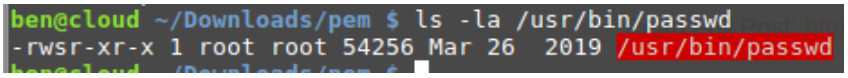
Set owner UserID up on execution is a special type of file permission given to a file. When a user runs a program, given they have the correct reading/executing rights, it will run using their account privileges. SUID allows a user to run a program using another users privileges. To further understand file privileges, complete challenge 4 in the Christmas room or read the supporting material here.

In some cases, we can take advantage of having a file run as another user, to execute commands as them. You might be thinking, why allow anyone to run a file as another user in the first place? However, we need to have certain binaries run as root by a non-privileged user.

For example, if we change our password on Linux, the program that does

this needs the permissions to right to the file system. You might not have permissions to write to the /etc/ directory, but root does. This is why the passwd binary has the SUID bit set.

If a binary has the SUID bit set, it will have an **s** appear. If we check the file permissions of the passwd binary, we can see the permissions are - **rwsr-xr-x**.



Permissions of the passwd binary

The SUID bit is set on the execute permission, meaning when a user runs this, it will run as the file owner (which is root).

In essence, SUID files execute with the permission of the file owner.

Taking advantage of SUID files

Some administrators will set the SUID bit manually to allow certain programs to be run as them. Lets say you're a system administrator and a non-privileged user wants to program that requires it to be run with higher privileges. They can set the SUID bit, then the non-privileged user can execute the program without having any extra account permissions set.

We can scan the whole file system to find all files with the SUID bit set, with the following code:

```
find / -user root -perm -4000 -exec ls -ldb {} \;
```

The find command has a parameter where it can execute commands. So when it finds a file, it will list its permissions. The output will reveal something similar to this:

```
ben@cloud ~/Downloads/pem $ find / -user root -perm -4000 -exec ls -ldb {} \; 2>/dev/null
-rwsr-xr-x 1 root root 40152 May 15  2019 /bin/mount
-rwsr-xr-x 1 root root 44168 May  7  2014 /bin/ping
-rwsr-xr-x 1 root root 27608 May 15  2019 /bin/umount
-rwsr-xr-x 1 root root 44680 May  7  2014 /bin/ping6
-rwsr-xr-x 1 root root 30800 Jul 12  2016 /bin/fusermount
-rwsr-xr-x 1 root root 40128 Mar 26  2019 /bin/su
```

More files with the SUID permission bit

We can see a few binaries that run as the root user, which are legitimate programs that have the right permissions set to properly perform a task.

If a sysadmin has manually set an SUID bit on a binary, the code above will find it. Perhaps there is a custom file that has been created by another user that runs as root? You might be able to leverage this program to escalate your privileges or run commands you'd not normally be able to.

Tip for SUID Challenge, Question 1

A normal standard Linux binary (such as the find command), can have its file owner changed and have an SUID bit set.

For example, if we wanted to see what user is the find command running as, we could do:

```
touch foo
find foo -exec whoami \;
```

This will find the file foo (which we've just created), then run the execute the code you have stated in -exec parameter.

Tip for SUID Challenge, Question 2

Found a file that looks suspicious? Try running it and seeing what you can do with it? Run whoami to see if the file actually runs as the file owner.

If you're given an option to run a command as another user... Why not run /bin/bash to run bash (to get a shell) as another user..

Ben Spring
Read more posts by this author.

Read More