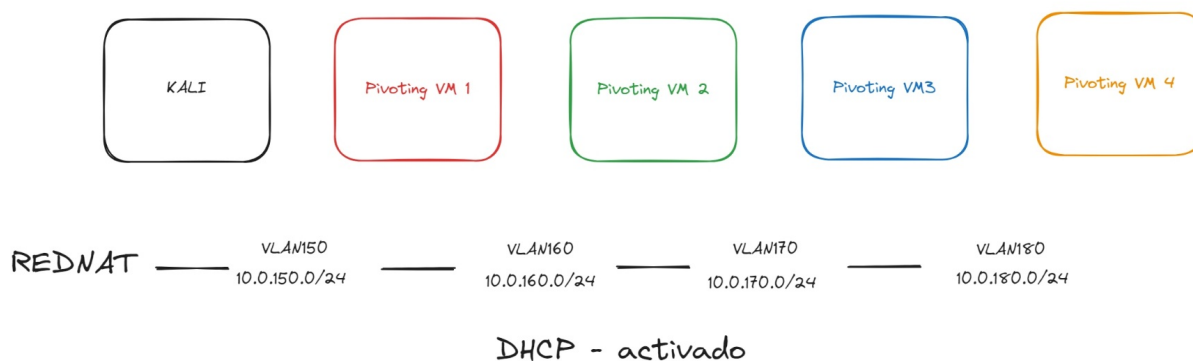


**SPRINGO 18**

**TEAM CHALLENGE**

**RETO SSH - PIVOTING**

Para la realización de este Team Challenge, se han conectado 5 máquinas virtuales a la red como se muestra en la imagen:



Para llevar a cabo la conexión de las máquinas, así como para conseguir las distintas credenciales de acceso a cada una de las máquinas y los archivos “flag.zip” para poder conseguir responder a la pregunta final que se formula en este Team Challenge, se han realizado las siguientes gestiones:

1. Se procede a un escaneo mediante la herramienta “arp-scan” desde la máquina Kali para conocer la IP de la máquina “Pivoting VM 1” (PV1, en adelante):

```
[10.0.150.4] * [No-IP] VicEvil ~ % sudo arp-scan --localnet
Interface: eth0, type: EN10MB, MAC: 08:00:27:f6:76:04, IPv4: 10.0.150.4
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.150.1      52:54:00:12:35:00      (Unknown: locally administered)
10.0.150.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.150.3      08:00:27:fd:73:71      (Unknown)
10.0.150.5      08:00:27:f4:61:f9      (Unknown)
```

Imagen 1.- escaner de IPs en la misma red, siendo la PV1: 10.0.150.5

2. - Se conecta por el protocolo SSH (Secure Shell) a la máquina PV1, dado que conocemos el usuario y contraseña: “ubuntu”, estableciendo un túnel dinámico por el puerto 9050, redirigiendo el tráfico de manera segura a través de un servidor intermediario, permitiendo el acceso a recursos de otra red, como si fuera un proxy.

```
[10.0.150.4] * [No-IP] VicEvil ~ % ssh -D 9050 ubuntu@10.0.150.5
The authenticity of host '10.0.150.5 (10.0.150.5)' can't be established.
ED25519 key fingerprint is SHA256:FKke4thhVCnDGzCdcOfF5AiItc8naC9zsRaUXVzZjrE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.150.5' (ED25519) to the list of known hosts.
ubuntu@10.0.150.5's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 579 paquetes.
371 actualizaciones son de seguridad.

Last login: Fri Feb 12 22:23:16 2021 from 10.0.30.5
ubuntu@ubuntu:~$ ip a
```

Imagen 2.- establecimiento túnel remoto a través de la PV1, para poder conseguir acceder a otro rango de red



3. Se visualiza las IPs asignadas a la maquina PV1; 10.0.150.5 y 10.0.160.6, siendo esta ultima interfaz de red la que conectará con la nueva máquina “*pivoting VM 2*” (PV2, en adelante)

```
ubuntu@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f4:61:f9 brd ff:ff:ff:ff:ff:ff
    inet 10.0.150.5/24 brd 10.0.150.255 scope global dynamic noprefixroute enp0s3
        valid_lft 526sec preferred_lft 526sec
    inet6 fe80::a00:27ff:fef4:61f9/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:92:b2:ca brd ff:ff:ff:ff:ff:ff
    inet 10.0.160.6/24 brd 10.0.160.255 scope global dynamic noprefixroute enp0s8
        valid_lft 527sec preferred_lft 527sec
    inet6 fe80::a00:27ff:fe92:b2ca/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Imagen 3.- resultado de ejecutar el comando IP a de la máquina Linux PV1

4. Se inspecciona los archivos en el interior de la maquina PV1, encontrando en el escritorio de la misma, el primer archivo “flag.zip”, el cual el transferido a la Kali, usando el método “servidor python -wget”.

```
ubuntu@ubuntu:~$ ls
Descargas Desktop Documentos Imágenes Música Plantillas Público Videos
ubuntu@ubuntu:~$ cd Desktop/
ubuntu@ubuntu:~/Desktop$ ls
flag.zip
ubuntu@ubuntu:~/Desktop$ unzip flag.zip
Archive:  flag.zip
[flag.zip] flag.txt password:
extracting: flag.txt
ubuntu@ubuntu:~/Desktop$ cat flag.txt
Cuantos
ubuntu@ubuntu:~/Desktop$

ubuntu@ubuntu:~/Desktop$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.150.4 - - [27/Oct/2024 20:25:33] "GET /flag.zip HTTP/1.1" 200 -
```

Imagen 4.- Se observa la ruta del primer archivo “flag.zip”, así como la trasferencia hacia Kali

5. Se escanea mediante la herramienta “Nmap” todos los puertos y servicios disponibles de la máquina PV1, encontrando dos puertos abiertos (**22 y 80**) con los servicios openSSH 7.6p1 y **Apache** 2.4.49, respectivamente.

```
[10.0.150.4] > [VicEvil ~/tools/dirsearch % nmap -A -p- -T5 10.0.150.5

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 15:39 EDT
Nmap scan report for 10.0.150.5
Host is up (0.00099s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 2048 6c:f4:06:9a:a7:1a:3d:21:4b:9b:b0:48:25:84:d6:a1 (RSA)
|_ 256 c0:fd:e8:f3:69:03:0d:98:32:d9:ae:9c:5a:23:2a:e3 (ECDSA)
|_ 256 59:04:b0:6c:80:45:fe:29:a1:04:f0:87:95:31:76:4b (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-title: Apache2 Ubuntu Default Page: It works
|_ http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Imagen 5.- Resultado de la ejecución del comando “Nmap” con sus respectivos parámetros de la PV1

6. Al comprobar que hay servicio web por el puerto 80 de la PV1, se procede a un escaneo con la herramienta “**dirsearch**” para encontrar directorios y archivos ocultos en el servidor web, encontrando un directorio llamado “flag.html”, el cual contiene la información con la **contraseña de la PV2**.

```
[10.0.150.4] > [VicEvil ~/tools/dirsearch % python3 dirsearch.py -u http://10.0.150.5 -i 200 -e html

dirsearch v0.4.3
Extensions: html | HTTP method: GET | Threads: 25 | Wordlist size: 9602
Target: http://10.0.150.5/

[15:45:34] Scanning:
[15:45:40] 200 - 74B - /flag.html
[15:45:40] 200 - 11KB - /index.html
```

Imagen 6.- resultado de ejecutar la herramienta “dirsearch” sobre el servidor web de PV1

```
[10.0.150.4] > [VicEvil ~/tools/dirsearch % curl http://10.0.150.5/flag.html
<html>
<body>
<b> La pass de la VM2 es: Pivoting2341 </b>
</body>
</html>
```

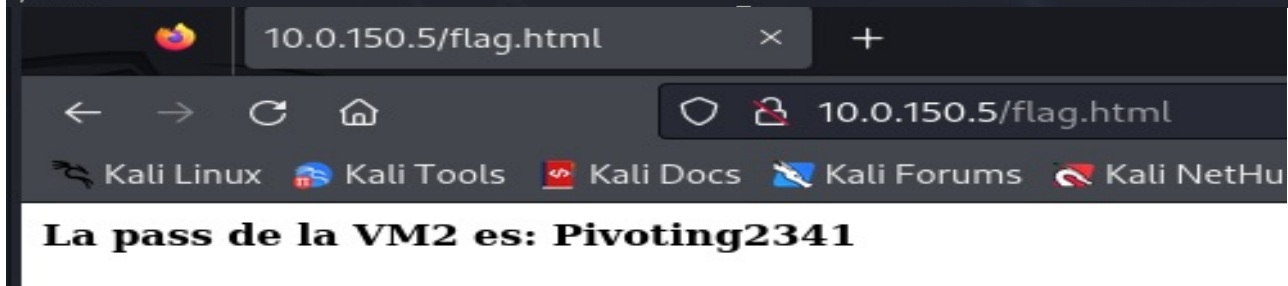


Imagen 7.- Contenido del directorio encontrado “/flag.html” con la contraseña de la PV2.

7. Mediante la herramienta **Proxychains**, la cual, permite el enrutamiento del tráfico de red a través de diferentes proxies en serie (como HTTP, SOCKS4, SOCKS5), consiguiendo ocultar la dirección IP del usuario, mejorar la privacidad y, en algunos casos, eludir restricciones de red. En nuestro caso, hemos usado sólo el **socks4** por el puerto **9050**, permitiendo un **enmascaramiento de la IP**, y de esta forma poder ejecutar diferentes herramientas y llegar a redes mas lejanas o con diferente rango de IP, como en nuestro caso, ejecutamos **“Nmap”** sobre el rango de red 10.0.160/24, aprovechando el túnel dinámico que tenemos abierto desde Kali hasta el final de la PV1, con lo cual, podemos escanear todos los hosts activos que se encuentren en dicho rango de IP, **localizando la IP de la “pivoting VM 2”**.

```
[10.0.150.4] > [127.0.0.1] VicEvil ~/tools/dirsearch % proxychains nmap 10.0.160.0/24

[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 15:53 EDT
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.1:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.2:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.3:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:80 ... OK
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.5:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.8:80
```

Imagen 8.- Ejecución de la herramienta Nmap . a través de Proxychains y el túnel dinámico que tenemos abierto

8. Se escanea mediante la herramienta **“Nmap”**, usando las ventajas y servicios que proporciona **Proxychains** y el túnel dinámico aperturado en PV1, todos los puertos y servicios disponibles de la máquina PV2, encontrando, dos puertos abiertos (**22 y 80**) con los servicios openSSH 7.6p1 y **Apache** 2.4.29, respectivamente.

```
[10.0.150.4] > [10.0.150.4] VicEvil ~ % proxychains nmap -sV 10.0.160.4
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-25 17:02 EDT
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:80 ... OK

Nmap scan report for 10.0.160.4
Host is up (0.00045s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Imagen 9.- Ejecución de Nmap a través de Proxychains para conocer los servicios y puertos abiertos de la PV2

9. Se procede a la **conexión mediante SSH a la PV2**, una vez conseguidas la IP, la contraseña y sabiendo que el puerto 22 esta abierto, inspeccionado los archivos del sistema en busca del **segundo “flag.zip”**, siendo encontrado también en el escritorio de PV2, siendo transferido a la Kali mediante el método **“server python wget”**

```
ubuntu@ubuntu:~$ cd /home/ubuntu/Desktop/
ubuntu@ubuntu:~/Desktop$ ls
flag.zip de la VM2 es: Pivoting2341
ubuntu@ubuntu:~/Desktop$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.160.6 - - [27/Oct/2024 21:03:58] "GET /flag.zip HTTP/1.1" 200 -
```



```
[10.0.150.4] * [10.0.150.4] VicEvil ~/reto_18 % proxychains wget http://10.0.160.4:8000/flag.zip
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
--2024-10-27 16:12:56-- http://10.0.160.4:8000/flag.zip
Connecting to 10.0.160.4:8000 ... [proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:8000 ... OK
connected.
HTTP request sent, awaiting response ... 200 OK
Length: 202 [application/zip]
Saving to: 'flag.zip'

flag.zip 100%[=====]
2024-10-27 16:12:56 (37.4 MB/s) - 'flag.zip' saved [202/202]

[10.0.150.4] * [10.0.150.4] VicEvil ~/reto_18 % ls
flagPV1.zip flagPV2.zip flag.zip
[10.0.150.4] * [10.0.150.4] VicEvil ~/reto_18 % rm flag.zip
[10.0.150.4] * [10.0.150.4] VicEvil ~/reto_18 % ls
flagPV1.zip flagPV2.zip
[10.0.150.4] * [10.0.150.4] VicEvil ~/reto_18 %
```

Imagen 10.- Transferencia segundo flag hacia la Kali con el método server python -wget

10. Utilizando nuevamente Proxychains, utilizamos la herramienta **dirsearch** sobre el **servidor web de la PV2**, como se ha comprobado anteriormente mediante Nmap, encontrando otro directorio **“/flag.html”**.

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains python3 dirsearch.py -u http://10.0.160.4:80 -i 200 -e html
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17

dirsearch v0.4.3

Extensions: html | HTTP method: GET | Threads: 25 | Wordlist size: 9602
Target: http://10.0.160.4/

[21:19:40] Scanning:
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:80 ... OK
[ ] 0% 3/9602 11/s job:1/1 errors:0 ... OK

[#####] 50% 4877/9602 734/s job:1/1 errors:0 ... OK
[#####] 51% 4921/9602 734/s job:1/1 errors:0 ... OK
[#####] 51% 4946/9602 734/s job:1/1 errors:0 [proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:80 ... OK
[#####] 51% 4949/9602 734/s job:1/1 errors:0 ... OK
[21:19:46] 200 - 748 - /flag.html
[#####] 52% 5006/9602 740/s job:1/1 errors:0 ... 10.0.160.4:80 ... OK
[#####] 55% 5310/9602 749/s job:1/1 errors:0 ... OK
[#####] 55% 5363/9602 749/s job:1/1 errors:0 ... OK
[#####] 56% 5407/9602 749/s job:1/1 errors:0 ... 10.0.160.4:80 ... OK
[21:19:47] 200 - 11KB - /index.html
[#####] 57% 5563/9602 751/s job:1/1 errors:0 [proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:80 ... OK
[#####] 60% 5822/9602 757/s job:1/1 errors:0 ... OK
[#####] 61% 5867/9602 756/s job:1/1 errors:0 ... OK
```

Imagen 11.- Uso de dirsearch a través de Proxychains, consiguiendo el directorio /flag.html

11. Se procede a consultar el directorio encontrado en el punto anterior, aportando información con la **contraseña de la maquina “pivoting VM 3(PV3, en adelante)”**

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains curl http://10.0.160.4/flag.html
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 10.0.160.4:80 ... OK
<html>
<body>
<b> La pass de la VM3 es: 3412Pivoting </b>
</body>
</html>
```




Imagen 12.- Contraseña de acceso a la maquina PV3

12. Una vez llegados a este punto, necesitamos obtener un nuevo túnel dinámico que vaya desde la PV1 hasta le final de la PV2 y que conecte con el que ya teníamos abierto entre Kali y PV1.

Para construir el nuevo túnel, en la terminal donde tenemos abierto el primer túnel dinámico, es decir en la PV1, y usando las técnicas de ssh-tunneling, ejecutaremos el **nuevo túnel dinámico, utilizando la IP de PV2**, no sin antes haber modificado el archivo de configuración en la ruta **“/etc/proxychains.conf”**, agregando una nueva instrucción: **“socks4 127.0.0.1 9051”**, todo con la finalidad de poder llegar a la PV3.

13. Se procede a realizar un escaneo **Nmap** mediante proxychains, que como se puede observar **enlaza el socks 9050 y 9051**, resultando que la IP de la **PV3 es la 10.0.170.5**, ya que la 10.0.170.4, pertenece a la segunda interfaz de red de PV2. Además muestra que tiene abiertos los puertos **22 y 80**, con los mismo servicios que las anteriores máquinas (SSH y Apache)

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains nmap -A -p- -T5 10.0.170.1-10

[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 19:46 EDT
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.2:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.3:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.6:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.9:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.4:80 ... OK
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.5:80 ... OK
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.7:80 ... OK
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.8:80 ... OK
PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 c6:f4:06:9a:a7:1a:3d:21:4b:9b:b0:48:25:84:d6:a1 (RSA)
|   256  c0:fd:e8:f3:69:03:0d:98:32:d9:ae:9c:5a:23:2a:e3 (ECDSA)
|_  256  59:04:b0:6c:80:45:fe:29:a1:04:f0:87:95:31:76:4b (ED25519)
53/tcp    closed domain
80/tcp    open  http              Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Apache2 Ubuntu Default Page: It works
443/tcp   closed https
445/tcp   closed microsoft-ds
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Imagen 13.- Ejecución de Nmap a través de los dos túneles establecidos para alcanzar PV3

14. Se procede a la **conexión mediante SSH a la PV3**, una vez conseguidas la IP, la contraseña y sabiendo que el puerto 22 esta abierto, inspeccionado los archivos del sistema en busca del **tercer archivo “flag.zip”**, siendo encontrado también en el escritorio de PV2, siendo transferido a la Kali mediante el método **“server python\_wget”**

```
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % ls
'Apache Status_PV3.pdf' flagPV1.zip flagPV2.zip
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % proxychains wget http://10.0.170.5:8000/flag.zip
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
--2024-10-27 20:45:23-- http://10.0.170.5:8000/flag.zip
Connecting to 10.0.170.5:8000 ... [proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.170.5:8000 ... OK
connected.
HTTP request sent, awaiting response... 200 OK
Length: 209 [application/zip]
Saving to: 'flag.zip'

flag.zip
100%[=====]
2024-10-27 20:45:23 (37.3 KB/s) - 'flag.zip' saved [209/209]

[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % mv flag.zip flagPV3.zip
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % ls
'Apache Status_PV3.pdf' flagPV1.zip flagPV2.zip flagPV3.zip
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 %
```

15. Por cambiar la la manera de conseguir acceso al puerto 80, se crea un túnel local entre la PV3 y la Kali, mediante la conexión por SSH a la máquina PV3.

```
[10.0.150.4] > [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains ssh -L 8080:10.0.170.5:80 ubuntu@10.0.170.5
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 10.0.170.5:22 ... OK
The authenticity of host '10.0.170.5 (10.0.170.5)' can't be established.
ED25519 key fingerprint is SHA256:FKke4thhVCnDGzCdcOfF5AiItc8naC9zsRaUXVzZjrE.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:1: [hashed name]
~/.ssh/known_hosts:3: [hashed name]
~/.ssh/known_hosts:4: [hashed name]
~/.ssh/known_hosts:5: [hashed name]
~/.ssh/known_hosts:6: [hashed name]
~/.ssh/known_hosts:7: [hashed name]
~/.ssh/known_hosts:8: [hashed name]
~/.ssh/known_hosts:9: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.170.5' (ED25519) to the list of known hosts.
ubuntu@10.0.170.5's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Pueden actualizarse 579 paquetes.
371 actualizaciones son de seguridad.

Last login: Fri Feb 12 22:05:07 2021 from 10.0.40.7
```

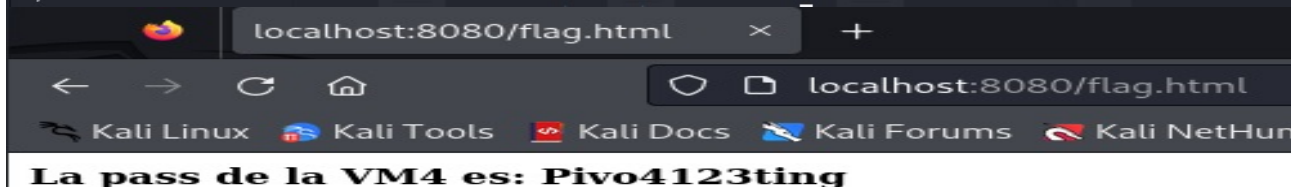
Imagen 14.- Ejecución del túnel local entre PV3 y Kali

16. Se usa la herramienta **dirsearch** sobre el **servidor web de la PV3**, encontrando el **tercer directorio “/flag.html”**. En este caso **no** ha hecho falta usar la herramienta **Proxychains**, debido q que hemos establecido un túnel local entre ambas maquinas, **redirigiendo** todo el **puerto 80** de la PV3 al puerto **8080** de nuestra maquina kali, es decir, **nuestro hosts**.

```
[10.0.150.4] > [10.0.150.4] VicEvil ~/tools/dirsearch % python3 dirsearch.py -u http://localhost:8080 -i 200 -e html
dirsearch v0.4.3
Server Status for localhost (via 10.0.170.5)
Extensions: html | HTTP method: GET | Threads: 25 | Wordlist size: 9602
Server version: Apache/2.4.29 (Ubuntu)
Target: http://localhost:8080/
Scanned By: 2020-08-12T21:33:25
[20:06:49] Scanning:
[20:06:56] 200 - 74B - /flag.html
[20:06:56] 200 - 11KB - /index.html
[20:06:59] 200 - 15KB - /server-status
[20:06:59] 200 - 15KB - /server-status/
```

17. Se procede a consultar el directorio encontrado en el punto anterior, aportando información con la **contraseña de la maquina “pivoting VM 4(PV4, en adelante).**

```
[10.0.150.4] > [10.0.150.4] VicEvil ~/tools/dirsearch % curl http://localhost:8080/flag.html
<html>
<body>
<b> La pass de la VM4 es: Pivo4123ting </b>
</body>
</html>
```





18. Para continuar hacia la ultima máquina, se construye un nuevo túnel, en la terminal donde tenemos abiertos los dos túneles dinámicos, es decir en la PV1 y PV2, ejecutando el **tercer nuevo túnel dinámico**, sobre el ultimo incorporado (PV2), **usando la IP de PV3**, no sin antes haber modificado el archivo de configuración en la ruta **“/etc/proxychains.conf”**, agregando una nueva instrucción: **“socks4 127.0.0.1 9052”**, todo con la finalidad de poder llegar a la PV4.

19. Se continua con la ejecución de **Nmap** a través de la cadena de proxies (socks4 9050,9051 y 9052), establecidos en los túneles dinámicos creados al efecto, resultando que la **IP de PV4 es 10.0.180.5**, teniendo abiertos los **puerto 22 y 80**, como todas las máquinas anteriores.

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains nmap -A -T5 -p 22,80,21 10.0.180.1-6

[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 20:49 EDT
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.2:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.3:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.6:80 ←socket error or timeout!
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.5:80 ... OK
Nmap scan report for 10.0.180.5
Host is up (0.070s latency).

PORT      STATE SERVICE VERSION
21/tcp    closed ftp
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 6c:f4:06:9a:a7:1a:3d:21:4b:9b:b0:48:25:84:d6:a1 (RSA)
|   256 c0:fd:e8:f3:69:03:0d:98:32:d9:ae:9c:5a:23:2a:e3 (ECDSA)
|   256 59:04:b0:6c:80:45:fe:29:a1:04:f0:87:95:31:76:4b (ED25519)
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 10.0.180.6
Host is up (3.1s latency).

PORT      STATE SERVICE VERSION
21/tcp    closed ftp
22/tcp    closed ssh
80/tcp    closed http

Post-scan script results:
|_ ssh-hostkey: Possible duplicate hosts
|_ Key 256 c0:fd:e8:f3:69:03:0d:98:32:d9:ae:9c:5a:23:2a:e3 (ECDSA) used by:
|   10.0.180.4
|   10.0.180.5
|_ Key 2048 6c:f4:06:9a:a7:1a:3d:21:4b:9b:b0:48:25:84:d6:a1 (RSA) used by:
|   10.0.180.4
|   10.0.180.5
|_ Key 256 59:04:b0:6c:80:45:fe:29:a1:04:f0:87:95:31:76:4b (ED25519) used by:
|   10.0.180.4
|   10.0.180.5
```

20. Se procede a la **conexión mediante SSH a la PV4**, una vez conseguidas la IP, la contraseña y sabiendo que el puerto 22 esta abierto, inspeccionado los archivos del sistema en busca del **cuarto archivo “flag.zip”**, siendo encontrado también en el escritorio de PV4, transferido a la Kali mediante el método **“server python\_wget”**

```
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % proxychains wget http://10.0.180.5:8000/flag.zip
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
--2024-10-27 20:57:11-- http://10.0.180.5:8000/flag.zip
Connecting to 10.0.180.5:8000 ... [proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.5:8000 ... OK
connected.
HTTP request sent, awaiting response... 200 OK
Length: 205 [application/zip]
Saving to: 'flag.zip'

flag.zip
100%[=====]

2024-10-27 20:57:11 (34.2 KB/s) - 'flag.zip' saved [205/205]

[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % mv flag.zip flagPV4.zip
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 % ls
'Apache Status_PV3.pdf' flagPV1.zip flagPV2.zip flagPV3.zip flagPV4.zip
[10.0.150.4] * [127.0.0.1] VicEvil ~/reto_18 %
```

21. Se procede a ver el contenido del **servidor web de la PV4**, se apertura un **túnel local** entre la **Kali y PV4**, a través de la conexión SSH de la IP de la PV4, como se ha explicado anteriormente, teniendo el siguiente **Flujo de Tráfico**:

Kali -> PV1 (SOCKS en 9050 -red 10.0.150.0/24 y 10.0.160.0/24)  
 PV1 -> PV2 (SOCKS en 9051- red 10.0.160.0/24 y 10.0.170.0/24)  
 PV2 -> PV3 (SOCKS en 9052 -red 10.0.170.0/24 y 10.0.180.0/24)  
 PV4 -> red 10.0.180.0/24

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains ssh -L 8080:10.0.180.5:80 ubuntu@10.0.180.5
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.5:22 ... OK
ubuntu@10.0.180.5's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 579 paquetes.
371 actualizaciones son de seguridad.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 28 02:05:12 2024 from 10.0.180.4
ubuntu@ubuntu:~$
```

22. Se usa la herramienta **dirsearch** sobre el **servidor web de la PV4**, encontrando el **cuarto directorio "/flag.html"**. En este caso **no** ha hecho falta usar la herramienta **Proxychains**, como se ha explicado anteriormente.

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains python3 dirsearch.py -u http://10.0.180.5:80 -i 200 -e html
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17

dirsearch v0.4.3

Extensions: html | HTTP method: GET | Threads: 25 | Wordlist size: 9602
Target: http://10.0.180.5/

[21:20:02] Scanning:
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.5:80 ... OK
##### ] 51% 4971/9602 747/s job:1/1 errors:0 ... OK
[21:20:08] 200 - 102B - /flag.html
##### ] 52% 5045/9602 747/s job:1/1 errors:0 ... OK
##### ] 53% 5156/9602 741/s job:1/1 errors:0 ... 10.0.180.5:80 ... OK
##### ] 55% 5331/9602 742/s job:1/1 errors:0 ... OK
##### ] 56% 5382/9602 742/s job:1/1 errors:0 ... OK
##### ] 56% 5411/9602 742/s job:1/1 errors:0
[proxychains] Dynamic chain ... 127.0.0.1:9050 ...
OK
[21:20:09] 200 - 11KB - /index.html
```

23. Se procede a consultar el directorio encontrado en el punto anterior, aportando información sobre la **consecución del reto**, solicitando **responder a la pregunta** que están en los **archivos "flag.zip"** que se ha ido transfiriendo hacia la Kali.

```
[10.0.150.4] * [10.0.150.4] VicEvil ~/tools/dirsearch % proxychains curl http://10.0.180.5/flag.html
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9051 ... 127.0.0.1:9052 ... 10.0.180.5:80 ... OK
<html>
<body>
<b> CONSEGUIDO! Pide la clave de los ZIP y responde a la pregunta! </b>
</body>
</html>
```

24. Una vez en nuestra máquina Kali, abrimos el directorio donde se ha ido almacenando los distintos archivos flag.zip, ejecutando en un **bucle “for”** para automatizar el la descompresión de los archivos y, como todos tiene el mismo nombre, que se vaya guardando en carpetas diferentes.

```
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % for PV in *.zip;do unzip "$PV" -d "${PV%.zip}"; done
Archive:  flagPV1.zip
[flagPV1.zip] flag.txt password:
extracting: flagPV1/flag.txt
Archive:  flagPV2.zip
[flagPV2.zip] flag.txt password:
extracting: flagPV2/flag.txt
Archive:  flagPV3.zip
[flagPV3.zip] flag.txt password:
extracting: flagPV3/flag.txt
Archive:  flagPV4.zip
[flagPV4.zip] flag.txt password:
extracting: flagPV4/flag.txt
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % ls
'Apache Status_PV3.pdf'  flagPV1  flagPV1.zip  flagPV2  flagPV2.zip  flagPV3  flagPV3.zip  flagPV4  flagPV4.zip
```

25. Finalmente, se procede a la lectura de los diferentes archivos de texto, siendo la pregunta: **¿Cuántos puertos diferentes hay en un sistema?**

-- En un sistema, **existen 65,536 puertos en total**, numerados del 0 al 65,535, los cuales, se dividen en tres rangos principales:

- Puertos Bien Conocidos (Well-Known Ports)(0 - 1023)
- Puertos Registrados (1024- 49,151)
- Puertos Dinámicos o Privados (49,152- 65,535)

-- En relación a los **sistemas explotados**, los **únicos puertos abiertos** han sido el **22 y el 80**, que pertenecen a los puertos bien conocidos.

```
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % ls
'Apache Status_PV3.pdf'  flagPV1  flagPV1.zip  flagPV2  flagPV2.zip  flagPV3  flagPV3.zip  flagPV4  flagPV4.zip
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % cat flagPV1
cat: flagPV1: Is a directory
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % cd flagPV1
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV1 % cat flag.txt
Cuantos
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV1 % cd ..
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % cd flagPV2
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV2 % cat flag.txt
Puertos
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV2 % cd ..
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % cd flagPV3
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV3 % cat flag.txt
Diferentes hay
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV3 % cd ..
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18 % cd flagPV4
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV4 % cat flag.txt
Un sistema
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV4 % #EN UN SISTEMA HAY 65.536 PUERTOS, SIENDO LOS 1024 PRIMEROS LOS LLAMADOS "BIEN CONOCIDOS"
[10.0.150.4] > [127.0.0.1] VicEvil ~/reto_18/flagPV4 % #EN LOS SISTEMAS PIVOTING (1-4) SE HA IDO REPITIENDO LOS PUERTOS 22 Y 80.
```