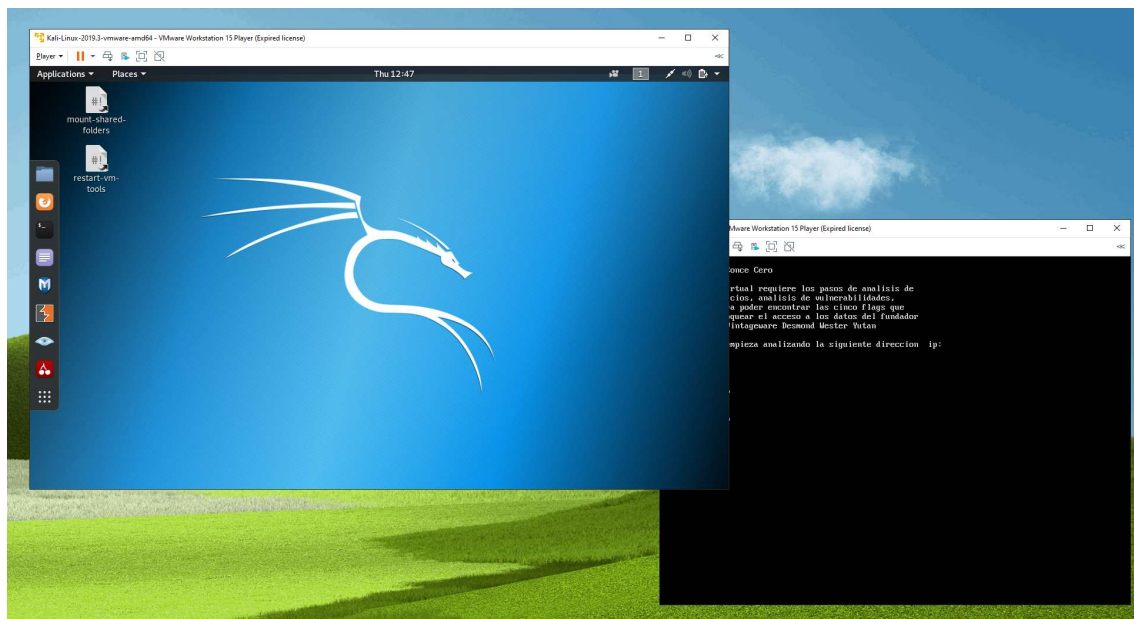


# Escape room CONDECERO – Bandera/clave 1

## Introducción

Para comenzar, debemos lanzar las dos máquinas virtuales correspondientes, utilizando el programa VMware tal y como se explica en la guía de instalación del entorno para la resolución de la práctica.

De esta manera, deberíamos tener en nuestro ordenador una visión similar a esta:



En la ventana de la izquierda será donde realicemos nuestras acciones para acceder a la información de la máquina de la derecha.

## Primeros pasos

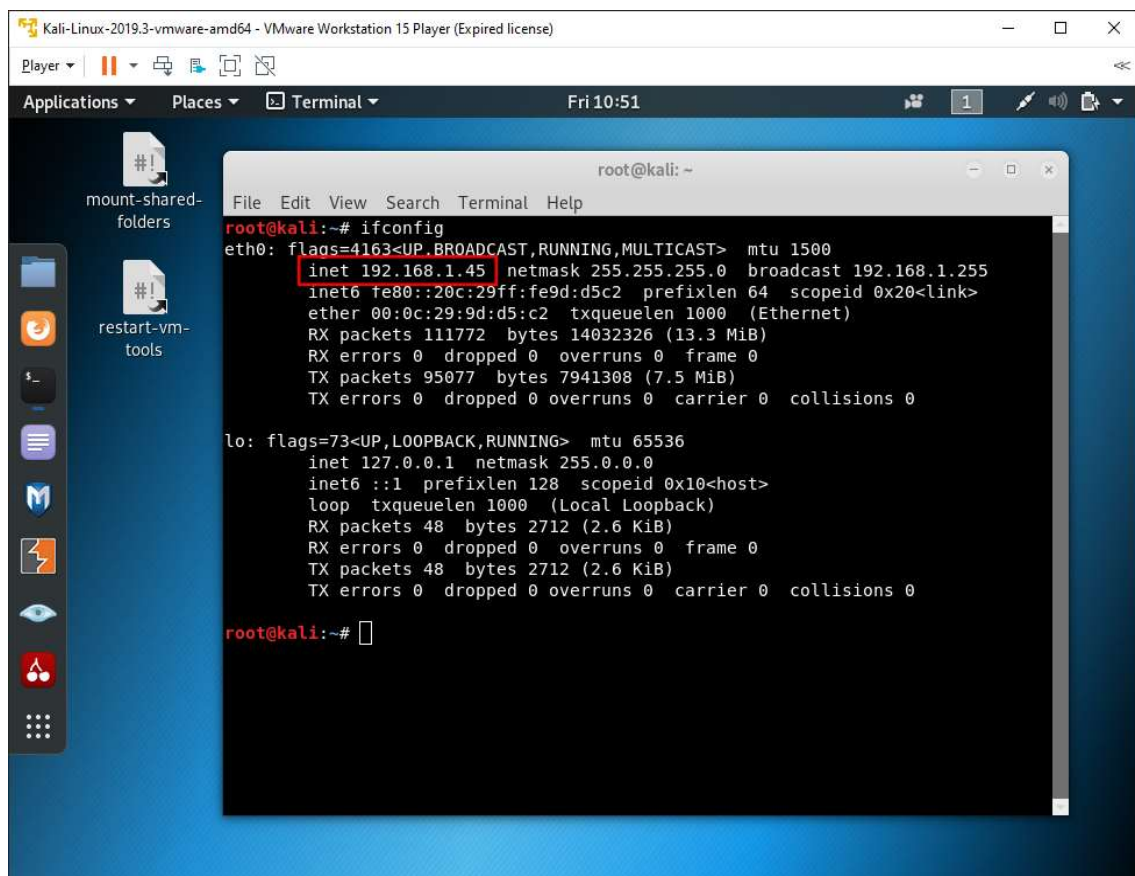
Lo primero que necesitamos saber es la dirección ip de nuestra máquina, para conocer en qué red nos estamos moviendo.

Para ello, abrimos una ventana de terminal y utilizamos el comando **ifconfig**, que nos proporciona información sobre la configuración de las interfaces de red.

Introducimos el comando

**ifconfig**

y pulsamos enter



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.45 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::20c:29ff:fe9d:d5c2 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:9d:d5:c2 txqueuelen 1000 (Ethernet)  
    RX packets 111772 bytes 14032326 (13.3 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 95077 bytes 7941308 (7.5 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 48 bytes 2712 (2.6 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 48 bytes 2712 (2.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
root@kali:~#
```

Podemos observar que, en nuestro caso, la dirección es: 192.168.1.45, pero cada alumno podría tener un valor diferente.

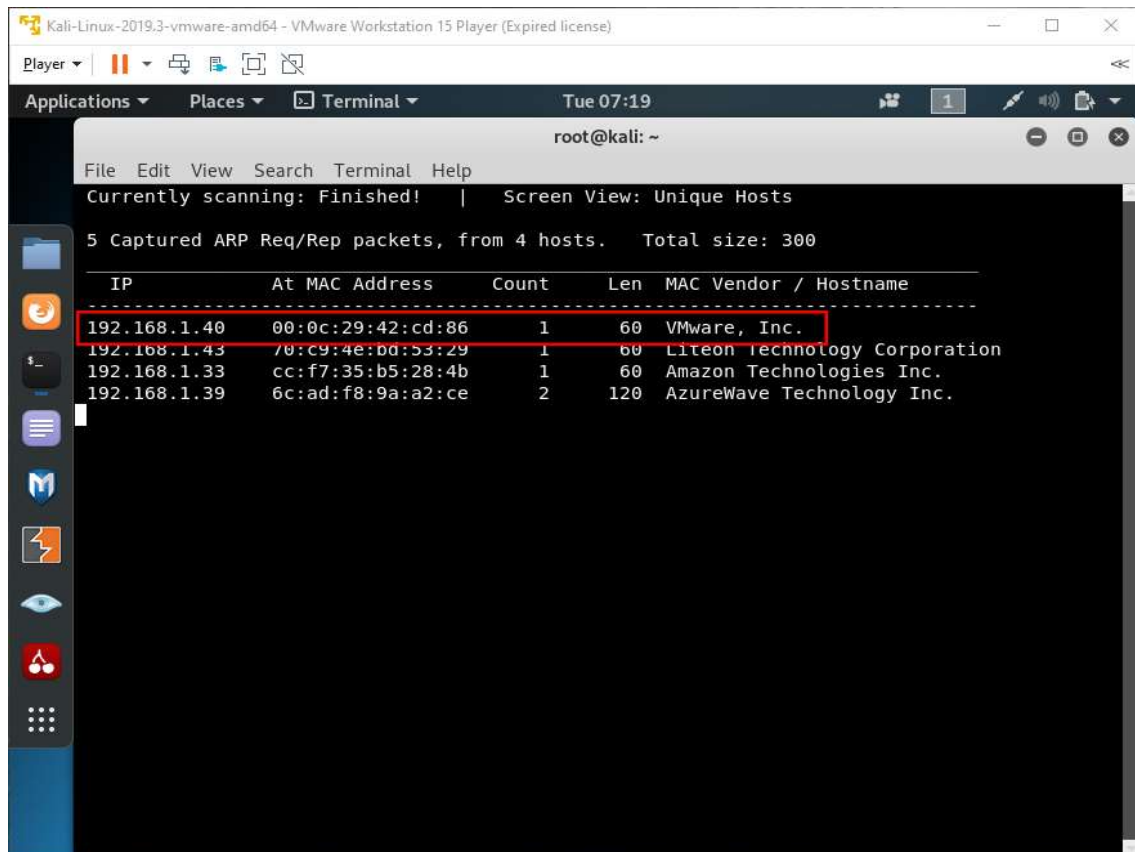
Para descubrir nuestro objetivo, buscaremos máquinas conectadas a nuestra misma red, utilizando el comando **netdiscover**, que realiza un barrido de paquetes ARP en la subred que se indica con el parámetro -r

En nuestro caso queremos analizar la subred 192.168.1.45/16, es decir, todas las direcciones que tengan la forma 192.168.xxx.xxx

Introducimos el comando

```
sudo netdiscover -r [dirección ip]/16
```

y pulsamos enter



```
root@kali: ~  
File Edit View Search Terminal Help  
Currently scanning: Finished! | Screen View: Unique Hosts  
5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 300  


| IP           | At MAC Address    | Count | Len | MAC Vendor / Hostname         |
|--------------|-------------------|-------|-----|-------------------------------|
| 192.168.1.40 | 00:0c:29:42:cd:86 | 1     | 60  | VMware, Inc.                  |
| 192.168.1.43 | 70:c9:4e:bd:53:29 | 1     | 60  | Liteon technology Corporation |
| 192.168.1.33 | cc:f7:35:b5:28:4b | 1     | 60  | Amazon Technologies Inc.      |
| 192.168.1.39 | 6c:ad:f8:9a:a2:ce | 2     | 120 | AzureWave Technology Inc.     |


```

La dirección IP de la máquina que queremos atacar será una perteneciente a **VMware**, el resto son diferentes dispositivos que también se encuentran en la red en la que estemos trabajando.

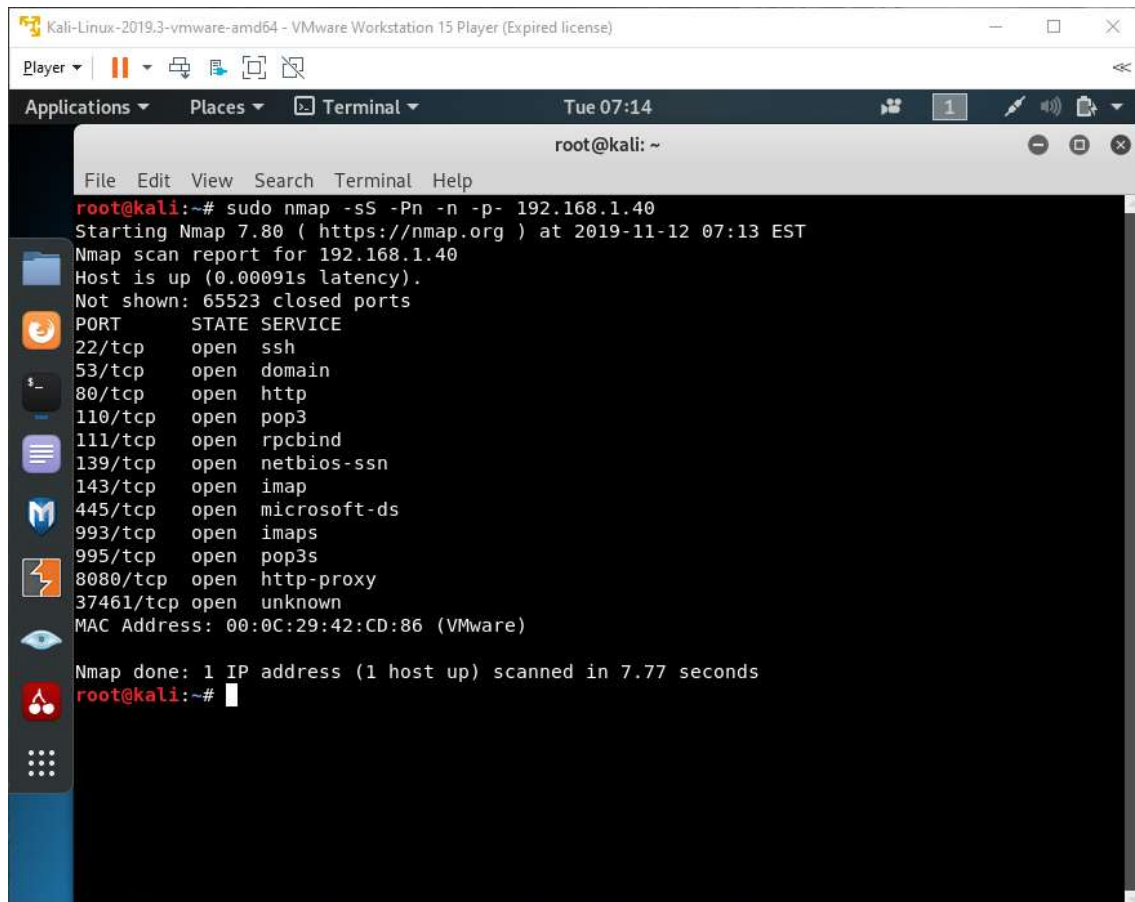
En nuestro caso la dirección sería 192.168.1.40

Una vez conseguido este dato, comenzaremos el análisis de puertos abiertos, que son los que aceptan conexiones TCP o paquetes UDP e indican los servicios disponibles para ser utilizados en una red.

Pulsamos Ctrl+C para dejar de ejecutar el comando anterior e introducimos el comando

```
sudo nmap -sS -Pn -n -p- [dirección ip]
```

y pulsamos enter



```
root@kali:~# sudo nmap -sS -Pn -n -p- 192.168.1.40
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-12 07:13 EST
Nmap scan report for 192.168.1.40
Host is up (0.00091s latency).
Not shown: 65523 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
110/tcp   open  pop3
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
993/tcp   open  imaps
995/tcp   open  pop3s
8080/tcp  open  http-proxy
37461/tcp open  unknown
MAC Address: 00:0C:29:42:CD:86 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 7.77 seconds
root@kali:~#
```

Como se puede apreciar, en este caso los puertos abiertos son el puerto 22, el puerto 53, el puerto 80, el puerto 110, el puerto 111, el puerto 139, el puerto 143, el puerto 445, el puerto 993, el puerto 995, el puerto 8080 y el puerto 37461.

El siguiente paso es analizar los servicios que funcionan en los puertos abiertos que hemos descubierto y sus respectivas versiones.

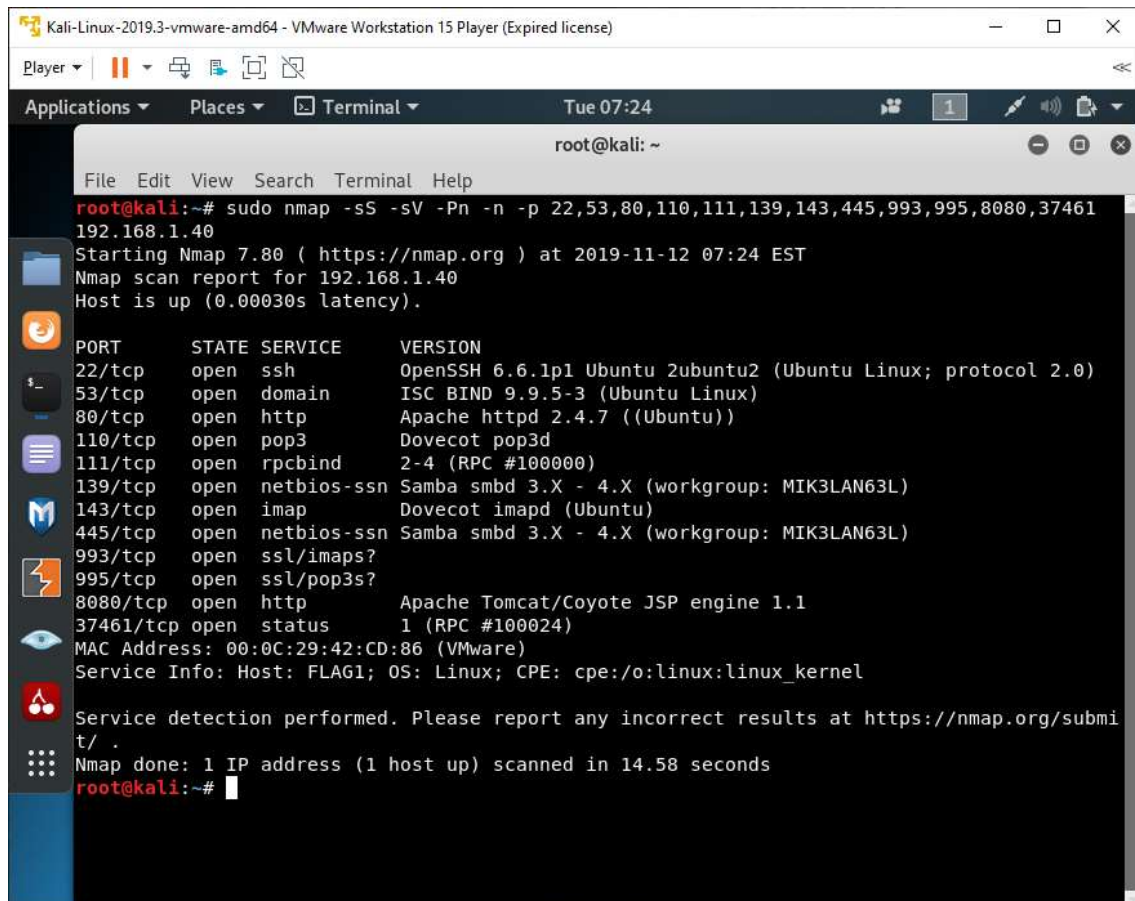
Para esto utilizaremos el mismo comando **nmap** pero añadiendo el parámetro **-sV** y especificando los puertos a analizar mediante **-p**  
22,53,80,110,111,139,143,445,993,995,8080,37461

Introducimos el comando

```
sudo nmap -sS -sV -Pn -n -p [puertos abiertos] [dirección ip]
```

y pulsamos enter

Se pueden observar los distintos servicios y versiones que se están ejecutando en los puertos abiertos.



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# sudo nmap -sS -sV -Pn -n -p 22,53,80,110,111,139,143,445,993,995,8080,37461 192.168.1.40  
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-12 07:24 EST  
Nmap scan report for 192.168.1.40  
Host is up (0.00030s latency).  
  
PORT      STATE SERVICE      VERSION  
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)  
53/tcp    open  domain       ISC BIND 9.9.5-3 (Ubuntu Linux)  
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))  
110/tcp   open  pop3         Dovecot pop3d  
111/tcp   open  rpcbind      2-4 (RPC #100000)  
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: MIK3LAN63L)  
143/tcp   open  imap         Dovecot imapd (Ubuntu)  
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: MIK3LAN63L)  
993/tcp   open  ssl/imap     Samba smbd 3.X - 4.X (workgroup: MIK3LAN63L)  
995/tcp   open  ssl/pop3     Samba smbd 3.X - 4.X (workgroup: MIK3LAN63L)  
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1  
37461/tcp open  status       1 (RPC #100024)  
MAC Address: 00:0C:29:42:CD:86 (VMware)  
Service Info: Host: FLAG1; 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 14.58 seconds  
root@kali:~#
```

Ahora evaluaremos las vulnerabilidades en estos servicios a través de los scripts de automatización de **nmap**, una de sus funciones más poderosas y flexibles.

Utilizaremos de nuevo el comando **nmap** en los mismos puertos, añadiendo esta vez los parámetros de los scripts correspondientes. En este caso utilizaremos **auth** para evaluar la autenticación, **discovery** para descubrir más sobre la red, **exploit** para descubrir vulnerabilidades de ese tipo y **vuln** para vulnerabilidades específicas.

Introducimos el comando

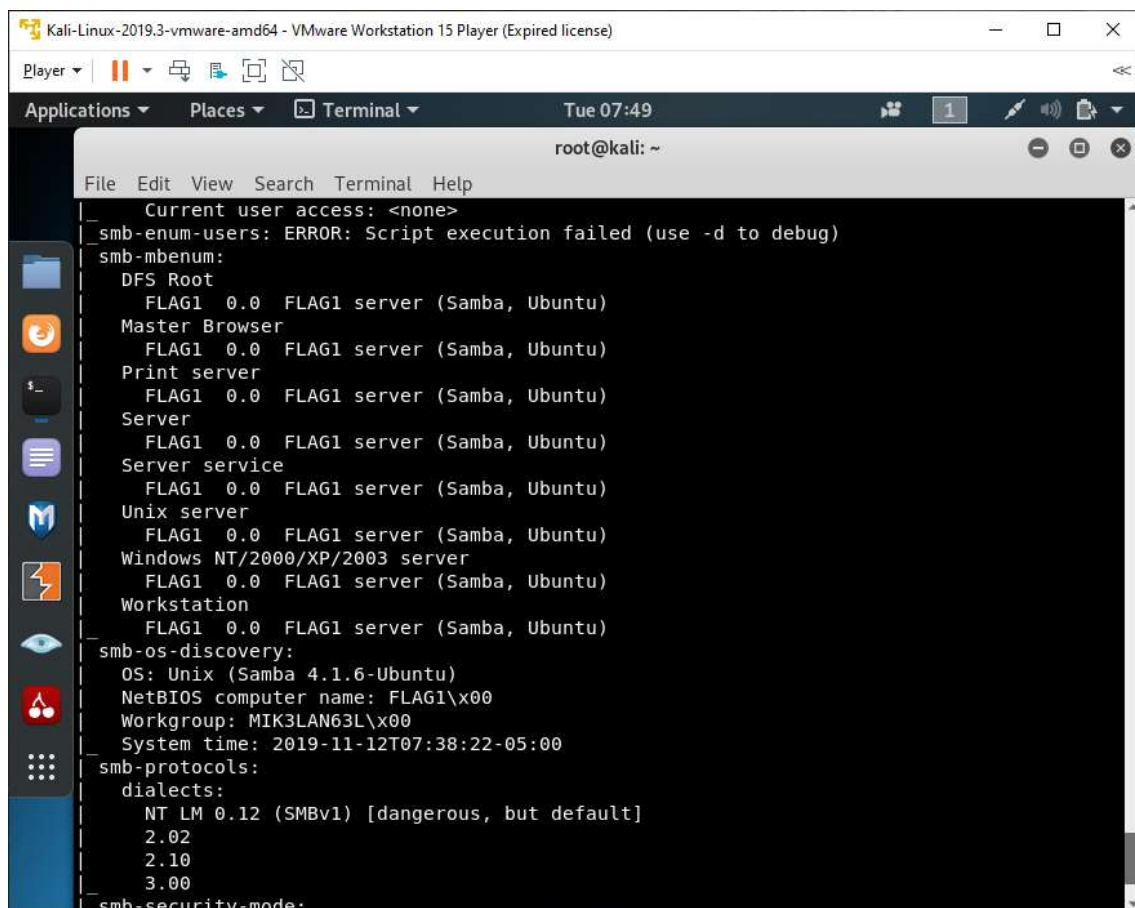
```
sudo nmap -sS -sV -Pn -n -p [puertos abiertos] --script=auth,exploit,discovery,vuln [dirección ip]
```

y pulsamos enter

Esperamos un tiempo hasta que se ejecuten todos los scripts y obtendremos una gran cantidad de información que tendremos que revisar para conseguir todas las claves ocultas en banderas.

## Bandera/Clave 1

Comenzaremos por investigar la información que se ofrece del servicio **Samba**, que se utiliza para la interconexión de redes con sistemas basados en Unix y otras basadas en Windows, ya que podemos ver que hay menciones a FLAG1.



```
Kali-Linux-2019.3-vmware-amd64 - VMware Workstation 15 Player (Expired license)
root@kali: ~
File Edit View Search Terminal Help
Current user access: <none>
smb-enum-users: ERROR: Script execution failed (use -d to debug)
smb-mbenum:
DFS Root
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Master Browser
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Print server
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Server
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Server service
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Unix server
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Windows NT/2000/XP/2003 server
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
Workstation
FLAG1 0.0 FLAG1 server (Samba, Ubuntu)
smb-os-discovery:
OS: Unix (Samba 4.1.6-Ubuntu)
NetBIOS computer name: FLAG1\x00
Workgroup: MIK3LAN63L\x00
System time: 2019-11-12T07:38:22-05:00
smb-protocols:
dialects:
NT LM 0.12 (SMBv1) [dangerous, but default]
2.02
2.10
3.00
smb-security-mode:
```

Para ello, utilizaremos la herramienta **enum4linux**, que enumera información sobre sistemas Windows y Samba.

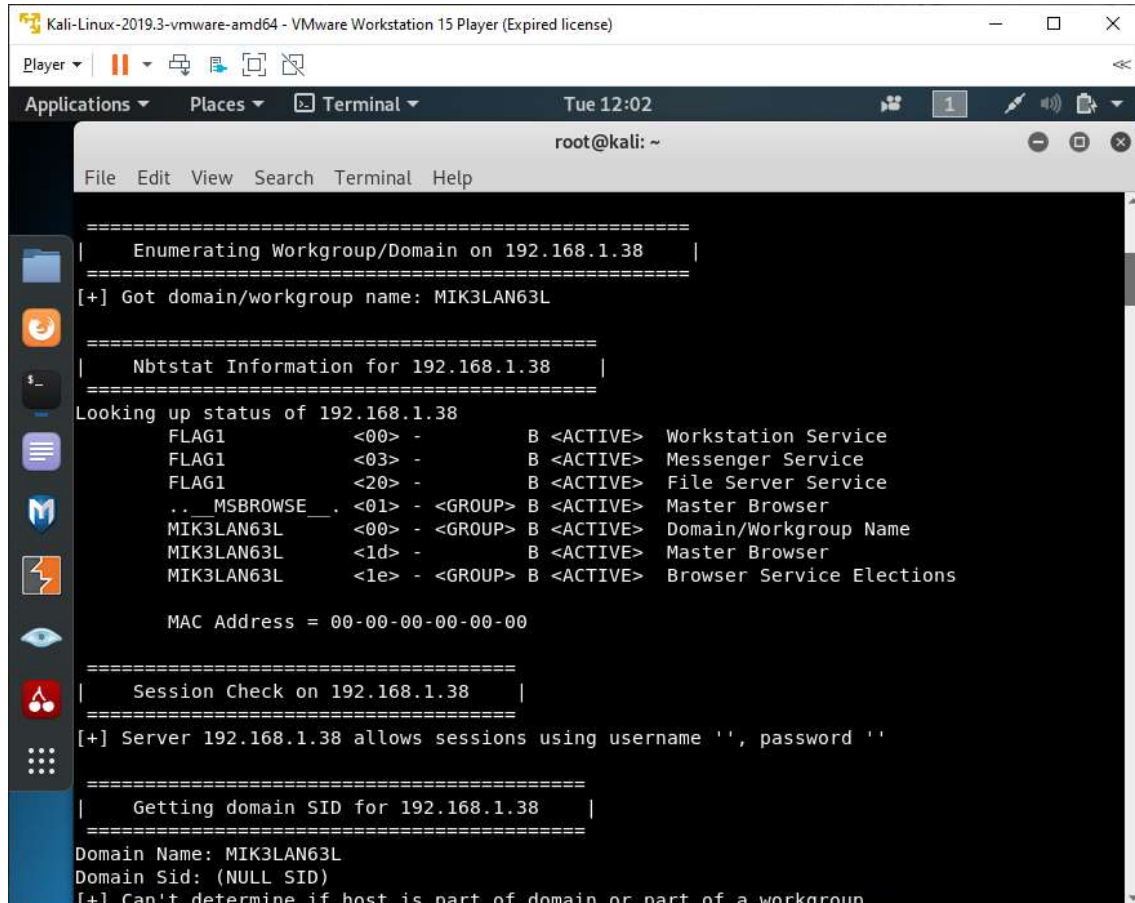
<https://labs.portcullis.co.uk/tools/enum4linux/>



Introducimos el comando

`enum4linux -a [dirección ip]`

y pulsamos enter



```
Kali-Linux-2019.3-vmware-amd64 - VMware Workstation 15 Player (Expired license)
Tue 12:02
root@kali: ~
File Edit View Search Terminal Help

=====
| Enumerating Workgroup/Domain on 192.168.1.38 |
=====
[+] Got domain/workgroup name: MIK3LAN63L

=====
| Nbtstat Information for 192.168.1.38 |
=====
Looking up status of 192.168.1.38
FLAG1 <00> - B <ACTIVE> Workstation Service
FLAG1 <03> - B <ACTIVE> Messenger Service
FLAG1 <20> - B <ACTIVE> File Server Service
.._MSBROWSE_ <01> - <GROUP> B <ACTIVE> Master Browser
MIK3LAN63L <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
MIK3LAN63L <1d> - B <ACTIVE> Master Browser
MIK3LAN63L <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00

=====
| Session Check on 192.168.1.38 |
=====
[+] Server 192.168.1.38 allows sessions using username '', password ''

=====
| Getting domain SID for 192.168.1.38 |
=====
Domain Name: MIK3LAN63L
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
```

Encontramos la primera bandera en el nombre del grupo de trabajo/dominio

**CÓDIGO BANDERA 1: MIK3LANG3L**