

2.1 HOST DISCOVERY EN RED DESCONOCIDA



1. UN POCO DE TEORÍA

Antes de comenzar con la parte práctica sobre Kali vamos a empezar hablando de que es un “host discovery” y como se consigue.

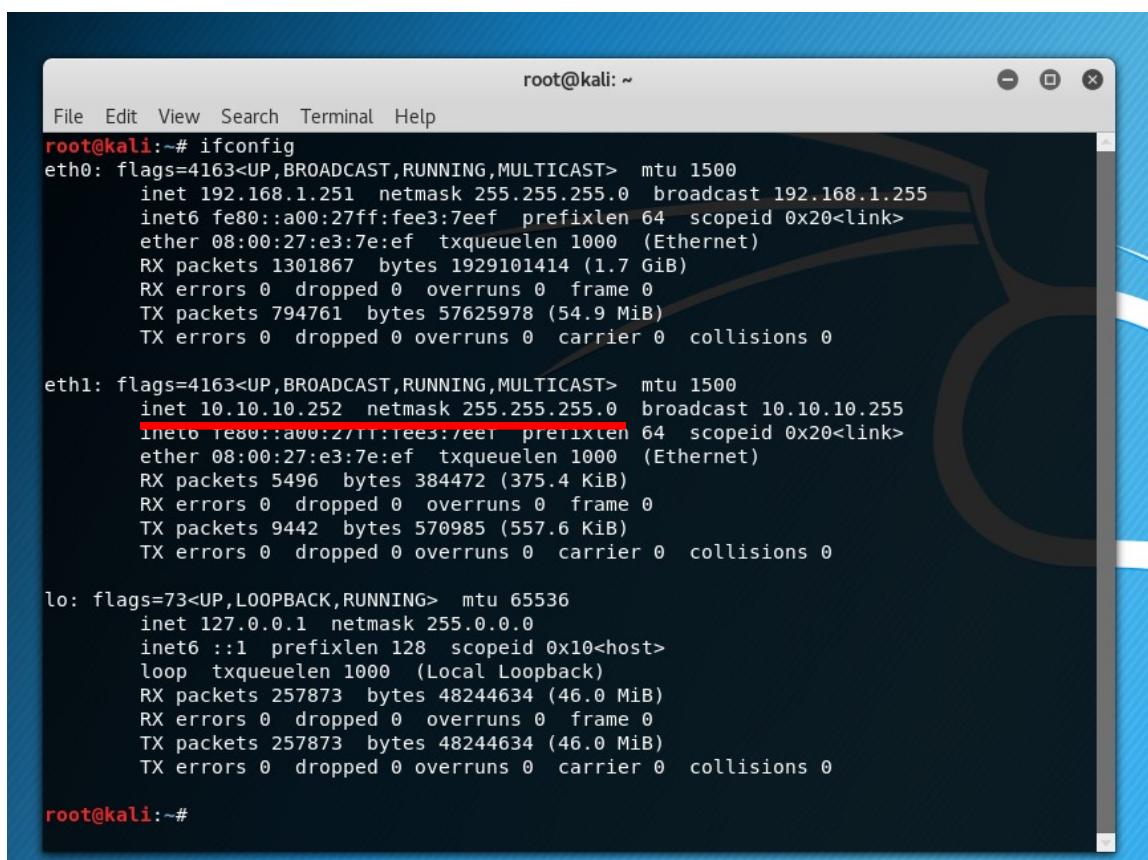
Host discovery es una técnica que permite conocer los dispositivos que están conectados a una red. Este es uno de los primeros pasos a la hora de analizar una red y conocer la cantidad de dispositivos que conviven en esta.

¿Y CÓMO SE LOGRA DESCUBRIR LOS HOSTS DE UNA RED?

Es necesario, como condición indispensable e inicial, conocer la IP de red y su máscara. Esto no debería ser un reto ya que para poder hacer un host-discovery a una red debes estar conectado a ella. Una vez conectado a esta, simplemente en nuestro Kali teclearemos el comando:

```
ifconfig
```

Este comando sacará una lista de interfaces de red. En una de ellas, dependiendo de nuestras interfaces y como las hayamos asignado, aparecerá la IP de tu dispositivo y la máscara:



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.251  netmask 255.255.255.0  broadcast 192.168.1.255
              inet6 fe80::a00:27ff:fe3:7eef  prefixlen 64  scopeid 0x20<link>
                ether 08:00:27:e3:7e:ef  txqueuelen 1000  (Ethernet)
                  RX packets 1301867  bytes 1929101414 (1.7 GiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 794761  bytes 57625978 (54.9 MiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.252  netmask 255.255.255.0  broadcast 10.10.10.255
              inet6 fe80::a00:27ff:fe3:7eef  prefixlen 64  scopeid 0x20<link>
                ether 08:00:27:e3:7e:ef  txqueuelen 1000  (Ethernet)
                  RX packets 5496  bytes 384472 (375.4 KiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 9442  bytes 570985 (557.6 KiB)
                  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 257873  bytes 48244634 (46.0 MiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 257873  bytes 48244634 (46.0 MiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

root@kali:~#
```

Voy a suponer en este momento que se conocen conceptos muy básicos de redes (IP, máscaras, subredes...), por tanto se deduce que nuestra red es la 10.10.10.0/24.

Una vez que conocemos la red a escanear, la pregunta es... ¿cómo hacerlo?. La solución es muy sencilla. Nuestro dispositivo manda tramas (paquetes) a los 256 posibles hosts que pueden convivir en la red (recordemos que es una máscara /24). Para cada posible host (10.10.10.1, 10.10.10.2, 10.10.10.3... hasta 10.10.10.255) hemos de enviar paquetería a ese host esperando su respuesta.

Evidentemente, no seremos nosotros quien lo hagamos, lo hará una herramienta dedicada a automatizar todo el trabajo. La herramienta más conocida (y presentada más abajo) llamada Nmap envía por defecto los siguientes paquetes a cada dispositivo:

- TCP SYN al puerto remoto 443 (HTTPS) → espera respuesta TCP RST o TCP SYN ACK
- TCP ACK al puerto remoto 80 (HTTP) → espera respuesta TCP RST
- ICMP Timestamp → espera respuesta ICMP Echo-Request

Se realizan estos 3 envíos debido a posibles filtrados o pérdidas de paquetería, con lo cual nos aseguramos obtener respuesta si está vivo. Se pueden configurar estos paquetes.

En cuanto alguna de las siguientes respuestas se den, sabremos que el host está vivo y lo incluiremos en la lista de dispositivos en red. Debemos saber que aunque solo existan N dispositivos encendidos, no sabremos nunca con certeza cuantos existen apagados.

Las herramientas que os mostraré serán:

- **Nmap:** la herramienta por excelencia para escaneo de puertos. Nmap es una herramienta que permite realizar una recogida de información de dispositivos de gran calidad, pudiendo especificar y configurar toda la paquetería: protocolo, puertos de envío y de destino, tipos de paquete...
- **Zenmap:** una herramienta GUI que trabaja por debajo con Nmap. Permite a los usuarios menos experimentados hacer un host discovery por interfaz de una forma muy sencilla.
- **Metasploit:** un framework dedicado al pentesting y al análisis de vulnerabilidades que entre sus funcionalidades permite un scanner de la red.
- **Nessus:** una herramienta privativa comercial (aunque con versión gratuita que es la que usaremos en este artículo) que permite el uso de muchos plugins para hacer análisis de vulnerabilidades, escaner profundo de redes...
- **Net Analyzer:** herramienta para teléfonos Android o iOS

Entre estas herramientas personalmente recomiendo **Nmap o Metasploit**, debido a ser un software muy potente, muy usado en distintas cosas más allá de host-discovery y su licencia gratuita y libre. Aún así, cualquiera de las herramientas presentadas son muy útiles a la hora de conocer vulnerabilidades y descubrir el entorno de red, por tanto dedicaremos el tiempo acorde a todas.

2. PASANDO DIRECTAMENTE A LA PRÁCTICA

A) NMAP

Para mi la herramienta más importante a la hora de hacer Information Gathering. Tiene un gran potencial, ya que logra saber hasta qué aplicaciones están corriendo en una máquina determinada.

INSTALACIÓN

En Kali Linux viene por defecto. Para otras versiones de Linux se encuentra en los repositorios oficiales, simplemente se necesita usar el gestor de paquetes (apt, yum...).

Para Windows o para obtener binarios, consultar <https://nmap.org/>

CONFIGURACIÓN

La de por defecto funciona correctamente.

EJECUCIÓN

Abrimos un nuevo terminal.

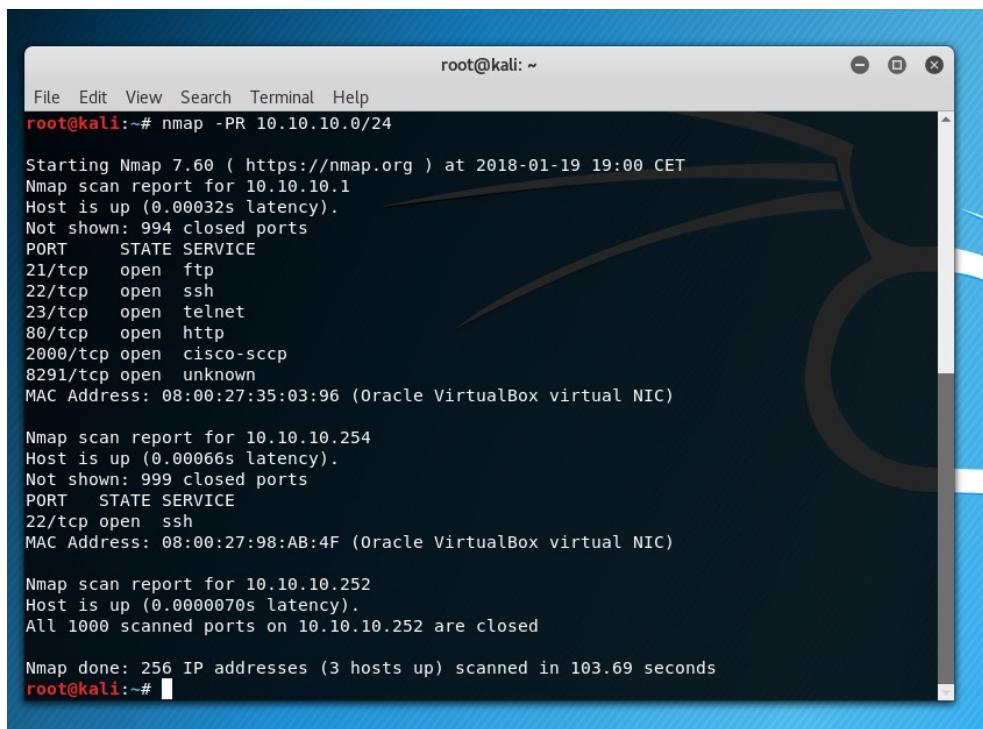
Para Host Discovery, nmap únicamente implementa 2 opciones por defecto:

nmap -PR 10.10.10.0/24

o

nmap 10.10.10.0/24

Efectúa los envíos de los paquetes explicados en la parte teórica. Esto requiere un tiempo, y muestra un desglose de puertos abiertos por host.



The screenshot shows a terminal window titled 'root@kali: ~' running on Kali Linux. The user has run the command 'nmap -PR 10.10.10.0/24'. The output is as follows:

```
root@kali:~# nmap -PR 10.10.10.0/24
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-19 19:00 CET
Nmap scan report for 10.10.10.1
Host is up (0.00032s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
2000/tcp  open  cisco-sccp
8291/tcp  open  unknown
MAC Address: 08:00:27:35:03:96 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.10.10.254
Host is up (0.00066s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:98:AB:4F (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.10.10.252
Host is up (0.0000070s latency).
All 1000 scanned ports on 10.10.10.252 are closed

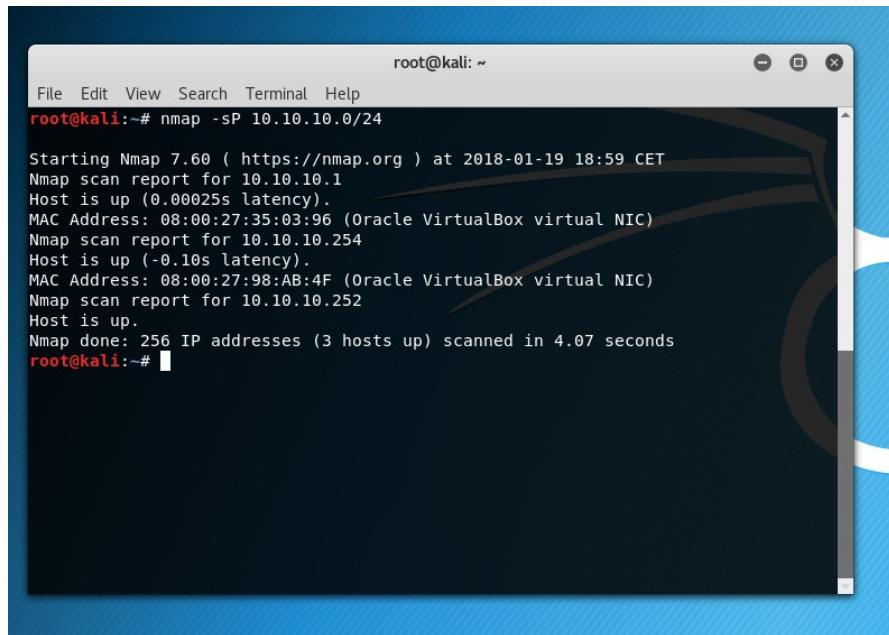
Nmap done: 256 IP addresses (3 hosts up) scanned in 103.69 seconds
root@kali:~#
```

Vamos a analizar la salida. Nmap termina mostrando una lista con 3 items, cada item es un host de la red. Por cada host, muestra la IP, la latencia, puertos abiertos y que suele correr en ellos y la MAC con su posible marca.

Cabe destacar que con este nmap no es 100% seguro que en el puerto XX esté corriendo un servicio Y debido a que cualquier servicio corre en cualquier puerto, aunque suele acertar debido a que la relación servicio-puerto suele seguir el estándar que presenta Nmap. Además, la MAC tampoco tiene que ser de esa marca debido a que cualquier dispositivo puede asignar una MAC distinta a la ofrecida por defecto para enmascarar la empresa distribuidora y no ofrecer vulnerabilidades sobre ellas.

nmap -sP 10.10.0/24

Unicamente lanza un ICMP para obtener resultados rápidos y simples:



```
root@kali:~# nmap -sP 10.10.0/24
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-19 18:59 CET
Nmap scan report for 10.10.10.1
Host is up (0.00025s latency).
MAC Address: 08:00:27:35:03:96 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.10.10.254
Host is up (-0.10s latency).
MAC Address: 08:00:27:98:AB:4F (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.10.10.252
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 4.07 seconds
root@kali:~#
```

Debemos destacar que un host discovery de Nmap muestra también nuestro propio dispositivo, pero sin desglosar MAC.

Como podemos ver, Nmap es una herramienta simple que consta de un par de comandos para el host discovery, y tiene un gran potencial, por eso es de las más usadas para estos casos.

Además, Nmap ofrece ciertas características para enmascarar nuestro host discovery. Estas características vienen dadas para evitar monitorizaciones o sensores dentro de la propia red. Para escapar de estos sensores que captan gran cantidad de paquetería de un host hacia los demás se usan los temporizadores que ofrece Nmap.

nmap -Tx 10.10.0/24

En este caso, nmap hace su función pero con la intensidad que le marquemos en x. El valor de x debe estar comprendido entre 0 y 4, donde 0 es un modo paranoico (muy sigiloso) y 4 es un envío masivo (muy rápido). Por defecto, Nmap viene en el modo 2, pero si sospechamos de sensores debemos colocar el modo 1 al menos.

*** Si quieres conocer más acerca de nmap, en este directorio git existe la guia_nmap.pdf que ilustra todos los conceptos de esta herramienta.

B) ZENMAP

Zenmap es la interfaz gráfica para Nmap más usada. Ofrece las mismas funcionalidades que Nmap pero con un entorno sencillo.

INSTALACIÓN

En Kali Linux viene por defecto. Para otras versiones de Linux se encuentra en los repositorios oficiales, simplemente se necesita usar el gestor de paquetes (apt, yum...).

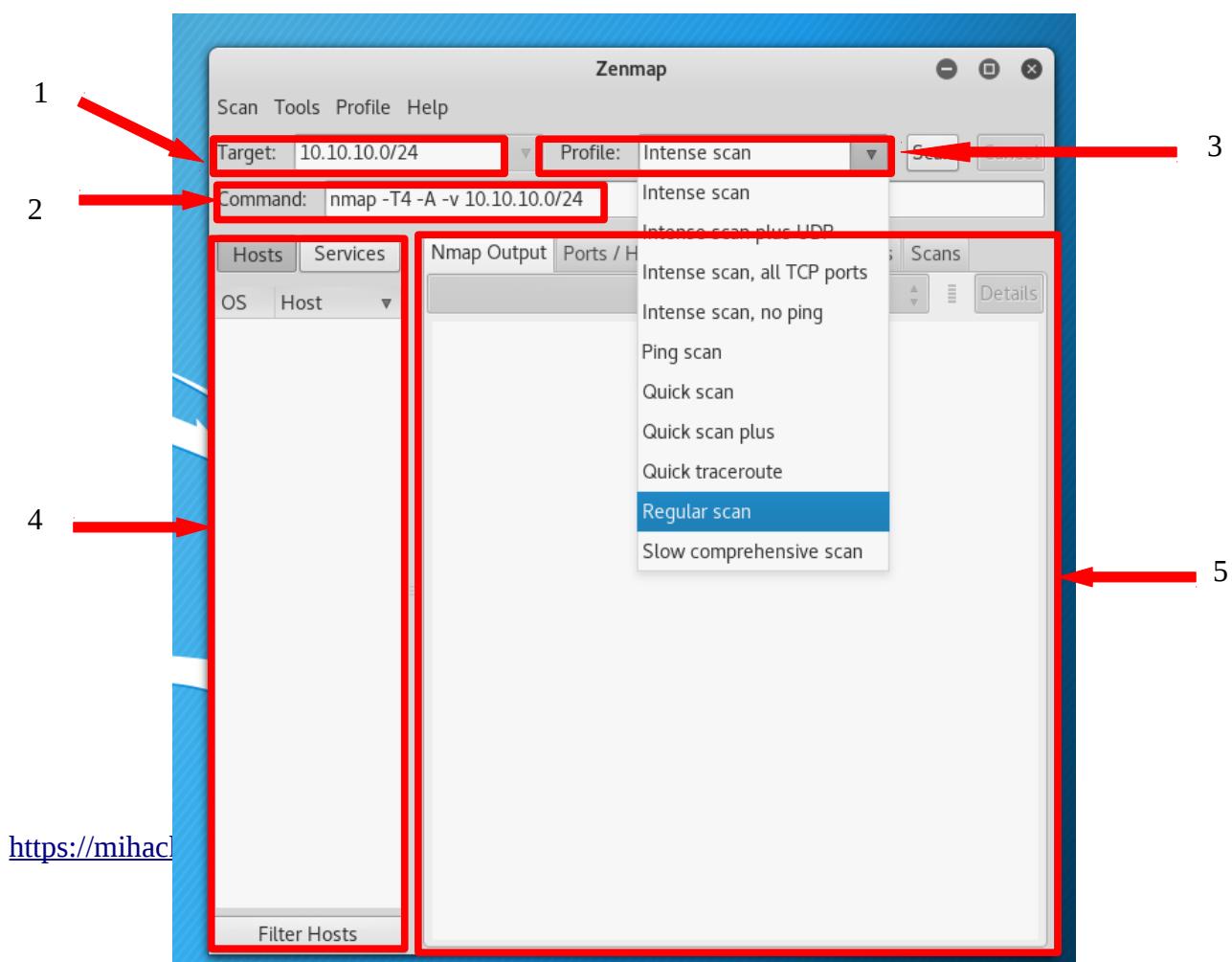
Para Windows o para obtener binarios, consultar <https://nmap.org/>

CONFIGURACIÓN

No necesita configuración específica más allá que la de por defecto.

EJECUCIÓN

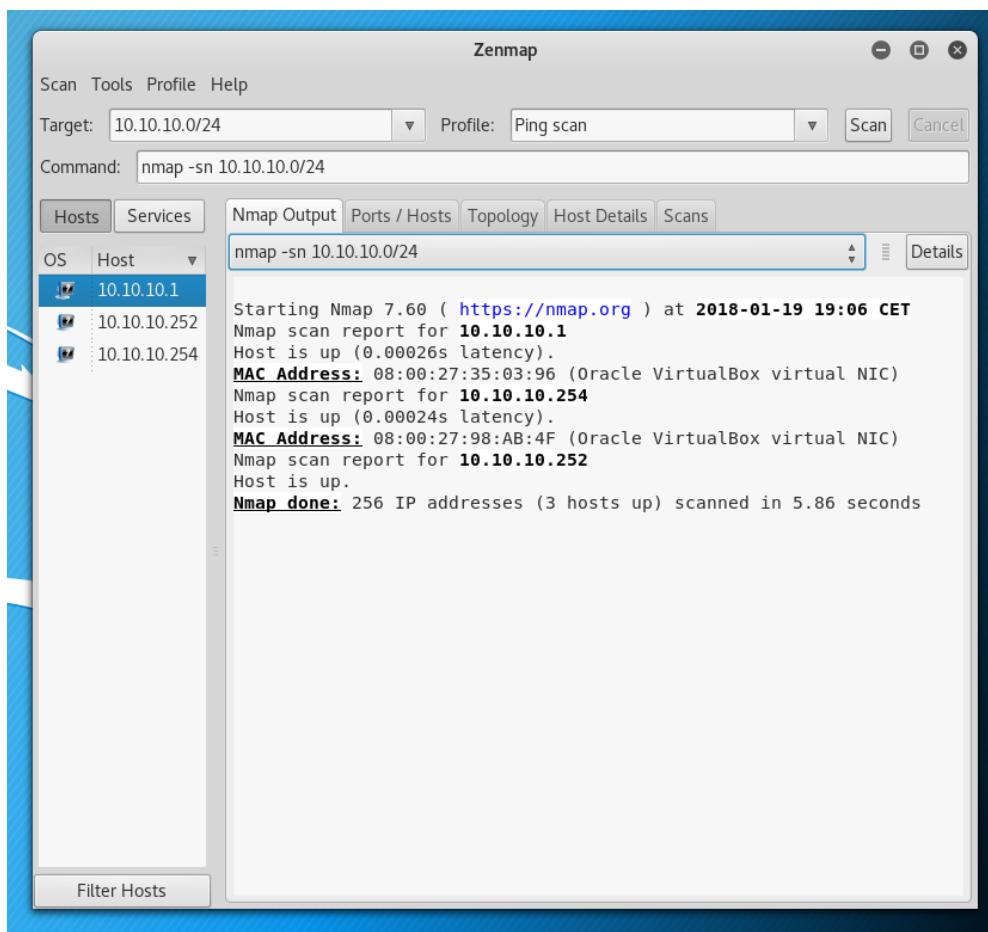
En nuestras aplicaciones, abrimos Zenmap. La interfaz se basa en 5 grandes bloques de utilidades:



<https://mihac1>

1. Target: Debemos escribir la red que queremos auditar y su mascara, en el formato de Nmap.
2. Command: Este comando se autogenera, es decir, no se suele poner a mano. Se puede escribir el comando pero entonces, ¿para qué la GUI?
3. Profile: Aquí tenemos distintos métodos para realizar es escaneo. El escaneo que haremos es el “Regular scan” que es el Nmap clásico.
4. Hosts/Services: Es una lista resumen de los hosts encontrados por Nmap y su posible SO deducido (tema de fingerprinting)
5. Output: Muestra la salida de los hosts. En el caso de host discovery no necesitamos analizar nada más en el output.

Una vez preparado el tipo de escaneo y el target, le damos al botón de “Scan” y, tras unos segundos, Zenmap sacará una salida similar a esta:



Zenmap reconoce 3 hosts que incluye en su lista y muestra la salida de Nmap correspondiente.

C) METASPLOIT

Para conocer el host discovery sobre la herramienta debemos saber la funcionalidad de metasploit. Metasploit es una herramienta de análisis de vulnerabilidades y pentesting, basada en ejecución de exploits sobre posibles vulnerabilidades es un sistema.

<https://mihackeo.blogspot.com.es/>

Como podemos vislumbrar, metasploit no nace para hacer host discovery, pero se acabó añadiendo esta funcionalidad debido al uso clave en el pentesting.

Metasploit puede hacer host discovery mediante dos métodos: mediante comandos nmap o mediante un “exploit”. Como Nmap ya lo hemos visto, voy explicar la ejecución sobre el exploit.

INSTALACIÓN

En Kali Linux viene por defecto. Para otras versiones de Unix se puede encontrar en el Git-Hub de rapid7: <https://github.com/rapid7/metasploit-framework>

Para instalarlo simplemente abrimos un terminal y escribimos:

```
curl https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall
```

Posteriormente damos permisos con **chmod** a msfinstall para ejecutarlo y, finalmente, lanzamos el script mediante **./msfinstall**.

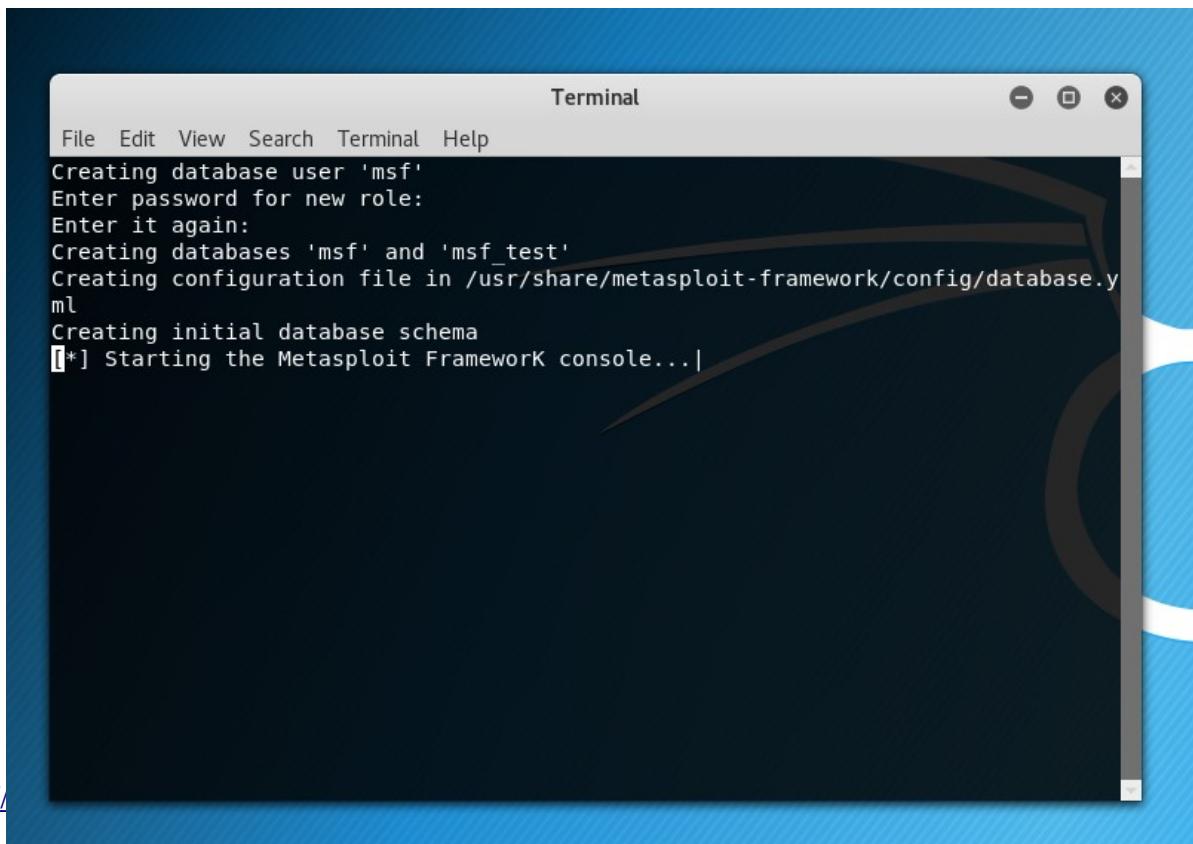
Una vez ejecutados estos pasos se instalará msfconsole, la consola de metasploit framework, que usaremos para lanzar los exploits.

CONFIGURACIÓN

No necesita configuración específica más allá que la de por defecto.

EJECUCIÓN

Escribimos en una terminal msfconsole:



```
Terminal
File Edit View Search Terminal Help
Creating database user 'msf'
Enter password for new role:
Enter it again:
Creating databases 'msf' and 'msf_test'
Creating configuration file in /usr/share/metasploit-framework/config/database.yml
Creating initial database schema
[*] Starting the Metasploit Framework console...|
```

<https://>

Tras haber creado una base de datos y haberse configurado (imagen anterior), usaremos el exploit auxiliary/scanner/discovery/arp_sweep. Vamos a usar el protocolo ARP ya que es más lógico que contesten a esta trama que no a envíos UDP, por ejemplo.

Para especificar el exploit haremos

```
use auxiliary/scanner/discovery/arp_sweep
```

Se cambia el prompt y estamos listos para configurar el escaneo.

Debemos ahora especificar los dispositivos remotos que queremos analizar (rhosts) y el dispositivo fuente desde el cual analizaremos (shost y smac) de la siguiente forma:

The image shows two terminal windows side-by-side. The left window is a Metasploit session (Terminal) with the following content:

```
File Edit View Search Terminal Help
[*****] ^"a, | | |
[%%%] ^"a,$$ | | |
[%%%] ^"$ | | |
[*****] ^"a, | | |
[%%%] ^"a,$$ | | |
[%%%] ^"$ | | |

=[ metasploit v4.16.6-dev
+ -- --=[ 1682 exploits - 964 auxiliary - 297 post
+ -- --=[ 498 payloads - 40 encoders - 10 nops
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp

msf > use auxiliary/scanner/discovery/arp_sweep
msf auxiliary(arp_sweep) > set interface eth1
interface => eth1
msf auxiliary(arp_sweep) > set rhost 10.10.10.0/24
[!] RHOST is not a valid option for this module. Did you mean RHOSTS?
rhost => 10.10.10.0/24
msf auxiliary(arp_sweep) > set rhosts 10.10.10.0/24
rhosts => 10.10.10.0/24
msf auxiliary(arp_sweep) > set shost 10.10.10.252
shost => 10.10.10.252
msf auxiliary(arp_sweep) > set smac 08:00:27:e3:7e:ef
smac => 08:00:27:e3:7e:ef
msf auxiliary(arp_sweep) > 
```

The right window is a root shell on Kali Linux (root@kali: ~) with the following netstat output:

```
File Edit View Search Terminal Help
RX packets 224106 bytes 333261326 (317.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 132397 bytes 9211133 (8.7 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.10.10.252 netmask 255.255.255.0 broadcast 10.10.10.255
inet6 fe80::a00:27ff:fee3:7eef prefixlen 64 scopeid 0x20<link>
ether 08:00:27:e3:7e:ef txqueuelen 1000 (Ethernet)
RX packets 21 bytes 2471 (2.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 23 bytes 1738 (1.6 KiB)
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 189553 bytes 31550659 (30.0 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 189553 bytes 31550659 (30.0 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# 
```

Recordamos que para hallar la IP de shost y la MAC podemos usar ifconfig.

Para ejecutar el exploit, escribimos en el shell “run” y procede a hacer un host discovery.

```
msf auxiliary(arp_sweep) > run
[+] 10.10.10.1 appears to be up (CADMUS COMPUTER SYSTEMS).
[+] 10.10.10.254 appears to be up (CADMUS COMPUTER SYSTEMS).
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(arp_sweep) > 
```

Podemos ver que muestra una lista simple con cada IP de cada host.

D) NESSUS

Nessus es una herramienta que permite escanear vulnerabilidades, de una forma muy similar a Metasploit. En este caso, os muestro esta herramienta ya que tiene una interfaz a través de navegador web, de forma que es más intuitivo.

INSTALACIÓN

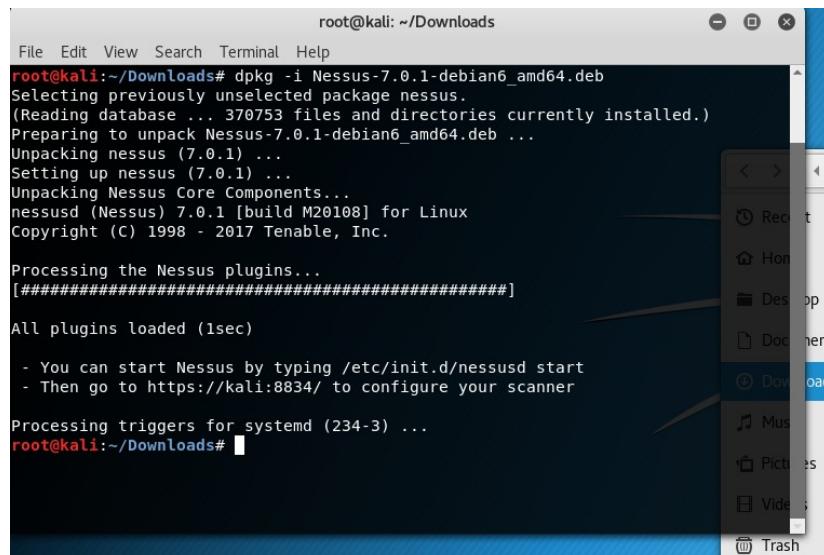
Este software no es abierto, está desarrollado por la empresa Tenable, pero tiene una versión para equipos particulares gratuita. Esta versión se llama Nessus Home y se puede encontrar en <https://www.tenable.com/products/nessus-home>. Esta versión está disponible para todos los Sistemas Operativos.

Es necesario registrarse para obtener un código de activación por correo electrónico.

- En Kali

Tanto para Kali como para cualquier S.O. basado en Debian se descarga un paquete .deb y se instala mediante

```
dpkg -i nombre_paquete.deb
```



```
root@kali:~/Downloads
File Edit View Search Terminal Help
root@kali:~/Downloads# dpkg -i Nessus-7.0.1-debian6_amd64.deb
Selecting previously unselected package nessus.
(Reading database ... 370753 files and directories currently installed.)
Preparing to unpack Nessus-7.0.1-debian6_amd64.deb ...
Unpacking nessus (7.0.1) ...
Setting up nessus (7.0.1) ...
Unpacking Nessus Core Components...
nessusd (Nessus) 7.0.1 [build M20108] for Linux
Copyright (C) 1998 - 2017 Tenable, Inc.

Processing the Nessus plugins...
[#####
All plugins loaded (1sec)

- You can start Nessus by typing /etc/init.d/nessusd start
- Then go to https://kali:8834/ to configure your scanner

Processing triggers for systemd (234-3) ...
root@kali:~/Downloads#
```

Y posteriormente arrancamos el servicio de nessus

```
root@kali:~/Downloads# /etc/init.d/nessusd start
Starting Nessus : .
root@kali:~/Downloads#
```

CONFIGURACIÓN

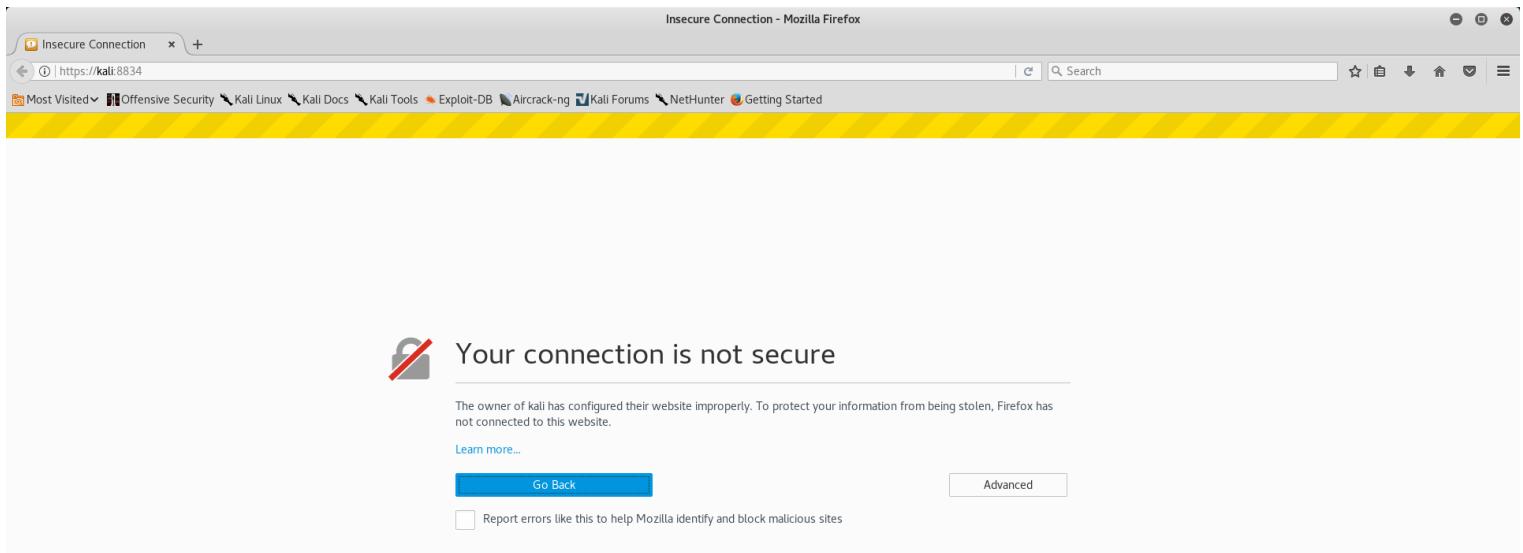
No necesita configuración específica más allá que la de por defecto.

<https://mihackeo.blogspot.com.es/>

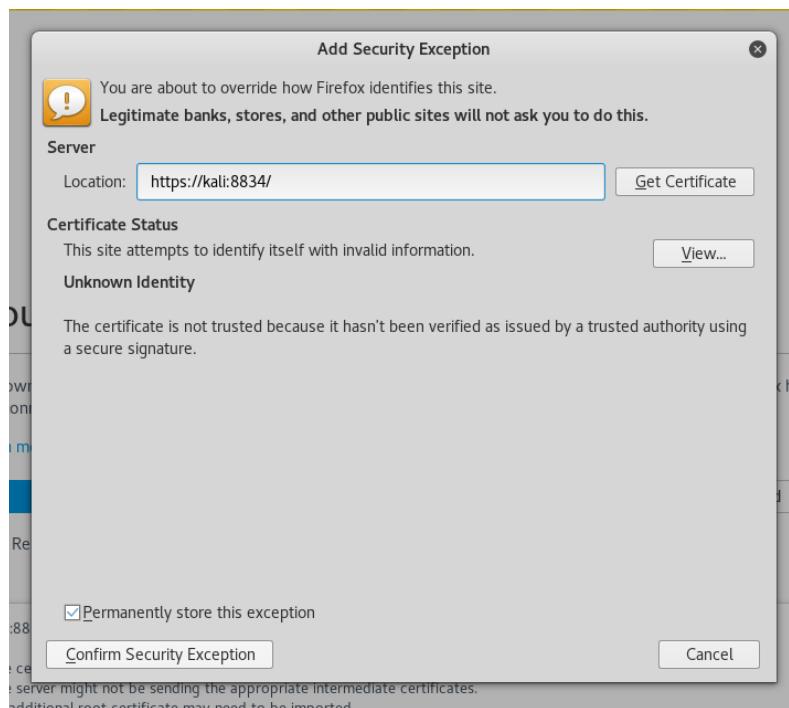
EJECUCIÓN

Abriremos ahora un navegador, en mi caso Firefox, y pondremos <https://kali:8834>

Lo que estamos haciendo es acceder a un servicio que tenemos en nuestra propia máquina (ya que mi máquina se llama kali) que está corriendo por el puerto 8834.



El certificado NO es seguro, pero es algo normal ya que no está firmado, es un servicio de tu máquina. Vamos pues a Avanced → Confirm Security Exception



Una vez se acceda al servicio habrá que loguearse e introducir el código de activación. A continuación se descargarán y instalarán plugins de análisis de vulnerabilidades.

<https://mihackeo.blogspot.com.es/>

Una vez instalado, nos aparecerá la siguiente interfaz:

The screenshot shows the Nessus web interface. On the left, there's a sidebar with 'FOLDERS' containing 'My Scans' (which is selected), 'All Scans', and 'Trash'. Under 'RESOURCES', there are 'Policies', 'Plugin Rules', and 'Scanners'. The main area is titled 'My Scans' and contains a message: 'This folder is empty. Create a new scan.' with three buttons: 'Import', 'New Folder', and 'New Scan'.

Y hacemos un nuevo escaneo clickando en “New scan”.

The screenshot shows the 'Scan Templates' page. It has a sidebar with 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Scanners). The main area is titled 'Scan Templates' and shows a grid of 16 scanning templates. Some templates have 'UPGRADE' arrows pointing to them. The templates include: Advanced Scan, Audit Cloud Infrastructure, Badlock Detection, Bash Shellshock Detection, Basic Network Scan, Credentialed Patch Audit, DROWN Detection, Host Discovery, Intel AMT Security Bypass, Internal PCI Network Scan, Malware Scan, MDM Config Audit, Mobile Device Scan, Offline Config Audit, PCI Quarterly External Scan, Policy Compliance Auditing, SCAP and OVAL Auditing, Shadow Brokers Scan, Spectre and Meltdown, WannaCry Ransomware, and Web Application Tests.

Escogemos “Host Discovery”.

The screenshot shows the 'New Scan / Host Discovery' configuration page. It has a sidebar with 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Scanners). The main area is titled 'New Scan / Host Discovery' and shows a form with tabs for 'Settings' (selected) and 'Plugins'. Under 'Settings', there are sections for 'BASIC' (General, Schedule, Notifications), 'DISCOVERY' (with a 'Discover' button), and 'REPORT' (with a 'Report' button). The 'BASIC' section includes fields for 'Name' (MiRed), 'Description', 'Folder' (My Scans), and 'Targets' (10.10.10.0/24). At the bottom are 'Save' and 'Cancel' buttons.

Asignamos un nombre y el target a escanear. Luego clickamos sobre “Scan”.

<https://mihackeo.blogspot.com.es/>

Lanzamos en el icono “Launch” y esperamos a que termine de escanear. Finalmente obtenemos el resultado:

Las barras representan vulnerabilidades encontradas de forma automática en el escaneo.

E) NET ANALYZER

Si lo que deseas es poder analizar una red de forma rápida y sencilla desde tu teléfono, la aplicación más completa es Net Analyzer.

Net Analyzer es una app para dispositivos móviles que permite host discovery, información completa de WiFi, ping, traceroute, DNS resolve...

INSTALACIÓN

Desde la Apple Store o la Play Store de forma gratuita y verificada por Play Secure.

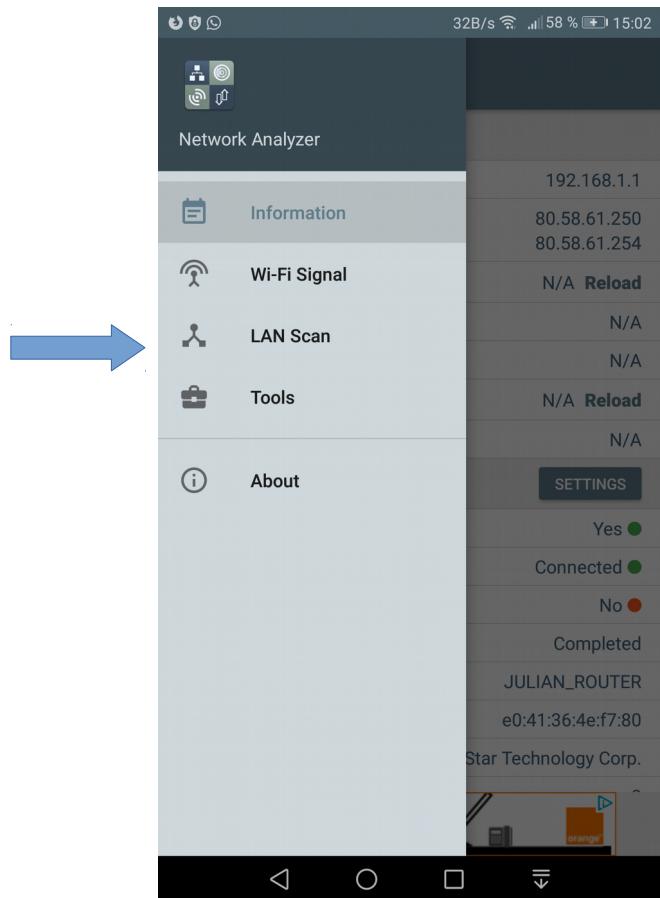
CONFIGURACIÓN

No necesita configuración específica más allá que la de por defecto.

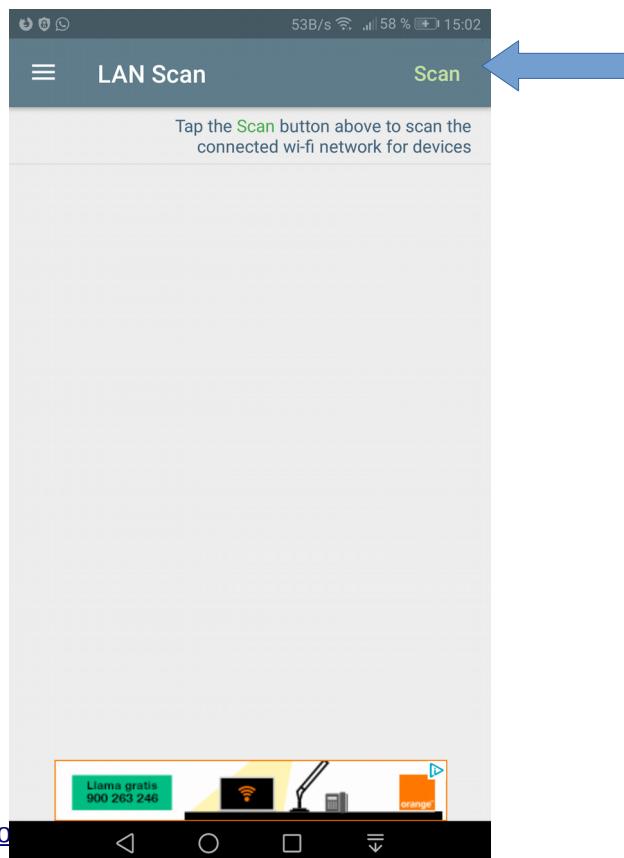
EJECUCIÓN

Para hacer un host discovery, iremos al menú de la app y escogeremos LAN Scan.

<https://mihackeo.blogspot.com.es/>

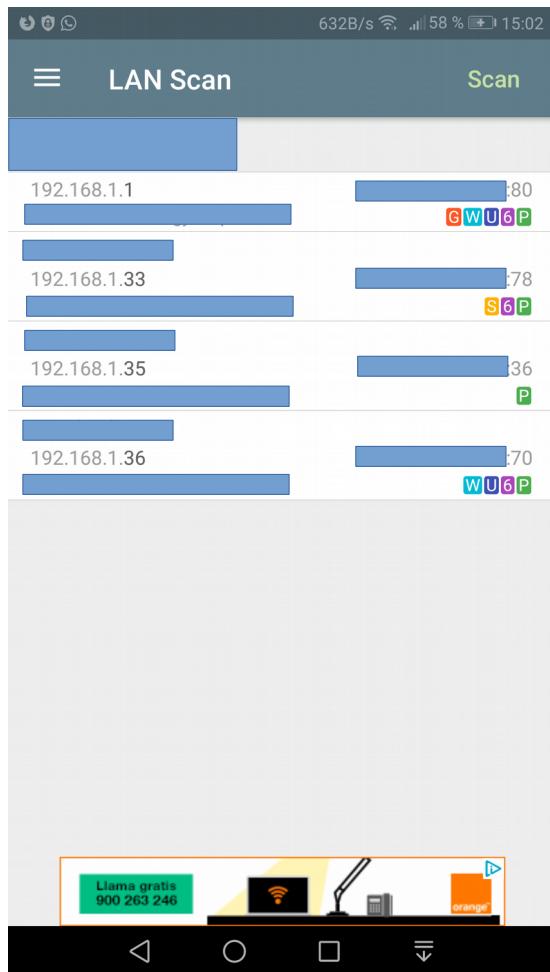


Luego pulsaremos “Scan”



<https://mihackeo.blogspot.com>

Tras esperar unos segundos aparecerán los dispositivos:



En este caso se muestran las IPs, sus MAC, el nombre del host y la posible marca de la tarjeta de red.

3. CONCLUSIONES

Como vemos, no hay una única forma de realizar un host discovery, y es un mecanismo muy simple y que logra un gran resultado. A la vez, es un proceso que sin duda es un pilar clave en el hacking, para conocer el entorno a atacar/defender.

Sin duda, Nmap y Metasploit serán usados en muchos otros artículos de Information Gathery y pentesting, así que aprender sobre estas herramientas es esencial.