DEFENSA DE UN SERVIDOR PÚBLICO FRENTE A ATAQUES A TRAVÉS DE RED MEDIANTE IPTABLES

Alumno: Eduardo Rodríguez Hernández
Tutor: Francisco Javier Alayón Hernández

Grado en Ingeniería Informática (Tecnologías de la información)

Julio 2019 – Las Palmas de Gran Canaria

AGRADECIMIENTOS.

A mi familia por haberme apoyado siempre y haber estado ahí cuando los necesitaba.

A mis compañeros por haberme ayudado y acompañado a lo largo de estos años de universidad.

A los profesores que han ayudado a mi desarrollo en el campo de estudio de este trabajo y en especial a mi tutor por haberme guiado e incentivado a lo largo del mismo.

RESUMEN:

Cuando se ofrece servicios en Internet a todo el mundo se está expuesto a cualquier tipo de interacciones tanto para obtener resultados del servicio prestado como a actuaciones buscando menoscabar el servicio o tener acceso no autorizado a los datos que se encuentra en el servidor. Por ello, en un servicio dirigido a toda la comunidad, la seguridad es de vital importancia. Hay que distinguir las interacciones orientadas a obtención del servicio de aquella que buscan bien el entorpecimiento del servicio o el acceso a datos no autorizados.

Una manera de intentar resolver esta cuestión son los cortafuegos, impidiendo en la medida que sea posible el que este tráfico malintencionado llegue a las aplicaciones que prestan el servicio. En este proyecto vamos a utilizar IPtables, cortafuego que implementan los equipos con las distribuciones de Linux, para detectar y contener los ataques más frecuentes a servidores expuestos por prestar servicios a todo el mundo.

Para la realización se realizará el estudio de los ataques más importantes, utilización o/y desarrollo de herramientas que implementen estos ataques y a creación de un cortafuego implementado con IPtables para la protección contra diferentes ataques. Estas herramientas de ataques garantizarán el correcto funcionamiento del sistema creado y así crear un entorno seguro en el que el servidor podría ofrecer un servicio sin miedo a verse afectado por los diferentes ataques que se estudiarán.

ABSTRACT:

When some services are publicly offered on the Internet, exposure to any interaction is inevitable, either it being with honest intentions, seeking to undermine the service or trying to gain unauthorized access to data stored on the server. Therefore, in a service aimed at the entire community, security is of vital importance. It is necessary to be able to distinguish interactions aimed at obtaining data from the service from interactions that seek either to hinder the service or to access unauthorized data.

One way of attempting to resolve this issue is through firewalls, preventing this malicious traffic from reaching the applications that provide the service as far as

possible. IPTABLES will be used in this project as firewall, found in every Linux-based distribution computer to detect and contain the most frequent attacks on servers exposed to the Internet for providing services to everyone.

Attacks of the utmost importance will be studied and launched against a simulated computer that provides a service. An IPTABLES-based firewall will be configured to be able to mitigate or fully eradicate (if possible) the attack and thus create a secure environment in which the server could offer a service without fear of being affected by the different attacks found in this study.

Contenidos

1 Introducción

Vivimos en una época en la que el uso de las nuevas tecnologías avanza a pasos de gigante, cada vez más usamos la tecnología para solucionar problemas del día a día, convirtiéndola prácticamente en una forma de vida. Esto conlleva increíbles avances y mejoras en la calidad de vida de las personas, pero también mayores riesgos, debido a la exposición que sufren nuestros datos personales al estar guardados en sistemas que podrían ser inseguros a priori.

Debería ser conocimiento general el hecho de que la seguridad en el campo de la informática es algo imposible de garantizar, tantas son las vulnerabilidades conocidas de los sistemas informáticos y tantas son las que quedan por conocer, que genera una incertidumbre sobre hasta qué punto se puede estar seguro depositando datos en un servidor.

En el mundo de la informática, cuando se ofrece un servicio público, siempre cabe la posibilidad de que alguien intente mermar el servicio o intente recabar datos a los que no está autorizado a acceder. Es por esto por lo que es de máxima prioridad tener un sistema de seguridad para asegurar que los datos guardados por el servicio, y el servicio en sí, tengan el menor número de problemas posible.

Por otro lado, el abuso del uso del servicio de forma malintencionada puede provocar el colapso de este servicio, haciendo que deje de funcionar, por lo que en la medida de lo posible hay que intentar que esto no ocurra.

Hoy en día se encuentran diversas herramientas para asegurar los servicios. Este estudio se va a centrar en una de ellas, los cortafuegos. Estos se encargan de poner medidas para que, dentro de lo posible, el tráfico de la red cuya intención sea maligna, no llegue a causar ningún daño real.

1.1 Estado Actual

En la actualidad las organizaciones debido a su exposición al ofrecer un servicio al público en internet necesitan algo más que una estructura de red simple. Por otro lado, dicha estructura de red estará completamente expuesta a intentos de ataques por parte de terceros, para esto se cuenta con diferentes sistemas de defensa.

Los cortafuegos son uno de los mencionados sistemas que se usan para frenar posibles ataques que busquen menoscabar el estado de un servicio. Las organizaciones encuentran a su disposición muchas ofertas de diferentes tipos para poder defenderse de ciberataques, tanto en cortafuegos hardware como software. Las diferencias básicas se encuentran en que a veces un negocio no necesita de un sistema hardware que incluya preinstalado el software del cortafuego, sino que conviene más instalar un software de propósito general, ya sea por coste o complejidad.

A la hora de elegir un cortafuego siempre se debe tener en mente las necesidades del sistema y la organización, puesto que todos los cortafuegos ofertados tienen puntos fuertes y débiles, por ejemplo, los sistemas de la compañía ‘Cisco’ son conocidos por su confiabilidad mientras que los de la compañía ‘SonicWall’ lo son por su más asequible apartado económico.

Todas las empresas ofertan por lo general similares ventajas, pero a diferentes escalas dependiendo obviamente del valor económico del cortafuego elegido, ventajas como pueden ser:

- Seguridad para sistemas de tamaños variables.
- Estudios sobre posibles amenazas para los sistemas que se deben proteger.
- Control sobre el acceso a las aplicaciones.
- Ancho de banda de transmisión de datos a través del cortafuego.

Hoy en día en el mercado se pueden encontrar ofertas de cortafuegos que oscilan entre los 800€ y los 10.000€ para organizaciones de tamaño medio, siempre dependiendo del sistema que se deba defender.

Varias opciones pueden ser apropiadas para las necesidades de una organización, solo se debe encontrar la que más se adapte a dichas necesidades. En este documento se va a operar con el cortafuego 'IPTables' encontrado gratuitamente en todas las distribuciones de Linux de forma que es accesible a toda organización.

1.2 Objetivos

El objetivo de este estudio es proveer unos conjuntos de reglas basadas en IPtables, que sean capaces de garantizar el correcto funcionamiento de un sistema ante la posible amenaza de un ataque por parte de un software automatizado o de un atacante físico con propósitos malignos.

Por otro lado, realizar este proyecto tiene varios objetivos intermedios para el alumno como son:

- Profundizar en el funcionamiento de un cortafuego como es IPtables con muchos de sus posibles módulos que le brindan amplias capacidades y formas de operar.
- Obtener el conocimiento necesario para operar con redes de computadores en máquinas virtuales.
- Conseguir conocimiento en sistemas operativos orientados a la penetración como es Kali Linux
- Aprender sobre técnicas ofensivas de seguridad.
- Obtener conocimiento sobre vulnerabilidades en sistemas y como prevenir daños en sistemas de forma que no se generen problemas en la máquina.

1.3 Competencias específicas cubiertas.

Con este proyecto se tiene la finalidad de alcanzar las siguientes competencias específicas referentes a la mención de tecnologías de la información:

- TI02: Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

Con la realización del proyecto el alumno deberá aprender a trabajar con sistemas de bajo coste y hacer uso de máquinas virtuales para el despliegue de los sistemas necesarios para realizar las pruebas convenientes. De forma que se presupone un aprendizaje sobre despliegue, explotación y utilización de sistemas de bajo coste y manteniendo la calidad de los sistemas.

- TI04: Capacidad para seleccionar, diseñar, desplegar, integrar y gestionar redes e infraestructuras de comunicaciones en una organización.

Al realizar este proyecto se debe trabajar con redes de computadores, en este caso la simulación se realizará mediante máquinas virtuales con el gestor *VirtualBox*, esto garantiza que el alumno sea capaz de desplegar sistemas dentro de redes y hacer uso de ellas sin ninguna clase de problema.

- TI07: Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.

El objetivo último del trabajo como se vio en el punto anterior es hacer uso de un cortafuego para garantizar la seguridad de un sistema dentro de una red, durante el estudio se ha tenido que llevar a cabo una gran investigación de las vulnerabilidades y herramientas que explotan las mismas para poder garantizar la seguridad del sistema que se intenta defender.

1.4 Aportaciones

La necesidad de elegir un cortafuego para defender un servicio proporcionado por una pequeña o mediana organización puede ser una ardua tarea, debido al estudio que se debe realizar para averiguar la viabilidad de los incontables productos que se tienen a disposición en el mercado en la actualidad.

Garantizar la seguridad de un servicio y los datos que alberga es un punto primordial en la existencia de un servicio y en especial de la organización, puesto que una vulnerabilidad no controlada y explotada por algún atacante, puede llevar a la organización no solo a perder dinero por los daños al servicio, sino a ser sancionados por el incumplimiento de algunas de las leyes de regulación con respecto a la información en servicios informáticos.

Este proyecto ofrecerá un acercamiento a un sistema de protección sin coste alguno y con grandes capacidades para las pequeñas y medianas organizaciones. Por otro lado, ofrecerá a los administradores de sistemas que trabajen para las organizaciones, un sistema fácilmente regulable y accesible telemáticamente en caso de necesitarlo para solucionar otras incidencias con total disponibilidad.

1.5 Desarrollo

Este proyecto se ha llevado a cabo de manera iterativa, donde en cada iteración se mostraban los avances y problemas encontrados. Durante las reuniones con el tutor se revisaban las soluciones propuestas y se establecían nuevos objetivos para la siguiente iteración.

En primera instancia, se llevó a cabo un estudio de los ciberataques más comunes en la actualidad. Además, se refrescaron conocimientos sobre el cortafuego IPTables y sus posibilidades frente a algunos ciberataques.

Posteriormente se realizó la instalación y configuración de las máquinas virtuales y su estructura de red para llevar a cabo el proyecto. Se tuvo que llevar a cabo una familiarización con la distribución de Kali Linux para su uso.

Una vez estaba todo preparado para la realización del proyecto, se comenzó a experimentar con los diferentes ataques y se estudiaron sus efectos en la máquina atacada.

Cuando se habían llevado a cabo las suficientes pruebas con los ciberataques, se comenzó con la implementación de las reglas de IPTables para la mitigación de estos.

Por cada ciberataque mitigado, se estudiaban los resultados obtenidos y se planteaban posibles mejoras en el sistema de mitigación.

Por último, se implementaron las posibles mejoras y se realizaron nuevamente las pruebas para comprobar su funcionamiento.

2 IPtables

En este capítulo de la memoria se verá lo que es un cortafuego, con especial mención a las IPTables y algunos de sus módulos más útiles.

2.1 Cortafuegos (Firewall)

Un cortafuego o Firewall es uno de los sistemas de seguridad de las redes informáticas cuya finalidad es evitar accesos no autorizados y conexiones fraudulentas

Los cortafuegos actúan de filtro, examinan todo el tráfico que pasa por una máquina y lo contrastan con unas reglas bajo una política de acceso definidas por el administrador del sistema.

Debemos entender que un cortafuego no tiene por qué constar solamente de una máquina, sino que se puede componer de varios elementos, entre ellos los hosts bastión (máquinas que conectan zonas internas con externas por lo general y se deben proteger), Proxies (intermediario de peticiones de recursos entre cliente y servidor), etc.

2.2 IPtables

IPtables es una utilidad de consola que permite, a los administradores de un sistema, hacer uso de las capacidades en filtrado de paquetes encontradas en Netfilter (framework del kernel de Linux que permite realizar múltiples acciones, entre ellas el filtrado de paquetes de red). Sin embargo, en este trabajo no se hará distinción entre IPtables y Netfilter, simplemente nos referiremos de ahora en adelante a todo este sistema como IPtables.

IPtables es una herramienta estructurada en tablas que permite definir y modificar las cadenas que contienen las reglas para el tratamiento de paquetes de red. Estos paquetes se examinan secuencialmente y se contrastan con las reglas existentes en las cadenas de las diferentes tablas. Existen 5 cadenas principales:

- **Prerouting**: se evalúan paquetes en el momento previo a su enrutamiento.
- **Postrouting**: se evalúan paquetes justo después de decidir su enrutamiento.
- **Input**: se evalúan los paquetes que se reciban para ser entregados localmente, es decir, en la máquina que los está evaluando.
- **Output**: se evalúan los paquetes salientes de la máquina local.
- **Forward**: se evalúan paquetes que se han enrutado pero que no tienen como destino la máquina que los está evaluando actualmente.

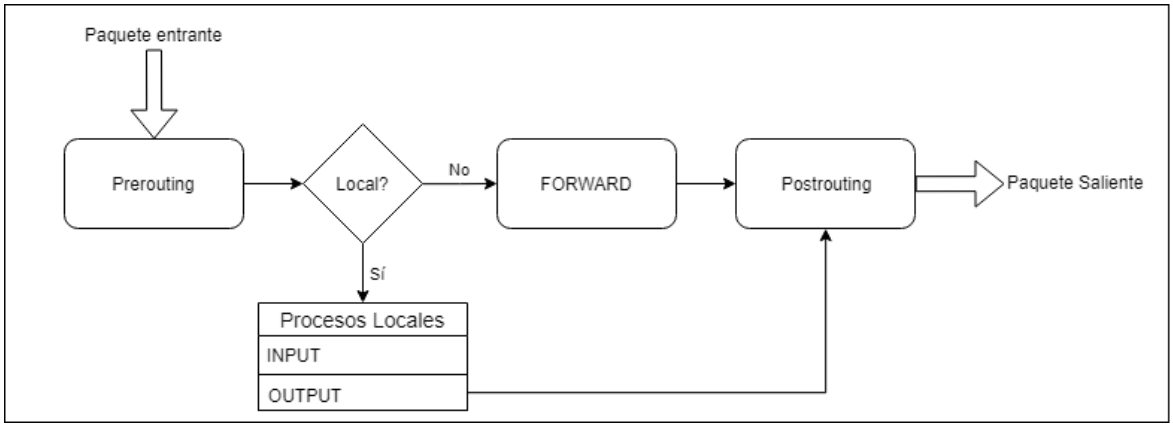

Ilustración 1: IPTables flowchart para paquetes entrantes a una máquina.

Cada paquete que circule por la máquina pasa por las tablas del cortafuego y por alguna de las cadenas que se encuentren dentro de las mismas, hasta que se da una de las siguientes situaciones:

- Una de las reglas coteja el paquete y decide finalmente si se acepta o desestima (ACCEPT o DROP).
- Una de las reglas usa la llamada RETURN para volver a la cadena padre en caso de haber sido llamada previamente.
- Se alcanza el final de la cadena, en cuyo caso, si la cadena había sido llamada por otra se retorna a la cadena de origen o se aplica la política por defecto de la cadena.

2.3 Módulos

IPTables permite el uso de módulos de extensión para realizar la evaluación de paquetes entrantes y salientes de una máquina de la forma más completa y eficiente posible. Estos módulos se activan con la opción '-m' o '--match' seguida del nombre del módulo que se va a usar. Dependiendo del módulo, se habilitan diferentes funcionalidades relacionadas con el módulo en uso. Se permite el uso de varios módulos a la vez y también el uso del operador *'!'* delante de la declaración para invertir el sentido del operando en las reglas usadas.

Entre estos módulos encontramos algunos de distinguida importancia en el desarrollo de este proyecto como son:

2.3.1 Recent

El módulo *'Recent'* permite crear listas dinámicas de direcciones IP y cotejar los datos de paquetes entrantes con el contenido de dicha lista de diversas formas.

Por ejemplo, se puede usar para crear una lista con las direcciones IP de los clientes que intenten acceder a un puerto específico, y desechar todos los paquetes futuros sin tener en cuenta otras reglas.

Entre las opciones disponibles más comunes para este módulo encontramos:

--name	Indica el nombre de la lista a usar con el comando, si no se especifica se usa *'DEFAULT'.*
--set	Añade la dirección fuente del paquete a la lista designada, en caso de ya existir, actualizará la entrada en la lista.
--rcheck	Comprueba si la dirección fuente del paquete se encuentra actualmente en la lista.
--update	Comprueba si la dirección fuente del paquete se encuentra, igual que la opción anterior, pero esta vez, si se encuentra, actualiza la marca de tiempo de la entrada en la lista.
--remove	Si la dirección fuente del paquete se encuentra en la lista, la elimina.
--seconds X	Comprueba si la dirección fuente del paquete se encuentra en la lista y su última aparición fue hace menos de X segundos. Requiere el uso de '--update' o '--rcheck'.
--hitcount X	Comprueba si la dirección fuente del paquete se encuentra en la lista y se han recibido X o más paquetes. Se puede usar con la opción '--seconds' para acotar aún más el filtro. Requiere el uso de '--update' o '--rcheck'.

2.3.2 Limit

El módulo *'Limit'* establece un contador de frecuencia sobre el número de veces que un paquete es aceptado en el sistema. Una regla que use este módulo se ejecutará hasta que se alcance el límite especificado.

Para el módulo limit se encuentran 2 opciones principales:

--limit X	Opción que nos sirve para representar el límite establecido por el usuario. X se establece como un valor numérico con la opción de añadir los sufijos '/second', '/minute', '/hour' o '/day' para establecer un promedio (por defecto se establece en '3/hour').
--limit-burst Y **(capacidad de ráfaga)**	Opción que establece el número de paquetes coincidentes con la regla a los que se aceptan antes de activar el contador establecido en la opción 'limit', el contador decrementa a cada paquete coincidente con la regla y se recarga en 1 cada vez que el límite establecido en la opción 'limit' se alcance (hasta un máximo de **Y)**. Su valor por defecto es 5.

Para entender mejor este módulo debemos tener en cuenta lo siguiente, el parámetro establecido en la opción limit denota el máximo de actuación de la regla, es decir, siendo el valor por defecto '3/h' en una regla de aceptación, la regla permitirá el paso de 3 paquetes por hora. Las reglas con el módulo limit se complican al entrar en juego el segundo parámetro.

El parámetro '–limit-burst' representa una ráfaga de paquetes que se aceptan antes de lanzar el límite que se establece en el parámetro –limit, es decir, si el parámetro '—limit-burst' está establecido a cinco en una regla de aceptación, esto significa que los primeros cinco paquetes que se cotejen con esta regla, se aceptaran directamente y una vez aceptados, se comienza a aceptar en la medida que indique el parámetro '–limit'. Por otro lado, cada paquete que llegue reduce en uno la capacidad de la ráfaga hasta llegar a cero, esta capacidad se va recuperando en uno cada 1/X (siendo X el valor de la opción –limit) intervalos de tiempo que no se reciban paquetes para cotejar.

En un ejemplo completo se supone la regla de IPTables siguiente:

iptables -A INPUT -p icmp --icmp-type 8 -m limit --limit 4/h –limit-burst 10 -j ACCEPT

La anterior regla acepta cuatro paquetes de protocolo ICMP por hora y dispone de una ráfaga de diez de estos paquetes, esto significa que los diez primeros que lleguen se aceptarán directamente y a partir de ahí se aceptarán cuatro por hora. La capacidad de la ráfaga ha pasado de diez a cero y como se explicó en el párrafo anterior, se irá recuperando uno a uno cada quince minutos (1/4 de hora) que no se coteje ningún paquete con esta regla. Esto quiere decir que, si pasa una hora completa sin recibir ningún paquete correspondiente a esta regla, la capacidad de la ráfaga estará en 4 de nuevo y los próximos 4 paquetes que lleguen se aceptarán directamente.

2.3.3 Conntrack

Conntrack es un módulo que permite acceder a un mayor número de opciones en lo que al seguimiento de conexiones establecidas se refiere. Este módulo se suele combinar con otros complementarios como son ‘connlimit’, ‘connrate’, etc.

Entre sus opciones más importantes vemos:

‘--ctstate State’	Indica el estado de la conexión en el parámetro ‘State’ (posibles estados INVALID, ESTABLISHED, NEW, RELATED, SNAT y DNAT).
***‘--ctproto* Proto’**	Indica el protocolo a filtrar en el parámetro ‘Proto’.
‘--ctorigsrc’	Dirección de origen con la que cotejar.
‘--ctorigdst’	Dirección de destino con la que cotejar.
‘--ctreplsrc’	Dirección de origen de la respuesta para cotejar.
‘--ctrepldst’	Dirección de destino de la respuesta para cotejar.

2.3.4 Hashlimit

El módulo 'hashlimit' de IPTables es un módulo muy útil en lo referente a filtrado de paquetes por grupos, el funcionamiento es muy similar al módulo 'limit' explicado anteriormente, pero con el añadido de que permite realizar un seguimiento de las conexiones establecidas.

La principal ventaja de este módulo es que el filtrado se puede realizar por agrupaciones de distintos tipos, como por ejemplo de direcciones IP, lo cual da la opción de realizar un filtrado para X paquetes durante Y tiempo por Z grupo de clientes.

El módulo hace uso de tablas de sistema para llevar un seguimiento de las conexiones establecidas y así poder controlar si un paquete debe ser aceptado o desestimado en una situación determinada.

Entre las opciones más comunes para este módulo se encuentran:

--hashlimit-upto X [requerida]	Opción que indica el valor máximo de paquetes por unidad de tiempo coincidentes con la regla, se especifica como un número con la posibilidad de añadir un sufijo de tiempo.
--hashlimit-above X [requerida]	Opción que indica el número de paquetes por unidad de tiempo mínimos para que la regla comience a realizar el filtrado especificado. Se especifica como un número con la posibilidad de añadir un sufijo de tiempo. Si se utiliza esta opción no es necesario usar 'upto' y viceversa.
--hashlimit-name nombre [requerida]	Indica el nombre de la entrada en la tabla de sistema que se va a usar/crear.
--hashlimit-burst X	Opción que establece el número de paquetes coincidentes con la regla a los que se acepta antes de activar la limitación de las opciones 'above' y 'upto'. Funciona igual que la opción 'burst' del módulo limit.
--hashlimit-mode modo	Una lista de al menos uno de los parámetros a tener en cuenta en la regla en caso de querer usar seguimiento de conexiones, por ejemplo, 'srcip', 'dstip','srcport', etc.
--hashlimit-htable-max X	Número máximo de entradas en la tabla de sistema usada por la regla, se recomienda establecer este parámetro lo suficientemente alto como para no influir en el tráfico normal de la máquina.

3 Estructura

En este capítulo de la memoria se explicará la arquitectura de red usada en las simulaciones que veremos más adelante en este documento.

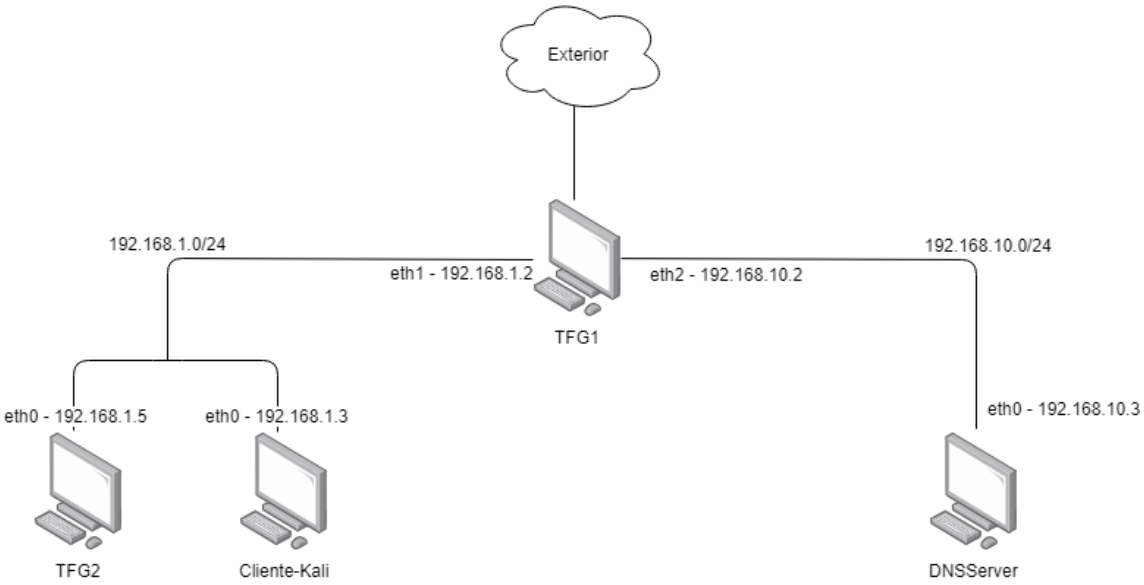

Ilustración 2: Estructura de red.

Para la estructura general contamos con 4 máquinas virtuales en el entorno VirtualBox para Windows, cada una con la siguiente estructura:

- **Máquina 'TFG1'**: Esta máquina virtual (Centos 6.10) simula la existencia de un Router, es decir, hace de cortafuego principal, habilita la conexión con el exterior para las máquinas de las subredes que se conecten a él. Posee 3 interfaces de red (eth0, eth1 y eth2). eth0 conecta con una red de tipo NAT para permitir el acceso al exterior, eth1 conecta con la primera subred privada donde se encuentra el servidor que vamos a defender y eth2 conecta con una segunda subred de tipo privada donde se aloja el servidor DNS.
- **Máquina 'TFG2'**: Esta máquina virtual (Centos 6.10) será nuestro servidor, donde se montarán los servicios necesarios sobre los que simular los ataques. Posee 1 sola interfaz de red que lo conecta a la subred privada del Router.

- **Máquina 'DNSserver'**: Esta máquina virtual (Centos 6.10) aloja nuestro servidor de nombres (DNS), puesto que es más seguro y eficiente tener uno propio en todas las subredes, además de ser una práctica común. Posee solo 1 interfaz de red que se conecta a la segunda subred privada del Router.
- **Máquina 'Cliente-kali'**: Esta máquina virtual (Kali Linux 2019.1) representa nuestro atacante, se encargará de simular todos los ataques, su situación habitual será la de la red interna donde se encuentre el servidor, puesto que los ataques normalmente se intentan realizar desde dentro, es decir, debemos situarlo al mismo nivel que nuestro servidor y suponer que el atacante ya ha hecho un estudio previo que le capacite para realizar los ataques.

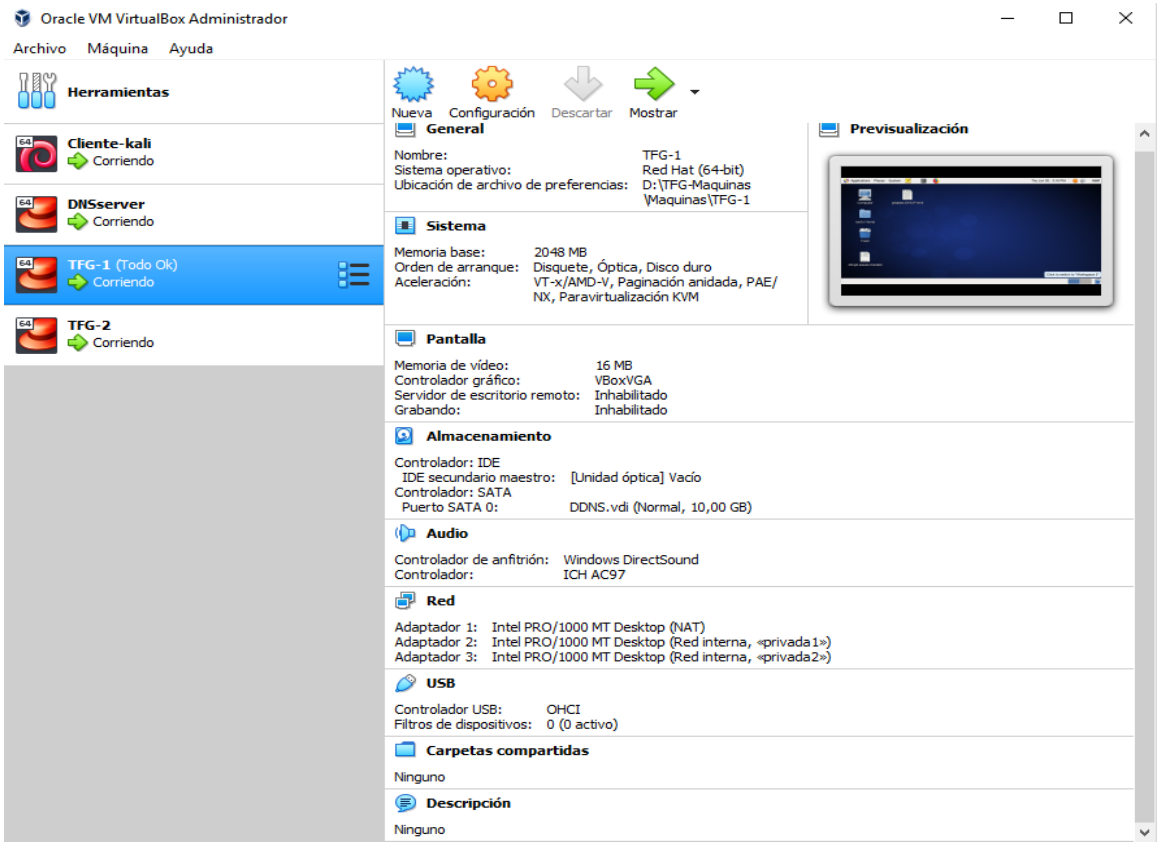

Ilustración 3: Máquina TFG1 - datos.

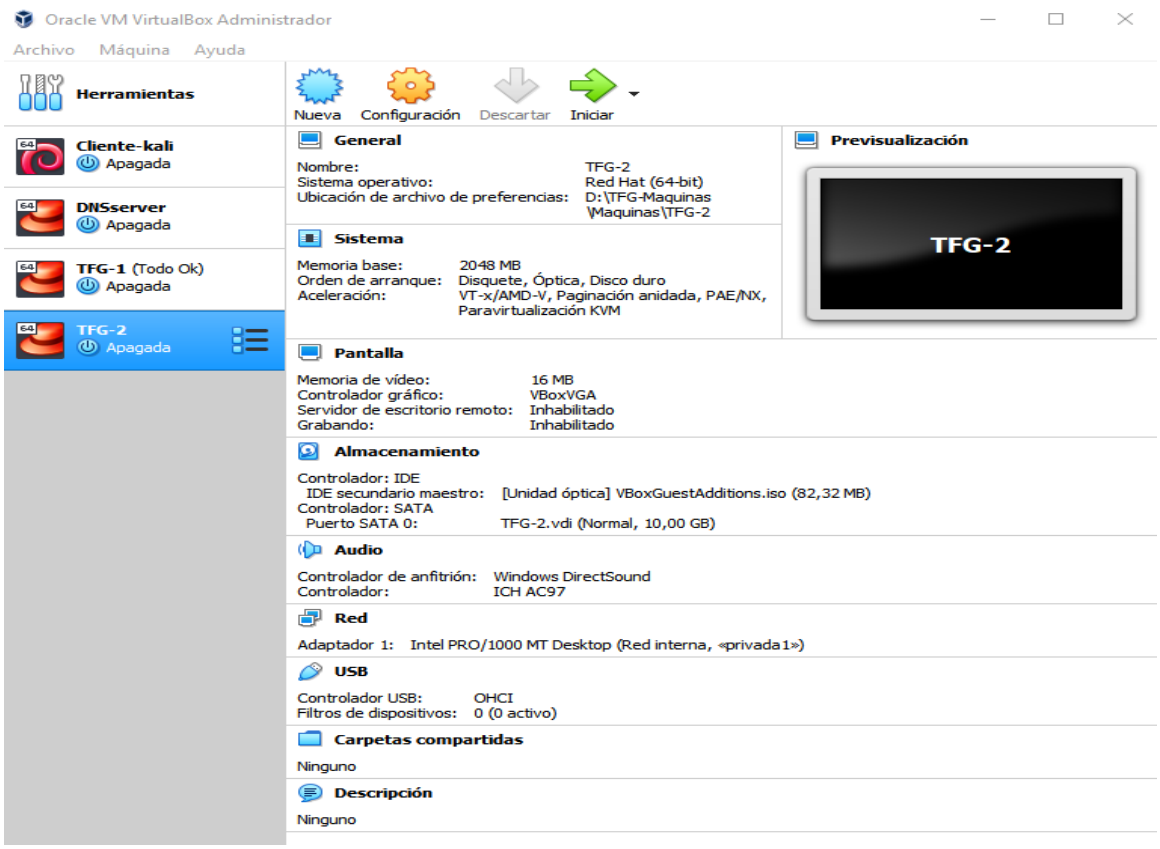

Ilustración 4: Máquina TFG2 - datos.

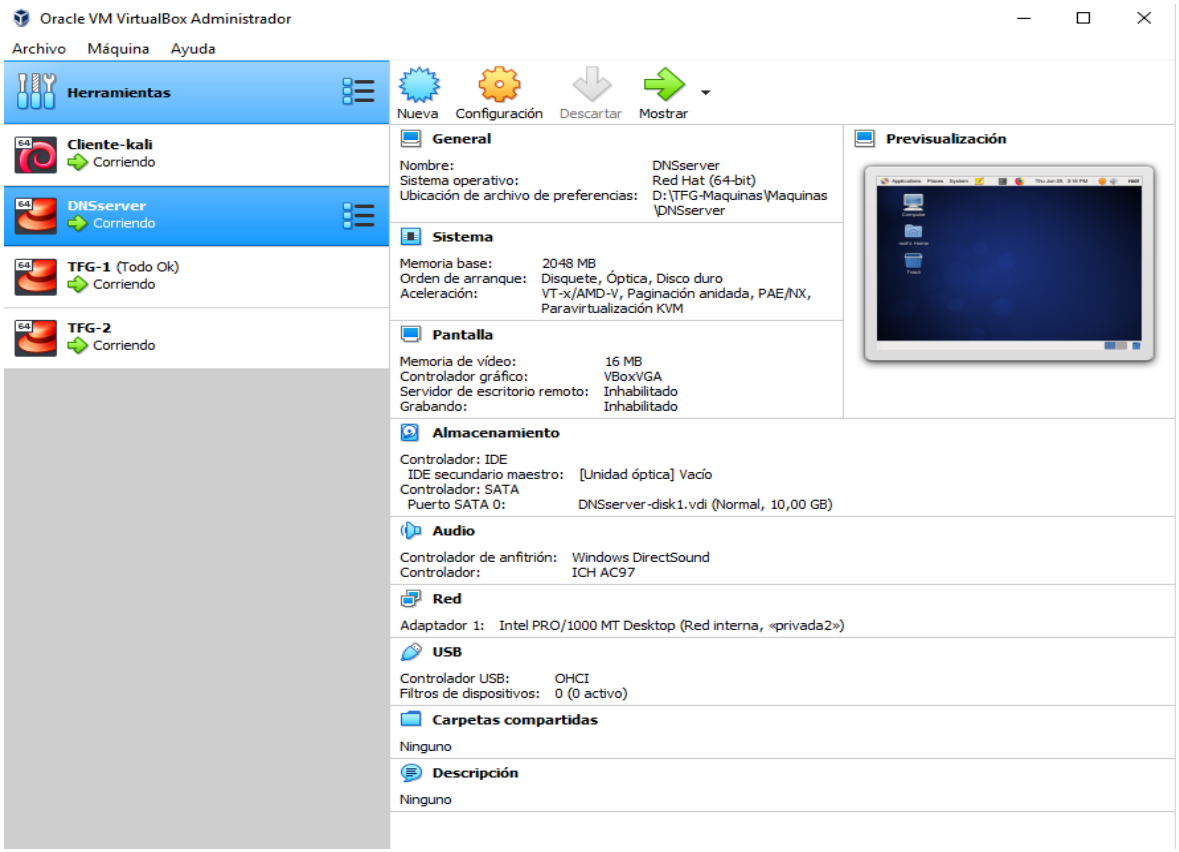

Ilustración 5: Máquina DNSServer - datos.

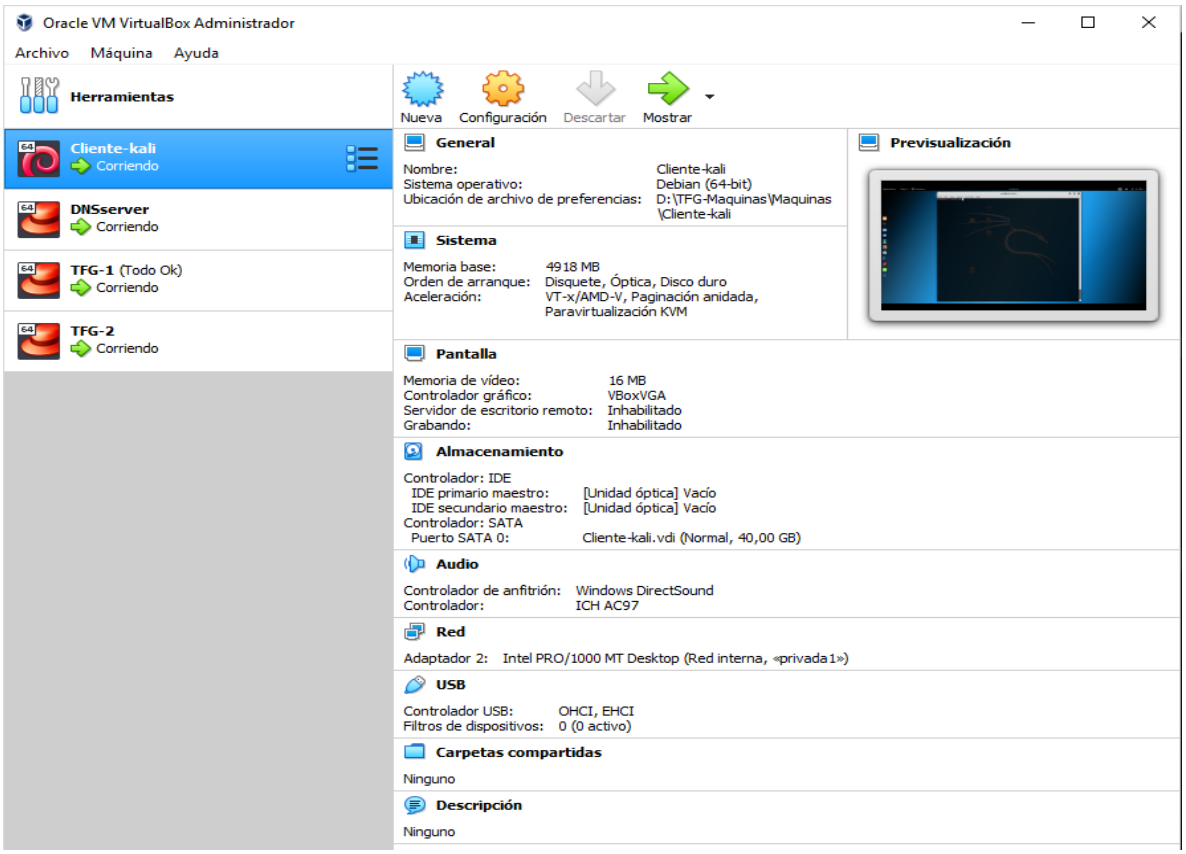

Ilustración 6: Máquina Cliente-Kali - datos.

4 Distribución Kali

4.1 Qué es Kali Linux

Kali Linux es una de las muchas distribuciones del sistema operativo de código abierto “Linux”, en este caso destinada a la realización de pruebas de penetración en sistemas (Pentesting) y auditorías de seguridad avanzadas. Para ello, Kali proporciona cientos de herramientas orientadas hacia diversas tareas de seguridad de la información, como pueden ser pruebas de penetración, investigaciones de seguridad, informática forense, etc.

La versión de Kali Linux usada para este trabajo es la 2019.1, la versión más reciente de la distribución en el momento de realización de este. Esta distribución en especial nos proporciona una serie de ventajas como, por ejemplo:

- Uso libre, sin coste alguno para el usuario.
- Mas de 600 herramientas para realizar pruebas de penetración en sistemas.
- Repositorios de Git de código abierto.
- Compatible con FHS (Filesystem Hierarchy Standard), permitiendo a los usuarios la fácil localización de ficheros binarios, librerías, etc.
- Núcleo del sistema personalizado con los últimos parches para inyección en caso de necesitarlos.
- Sistema completamente personalizable para poder cambiar partes del diseño en caso de necesitarlo.

4.2 Conclusión

Kali Linux es una distribución que no se recomienda a usuarios poco experimentados con Linux, pero debido a su orientación, a estar específicamente creada para colmar los requerimientos de los profesionales del sector y a la ventaja que supone tener muchas herramientas para pruebas ya instaladas. Se convierte en una distribución idónea para este proyecto.

5 Rastreador de puertos

Para poder hacer uso de otras herramientas de testeo, al principio siempre es necesario tener el conocimiento de la estructura de la red y lo que esta alberga. Para ello se hace uso de rastreadores de puertos. Los rastreadores de puertos proporcionan información de inestimable valor que podría ser usada tanto para fines honrados como para fines nocivos.

La herramienta elegida para realizar la simulación en este documento ha sido NMap.

5.1 NMap

NMap es una herramienta de código abierto creada originalmente para Linux, pero que se ha convertido en la actualidad en una herramienta multiplataforma.

Su uso principal hoy en día se encuentra en la evaluación de la seguridad de sistemas informáticos, basándose en envío de paquetes predefinidos a equipos y estudiando su respuesta frente a ellos.

La herramienta NMap es prácticamente una herramienta indispensable para toda persona que se dedique a la administración de sistemas, puesto que proporciona un amplio sistema de detección de puertos y servicios con una gran variedad de posibilidades que van desde escaneos específicos a la aplicación de scripts propios o genéricos con distintos fines.

5.2 Funcionamiento.

NMap funciona en línea de comandos de Linux (Aunque también posee interfaz gráfica) y viene instalado por defecto en la distribución “Kali” usada para esta simulación. Vale con usar el comando ***<nmap –version>,*** para ver su versión, plataforma y compilación en la consola.

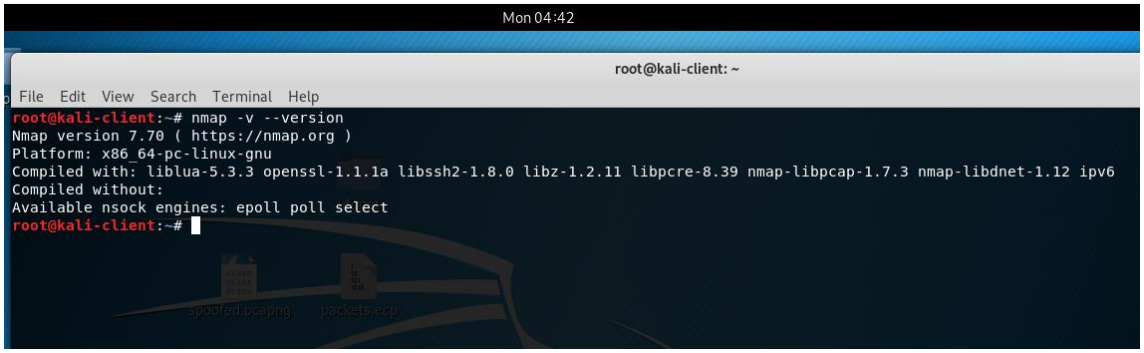

Ilustración 7: Versión NMap.

El comando NMap tiene la siguiente estructura:

- Nmap [tipo de escaneo] [opciones] {objetivo del escaneo}

Esta estructura permite realizar escaneos a medida de lo que necesitemos, por ejemplo, si se quisiera ejecutar un barrido de la red a base de ping (protocolo ICMP), valdría con ejecutar el comando (Como se ve en la ilustración 8):

- Nmap -sn <red a escanear>

Y si se quisiera escanear una máquina específica para ver sus servicios y puertos disponibles, valdría con ejecutar el comando a secas pasando como objetivo del escaneo la dirección IP de la máquina.

- Nmap <IP objetivo>

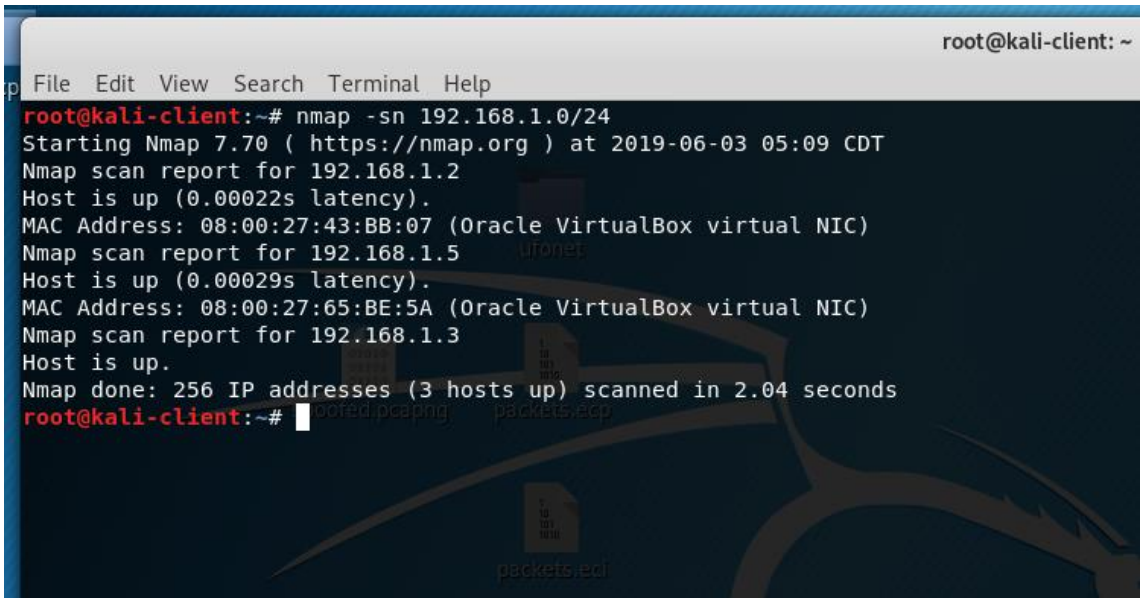

Ilustración 8: NMap - escaneo de red.

En este caso, el comando ejecutado con permisos de administrador en el sistema ha realizado las siguientes tareas:

1. Envio de paquetes “TCP SYN” al puerto 443
2. Envío de paquetes “TCP ACK” al puerto 80
3. Envío de paquetes “ICMP ECHO” y “TIMESTAMP REQUESTS”
4. Uso del protocolo ARP (Address Resolution Protocol) para descubrimiento de posibles vecinos en la red.

Debido a esto se encuentran las direcciones MAC y si es posible algunos servicios en la respuesta del comando.

NMap dispone de varias técnicas de descubrimiento de hosts en una red y los parámetros pueden ser modificados a merced de la persona que lo ejecute, pero siempre es importante conocer y entender el funcionamiento de las técnicas usadas por la herramienta para poder aplicarlas correctamente.

Algunas técnicas de NMap que pueden ser útiles incluyen las siguientes.

5.2.1 Trazado de rutas

NMap usa escaneo por ping (protocolo ICMP), por tanto, permite ver un trazado de la ruta desde el host que realiza el escaneo hasta el objetivo usando la opción "*--traceroute*". Esto da información importante puesto que podría delatar donde se encuentra una puerta de enlace o un servidor DNS que se haya pasado por alto, como se puede ver en la ilustración 9**.**

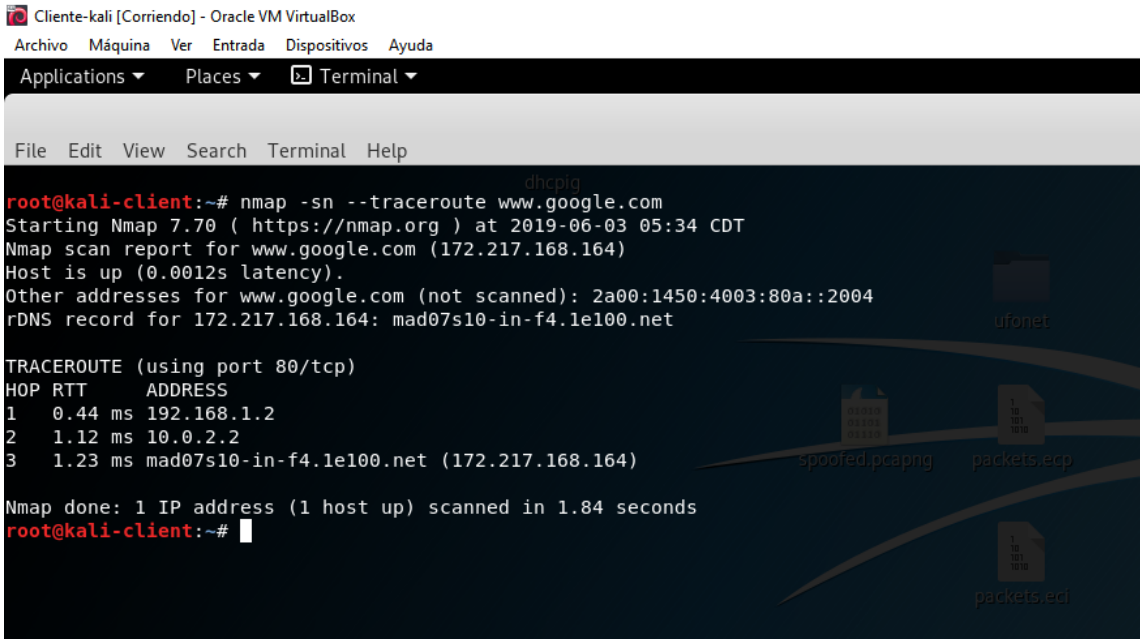

Ilustración 9: NMap - trazado de ruta.

5.2.2 Uso del motor de scripts (NSE, NMap scripting engine)

NMap dispone de un motor de scripts que se puede activar durante el escaneo mediante ping (protocolo ICMP) para obtener información adicional que puede ser de ayuda. Para hacer uso del NSE con los escaneos mediante ping simplemente haremos uso de la opción "*—script <fichero, directorio, categoría>*".

Los scripts del NSE conforman una de las características más potentes de la herramienta NMap, permite que los usuarios creen y compartan pequeños programas codificados en 'Lua' (potente y ligero lenguaje de programación para scripts), con el objetivo de automatizar tareas.

Como se ve en la siguiente ilustración se hace uso de un script que intenta enumerar los nombres de los hosts DNS intentando sacarlos por "*fuerza bruta*" (adivinando subdominios comunes).

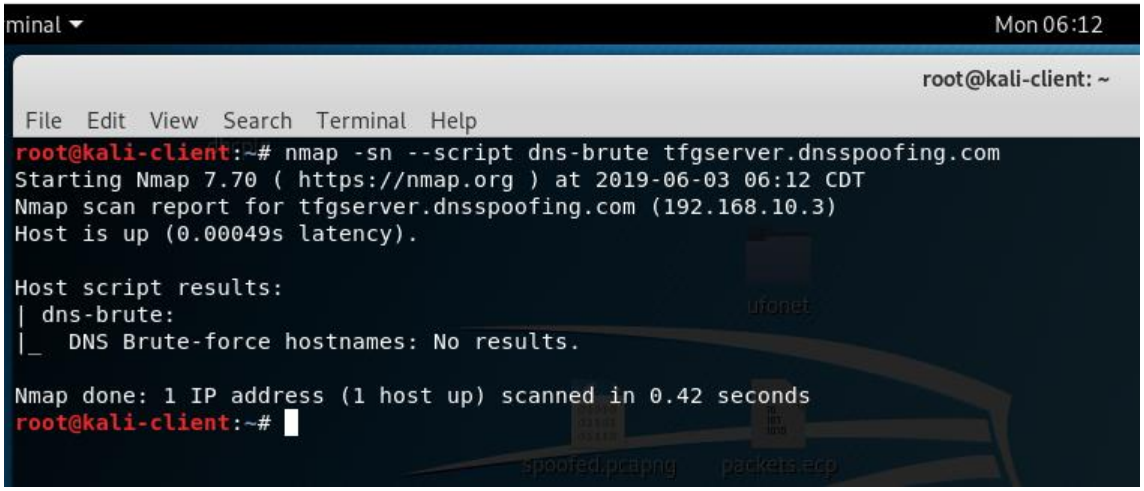

Ilustración 10: NMap - Scripts.

5.2.3 Listado de puertos

NMap permite escanear una red completa en búsqueda de los hosts que la habitan, una vez se conocen, se puede identificar un objetivo y escanear los puertos de dicho objetivo para comprobar el estado y los servicios disponibles.

Para escanear los puertos de una máquina específica, bastará con dirigir el comando NMap a la dirección IP de la máquina, esto devolverá un listado con el estado de los puertos de dicha máquina.

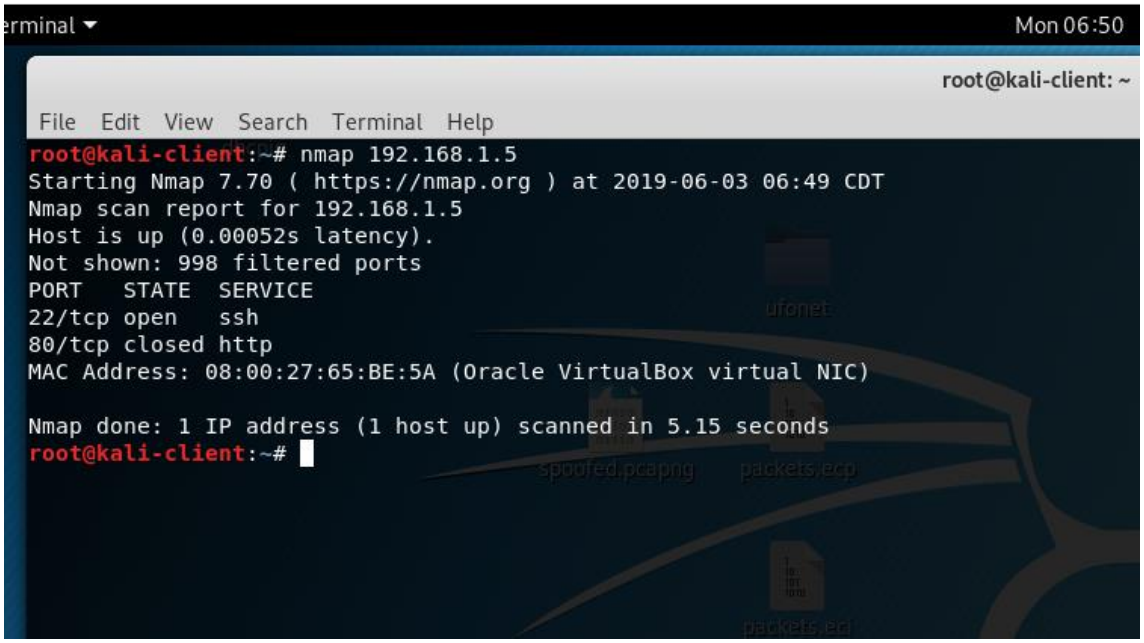

Ilustración 11: Puertos de una máquina.

Como se ve en la ilustración 11, los puertos encontrados tienen diferentes estados, esto se debe a que los puertos no siempre se encuentran abiertos y en escucha, tienen distintos estados habituales, los cuales pueden ser:

- **Abierto** (Open): Indica que hay un servicio en escucha en ese puerto.
- **Cerrado** (Closed): Indica que se recibieron las tramas, pero no hay ningún servicio escuchando en ese puerto.
- **Filtrado** (Filtered): Indica que no hay indicios de recepción de tramas por parte del objetivo y por tanto el estado no se pudo determinar, también podría indicar que las tramas están siendo rechazadas por algún filtro.
- **Sin Filtrar** (Unfiltered): Indica que las tramas fueron recibidas pero que no se pudo establecer un estado.

Cabe destacar que los escaneos de puertos también tienen parámetros ajustables mediante la opción "*-p*", como por ejemplo "*nmap -p80 <objetivo>*" para escanear solamente el puerto 80, o "*nmap -p100-1500 <objetivo>*" para escanear los puertos del 100 al 1500, o "*nmap -p smtp <objetivo>*" para escanear los posibles puertos del protocolo SMTP.

5.3 Explicación de la defensa.

NMap proporciona una serie de utilidades de gran valor para los administradores de sistemas, pero también una gran fuente de conocimiento a un atacante. Para poder realizar un ataque de cualquier característica siempre conviene conocer la estructura de red a la que vas a atacar, al igual que la situación de los servicios que deseas atacar. NMap nos permite conocer todos estos datos incluso antes de infiltrarnos. Una vez dentro todavía es más fácil llevar a cabo un escaneo con otros fines. Por esta razón a veces conviene más tener el servidor protegido de forma que no sea posible realizar un escaneo de puertos.

5.3.1 Procedimiento de la defensa.

Para la defensa de una máquina del escaneo de puertos de NMap se debe tener en cuenta lo explicado en los puntos anteriores, puesto que una vez se sabe cómo funciona la herramienta, será más fácil defenderse de ella haciendo uso de los módulos correspondientes de IPTables.

El objetivo en este caso es hacer que el “atacante” o la persona que esté intentando escanear los puertos de nuestra máquina, reciba como respuesta que todos los puertos de la máquina que está escaneando están filtrados. Para ello hay que configurar el cortafuego con las reglas necesarias para los siguientes casos:

1. Mantendremos todo el tráfico de circuito cerrado (Loopack) y habilitaremos las conexiones con protocolo de negociación a 3 vías (3-way handshake).
2. Descartaremos todos los paquetes inválidos por definición y paquetes entrantes a la máquina que puedan venir de direcciones falsificadas típicamente.
3. Bloquearemos la IP del “atacante” por un periodo de seguridad.
4. Crearemos entradas en el registro con los datos del escaneo de puertos recibido.
5. Permitiremos el resto de tráfico habitual para que la máquina siga funcionando con normalidad para los clientes “legítimos”.

5.4 Reglas de la defensa

A continuación, se exponen las reglas correspondientes a los puntos explicados en la lista del apartado anterior:

```
# Reglas de cadena INPUT
# Aceptar tráfico de Loopback y habilitar conexiones con
# 3-way handshake
-A INPUT -i lo -p all -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEP
# Descartar paquetes inválidos por definición y descartar
# paquetes de redes falsificadas típicamente, descartadas directamente y direcciones
# de broadcast.
-A INPUT -m state --state INVALID -j DROP
-A INPUT -s 0.0.0.0/8 -j DROP
-A INPUT -d 0.0.0.0/8 -j DROP
-A INPUT -s 192.168.0.0/24 -j DROP
-A INPUT -s 10.0.0.0/8 -j DROP
-A INPUT -d 255.255.255.255 -j DROP
-A INPUT -s 169.254.0.0/16 -j DROP
-A INPUT -d 239.255.255.0/24 -j DROP
-A INPUT -s 172.16.0.0/12 -j DROP
-A INPUT -s 127.0.0.0/8 -j DROP
-A INPUT -s 224.0.0.0/4 -j DROP
-A INPUT -d 224.0.0.0/4 -j DROP
-A INPUT -s 240.0.0.0/5 -j DROP
-A INPUT -d 240.0.0.0/5 -j DROP
# Bloquear la dirección del atacante durante 12h ~ 43200 segundos
# comprueba que la dirección atacante no se haya visto en las últimas 24h.
-A INPUT -m recent --name listaScanner --rcheck --seconds 43200 -j DROP
# Se añade el "atacante" a la lista y se añade una entrada con los datos al registro
-A INPUT -p tcp -m tcp -m recent --name listaScanner --set -j LOG --log-prefix "listaScanner:"
-A INPUT -p tcp -m tcp -m recent --name listaScanner --set -j DROP
# Permitimos la entrada de tráfico común (SMTP, DNS, HTTP, HTTPS y SSH)
-A INPUT -p tcp -m multiport --dports 25,53,80,443,22 -j ACCEPT
# Rechazamos todo tráfico no filtrado previamente.
-A INPUT -j REJECT
```

5.5 Resultado.

Una vez añadidas las reglas al cortafuego del servidor, en caso de ser objetivo de un escaneo de puertos, se vería en el registro de sistema una entrada que indicaría los datos del atacante y el escaneo, como vemos en la ilustración 12. Además, la IP del atacante será bloqueada durante 12h y todo paquete procedente de dicha dirección será rechazado.

El atacante por su parte solo será capaz de ver en el resultado del escaneo que todos los puertos de la máquina objetivo están filtrados sin posibilidad de explotar ningún servicio, aunque exista dentro de la máquina, exactamente como se ve en la ilustración 13.

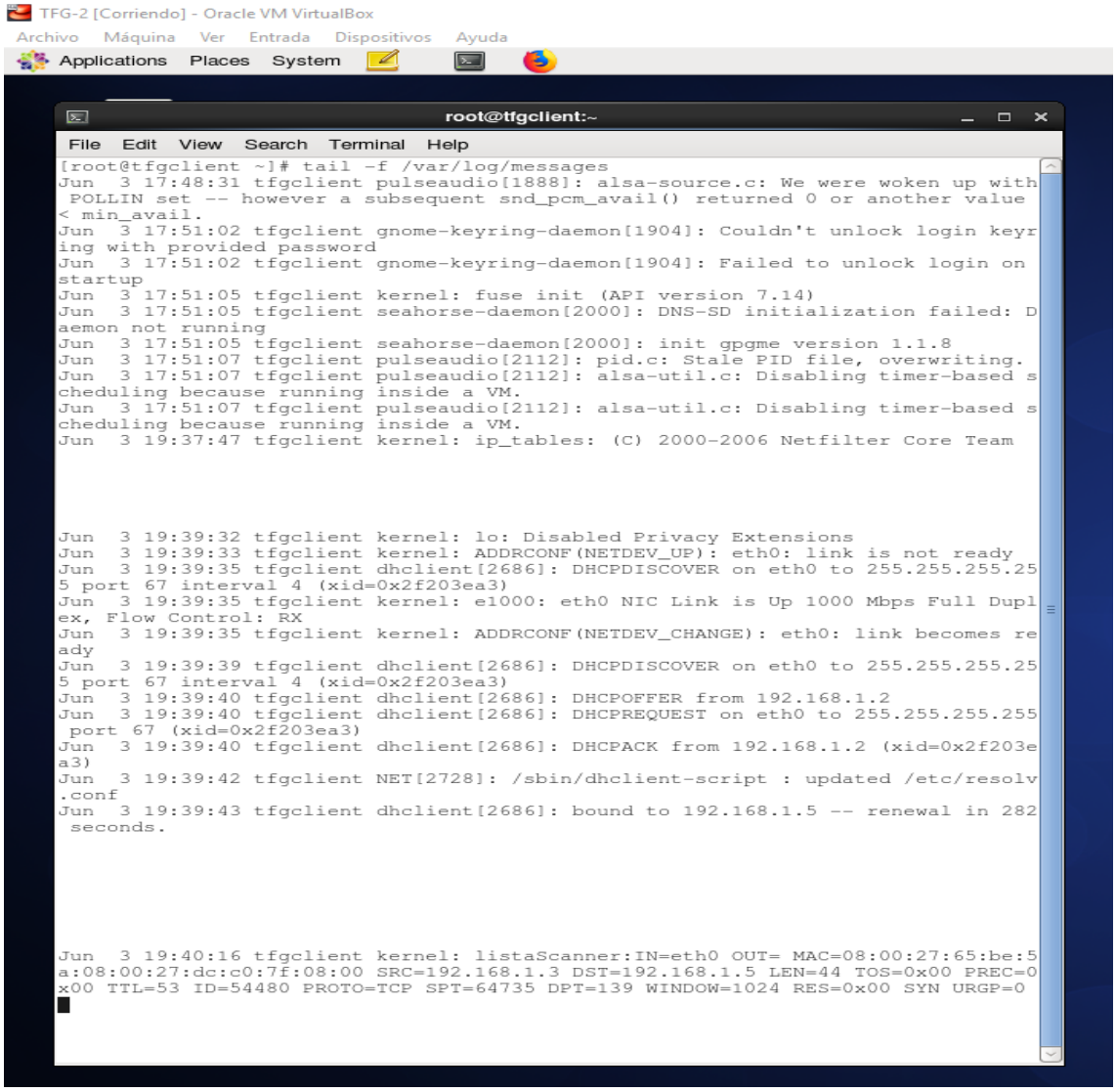

Ilustración 12: LOG listaScanner

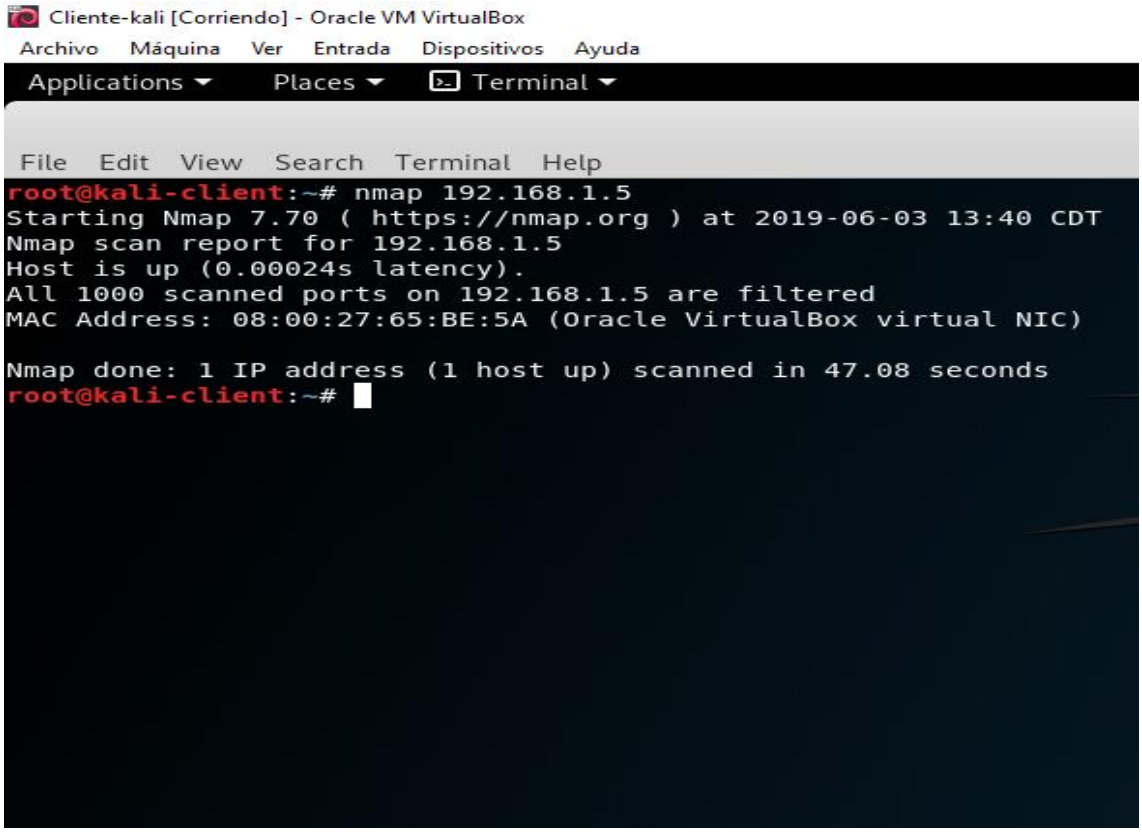

Ilustración 13: NMap - resultado.

6 Fuerza Bruta

6.1 Qué es un ataque de fuerza bruta

Los ataques de fuerza bruta son los métodos más populares para llevar a cabo cracking sobre contraseñas (descifrado de contraseñas), aunque no son su único propósito. Pueden ser utilizados para descubrir paginas ocultas y otro contenido dentro de aplicaciones web. Se basan en el principio de lanzar pruebas hasta que se obtiene el resultado esperado, estos ataques poseen tiempos de ejecución más elevados que el resto de los tipos de ataques informáticos, pero a su vez poseen también una probabilidad de éxito mayor.

Es habitual tener que realizar conexión a un servidor mediante protocolos como, por ejemplo, SSH (Secure Shell) para administrar la máquina y hacer cambios pertinentes. Un atacante puede intentar ganar acceso a dicha máquina y sus recursos e información haciendo uso de ataques de fuerza bruta.

En este tipo de ataques encontramos que las dos formas más comunes de realizar el ataque son haciendo uso de un diccionario o mediante generación de contraseñas con o sin uso de un patrón.

6.1.1 Usando un diccionario.

Las herramientas para realización de ataques de fuerza bruta en la mayoría de los casos admiten la recepción de ficheros de texto con un diccionario generado de posibles contraseñas o recursos. El objetivo en este caso sería probar sistemáticamente todas las palabras del diccionario con la esperanza de que alguna de estas coincida.

Los diccionarios pueden ser generados por el propio atacante en base a datos obtenidos sobre el objetivo del ataque o pueden ser diccionarios genéricos como los que se encuentran en la propia distribución de Kali Linux bajo el directorio *'/usr/share/wordlists/metasploit'* por ejemplo.

El hacer uso de un diccionario con palabras genéricas y específicas aporta versatilidad al ataque, debido a que muchas veces los usuarios simplemente utilizan contraseñas sencillas de recordar, aunque no sostengan ninguna relación con el usuario. El uso de un diccionario genérico en ese caso aportaría contraseñas comunes para probar, como por ejemplo ‘123456’, ‘toor’ o ‘admin’, ahorrando así una gran cantidad de tiempo al atacante.

6.1.2 Generación de contraseñas por fuerza bruta.

Este ataque es por así decirlo “el infalible”, probará todas las contraseñas posibles, sean cuantas sean. La idea de este ataque es que la herramienta del atacante genere todas las posibles combinaciones y las pruebe, comenzando con longitud de cadena 1 y aumentándola hasta dar con la contraseña correcta.

Debido a la forma de operar, es de suponer que contraseñas cuyo número de caracteres sea bajo, serán más fáciles de romper que las que posean un mayor número de caracteres

En este tipo de ataques normalmente se intenta acotar haciendo uso de un patrón con datos conocidos por el atacante (obtenidos mediante otros medios), de forma que este pueda predefinir que el recurso tenga de X a Y caracteres (mayúsculas o minúsculas), números, caracteres especiales, etc., Todo dependiendo del grado de conocimiento que tenga el atacante sobre la posible contraseña o recurso del objetivo.

6.2 Herramienta y simulación

La herramienta elegida para la simulación del ataque de fuerza bruta en este caso ha sido *'THC HYDRA'*, herramienta instalada por defecto en la distribución Kali Linux usada para el proyecto.

6.2.1 THC Hydra

THC Hydra (a partir de ahora Hydra, por su nombre común) es un descifrador de contraseñas, actualmente una de las mejores herramientas para realizar ataques remotos contra una gran variedad de servicios. Su principal ventaja es que es una herramienta paralelizada, es decir, permite realizar varias conexiones simultáneas con la intención de acelerar la prueba.

Hydra es una herramienta de consola, por tanto, para ver su versión e información sobre opciones bastará con lanzar el comando 'hydra -h*'* desde la consola de nuestra máquina virtual con Kali Linux. Ahí veremos todas las opciones disponibles con su información correspondiente. Entre las opciones más útiles se destacan:

-S	realiza una conexión SSL (Secure Socket Layer)
-s puerto	realiza una conexión al puerto especificado, útil si conocemos que el servicio se encuentra en un puerto distinto al por defecto.
-l login* o *-L fichero	realiza un intento de conexión con nombre de usuario especificado o con varios nombres de usuario si se usa la opción -L
-x MIN:MAX:CHARSET	realiza una generación de contraseñas por fuerza bruta siendo 'MIN' el número mínimo de caracteres, 'MAX' el número máximo de caracteres y 'CHARSET' el set de caracteres que se van a usar de patrón.

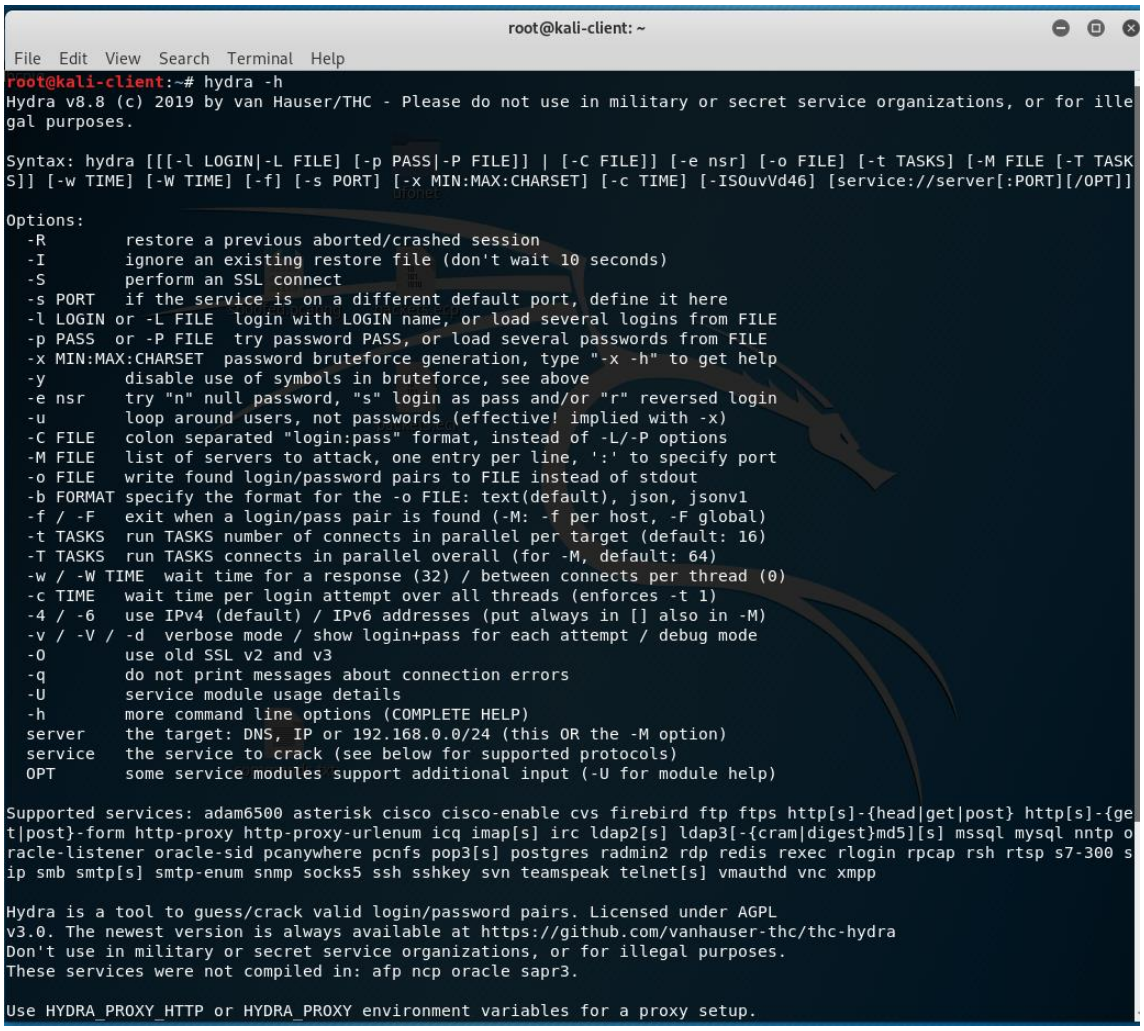

Ilustración 14: Hydra - información.

6.2.2 Simulación

Para la simulación de este ataque se ha optado por atacar al protocolo SSH, protocolo ampliamente usado para el acceso remoto a un servidor mediante un canal seguro de conexión, debido a que la información se encuentra cifrada en su totalidad. El objetivo es hacer uso de la herramienta Hydra mencionada anteriormente para conseguir la contraseña del usuario root del servidor y con ella, el acceso al servidor y sus recursos o información.

Para ello desde nuestra máquina atacante con Kali Linux se lanzará la herramienta Hydra con las opciones correspondientes. En este caso se probó a lanzarla tanto con un diccionario de palabras como generando automáticamente las contraseñas con la opción -x como se explicó en el apartado anterior.

El ataque consiste en enviar peticiones de conexión SSH a la máquina objetivo, con el nombre de usuario root y cada una de las contraseñas encontradas en el diccionario o generadas a partir del patrón.

Los resultados son los siguientes:

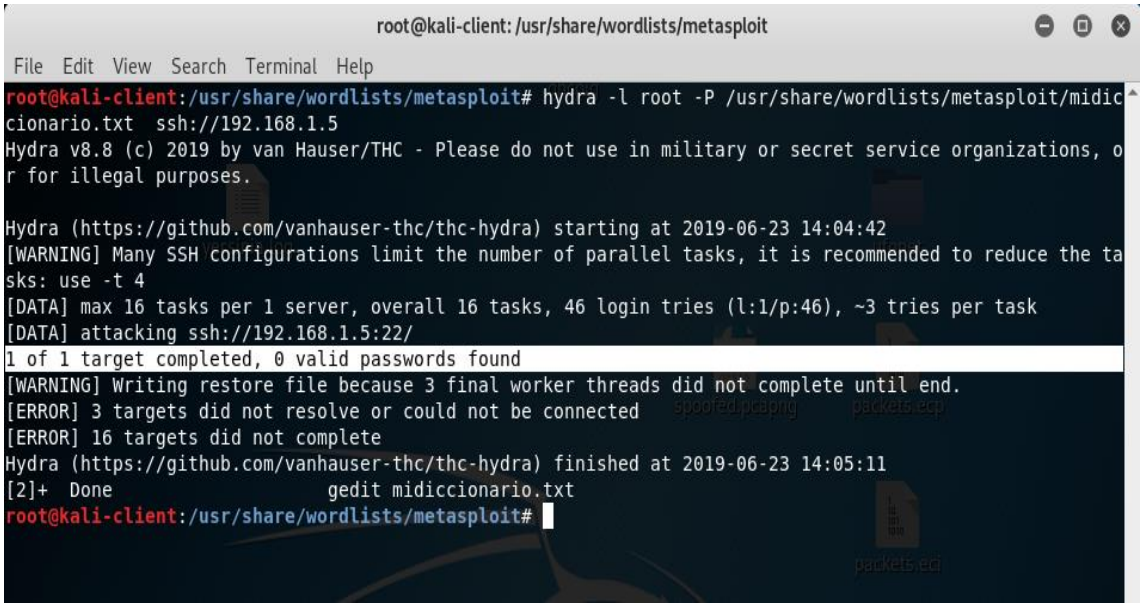

Ilustración 15: Ataque de fuerza bruta con diccionario.

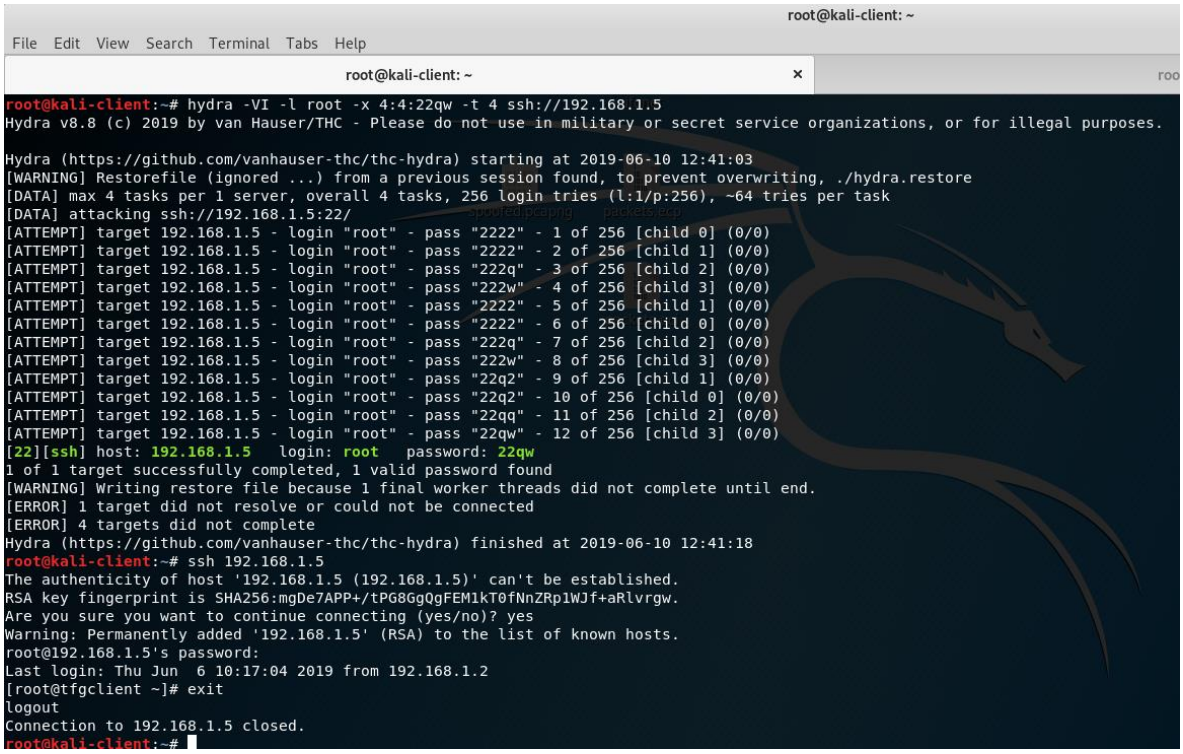

Ilustración 16:Ataque de fuerza bruta por generación de contraseñas

Como se ve en las ilustraciones 15 y 16, el ataque con el diccionario no encuentra la contraseña porque no es ninguna de las que se encuentra en el diccionario, pero el ataque de fuerza bruta con contraseñas autogeneradas a partir de un patrón dado por el atacante, sí que la encuentra. Hay que destacar que la contraseña ha sido elegida para garantizar el éxito del ataque en esta simulación y que una contraseña nunca debe ser tan simple como la que se ve en el resultado del ataque.

6.3 Defensa

Con los ataques de fuerza bruta se tienen varias opciones de defensa, como podrían ser:

- Restringir el acceso mediante SSH por IP, aunque esto puede no ser completamente útil puesto que las direcciones IP en un entorno real suelen ser dinámicas, así que la máquina que vaya a realizar la conexión puede variar su dirección IP.
- Cambiar el protocolo SSH de puerto puede ayudar, pero en un entorno donde un atacante puede poseer rastreadores de puertos, puede seguir siendo inseguro.
- Usar software de terceros para protegerse ante ataques de estas características, puede ser una opción viable, pero estos programas pueden ser fácilmente esquivados por un atacante con la experiencia suficiente.

El método más seguro es configurar reglas de cortafuegos que controlen el sistema y en caso de recibir demasiados accesos de protocolo SSH fallidos, bloquee el acceso dejando constancia en el sistema.

6.4 Reglas aplicadas y efecto.

Como se ve a continuación, se mantienen las políticas por defecto de las cadenas lo más cerradas posibles y se controla todo acceso SSH realizado a la máquina. Para ello, se registra toda conexión nueva realizada mediante protocolo SSH en una lista dinámica con su dirección de origen, si la misma dirección de origen realiza más de 3 intentos erróneos de conexión se registrará como intento de ataque y se eliminará toda conexión entrante proveniente de esta dirección fuente. Se encuentran las reglas aplicadas a continuación.

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:SSHLOG - [0:0]
#nuevas conexiones entrantes (SSH) registradas en lista dinámica (SSH_CONNTRACK)
-A INPUT -i eth0 -p tcp -m tcp --dport 22 -m state --state NEW -m recent --set --name SSH_CONNTRACK --rsource
#nuevas conexiones cuya Fuente se encuentra ya en la lista SSH_CONNTRACK
#y han realizado más de 3 accesos fallidos en la última hora se mandan a cadena SSHLOG
-A INPUT -i eth0 -p tcp -m tcp --dport 22 -m state --state NEW -m recent --update --seconds 3600 --hitcount 4 --
name SSH_CONNTRACK --rsource -j SSHLOG
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 22 -j ACCEPT
#SSHLOG → dejamos constancia del intento de conexión y desestimamos todos los demás intentos.
-A SSHLOG -j LOG --log-prefix "INTENTOSSH"
-A SSHLOG -j DROP
COMMIT
```

6.5 Conclusión

Esta clase de ataques, aunque fáciles de detectar y no muy aparatosos para mitigar, constituyen un gran peligro para cualquier sistema. Es por ello por lo que no solo se debería aplicar lo expuesto en el apartado anterior para mitigarlos, sino que se recomienda completamente el hecho de tener unas políticas de seguridad adecuadas para prevenir el acceso no autorizado. Puesto que las contraseñas son consideradas la parte más débil en un sistema, se hace necesaria la concienciación de los usuarios para mantener las contraseñas en secreto, elegir contraseñas robustas para los sistemas y, en caso de ser necesario, incitar a los usuarios a realizar un cambio periódico de contraseñas.

7 Inanición de servidor DHCP

En este capítulo se verán los efectos de un ataque de inanición sobre un servidor DHCP (Dynamic Host Configuration Protocol) y como mitigar el efecto de este ataque, además de otros consejos al respecto de su identificación y defensa.

7.1 DHCP

El “dynamic host configuration protocol” o DHCP por sus siglas en inglés es un protocolo de administración de redes basado en redes UDP/IP (User datagram protocol/Internet Protocol), en las cuales un servidor, denominado “servidor DHCP”, asigna dinámicamente direcciones IP y otros parámetros de configuración de red a los clientes. En caso de no existir este tipo de servidores se encontraría el administrador de la red en la situación de tener que configurar manualmente cualquier máquina que se conecte a la red.

Los servidores DHCP son utilizados tanto en redes de gran tamaño (nivel universitario o incluso regional) como en redes domésticas, habitualmente los Routers se encargan de actuar de servidores DHCP.

7.1.1 Anatomía del protocolo DHCP

El protocolo DHCP hace uso de protocolo UDP para configurar las conexiones, y actúa en los puertos 67 (puerto destino del servidor) y 68 (puerto usado por el cliente).

Las operaciones realizadas por el protocolo DHCP se basan en 4 fases (más una fase de liberación de conexión) las cuales son:

1. **Discovery**: El cliente lanza una petición con mensaje DHCPDISCOVER a la dirección de difusión (255.255.255.255 o la dirección de difusión especifica de la subred).
2. **Offer**: Cuando un servidor DHCP reciba dicho mensaje, preparará una oferta al cliente con una de las direcciones IP disponibles (en caso de haberlas), reservando dicha dirección IP para ese cliente en este momento y enviándole un mensaje DHCPOFFER.
3. **Request**: Una vez el cliente recibe la oferta del servidor, responderá a ese mensaje con un mensaje DHCPREQUEST por difusión, para que en caso de haber más de un servidor DHCP todos sepan que dirección ha seleccionado el cliente y en caso de no ser elegidos, deshacer la reserva de dirección y así esté disponible para otros clientes.
4. **Acknowledgement**: Cuando el servidor recibe el mensaje DHCPREQUEST por parte del cliente, el servidor le manda al cliente un mensaje DHCPACK en el que incluye datos referentes a los tiempos de conexión y otra conexión necesaria. Llegado este punto, la comunicación está terminada y el cliente configurará su interfaz de red con los parámetros acordados y sus comprobaciones pertinentes.
5. **Release**: Esta quinta fase no es estrictamente necesaria puesto que su propósito es mandar un mensaje al servidor (DHCPRELEASE) cuando el cliente se desconecte de la red, para que el servidor libere la dirección IP, la fase no es necesaria debido a que la máquina cliente no es capaz de predecir cuando el usuario lo va a desconectar de la red y por tanto no sabe si tiene que mandar el mensaje o esperar.

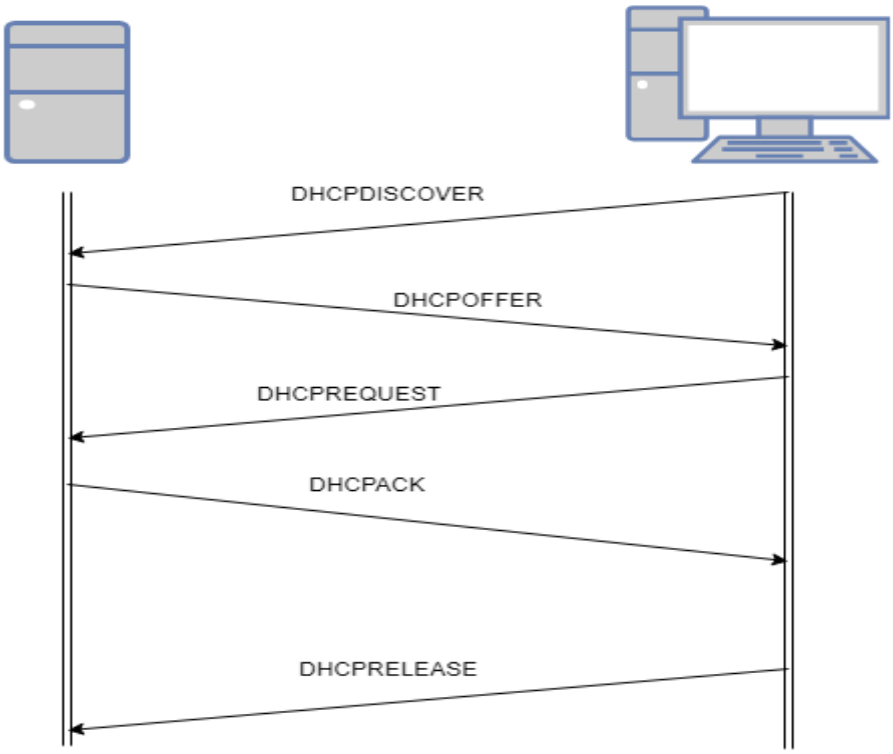

Ilustración 17: DHCP – comunicaciones

7.2 Simulación y herramientas usadas.

Un ataque de inanición de un servidor DHCP se basa en el principio de que las direcciones que posee el servidor para repartir entre los clientes son limitadas. Por otro lado, las direcciones que el servidor DHCP oferta a un cliente quedan reservadas durante un tiempo antes de que el servidor las vuelva a poner disponibles como se explicó en el apartado anterior.

El ataque comúnmente consiste en inundar el servidor DHCP a solicitudes de conexión (DHCPDISCOVER) para que este responda con las direcciones ofertadas (DHCPOFFER) y las reserve.

Esta clase de ataques por su cuenta no tienen demasiado valor. La clave está en acompañarlos de otras técnicas de ataque como pueden ser la creación de un servidor DHCP propio por parte del atacante de forma que una vez el servidor DHCP principal de la red está inundado, las peticiones de nuevos clientes sean respondidas por el servidor DHCP del atacante, permitiendo así al atacante espiar todo el tráfico, poniéndose a sí mismo como puerta de enlace de la red, o

simplemente engañando al cliente con redirecciones maliciosas a páginas mediante resoluciones falsificadas de DNS (Domain Name System).

La herramienta elegida para esta simulación ha sido 'DHCPig', aunque se debe mencionar especialmente la herramienta 'Yersinia', encontrada en la distribución Kali Linux por defecto y que demuestra lo sencillo que es realizar este tipo de ataques con solo seleccionarlo en la interfaz gráfica y lanzarlo.

7.2.1 DHCPig.

DHCPig es una herramienta avanzada de consola para realización de ataques de inanición de servidores DHCP. Su objetivo como ya se ha explicado es consumir todas las posibles IPs dentro de una red de área local (LAN) para así evitar que otros usuarios consigan sus parámetros de red entre otras cosas. La herramienta en este caso no se encontraba instalada por defecto en la distribución Kali usada, su descarga se realizó desde el repositorio remoto oficial de la herramienta que proporciona un fichero con el código escrito en Python de esta.

La elección de esta herramienta frente a las demás se debe a que es una herramienta más avanzada, por ejemplo, esta herramienta completa las 4 operaciones del protocolo DHCP de forma que, una vez terminado el ataque, si se accede al fichero de préstamos de direcciones del servidor DHCP, localizado en la ruta "*/var/lib/dhcpd/dhcpd.leases*" por defecto, se ven las concesiones de las direcciones a unas direcciones MAC inventadas por el atacante. Por otro lado, la herramienta posee varias opciones para forzar a los demás clientes que ya estaban en la red a soltar sus direcciones IP y así inundar completamente el servidor por ejemplo. Las opciones se muestran en la ilustración 18.

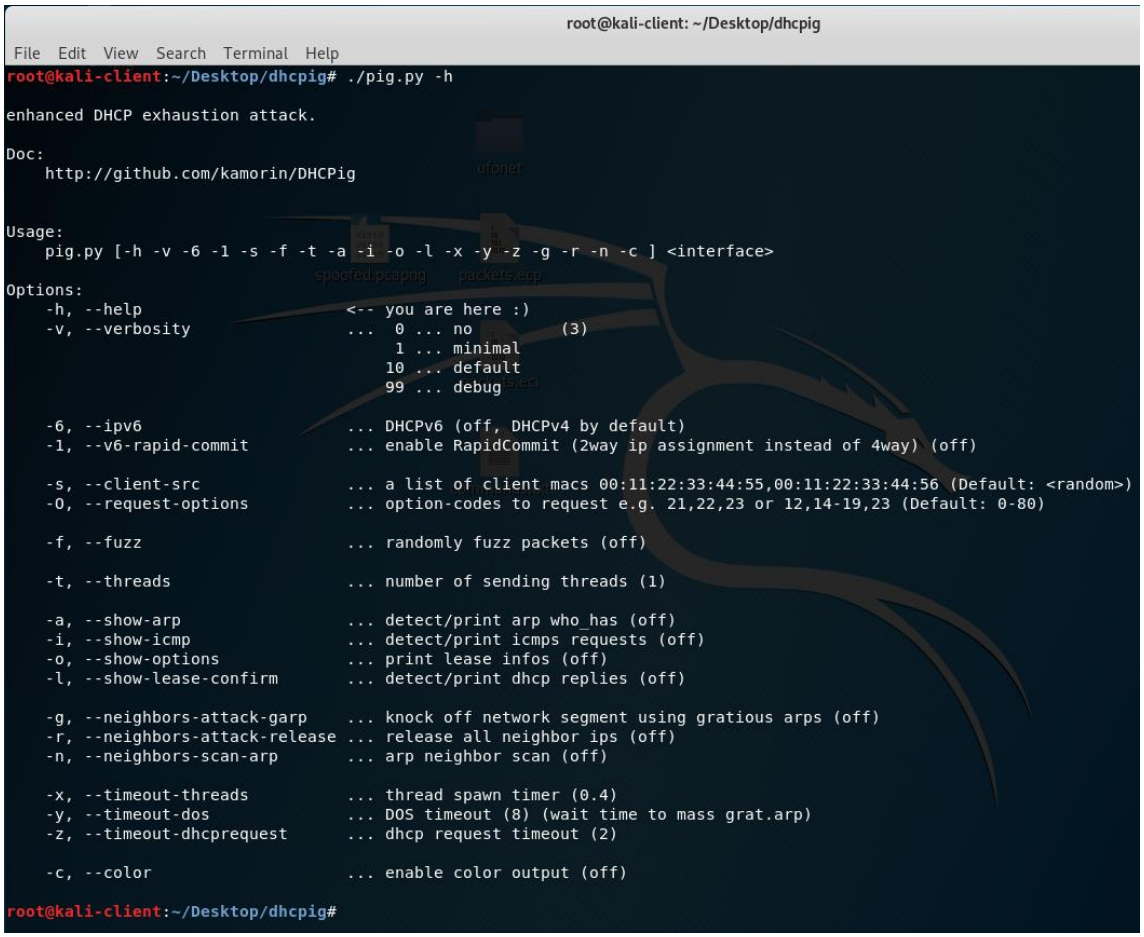

Ilustración 18: DHCPig – opciones.

7.2.2 Simulación.

Para realizar la simulación final del ataque, basta con lanzar la herramienta indicando únicamente la interfaz en la que debe actuar, como se ve en la ilustración 19.

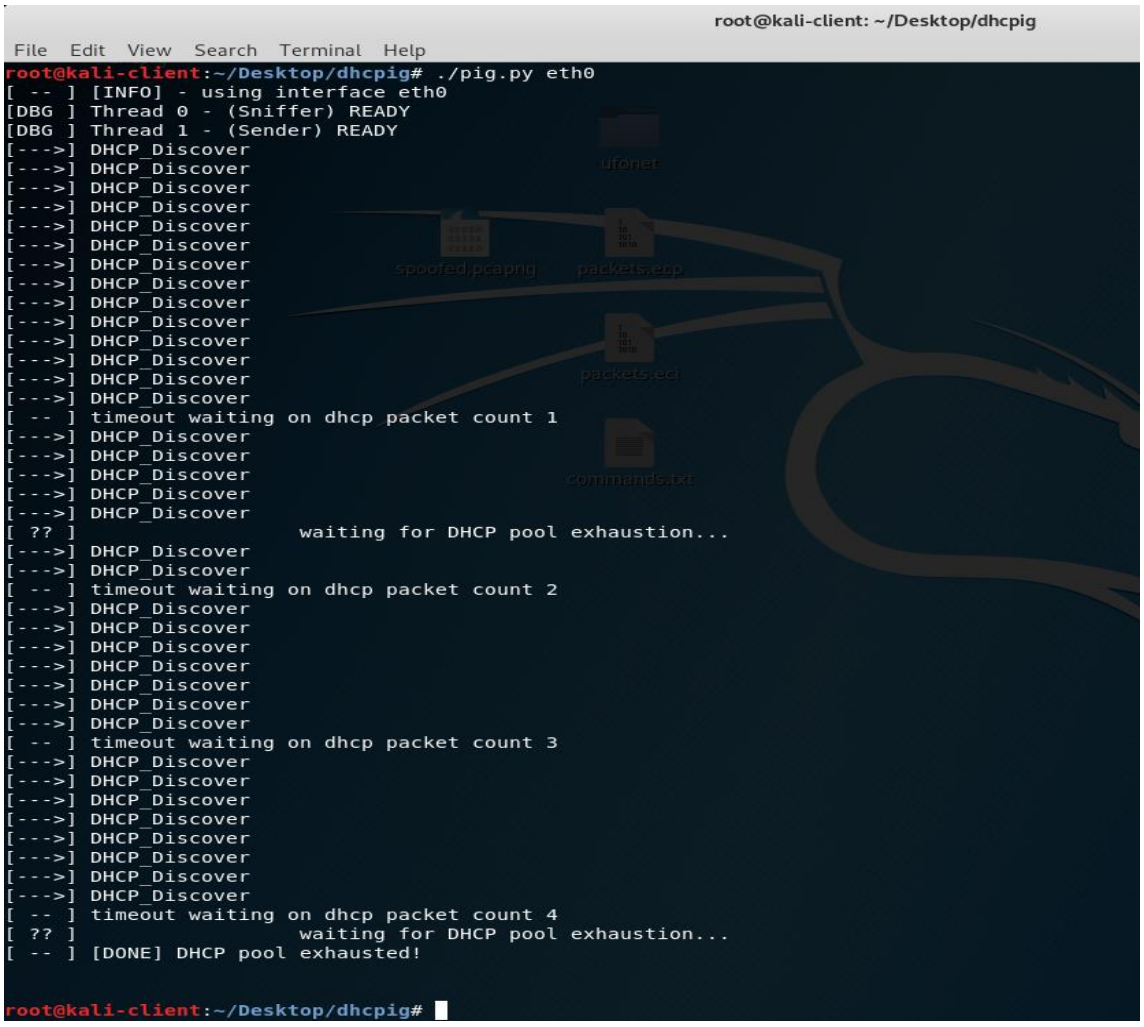

Ilustración 19: DHCPig - funcionamiento.

Como se ve en la consola, la herramienta funciona paralelamente (mediante hilos), simplemente envía los mensajes DHCPDISCOVER y espera respuesta para acabar cerrando la conexión, una vez ha terminado se muestra un mensaje explicando que ha acabado y si se busca el fichero de préstamos de direcciones IP del servidor, se verán todas las concesiones realizadas (Véase Anexo I para un pequeño extracto del fichero después del ataque).

7.3 Mitigación

Con esta clase de ataques se encuentra un gran problema a la hora de realizar la defensa, la herramienta falsifica direcciones desde las cuales lanza las peticiones al servidor DHCP, lo cual hace casi imposible que un cortafuego descifre si esa dirección es real o no. Lo único que se puede hacer en este caso desde el punto de vista de IPTables como cortafuego, es establecer un límite a las peticiones entrantes. En vista de que la herramienta lanza todas las peticiones secuencialmente y sin tiempo de espera entre una y otra, lo que se debe hacer es hacer uso del módulo 'limit' de IPTables para evitar que las direcciones disponibles se agoten en un tiempo mínimo y dar capacidad de respuesta al administrador del sistema atacado.

A continuación, se exponen las reglas elegidas para mitigar el ataque:

```
# Generated by iptables-save v1.4.7 on Mon Jun 10 12:07:55 2019
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
# Se limitan las peticiones entrantes DHCP a 1/sg y con limitador de ráfaga de 1
-A INPUT -i eth1 -p udp -m udp --sport 68 --dport 67 -m limit --limit 1/sec --limit-burst 1 -j ACCEPT
-A INPUT -p udp -m multiport --dports 67 -j DROP
# Se limitan las respuestas DHCP a 1/sg y con limitador de ráfaga de 1
-A OUTPUT -o eth1 -p udp -m udp --sport 67 --dport 68 -m limit --limit 1/sec --limit-burst 1 -j ACCEPT
-A OUTPUT -p udp -m multiport --sports 67 -j DROP
COMMIT
# Completed on Mon Jun 10 12:07:55 2019
```

7.4 Conclusión

Como se ha visto a lo largo del desarrollo de este capítulo, un ataque de inanición de DHCP es algo sencillo de llevar a cabo y puede tener consecuencias devastadoras en una red de área local y a mayor escala. Las posibilidades que le da al atacante de convertirse en dueño y señor de la red y todos los clientes conectados a ella, lo convierten en una amenaza de grandes proporciones, permitiendo al atacante espiar a todos los clientes a la vez, e incluso llevar a cabo técnicas de spoofing (suplantación de identidad) y phishing (obtención de información confidencial).

Con el solo uso de un cortafuego, será casi imposible frenar un ataque de estas características llevado a cabo por un atacante con cierto conocimiento, puesto que será capaz de burlar al sistema sin mucha dificultad. Sin embargo, se pueden poner reglas como las vistas en el apartado anterior para dar más tiempo a los administradores para darse cuenta de que están siendo atacados y tomar las medidas necesarias para acabar con el ataque satisfactoriamente.

7.5 Recomendaciones.

A pesar de no poder tener una solución completa para este tipo de ataques con solo IPTables como cortafuego, se debe destacar el hecho de que esta clase de ataques se pueden resolver de varias maneras como, por ejemplo:

- ‘rp_filter’: rp_filter es el nombre de un mecanismo del kernel de Linux, su nombre proviene de ‘reverse path filtering’ o filtrado inverso. Este mecanismo se puede usar para conocer si la dirección fuente de un paquete es real, es decir, si una máquina recibe un paquete, antes de aceptarlo se comprobará que la máquina de origen de este, es alcanzable a partir de la interfaz por la cual llegó el paquete. En caso de no ser alcanzable, el paquete simplemente se desestima. Este mecanismo podría ayudarnos a solucionar en parte el hecho de la falsificación de direcciones que envían el DHCPDISCOVER en el ataque.

- Scripting o creación de scripts: Este método consiste en elaborar un script en bash y hacer que es ejecute periódicamente (Cron). El objetivo sería comprobar que las direcciones IPs cedidas por el servidor DHCP sean legítimas, ya sea comprobando las direcciones mac con las que se solicita o comprobando si hay respuesta de esa dirección IP con otros protocolos (ICMP, por ejemplo).

Estos métodos pueden ayudar a solucionar el problema planteado por este ataque a costa de rendimiento en ambos casos, puesto que, a mayor número de comprobaciones realizadas, más lento se hace el filtrado, pero si se usan estos métodos junto con un cortafuego bien configurado, se reduce en gran medida el efecto de este tipo de ataques sobre una máquina.

8 Suplantación de identidad de servidor de nombres.

8.1 DNS.

Antes de poder comenzar a explicar el ataque en sí, se debe tener unas nociones básicas sobre el sistema de nombres de dominio y para que se usa de forma que así entendamos de mejor manera la existencia y relevancia de este tipo de ataques.

El sistema de nombres de dominio o DNS (Domain Name System) se encarga de traducir de nombres entendibles por los humanos a direcciones IP de máquinas que se encuentren asociadas a esos nombres. El objetivo es poder localizar y conectar estos equipos a gran escala, habitualmente se usa la comparación con una guía telefónica para las redes informáticas, puesto que estas siguen un sistema similar.

Este sistema garantiza un nivel de fiabilidad puesto que la dirección IP asociada a un nombre de dominio puede cambiar debido a varias razones sin significar un cambio en el nombre del dominio correspondiente. Además, para los humanos es más sencillo de recordar un nombre de dominio como puede ser *"www.youtube.com"* o "*www.google.com"* que sus respectivas direcciones IP.

La base de datos del sistema de nombres de dominio usa un sistema distribuido apoyado de un modelo Cliente-Servidor, donde los clientes, también llamados '*resolvers'*, solicitan información a los servidores de nombres, para luego comunicársela a las aplicaciones que harían uso de esa información.

El espacio de nombres de dominio se basa en una estructura en forma de árbol, como se ve en la ilustración 21. Comenzando en una ristra vacía denominada raíz y a partir de ahí se extenderían los demás nodos que representarían a los demás dominios existentes.

Los dominios se pueden dividir en subdominios, donde cada organización es la encargada de mantener su propio subdominio, esta subdivisión hace que, si cada nodo representa un sistema existente, la ristra en representación del nodo se

forma uniendo todos los nodos por los que se debe pasar para llegar hasta el nodo especifico desde la raíz.

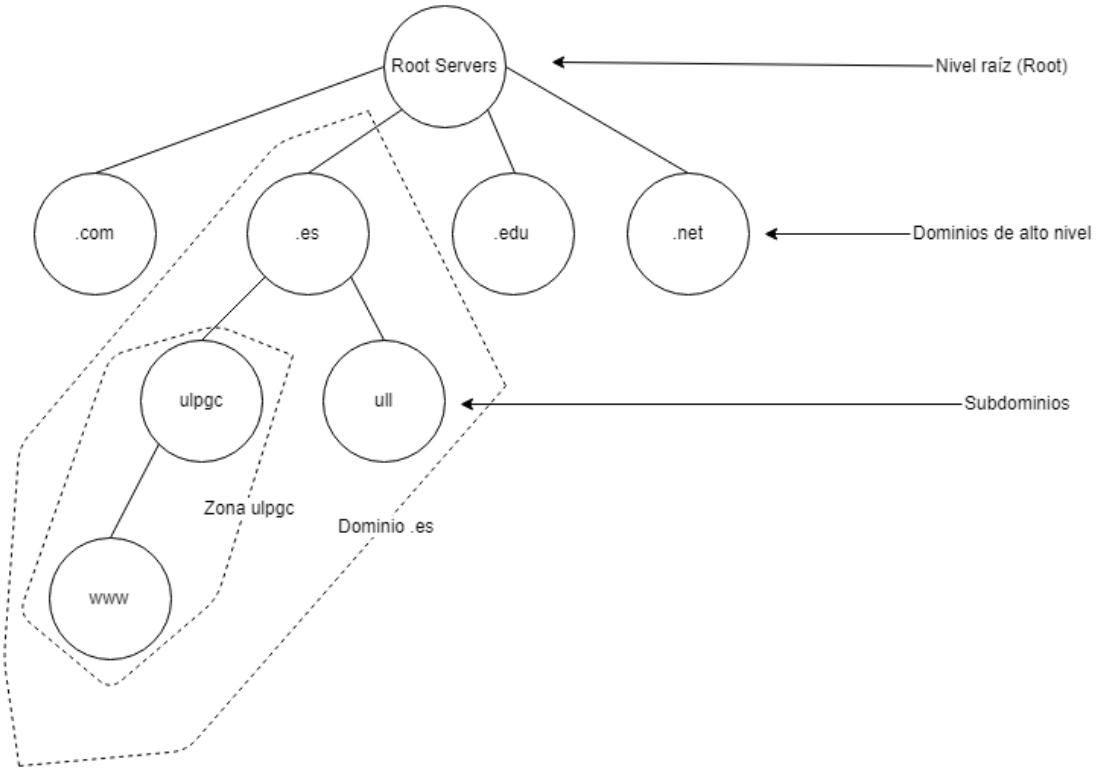

Ilustración 20: Jerarquía DNS

Debajo de los nodos raíz, en el segundo nivel de la estructura se encuentran los dominios de nivel superior (TLDs por sus siglas en inglés). Los cuales dividen los dominios en función del servicio que ofertan o de la región a la que pertenecen geográficamente, aunque se debe destacar que este planteamiento en realidad no se sigue demasiado.

Los servidores DNS realizan resoluciones de nombre de dominio de dos formas:

- **Recursiva:** El cliente solicita un dato al servidor y el servidor por su propia cuenta debe realizar las cuestiones pertinentes para devolverle al cliente el dato que precisa directamente.
- **Iterativa:** El cliente solicita un dato, pero el servidor no da una respuesta completa a la cuestión a menos que el dato se encuentre en su poder, en caso contrario simplemente se remite al cliente al próximo servidor que pueda ayudarle a encontrar la solución.

Los servidores de DNS llevan a cabo una práctica común para reducir la carga sobre los servidores, esta práctica se basa en guardar en una **memoria cache** los resultados de las cuestiones realizadas por los clientes. Si un cliente interroga en busca de un nombre de dominio o dirección que ya se ha buscado previamente, el servidor puede dar una respuesta inmediata sin tener que ir a buscar el dato nuevamente, este tipo de práctica se asocia con la existencia de un tiempo máximo de vida (TTL, Time To Live), establecido para esos registros en memoria de forma que pasado este tiempo, los datos se descartan y se tienen que refrescar de nuevo.

8.2 Man in the middle.

Un ataque Man in the middle (Mitm) o ataque de intermediario es un tipo de ataque cuyo objetivo es obtener la capacidad de leer y modificar información compartida entre 2 víctimas, procurando siempre que ninguna de las 2 víctimas se percate de la existencia de un intermediario indeseado.

Esta clase de técnicas son especialmente útiles en protocolos de intercambio de claves simétricas, especialmente cuando no se hace uso de ningún tipo de autenticación.

La forma de operar se puede ver a modo de guía en la ilustración 22, donde Ana y Berto intentan llevar a cabo un intercambio de claves para poder realizar conexiones seguras, pero un atacante (Carlos) intercepta y modifica sus claves para así quedarse con toda la información que intenten comunicarse Berto y Ana.

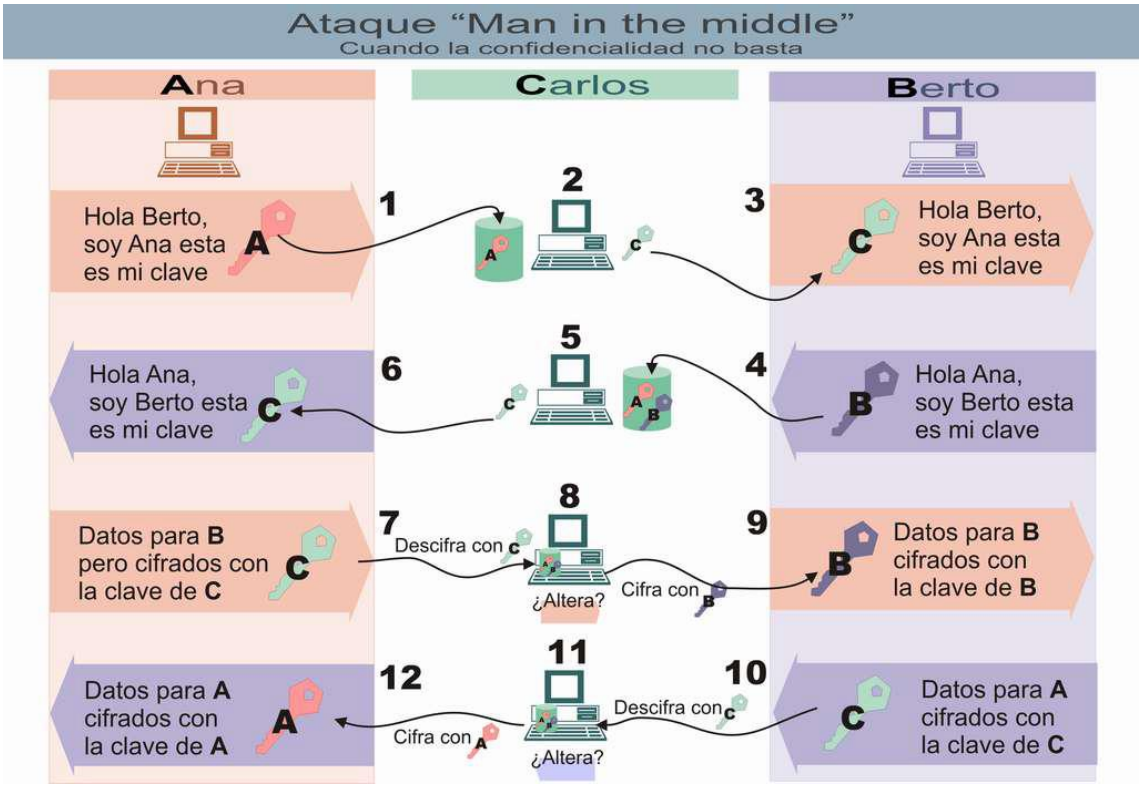

Ilustración 21: Mitm para un sistema de intercambio de claves.

Este tipo de técnicas se usan frecuentemente cuando el atacante ya está infiltrado en la red desde un principio y se basan en la intercepción de información, para poder realizar la intercepción se hace uso de técnicas de spoofing como la que se explica en el punto 8.3.

8.3 ARP Spoofing.

Para explicar esta técnica de ataque primero se debe conocer el protocolo ARP mínimamente y como funciona.

ARP (Address resolution protocol) es un protocolo de comunicaciones que opera en la capa de red del modelo OSI (Open System Interconnection), cuya función es resolver la dirección de hardware (MAC) correspondiente a una dirección IP específica. Para ello se envía un paquete ARP REQUEST a la dirección de difusión de la red con la dirección IP, de la cual se solicita su dirección hardware y se espera que esa máquina u otra responda con la dirección correspondiente en un mensaje ARP REPLY.

En este caso el atacante lo que hará será enviar paquetes ARP infectados a la red de forma que consiga engañar a los clientes, haciéndoles creer que la dirección hardware del atacante se corresponde con la dirección IP de otro host, como podría ser por ejemplo la dirección de la puerta de enlace por defecto de la red, haciendo que cualquier tráfico dirigido hacia esa dirección IP acabe en poder del atacante.

8.4 Explicación del ataque

Una suplantación de identidad de un servidor DNS, desde ahora DNS spoofing por su nombre en inglés y también referido como envenenamiento de memoria cache de DNS, es una forma de ataque informático que se basa en falsificar la resolución de DNS realizada por un servidor ante la petición de un cliente, resultando en una redirección del tráfico a elección del atacante.

Estos ataques se pueden llevar a cabo de 2 maneras distintas:

8.4.1 Envenenamiento de la caché de un servidor DNS

Esta primera variante del ataque se basa en la que probablemente es la característica más peligrosa de un DNS, su memoria caché. Los servidores DNS guardan una memoria cache para así resolver con mayor agilidad las peticiones realizadas por los clientes, puesto que en una organización que posea un alto número de clientes, si un cliente interroga por el mismo nombre que otro previamente, se podría responder al instante sin necesidad de ir a buscar ese nombre de dominio a otro lado, agilizando así el tiempo de respuesta y el de utilización de la red.

Por otro lado, esta característica lo convierte en extremadamente peligroso, puesto que, si un atacante consiguiera introducir datos falsos en esa memoria caché, cualquier cliente que preguntara al servidor de nombres infectado recibiría una dirección falsa.

La razón real por la cual esta es una amenaza muy peligrosa es porque la caché de los servidores DNS puede compartirse, de forma que, si en un entorno real un proveedor de servicio como puede ser Movistar, Vodafone, Orange, etc. recopilaran información de un servidor DNS infectado, las entradas infectadas podrían llegar a los routers domésticos de sus clientes generando un problema masivo.

Esta clase de ataque conlleva un problema, atacar a un servidor DNS no es una tarea sencilla, puesto que al conocerse la amenaza a la que se enfrentan, los servidores DNS se suelen proteger con especial esmero, es por esto por lo que esta técnica no se usa con tanta frecuencia como la segunda que se va a explicar.

8.4.2 Suplantación del servidor de nombres de dominio.

Esta es la forma más común de encontrar este ataque, se basa en hacer uso de técnicas de ataque “Man in the middle o ataque de intermediario” y técnicas de suplantación de identidad como se vieron en los puntos 8.2 y 8.3, para hacer creer al cliente que está recibiendo una respuesta legítima a una petición realizada cuando en realidad está recibiendo una respuesta por parte del atacante.

Desde un comienzo el objetivo es hacer uso de la técnica de ARP spoofing, para hacer creer a un ‘cliente’ que el atacante posee la dirección del servidor DNS (o la puerta de enlace en caso de encontrarse el servidor DNS fuera de la red actual). Por tanto, cada vez que el cliente intente realizar una resolución DNS el atacante recogerá esa petición (simulando un ataque de intermediario) y en este momento es donde entra el ataque de suplantación de identidad del DNS, puesto que el atacante generará una respuesta a esa cuestión DNS, pero con la respuesta que él considere oportuna.

8.5 Herramientas.

La herramienta elegida en este caso para realizar el ataque es “Ettercap”, una herramienta de código abierto encontrada por defecto en la distribución de Kali y usada mayoritariamente para la realización de ataques de tipo “Man in the middle” sobre redes de área local.

Ettercap funciona poniendo la interfaz de red en modo promiscuo, que es un modo de interfaz que provoca que el controlador pase a la CPU la totalidad del tráfico recibido en lugar de pasar sólo las tramas que está específicamente programado para recibir (usado normalmente en rastreo de paquetes), y utilizando técnicas de ARP spoofing para infectar las tablas de ARP de las máquinas víctima.

Ettercap posee también soporte para plugins de forma que no solo sirve para un tipo de ataque, sino que se convierte en una herramienta realmente versátil y capaz de combinar ataques de tipo Man in the middle con otros de manera muy simple.

Por otro lado, esta herramienta también soporta análisis de diversos protocolos (incluidos protocolos cifrados) y funcionalidades para el análisis de redes y máquinas, además de una de las cosas más importantes, es capaz de identificar activa o pasivamente a otros individuos tratando de infectar la red.

8.6 Ataque y efecto.

Lo primero será iniciar la herramienta, Ettercap funciona desde la línea de comandos de Kali Linux, pero también se puede hacer uso de la interfaz gráfica proporcionada por la herramienta. Para lanzarla bastará con utilizar el comando *"ettercap -G"* y se abrirá directamente la ventana que vemos en la ilustración 22.

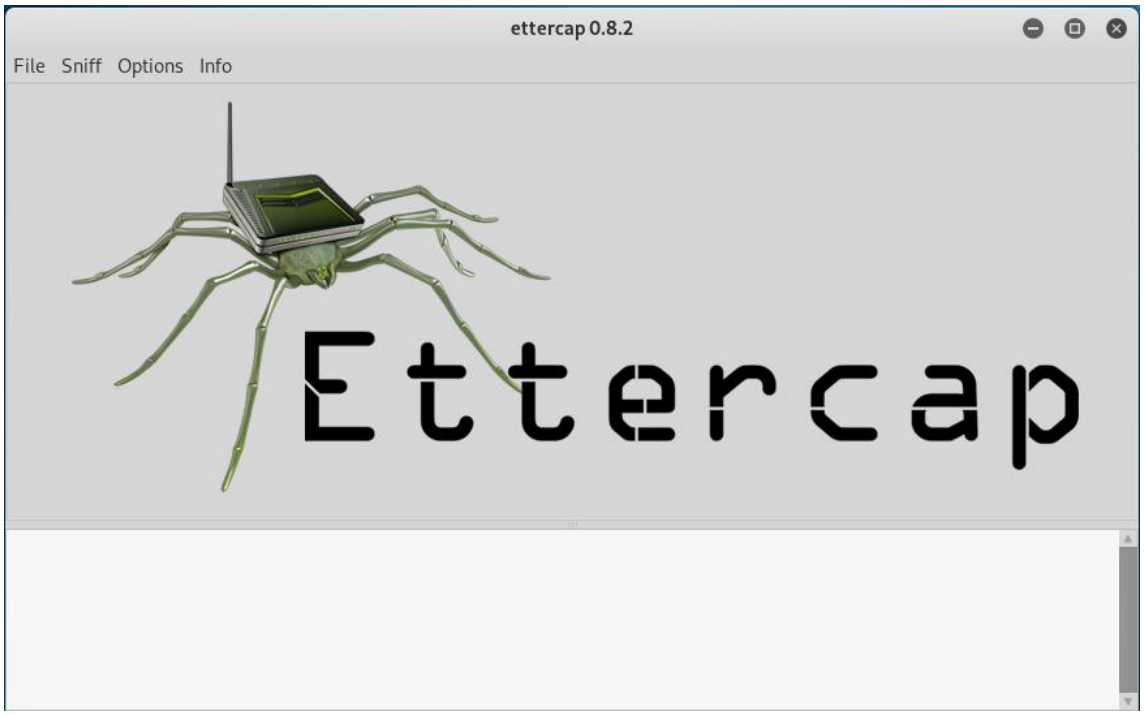

Ilustración 22: Ettercap GUI.

Ahora, se debe especificar a la herramienta por qué interfaz debe operar. Para eso bastará con ir a la opción "Sniff" en la barra superior e indicar "Unified Sniffing" como se ve en la ilustración 23. En este caso solo se tiene una interfaz de red, así que la seleccionada será eth0.

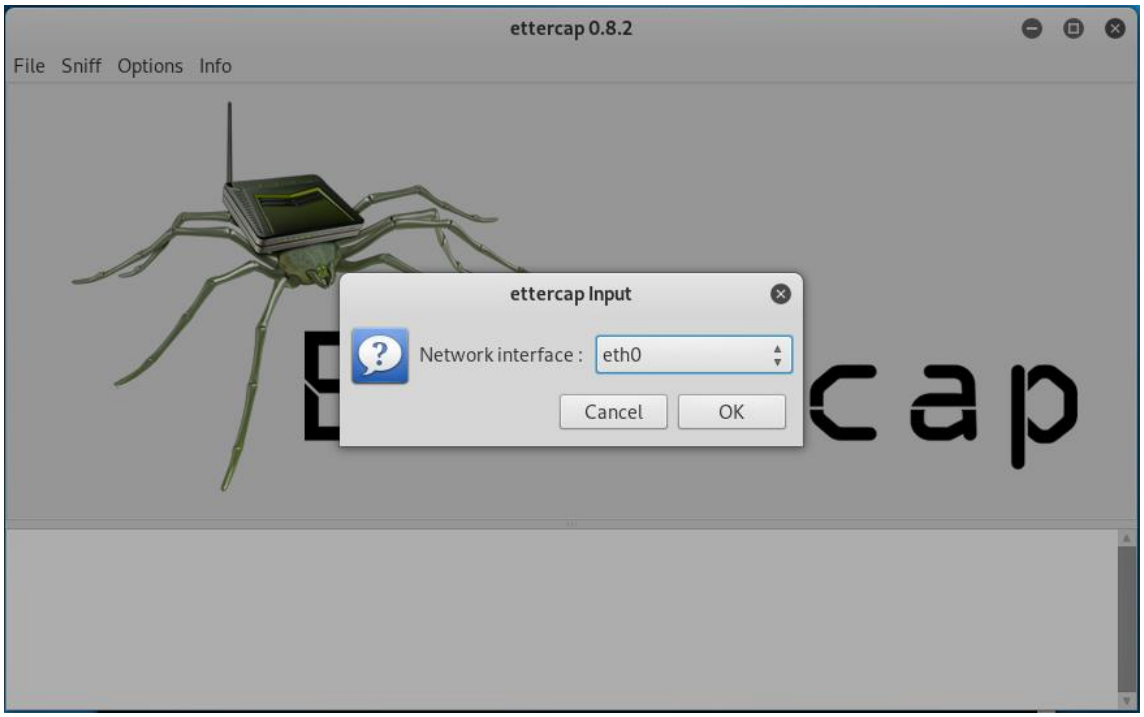

Ilustración 23: Ettercap - interfaz.

Lo siguiente será investigar la red en busca de los hosts existentes en ella para poder atacar a uno especifico, en este caso, el servidor. Para ello, ir a la opción “Hosts” y seleccionar “Scan for hosts”, de esta forma Ettercap automáticamente generará un listado con los hosts encontrados en la red.

Para poder ver ese listado se debe ir nuevamente a la opción “hosts”, pero esta vez entrar a la opción de “Hosts List” y se abrirá el listado de los hosts encontrados por la herramienta en la red como se ve en la ilustración 24.

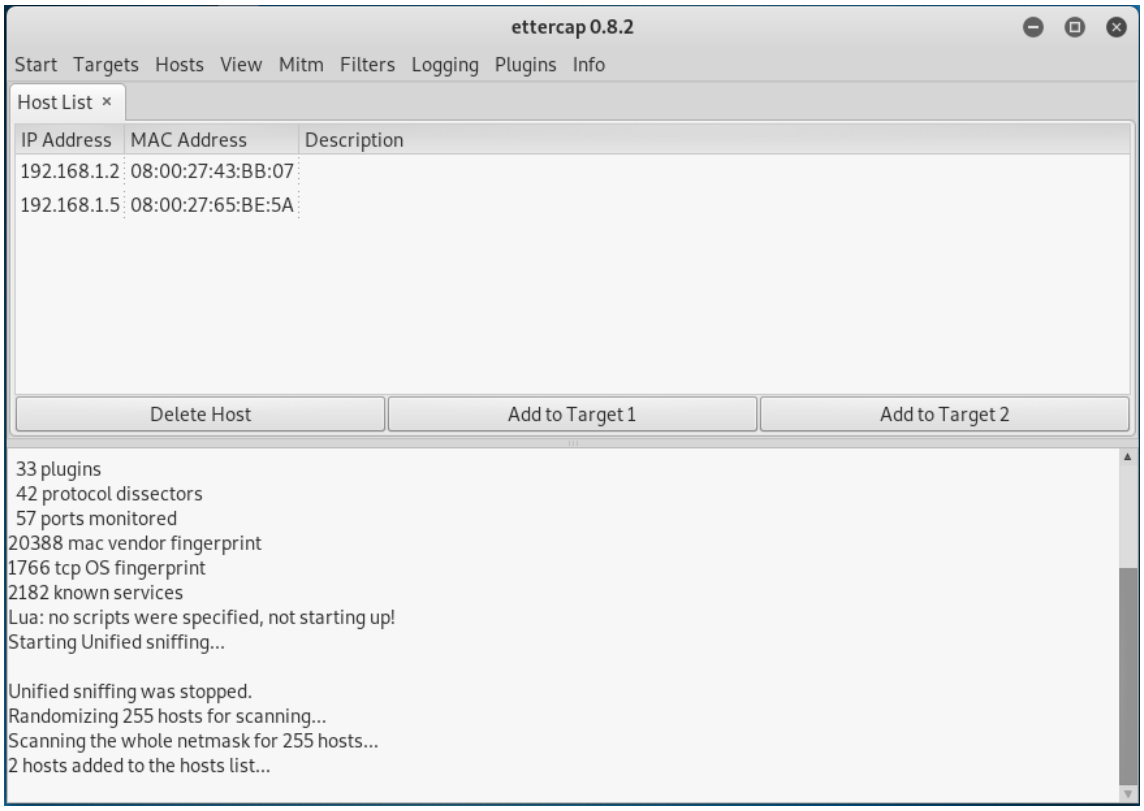

Ilustración 24: Ettercap - Hosts.

A continuación, viene un paso crucial, identificar al objetivo y situarlo en el punto de mira de la herramienta. Para ello se deben establecer 2 objetivos, el primero debe ser la máquina que va a ser víctima del ataque y el segundo debe ser en este caso la puerta de enlace de la red. En el caso de este trabajo la herramienta identifica 2 hosts en la red, uno en la dirección 192.168.1.5 y otro en la 192.168.1.2. Para identificar cuál de los dos es la puerta de enlace de la red se hace uso del comando “route -n” en un terminal y se identifica la puerta de enlace como la dirección 192.168.1.2 por tanto, se deduce que la víctima es la máquina encontrada en la dirección 192.168.1.5.

Una vez se han identificado a la víctima y puerta de enlace, el siguiente paso es añadir a los objetivos de la herramienta los hosts correspondientes. Para ello basta con seleccionar la opción “Add to target 1” y “Add to target 2” respectivamente para cada dirección del listado, como se ve en la ilustración 25.

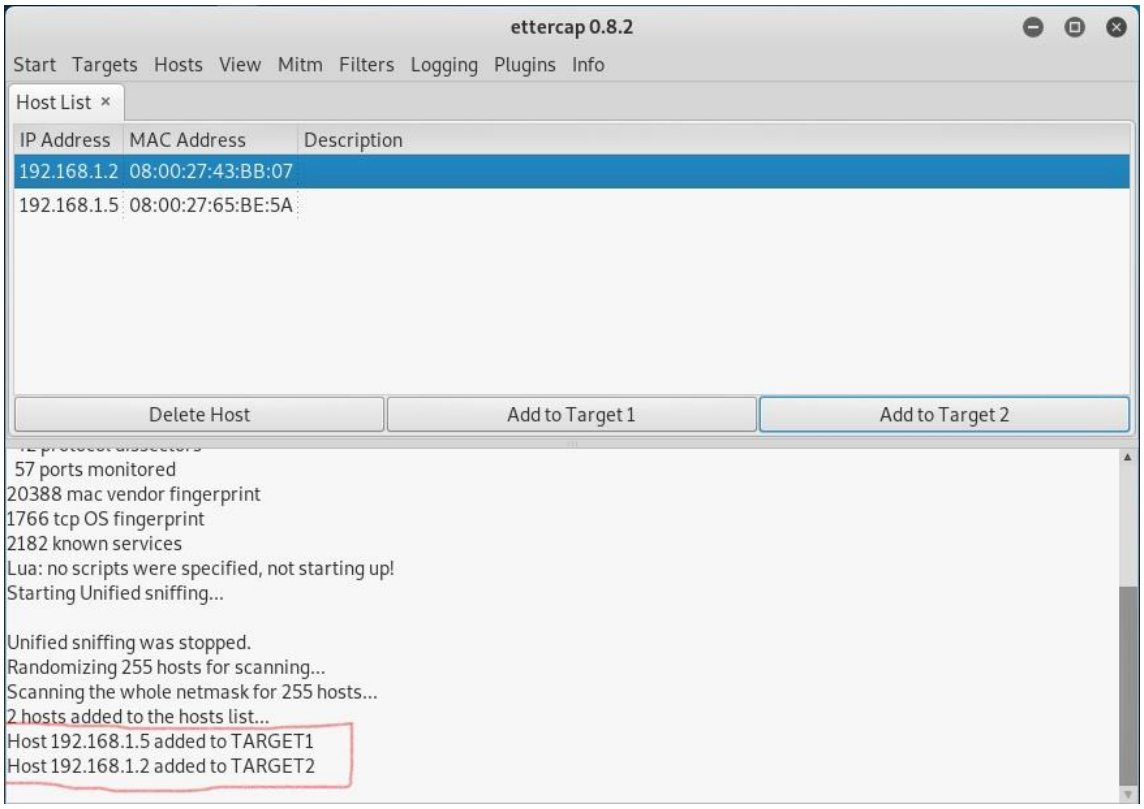

Ilustración 25: Ettercap - Objetivos.

Ahora que ya se tiene al menos un objetivo establecido para la realización del ataque, lo siguiente es configurar el ataque. Se debe ir a la opción “Mitm” y posteriormente “ARP Poisoning” y una vez dentro indicarle que espíe las conexiones remotas con la opción “Sniff remote connections”, puesto que el objetivo actual, es engañar a la víctima haciéndole creer que el atacante es su nueva puerta de enlace y así poder espiar todo su tráfico (Véase ilustración 26).

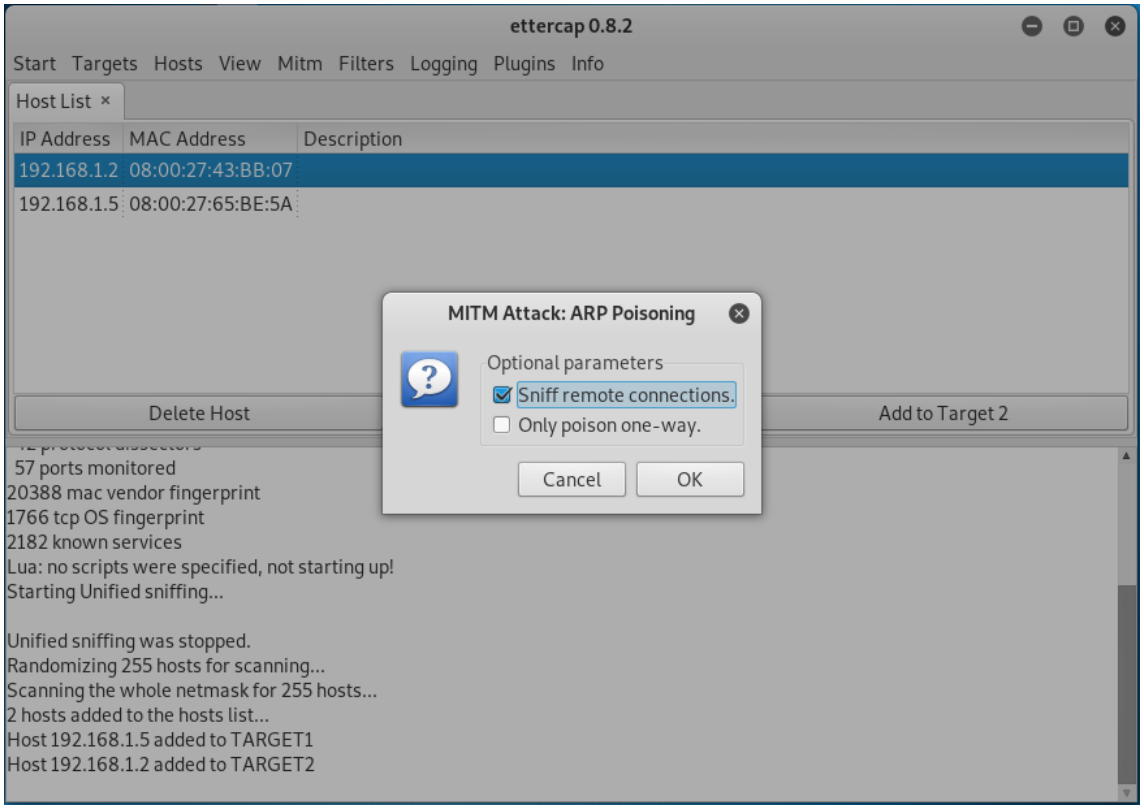

Ilustración 26: Ettercap - Man in the middle.

En este momento la herramienta ha llevado a cabo un ataque de tipo ARP Spoofing para indicarle a la víctima que la puerta de enlace se corresponde con la máquina atacante. De hecho, si en la máquina víctima a modo de comprobación se muestra la tabla de resoluciones ARP de la máquina, se encontrará que la puerta de enlace (192.168.1.2) se encuentra en la misma dirección hardware que el atacante como se ve en la ilustración 27.

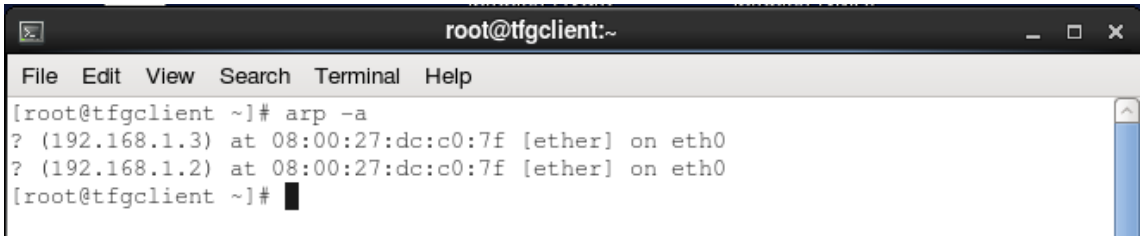

Ilustración 27: Tabla ARP de la víctima.

El siguiente paso es indicar a la herramienta que tipo de ataque se va a realizar, activando el plugin correspondiente al ataque, para ello basta con ir a la opción “Plugins” de la barra de navegación y seleccionar la opción “dns_spoof” del listado, como se ve en la ilustración 28.

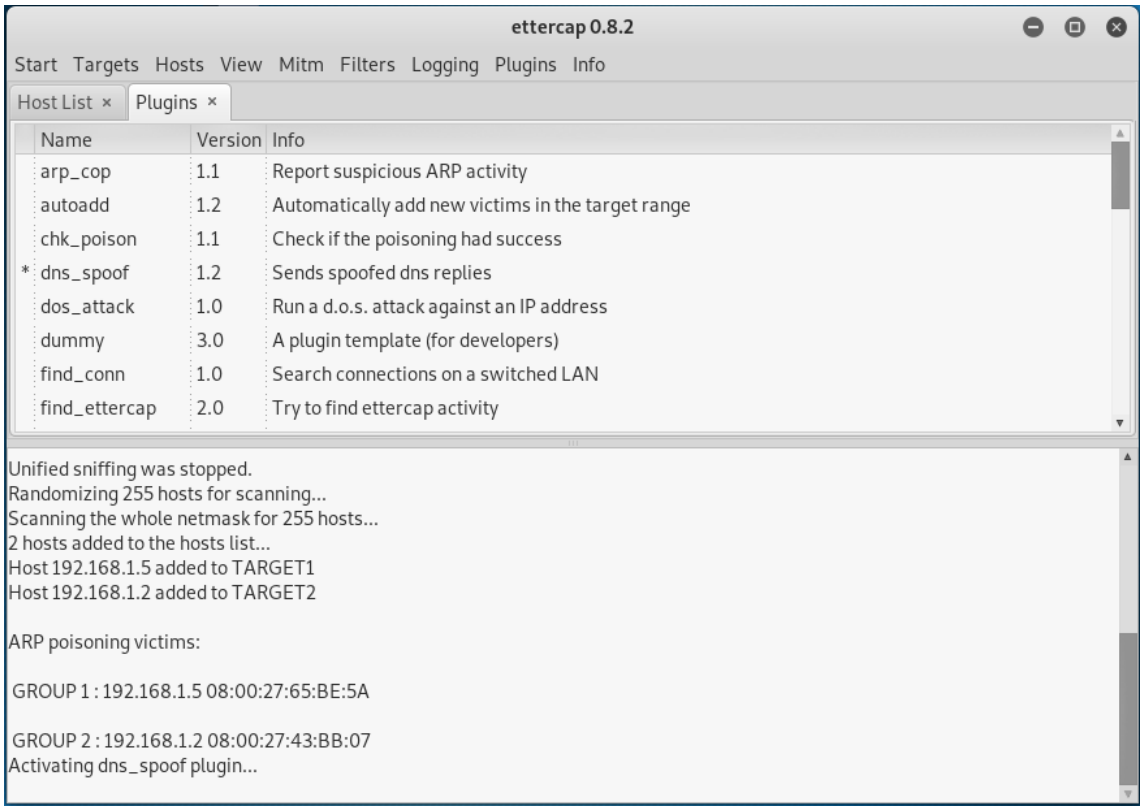

Ilustración 28: Ettercap - dns_spoof plugin.

En este momento la herramienta está lista para realizar el ataque. Ettercap simula la existencia de una caché DNS, que sencillamente es un fichero de configuración de la herramienta, donde Ettercap se basará para formar las respuestas DNS que le enviará a la víctima simulando ser el servidor DNS.

El fichero se encuentra en la ruta "*/etc/ettercap/etter.dns"* por defecto y lo único que se debe hacer es añadir los dominios que se quiera infectar al listado de los ya encontrados inicialmente en el fichero como se ve en la ilustración 30. En este caso se han añadido los dominios de YouTube, Gmail y la Universidad de Las Palmas de Gran Canaria y se han redirigido a una página de prueba montada en un servidor apache en la máquina atacante (Véase un extracto del fichero etter.dns en el anexo II).

```
#       host. (look at the www.microsoft.com example)                    #
#                                                                        #
##########################################################################

################################
# microsoft sucks ;)
# redirect it to www.linux.org
#
#
*.youtube.*     A       192.168.1.3
www.gmail.com   A       192.168.1.3
*.ulpgc.*       A       192.168.1.3
microsoft.com      A   107.170.40.56
*.microsoft.com    A   107.170.40.56
www.microsoft.com  PTR 107.170.40.56      # Wildcards in PTR are not allowed

########################################
# no one out there can have our domains...
#
```

Ilustración 29: Ettercap - dominios spoofeados.

En el momento que la víctima lance una petición para que el DNS le responda, Ettercap se encargará de recibir esa petición y en función de la configuración del fichero *"etter.dns"* responder con un dominio falso (spoofeado) o redirigir la petición al servidor DNS de la red para que este dé una respuesta legítima. Las dos formas de comprobar que esto funciona son, lanzando el comando 'dig' (utilidad de los sistemas Unix para cuestionar por servidores DNS) con una de las direcciones que de antemano se sabe que están en el fichero '*etter.dns*' (Véase ilustración 30), o simplemente accediendo a una de las páginas desde el navegador (Véase ilustración 31).

Ilustración 30: Ettercap – Dig.

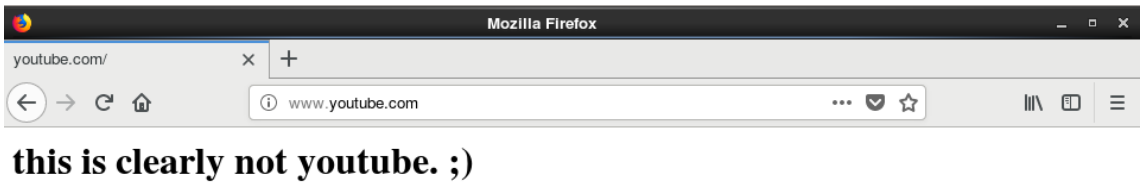

Ilustración 31: Ettercap - Navegación.

8.7 Defensa.

Esta clase de ataques, basados en técnicas Man in the middle y ARP Spoofing, son ataques cuya mitigación con el solo uso de un cortafuego es prácticamente imposible, puesto que la herramienta usada, suplanta identidades para llegar a cumplir su objetivo. Un cortafuego no es capaz de detectar si la dirección hardware de la cual procede un paquete es legítima o no, solamente permite comprobar si es o no una dirección hardware específica.

Aun así, en el caso de una organización que posee un servidor de nombres privado, es posible conocer la dirección hardware exacta de la máquina donde se encuentra alojado el servidor de nombres, por tanto, la solución proporcionada se basa en desestimar cualquier respuesta a una cuestión de DNS que no provenga de ese servidor.

En el caso de que el servidor de nombres esté en la misma red, la dirección que debemos especificar es exactamente la del servidor de nombres, en caso de que se encuentre fuera de la red privada en la que se encuentra el servidor, se puede hacer uso de la dirección del Router inmediatamente superior.

La regla correspondiente quedaría de la siguiente manera para la estructura de red con la que contamos:

```
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -i eth0 -p tcp -m tcp --dport 21 -j ACCEPT
-A INPUT -i eth0 -p udp -m udp --sport 53 -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --sport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
# Esta regla nos permite recibir respuestas solo desde nuestro servidor DNS
-A INPUT -p udp -m udp --sport 53 -m mac ! --mac-source 08:00:27:43:BB:07 -j REJECT
-A OUTPUT -o eth0 -p udp -m udp --dport 53 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --sport 80 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --dport 80 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --sport 21 -j ACCEPT
COMMIT
```

Cabe destacar que, si aplicamos dicha regla de seguridad en nuestro cortafuegos y somos atacados por alguien de la misma manera expuesta en este trabajo, el atacante seguirá siendo capaz de infectar la tabla ARP, pero las respuestas DNS generadas por él nunca nos llegarán. También cabe la posibilidad de que el atacante se dé cuenta de dicha regla y suplante la dirección hardware de la máquina donde se encuentra nuestro servidor DNS, para lo cual no se puede de ninguna manera protegerse mediante un cortafuego.

8.8 Conclusiones

La verdadera razón por la cual un ataque de suplantación de identidad a un servidor DNS es tan peligroso se debe a que al recibir una resolución de nombre por parte de un servidor no se puede saber si esa respuesta es legítima en su totalidad o ha sido manipulada por alguien de por medio.

A pesar de esto, se puede poner algún pequeño nivel de protección como el que se vio en el apartado anterior que por pequeño que sea, podría ayudar a identificar el ataque y dar pie a que el administrador del sistema atacado ponga fin al ataque por otros métodos.

8.9 Recomendaciones.

En vista de los problemas que generan este tipo de ataques a los cortafuegos como es IPtables, se pueden proponer otras soluciones que sí que serían capaces de erradicar el problema como serían:

8.9.1 DNSSEC

Domain name system security extensions o DNSSEC Es un conjunto de herramientas diseñadas para asegurar ciertos tipos de información que provee el sistema de nombres de dominio, esto se consigue dándoles a los clientes una certificación de origen de la información que reciben y garantizándoles la integridad de los datos.

DNSSEC fue creado para evitar problemas como los planteados por ataques de envenenamiento de la memoria cache de un servidor DNS visto en este capítulo de este documento, puesto que todas las respuestas de un servidor de nombres cuya zona está protegida por DNSSEC se entregan firmadas digitalmente, de forma que con realizar una comprobación de firma se es capaz de comprobar que la información recibida es legítima y correcta.

8.9.2 Uso de programas de terceros

En el caso de un ataque de suplantación completa del servidor DNS como se ha visto en este documento, se destaca la existencia de diversas aplicaciones software cuyo objetivo es la detección de intrusos en una red.

Para ello estos sistemas se encargarían de realizar comprobaciones en el protocolo ARP de diversas formas de manera que sea capaz de impedir una suplantación de identidad por ARP y, por tanto, estarían frenando el hecho de que el atacante se infiltre y desvíe el tráfico de la red.

9 Ataques de denegación de servicio distribuidos (DDoS).

9.1 Descripción

Los ataques de denegación de servicio distribuidos (DDoS) son los ataques de mayor popularidad de la última década. Para entender lo que es un ataque de denegación de servicio distribuido, primero se debe conocer mínimamente los ataques en los que se basan, que son ataques de denegación de servicio o DoS.

Los ataques de denegación de servicio buscan mermar un servicio, un sistema o una red de forma temporal o permanente. La forma de conseguir esto típicamente es inundando la máquina o red objetivo con peticiones sin objetivo real de hacer uso de algún servicio. El objetivo siempre es privar a usuarios legítimos de hacer uso del servicio o de la red a la que intentan acceder.

Un ataque de denegación de servicio depende en gran medida de los recursos de la máquina atacante, es por esto por lo que, en la mayoría de los casos, estos ataques se lanzan de forma distribuida para aumentar la potencia y evitar que la víctima se defienda del ataque solamente bloqueando a la dirección IP de la que procede, convirtiéndose así en uno de los tipos de denegación de servicio más conocidos, llamados ataques de denegación de servicio distribuidos o DDoS por sus siglas en inglés, que será el tipo de ataque en el que se centra este capítulo.

Los ataques **DDoS** son, en esencia, ataques de denegación de servicio, pero a una escala mucho mayor, donde el ataque de denegación de servicio puede provenir de hasta miles de fuentes distintas. Teniendo en cuenta que el ataque se lanza desde distintos orígenes y que los paquetes pueden llegar a la víctima con direcciones de origen suplantadas (spoofeadas), podría ser casi imposible frenar un ataque de denegación de servicio distribuido por métodos convencionales.

Los ataques de denegación de servicio distribuido se pueden separar en 3 grandes grupos dependiendo del objetivo del ataque:

- **DDoS por tráfico de protocolo:** El objetivo de este ataque es reducir el rendimiento de una máquina haciendo un envío masivo de tráfico que podrían ser paquetes TCP o UDP, los cuales requerirían atención por parte de la víctima generando así una ralentización en el sistema e incluso una parada completa del servicio atacado. Incluyen inundaciones SYN, inundaciones de paquetes fragmentados, ping de la muerte, etc.
- **DDoS contra el ancho de banda o volumétricos:** En este tipo de ataque de denegación de servicio distribuido el objetivo es inundar la red de cualquier manera posible para dificultar el acceso al servicio de los clientes legítimos. Este tipo de ataque además puede generar una caída del servidor debido a excesiva afluencia de paquetes que por lo general son anónimos o con identidad suplantada. Incluyen inundaciones ICMP y otras inundaciones de paquetes con suplantación de identidad.
- **DDoS contra una aplicación:** De los 3 tipos encontrados este es probablemente el peor para los administradores de un sistema, opera directamente en la capa de aplicación intentando causar un agotamiento de los recursos al sistema o la red, no son muy habituales debido a su complejidad comparado con los demás, pero son los más difíciles de detectar y defender. Incluyen ataques del tipo inundación de GET/POST, ataques contra servidores Apache, etc.

Algunos de los ataques DDoS más comúnmente usados en el mundo incluyen los siguientes:

9.1.1 SYN flood:

El protocolo TCP hace uso de una negociación en 3 pasos para la transferencia de datos de un cliente a un servidor y viceversa de la siguiente manera:

1. El cliente envía una petición de sincronización al servidor (SYN)
2. El servidor responde con un reconocimiento de esa petición (SYN-ACK) y reserva los datos de conexión.
3. El cliente reconoce la respuesta del servidor (ACK)

Tanto cliente como servidor reciben un reconocimiento de la conexión que se intentó realizar, una vez se finaliza el tercer paso, queda establecida una conexión de tipo dúplex completo (habilitado un canal de recepción y envío completos) entre el cliente y el servidor.

Teniendo en cuenta el funcionamiento de la negociación en 3 pasos, es sencillo entender el ataque. El objetivo de un ataque de tipo 'SYN flood' es realizar una inundación de peticiones SYN (las cuales pueden venir de direcciones suplantadas), ante las cuales el servidor reservará los datos de conexión y responderá con un SYN-ACK, pero el atacante o nunca responderá o nunca recibirá el SYN-ACK puesto que las direcciones de origen eran falsas, dejando así una inundación de conexiones semiabiertas que consumen recursos en el servidor y pueden llegar a generar una caída del servicio.

9.1.2 ICMP flood:

Estos ataques están basados en el protocolo ICMP y su objetivo es llevar a cabo una inundación del objetivo a base de mensajes de ICMP Echo Request (ping) sin siquiera esperar respuesta, de forma que la víctima se vea desbordada.

Esta clase de ataques, aunque no son excesivamente peligrosos, puesto que el protocolo ICMP está considerado ampliamente como inseguro y por tanto no se permite en la mayoría de los sistemas, pero realizados sobre una máquina desprotegida podría llegar a causar un intento de respuesta a todas las peticiones por parte del servidor con la consecuente y notable ralentización del sistema.

9.1.3 Ping of death:

Este ataque explota una conocida vulnerabilidad del protocolo IP y es que el protocolo IP a pesar de tener un tamaño de paquete standard de 64 bytes (incluyendo la cabecera), se permite que los paquetes tengan tamaño de hasta 65.535 bytes (incluyendo la cabecera), debido a que un tamaño superior se saldría del establecido en el protocolo, algunos sistemas simplemente no están preparados para soportar un tamaño de paquete superior.

Este ataque consiste en generar un paquete de gran tamaño, mal formado y fragmentado de forma que al ser reconstruido en el sistema destino, este sufra un desbordamiento de buffer de memoria, generando así una caída del sistema con sus consecuentes vulnerabilidades.

9.1.4 Smurf:

Este tipo de ataques son sencillos de realizar, pero son dependientes de la red en la que se encuentran y el número de hosts que haya en ella.

Los ataques de tipo smurf son un tipo de ataque DDoS que se basa en enviar un mensaje ICMP Echo Request a la dirección de difusión de la red desde una máquina atacante, pero con un pequeño añadido, antes de enviar el paquete, el atacante suplanta la identidad de la máquina víctima, de forma que el paquete sale con la dirección de la víctima como dirección de origen del paquete.

Debido a que ese paquete se envía a la dirección de difusión de la red, todos los hosts que formen parte de la red lo recibirán y responderán a ese paquete inundando a la víctima y dependiendo del número de hosts que respondan, eso generará una ralentización en el sistema objetivo e incluso hasta una caída completa del sistema.

9.1.5 Botnets

El concepto “distribuido” en este tipo de ataques hace referencia a que el ataque se lanza desde diversas máquinas que en principio no tienen relación y actuarían bajo un sistema de estructura maestro-esclavos. Para este tipo de tareas lo habitual es hacer uso de redes de máquinas conectadas entre sí para un mismo propósito, llamadas ‘botnets’.

Una botnet es una red de sistemas conectados a través de internet los cuales estarían “infectados” con un programa maligno, de forma que un atacante desde un software de control es capaz de lanzar ordenes que estos sistemas ejecutarán.

Por norma general, estas botnets, se usan con fines malignos, como puede ser realizar ataques de denegación de servicios distribuidos, enviar spam, etc.

Este tipo de sistemas, se pueden generar teniendo los recursos o se pueden adquirir sencillamente a través de internet, con programas como puede ser UFONET, un set de herramientas para realizar ataques DDoS a gran escala y que provee al atacante con una botnet para uso completo.

En la ilustración 32 se ve un ejemplo de la arquitectura de una botnet.

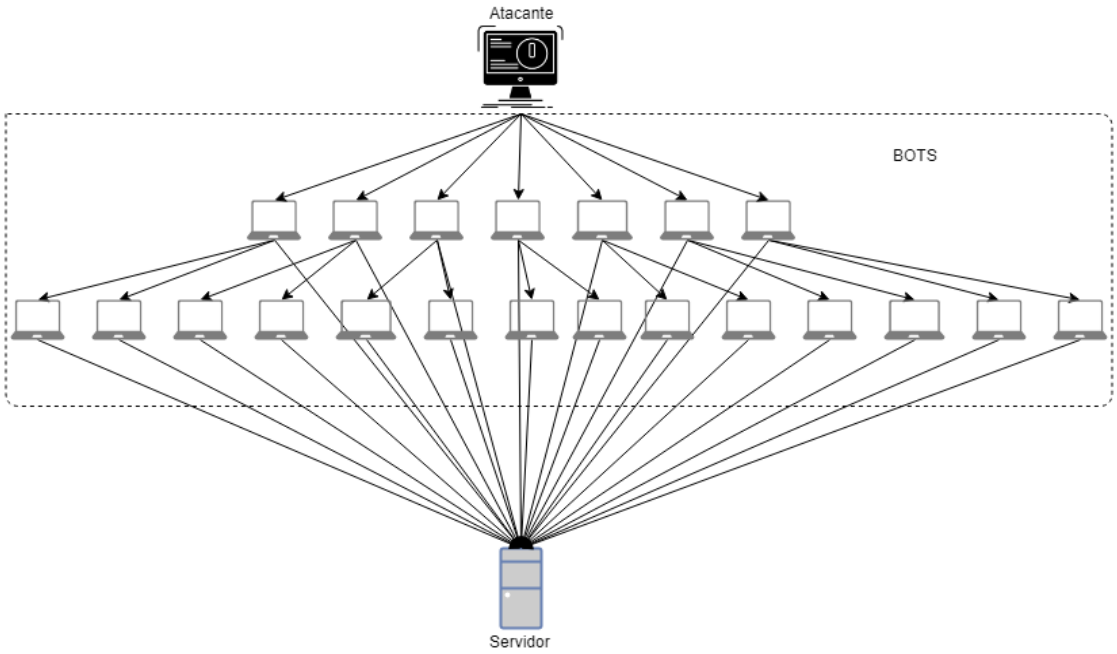

Ilustración 32: Botnet - arquitectura.

9.2 Herramienta usada.

La herramienta usada para el lanzamiento de este ataque ha sido Metasploit en su versión 5.0.34 (Véase ilustración 33), la cual es la herramienta más usada del mundo en el campo del Pentesting (penetración de máquinas).

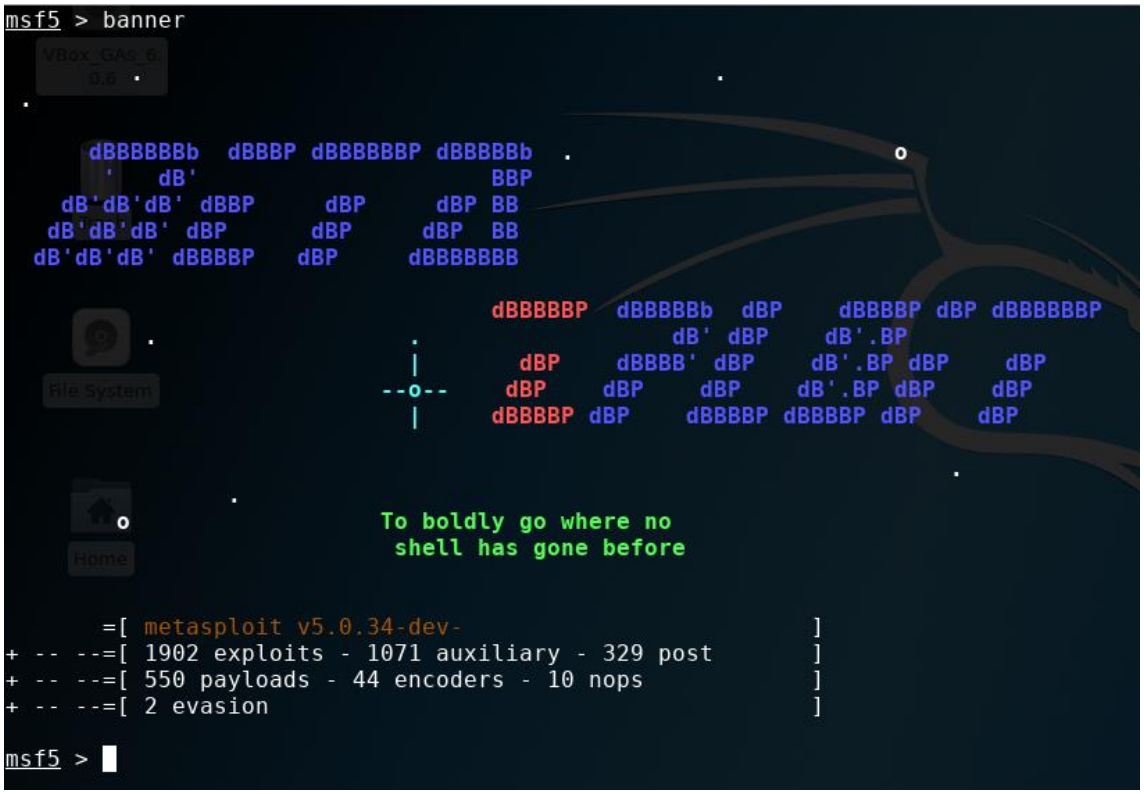

Ilustración 33: Metasploit - información.

Como framework provee a los usuarios que hagan uso de esta herramienta con módulos, contenido y otras herramientas diseñadas para realizar pruebas de penetración de máquinas y auditorías informáticas. Por otro lado, es una herramienta de código abierto, lo cual permite que tenga una comunidad detrás implicada en el desarrollo de nuevos módulos que están disponibles al instante para su uso, de forma que nuevas vulnerabilidades encontradas en sistemas se ven con regularidad añadidas a la herramienta para hacer uso de ellas y realizar pruebas.

Metasploit hace uso de módulos, que son básicamente software o código que realizan una función. En Metasploit encontramos distintos tipos de módulos entre los que se destacan:

Auxiliares	Módulos para realización de algunas tareas específicas, como pueden ser tareas de escaneo, espionaje de tráfico, etc.
Encoders	Los enconders son usados para enmascarar los módulos de Metasploit de forma que sean más difíciles de detectar o incluso indetectables por parte de cortafuegos y antivirus.
Exploits	Como su nombre indica, son trozos de código que usados se encargan de explotar vulnerabilidades de los sistemas.
Payloads	Los payloads son trozos de código que se ejecutarán en la máquina víctima del ataque una vez explotada una vulnerabilidad.

Para hacer uso de cualquier módulo basta con usar el comando:

- use <Módulo>

Si no se conoce la ubicación especifica de un módulo podemos usar el comando 'search <módulo>' como se ve en la ilustración 34 y Metasploit nos indicará la ubicación.

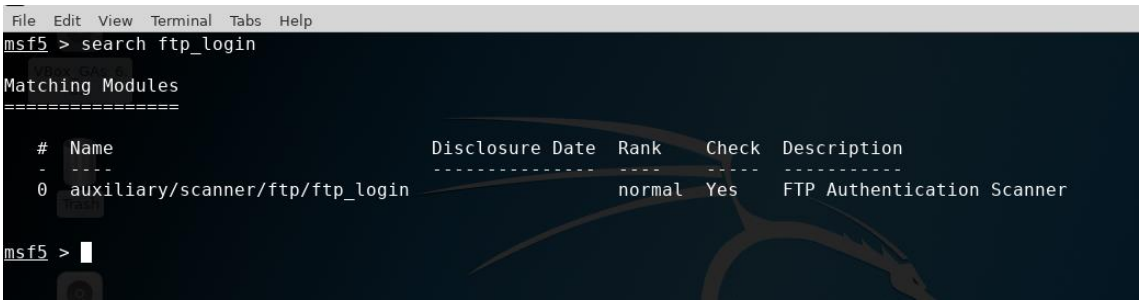

Ilustración 34: Metasploit – search.

Una vez encontrado el módulo se puede hacer uso del comando 'info' para ver toda la información referente al módulo, los parámetros que deben establecerse, etc. Para indicar los parámetros al programa se hace uso del comando set como se verá más adelante en la implementación.

Una vez configurado todo bastará con lanzar el comando ‘exploit’ o ‘run’ para iniciar el ataque.

9.3 Estructura

Los ataques de denegación de servicio distribuidos requieren de una estructura igual a la que se ha usado durante este proyecto, pero con una variación, se deben introducir nuevas máquinas virtuales para simular la naturaleza distribuida del ataque, debido a los recursos que consumen dichas máquinas se ha debido ajustar la distribución de Kali-linux usada a la versión de escritorio más sencilla (Kali Linux Light 2019.2), de forma que fuera viable ejecutar varias máquinas a la vez sin generar problemas de rendimiento al host anfitrión de las máquinas.

En este caso la estructura ha contado con 5 máquinas de igual configuración para la realización de los ataques, en el anexo 3 de este documento se encuentran ilustraciones con las máquinas usadas de atacante con su configuración de red.

La estructura de red queda de la forma que se puede ver en la ilustración 35.

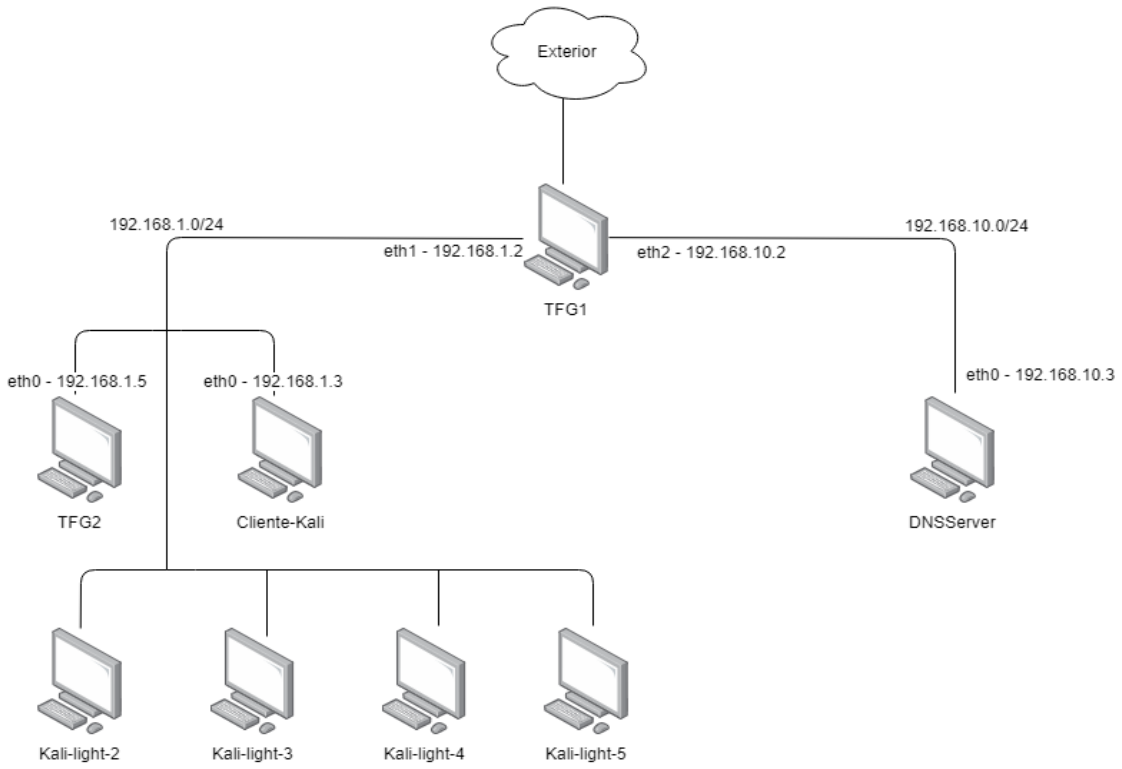

Ilustración 35: DDoS - estructura de red.

9.4 Simulación

El ataque elegido para esta simulación ha sido un ataque de tipo 'SYN flood' principalmente, debido a que al estar basado en protocolo TCP y ser uno de los más populares, será de los que más se encuentren y para los cuales este documento ayudará más, a pesar de haber elegido este, también se plantearán soluciones a algunos de los ataques de denegación de servicio que explotan el protocolo ICMP.

Para preparar el ataque debemos iniciar la consola de Metasploit y seleccionar el ataque synflood (encontrado en el módulo auxiliar, bajo la ruta *'auxiliary/dos/tcp/synflood'*), este módulo tiene solamente 4 parámetros necesarios, 2 de ellos vienen establecidos por defecto, se deben indicar los 2 faltantes que son la máquina y el puerto a los que lanzar el ataque (Véase ilustración 36).

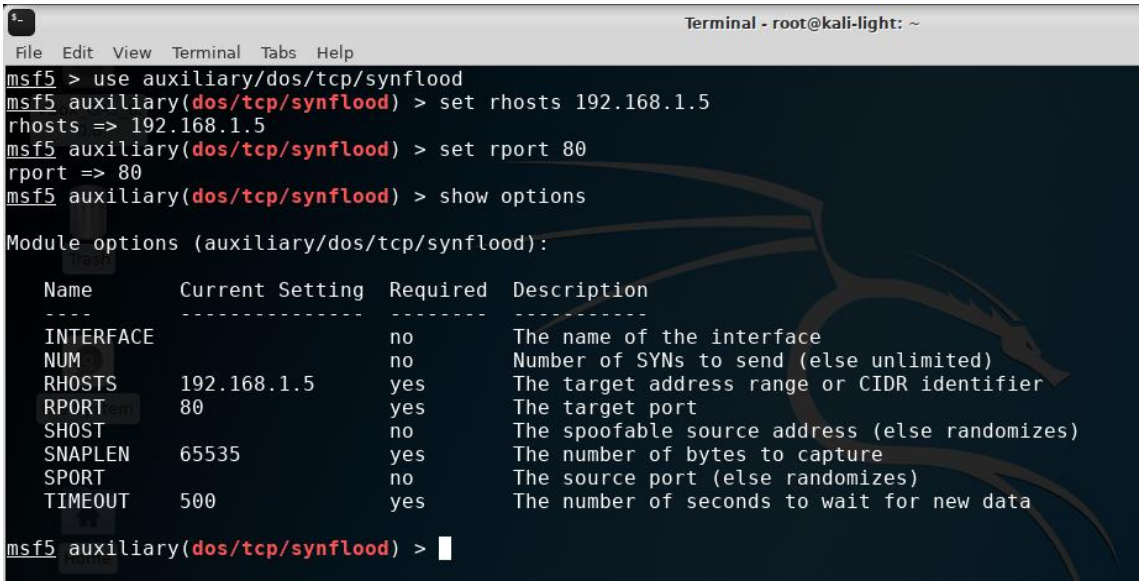

Ilustración 36: DDoS - Synflood preparation.

El último paso es lanzar el ataque, desde todas las máquinas virtuales que van a realizar el ataque a la vez, se debe lanzar el comando 'exploit' o 'run' (Véase ilustración 37).

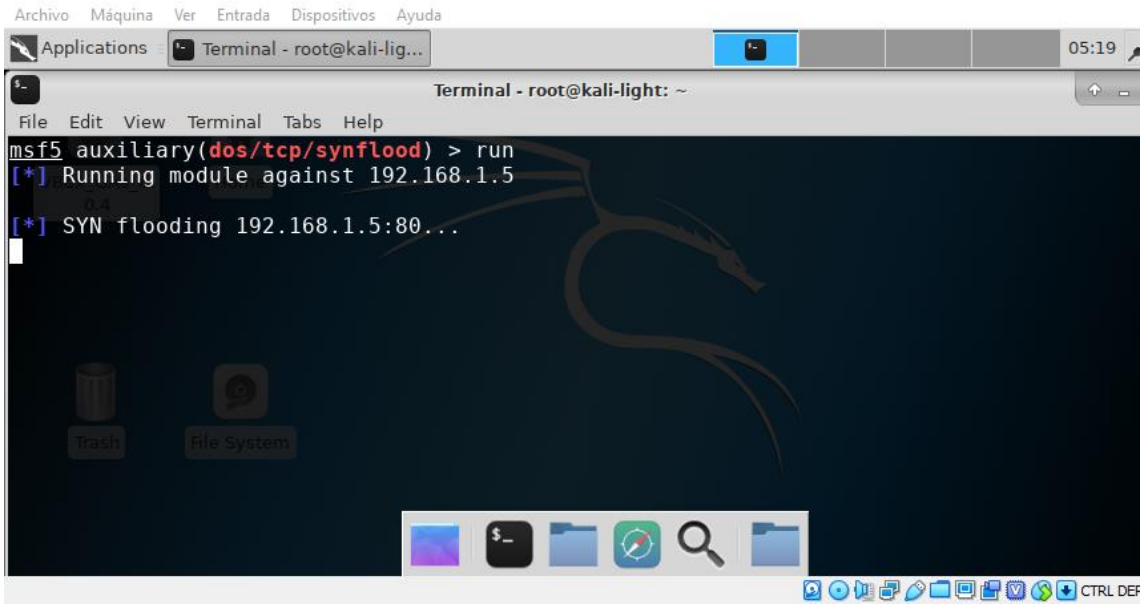

Ilustración 37: Synflood - run.

El ataque resulta en un exceso de tráfico TCP que resulta en un colapso del servicio ofrecido, lo cual inhabilita a usuarios legítimos a acceder a él.

9.5 Defensa

Un ataque de tipo syn flood se basa en, como se explicó anteriormente, abrir conexiones de TCP, pero nunca cerrarlas por parte del atacante. Esto significa que desde la máquina atacada solo se verá una entrada masiva de paquetes de tipo TCP en cuya cabecera está habilitado el flag SYN.

Para poder mitigar esta clase de ataques sin que haya una perdida en el servicio para usuarios legítimos, se han implementado reglas en el cortafuego teniendo en cuenta los siguientes puntos:

- Se deben eliminar directamente paquetes de conexión TCP que posean flags sospechosos, en este caso hablamos de paquetes con todos o ninguno de los flags TCP activados y paquetes con el flag de SYN y FIN activados simultáneamente, puesto que son mutuamente excluyentes.
- Se debe eliminar todo paquete ICMP, por seguridad para evitar inundaciones de protocolo ICMP.
- Se deben descartar todos los paquetes fragmentados, para evitar problemas con ataques de fragmentación de protocolo IP comunes.
- Se desechan paquetes de conexión TCP cuyo estado de conexión sea invalido.
- Se establece un límite de conexiones TCP establecidas con el servidor de forma que, si se sobrepasa, el sistema no se desborde con conexiones entrantes.
- Cualquier conexión entrante que en principio sea legítima, debe filtrarse y limitarse por tiempo, de forma que el servidor no se vea ahogado en peticiones por segundo.

Las reglas implementadas para mitigar el efecto del ataque han sido:

```
# Generated by iptables-save v1.4.7
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -p icmp -j DROP
-A PREROUTING -f -j DROP
-A PREROUTING -p tcp --tcp-flags ALL NONE -j DROP
-A PREROUTING -p tcp --tcp-flags ALL ALL -j DROP
-A PREROUTING -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
COMMIT
# Generated by iptables-save v1.4.7
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:CONN_TRACKING - [0:0]
-A INPUT -m conntrack --ctstate INVALID --ctproto 6 -j DROP
-A INPUT -p tcp -m connlimit --connlimit-above 1000 --connlimit-mask 32 -j REJECT --reject-with tcp-reset
-A INPUT -p tcp -m conntrack --ctstate NEW -j CONN_TRACKING
-A INPUT -i eth0 -p tcp -m tcp --sport 80 -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 80 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --sport 80 -j ACCEPT
-A OUTPUT -o eth0 -p tcp -m tcp --dport 80 -j ACCEPT
-A CONN_TRACKING -m hashlimit --hashlimit-upto 5/sec --hashlimit-mode srcip --hashlimit-name limitador --hashlimit-
htable-max 1000000 --hashlimit-htable-size 250000 -j ACCEPT
-A CONN_TRACKING -j DROP
COMMIT
```

Las reglas se han aplicado de la siguiente forma, desde la tabla mangle (por razones de eficiencia al actuar antes que la tabla de filtros) se eliminan paquetes de protocolo ICMP y paquetes fragmentados, además de los paquetes de conexiones TCP con flags sospechosos activados. Desde la tabla de filtros se eliminan conexiones TCP de estado inválido y se establece un límite superior de 1000 conexiones simultáneas, a partir de ahora toda conexión TCP nueva entrante a la máquina será tratada por la regla de la cadena "CONN_TRACKING". En dicha cadena se crea una tabla de sistema donde se registran los datos de la conexión, esta regla además de registrar los datos, impide el paso de más de 5 conexiones TCP nuevas por segundo que vengan desde la misma dirección IP. Solucionando así el problema del exceso de conexiones entrantes y manteniendo el servicio disponible para usuarios legítimos.

Cabe destacar que el módulo usado, 'hashlimit', crea tablas de sistema para hacer las comprobaciones, se debe tener cuidado con el tamaño de tabla y el tamaño que se le da a la ráfaga de entrada en caso de especificarlo, estos valores se establecen con las opciones '--hashlimit-htable-max' y '--hashlimit-htable-size' respectivamente.

9.6 Resultados

Después de la aplicación de las reglas expuestas en el anterior punto, todo el tráfico se ve encauzado, de forma que la máquina no se vea inundada y por tanto es capaz de seguir dando el servicio a pesar de la simulación realizada.

Para comprobar esto se lanzó el ataque de nuevo y desde otra máquina virtual nueva se consiguió acceder al servicio alojado en el puerto del ataque, en este caso era un servidor apache. Además, si buscamos la mencionada tabla de sistema que genera el módulo usado de IPTables, veremos los registros del ataque (Véase ilustraciones 38 y 39 respectivamente).

Ilustración 38: Acceso http.

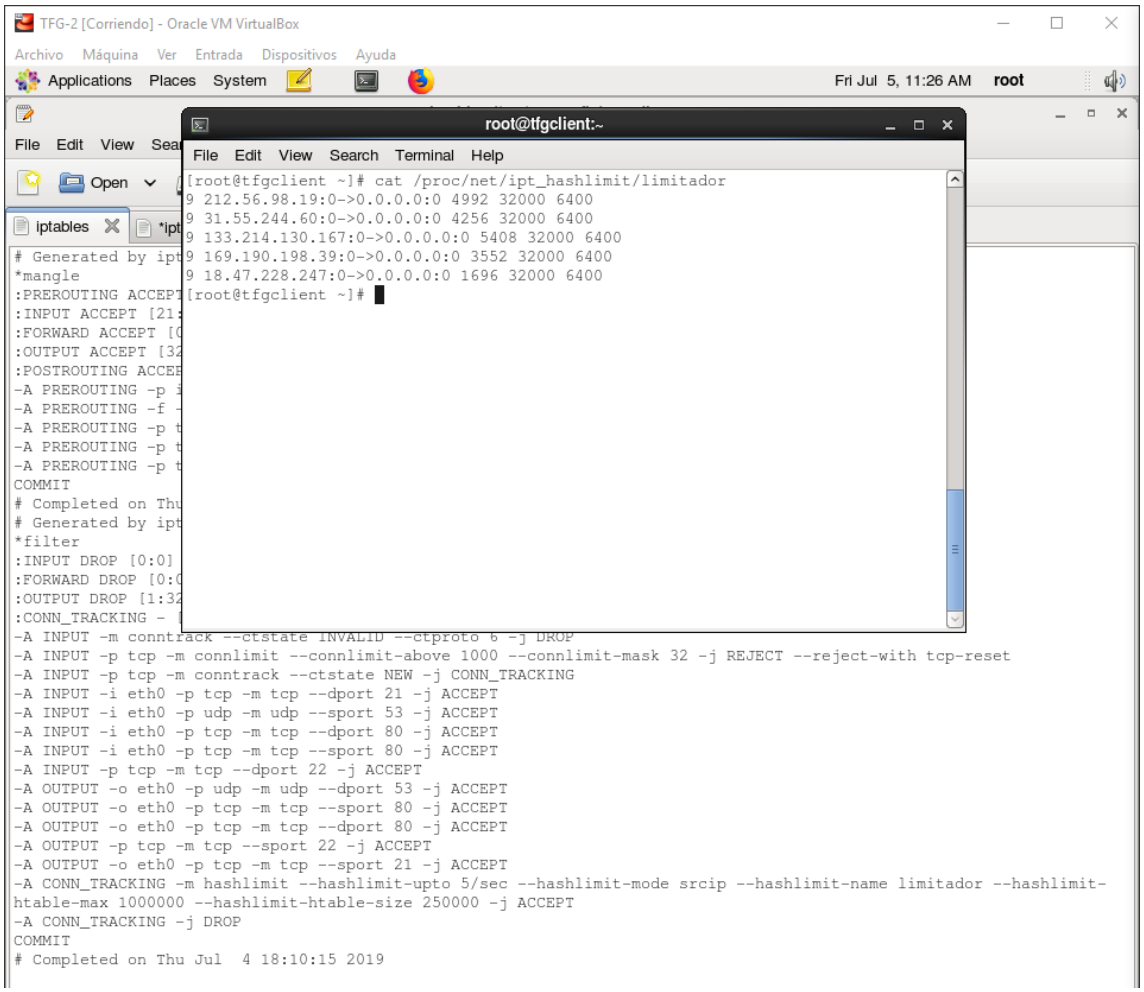

Ilustración 39: Tabla hashlimit.

9.7 Otras recomendaciones.

IPtables es una herramienta bastante útil para hacer frente a varios ataques de denegación de servicio distribuidos, pero estos ataques muchas veces son demasiado potentes en términos de transmisión por segundo de paquetes de datos. Por tanto, se recomienda no solo confiar en el uso de unas reglas de IPTables como las que se han visto en este documento. Además de estas reglas debemos tener a disposición una estructura de red capaz de soportar el ataque y, en caso necesario, dicha red debe estar bien preparada para desestimar tráfico mediante routers intermedios y de esa forma proteger el servidor al máximo antes de que las reglas establecidas en él tomen partido.

Los ataques DDoS varían normalmente en escala, pueden llegar a ser ataques en los que toman parte miles de máquinas distintas. Esto los convierte en impredecibles a ojos de los administradores de sistema que deben defenderse de ellos. Contra estos ataques toda seguridad es poca y aunque siempre se puede intentar estar preparado para estas situaciones, ni siquiera reglas como las implementadas en este estudio son completamente seguras.

10 Conclusiones y trabajos futuros.

10.1 Conclusiones.

Tras la finalización de este trabajo de fin de grado se puede concluir que los objetivos establecidos al comienzo han sido cumplidos, siempre en la medida de lo posible, puesto que muchas veces se ven problemas que necesitan de otras soluciones no alcanzables por unos medios específicos.

Como se ha visto a lo largo de este documento, un cortafuego no es infalible, pero es una herramienta de gran valor a la hora de defender un sistema. IPtables provee una gran utilidad como cortafuegos para poder asegurar los sistemas en los cuales se instala, no solo por el hecho de poseer unas cadenas de reglas que controlan los paquetes de red, sino porque posee varios módulos que ayudan a controlar muchos comportamientos de los paquetes, sin dejar atrás el hecho de que posee mucha documentación y muchas pruebas debido a la longevidad de su existencia y su amplia accesibilidad, por estas razones se eligió como cortafuegos para este proyecto.

Por otro lado, queda claro que existen situaciones en las cuales es necesario implementar otras medidas de seguridad complementarias de forma que, el cortafuego sea solo una primera instancia en la defensa de una máquina y detrás de él existan sistemas encargados de ayudar a mitigar posibles ataques.

La realización de este proyecto me ha ayudado con creces al desarrollo en el campo de la ciberseguridad. Un trabajo de este tipo de extensión ha requerido no solo de los conocimientos adquiridos en diversas asignaturas a lo largo del grado para poder llevar a cabo el montaje de la red, servicios, etc. sino de la lectura de mucha documentación para poder llevar a cabo las pruebas y simulaciones explicadas en este documento. Además, me ha ayudado a comprender algunas de las vulnerabilidades de los sistemas informáticos y a ver lo sencillo que es a veces explotar una de esas vulnerabilidades, llegando hasta el punto de que cualquier usuario, sin el más mínimo conocimiento en la materia, siguiendo unos pasos concretos es capaz de generar un daño muy considerable.

Muy utilizado está el dicho ***"la mejor defensa es un buen ataque"***, la realización de este trabajo como alumno de un grado en ingeniería informática, me ha ayudado a ver que en el campo de la seguridad informática, es necesario saber explotar una vulnerabilidad de un sistema para saber cómo defenderse de ella y este campo es a menudo pasado por alto en los grados universitarios, cosa que se me plantea irrisoria debido a la importancia que tiene y a que lo único que se consigue con eso es formar gente capaz de montar servicios y redes pero no de protegerlos correctamente.

Después de esta extensa conclusión y en vista de los resultados de las simulaciones llevadas a cabo en este trabajo, no se me ocurre mejor frase para cerrar el documento que la frase del profesor de la universidad de Purdue, Eugene Spafford:

"El único sistema totalmente seguro es el que está apagado, encerrado en un bloque de cemento, guardado en una habitación forrada de plomo y protegida por guardias armados. Incluso entonces, no apostaría mi vida por ello."

10.2 Trabajos Futuros.

A pesar de que muchos de los puntos en este trabajo de fin de grado sean finales, siempre se puede ahondar mucho más en la investigación realizada, como posibles ampliaciones de este proyecto deben encontrarse los siguientes puntos:

- Simulaciones de otros ataques existentes o los mismos, pero con otras técnicas/metodologías de ataque, con su correspondiente defensa
- Aplicación de posibles métodos de defensa complementarios a IPTables.
- Defensa de ataques mediante otros métodos, que podrían ser sistemas de detección de intrusos, monitorización de ataques, etc.
- Aplicación de las técnicas de ataque sobre un entorno real y no simulado.

11 Bibliografía

Calderon, P. (May, 2017 (Second Edition)). *Nmap: Network Exploration and Security Auditing CookBook.* Birmingham, United Kingdom: Packt Publishing Ltd.

Cheswick, W. R., & Bellovin, S. M. (1994). *Firewalls and Internet Security: Repelling the Wily Hacker.* AT&T and Lumeta Corporation.

Comer, D. E. (2000). *Internetworking with TCP/IP, Vol. I: Principles, Protocols and Architecture.* West Lafayette, Indiana, USA: Prentice-Hall, Inc.

Corp., O. (2019). *Oracle VirtualBox Documentation.* Obtenido de Oracle VirtualBox Documentation: https://docs.oracle.com/cd/E97728_01/E97727/E97727.pdf

Corp., O. (2019). VirtualBox UserManual. In O. Corp., *VirtualBox UserManual* (pp. 101-113). Oracle Corp.

Corp., O. S. (2019). *Kali Docs Official Documentation*. Retrieved from Should I use Kali Linux?: https://docs.kali.org/introduction/should-i-use-kali-linux

Corp., O. S. (2019). *Kali Docs Official Documentation*. Retrieved from What is Kali Linux?: https://docs.kali.org/introduction/what-is-kali-linux

Corp., O. S. (2019). *Kali Tools*. Retrieved from Hydra Package: https://tools.kali.org/password-attacks/hydra

Corp., O. S. (2019). *Official Kali Linux Documentation*. Retrieved from Documenting all the things: https://www.kali.org/kali-linux-documentation/

CSIRT-cv. (Marzo de 2018). *CSIRT-cv: Centro de Seguridad TIC de la Comunidad Valenciana.* Obtenido de NMAP 6: Listado de comandos: https://concienciat.gva.es/wp-content/uploads/2018/03/infor_nmap6_listado_de_comandos.pdf

Eychenne, H. (1998, 11 13). *iptables(8) - Linux man page*. Retrieved from Linux IPtables Manual.: https://linux.die.net/man/8/iptables

Firewall (Computing). (2019). Retrieved from Wikipedia.org: https://en.wikipedia.org/wiki/Firewall_(computing)

How to Create Botnet for D-Dos Attack with UFONet. (2016, Junio 15). Retrieved from Hacking Articles: https://www.hackingarticles.in/create-botnet-d-dos-attack-ufonet/

Kim, P. (2015). *The Hacker Playbook 2, Practical guide to penetration testing.* North Charlestone, South Carolina: Secure Planet LLC.

Messier, R. (2018). *Learning Kali Linux.* O'Reilly Media Inc.

Muñoz Blanco, J. A., & Henríquez Henríquez, V. M. (2007). *Seguridad en sistemas de red.* Las Palmas de Gran Canaria: Universidad de Las Palmas de Gran Canaria.

Networks, P. A. (2019). What is a denial of service attack? USA.

NMap.org. (2017). *Nmap Network Scanning*. Retrieved from NMap Scripting Engine - Script Format: https://nmap.org/book/nse-script-format.html

NMap.org. (2017). *NMap Network Scanning*. Retrieved from Nmap Scripting Engine (NSE): https://nmap.org/book/man-nse.html

Ocon Carreras, A. (1 de Marzo de 2012). Fundamentos en seguridad digital. Las Palmas de Gran Canaria, Las Palmas, España.

Packet Filtering HOW TO. (n.d.). Retrieved from netfilter.org: https://netfilter.org/documentation/HOWTO/packet-filtering-HOWTO-7.html

Rahalkar, S. (2017). *Metasploit for Beginners.* Birmingham, United Kingdom: Packt Publishing Ltd.

Rash, M. (2008, Julio 14). Mitigating DNS Cache Poisoning Attacks with iptables.

Sharma, H. (2017). *Kali Linux - An Ethical Hacker's Cookbook.* Birmingham, United Kingdom: Packt Publishing Ltd.

SoftwareTesting. (2019, Junio). *8 Best DDoS Attack Tools (Free DDoS Tool Of The Year 2019)*. Retrieved from SoftwareTesting.com: https://www.softwaretestinghelp.com/ddos-attack-tools/

SoftwareTesting. (2019, Junio). What Is DDoS Attack And How To DDoS? United States of America.

Solutions, K. D. (8 de april de 2017). Seguridad en un servidor DNS. Spain.

TEAM, S. (04 de Julio de 2018). 8 tips to prevent DNS attacks.

Thompson, J. (19 de Febrero de 2019). How to Prevent DNS Poisoning and Spoofing in 2019.

Tutorials, H. (2015, May 30). *Hacking Tutorials*. Retrieved from Open Port Scanning and OS Detection with Nmap in Kali Linux: https://www.hackingtutorials.org/scanning-tutorials/port-scanning-and-os-detection-with-nmap/

TutorialsPoint.com. (2019). *TutorialsPoint.com*. Obtenido de Kali Linux - Password Cracking Tools: https://www.tutorialspoint.com/kali_linux/kali_linux_password_cracking_tools.htm

12 Anexo I – Concesiones DHCP

```
# This lease file was written by isc-dhcp-4.1.1-P1
lease 192.168.1.8 {
  starts 5 2019/05/31 11:10:57;
  ends 5 2019/05/31 11:12:57;
  cltt 5 2019/05/31 11:10:57;
  binding state free;
  hardware ethernet de:ad:07:4f:5c:11;
  client-hostname "RU4OCYPC";
}
lease 192.168.1.7 {
  starts 5 2019/05/31 11:10:57;
  ends 5 2019/05/31 11:12:57;
  cltt 5 2019/05/31 11:10:57;
  binding state free;
  hardware ethernet de:ad:08:0b:c8:de;
  client-hostname "N4NOOM2R";
}
lease 192.168.1.11 {
  starts 5 2019/05/31 11:10:58;
  ends 5 2019/05/31 11:12:58;
  cltt 5 2019/05/31 11:10:58;
  binding state free;
  hardware ethernet de:ad:03:0c:06:08;
  client-hostname "0PPUYJHO";
}
lease 192.168.1.5 {
  starts 5 2019/05/31 11:39:59;
  ends 5 2019/05/31 11:49:59;
  cltt 5 2019/05/31 11:39:59;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:65:be:5a;
  client-hostname "tfgclient";
}
lease 192.168.1.6 {
  starts 5 2019/05/31 11:10:56;
  ends 5 2019/05/31 11:12:56;
  cltt 5 2019/05/31 11:10:56;
  binding state free;
  hardware ethernet de:ad:1a:0b:e0:8a;
  client-hostname "9B4Z5Y54";
```

13 Anexo II – fichero etter.dns configurado.

```
###############################################################################
# microsoft sucks ;)
# redirect it to www.linux.org
*.youtube.*          A          192.168.1.3
www.gmail.com        A          192.168.1.3
*.ulpgc.*    A          192.168.1.3
*.microsoft.com    A   107.170.40.56
www.microsoft.com  PTR 107.170.40.56     # Wildcards in PTR are not allowed
#########################################
# no one out there can have our domains...
www.alor.org  A 127.0.0.1
www.naga.org  A 127.0.0.1
www.naga.org  AAAA 2001:db8::2
#########################################
# dual stack enabled hosts does not make life easy
# force them back to single stack
www.ietf.org   A    127.0.0.1
www.ietf.org   AAAA ::
www.example.org  A    0.0.0.0
www.example.org  AAAA ::1
#################################################
# some MX examples
alor.org   MX  127.0.0.1
naga.org   MX  127.0.0.1
example.org MX 127.0.0.2
microsoft.com MX 2001:db8::1ce:c01d:bee3
#################################################
# This messes up NetBIOS clients using DNS
# resolutions. I.e. Windows/Samba file sharing.
LAB-PC*  WINS  127.0.0.1
#################################################
# some service discovery examples
xmpp-server._tcp.jabber.org SRV 192.168.1.10:5269
ldap._udp.mynet.com SRV [2001:db8:c001:beef::1]:389
#################################################
# little example for TXT records
naga.org TXT "v=spf1 ip4:192.168.1.2 ip6:2001:db8:d0b1:beef::2 -all"
# vim:ts=8:noexpandtab
```

14 Anexo III – Atacantes y configuraciones de red.

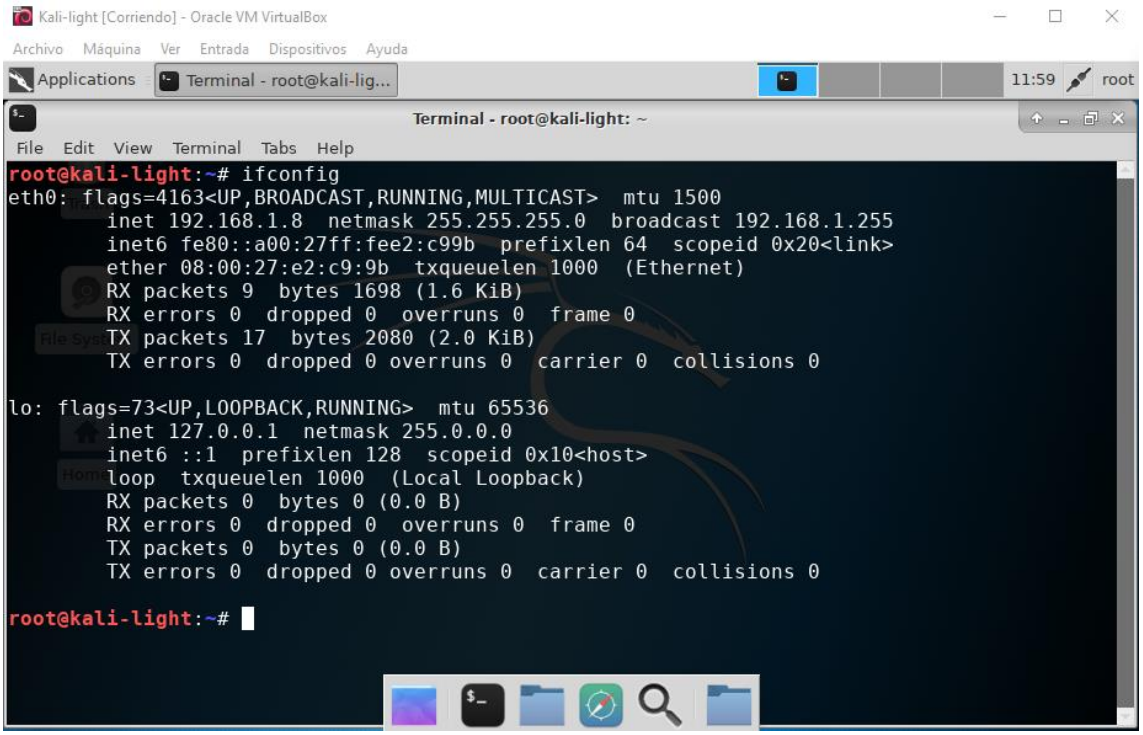

Ilustración 40: atacante 1.

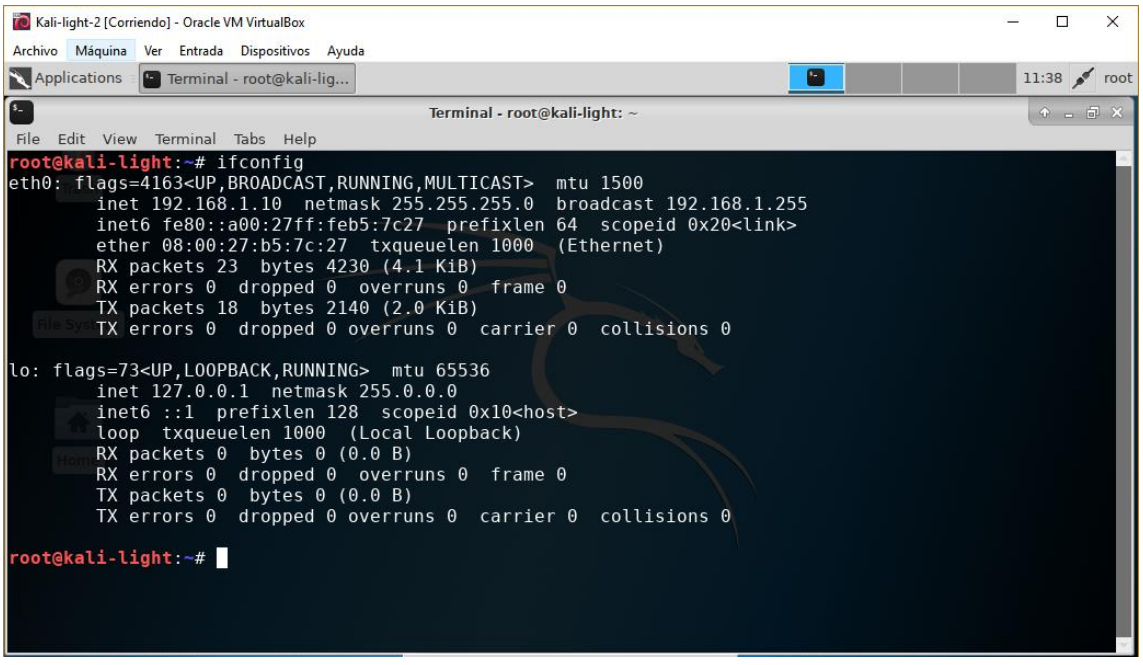

Ilustración 41: atacante 2.

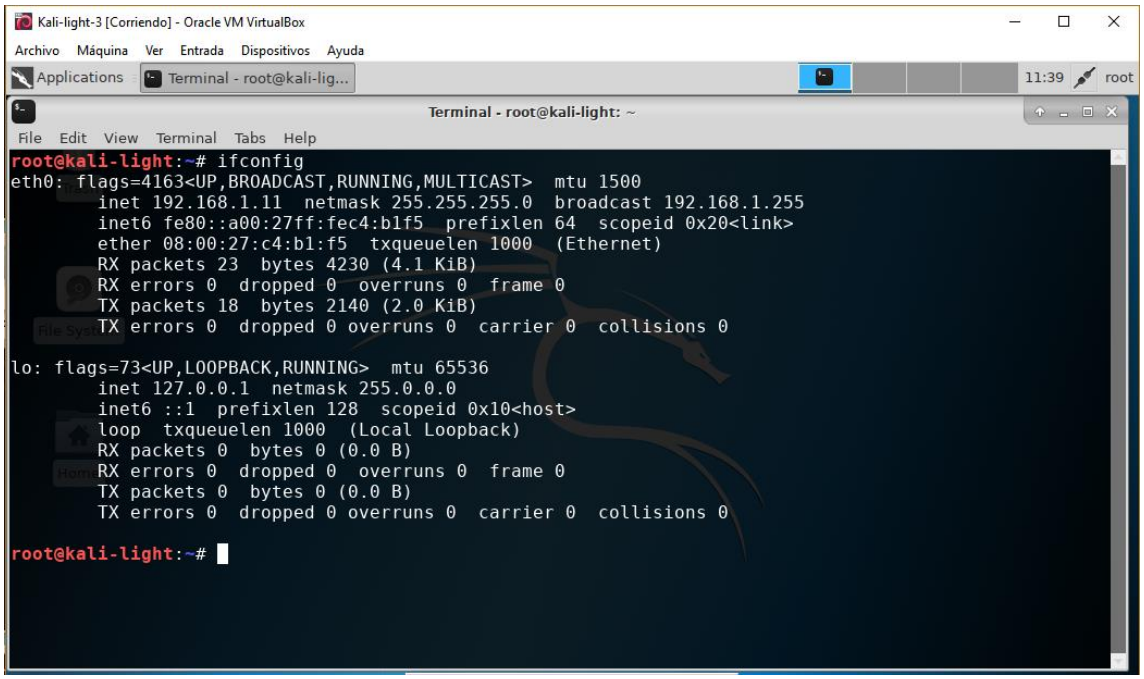

Ilustración 42: atacante 3.

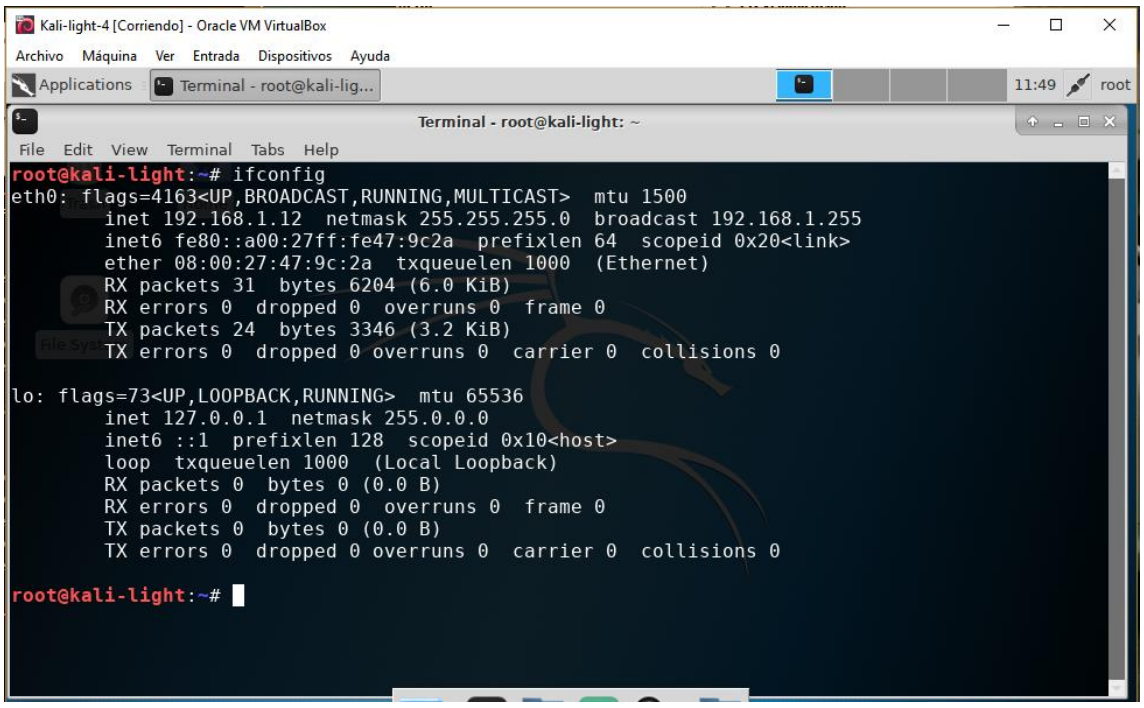

Ilustración 43: atacante 4.

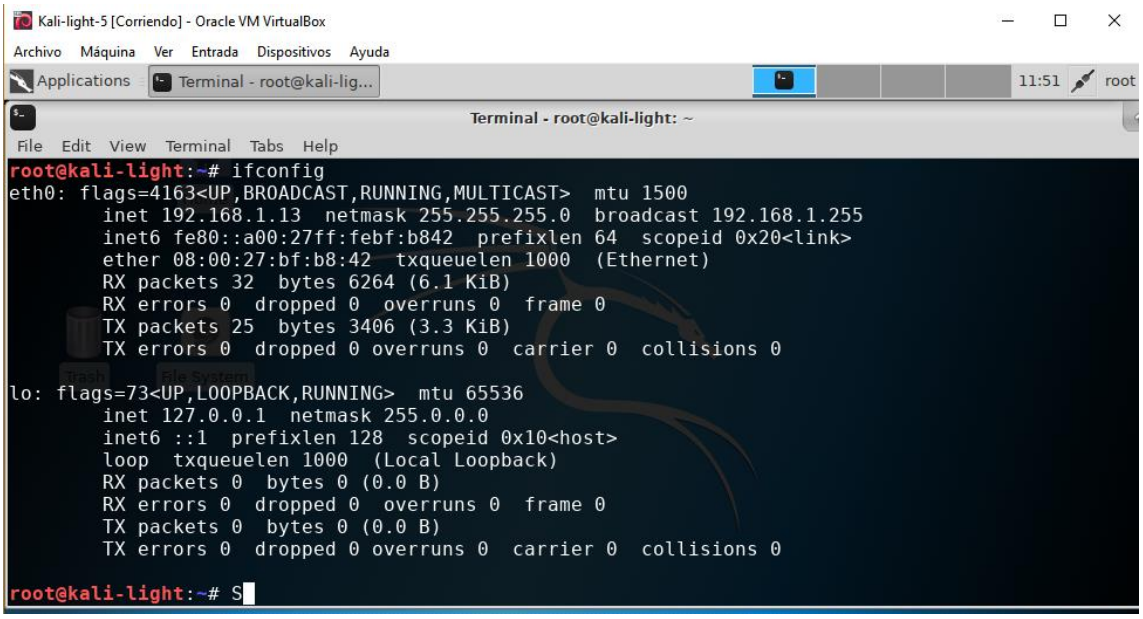

Ilustración 44: atacante 5.