UNIVERSIDAD DE GRANADA

Seguridad en Sistemas Operativos

GRADO EN INGENIERÍA INFORMÁTICA

Análisis de Ataques en un entorno gestionado por PfSense

Autores

Raúl Sánchez Fernández

Nazaret Román Guerrero

Adrián Ruiz López

Práxedes Martínez Moreno

Alejandro Poyatos López

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2019-2020

Índice:

1 Introducción

En todas las empresas y organizaciones se necesita proteger el sistema de ataques externos e internos para asegurarse de que no hay problemas de mal funcionamiento, filtración de datos, suplantación… y otros muchos problemas posibles.

Para protegernos es necesario tener un bloqueador de tráfico, un *firewall* o cortafuegos, que filtre la entrada de paquetes en nuestro sistema según lo que deseemos dejar pasar y lo que no. Éste actúa como una primera instancia de seguridad, que se utiliza en conjunto con con otras medidas de seguridad para que el sistema sea lo más robusto posible y puedan ser explotadas las mínimas vulnerabilidades.

En esta memoria vamos a tratar el funcionamiento de un router que actúa como *firewall* y que nos va a permitir filtrar el tráfico. Hablaremos de la estructura general del sistema que hemos levantado y configurado para esclarecer cómo llevaremos a cabo los ataques, de los cuales hablaremos en breve.

Vamos a llevar a cabo tres ataques diferentes a la estructura de red que hemos creado en el sistema y para demostrar las vulnerabilidades que se pueden explotar si no se configura correctamente el cortafuegos; entre esos ataques está: un túnel SSH, un ARP Spoofing y escaneo de puertos con NMAP.

2 Objetivo del Proyecto

El objetivo del proyecto es simular, monitorizar y asegurar un entorno lo más cercano a la red de una empresa mediante PfSense. Probar en él distintos ataques y observar qué marcas dejan estos en el sistema, de cara a poder informar de estos, mitigarlos o bloquearlos.

3 Estructura del Entorno

Vamos a simular una estructura típica de un entorno dentro de una empresa, por tanto, PfSense será nuestro *firewall*, a la vez que router y servidor DNS. PfSense contará con 3 interfaces:

- WAN: puerta de enlace con internet, modo puente/NAT con el host.

- LAN: subred con los equipos de trabajo de escritorio.
    - Gateway: 192.168.10.1
    - Hosts: 192.168.10.10-20
- OPT1: subred con el servidor web.
    - Gateway: 192.168.20.1
    - Hosts: 192.168.20.10-20

La red LAN contará con 2 clientes:

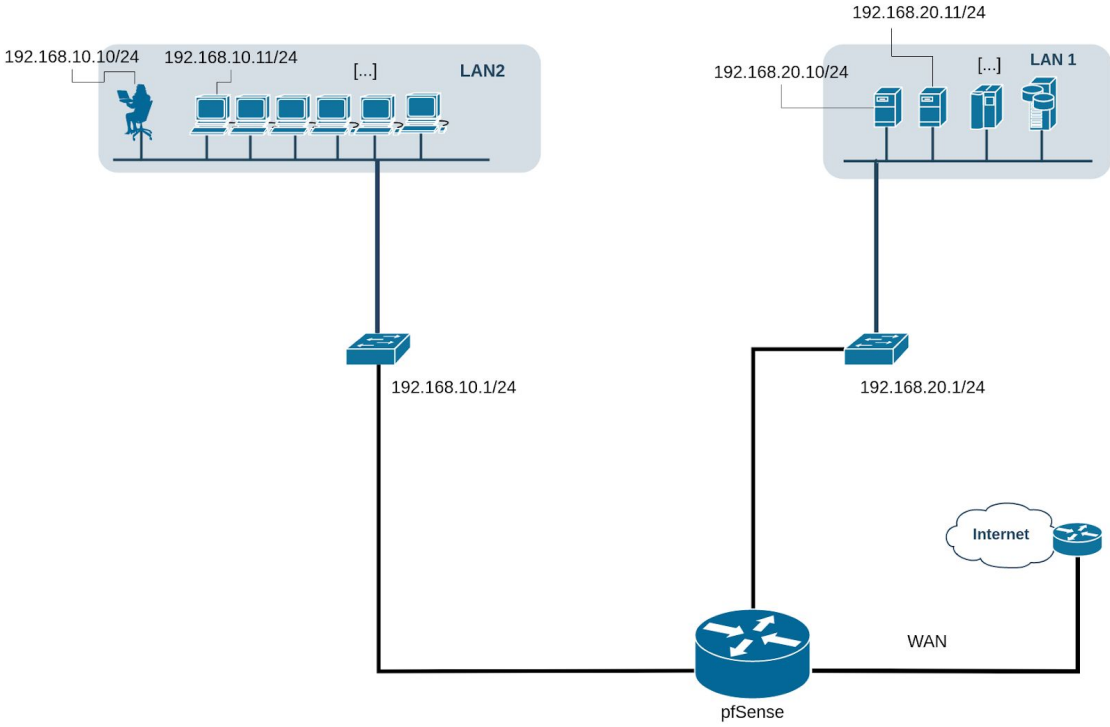
- Alice (Ubuntu 18.04 desktop), administradora de PfSense.
- Kali (Kali Linux 2019.4), equipo comprometido en el sistema.

La red OPT1 contará con un servidor web (Ubuntu Server 18.04) del cual Alice tendrá acceso SSH. Además, estará ejecutando un servicio web que consiste en una API RESTFUL en el puerto 3000.

PfSense actúa como servidor DNS para todas las máquinas de la red, sin embargo no es realmente un resolver, sino que redirige todas las peticiones a los servidores DNS de Google.

Por otro lado, PfSense utiliza Snort como sistema NIDS. De forma que analizará todos los paquetes salientes y entrantes en las interfaces WAN, LAN Y OPT1.

Los logs de PfSense son redirigidos y un servidor de logs remoto alojado en OPT1. El mismo que realiza la función de servidor web.



4 Fundamentos teóricos

¿Qué es un *firewall*?

Un *firewall* es un dispositivo de seguridad de la red que monitoriza el tráfico entrante y saliente y decide si debe permitir o bloquear un tráfico específico en función de un conjunto de restricciones de seguridad ya definidas. Estas reglas las define el administrador de seguridad del sistema en función de las necesidades de la empresa u organización.

Los *firewalls* han sido la primera línea de defensa en seguridad de la red durante más de 25 años. Establecen una barrera entre las redes internas seguras, controladas y fiables y las redes externas poco fiables como Internet.

Un *firewall* puede ser *hardware*, *software* o ambos.

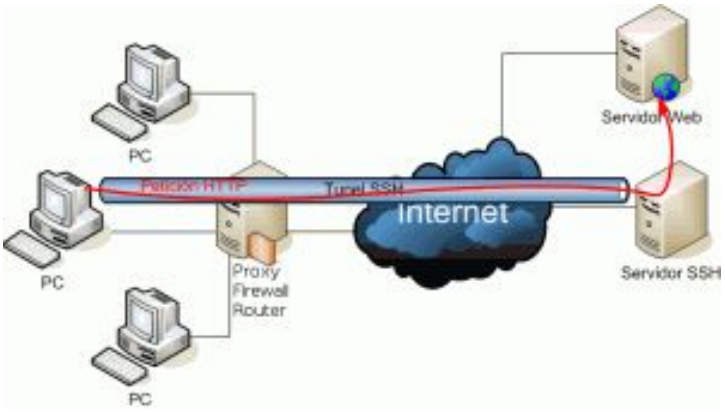
¿Qué es PfSense?

PfSense es una distribución basada en FreeBSD utilizada como *firewall* y router. Se caracteriza por ser de código abierto, por su portabilidad (puede ser instalado en una gran variedad de máquinas diferentes) y además cuenta con una interfaz web sencilla para su configuración. También existe *hardware* dedicado para la instalación de dicha distribución, el cual ha sido testeado, por lo que podemos confiar en que funciona correctamente.

Las principales funciones que realiza son:

- *Firewall*
- State Table
- Network Address Translation (NAT)
- Alta disponibilidad
- Multi-WAN
- Balance de carga
- VPN que puede ser desarrollado en IPsec, OpenVPN y en PPTP
- Servidor PPPoE
- Servidor DNS
- Portal Cautivo
- Servidor DHCP

Ataque túnel SSH



Cuando queremos acceder a un servicio web pero está restringido, podemos utilizar un túnel que nos conecte a través de localhost a dicho servicio, como si fuera nuestra propia máquina la que está sirviendo el contenido.

Establecer un túnel SSH nos puede proteger de una serie de ataques, sin embargo, también puede ponerse en contra nuestra si lo utiliza alguien malicioso. Es decir, se pueden crear túneles descontrolados de forma que perdamos la pista al atacante de nuestro sistema.

Si el atacante consigue comprometer nuestro servidor, este podría hacer peticiones a un servicio redirigiéndolas desde nuestro servidor a sí mismo y evitando que pasen por el *firewall*. Esto supone un fallo de seguridad, puesto que toda petición hecha desde localhost no es analizada por el *firewall*. Además aplicaciones que en un principio solo son accesibles en el servidor desde localhost, se verían expuestas al túnel.

Rastro que deja el ataque

Este ataque producirá un gran tráfico SSH en la red. Si no es común, será muy sospechoso. Por otro lado, los paquetes SSH serán más grandes de lo habitual, ya que en realidad contendrán paquetes HTTP en su interior.

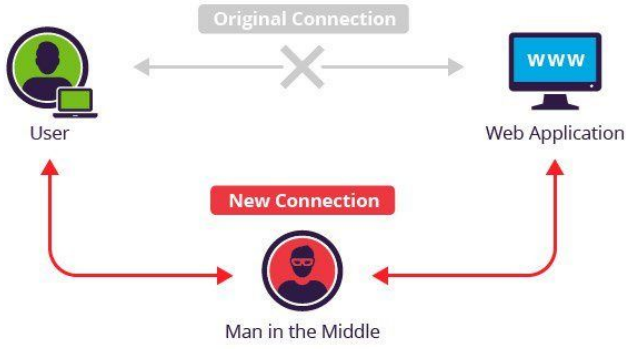
Protegerse contra el ataque

Para protegerse contra este ataque lo más conveniente es limitar las conexiones mediante reglas del *firewall* a los usuarios de confianza del sistema, incluso a máquinas únicamente destinadas a ello, para evitar cualquier tipo de intrusión. También conviene que el servidor SSH no exponga claves y tenga una lista blanca de hosts que puedan acceder al sistema por SSH.

Ataque MITM

Un ataque *Man In the Middle* es aquel que sitúa la máquina atacante en medio de una conexión, en este caso, la conexión será entre PfSense, que actúa como router, y un host que está navegando normalmente. Este último, en nuestra red del proyecto es una máquina virtual con Ubuntu instalado.

El atacante en nuestro caso es una máquina virtual con *Kali Linux* instalado y Ettercap que es la herramienta utilizada para interceptar el tráfico.



En concreto nuestro ataque se basa en *ARP Poisoning*, llamado así porque su cometido es alterar la tabla ARP para que el sistema "*malicioso*" duplique la dirección MAC para suplantar la de la víctima y recibir el tráfico entrante.

Esto se hace mediante el envío de mensajes ARP falsificados a una LAN y, como resultado, el atacante vincula su dirección IP con la de un equipo legítimo. Si lo consigue va a empezar a recibir todo el tráfico correspondiente a dicha IP y hacer con él lo que quiera.

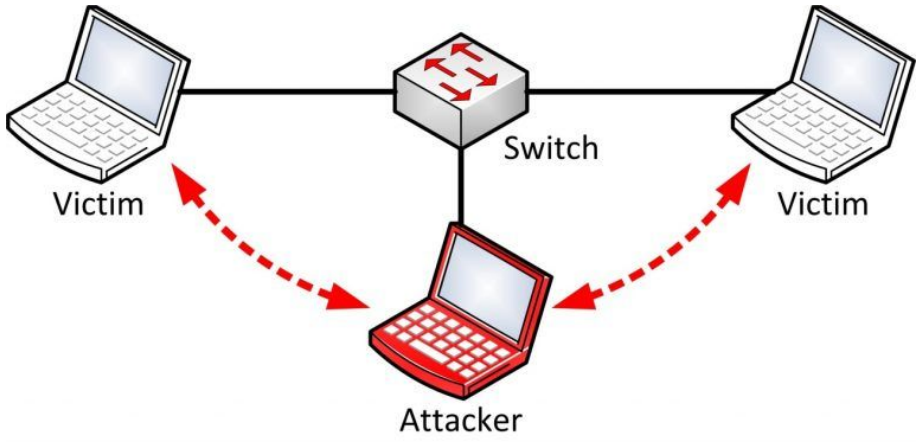
Algunos ejemplos de efectos que puede causar son:

- Ataques *DDOS* enlazando múltiples direcciones IP a la dirección MAC de una misma máquina, sobrecargando ésta e impidiendo su uso.
- Secuestro de sesiones; capturando el tráfico podemos acceder a instancias de sistemas donde se haya iniciado sesión, dándole acceso al atacante.
- Alteración de datos, dado que todo el tráfico pasa por la máquina atacante, esta puede retrasar o modificar los datos que recibe.

ARP Spoofing

Un ataque *ARP Spoofing* consiste en el envío de mensajes ARP (Address Resolution Protocol) falsificados a una LAN. Como resultado, el atacante autor del mismo vincula su dirección MAC con la dirección IP de un equipo legítimo (o servidor) en la red. Si esto se ha conseguido realizar con éxito, el atacante comenzará a recibir cualquier dato al que se pueda acceder mediante la mencionada dirección IP. El autor del ataque ahora puede elegir entre reenviar el tráfico a la puerta de enlace predeterminada real (a lo cual denominamos ataque pasivo) o modificar los datos antes de reenviarlos (ataque activo). Por otro lado, cabría la posibilidad de que el mismo atacante realizará un ataque de tipo *DoS* (denegación de servicio) contra la víctima, asociando a su dirección IP de la puerta de enlace predeterminada de la víctima con una dirección MAC inexistente.

Este tipo de ataque puede lanzarse tanto desde una máquina controlada, es decir, que el propio atacante ha conseguido controlar, lo cual conocemos como intrusión; como desde la máquina del autor del ataque si está conectada directamente a la red local Ethernet.



Algunos de los objetivos de este ataque suelen ser robar información sensible de una empresa, en su nivel más básico, o, aparte de esto, facilitar otros ataques tales como:

- Denegación de servicio.
- Secuestro de sesiones.
- Ataques de tipo *Man in the Middle* (del que hablábamos anteriormente).

Algunas formas de detectar, prevenir y proteger contra el *ARP Spoofing* son:

- Filtrado de paquetes.
- Utilizar software de detección de *ARP Spoofing*.
- Utilizar protocolos de red criptográficos.

Proceso para ejecutar el ataque

1. Situar la máquina atacante dentro de la red en la que realizará la intercepción de una conexión, que es requisito indispensable para ejecutar el ataque.
2. Seleccionar una víctima dentro de la red mencionada.
3. Encontrar la dirección IP de la puerta de enlace.
4. Iniciar el ataque mediante una herramienta, en nuestro caso `Ettercap`.
5. Seleccionar los dos objetivos del *ARP Poisoning*, siendo uno de ellos la víctima y el otro la puerta de enlace.
6. Empezar a recibir paquetes y analizar el tráfico.
7. (Opcional) Ejecutar alguna acción maliciosa como ataques de denegación de servicio, de alteración de datos o de secuestro de sesiones,

En nuestro caso el ataque elegido ha sido un secuestro de sesión para acceder a una web con las credenciales de la víctima, así como observar todas aquellas webs que vaya visitando, mostrandose en el navegador web del atacante (*mirroring*).

En concreto de una web `HTTP` (dtstc.ugr.es), ya que todo el tráfico de web `HTTPS` van cifrados con `SSL` y son muy complicados de descifrar.

Para intentar acceder a el tráfico de webs `HTTPS` hemos usado la herramienta `Bettercup` que cuenta con `SSLstrip` para intentar sortear este problema, pero actualmente la mayoría de navegadores interceptan el ataque *phishing*, mostrando la web como no segura por error de certificado inválido.

Rastro que deja el ataque

Teóricamente un ataque MITM es indetectable en webs que no usen `HTTPS`, en estas si el atacante intenta captar el tráfico es a través de otra web que se le muestra a la víctima, y que no tendrá los certificados que se esperan de la página, además esta será `HTTP` en lugar de `HTTPS`.

No es así para el caso del *ARP Poisoning*, donde existen diversas maneras de detectar que se está sufriendo un ataque de este tipo.

Dos de ellas son:

1. Observar la tabla ARP desde la configuración del router o desde cualquier host con acceso a la línea de comandos, con el comando `arp -a`, tanto en sistemas Windows como Unix accederemos a la tabla de direcciones donde podremos comprobar si existen varias direcciones IP asignadas a la misma MAC, en cuyo caso es muy probable que se esté sufriendo un ataque de este tipo.
2. Con herramientas como Wireshark que avisan con mensajes del tipo: (`duplicate use of <ip> detected!`).

Protegerse contra el ataque

La única forma conocida de protegerse contra un ataque MITM es desde la propia red, poniendo los medios físicos y digitales para evitar el acceso a nuestra red interna.

- En la parte física: Protección de entradas de los equipos y protección de elementos como los cables para la parte física.

- En la parte digital: Contraseñas fuertes, desactivación de la red wifi solo permitiendo el acceso a través de ethernet, eliminar la opción de conectar por Pin WPS.

Una vez el atacante se encuentra en nuestra red el ataque MITM es inevitable, pero de cara al *ARP Poisoning* hay distintas medidas efectivas para evitar su ejecución.

Algunas de ellas son:

- Asignación estática de direcciones en la tabla ARP: Este método es el más eficaz ya que asigna una IP estática a cada dispositivo y la asocia a su MAC. El problema es que obliga a operar de una manera muy concreta sólo sostenible en redes pequeñas donde no existan demasiados equipos.
- Conexiones encriptadas: Proteger el envío y la recepción de información imposibilita al atacante poder conocer el contenido de los datos más allá de saber que estos se estén enviando o no, aquellas webs con HTTPS y las transferencias efectuadas mediante SSH están protegidas con el protocolo SSL y es muy complicado de descifrar el contenido que protegen.
- Conexión mediante túneles VPN: Protege a equipos concretos, cifrando toda la conexión entre el cliente y el servidor final, pero no es viable dentro de una red grande de empresa.
- Filtrado de paquetes: Algunas redes pueden contar con algún tipo de protección contra paquetes maliciosos evitando los accesos a las tablas ARP y por ende el ataque, aunque este análisis continuo de los paquetes puede degradar el rendimiento o bloquear paquetes que realmente deberían haber sido enviados.

Escaneo de puertos con NMAP

Otro ataque que podríamos llevar a cabo es el ataque de escáner de puertos, que es uno de los más explotados a día de hoy; es utilizado por los administradores de seguridad de un sistema para saber qué puertos hay abiertos o cerrados y qué posibles fallos de seguridad puede ocasionar dicha configuración.

No obstante, también es utilizado por atacantes que desean explotar vulnerabilidades del sistema y que, como primer paso, deciden escanear los puertos abiertos, cerrados o protegidos con un *firewall* para diversos propósitos, como saber qué servicios hay activos y qué puertos están utilizando, qué puertos están abiertos y no protegidos para poder infiltrarse a través de ellos o incluso averiguar el sistema operativo de la máquina según los puertos que hay.

Según la respuesta que recibe el atacante de la máquina atacada, puede saber:

- El puerto está abierto si recibe un paquete SYN/ACK del atacado.
- El puerto está cerrado si recibe un paquete RST (un bit perteneciente al protocolo TCP utilizado para reiniciar la conexión).
- Está protegido si no devuelve contestación o si recibe un paquete ICMP con el puerto inalcanzable.

Hay diversas herramientas con las que llevar a cabo un escáner de puertos, entre ellas, arp-scan, Angry IP Scanner o NMAP.

```
nmap <IP a escanear>
```

Podemos utilizar distintas opciones para especificar más concretamente qué queremos analizar: qué redes, si queremos ver solo los puertos que cumplen una condición (como estár en un rango determinado, estar abiertos, ver los puertos de una red concreta…).

Rastro que deja el ataque

Hay que tener en cuenta que este ataque es muy ruidoso, puesto que se generan muchas peticiones SYN entre la máquina atacada y el atacante, por lo que es relativamente fácil de saber si alguien está escaneando los puertos de un sistema.

Protegerse contra el ataque

Para protegerse conviene establecer reglas que permita acceso solo a los puertos indispensables para que la máquina trabaje. El resto deben estar protegidos de

manera que la regla por defecto sea bloquear todo el tráfico que no se dirija a un puerto especificado en alguna regla que nosotros hayamos definido.

5 Resultados Obtenidos

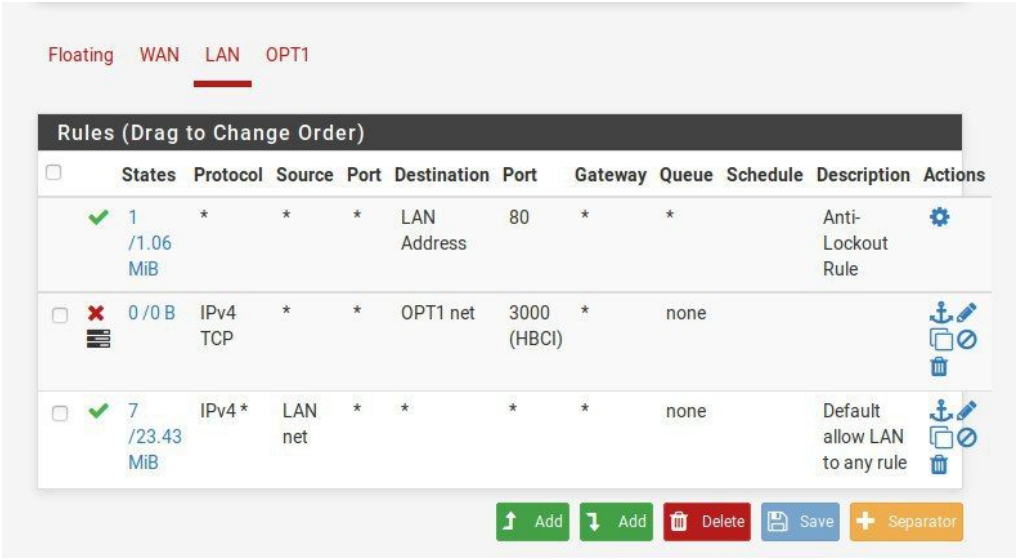
Nada más conectar nuestro *Kali Linux* dentro del sistema, podemos observar como PfSense detecta que existe dicha distribución y nos lo informa:

```
2019-12-10T14:33:47+00:00 192.168.20.1 snort[80331]: [1:2022973:1] ET POLICY Possible Kali Linux ho
name in DHCP Request Packet [Classification: Potential Corporate Privacy Violation] [Priority: 1]
UDP} 192.168.10.11:68 -> 192.168.10.1:67
```

Túnel SSH

Vamos a crear un túnel SSH que conecte dos máquinas.

Es importante tener el puerto 22 abierto y permitiendo el paso de tráfico a través de él, tal y como se ve en la imagen siguiente, donde la última regla permite el paso por todos los puertos que no tengan una regla previa, es decir, que esa última regla permite el paso de tráfico por el puerto 22. Sin embargo, no se permite el tráfico al puerto 3000, que es donde opera la aplicación.



Tras esto, creamos el túnel. Para ello basta con ejecutar:

```
ssh -L 3000:localhost:30000 webserver@192.168.10.11
```

De esta forma se abrirá una sesión SSH y todo el tráfico que reciba nuestro puerto 3000 será redirigido por el túnel al puerto 3000 del servidor, que nos devolverá la respuesta. Además el servidor verá la petición como si la recibiera desde localhost, y no desde nuestra máquina.

Comprobamos que no tenemos acceso directo, pero sí a través del túnel SSH que conecta localhost con la máquina que sirve las peticiones:

```
alice@alice-VirtualBox:~$ curl 192.168.20.11:3000
^C
alice@alice-VirtualBox:~$ curl localhost:3000
OKalice@alice-VirtualBox:~$
```

Además, podemos comprobar que el servidor de las peticiones las está recibiendo y procesando ya que el servidor está en modo desarrollo:

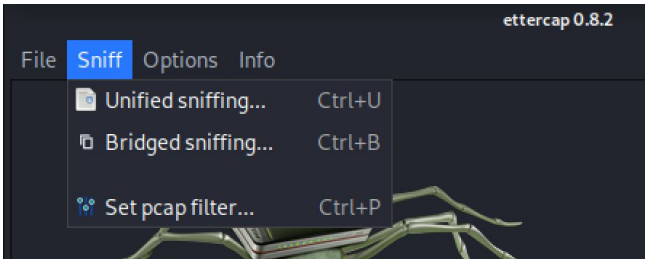
```
webserver@webserver:~/Proyecto-IV$ npm run start-dev

> prodproject@0.0.0 start-dev /home/webserver/Proyecto-IV
> nodemon -e js,yaml --ignore products.json ./src/bin/www

[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,yaml
[nodemon] starting `node ./src/bin/www`
Server in mode development
GET / 200 4.319 ms - 2
GET / 200 0.512 ms - 2
```

Ataque *Man In The Middle*

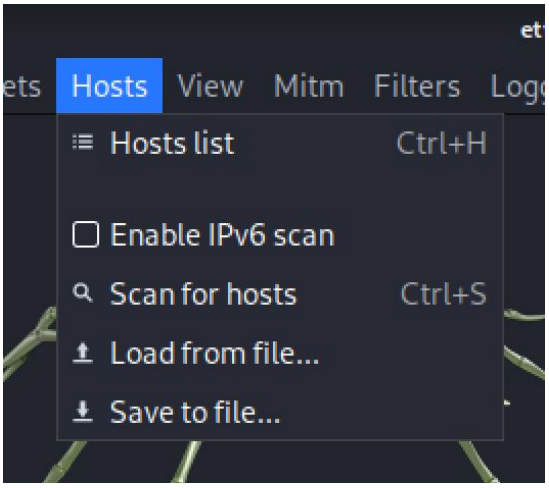
El primer ataque que hemos probado sobre nuestra infraestructura es un *ARP Spoofing* para ello, hemos iniciado un ataque desde la red LAN de *Man In The Middle* usando la herramienta Ettercap que trae *Kali Linux* instalado por defecto.

Partimos de una máquina ya situada dentro de la red, ya que para la demostración conviene centrarnos solo en el ataque.

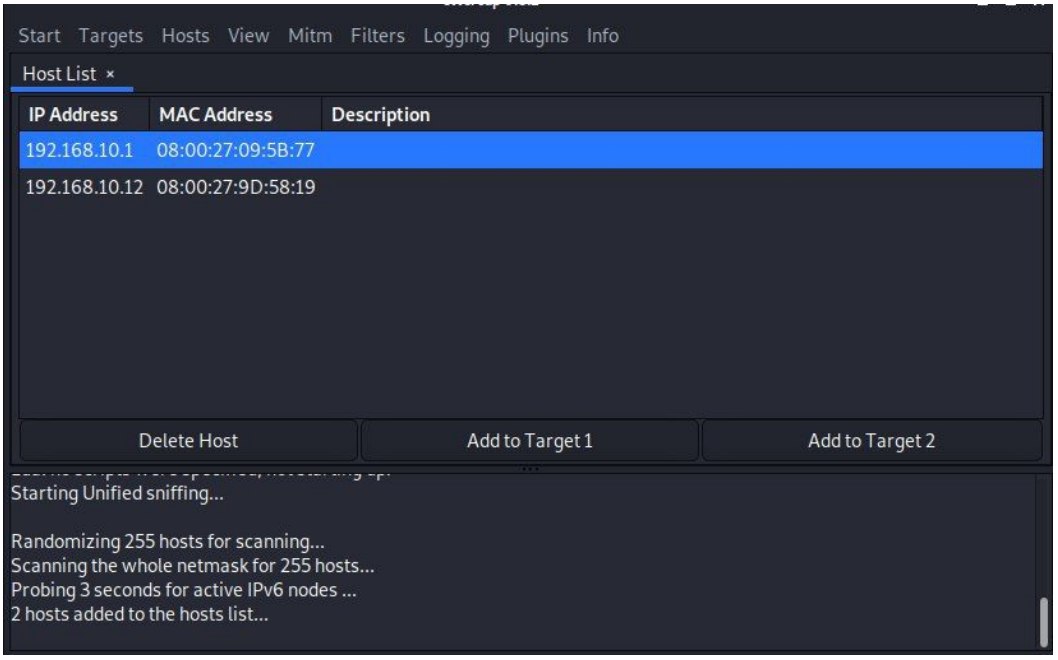
Para detectar a los hosts vulnerables en nuestra red empezamos a analizar el tráfico con Unified sniffing, y seleccionando la interfaz donde deseemos analizar el tráfico:



También podemos forzar su aparición realizando un escaner completo de la red con `Scan for hosts`:



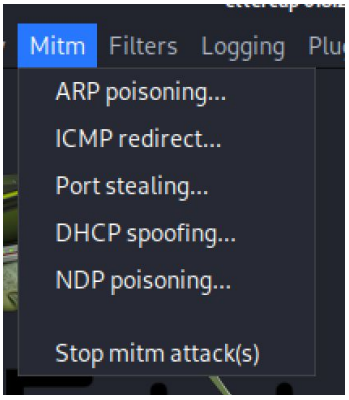
Tras esto si esperamos un poco los hosts irán realizando peticiones `HTTP`, momento en el que aparecerán en la lista de hosts para ser seleccionados para el ataque.



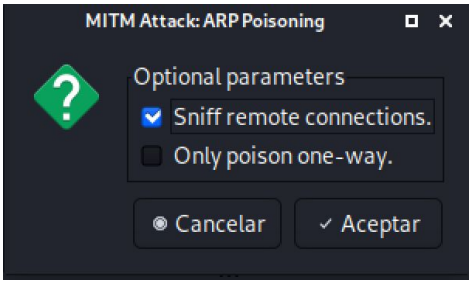
El siguiente paso es elegir nuestros objetivos, en nuestro caso solo hemos obtenido dos objetivos, pero en redes grandes es habitual ver muchos más.

Tras esto seleccionamos a la víctima, y la marcamos como Objetivo 1 en nuestro caso sería la IP (192.168.10.12) y la puerta de enlace cuya tabla `ARP` deseamos atacar como Objetivo 2, sería la IP (192.168.10.1) .

Tras esto iniciamos el ataque *ARP poisoning* en sí:



Seleccionamos en ambas direcciones, para disponer de toda la información posible:



Tras esto empezará a analizar por completo el tráfico de las dos IPs seleccionadas, ya que todo el tráfico pasará por la máquina atacante.

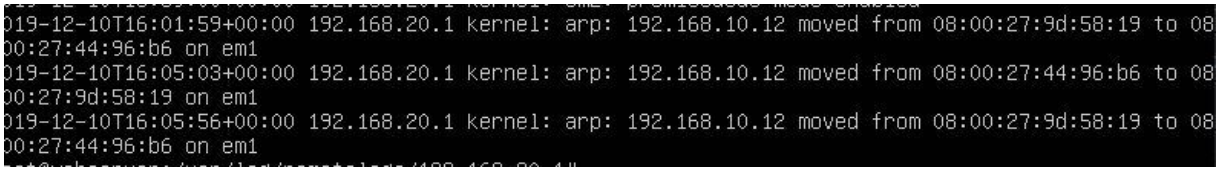
Permitiéndonos obtener cualquier contraseña que no esté protegida por el protocolo `HTTPS`, muestra de esto es un login realizado para entrar a administrar `PfSense`, y que ha sido capturada por `Ettercap`, donde como vemos hemos obtenido tanto el usuario, en este caso “admin” como la contraseña “secretpassword”.

```
HTTP : 192.168.10.1:80 -> USER: Sign+In PASS:  INFO: http://192.168.10.1/index.php
CONTENT: __csrf_magic=sid%3A6216f0457848a4bbc687b368c022cf7af30878a2%2C1575994656&usernamefld=admin&passwordfld=secretpassword&login=Sign+In
```

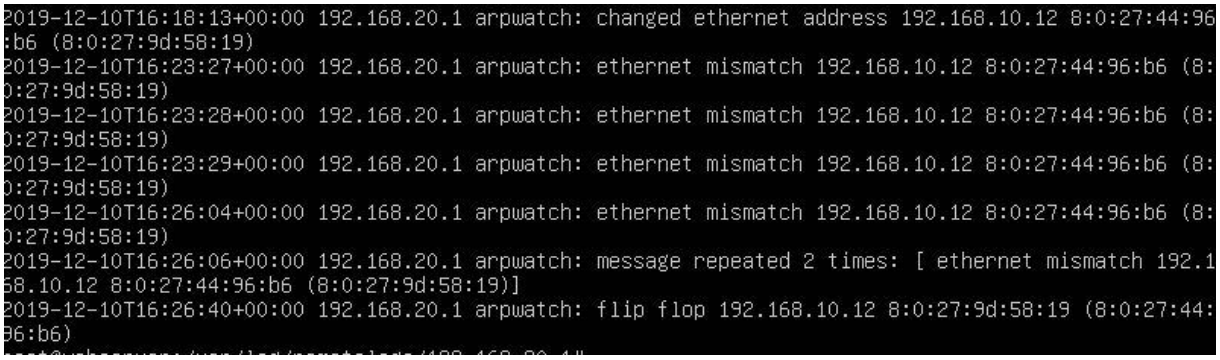
Este método hubiera funcionado para capturar cualquier formulario que hubiera rellenado la víctima en una web no protegida.

Detección ataque *ARP Spoofing*

Podemos observar en los logs de PfSense como nos detecta que se han modificado las tablas ARP:



También se pueden observar los cambios detectados por arpwatch, una extensión de PfSense encargada de monitorizar los cambios en la tabla ARP.



6 Conclusiones

Finalmente, hablaremos sobre lo que hemos sacado en claro durante la configuración del sistema y la ejecución de los distintos ataques orquestados entre las máquinas configuradas.

La configuración ha sido compleja, especialmente en la parte de la configuración del DNS puesto que PfSense es de por sí un servidor DNS pero por defecto no redirige el tráfico los servidores de DNS (el de google) y hay que habilitar esta opción manualmente.

Durante el ataque de construcción del túnel SSH nos hemos dado cuenta de que crear un túnel entre dos máquinas es sencillo, y supone un fallo de seguridad el hecho de que no se analice el tráfico de localhost, puesto que con SSH podemos acceder a ciertos sitios a través de localhost sin que se detecte.

Durante el ataque Man In The Middle hemos comprobado cómo de sencillo es que un atacante se interponga en mitad de una conexión establecida entre dos máquinas y reciba todas las peticiones y respuestas; observando como cambia la tabla ARP podemos ver que lleva a cabo el ataque y se sitúa en el centro de las comunicaciones.

Después de realizar el ataque de ARP-Spoofing nos hemos dado cuenta que no es seguro conectar con el webAdmin desde redes que no sepamos que son seguras, ya que al no tener una interfaz web HTTPS, si un atacante está llevando a cabo un análisis de la red, podrá obtener los credenciales de inicio sesion y podrían acceder dentro de nuestro firewall y modificar todo a su antojo, puesto que entrará con privilegios.

Por lo que la protección debería realizarse antes y no durante el ataque, lo cual puede empeorar aún más la situación.

También hemos comprobado que en estos últimos años la seguridad en general de internet ha aumentado bastante porque la grandísima mayoría de webs que visitamos están protegidas con el protocolo https, siendo muy complicado acceder a la información, obligando al atacante a tomar actitudes más drásticas que pueden hacer que se exponga o que alerten al usuario del peligro, ejemplo de esto es el intentar redirigir el tráfico https a uno no protegido (SSLStrip), que en los navegadores actuales advierte que la web que se visita no tiene adjunto un certificado válido.

Además, de forma general, nos hemos dado cuenta de que es muy compleja la administración de un sistema real si se cometen fallos en la configuración de las herramientas que gestionan la seguridad.

Es muy importante que la primera defensa existente en una red, es decir, el firewall, esté correctamente configurado para nuestras necesidades y que sea capaz de reducir al máximo las vulnerabilidades explotables en nuestra red o sistema, ya que, si éste falla, nuestra primera barrera ante ataques habrá caído y estaremos expuestos.

PfSense es una herramienta potente y con muchas posibilidades, que ofrece una amplia gama de plugins y módulos que ayudan a la securización de un sistema con conexiones a distintas redes y que puede evitar, o al menos, reducir los ataques que se pueden explotar. El funcionamiento de un firewall es simple pero efectivo, y teniendo un conjunto de reglas bien definidas puede facilitar mucho la gestión de la seguridad.

7 Bibliográfica

- https://itpro.outsidesys.com/2015/02/19/home-lab-with-pfsense-workstation/
- https://resources.infosecinstitute.com/setting-pentest-lab-pfsense-virtualbox/#gref
- https://medium.com/@marvin.soto/qu%C3%A9-es-el-envenenamiento-arp-o-ataque-arp-spoofing-y-c%C3%B3mo-funciona-7f1e174850f2
- https://es.wikipedia.org/wiki/Suplantaci%C3%B3n_de_ARP#Defensas
- https://es.wikipedia.org/wiki/Esc%C3%A1ner_de_puertos
- https://es.wikipedia.org/wiki/RST_(flag)
- https://linux.die.net/man/1/nmap