Universidad de Vigo

Escuela de Ingeniería de Telecomunicación

MEMORIA GRUPO C SEGURIDAD

Autores:

Axel Valladares Pazó
Juan Anselmo López Gómez
Samuel Yáñez Delgado

Curso 2021/2022

1. Introduction

En esta memoria se recogen la implementación, puesta en escena y contramedidas de los ataques llevados a cabo sobre una topología de red (Figura 1) en el programa GNS3.

Haciendo uso de las herramientas de kali (Yersinia) hemos podido llevar a cabo una serie de ataques para demostrar la vulnerabilidad de una red Ethernet y que contramedidas podemos llevar a cabo para evitar dichas fisuras. La topología consta de 3 switches haciendo un spanning-tree: el switch 1 (SW1) será nuestro raíz, y estará conectado al router principal. El router principal hace de servidor DHCP y hace NAT para la conexión con Internet. Además contamos con una VLAN en la que se alojan todos los switches y Host de la topología. Nuestros Host constan de un ATACANTE (Kali) y dos VICTIMAS (Host1 y Host2), las cuales estarán conectados a distintos switches dentro de la misma VLAN. La subred en la que se encuentran todos los miembros de la topología es 198.168.100.0/28 El atacante cuenta con la IP 192.168.100.4 La IP asociada a la interfaz del router es 192.168.100.1

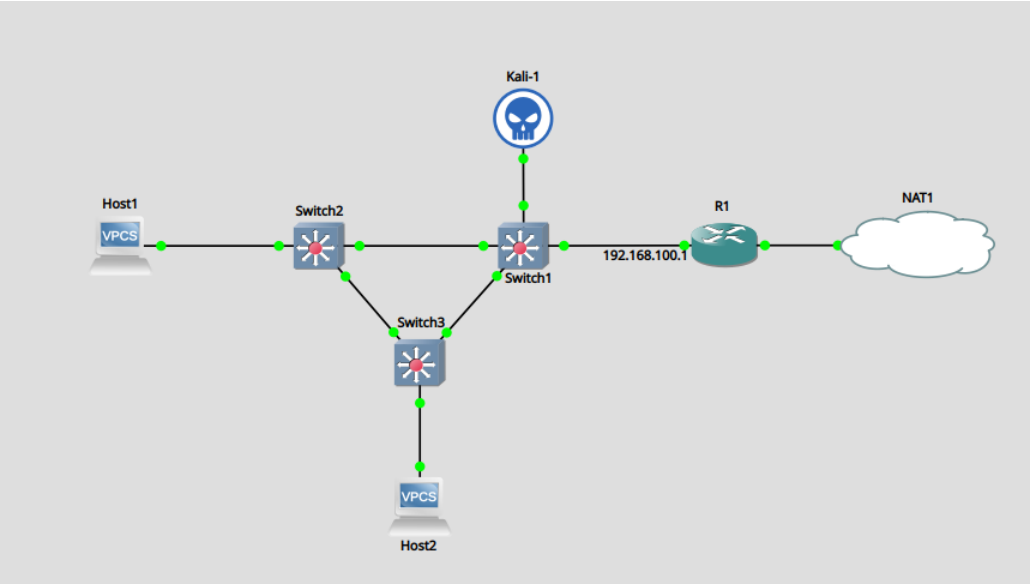


Figura 1: Topología de la red utilizada para la experimentación del proyecto

2. DHCP Starvation

Descripción: Consiste en agotar la pool de direcciones IP del servidor DHCP (router1) impidiendo así que nuevos dispositivos se puedan conectar a la subred de forma dinámica. Haremos uso de la herramienta Yersinia para saturar al router generando de forma automática MACs aleatorias que soliciten una IP.

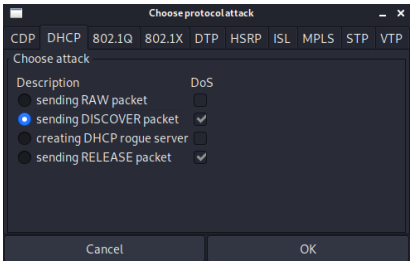
Puesta en práctica: Mediante una serie de capturas veremos la funcionalidad del ataque DHCP. Primero, desde la máquina del atacante (Kali), preparamos el escenario en Yersinia, donde nuestro objetivo será generar una gran ristra de paquetes DISCOVER que lleguen hasta el servidor DHCP.

Segundo, capturando los paquetes en la interfaz de entrada del router observamos como se llena de paquetes DISCOVER, todos ellos con una MAC distinta y que provienen del mismo destino, en este caso con IP 0.0.0.0 (un host sin IP).

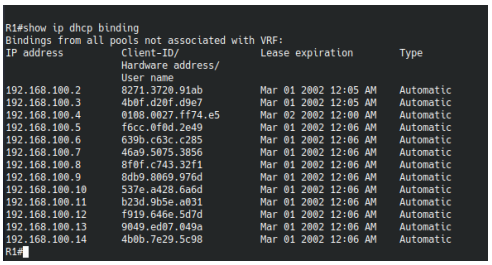
Tercero y como resultado del ataque, observamos que haciendo uso del comando **show ip dhcp binding**, la pool del router está toda en uso, impidiendo así que cualquier persona que quiera conectarse a la subred pueda obtener de forma dinámica su IP.

Una variante de este ataque es el **DHCP Exhaustion**, que básicamente su implementación es la misma que la del Starvation, con la diferencia de que dejaremos que Yersinia genere una cantidad ingente de paquetes durante un tiempo. Cuando alguien haga una solicitud IP al router y este mande el paquete OFFER , la red estará tan saturada que será imposible que el paquete llegue hasta el Host que solicitó la IP, generando así una denegación de servicio.

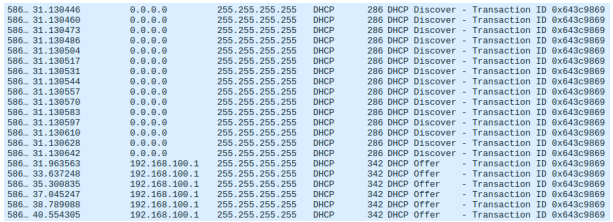
Contramedidas:DHCP Snooping es lo suficientemente inteligente como para leer la carga útil del protocolo DHCP y verificar que la dirección MAC de origen y CHADDR (campo Client Hardware Ip Addres de DHCP) sean iguales (comando opcional ip dhcp snooping verificar mac-address). También es posible establecer un umbral máximo, o el número de paquetes por segundo que el switch puede recibir en un puerto determinado, de modo que, si el número de paquetes DHCP alcanza este umbral, el puerto entra en modo de apagado (bloqueo) y generaría una advertencia sobre el DoS.



(a) Configuración del Atacante



(b) Configuración de las IPs del servidor DHCP tras el ataque



(c) Paquetes DHCP enviados por el atacante

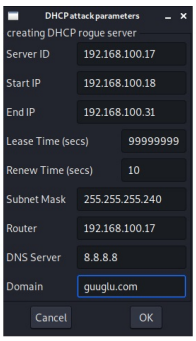
Figura 2: Demostración de DHCP Starvation

3. DHCP Rogue Server

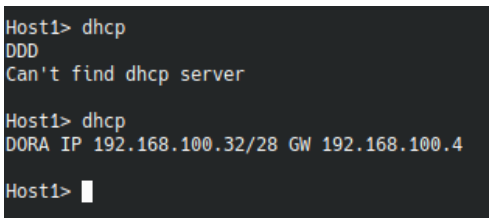
Descripción: El objetivo principal es convertirse en el servidor DHCP. Para poder lograr esto debemos juntar este ataque con otro para poder denegar al router que suministra las IPs, ya que el primero que mande el OFFER será el servidor DHCP. Para impedir esto podemos hacer uso del ataque mostrado en el punto 2 (DHCP Starvation), para dejar al servidor DHCP sin IPs y así poder inyectar nuestros propios paquetes OFFER. Otro ataque con el que podemos convinar el Rogue Server es un ataque al Spanning-Tree, donde hagamos que el tráfico de paquetes pase por le atacante y así ganar la çarrera"del OFFER del DHCP. Para nuestra puesta en práctica haremos uso del ataque Starvation como ataque auxiliar.

Puesta en práctica: Tenemos un router (router1) como servidor DHCP. Agotamos todas las IPs de este servidor con el ataque DHCP Starvation y configuramos el Rogue Server con la herramienta Yersinia de la forma:
Una vez que el host hace una petición DHCP, esta no será contestada por el router1, ya que tiene todas las IPs dadas a las MACs aleatorias creadas por el DHCP Starvation. Por ende, le responderá la maquina del atacante (Kali), quedando asignada la IP que se haya indicado en el DHCP Rogue Server y definiendo como Default Gateway la maquina virtual, por lo que a su vez estaremos haciendo un Man In The Middle.

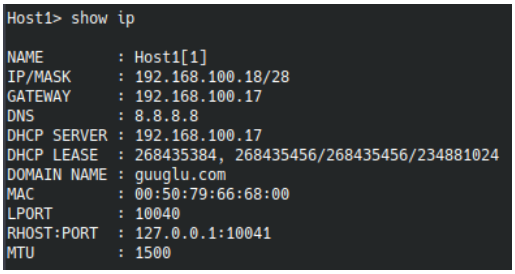
Contramedida: Al no haber una autenticación entre el cliente y el servidor DHCP, es muy sencillo realizar este ataque sin las medidas de seguridad tomadas por parte del administrador de la red. La mejor forma de impedir este ataque es haciendo uso de **DHCP Snooping**. Principalmente lo que hace es impedir que los puertos de un switch admitan respuestas DHCP menos el que está conectado al router y los del Spanning-Tree. De esta forma impides que un atacante pueda inyectar sus propios paquetes de respuesta a la petición DHCP.



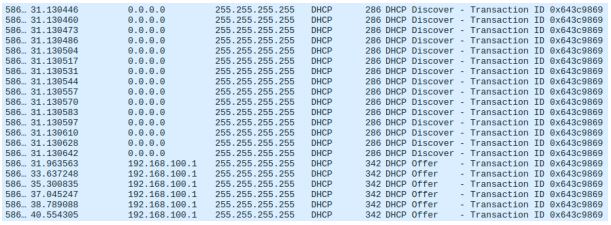
(a) Configuración del servidor DHCP del Atacante



(b) Host adoptando una IP del servidor DHCP del Atacante



(c) Configuración IP del Host atacado



(d) Paquetes DHCP enviados por el atacante

Figura 3: Demostración de DHCP Rogue

4. ARP Poisoning

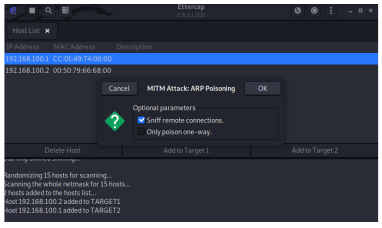
El protocolo ARP permite que las comunicaciones de red puedan llegar correctamente a su destino. Su objetivo es traducir las direcciones IP en una dirección MAC y viceversa. Se encarga de formar una tabla con la pareja IP-MAC, y que los diferentes equipos de la red local puedan comunicarse entre ellos y con el router (para acceder a Internet) sin problema. El router también cuenta con una IP de la LAN y una dirección MAC.

Descripción: El ataque se basa en envenenar la tabla ARP de la víctima. Haciendo esto podemos lograr 2 cosas:

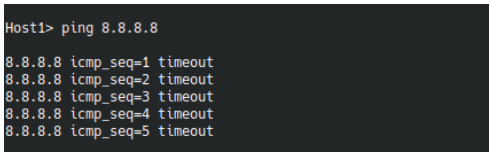
- **DoS:** si el atacante no reenvía las conexiones al router para salir a Internet, estaremos haciendo una denegación de servicio.
- **Man In The Middle:** si el atacante reenvía las conexiones al router para salir a Internet, será capaz de interceptar cualquiera de estos paquetes para su posterior estudio o modificar cualquier tipo de información que vaya por ellos.

Puesta en práctica: Trabajaremos con Ettercap_graphical. Seleccionamos la tarjeta de red, escaneamos la red local y elegimos la IP de la víctima como TARGET 1 y la del router a suplantar como TARGET 2. En el menú de selección de MITM elegimos ARP Poisoning y Sniff Remote Connections y desmarcamos la opción de only poison one_way (ya que queremos hacer MITM).

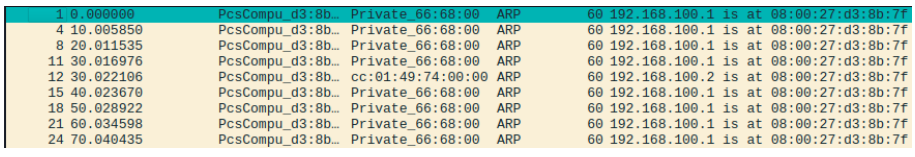
Contramedidas: Una forma es cifrar todas nuestras comunicaciones para que, aunque puedan capturar toda la información, no puedan leer absolutamente nada. Lo más sencillo será comprobar la tabla de ARP, si hay dos direcciones MAC con la misma IP significará que alguien nos está haciendo un Man In The Middle.
Lo más útil es utilizar tablas estáticas y renegar de cualquier petición ARP entrante que no se encuentre en la tabla. El problema de esta medida es que añade latencia por culpa de los tiempos de búsqueda y requiere mantener actualizadas las entradas en todas las maquinas de la subred. Otra forma de mitigar este ataque es haciendo uso de software que nos comprueben la tabla ARP y nos notifiquen cuando una IP tenga varias MACs asociadas.



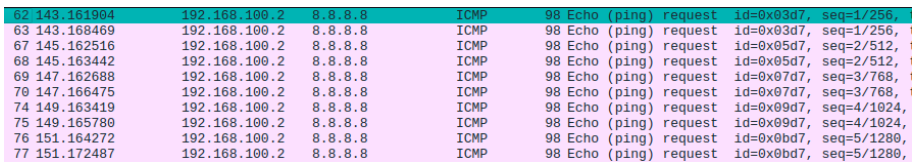
(a) Configuración del ataque a ARP



(b) Host víctima sin conexión



(c) Paquetes ARP enviados a la víctima



(d) Paquetes de ICMP de la víctima enviados al atacante

Figura 4: Demostración de ARP Poisoning

5. Ataques al Spanning-Tree (STP)

El Spanning-Tree se da cuando se conectan más de 2 switches juntos. El protocolo permite a los switches activar o desactivar automáticamente los enlaces de conexión, de forma que se garantice la eliminación de bucles. Para ello se estipula un switch como switch raíz.

Nota: Para la prueba del ataque, el atacante estará conectado al Switch2 en lugar del Switch1 como en el resto de casos.

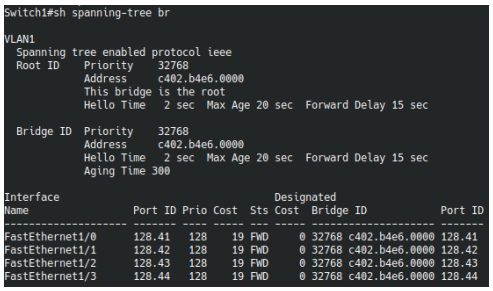
Descripción: El principio de este ataque es intercambiar la prioridad del switch al que está conectado el atacante, de forma que todo el árbol de switches deba modificarse para adaptarse a las nuevas características del árbol. El hacer esto propicia a realizar ataques DoS o MITM con mayor facilidad, ya que el árbol STP está redirigiendo el tráfico por el switch donde está conectado el atacante.

Puesta en práctica: Haciendo uso del programa Yersinia se puede reclamar el ser el switch raíz de la forma (Figura a). Observamos en los switches que el árbol ha sido modificado mediante el comando :

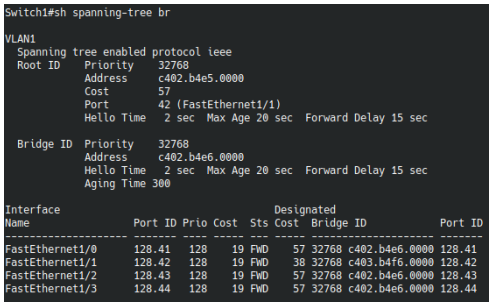
```
Switch# show spanning-tree brief
```

Contramedidas: BPDU Guard es una funcionalidad que al detectar que se reciben tramas BPDU en un puerto lo deshabilita, de forma que tengamos que habilitarlo de forma manual.

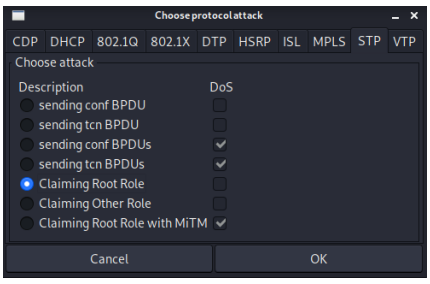
```
Switch(config-if)# spanning-tree bpduguard enable
```



(a) Configuración del Switch1 antes del ataque



(b) Configuración del Switch2 después del ataque



(c) Inicio del ataque

Figura 5: Demostración de ataque al STP

6. DNS Spoofing

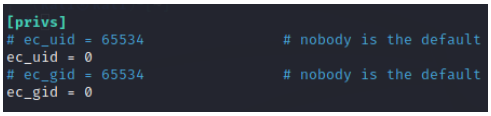
Descripción: Consiste en suplantar la identidad de un servidor legítimo DNS para capturar peticiones DNS de nuestra víctima, y de esta forma poder desviar el tráfico hacia servidores propios del atacante (Kali). Ademas, realizando un MITM podremos observar las credenciales intercambiadas entre la víctima y el servidor legítimo.

Puesta en práctica: Cambiamos los ficheros /etc/ettercap/etter.conf y /etc/ettercap/etter.dns y añadimos (Figura 1) y (Figura 2) respectivamente.
Tras modificar estos dos archivos, bastará con ejecutar el comando :

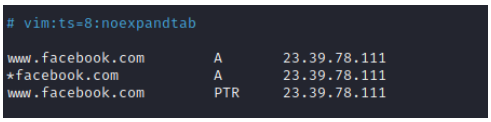
```
> ettercap -T q -i eth0 M arp -P dns_spoof
```

Si realizamos un ping a facebook (facebook.com, www.facebook.com, etc) desde el host víctima, observamos que el ping se redirige a la ip de nuestra elección (en este caso la 23.39.78.111).

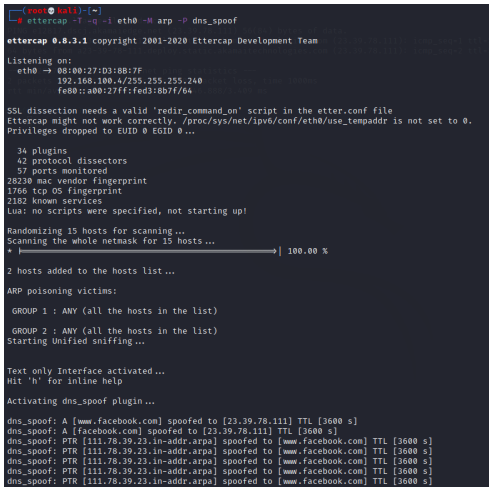
Contramedidas: Si queremos evitar sufrir algún tipo de ataque de este estilo podemos tomar una serie de medidas, como podría ser el caso de comprobar los indicadores de HTTP/HTTPS. Si una página usa el indicador HTTPS y nos sale que es HTTP, podremos observar que no se trata de la página que hemos solicitado, y que estaremos bajo un ataque de este estilo. Otra medida a tomar sería observar el tráfico de la red y observar como las peticiones DNS van hacia el dispositivo que nosotros queremos, y no hacia alguien externo.



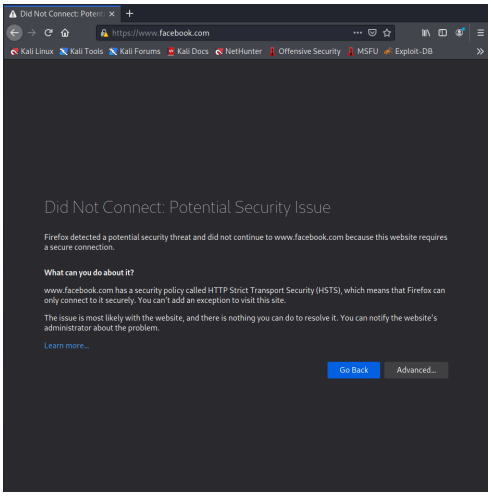
(a) Configuración de etter.conf



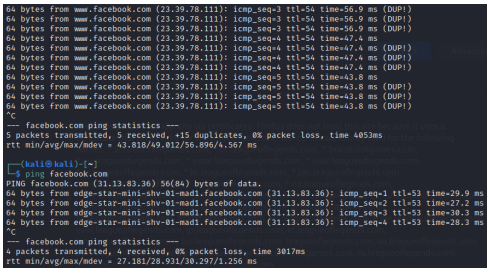
(b) Configuración de etter.dns



(c) Recogida del tráfico a la web cambiada



(d) Fallo en el certificado de la página



(e) Ping de la víctima a la IP cambiada

Figura 6: Demostración de DNS Spoofing

7. Smurfing

Descripción: El ataque Smurf es un ataque de denegación de servicio distribuido (DDoS) que deja inoperativa la red. Este ataque se lleva a cabo aprovechando las vulnerabilidades del protocolo IP e ICMP. El ataque consiste en inundar con paquetes ICMP la dirección Broadcast con la IP spoofeada de la víctima a modo de origen. Si el Router envía el tráfico a esas direcciones de Broadcast, la mayoría de los host tomarán los mensajes echo request de ICMP y lo responderán, multiplicando el tráfico por cada host de la subred, provocando que todas esas respuestas vuelvan a la IP de origen (la IP de la víctima atacada).

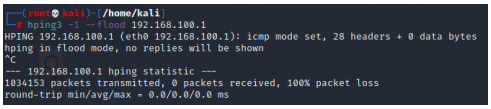
Puesta en práctica: Para realizar este ataque haremos uso de la herramienta hping3 de la forma:

```
> hping3 -1 --flood IProuter
```

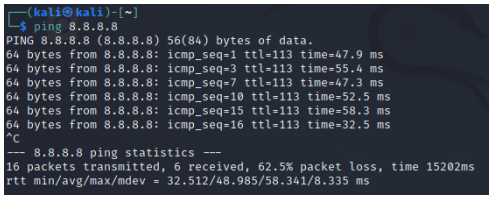
Una vez spoofeada la dirección IP de la víctima ejecutamos el comando anterior y ya se produciría el ataque. Si abrimos un Wireshark desde la víctima, podemos ver la inundación de paquetes ICMP producidos por el ataque, dando lugar a una denegación de servicio.

Contramedidas: Para protegernos de los ataques Smurf podemos optar por las siguientes medidas para limitar la probabilidad e impacto de este ataque:

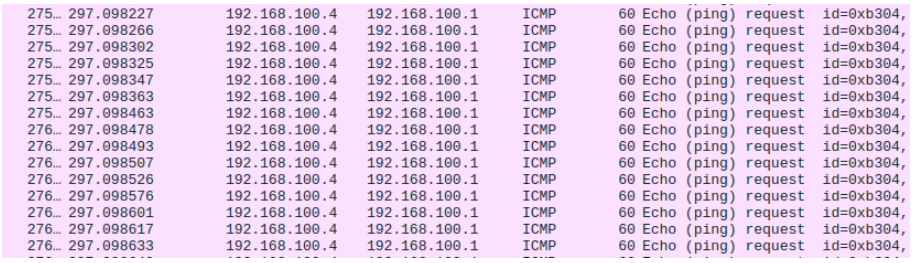
- Bloquear el tráfico de difusión directa que ingresa en la red.
- Configurar los hosts y enrutadores para que no respondan solicitudes de echo request de ICMP.



(a) Configuración del atacante



(b) Pérdida de paquetes de la víctima



(c) Paquetes enviados a la víctima

Figura 7: Demostración de Smurfing

8. Fraggle:

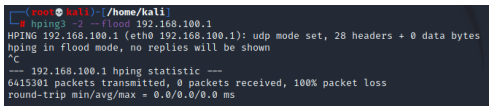
Descripción: Una vez visto el ataque Smurf, se creó una variante a este (por el mismo creador de Smurf) llamado ataque Fraggle. Este ataque es una variante del Smurf, siendo prácticamente idéntico a este. Lo que cambia con respecto a Smurf es que en lugar de realizar el ataque usando ICMP se hace mediante paquetes UDP, provocando el mismo resultado que el ataque original.

Puesta en práctica: Para ralizar Fraggle haremos uso una vez más del comando hping3 de la siguiente manera:

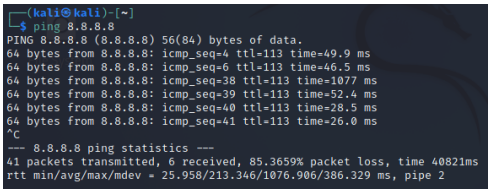
```
> hping3 -2 --flood IProuter
```

Contramedidas:

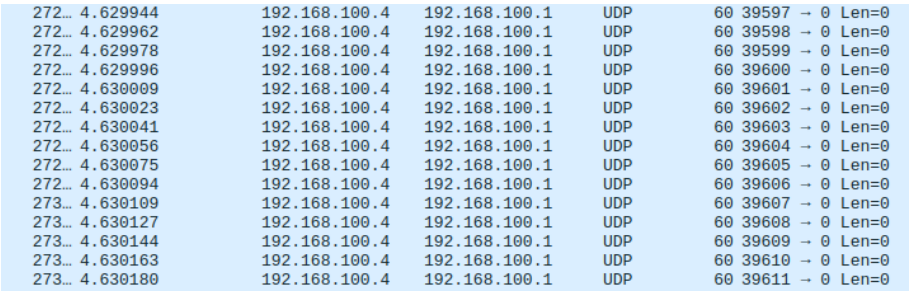
- Limitación de la velocidad de las respuestas ICMP por unidad de tiempo: esta limitación de las respuestas ICMP suele llevarse a cabo en el nivel del sistema operativo.
- Filtrado en el nivel del cortafuegos en el servidor: esto permite descartar paquetes sospechosos. Sin embargo, el cortafuegos también puede colapsarse bajo el volumen de datos de la inundación UDP.
- Filtrado de paquetes UDP, excepto en DNS, a nivel de red: las consultas DNS suelen ejecutarse mediante UDP. Con esta medida, cualquier otra fuente que genere una cantidad masiva de tráfico UDP se considerará sospechosa y los paquetes en cuestión se descartarán.



(a) Configuración del atacante



(b) Ping de la víctima a la IP cambiada



(c) Paquetes enviados a la víctima

Figura 8: Demostración de Fraggle

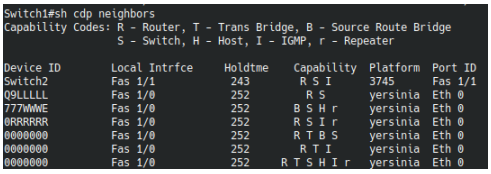
9. CDP Flooding:

Descripción: El ataque se basa en la inundación de la tabla CDP del switch conectado al atacante (Switch1) con mensajes CDP, que haciendo uso de otros ataques, es posible denegar el servicio de la víctima. El primer objetivo (el otro ataque) es eliminar las entradas de los otros switches para, posteriormente, inundar la tabla CDP y evitar que puedan llegar solicitudes del switch de la víctima (Switch2). Si se mantiene activo este ataque de forma prolongada, puede afectar al funcionamiento del dispositivo ya que puede dañar componentes de este al estar expuesto a una sobrecarga de los recursos del equipo.

Puesta en práctica: Haciendo uso del programa Yersinia y su función de ataque CDP (flooding CDP table) comenzaremos a inundar la tabla con paquetes, los cuales son vecinos falsos creados aleatoriamente y enviados directamente al Switch1. La tabla se llenará y ,al no existir entradas para los otros switches de la topología (previamente fueron eliminadas), impedirán al Switch1 conocer a los otros switches.

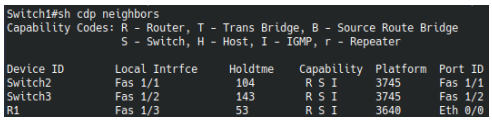
Contramedidas: Configurar los switches y demás dispositivos para que reenvíen un mensaje CDP antes de que expire la entrada de la tabla del switch inutiliza la denegación de servicio con el ataque individual. Otra opción es desactivar el protocolo CDP en una interfaz en específico (las interfaces que no pertenezcan a switches), ya que no es imprescindible para el correcto funcionamiento de la subred. Se desactivaría ejecutando el siguiente comando en el switch deseado:

```
Switch1(config-if)# no cdp enable
```



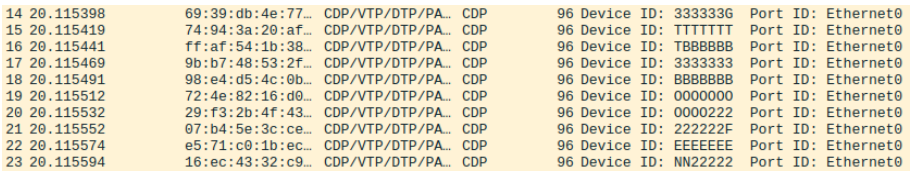
```
Switch1#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Switch2          Fas 1/1           243         R S I      3745      Fas 1/1
Q9LLLLL          Fas 1/0           252           R S      yersinia  Eth 0
777WWWE          Fas 1/0           252        B S H r     yersinia  Eth 0
0RRRRRR          Fas 1/0           252        R S I r     yersinia  Eth 0
0000000          Fas 1/0           252        R T B S     yersinia  Eth 0
0000000          Fas 1/0           252         R T I      yersinia  Eth 0
0000000          Fas 1/0           252     R T S H I r    yersinia  Eth 0
```

(a) Tabla CDP del switch antes del ataque



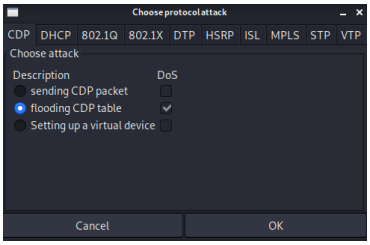
```
Switch1#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Switch2          Fas 1/1           104         R S I      3745      Fas 1/1
Switch3          Fas 1/2           143         R S I      3745      Fas 1/2
R1               Fas 1/3           53          R S I      3640      Eth 0/0
```

(b) Tabla CDP del switch después del ataque



```
14 20.115398        69:39:db:4e:77… CDP/VTP/DTP/PA… CDP      96 Device ID: 333333G  Port ID: Ethernet0
15 20.115419        74:94:3a:20:af… CDP/VTP/DTP/PA… CDP      96 Device ID: TTTTTTT  Port ID: Ethernet0
16 20.115441        ff:af:54:1b:38… CDP/VTP/DTP/PA… CDP      96 Device ID: TBBBBBB  Port ID: Ethernet0
17 20.115469        9b:b7:48:53:2f… CDP/VTP/DTP/PA… CDP      96 Device ID: 3333333  Port ID: Ethernet0
18 20.115491        98:e4:d5:4c:0b… CDP/VTP/DTP/PA… CDP      96 Device ID: BBBBBBB  Port ID: Ethernet0
19 20.115512        72:4e:82:16:d0… CDP/VTP/DTP/PA… CDP      96 Device ID: OOOOOOO  Port ID: Ethernet0
20 20.115532        29:f3:2b:4f:43… CDP/VTP/DTP/PA… CDP      96 Device ID: OOOO222  Port ID: Ethernet0
21 20.115552        07:b4:5e:3c:ce… CDP/VTP/DTP/PA… CDP      96 Device ID: 222222F  Port ID: Ethernet0
22 20.115574        e5:71:c0:1b:ec… CDP/VTP/DTP/PA… CDP      96 Device ID: EEEEEEE  Port ID: Ethernet0
23 20.115594        16:ec:43:32:c9… CDP/VTP/DTP/PA… CDP      96 Device ID: NN22222  Port ID: Ethernet0
```

(c) Paquetes enciados al switch víctima



(d) Inicio del ataque

Figura 9: Demostración de CDP Flooding

10. ICMP Redirection:

Este ataque lo contemplamos de forma teórica.

Descripción: Este ataque se basa en el uso del mensaje de ICMP tipo 5, el ICMP Redirect, con la finalidad de modificar la tabla de ruta de una máquina víctima para redirigirla hacia otros destinos en beneficio del atacante, pudiendo ser estos destinos una subred o a un host en particular, pudiendo conseguir realizar un ataque Man In The Middle gracias a ICMP. Al realizar este tipo de ataques podemos conseguir un ataque bidireccional implementando las distintas tecnologías que hemos visto previamente.

Puesta en práctica: A traves del comando:

```
> icmp_redirect -v -s ipVictima/mask -G ipAtacante -i interface

    -s ipVictima/mask nos indica la red fuente y su máscara de red
    -G ipAtacante nos indica el Gateway que recibe los paquetes
       que se envíen desde la red víctima hacia la máquina atacante
    -i nos indica la interfaz de escucha en la máquina atacante
```

Una vez ejecutado este comando se ya estaremos realizando el Man In the Middle, por lo que todos los paquetes pasarán por la máquina atacante.

Contramedidas: Para protegernos contra este ataque es posible configurar el equipo de forma local para evitar que los mensajes icmp redirect tengan efecto. Para ello debemos editar el fichero /etc/sysctl.conf añadiendo la línea:

```
net.inet.icmp.drop_redirect=1 (se puede hacer directamente desde
terminal con el comando sudo sysctl -w net.inet.icmp.drop_redirect=1).
```

Cuando nos llegue un mensaje icmp redirect con valor 1 se aceptará y cuando sea 0 se descartará el mensaje. La mayoría de distribuciones actuales vienen con el valor en 0 por lo que estarán protegidas por defecto.

11. SSLsplit:

Este ataque lo contemplamos de forma teórica.

Descripción: Es una herramienta para ataques **MITM (man-in-the-middle)** en conexiones de red cifradas en **SSL/TLS**. Las conexiones se interceptan de forma transparente a través de un motor de traducción de dirección de red redirigido a SSLsplit. La herramienta termina SSL/TLS e inicia una nueva conexión (SSL/TLS) con la dirección original de destino, mientras se registran todos los datos transmitidos. SSLsplit está destinado al análisis forense de red y pruebas de penetración.

SSLsplit trabaja con los protocolo **TCP, SSL, HTTP y HTTPS,** a través de **IPv4 e IPv6** .
Para las conexiones SSL y HTTPS, genera certificados X509v3 sobre la marcha, con base en el certificado de servidor original Subject DN. SSLsplit es capaz de trabajar con RSA , DSA y las claves ECDSA, DHE y Suites de cifrado ECDHE. SSLsplit también pueden usar certificados existentes del que se disponga la clave privada, en lugar de generarlo. También soporta certificados NC NULL -prefix y puede denegar las solicitudes de OCSP de forma genérica.

Puesta en práctica:

1. Utilizando openssl generaremos la clave privada para el certificado de la siguiente forma:

```
> openssl genrsa -out ca.key
(-out: salida como archivo ca.key)
```

2. Utilizamos la firma de clave privada para generar el certificado:

```
> openssl req -new -x509 -days 1096 -key ca.key -out ca.crt \\
    -req: solicitud;
    -nuevo: nuevo;
    -x509: formato;
    -días: período válido;
    -clave especifica la clave privada;
    -out: certificado raíz
```

3. Activamos el enrutamiento y reenvío en la máquina atacante:

```
> sysctl - w net.ipv4.ip_forward=1
```

4. Establecemos las reglas de reenvío de puertos de iptables:

```
> iptables -t nat -F
> iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080 [#HTTP]
> iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8443 [#HTTPS]
> iptables -t nat -A PREROUTING -p tcp --dport 587 -j REDIRECT --to-ports 8443 [#MSA]
> iptables -t nat -A PREROUTING -p tcp --dport 465 -j REDIRECT --to-ports 8443 [#SMTPS]
> iptables -t nat -A PREROUTING -p tcp --dport 993 -j REDIRECT --to-ports 8443 [#IMAPS]
> iptables -t nat -A PREROUTING -p tcp --dport 995 -j REDIRECT --to-ports 8443 [#POP3S]
```

```
    -t: especificar la tabla;
    -A PREROUTING: entrar en vigor antes del enrutamiento;
    -p: especificar el protocolo;
    -dport: aceptar el puerto del tráfico;
    -j: especificar el método de procesamiento (redirección REDIRECT);
    -to-ports: puertos de reenvío
```

Con el comando:

```
> iptables -t nat -L
```

podemos ver todas las reglas recién configuradas.

5. Con Arp spoofing interceptamos el tráfico de red del host de destino:

```
> arpspoof -i eth0 -t IpDestino -r PuertaEnlaceHostDestino
    -i: especificar la tarjeta de red;
    -t: falsificar el destino;
    -r: dirección de la puerta de enlace
```

6. Iniciamos SSLsplit para el ataque: Primero creamos un directorio (p. e.):

```
> mkdir -p test/logdir
```

Escuchamos en el puerto del set:

```
> sslsplit -D -l connect.log -j /root/test -S /root/test/logdir/
  -k ca.key -c ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080
```

7. Resultados:
Cuando la víctima acceda a un sitio web https, habrá un error con el certificado de seguridad. Si la víctima hace click en Aceptar, el enlace se secuestra con éxito y el atacante puede obtener la información transmitida. Cuando el cliente accede al atacante, sslsplit recopila y registra la información real del certificado del sitio real en un instante y falsifica el certificado que se puede ver en el navegador para que el usuario lo vea. La información de conexión se puede ver en connect.log. Los datos de transmisión se pueden ver en el directorio que creamos en primer lugar en el paso 6: test/ logdir (usando el comando grep podemos filtrar datos).

Contramedidas: Para protegernos contra este tipo de ataques al SSL tenemos varias posibilidades que vamos a ver a continuación:

- A la hora de estar visitando sitios web una opción segura es el uso de VPN's, una herramienta que encripta nuestra conexión extremo a extremo, desde antes incluso que accedamos al sitio web. Esto impedirá que cualquier atacante reciba nuestros datos sin encriptar.
- Si somos nosotros quienes gestionamos una web, una buena opción es contar con un certificado SSL que abarque la totalidad de la web y no solo su página de acceso. Gracias a esto conseguimos que no existan enlaces sin encriptar.
- Evitar las redes poco seguras, como pueden ser las redes Wi-Fi de cafeterías, bibliotecas, aeropuertos... Estas redes son espacios donde resulta mucho más sencillo realizar ataques MITM como SSLsplit, SSLstrip...