VNiVERSiDAD D SALAMANCA
CAMPUS DE EXCELENCIA INTERNACIONAL

DPTO. INFORMÁTICA Y AUTOMÁTICA
Plaza de los Caídos, s/n. 37008 Salamanca
Tel.: +34 923 29 46 53 Fax:+34 923 29 45 14
diaweb.usal.es sec_dia@usal.es

MEMORIA FINAL DE EJECUCIÓN

Código del proyecto: ID2017/029

Diseño de materiales prácticos para la asignatura Seguridad en Sistemas Informáticos del grado en Ingeniera Informática

Responsable: Ángeles Mª Moreno Montero

Miembros del equipo de trabajo:

Francisco Javier Blanco Rodríguez

Belén Curto Diego

Vidal Moreno Rodilla

María José Polo Martín

Salamanca, 12 de julio de 2018

Contenido

1. Introducción

Este documento constituye el informe final de ejecución del proyecto de Innovación Docente realizado durante el curso académico 2017-2018 en el Departamento de Informática de la Facultad de Ciencias, financiado mediante la convocatoria de 2017 de Ayudas de la Universidad de Salamanca para los Proyecto de Innovación y Mejora Docente.

Si buscamos un aspecto esencial dentro de la rama de las redes de computadores encontraremos el de la seguridad de las mismas. En sus inicios, los protocolos que permiten la conexión entre nodos de una red fueron creados para desarrollar una serie de funcionalidades, y no para ser seguros. Sin embargo, el número de nodos que componen Internet (la red de redes) ha experimentado un enorme crecimiento en los últimos años, lo que ha favorecido la aparición de atacantes que buscan debilidades que les permitan obtener información sensible. Esto, como es de esperar, es un riesgo y amenaza constante para las empresas, al igual que para cualquier usuario de a pie. Pero las amenazas no solo se generan desde fuera de la empresa, o fuera de la red local, como ampliamente se cree.

Es por todo esto, por lo que el conocimiento de ataques, vulnerabilidades y soluciones a ellas deben serle familiares al estudiante de Ingeniería Informática, ya que la información cada vez es más sensible y es necesario poner todos los esfuerzos para protegerla a través de la red.

En el ámbito del Grado en Ingeniería Informática, la materia Redes está formada por: Señales y Sistemas, Redes de Computadores I, Redes de Computadores II y Seguridad en Sistemas Informáticos. Estas asignaturas constan de partes teóricas y de partes prácticas, donde se pretende que el alumno aplique lo aprendido en la teoría a escenarios cercanos a la realidad.

En el caso de la asignatura de Seguridad en Sistemas Informáticos, la seguridad en redes es solo un apartado de la asignatura, debido al hecho de que se aborda la seguridad desde un punto de vista general y no es posible profundizar en este aspecto ni practicar con distintas configuraciones de red. Es importante destacar que la parte práctica de la asignatura de Seguridad en Sistemas Informáticos en el grado de Ingeniería Informática de la Universidad de Salamanca se realiza en el Laboratorio de Informática de la Facultad de Ciencias, cuya configuración no puede ser alterada por el alumno.

Por todo lo expuesto anteriormente, las aplicaciones de simulación resultan un aliado para el estudiante, ya que ofrecen la posibilidad de crear topologías de red completamente simuladas sin limitaciones físicas ni económicas.

Existen varias herramientas de simulación que permiten una experiencia real y cercana a entornos que pueden darse en cualquier organización. Estas herramientas simulan los

elementos de red (switches, routers, etc.) como elementos de Cisco Systems, uno de los fabricantes más importantes en esta área. Este aspecto constituye un beneficio para el alumno, que podrá experimentar cómo se configuran equipos Cisco ampliamente utilizados en el ámbito empresarial.

Además, en el curso académico 2011/2012, resultado de la financiación conseguida en el proyecto: "Elementos de Interconexión de Redes: Integrando Voz y Datos" (PL11/023) del programa de Innovación Docente USAL 2011, fueron adquiridos equipos utilizados para crear un prototipo de red que integra voz y datos. Estos equipos son de gama media-alta, siendo similares a los utilizados en medianas y grandes empresas.

La disponibilidad de estos equipos reales va a dotarnos de otro entorno sobre el que aplicar medidas de seguridad, permitiendo realizar una demostración de un laboratorio centrado en la seguridad sobre una red avanzada.

Así, este proyecto de innovación se basa en la idea de elaborar material didáctico para la que el estudiante pueda profundizar en el ámbito de la seguridad en redes mediante guiones teóricos y prácticos, utilizando software de simulación, además de aplicarse todos los conocimientos adquiridos durante la realización de los materiales didácticos sobre un laboratorio real.

El resto del documento se organiza como sigue. En primer lugar, se enumeran los objetivos del proyecto. A continuación, se examina el proceso de diseño del material docente que se desea generar. En el apartado cuarto se describen brevemente todos los materiales elaborados, tanto los materiales para entornos simulados, como para nuestro laboratorio de redes con equipos reales. En el apartado cinco, a modo de ejemplo, se incluye el material correspondiente a los ataques sobre HTTP/HTTPS. En el sexto se muestran los equipos adquiridos, sus principales características y se justifica su adquisición para finalizar con las principales conclusiones de este proyecto.

2. Objetivos

El objetivo principal de este proyecto es la elaboración de materiales didácticos que permitan a los estudiantes de Ingeniería Informática adquirir competencias en el ámbito de la seguridad en redes. Como se ha comentado en la introducción, la seguridad en redes se aborda de forma general en la asignatura de cuarto curso Seguridad en Sistemas Informáticos y la parte práctica de la asignatura se realiza en el Laboratorio de Informática de la Facultad de Ciencias, cuya configuración no puede ser alterada por el alumno. En este sentido, las aplicaciones de simulación son una buena opción, ya que ofrecen la posibilidad de crear topologías de red completamente simuladas sin limitaciones físicas ni económicas.

Otro objetivo planteado es trabajar con equipos reales y no solo con entornos simulados. Para ello, se dispone de un prototipo de laboratorio avanzado de redes formado por equipos CISCO de gama media-alta muy similares a los utilizados en entorno de medianas y grandes empresas. Este prototipo ha sido ampliado gracias a la financiación recibida en este proyecto con equipos más modernos que implementan medidas de seguridad no disponibles en los anteriores.

3. Diseño del material docente

Para conseguir los objetivos planteados ha sido necesario realizar una adecuada selección de contenidos y la elaboración de los materiales que permita al alumno realizar las prácticas tanto en entornos simulados como reales.

En la preparación de los guiones prácticos se han tenido en cuenta dos puntos de vista muy importantes para la comprensión de los contenidos escogidos:

1. Ataques a la red, donde se pondrán en práctica los ataques más relevantes que pueden realizarse en una red, no solo mediante el uso de herramientas de ataque, sino también a partir de la comprensión de las mismas y de los aspectos de los protocolos que permiten el éxito de estos ataques. Es muy importante conocer los ataques para saber defenderse frente a ellos.
2. Contramedidas para los ataques, donde se comprenderán y configurarán medidas tendentes a evitar o mitigar los efectos producidos por los ataques puestos en práctica anteriormente.

A continuación, se describe el proceso de diseño del material elaborado. El primer paso es definir qué contenidos se van a abordar en las prácticas y cómo clasificarlos de forma lógica. Una vez realizado este paso se han elaborado los guiones, los cuales cuentan con dos versiones, una con las soluciones del escenario y otra para el alumno, donde se marcarán los objetivos, los pasos, las pruebas y evaluaciones necesarias para comprobar que la solución al escenario propuesta por el alumno es correcta y que se han comprendido los conceptos teóricos.

Los contenidos se organizarán según los niveles del modelo de referencia OSI, conociendo las vulnerabilidades que afectan al nivel que se esté tratando. Esto permitirá un aprendizaje incremental, ya que a medida que se incrementa el nivel, los ataques parecen más elaborados, pero utilizan estrategias vistas en niveles inferiores.

En cada uno de los niveles se seleccionarán los ataques más conocidos y se escogerá el mejor entorno para realizarlo, ya que cada uno cuenta con ciertas limitaciones. El uso de varios entornos (simulaciones, emulaciones y equipos reales) permitirá abarcar las principales medidas de seguridad ofrecidas por la empresa Cisco.

Estos conjuntos de materiales irán abordando vulnerabilidades capa a capa, siguiendo el modelo OSI. El último grupo de materiales se trata de una demostración, explicada paso a paso, de la configuración de un laboratorio de redes que permita poner en práctica, en un entorno totalmente real, todos los conceptos adquiridos durante las simulaciones realizadas a lo largo de los laboratorios.

Los materiales han sido realizados siguiendo los mismos pasos y estructura que se puede observar en la Tabla 1. Se realiza un primer apartado teórico que introduzca las vulnerabilidades a tratar y permita entender el por qué y cómo pueden explotarse de forma malintencionada, así como sus medidas para protegernos frente a ellas.

El siguiente apartado, si procede con el tema tratado, da a conocer las herramientas que permiten explotar las vulnerabilidades tratadas anteriormente, y que se utilizarán para realizar los laboratorios. El tercer y último apartado será meramente práctico, dividido en dos guiones; el guion dirigido al alumno y el guion con la solución al escenario presentado. Los guiones constarán de una parte en la que se realizarán los ataques explicados en puntos anteriores y otra parte en la que se configurarán los elementos de red de forma que se pueda frenar la explotación de las vulnerabilidades presentadas, para así conocer de forma más cercana tanto los ataques como la forma de mitigarlos. Se busca con esto que el alumno encuentre y realice, de forma guiada, la mejor configuración para el escenario abordado en cada punto.

1. Selección y clasificación de contenidos según la vulnerabilidad de la capa que explote
2. Definición de objetivos del apartado
3. Elaboración de los apartados teóricos
4. Elaboración de figuras que ayuden a comprender los conceptos teóricos
5. Elección de la herramienta (Cisco Packet Tracer o GNS3)
6. Elección de herramientas de ataque para explotar vulnerabilidades
7. Guion práctico para el alumno
   - 7.1. Elaboración de escenarios preconfigurados y prediseñados.
   - 7.2. Conjunto de pasos a realizar.
   - 7.3. Elaboración de preguntas de evaluación.
8. Guion con la solución óptima
   - 8.1. Elaboración de los escenarios y configuraciones que den solución a la práctica en cuestión.
   - 8.2. Sugerencias para la evaluación de la práctica.
9. Enlaces para la ampliación de los temas tratados.

Tabla 1: Pasos seguidos para la elaboración de los materiales didácticos.

4. Resultados: Materiales elaborados

Los materiales generados agrupados en niveles se resumen en la siguiente tabla.

Nombre ATAQUE	Descripción	Herramien-tas de ataque	Herramien-tas de defensa	Defensas con equipos CISCO no disponibles
Nivel de enlace				
Inundación (*flooding*)	Se envía una gran cantidad de tramas Ethernet falseadas para que la tabla MAC del switch se sobrecargue y así deja de mostrar su funcionamiento normal	GNS3, utilidad *macof* de Kali Linux	Packet tracer	
Suplantación MAC (MAC *spoofing)*	Se utiliza la dirección física de otro elemento (o una dirección física falseada) con objeto de suplantarlo	GNS3, utilidades *arpspoof y ettercap* de Kali Linux	GNS3, *arpwatch* Debian-8 Linux	Dynamic ARP Inspection (DAI), IP Source Guard
El uso de VLANs para proteger la arquitectura de red local	Una incorrecta configuración de VLAN puede generar vulnerabilidades y ser por tanto la fuente de ataques	Cisco Packet Tracer	Cisco Packet Tracer	
Nivel de red				
Ataques sobre IPv4: - Ataque smurf, - ICMP redirect (spoofing) -Ataques de fragmentación	Buscan atacar las rutas de encaminamiento, hay vulnerabilidades que permiten la denegación de servicios o traspasar reglas de un firewall o listas de control de acceso	GNS3, utilidades hping3 icmpush y scapy de kali linux	Linux y Cisco IOS	
Ataques sobre IPv6: -Ataque sobre el protocolo Neighbor	IPv6 se diseñó para ser mucho más seguro que su predecesor IPv4 pero el desconocimiento	GNS3, herramient a IPv6 Attack	Herramient a ndpmon en Linux Ubuntu	Protocolo SeND (Secure Neighbor Discovery),

Nombre ATAQUE	Descripción	Herramien-tas de ataque	Herramien-tas de defensa	Defensas con equipos CISCO no disponibles
Discovery (neighbor spoofing) - Ataque contra la detección de direcciones IPv6 duplicadas (Router Advertisement spoofing) - Ataques de inundación (Router Advertisement flooding) -Advertencia de seguridad en redes en las que conviven IPv4 e IPv6.	de este relativamente nuevo protocolo puede provocar situaciones de vulnerabilidad	Toolkit de Kali Linux		IPv6 Snooping, IPv6 Source Guard, RA Guard
El uso de VPNs para proteger comunicaciones remotas	Las redes privadas virtuales conectan wide area networks (WAN) de forma segura, de tal forma que no se compromete la privacidad e integridad de los datos	GNS3, CISCO IOS		
Nivel de transporte				
Ataques sobre UDP: Ping-pong Attack,	Buscan, mediante inundaciones o sobrecargas, realizar una denegación de servicios sobre los objetivos del atacante	GNS3, herramient a hping3 de Kali Liunux	CISCO IOS:	Cisco IOS ofrece funcionalid ades que pueden filtrar el tráfico recibido que se encuentran también en

Nombre ATAQUE	Descripción	Herramien-tas de ataque	Herramien-tas de defensa	Defensas con equipos CISCO no disponibles
				sus firewalls ASA
Ataques sobre TCP: - SYN Flood - Secuestro de sesión	flooding o inundación es una de las técnicas más utilizadas a la hora de denegar el servicio también en TCP.	GNS3, herramient a *metasploit* de Kali Linux	CISCO IOS tcp-intercept	
Nivel de aplicación				
Ataques sobre HTTP/HTTPs: SSL Stripping	Hombre en medio. La víctima cree tener un dialogo seguro con el legítimo servidor, pero el atacante está en medio de ambos interceptando su comunicación.	GNS3, herramient a SSLStrip de Kali Linux	HSTS	
Ataques sobre DHCP: *Rogue* DHCP	La necesidad de los servidores DHCP hace que estos servidores sean un blanco para ataques de spoofing y denegación de servicios	Cisco Packet Tracer	CISCO IOS DHCP Snooping	
Ataques sobre DNS: -DNS ID Spoofing - DNS Cache Poisoning - DNS Flooding (DoS).	Otro servicio esencial y por tanto vulnerable	GNS3, herramient a dnsspoof de Kali Linux		DNSSEC
VoIP: -Denegación de servicio - escuchas, spoofing - fraudes	El hecho de que las señales de voz viajen a través de Internet puede generar problemas de seguridad	Packet Tracer	CISCO IOS, VLANs: Separando lógicament e voz y datos, encriptació n: SRTP y SDP, VPNs: Protegiendo	

Nombre ATAQUE	Descripción	Herramien-tas de ataque	Herramien-tas de defensa	Defensas con equipos CISCO no disponibles
			- Llamadas remotas - ACLs: Restringiendo accesos a/desde determinados segmentos - Protecciones contra fraude	

Tabla 2: Materiales elaborados agrupados por niveles.

En cuanto a los materiales para el aseguramiento de una red real (dos router Cisco 861, un router Cisco 1861 y tres teléfonos IP Cisco 6911 – Ver Figura 1) se han descrito todos los pasos para que el alumno pueda trabajar con las tres configuraciones de dificultad creciente que se detallan en la Tabla 3. Para trabajar con estos equipos reales el alumno ha de aplicar las técnicas aprendidas en las prácticas realizadas previamente en los entornos de simulación.

Figura 1: Trabajando con equipos reales.

Configuración básica	Esta configuración tiene como objetivo dotar a los equipos de un mismo edificio (esto es, los equipos conectados a cada router Cisco) de conectividad, así como aplicar buenas prácticas en la configuración de VLANs y asegurar los propios dispositivos de red de accesos indeseados (router Cisco 1861 y routers Cisco 861).
Configuración media	La configuración media aporta un estrato de seguridad mayor que el visto en la configuración básica. Los equipos de distintos edificios no pueden comunicarse aún entre sí, lo cual no es funcional, por lo que se habilitará el protocolo de encaminamiento RIP. Todos los protocolos de encaminamiento son susceptibles de recibir ataques, pudiendo un router o equipo malicioso indicar rutas falsas en sus mensajes. Para evitar que esto suceda debemos autenticar los mensajes del protocolo RIPv2. En este apartado también se deshabilitarán servicios no utilizados susceptibles de recibir ataques y se añadirán listas de acceso que limiten las conexiones entre distintos departamentos o segmentos de la red. Por último, se protegerá el servidor de la empresa de ataques SYN Flood y de falsificación de servidores DHCP (DHCP spoofing).
Configuración avanzada	Nuevas funcionalidades a la configuración media, para así permitir a nuestra maqueta el acceso a un escenario en GNS3. Para que los equipos conectados a la maqueta puedan comunicarse con el exterior debemos habilitar la traducción de direcciones o NAT. Debemos tener en cuenta en este paso el tipo de NAT utilizado, pues no queremos que los equipos finales de la empresa sean accesibles desde redes externas. Sin embargo, nuestra red cuenta con un servidor, el cual sí debe poder ser accesible por usuarios externos. También debemos habilitar OSPF para que nuestra maqueta obtenga dinámicamente las direcciones del escenario en GNS3. Por último, se simularán sedes remotas de la misma empresa que deben poder comunicarse de forma segura. Para ello, implementaremos una VPN (*Virtual Private Network*) compatible con la configuración NAT realizada anteriormente.

Tabla 3: Configuraciones de seguridad sobre los equipos CISCO.

A modo de ejemplo detallaremos en el siguiente apartado los materiales para *Ataques sobre HTTP/HTTPS*.

5. Caso de Estudio: Ataques sobre HTTP/HTTPS

Este ataque es parte del bloque de materiales del nivel de aplicación que busca introducir los ataques más dañinos de los principales y más utilizados servicios de internet: HTTP/HTTPS, DHCP y DNS. Al pertenecer a la capa superior, estos ataques se basan en conceptos vistos anteriormente, como son los ataques *Man In The Middle*. Es

importante conocer en qué vulnerabilidades se basan estos ataques para tener una base de conocimiento a la hora de defendernos.

Las aplicaciones web utilizan el protocolo HTTP (RFC2616) para recibir peticiones y responder a las mismas. Este protocolo no cifra los datos enviados a través de la red, lo que permitiría a un atacante leer los datos intercambiados entre usuario y servidor web mediante técnicas de "hombre en el medio", llegando a robar contraseñas simplemente leyendo el flujo de datos.

Es por esto por lo que la mayoría de servicios web utilizan el protocolo HTTPS (RFC2818) para intercambiar datos sensibles como pueden ser las credenciales de un usuario. Este protocolo cifra los datos intercambiados, haciendo prácticamente imposible a un atacante el leerlos. Sin embargo, veremos que el uso de HTTPS no es suficiente para evitar ataques de *sniffing*.

5.1. Conceptos teóricos

En este apartado se va a resumir el funcionamiento de la técnica de *hacking* denominada SSL *Stripping* (RFC7457), la cual fue introducida por primera vez por Moxie Marlinspike.

Una de las técnicas más utilizadas por servidores web para asegurar sus conexiones es la redirección de http a HTTPS, de esta forma un usuario inexperto que teclea el nombre de un sitio web sin comenzar por "https//:", por ejemplo www.example.com, será redirigido a https://www.example.com, obteniendo los certificados del servidor y cifrándose desde ese instante los datos intercambiados (Figura 2). Sin embargo, el propio funcionamiento incurre en una gran vulnerabilidad.

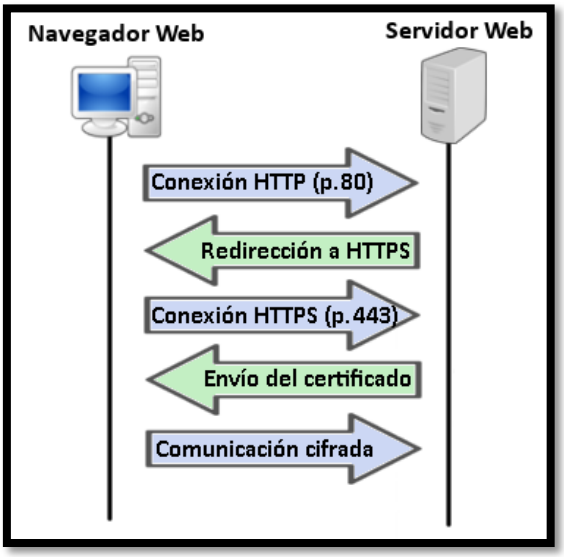


Figura 2: Redirección desde HTTP hacia HTTPS.

Un atacante puede aprovechar esa primera conexión mediante HTTP para interceptar la petición, y negociar él mismo el intercambio de certificados para empezar una conexión segura con el servidor.

Esto segmenta la comunicación, siendo la conexión entre víctima y atacante mediante HTTP y la conexión entre atacante y servidor mediante HTTPS. El atacante actúa como un puente, mandando los datos que le envía la víctima en texto plano (vía HTTP) al servidor (vía HTTPS).

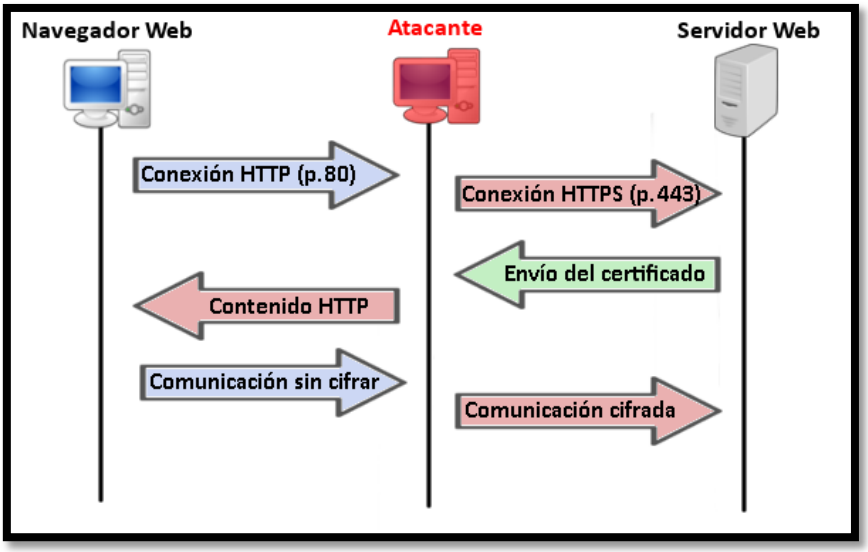


Figura 3. Ataque SSLStrip.

Para llevar a cabo este ataque es necesario encontrarse en la misma red que la víctima y haber realizado un ataque *Man In The Middle*. A su vez, este ataque solo es posible si la víctima comienza la conexión mediante HTTP, dado que si se comienza mediante HTTPS no habrá forma de que el atacante pueda interponerse, pues el cliente esperaría un certificado válido y el atacante no puede falsificar el certificado del servidor ni descifrar los datos intercambiados una vez iniciada la conexión, al estar asociado con una clave privada propiedad del servidor.

Si una víctima entra en un sitio web mediante un link que enlaza el sitio mediante HTTPS, el ataque tampoco surtirá efecto.

5.2. Herramienta de estudio: *SSLStrip*

SSLStrip es una herramienta escrita en Python que permite llevar acabo el ataque descrito anteriormente. Su funcionamiento es muy sencillo:

1. Todo el tráfico HTTP (puerto de destino 80) se redirige al puerto tras el que SSLStrip se encuentra funcionando.
2. SSLStrip se encarga de negociar con el servidor todos los parámetros HTTPS, haciéndole creer a éste que se trata de un usuario más de sus servicios.

3. Cuando se ha establecido la conexión segura entre atacante y servidor y el servidor responde a la primera petición, el atacante se encarga de responder mediante HTTP a la víctima con la información recibida.
4. El atacante se encarga de realizar el resto de peticiones de la víctima al servidor mediante HTTPS y de responder a la víctima mediante HTTP.

Un aspecto muy importante de esta herramienta es que, cuando recibe las páginas solicitadas al servidor, se encarga de renombrar todos los links que tiene esa página, convirtiendo los enlaces HTTPS en HTTP, de tal forma que el ataque siga teniendo efecto aun cuando la víctima navegue por medio de enlaces.

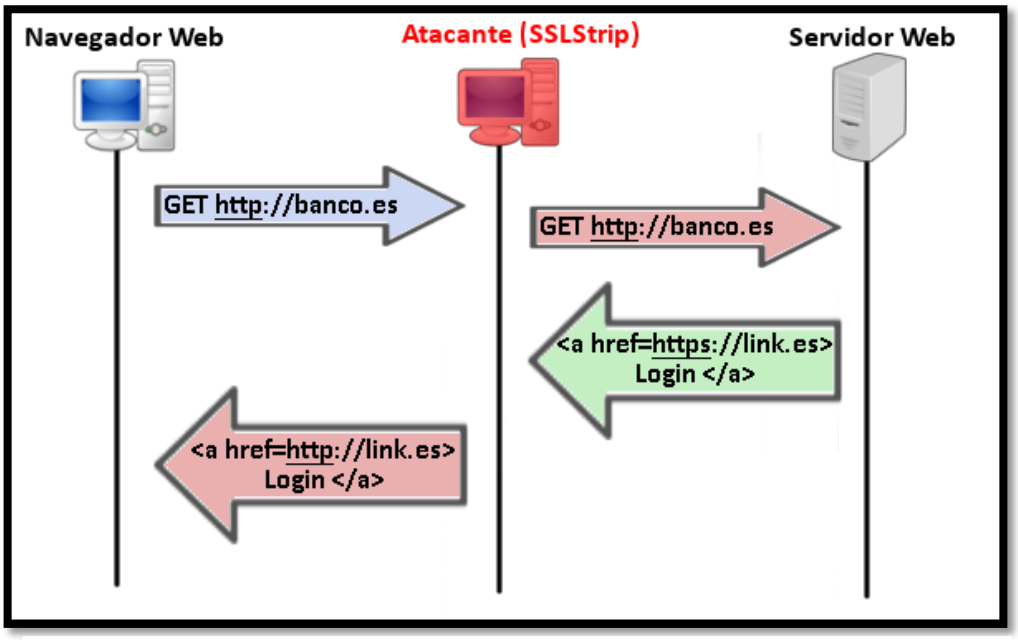


Figura 4. Renombramiento de enlaces por parte de SSLStrip.

Es necesario redirigir todo el tráfico cuyo puerto de destino sea el puerto 80 (HTTP) hacia el puerto tras el que se encuentra *SSLStrip* mediante reglas de firewall. Con *iptables* se realizaría de la siguiente forma:

iptables –t nat –A PREROUTING –p tcp --destination-port 80 –j REDIRECT --to-port <puerto>

Para lanzar SSLStrip basta con abrir una terminal y ejecutarlo mediante ***sslstrip –l <puerto>***, siendo "puerto" el puerto tras el que se ejecutará SSLStrip.

Para que el ataque funcione correctamente es necesario que el equipo atacante tenga activado el *forwarding* de paquetes y que se haya realizado con éxito un ataque *MitM*, de tal forma que el router por defecto de la red crea que el equipo atacante es, en verdad, el equipo víctima, consiguiendo así acceso a todo el tráfico proveniente del servidor web externo.

5.3. Guion del alumno

En esta práctica se realizará un ataque SSLStrip y se comprenderá HSTS, una forma de mitigar dicho ataque.

5.3.1. Ataque

Utilizando el escenario denominado *09-Application-SSLStrip.gns3* donde la dirección IP del router y el servicio DHCP se encuentran configurados, debes arrastrar un equipo ***Víctima***, ***Atacante*** y ***ServidorWeb*** y colocarlos en los huecos reservados para ellos, de tal forma que se obtenga un escenario similar al mostrado en la Figura 5.

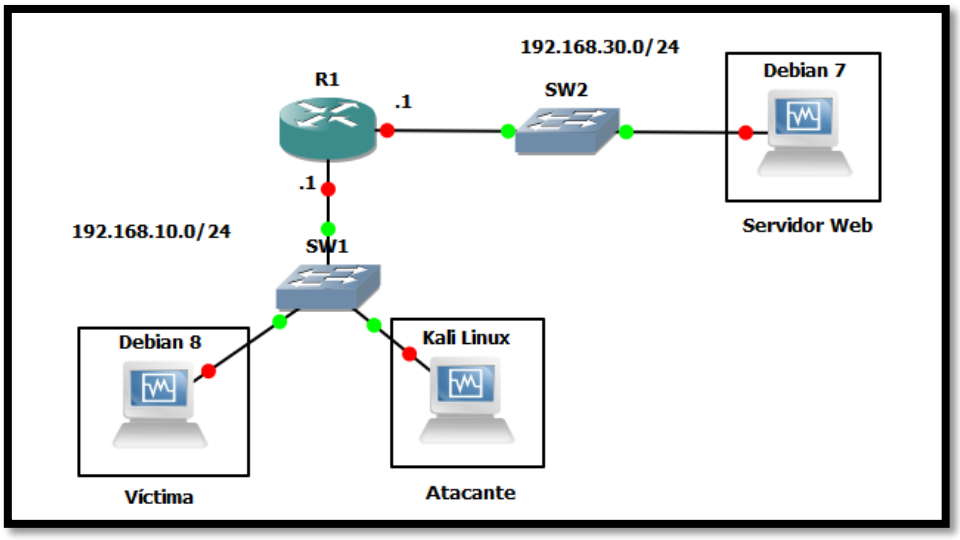


Figura 5: Escenario para ataques HTTPS.

Debes arrastrar al escenario una máquina virtual que contenga un servidor web con acceso mediante HTTPS y redirección hacia HTTPS si se intenta acceder mediante HTTP.

Arranca los equipos uno a uno, comenzando por el router R1 y acabando por los equipos finales. Comienza una captura en el segmento que une al equipo víctima con el switch SW1. Abre el navegador del equipo víctima y accede al servidor web mediante HTTP tecleando su dirección IP en la barra de navegación, ¿qué sucede?

¿Qué se observa en la captura de tráfico?

Nota: *ten en cuenta que el certificado del servidor web será autofirmado, y, en consecuencia, el navegador no lo tomará como seguro. Debes añadir una excepción. Sin embargo, la conexión HTTPS será completamente válida.*

A continuación, ejecuta el ataque SSLStrip desde el equipo ***Atacante***. Recuerda que debes realizar primero un ataque *Man In The Middle* y que debes activar el *forwarding* y redirigir el tráfico mediante *iptables* hacia el puerto de escucha de SSLStrip.

Comienza otra captura en el segmento que une el servidor web con el switch SW2. Vuelve a entrar en el servidor web desde el equipo víctima mediante HTTP. ¿Qué sucede en la captura que une el servidor web con SW2? ¿Y en la captura que une a la víctima con el SW1? ¿Quién negocia la conexión segura con el servidor web? Explica por qué los pasos realizados anteriormente en el equipo ***Atacante*** han conseguido este comportamiento. ¿Qué sucede si en vez de entrar al servidor web mediante HTTP, la víctima entra directamente mediante HTTPS? ¿Por qué?

¿Cómo compromete la seguridad este ataque?

5.3.2. Defensa

Como se ha podido comprobar, el origen de este ataque es el hecho de conectarse mediante HTTP por primera vez al servidor. Los usuarios no se preocupan en escribir las direcciones web completas (comenzando mediante "https://"), con lo cual, aunque posteriormente el servidor redirija las conexiones vía HTTPS, la conexión sigue estando en riesgo.

Es por esta vulnerabilidad por la que surge HSTS (RFC6797), que fuerza al propio navegador (no al usuario) a que, de forma interna, cambie una petición que en principio iba a ser realizada mediante HTTP (debido a que el usuario haya tecleado el nombre del sitio web sin utilizar "https://") hacia una petición HTTPS. Gracias a esto, todas las peticiones se realizarán de primeras mediante HTTPS.

Esto es posible gracias a una cabecera HTTPS denominada *Strict-Transport-Security*. Cuando un navegador recibe una respuesta de un sitio web con dicha cabecera, la almacena durante el tiempo que se especifique en la propia política *Strict-Transport-Security* (atributo *max-age*) y, a partir de ese momento, el navegador fuerza que todas las conexiones con dicho servicio web se hagan mediante HTTPS (aunque el usuario no lo especifique en la barra de dirección explícitamente). Así pues, a modo de ejemplo:

1. El usuario teclea **http://www.example.com**.
2. El navegador comprueba si dicho sitio web se encuentra bajo la política *Strict-Transport-Security*, y su campo, *max-age*, no ha expirado.
3. Si el sitio web se encuentra bajo la política, el navegador transforma la petición HTTP en una petición HTTPS, de tal forma que cuando la petición salga del ordenador del usuario, ésta será **https://www.example.com**.

Sin embargo, para recibir esa cabecera, debemos acceder al menos una vez al sitio web en cuestión. Ese primer acceso es vulnerable, ya que el servidor aún no conoce si tiene

que forzar la conexión hacia HTTPS, y, por tanto, podría realizarse mediante HTTP, siendo la conexión igualmente susceptible de ser atacada por SSLStrip.

Debido a esto, existen en los principales navegadores listas precargadas (*preloaded lists*), que contienen información sobre sitios web que implementan HSTS y hacia los que hay que forzar las conexiones HTTPS. Esto permite al navegador conocer a priori qué conexiones debe establecer mediante HTTPS de forma forzada, aun no habiendo visitado nunca dicho sitio (y, por tanto, aun no habiendo recibido la cabecera *Strict-Transport-Security*).

Visto esto, hay que tener siempre en cuenta que, **la primera conexión a un sitio web (que implemente HSTS) que no se encuentre en las listas precargadas y al que no se haya accedido anteriormente, es susceptible de ser atacada por las técnicas vistas anteriormente.**

Problemas con los certificados autofirmados

Esta medida no funciona correctamente si un servidor ofrece su servicio HTTPS por medio de un certificado autofirmado. Para que HSTS cumpla su labor, es necesario que el cliente instale el certificado del servidor para evitar excepciones de seguridad y falsificaciones de certificados.

Para configurar HSTS en Apache2 debes seguir dos pasos:

1. Habilitar el módulo *headers*.
2. Modificar el fichero *default-ssl* de *sites-available* en la carpeta de configuración de Apache2 para que incluya la siguiente línea:
    a. ***Header always set Strict-Transport-Security "max-age=VALIDEZ; includeSubDomains"***
    b. Donde VALIDEZ es el tiempo en segundos que se fuerza el tráfico mediante HTTPS. Este valor puede variar según la actividad del usuario, pero lo más seguro es que al menos sea de un año.

Debido a que en nuestro escenario no contamos con un certificado válido, el navegador utilizado hará caso omiso a la cabecera "*Strict-Transport-Security*", siendo, pues, el ataque aún exitoso.

Para comprender, entonces, cómo funciona HSTS estudiaremos páginas reales que actualmente lo implementan, como es el caso de *Facebook*. Lo primero que debes hacer es acceder al apartado de herramientas de desarrolladores de tu navegador y acceder al apartado *Network*. Este apartado permite ver con detalle todas las peticiones que se realizan al acceder a un sitio web.

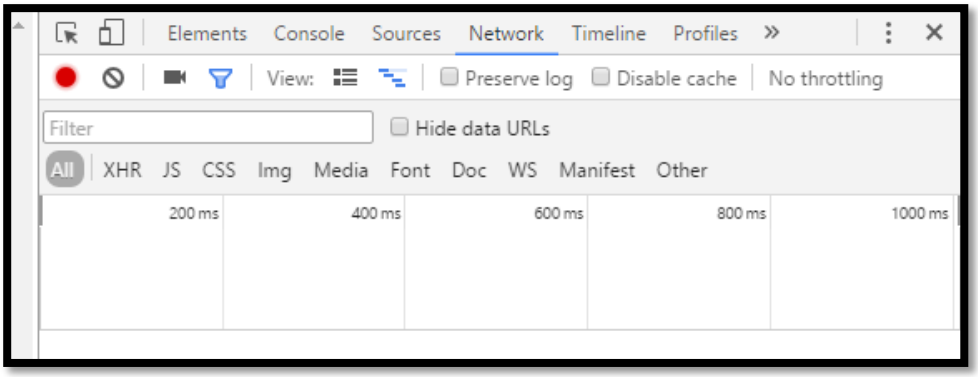


Figura 6. Inspector de red de las herramientas para desarrolladores en Google Chrome.

Una vez abierto el inspector de tu navegador, accede a *Facebook* mediante HTTP (es decir, tecleando únicamente *www.facebook.com* o bien *http://www.facebook.com*) y presta atención a la primera petición que se genera, accediendo a sus detalles clicando en ella. ¿Qué *status* tiene la petición? ¿Cuál es la URL que se pide (*request url*)? ¿Cuál es la respuesta (*response headers*)? Busca información sobre dicho estado y piensa cómo HSTS puede estar involucrado.

Ahora presta atención a la segunda petición, ¿qué URL se está pidiendo en este caso? ¿Existe algún *Response Header* llamativo? ¿Qué se está indicando con dicha cabecera? ¿Cuál es el *status* de la petición?

¿En algún momento han sido nuestras peticiones susceptibles de un ataque mediante SSLStrip? ¿Por qué?

Busca información sobre las listas precargadas HSTS en los navegadores e investiga si Facebook está entre ellas. ¿Podría alguien robar nuestras credenciales de Facebook mediante SSLStrip?

Nota: *actualmente existe SSLStrip2, que busca romper la seguridad que ofrece HSTS. Basa su funcionamiento en el hecho de que HSTS hace uso de los nombres de dominio en sus listas precargadas, con lo cual, se puede utilizar un servidor DNS malicioso que altere los nombres de dominio para que el navegador no los encuentre en sus listas, debido a que realmente no existen.*

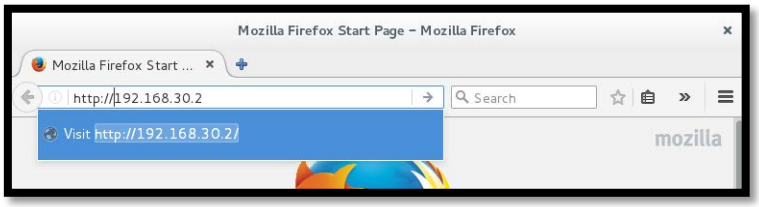
Más información: ***https://github.com/LeonardoNve/sslstrip2***

5.4. Guion con las soluciones

5.4.1. Ataque

Comienza una captura en el segmento que une al equipo víctima con el switch SW1. Abre el navegador del equipo víctima y accede al servidor web mediante HTTP tecleando su dirección IP en la barra de navegación, ¿qué sucede?

Especificamos en la barra de navegación la dirección del servidor Web y escribimos explícitamente que la conexión será mediante HTTP.



Sin embargo, al acceder al sitio web, observamos que se nos redirige hacia una conexión segura mediante HTTPS. Se nos advierte de que el certificado no está firmado por ninguna autoridad de certificación conocida por el navegador, aunque sabemos que el certificado es totalmente válido, así que lo aceptamos.

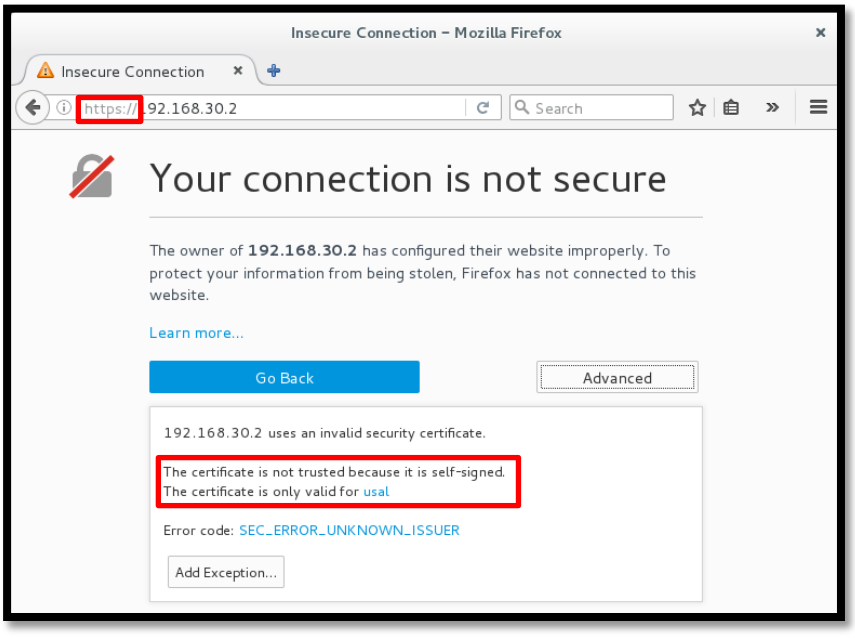
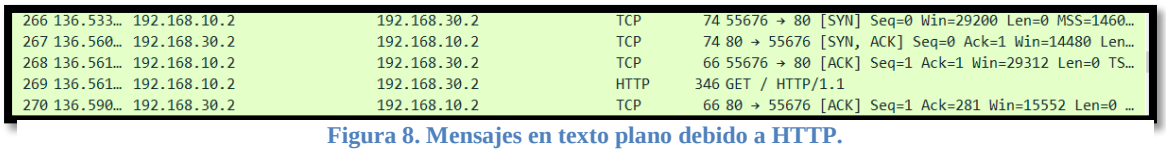


Figura 7. Advertencia de conexión no segura debido a los certificados autofirmados.

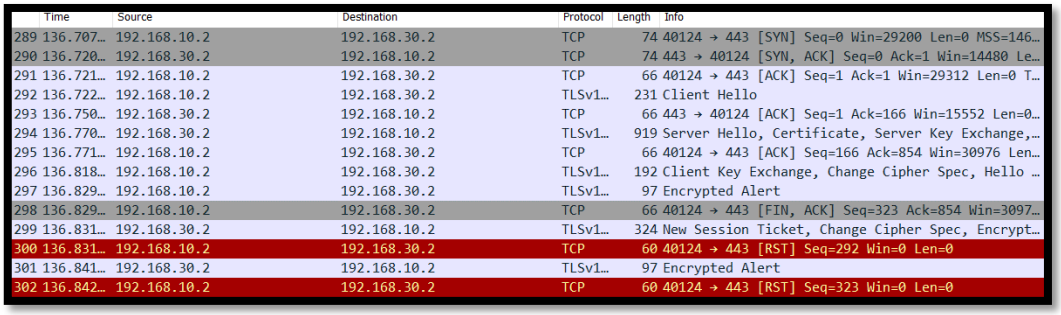
¿Qué se observa en la captura de tráfico?

1) Al realizar el equipo víctima la petición mediante HTTP, observamos mensajes TCP dirigidos al puerto 80 para iniciar la conexión. Todos estos mensajes viajan en texto plano entre la víctima (192.168.10.2) y el servidor web (192.168.30.2).
2) El servidor web, sin embargo, fuerza una redirección hacia HTTPS, con lo que



266	136.533…	192.168.10.2	192.168.30.2	TCP	74 55676 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460…
267	136.560…	192.168.30.2	192.168.10.2	TCP	74 80 → 55676 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len…
268	136.561…	192.168.10.2	192.168.30.2	TCP	66 55676 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TS…
269	136.561…	192.168.10.2	192.168.30.2	HTTP	346 GET / HTTP/1.1
270	136.590…	192.168.30.2	192.168.10.2	TCP	66 80 → 55676 [ACK] Seq=1 Ack=281 Win=15552 Len=0 …

Figura 8. Mensajes en texto plano debido a HTTP.

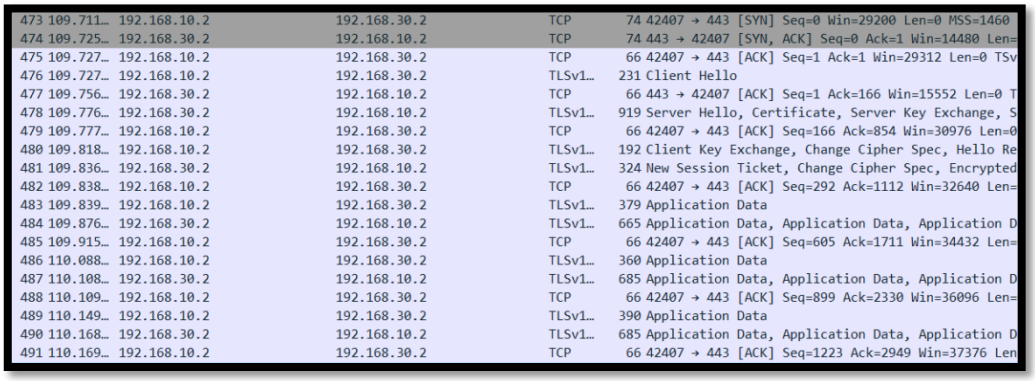


	Time	Source	Destination	Protocol	Length	Info
289	136.707…	192.168.10.2	192.168.30.2	TCP	74	40124 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=146…
290	136.720…	192.168.30.2	192.168.10.2	TCP	74	443 → 40124 [SYN, ACK] Seq=0 Ack=1 Win=14480 Le…
291	136.721…	192.168.10.2	192.168.30.2	TCP	66	40124 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 T…
292	136.722…	192.168.10.2	192.168.30.2	TLSv1…	231	Client Hello
293	136.750…	192.168.30.2	192.168.10.2	TCP	66	443 → 40124 [ACK] Seq=1 Ack=166 Win=15552 Len=0…
294	136.770…	192.168.30.2	192.168.10.2	TLSv1…	919	Server Hello, Certificate, Server Key Exchange,…
295	136.771…	192.168.10.2	192.168.30.2	TCP	66	40124 → 443 [ACK] Seq=166 Ack=854 Win=30976 Len…
296	136.818…	192.168.10.2	192.168.30.2	TLSv1…	192	Client Key Exchange, Change Cipher Spec, Hello …
297	136.829…	192.168.10.2	192.168.30.2	TLSv1…	97	Encrypted Alert
298	136.829…	192.168.10.2	192.168.30.2	TCP	66	40124 → 443 [FIN, ACK] Seq=323 Ack=854 Win=3097…
299	136.831…	192.168.30.2	192.168.10.2	TLSv1…	324	New Session Ticket, Change Cipher Spec, Encrypt…
300	136.831…	192.168.10.2	192.168.30.2	TCP	60	40124 → 443 [RST] Seq=292 Win=0 Len=0
301	136.841…	192.168.30.2	192.168.10.2	TLSv1…	97	Encrypted Alert
302	136.842…	192.168.10.2	192.168.30.2	TCP	60	40124 → 443 [RST] Seq=323 Win=0 Len=0

Figura 9. Redirección a HTTPS.

comienza la negociación de parámetros del protocolo entre el servidor web y el cliente. Esta negociación se ve interrumpida (mensajes TCP RST) debido a que el cliente no es capaz de validar el certificado recibido (momento en el que aparece la advertencia en el navegador). Se observa en la captura que no se ha transmitido ningún dato de la aplicación.

3) Cuando aceptamos desde el navegador el certificado y añadimos la excepción, es cuando realmente nos comunicamos con el servidor mediante HTTPS. Como vemos en la siguiente captura, sí que se transmiten datos de la aplicación. Estos datos están cifrados gracias al intercambio de claves anterior.



473	109.711…	192.168.10.2	192.168.30.2	TCP	74 42407 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
474	109.725…	192.168.30.2	192.168.10.2	TCP	74 443 → 42407 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=
475	109.727…	192.168.10.2	192.168.30.2	TCP	66 42407 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSv
476	109.727…	192.168.10.2	192.168.30.2	TLSv1…	231 Client Hello
477	109.756…	192.168.30.2	192.168.10.2	TCP	66 443 → 42407 [ACK] Seq=1 Ack=166 Win=15552 Len=0 T
478	109.776…	192.168.30.2	192.168.10.2	TLSv1…	919 Server Hello, Certificate, Server Key Exchange, S
479	109.777…	192.168.10.2	192.168.30.2	TCP	66 42407 → 443 [ACK] Seq=166 Ack=854 Win=30976 Len=0
480	109.818…	192.168.10.2	192.168.30.2	TLSv1…	192 Client Key Exchange, Change Cipher Spec, Hello Re
481	109.836…	192.168.30.2	192.168.10.2	TLSv1…	324 New Session Ticket, Change Cipher Spec, Encrypted
482	109.838…	192.168.10.2	192.168.30.2	TCP	66 42407 → 443 [ACK] Seq=292 Ack=1112 Win=32640 Len=
483	109.839…	192.168.10.2	192.168.30.2	TLSv1…	379 Application Data
484	109.876…	192.168.30.2	192.168.10.2	TLSv1…	665 Application Data, Application Data, Application D
485	109.915…	192.168.10.2	192.168.30.2	TCP	66 42407 → 443 [ACK] Seq=605 Ack=1711 Win=34432 Len=
486	110.088…	192.168.10.2	192.168.30.2	TLSv1…	360 Application Data
487	110.108…	192.168.30.2	192.168.10.2	TLSv1…	685 Application Data, Application Data, Application D
488	110.109…	192.168.10.2	192.168.30.2	TCP	66 42407 → 443 [ACK] Seq=899 Ack=2330 Win=36096 Len=
489	110.149…	192.168.10.2	192.168.30.2	TLSv1…	390 Application Data
490	110.168…	192.168.30.2	192.168.10.2	TLSv1…	685 Application Data, Application Data, Application D
491	110.169…	192.168.10.2	192.168.30.2	TCP	66 42407 → 443 [ACK] Seq=1223 Ack=2949 Win=37376 Len

Figura 10. Conexión cifrada gracias a HTTPS.

En este instante el navegador ya nos muestra que la conexión es segura.

A continuación, ejecuta el ataque SSLStrip desde el equipo *Atacante*. Recuerda que debes realizar primero un ataque Man In The Middle y que debes activar el *forwarding* y redirigir el tráfico mediante iptables hacia el puerto de escucha de SSLStrip.

Primero activamos el *forwarding* mediante *"echo 1 > /proc/sys/net/ipv4/ip_forward"* y posteriormente iniciamos *arpspoof* (ver punto 4.2.2 para más información) para hacerle creer al router (192.168.10.1) que nuestro equipo es la víctima (192.168.10.2).

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# arpspoof -i eth0 -t 192.168.10.2 192.168.10.1
8:0:27:77:a1:17 8:0:27:2b:c1:23 0806 42: arp reply 192.168.10.1 is-at 8:0:27:77:
a1:17
8:0:27:77:a1:17 8:0:27:2b:c1:23 0806 42: arp reply 192.168.10.1 is-at 8:0:27:77:
a1:17
8:0:27:77:a1:17 8:0:27:2b:c1:23 0806 42: arp reply 192.168.10.1 is-at 8:0:27:77:
```

Figura 11. Activación del *Forwarding* y ataque *MitM* en el equipo atacante.

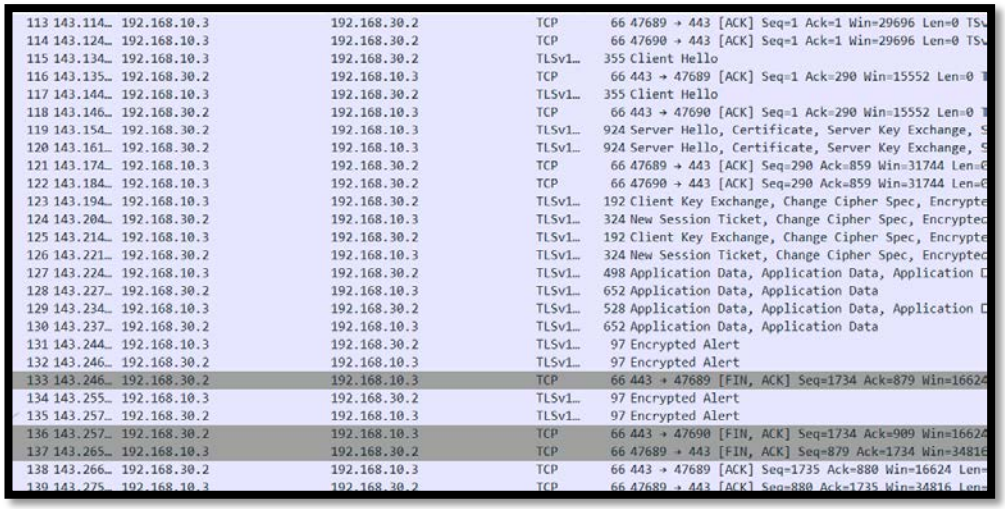
En otra terminal ejecutamos la orden de *iptables* vista anteriormente para redirigir el tráfico HTTP al puerto en el que estará escuchando SSLStrip (en este caso el 10000), para finalmente ejecutar SSLStrip.

```
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIR
ECT --to-port 10000
root@kali:~# sslstrip -l 10000
```

Figura 12. Redirección de conexiones HTTP al puerto de *SSLStrip* y puesta en marcha.

Comienza otra captura en el segmento que une el servidor web con el switch SW2. Vuelve a entrar en el servidor web desde el equipo víctima mediante HTTP. ¿Qué sucede en la captura que une el servidor web con SW2?

Realizando los mismos pasos para acceder al sitio web que en las cuestiones anteriores, obtenemos el siguiente resultado en la captura que une el servidor web con el switch SW2:



113	143.114…	192.168.10.3	192.168.30.2	TCP	66 47689 → 443 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TS…
114	143.124…	192.168.10.3	192.168.30.2	TCP	66 47690 → 443 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TS…
115	143.134…	192.168.10.3	192.168.30.2	TLSv1…	355 Client Hello
116	143.135…	192.168.30.2	192.168.10.3	TCP	66 443 → 47689 [ACK] Seq=1 Ack=290 Win=15552 Len=0 …
117	143.144…	192.168.10.3	192.168.30.2	TLSv1…	355 Client Hello
118	143.146…	192.168.30.2	192.168.10.3	TCP	66 443 → 47690 [ACK] Seq=1 Ack=290 Win=15552 Len=0 …
119	143.154…	192.168.30.2	192.168.10.3	TLSv1…	924 Server Hello, Certificate, Server Key Exchange, …
120	143.161…	192.168.30.2	192.168.10.3	TLSv1…	924 Server Hello, Certificate, Server Key Exchange, …
121	143.174…	192.168.10.3	192.168.30.2	TCP	66 47689 → 443 [ACK] Seq=290 Ack=859 Win=31744 Len=0
122	143.184…	192.168.10.3	192.168.30.2	TCP	66 47690 → 443 [ACK] Seq=290 Ack=859 Win=31744 Len=0
123	143.194…	192.168.10.3	192.168.30.2	TLSv1…	192 Client Key Exchange, Change Cipher Spec, Encrypte…
124	143.204…	192.168.30.2	192.168.10.3	TLSv1…	324 New Session Ticket, Change Cipher Spec, Encrypted…
125	143.214…	192.168.10.3	192.168.30.2	TLSv1…	192 Client Key Exchange, Change Cipher Spec, Encrypte…
126	143.221…	192.168.30.2	192.168.10.3	TLSv1…	324 New Session Ticket, Change Cipher Spec, Encrypted…
127	143.224…	192.168.10.3	192.168.30.2	TLSv1…	498 Application Data, Application Data, Application D…
128	143.227…	192.168.30.2	192.168.10.3	TLSv1…	652 Application Data, Application Data
129	143.234…	192.168.10.3	192.168.30.2	TLSv1…	528 Application Data, Application Data, Application D…
130	143.237…	192.168.30.2	192.168.10.3	TLSv1…	652 Application Data, Application Data
131	143.244…	192.168.10.3	192.168.30.2	TLSv1…	97 Encrypted Alert
132	143.246…	192.168.30.2	192.168.10.3	TLSv1…	97 Encrypted Alert
133	143.246…	192.168.30.2	192.168.10.3	TCP	66 443 → 47689 [FIN, ACK] Seq=1734 Ack=879 Win=16624…
134	143.255…	192.168.10.3	192.168.30.2	TLSv1…	97 Encrypted Alert
135	143.257…	192.168.30.2	192.168.10.3	TLSv1…	97 Encrypted Alert
136	143.257…	192.168.30.2	192.168.10.3	TCP	66 443 → 47690 [FIN, ACK] Seq=1734 Ack=909 Win=16624…
137	143.265…	192.168.10.3	192.168.30.2	TCP	66 47689 → 443 [FIN, ACK] Seq=879 Ack=1734 Win=34816…
138	143.266…	192.168.30.2	192.168.10.3	TCP	66 443 → 47689 [ACK] Seq=1735 Ack=880 Win=16624 Len=…
139	143.275…	192.168.10.3	192.168.30.2	TCP	66 47689 → 443 [ACK] Seq=880 Ack=1735 Win=34816 Len=…

Figura 13. Captura en el segmento que une al servidor web con su switch local.

Vemos que se realiza una conexión mediante HTTPS desde el primer momento, aunque el equipo víctima haya realizado la petición mediante HTTP. Esto es debido a SSLStrip, lo cual queda demostrado al revisar quiénes están negociando la conexión: 192.168.10.3 (el *Atacante*) y 192.168.30.2 (el servidor). Así pues, el *Atacante* está suplantando a la víctima en lo que a la conexión HTTPS se refiere.

¿Y en la captura que une a la víctima con el SW1? Se muestra una conexión HTTP sin interrupciones, debido a que SSLStrip está actuando como intermediario, tomando los datos que recibe desde el servidor mediante HTTPS y transformándolos en respuestas HTTP. Estos datos viajan en texto plano.

2…	972.218…	192.168.30.2	192.168.10.2	TCP	74 80 → 55683 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len…
2…	972.219…	192.168.10.2	192.168.30.2	TCP	66 55683 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TS…
2…	972.220…	192.168.10.2	192.168.30.2	HTTP	346 GET / HTTP/1.1
2…	972.220…	192.168.30.2	192.168.10.2	TCP	66 80 → 55683 [ACK] Seq=1 Ack=281 Win=30720 Len=0 …
2…	972.282…	192.168.30.2	192.168.10.2	HTTP	616 HTTP/1.1 302 Found (text/html)
2…	972.283…	192.168.10.2	192.168.30.2	TCP	66 55683 → 80 [ACK] Seq=281 Ack=551 Win=30336 Len=…
2…	972.388…	192.168.10.2	192.168.30.2	HTTP	346 GET / HTTP/1.1
2…	972.388…	192.168.30.2	192.168.10.2	TCP	66 80 → 55683 [ACK] Seq=551 Ack=561 Win=31744 Len=…
2…	972.569…	192.168.30.2	192.168.10.2	HTTP	556 HTTP/1.1 200 OK (text/html)
2…	972.570…	192.168.10.2	192.168.30.2	TCP	66 55683 → 80 [ACK] Seq=561 Ack=1041 Win=31488 Len…
2…	972.798…	192.168.10.2	192.168.30.2	HTTP	327 GET /favicon.ico HTTP/1.1
2…	972.799…	192.168.30.2	192.168.10.2	TCP	66 80 → 55683 [ACK] Seq=1041 Ack=822 Win=32768 Len…
2…	972.877…	192.168.30.2	192.168.10.2	HTTP	638 HTTP/1.1 302 Found (text/html)
2…	972.877…	192.168.10.2	192.168.30.2	TCP	66 55683 → 80 [ACK] Seq=822 Ack=1613 Win=32640 Len…
2…	972.880…	192.168.10.2	192.168.30.2	HTTP	327 GET /favicon.ico HTTP/1.1
2…	972.881…	192.168.30.2	192.168.10.2	TCP	66 80 → 55683 [ACK] Seq=1613 Ack=1083 Win=33792 Le…
2…	972.881…	192.168.10.2	192.168.30.2	TCP	74 55684 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460…
2…	972.882…	192.168.30.2	192.168.10.2	TCP	74 80 → 55684 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len…
2…	972.883…	192.168.10.2	192.168.30.2	TCP	66 55684 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TS…
2…	972.883…	192.168.10.2	192.168.30.2	HTTP	357 GET /favicon.ico HTTP/1.1

Figura 14. Captura en el segmento que une a la víctima con su switch local (Conexión en texto plano).

Vemos en el propio navegador de la víctima que la conexión ya no es HTTPS.

¿Quién negocia la conexión segura con el servidor web? Explica por qué los pasos realizados anteriormente en el equipo *Atacante* han conseguido este comportamiento.

Como se ha visto en la captura anterior, la negocia el equipo *Atacante*. Esto se ha conseguido gracias a los siguientes pasos:

1. **El uso de *arpspoof* y la habilitación del *forwarding* han conseguido que el equipo *Atacante* reciba todo el tráfico que la víctima envíe con dirección al router. El equipo puede modificar los paquetes que le llegan de la víctima y reenviarlos como si de un router se tratase.**
2. **La redirección hacia el puerto 10000 de paquetes dirigidos al puerto 80 consigue que todas las peticiones que realiza la víctima mediante HTTP se intercepten y se envíen a SSLStrip.**
3. **SSLStrip utiliza los datos de los paquetes HTTP interceptados para negociar una conexión HTTPS consistente con el servidor, y a su vez responder a la víctima mediante HTTP con los datos que le van llegando.**

¿Qué sucede si en vez de entrar al servidor web mediante HTTP, la víctima entra directamente mediante HTTPS? **Aun encontrándose el ataque en ejecución, la víctima entra sin problemas mediante HTTPS, haciendo imposible al *Atacante* descifrar la información que se intercambian el servidor y el cliente.**

¿Por qué? **Porque al iniciarse la conexión mediante HTTPS, el cliente queda a la espera de un certificado válido. Dicho certificado es imposible de falsificar porque está firmado con una clave privada que el *Atacante* no posee. A su vez, si el *Atacante* intentase enviar un certificado con la clave pública del servidor creado por el mismo tampoco sería capaz de descifrar el tráfico, pues necesita la clave privada asociada a la pública.**

¿Cómo compromete la seguridad este ataque? **Si la web que se accede necesita que sean introducidos datos personales, cuentas bancarias, contraseñas, etc., el *Atacante* podría sin ningún problema robar estos datos, pues viajarían en texto plano.**

5.4.2. Defensa

Una vez abierto el inspector de tu navegador, accede a *Facebook* mediante HTTP (es decir, tecleando únicamente *www.facebook.com* o bien *http://www.facebook.com*) y **presta atención a la primera petición que se genera, accediendo a sus detalles clicando en ella, ¿qué *status* tiene la petición? ¿Cuál es la URL que se pide (*request url*)? ¿Cuál es la respuesta (*response headers*)? Busca información sobre el estado obtenido y piensa cómo HSTS puede estar involucrado.**

La primera petición, como se ve en la captura, está dirigida hacia *http://www.facebook.com*, lo que genera un código 307 *Internal Redirect*. Este estado se obtiene cuando el propio navegador detecta que una petición debe ser redirigida incluso antes de realizarla.

El propio campo *Response Headers* nos da la respuesta; se nos devuelve una localización (*https://www.facebook.com*), la cual es aquella a la que se nos quiere redirigir. También hay un campo que indica la razón de esta redirección, y esta razón es HSTS. Es decir, el propio navegador ha interceptado nuestra petición insegura mediante HTTP, y, sabiendo que *Facebook* utiliza HSTS, ha redirigido nuestra petición hacia *https://www.facebook.com*, haciendo que nuestra primera petición hacia *Facebook* sea mediante HTTPS, aunque no se haya especificado de forma explícita.

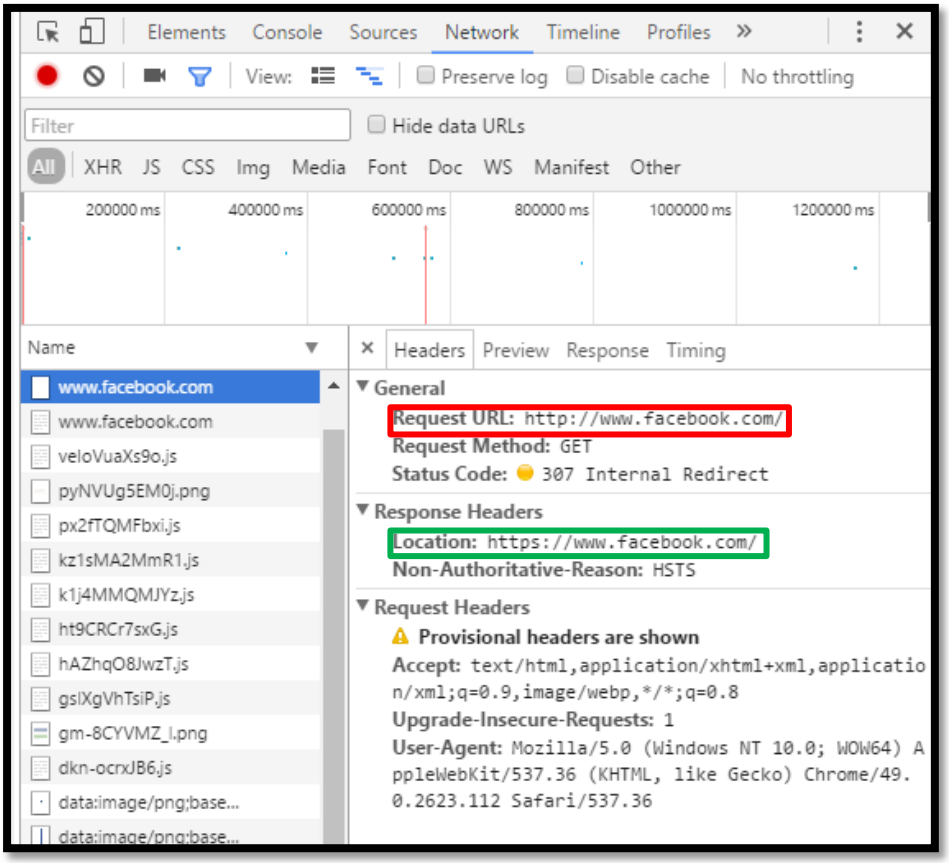


Figura 15. Primera petición realizada a facebook.com.

Ahora presta atención a la segunda petición, ¿qué URL se está pidiendo en este caso? ¿Existe algún *Response Header* llamativo? ¿Qué se está indicando con dicha cabecera? ¿Cuál es el *status* de la petición?

En este caso se pide directamente la URL *https://www.facebook.com*. Dado que el objeto de estudio es HSTS, sí que podemos observar una cabecera interesante: *strict-transport-security*. Como se ha indicado anteriormente, esta cabecera indica al navegador que debe forzar todas las conexiones a este sitio para que se realicen mediante HTTPS. También observamos que se indica el campo *max-age 15552000*, lo que quiere decir que el navegador forzará conexiones seguras a esta página durante los siguientes 15552000 segundos.

El estado 200 OK de esta petición nos indica que la petición se ha realizado satisfactoriamente y que hemos obtenido la información deseada (además, de forma segura).

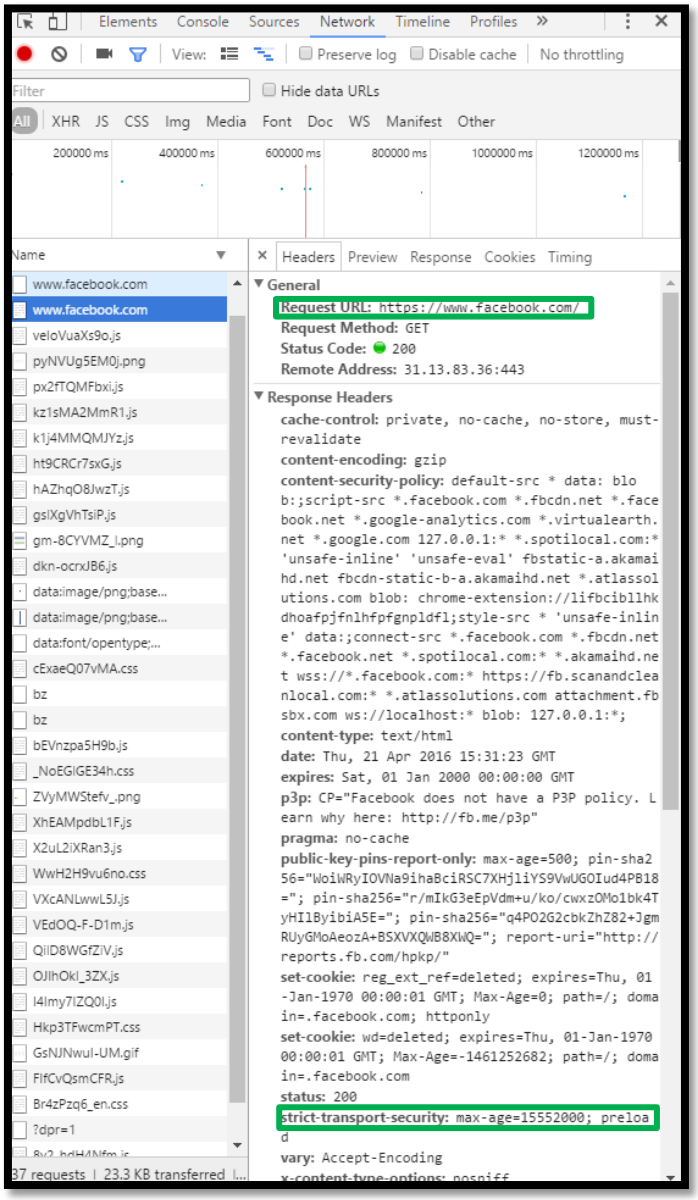


Figura 16. Segunda petición realizada a facebook.com (segura).

¿En algún momento han sido nuestras peticiones susceptibles de un ataque mediante SSLStrip? ¿Por qué?

No, porque el navegador ha redirigido de forma interna nuestra petición antes de realizarla. Dicha redirección ha conseguido que la primera petición efectiva (es decir, que se dirige a Internet) se haga mediante HTTPS, desbaratando el intento de SSLStrip de aprovecharse de las peticiones HTTP.

Busca información sobre las listas precargadas HSTS en los navegadores e investiga si Facebook está entre ellas. Las listas precargadas consiguen que no se realicen conexiones mediante HTTP a los sitios web que se encuentran en ellas, aunque no hayamos recibido nunca la cabecera *Strict-Transport-Security* de dichos sitios web (es

decir, aunque no los hayamos visitado nunca). El navegador forzará siempre el uso de HTTPS en las conexiones con los sitios web de las listas precargadas, protegiendo incluso la primera petición.

Facebook sí se encuentra en dichas listas. En el navegador *Google Chrome* podemos hacer consultas sobre qué dominios se encuentran en el apartado *chrome://net-internals/#hsts*. Al realizar una consulta sobre *Facebook* el resultado es positivo:

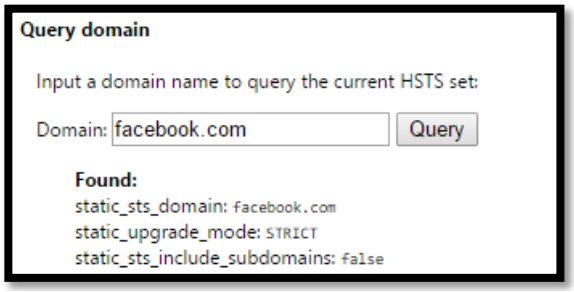


Figura 17. Facebook encontrado en las listas precargadas HSTS.

¿Podría alguien robar nuestras credenciales de Facebook mediante SSLStrip? No. Como se ha visto al principio de este guion, la debilidad de HSTS está en que la primera petición no está protegida si no se ha visitado el sitio anteriormente y, en consecuencia, recibido la cabecera *Strict-Transport-Security*. Sin embargo, *Facebook* se encuentra en las listas precargadas de los navegadores, con lo cual, aunque una persona no haya visitado *Facebook* todavía con cierto navegador, la primera petición (y las siguientes) se realizarán vía HTTPS. Todo esto sucede gracias a que el navegador busca primero si el dominio solicitado se encuentra en las listas precargadas y, si es así, redirige toda petición insegura hacia una segura.

6. Equipos adquiridos

En la Tabla 4 se muestran los equipos adquiridos con la financiación obtenida para este proyecto junto con sus principales características y las razones que justifican su adquisición.

 Figura 18: Cisco ASA 5506-X with FirePOWER Services.
Firewall adaptativo y orientado a amenazas de la industria diseñado para una nueva era de amenazas y protección avanzada contra malware. Ofrece una defensa de amenazas integrada para todo el ataque, antes, durante y después de un ataque. No disponíamos de ningún dispositivo de red del tipo Cortafuegos/Firewall.
 Figura 19: Cisco Catalyst 2960L-16TS-LL.
Conmutador Gigabit Ethernet que opera con el software Cisco IOS® y admite la gestión simple de dispositivos y la gestión de red. Conmutador administrado que ofrece funciones avanzadas de nivel 2. Esta serie ofrece seguridad de red mejorada, confiabilidad de la red y eficiencia operativa. Este conmutador pertenece a la serie *classic 2960* uno de los equipos más robustos de la marca CISCO. Ha sido elegido para este proyecto debido a sus características avanzadas de seguridad de las que no disponía ninguno de los equipos de nuestro laboratorio de redes.
 Figura 20: Cisco Small Business SG350-10P
Los switches de esta serie proporcionan características de seguridad avanzadas, así como funcionalidades de nivel 2 y 3. El hecho de proporcionar facilidades de nivel 3 es la razón por la que ha sido elegido para este proyecto a diferencia con el conmutador de la serie *classic* 2969 que solo tiene facilidades de nivel 2.

Tabla 4: Equipos adquiridos con la financiación de este proyecto.

7. Conclusiones

Con este proyecto se han elaborado materiales didácticos para poner en práctica ataques comunes en redes de computadores, así como razonar sus consecuencias y conocer formas de protegernos frente a ellos.

Para poder obtener estos materiales, se ha realizado un estudio de los diversos programas de simulación y emulación de redes disponibles, seleccionando aquellos que ofrecen más funcionalidades y brindan mayores posibilidades de aprendizaje. Los programas seleccionados han sido Cisco Packet Tracer y GNS3, por utilizar imágenes de Cisco Systems, siendo los dispositivos de esta empresa los más extendidos, así como por las intuitivas interfaces que ofrecen y la potencia de emulación y virtualización en el caso de GNS3.

Las vulnerabilidades, ataques y soluciones de seguridad en redes de computadores componen un campo muy extenso en cuanto a contenidos se refiere. Para evitar el desorden que esta cuestión puede generar, se ha realizado una selección de contenidos basada en la importancia e impacto de las vulnerabilidades en la red y en la popularidad de los ataques, clasificándose según la capa del modelo OSI en la que se explotan dichas vulnerabilidades.

El material didáctico elaborado se compone de un apartado de teoría donde se presentan las cuestiones a tratar en cada apartado, completándose con la elaboración de figuras que expliquen de forma visual los conceptos considerados. Tras el apartado de teoría, le siguen un guion donde el alumno pondrá en práctica los ataques y defensas vistas con anterioridad y un guion con soluciones que resuelve detalladamente las cuestiones presentadas.

Las cuestiones introducidas en los guiones buscan guiar al alumno en la resolución de escenarios pre-diseñados en entornos de simulación y emulación, para poder aplicar las órdenes y conceptos vistos en el apartado de teoría, y, de esta manera, asentar los conocimientos al poder estudiar de una forma más real y visual las consecuencias de los ataques y defensas realizadas.

Las defensas para los ataques se han centrado en los dispositivos de red, para conseguir estratos de seguridad a un nivel más bajo que el ofrecido por firewalls, permitiendo de esta forma conocer y razonar más profundamente las vulnerabilidades y las contramedidas para evitar la explotación de las mismas.

La elección del entorno de simulación se justifica en cada apartado, basándose, principalmente, en la necesidad de virtualizar equipos personales (funcionalidad que GNS3 ofrece de forma satisfactoria) o en la necesidad de simular dispositivos de conmutación, como los switches, de forma más fidedigna, como ofrece Cisco Packet Tracer.

Para cumplir con el objetivo de trabajar con equipos reales y no solo en entornos de simulación se han realizado configuraciones de seguridad de dificultad creciente en un prototipo de laboratorio de redes, formado por equipos Cisco de gama media-alta, que, además, permiten la integración de voz y datos. La primera de las configuraciones busca un aseguramiento básico, centrado en la creación de redes locales virtuales,

configuración del acceso al router y creación de usuarios y contraseñas. La configuración media, tomando como punto de partida la configuración básica, ofrece un nivel de seguridad mayor, conseguido gracias a la autenticación de los mensajes del protocolo de encaminamiento RIP, desactivación de servicios vulnerables e implementación de listas de acceso. La última de las configuraciones es una configuración avanzada, centrada en mostrar el funcionamiento de la traducción de direcciones, las redes privadas virtuales y el protocolo de encaminamiento OSPF.

La financiación conseguida en este proyecto ha permitido la adquisición de dispositivos que implementan medidas de seguridad no disponibles en los equipos anteriores permitiendo así ampliar los métodos de defensas de algunos de los ataques. Además, la adquisición de un firewall nos permitirá en el futuro el desarrollo de materiales para este nuevo tipo de dispositivo del que no disponíamos hasta ahora en nuestro prototipo de laboratorio de redes.