UNIVERSIDAD OBERTA DE CATALUNYA

UNIVERSIDAD ROVIRA I VIRGILI

UNIVERSIDAD AUTÓNOMA DE BARCELONA

MÁSTER INTERUNIVERSITARIO DE SEGURIDAD DE LAS TECNOLOGIAS DE LA INFORMACIÓN Y DE LAS COMUNICACIONES

(MISITIC)

TRABAJO FIN DE MÁSTER

"MAN IN THE MIDDLE ATTACKS ON SSL/TLS"

ANTONIO JESÚS CARO ALONSO-RODRÍGUEZ

01/2013

UNIVERSIDAD OBERTA DE CATALUNYA

UNIVERSIDAD ROVIRA I VIRGILI

UNIVERSIDAD AUTÓNOMA DE BARCELONA

TRABAJO FIN DE MÁSTER

"MAN IN THE MIDDLE ATTACKS ON SSL/TLS"

DIRECTOR DE TESIS: GUILLERMO NAVARRO ARRIBAS

AUTOR: ANTONIO JESÚS CARO ALONSO-RODRÍGUEZ

01/2013

RESUMEN

La seguridad es un problema que nos preocupa a todos, en los últimos años las comunicaciones seguras a través de internet se han convertido en una necesidad. Este proyecto trata de esclarecer un ataque a una conexión segura para comprobar los riesgos a los que somos expuestos diariamente, cuando accedemos a nuestra cuenta bancaria, a las redes sociales, o navegamos libremente por internet, confiando en una conexión segura.

Esas redes seguras utilizan un protocolo que proporciona la autentificación y la privacidad de la información entre extremos sobre internet. Uno de estos protocolos es SSL/TLS. Aunque SSL/TLS es un protocolo relativamente seguro es vulnerable a algunos ataques.

El ataque que se utilizará será el Man-in-the-middle que consiste en que el atacante se interpone entre el cliente y el servidor. Con este ataque demostraremos como se pueden obtener credenciales en una conexión segura.

En primer lugar el atacante se introducirá en la red del atacado a partir de un ataque vía WIFI, para acceder a la red local. Una vez introducidos en la red se realizará un envenenamiento del router, para que nos redireccione los paquetes del cliente, y finalmente utilizaremos una herramienta para ver el texto en plano. Ver figura 1.1.

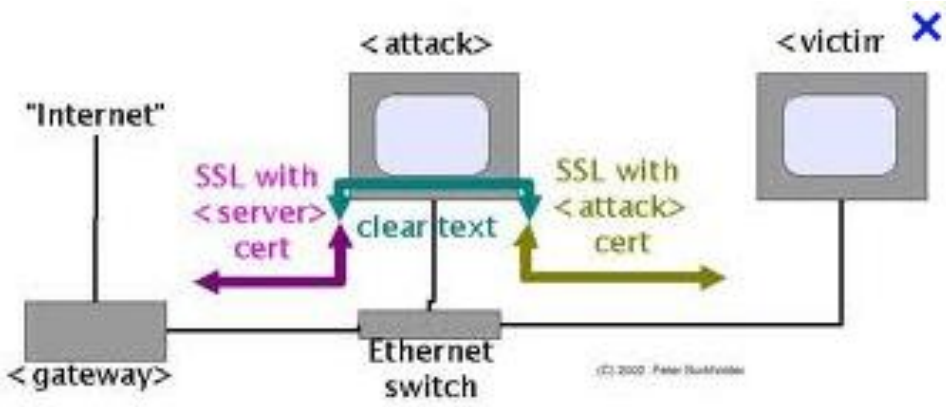


Figura 1.1 Ataque Man-in-the-middle sobre SSL [ALTAY]

ÍNDICE

LISTA DE FIGURAS

1. INTRODUCCIÓN

Descripción y objetivos

En este proyecto se realizará un ataque de Man-In-The-Middle sobre el protocolo SSL/TLS desde una red WIFI. Es decir se intentará situar al equipo atacante entre un cliente y un servidor sin que ambos sistemas tengan constancia de nosotros. Una vez allí podremos interceptar la comunicación entre ambos con una serie de herramientas.

El texto asume en el lector tiene unos conocimientos básicos sobre seguridad informática, por lo que no se introducirán términos básicos relacionados con la seguridad informática.

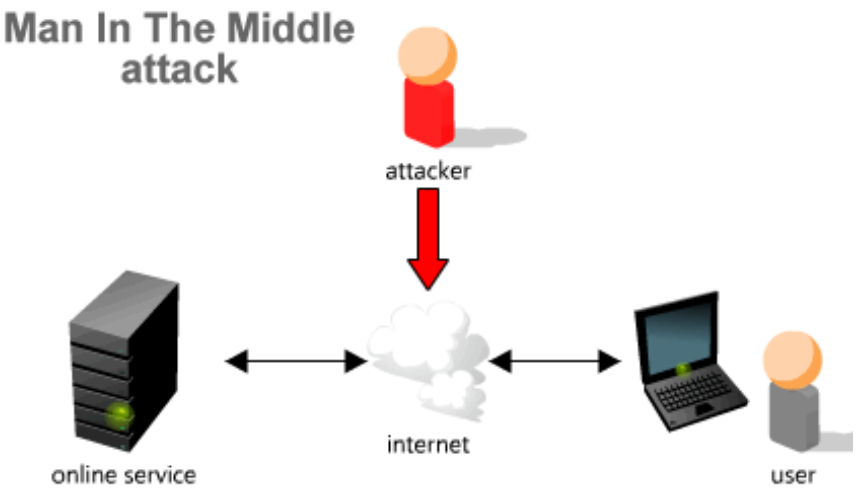


Figura 1.2: Man-In-The-Middle [PASSWIN]

El objetivo principal del proyecto, es demostrar las vulnerabilidades del protocolo SSL/TLS a partir de un ataque man-in-the-middle. Para conseguir este objetivo, se discernirá el proyecto en varios subapartados.

1. Estudio del protocolo SSL/TLS

2. Ataques sobre el protocolo SSL/TLS

3. Demostración de un ataque en un entorno de pruebas.

En el primer apartado se hará un estudio en profundidad sobre el protocolo SSL/TLS describiendo su funcionamiento, aplicaciones y estándares que

existen en la actualidad. Se hablará de las necesidades de los protocolos criptográficos y de las comunicaciones seguras en la red.

En el segundo apartado, se estudiarán las vulnerabilidades del protocolo SSL/TLS y los posibles ataques que puede sufrir dicho protocolo. Se estudiará en particular el funcionamiento del ataque MITM y sus consecuencias.

En el tercer apartado se realizará un ataque de MITM en un entorno de pruebas, donde se intentará ponerse entre un cliente y un servidor HTTPS que utilicen el protocolo SSL/TLS. Para ello se utilizará un servidor de pruebas y un PC en el que se intercambiarán información por el protocolo SSL/TLS. Por otro lado se utilizará un equipo atacante que interceptará dicha comunicación a través de WIFI y analizará el tráfico, obteniendo la información enviada por el emisor y el cliente. Para ello se deberá:

- Realizar un ataque WEP para acceder a la LAN.

- Realizar envenenamiento del router con arpspoof (MITM).

- Reedirecionar peticiones con iptables.

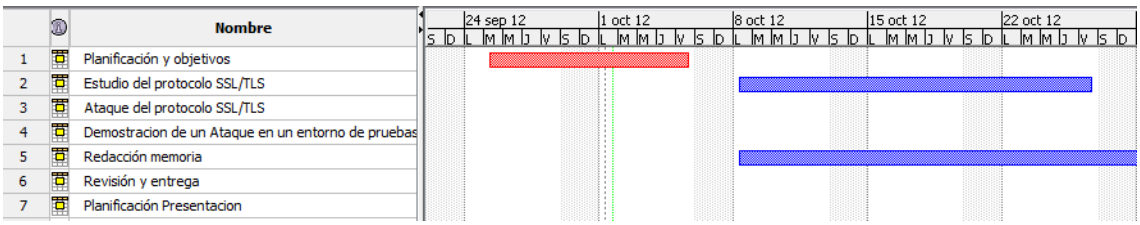
- Utilizar sslstrip para ver el texto en plano

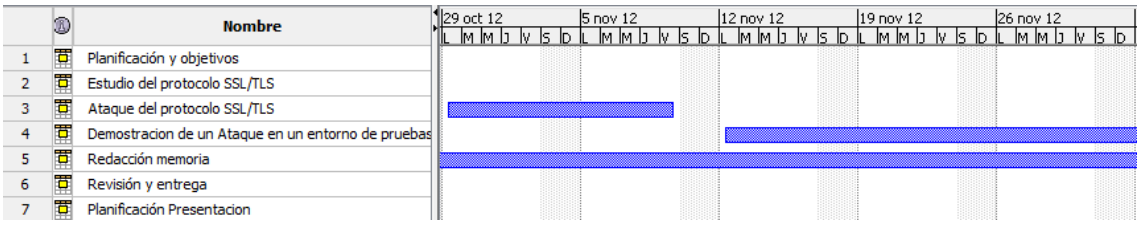
Para realizar dicho proyecto se ha realizado un estudio de la viabilidad y una planificación que se detallan a continuación:

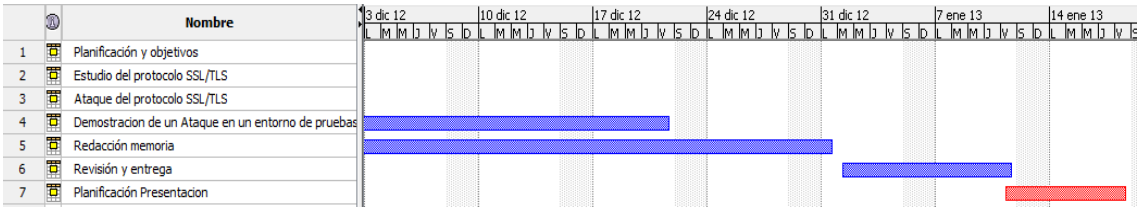
Planificación

Se utilizó la herramienta OpenProj para realizar la planificación del proyecto.

	Nombre	Duración	Inicio	Terminado
1	Planificación y objetivos	9 days?	25/09/12 8:00	5/10/12 17:00
2	Estudio del protocolo SSL/TLS	15 days?	8/10/12 8:00	26/10/12 17:00
3	Ataque del protocolo SSL/TLS	10 days?	29/10/12 8:00	9/11/12 17:00
4	Demostracion de un Ataque en un entorno de pruebas	30 days?	12/11/12 8:00	21/12/12 17:00
5	Redacción memoria	61 days?	8/10/12 8:00	31/12/12 17:00
6	Revisión y entrega	9 days?	1/01/13 8:00	11/01/13 17:00
7	Planificación Presentacion	6 days?	11/01/13 8:00	18/01/13 17:00







Estas fechas están sujetas a cambios excepto las entregas finales que corresponden a las evaluaciones del TFM

- 5 de Octubre: Entrega Planificación y Objetivos
- 14 de Enero: Entrega de la memoria TFM
- 21-25 de Enero: Exposición del TFM

Estudio de Viabilidad

En este apartado veremos los recursos necesarios, el coste y la viabilidad del proyecto.

Recursos necesarios:

- Una red LAN con Wireless (Encriptado WEP)
- Un equipo cliente (Windows 7 Entreprise con Internet explorer)
- Un servidor (HTTPS)
- Un equipo atacante (Backtrack 5)

Coste asociado:

Estudiaremos el coste que conlleva cada una de las tareas dispuestas en la planificación.

- Planificación y Objetivos: Se definirá un documento explicativo del proyecto. No conlleva coste económico si un coste de trabajo de nueve días.

- Estudio y Ataque del Protocolo SSL/TLS: Estas dos tareas corresponderán al descubrimiento de vulnerabilidades del protocolo SSL/TLS y en dar un mayor enfoque técnico de este protocolo. No tendrá coste económico pero si un coste de trabajo de veinticinco días.

- Demostración de un ataque en un entorno de pruebas: Para realizar la demostración del MITM se necesitarán recursos hardware y software. Estos recursos si llevan una carga económica, que son los siguientes:

  Una red LAN: Se necesita un router WIFI con cifrado WEP y un equipo mínimo conectado a esa red. El router será Router Wifi QuadBand DIR-825 cuyo precio es de 91,26 Euros

  Un equipo cliente: Se utilizará un PC de sobremesa con Windows 7 conectado a la red LAN. Este equipo será un MEDION AKOYA P5367E cuyo precio es de 849 Euros.

  Un servidor: Se utilizará una maquina virtual dentro del PC anterior con Apache que utilice conexiones seguras por SSL/TLS. El precio del servidor va incluido en el precio anterior

  Un equipo atacante (Backtrack 5): Se usará un portátil con el sistema operativo de backtrack. El coste de trabajo será de 30 días. El portátil será genérico y tiene un valor económico de 1140,50 Euros, con el software incluido.

- Redacción de memoria: Se irá realizando una memoria del proyecto a medida que se avance en el mismo. No tiene coste económico pero tiene una carga de trabajo de 61 días

- Revisión y entrega y planificación de la presentación: Se entregará la memoria y se preparará la presentación ante el tribunal. Llevará una carga de trabajo de 15 días.

  Coste de Trabajo: 9 + 61 + 9 + 6 = 85 días de trabajo

  Coste Económico: 91,26 + 849 +1140,50 = 2080,76 Euros

Viabilidad:

En primer lugar se disponen de los recursos y herramientas mencionadas anteriormente, por lo que se podría llevar a cabo el proyecto. Y atendiendo a la planificación planteada entra dentro de los plazos previstos para un TFM.

Para estudiar la viabilidad más allá de los recursos es ver si tenemos algún precedente, en el que se haya conseguido realizar un MITM con parecidos recursos anteriormente.

En 2009 Moxie Marlinspike, en una convención de blackhat, presentó la herramienta sslstrip para realizar ataques de MITM sobre SSL/TLS, donde el usuario piensa que está realizando una conexión por https cuando en realidad está usando http en texto claro.

2. ESTUDIO DEL PROTOCOLO SSL/TLS

2.1 Definición de SSL/TLS

El protocolo SSL (Secure Sockets Layer) y su sucesor TLS (Transport Layer Security) son protocolos criptográficos que proporcionan comunicaciones seguras por una red. Se basa en un proceso de cifrado de clave pública estableciendo un canal de comunicación seguro (cifrado) entre dos equipos después de una fase de autentificación [DAVBRU]. Esta comunicación se realiza a través de mensajes (serán explicados más adelante) entre el servidor y el cliente.

SSL proporciona confidencialidad, mediante el uso de encriptación, integridad, ya que los datos recibidos son exactamente iguales a los datos enviados y autentificación, ya que se realiza utilizando un certificado digital por parte del servidor. (Y cliente opcionalmente).

SSL se encuentra en la capa de transporte de la pila TCP/IP bajo la capa de aplicación, es decir bajo los protocolos HTTP, FTP, SMTP, etc. Es independiente del protocolo utilizado por lo tanto se pueden realizar transacciones a través de HTTP, FTP, POP e IMAP cifrando con el sistema SSL.

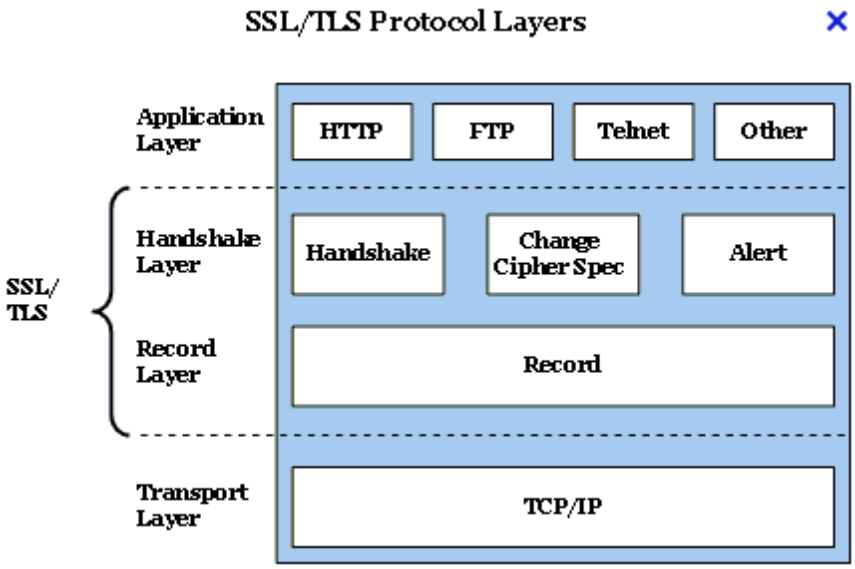


Figura 2.1: SSL en modelo TCP/IP. [MICRO]

El protocolo SSL está dividido en dos capas:

- SSL Record Layer
- SSL Handshake Layer

SSL Record Layer: Es usado para encapsular varios tipos de protocolo de mayor nivel. Proporciona una comunicación segura y principalmente esta capa toma los mensajes y los codifica con algoritmos de encriptación de llave simétrica aplicándole una MAC (Messsage Authentication Code).

SSL Handshake Layer: En la fase de negociación (Handshake) se negocian los algoritmos, se realiza la autenticación del servidor (opcionalmente el cliente) y se genera el secreto compartido. El alert protocol gestiona la sesión SSL, los mensajes de error y las advertencias El Change Ciphersuite Protocol es usado para indicar que una parte va a cambiar un ciphersuite que se ha negociado recientemente. Un ciphersuite es un nombre usado para la combinación de autenticación, encriptación y Message Authentication Code (MAC).

2.2 Funcionamiento de SSL/TLS

En primer lugar el protocolo negocia entre el cliente y el servidor los algoritmos que se utilizarán en la comunicación como (3DES, IDEA, AES, RSA, Diffie-Hellman, DSA, SHA-2, etc.). En segundo lugar se realizará el intercambio de claves y la autenticación basada en certificados digitales, utilizando una validación mediante una infraestructura de clave pública PKI cuando es necesario. Y tercero, el cifrado del tráfico basado en criptografía simétrica. Se genera una clave de sesión para la comunicación en función de los parámetros negociados. Esta clave facilitará el cifrado de los datos. La criptografía asimétrica sólo se utiliza en el intercambio de claves y en el firmado. Una vez concluye esta negociación comienza la conexión segura. [ANON]

Esta negociación entre el cliente y el servidor está basa en el intercambio de mensajes. En cada mensaje existe un campo (content_type) donde se especifica el protocolo de nivel superior utilizado. Estos mensajes pueden ser comprimidos, cifrados y empaquetados con un Message Authentication Code (MAC).

Existen dos tipos de negociación. Una en el que servidor se autentica con entrega de su certificado y el cliente no (Simple Handshake), y otra en la que se autentican con entrega de certificado las dos partes (Multiple Handshake).

Para la realización de la plataforma se utilizará Simple Handshake, así que se explicará a continuación cómo funciona la entrega de mensajes entre servidor y cliente para este tipo de negociación.

Simple Handshake

El servidor se autentica con entrega de su certificado y el cliente no.

Figura 2.2 Funcionamiento SSL/TLS[ANON]

1. El cliente inicia la comunicación enviando un mensaje “Client Hello” donde especifica una lista de conjunto de cifrados (Cipher Suites), métodos de compresión y la versión del protocolo SSL/TLS más alta permitida. También se envían bytes aleatorios que será usados más tarde (challege de Cliente).

2. El servidor responde con un mensaje “Server Hello” donde se indican los parámetros elegidos por el servidor a partir de las opciones ofertadas por el cliente. Se envía el ID de sesión, versión, compresión y un número aleatorio que se usará con el cliente en la creación de la clave simétrica.

3. Una vez establecidos los parámetros de la conexión, el servidor ofrece su certificado (Normalmente X.509) a través de un mensaje llamado “Certificate” Puede enviarse la clave pública o hacerlo en el paso 4.

4. Este mensaje se envía sólo si el anterior mensaje no incluye clave pública para el cifrado escogido. Con este mensaje el servidor ofrece para el cifrado asimétrico entre cliente y servidor la clave pública firmada con la clave del Certificado. De esta forma se separa la autenticación (paso 3, certificate) del cifrado. Lo más común es que el Certificado sea un certificado de autenticación de servidor que incluya la clave pública de cifrado

5. El servidor da por concluida su fase de negociación asimétrica.

6. El cliente tras haber comprobado y validado el certificado, genera el premaster secret que será el elemento secreto compartido que junto a otros datos intercambiados previamente darán lugar a un mismo master secret en la parte servidor y en el cliente y que será la clave simétrica utilizada para cifrar los datos. El premaster secret es cifrado con la clave pública del servidor y enviada. De este modo sabemos que solo el servidor legítimo puede descifrarlo y generar el master secret de la sesión.

7. Con este mensaje el cliente informa que los sucesivos datos estarán cifrados con el cifrado acordado.

8. El cliente da por finalizada su handshake. El mensaje que finaliza el Handshake incluye un hash de todos los mensajes de handshake que garantiza la integridad de la comunicación.

9. El servidor tras checkear que todo ha ido correctamente (en base al hashing MAC) , descifra con su clave privada el premaster secret enviado por el cliente en el paso 6 y genera el master secret del mismo modo que el cliente. En este momento cliente y servidor han logrado establecer una clave simétrica y acordado un cifrado. Con este mensaje el servidor indica que sus mensajes desde este momento se cifran.

10. El servidor finaliza el handshake

2.3 Aplicaciones de SSL/TLS

SSL/TLS fue diseñado para permitir protección de cualquier aplicación basada en un protocolo de transporte como TCP. Algunas de estas aplicaciones son [JOMAMO]:

HTTPS (HTTP sobre SSL/TLS): El protocolo más utilizado actualmente para la navegación web segura.

NNTPS (NNTP sobre SSL/TLS): Para el acceso seguro al servicio de News.

Estas aplicaciones con SSL/TLS funcionan exactamente igual que las originales. Las únicas diferencias son el uso de la capa de transporte seguro que proporciona SSL/TLS y la asignación de números de puerto TCP propios: 443 para HTTPS y 563 para NNTPS.

En muchos otros casos es preferible usar los mecanismos de extensión previstos en el propio protocolo de aplicación, si hay, para negociar el uso de SSL/TLS, a fin de evitar la utilización innecesaria de nuevos puertos TCP. Aplicaciones como:

TELNET, usando la opción de autenticación (RFC 1416)

FTP, usando las extensiones de seguridad (RFC 2228)

SMTP, usando sus extensiones para SSL/TLS (RFC 2487)

PO3 e IMAP, también usando comandos específicos para SSL/TLS (RFC 2595)

Algunos ejemplos de protocolos sobre SSL:

identificador	puerto TCP	descripción
https	443	HTTP sobre SSL
smtps	465	SMTP sobre SSL
nttps	563	NTTP sobre SSL
ldaps	646	LDAP sobre SSL
telnets	992	TELNET sobre SSL
imaps	993	IMAP sobre SSL
ircs	994	IRC sobre SSL
pop3s	995	POP3 sobre SSL
ftps-data	989	FTP-Datos sobre SSL
ftps-control	990	FTP-Control sobre SSL

Figura2.3: Aplicaciones sobre SSL [JOMAMO]

3. VULNERABILIDADES EN SSL/TLS

3.1 Vulnerabilidades

Pese a ser SSL/TLS un protocolo seguro, se han ido encontrando una serie de vulnerabilidades que explotan el propio protocolo SSL o alguna de las implementaciones que ha utilizado a lo largo de toda su historia.

Para encontrar las vulnerabilidades de dicho protocolo se usará una base de datos de vulnerabilidades de un CERT. En este caso la del INTECO (Instituto Nacional de tecnologías de comunicación) en colaboración con NVD (National Vulnerability Database).

Se mostrarán las vulnerabilidades más recientes sobre el protocolo SSL/TLS en la siguiente figura.

Resultado de la búsqueda de Vulnerabilidades

Mostrados 1 - 10 de 444 resultados

Vulnerabilidad en Tweepy (CVE-2012-5825)

Tipo: Validación de entrada

Gravedad: Media

Fecha de publicación: 04/11/2012

Última modificación: 05/11/2012

Descripción: Tweepy no comprueba si el nombre del servidor coincide con un nombre de dominio en el CN del sujeto o en el campo subjectAltName del certificado X.509, lo que permite ataques man-in-the-middle que permiten falsificar servidores SSL mediante un certificado válido de su elección. Se trata de un asunto relacionado con el uso de la biblioteca de Python httplib.

Vulnerabilidad en Trillian (CVE-2012-5824)

Tipo: Validación de entrada

Gravedad: Media

Fecha de publicación: 04/11/2012

Última modificación: 05/11/2012

Descripción: Trillian 5.1.0.19 no comprueba si el nombre del servidor coincide con un nombre de dominio en el nombre común (CN) del sujeto o en el campo subjectAltName del certificado X.509, lo que permite ataques man-in-the-middle que permiten falsificar servidores SSL a través de un certificado válido de su elección. Se trata de una vulnerabilidad diferente a CVE-2009-4831.

Vulnerabilidad en Open Source Classifieds (CVE-2012-5823)

Tipo: Validación de entrada

Gravedad: Media

Fecha de publicación: 04/11/2012

Última modificación: 05/11/2012

Descripción: Open Source Classifieds no comprueba si el nombre del servidor coincide con un nombre de dominio en el nombre común (CN) del sujeto o en el campo subjectAltName del certificado X.509, lo que permite ataques man-in-the-middle que permiten falsificar servidores SSL a través de un certificado válido de su elección. Se trata de un problema relacionado con el uso de la función PHP fsockopen.

Figura 3.1 Vulnerabilidades en SSL [INTEC]

Pese a encontrarse algunas vulnerabilidades sobre este protocolo desde 1995, muchas se han ido solucionando con las nuevas versiones o con las actualizaciones de dicho protocolo. En el siguiente apartado se verá que ataques ha sufrido SSL/TLS y en especial atención el ataque conocido como MITM (Man In The Middle).

3.2 Ataques en SSL/TLS (Man In The Middle)

Los protocolos SSL/TLS están diseñados para resistir los siguientes ataques. [XAPER]:

- **Lectura de los paquetes enviados por el cliente y servidor:** Al utilizar el protocolo SSL/TLS los datos se envían cifrados por lo que un atacante que utilice un sniffer para leer los paquetes se enfrenta al problema de romper el cifrado. Es preciso destacar que dependiendo la aplicación que utilice el protocolo SSL/TLS (Ej. HTTP para acceder a sistemas WEB) puede ser objeto de ataques engañando al usuario, como se verá en la plataforma de pruebas del punto 4.

- **Suplantación de servidor o cliente:** Cuando se realiza la autenticación del servidor (o cliente), el certificado digital debe estar firmado por la CA para verificar la identidad del propietario. Un posible ataque seria hacer que la CA firme un certificado no legítimo.

- **Alteración de los paquetes:** Un atacante puede modificar los paquetes para que lleguen a su destino con un contenido diferente del original. Por lo general el receptor detectará que el paquete viene alterado ya que el MAC será incorrecto.

- **Repetición, eliminación o reordenación de paquetes:** Aunque un atacante intentara enviar un paquete correcto que ya fue enviado o eliminar algún paquete haciendo que no llegue a su destino por lo general será detectado por el receptor ya que los códigos MAC no coincidirán con el valor esperado.

  Una vez visto los posibles ataques que puede recibir el protocolo SSL/TLS se entrará en más profundidad en el ataque Man In The Middle que será objeto de este trabajo.

Man-In-The-Middle con SSLSTRIP

Como hemos mencionado anteriormente, el ataque Man In the Middle consiste en situarse entre un equipo cliente y un servidor e interceptar la comunicación sin que ambas parte tengan constancia de lo que está ocurriendo.

Para ello se deberá situar al equipo atacante en la red LAN a atacar para que una vez introducidos en la red se pueda aplicar técnicas de ARP spoofing. Esta técnica consiste en envenenar la cache ARP de un equipo para hacerle creer que la MAC de la puerta de enlace es la dirección MAC del equipo atacante. [CHEALO]. Pudiendo de este modo situar la máquina atacante en medio de las comunicaciones efectuadas entre el cliente y el servidor.

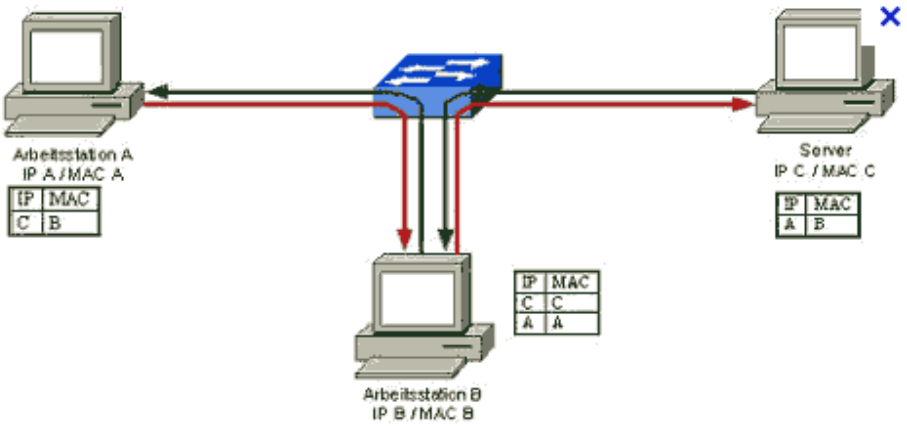


Figura 3.2: ARP spoofing [3MFUT]

Una vez realizado el ARP spoofing es hora de utilizar la herramienta SSLSTRIP para “descifrar todo el tráfico HTTPS”.

Lo primero que se debe decir es que SSLSTRIP no descifra el cifrado del protocolo SSL/TLS lo que hace es engañar al servidor y al cliente convirtiendo todo el tráfico HTTPS de una web en HTTP. Para que funcione el cliente debe acceder al servidor mediante una redirección o un link. [JALTHS]

En el siguiente apartado se explicará de forma más concisa y con la plataforma de ejemplo, el funcionamiento completo de este tipo de ataque y una descripción detallada de cada una de las herramientas.

3.3 Protección frente ataques

En primer lugar para protegerse frente a cualquier ataque, es recomendable tener actualizado el software con las últimas versiones proporcionadas por los fabricantes.

Para protegerse del ataque Man-In-The-Middle, que sigue siendo un problema potencial de seguridad en la actualidad lo mejor es usar protocolos de seguridad basados en clave pública, como por ejemplo SSL/TLS.

Aún así, se verá más adelante que es posible realizar un ataque MITM sobre este protocolo, en particular usando la herramienta SSLSTRIP engañando al usuario. Para protegerse de este ataque hay que escribir siempre la dirección en el navegador (https) cuando entremos en una Web que utilice SSL para cerciorarnos que realmente estamos utilizando una comunicación segura.

4. PLATAFORMA DE PRUEBAS

4.1 Descripción de la infraestructura

En este apartado se detallarán con atención los elementos necesarios para realizar la plataforma de prueba, describiendo uno a uno todos los componentes y la interconexión entre ellos.

- Servidor Web
- Equipo Victima (Windows 7)
- Equipo Atacante (BackTrack 5)

Servidor Web: Para el servidor web, se puede utilizar cualquier plataforma que soporte https, como un servidor local o un servidor web por internet. Como en el lado del servidor no se debe configurar ningún parámetro usaremos una aplicación web que esté en la red como puede ser paypal, gmail,tuenti…. Lo que realmente se busca es un servidor que proporcione la posibilidad de introducir un login en una comunicación aparentemente segura como puede ser SSL/TLS para poder descifrar los datos de un usuario. Ej. Login en tuenti:

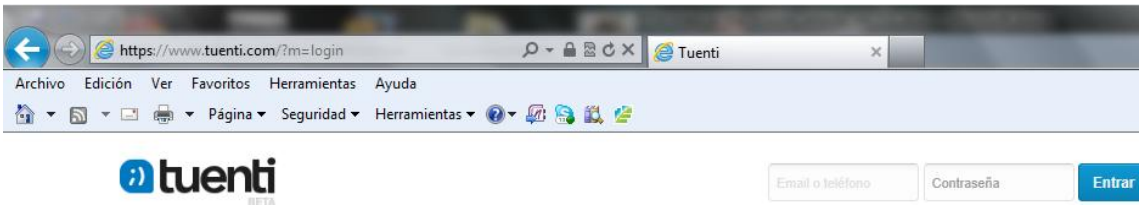


Figura 4.1: Login de Tuenti

Equipo Victima (Windows 7): El equipo víctima es un PC que debe disponer de un navegador web para conectarse al servidor web. En este caso utilizaremos Internet Explorer para acceder a dicho servidor. La conexión entre el equipo víctima y el router se realizará vía WIFI con un cifrado tipo WEP.

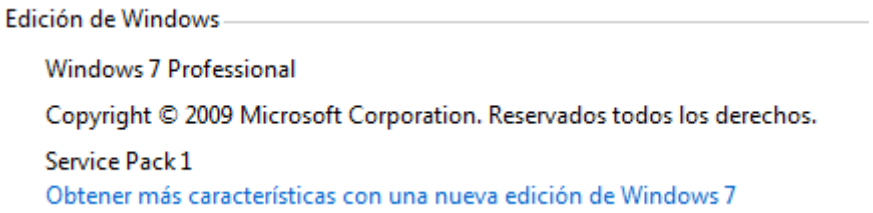


Figura 4.2: Sistema Windows 7 Professional

Equipo Atacante (BackTrack 5): Para realizar la parte de atacante se va a utilizar un portátil con el sistema operativo Backtrack que debe disponer de una serie de herramientas para la realización de la plataforma. Además debe disponer de antena WIFI para realizar el ataque WEP.

Las herramientas son las siguientes:

- **Aircrack-ng**: Es una suite de software de seguridad inalámbrica. Consiste en un analizador de paquetes de redes, un crackeador de redes WEP y WPA/WPA2-PSK y otro conjunto de herramientas de auditoría inalámbrica.

- **Arpspoof**: Es una técnica usada para infiltrarse en una red que puede permitir al atacante leer paquetes de datos en la LAN (red de área local), modificar el tráfico, o incluso detenerlo.

- **iptables**: Es una herramienta de cortafuegos que permite filtrar paquetes

- **sslstrip**: La herramienta sslstrip permite hacer creer a un usuario que está navegando por una conexión segura (HTTPS) cuando realmente no lo está haciendo (HTTP). El funcionamiento de sslstrip consiste en realizar una conexión HTTP entre el atacante y la victima y realizar una conexión segura (HTTPS) entre el atacante y el servidor Web. SSLStrip explota el paso entre HTTP y HTTPS para engañar a la víctima y recibir los credenciales en texto claro por HTTP

- **Wireshark**: Es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones

Figura 4.3: Backtrack 5

Interconexión: En primer lugar tendremos el PC de la victima conectado al router a través de la WIFI. La víctima se conectará al servidor web para acceder a algún contenido de su cuenta se usuario como puede ser paypal, gmail, twitter, etc... Por otro lado el atacante realizará un ataque WEP para conectarse a la red local de la víctima y desde ahí se realizará una serie de operaciones para poder deducir los credenciales que la víctima ha enviado al servidor WEP.

4.2 Ataque WEP a la red LAN

En primer lugar debemos poner nuestra tarjeta de red en modo monitor para poder observar todas las redes WIFI que tenemos a nuestro alcance.

Airmon-ng start wlan0

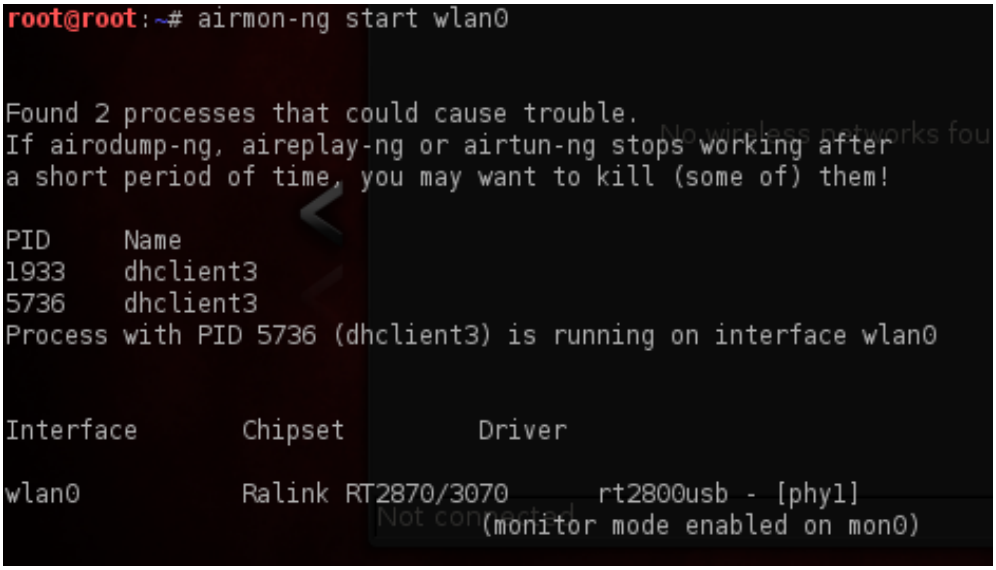


Figura 4.4: Airmon-ng

Una vez en modo monitor ya podemos rastrear la red en busca de posibles redes que podamos atacar para ello utilizamos el siguiente comando.

Airodump-ng mon0

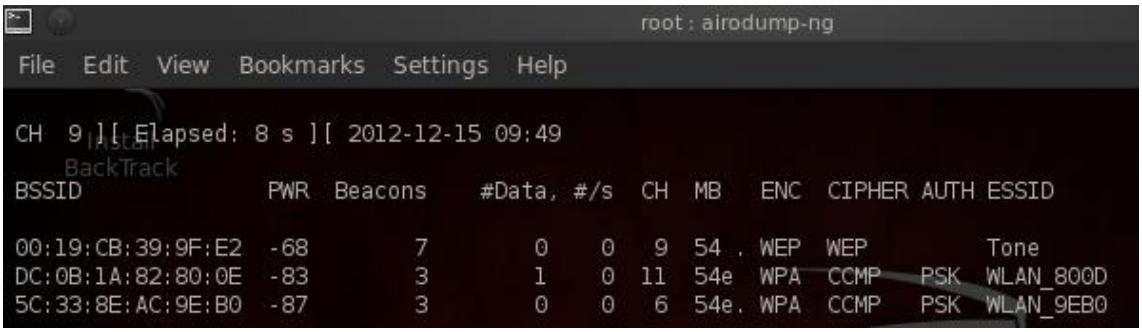


Figura 4.5: Airodump-ng

Como podemos observar hay tres redes a nuestro alcance, en particular interesa la que tiene el cifrado WEP por ser la más fácil de romper. A partir de ahora se monitorizará esa red en particular para capturar los máximos datos posibles.

Airdump-ng –c 9 –w PruebaMistic –bssid 00:19:CB:39:9F:E2 mon0

Figura 4.6: Airodump-ng 2

Ahora realizaremos una serie de ataques con aireplay para intentar capturar el mayor número de datos posible con el fin de obtener la clave. Por lo menos hay que tener unos 10000 data para obtener la clave.

Ataque 1: Falsa Autentificación: aireplay-ng -1 0 –a 00:19:CB:39:9F:E2 –h 00:0c:29:a4:2e:9b –e Tone mon0

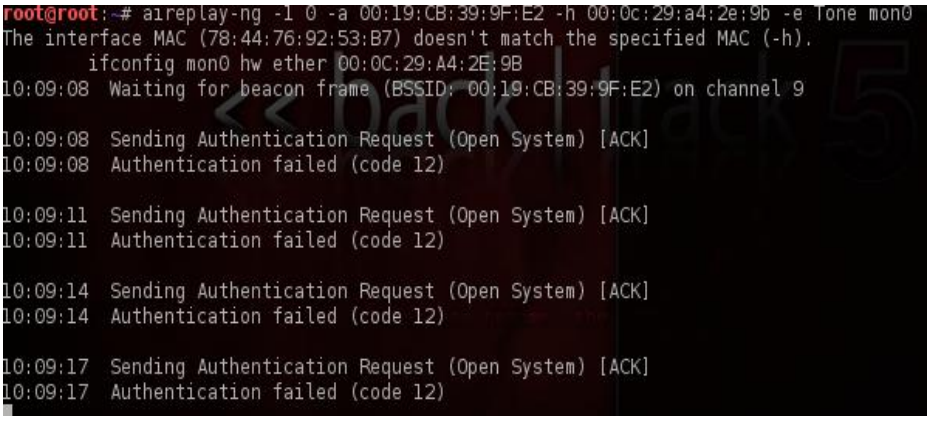


Figura 4.7: Falsa Autentificación

Modo autenticación falsa. En este modo se realiza un ataque de falsificación de autenticación como el que hemos visto anteriormente. Esto puede ser necesario para realizar posteriormente otros ataques si no hay ninguna otra estación asociada.

Ataque 2: Reinyección de una petición ARP: aireplay-ng -3 –b 00:19:CB:39:9F:E2 –h 00:0c:29:a4:2e:9b mon0

Figura 4.8: Reinyección de una petición ARP

Modo inyección de paquetes ARP. Este es probablemente el modo más efectivo para forzar el envío de tramas WEP y poder capturar así los IV necesarios para obtener la clave. En este modo, la herramienta escucha el medio hasta que detecta una petición ARP.

Ataque 3: Desauntetificación aireplay-ng -0 1 –a 00:19:CB:39:9F:E2 –c 00:13:F7:58:30:C3 mon0

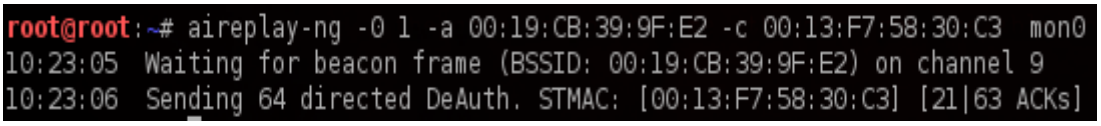


Figura 4.9: Desauntetificación

Modo desautenticación. En este modo se generan tramas falsas de desautenticación destinadas a una estación autenticada con el AP. El objetivo es provocar que esta estación inicie una nueva autenticación y, dependiendo del sistema operativo con que trabaje, envíe una petición ARP para averiguar la dirección IP del AP que hace de router.

Una vez que hemos alcanzado los 20000 lanzamos el aircrack-ng para intentar crackear la contraseña determinada. Si aún no podemos descifrar la clave habrá que esperar a alcanzar 25000 IVS para intentarlo otra vez y así sucesivamente.

Aircrack-ng PruebaMistic-01.cap

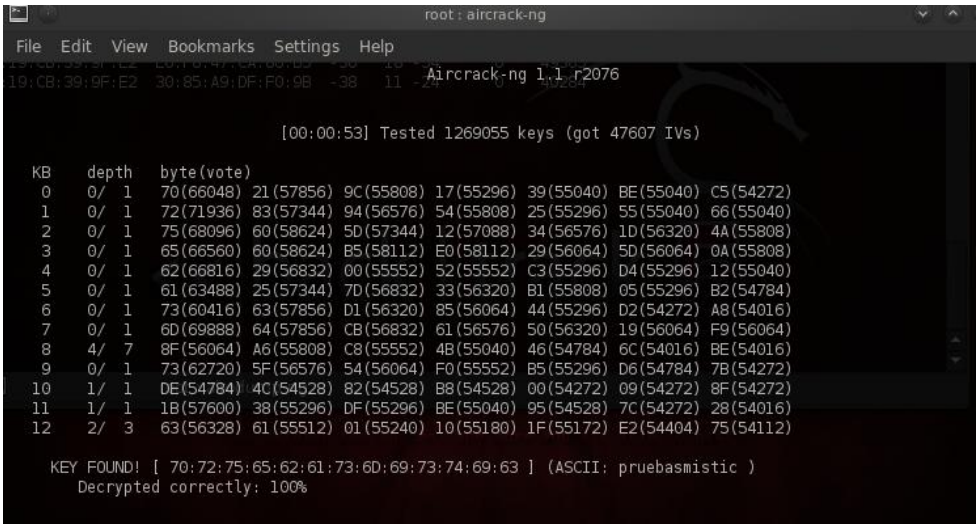


Figura 4.10: Airack-ng

Esta es la herramienta que a partir de los IV obtenidos implementa el ataque para recuperar la clave WEP. Como se puede observar en la figura ya tenemos la clave WEP tan ansiada**: pruebasmistic**, ahora nos conectamos a la red WIFI y podemos pasar al siguiente apartado.

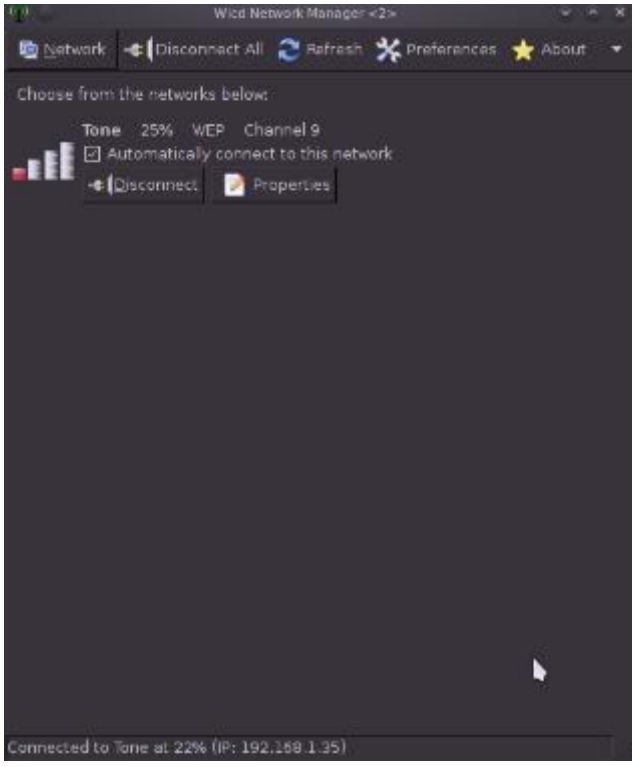


Figura 4.11: Wicid

4.3 Ataque Man-in-the-middle

Como se ha visto el equipo atacante ya se ha introducido en la red local, es hora de empezar a capturar el tráfico y obtener los credenciales de la víctima.

En primer lugar se debe conocer la IP privada de la víctima, para ello usaremos la herramienta nmap. Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos que se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática.

En este caso se escaneará toda la red local en busca de Host activos, y encontraremos IP que pertenezcan a nuestra red local.

Descubriendo IP víctima: nmap –T4 –A –v 192.168.1.*

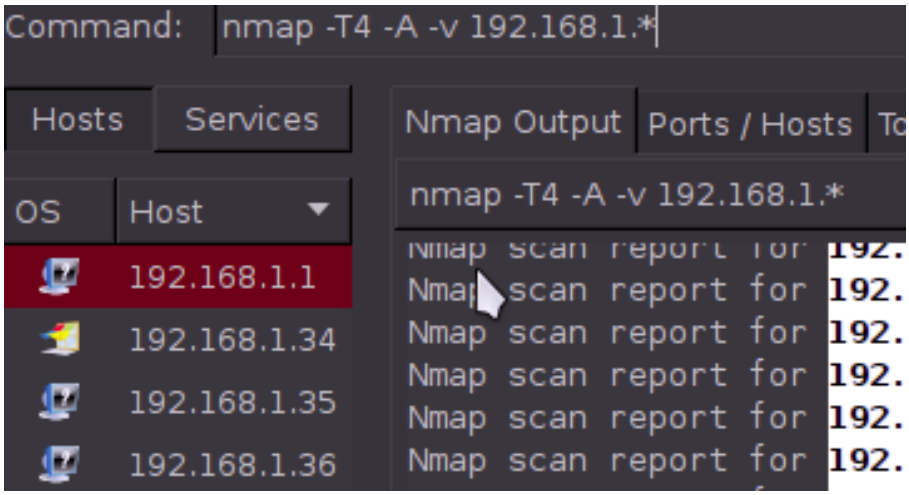


Figura 4.12: Nmap

Donde se puede comprobar que la dirección 192.168.1.34 corresponde con el equipo Windows 7 que queremos atacar.

Llegados a este punto debemos redireccionar el tráfico desde nuestra máquina atacante hacia la víctima, ya que si no hiciéramos este paso, la víctima no tendría acceso a internet y por lo tanto se percataría de que algo no anda bien.

En primer lugar activaremos el reenvío de paquetes con la siguiente directiva:

Echo 1 > /proc/sys/net/ipv4/ip_forward

```
root@bt:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Figura 4.13: ip_forward

Una vez que se ha activado la directiva, se deben configurar las iptables. Iptables es una herramienta de cortafuegos que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de log. Lo que se hara a continuación es usar las iptables para redirigir todo el tráfico del puerto 80 al puerto de escucha del SSLStrip, en este caso el puerto 10000.

Para ello se pone el siguiente comando:

Iptables –t nat –A PREROTUING –p tcp –destination-port 80 –j REDIRECT –to-port 10000

Figura 4.14: Iptables

Ya tenemos el tráfico redirigido, ahora toca envenenar el router en el cual están conectadas las máquinas de la víctima y del atacante. En este caso lo que haremos es enviar mensajes ARP falsos para que el router asocie nuestra MAC con la dirección IP de la víctima. De este modo todo el tráfico que debería ser enviado a la victima será enviado al atacante. Como anteriormente se ha permitido reenviar el tráfico de paquetes que nos llegue, la víctima no tendrá percepción alguna de que todo el tráfico que está transcurriendo entre él y el servidor web está siendo cazado por el atacante.

Arspoof –i wlan0 –t 192.168.1.34 192.168.1.1

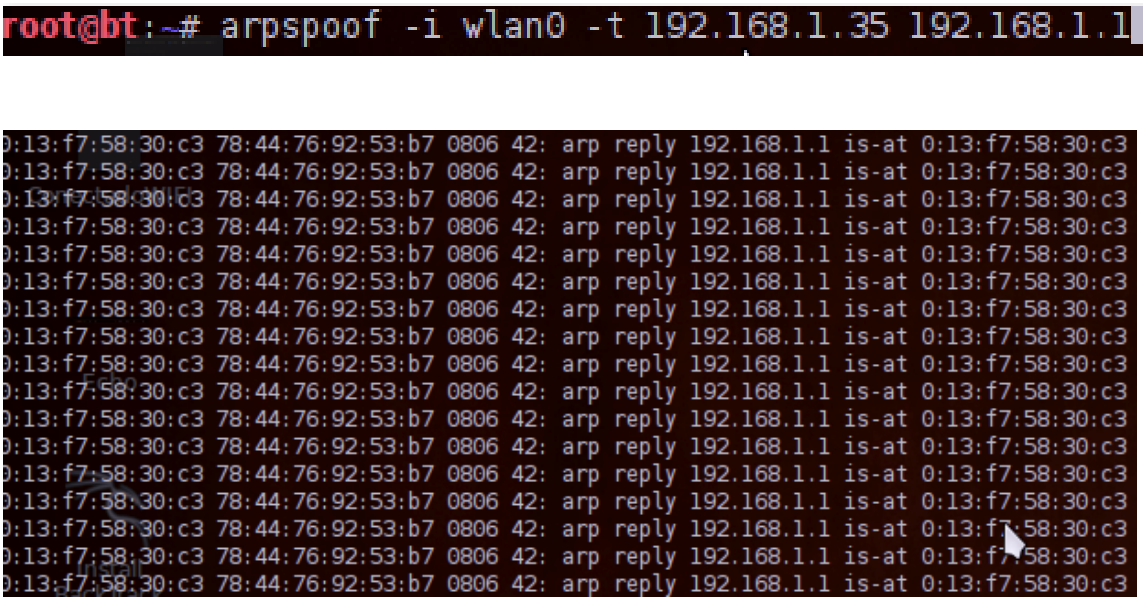


Figura 4.15: arspoof

Después de esperar un tiempo en que se envene la cache del router ya se pueden observar los datos que se envían entre el cliente y el servidor, si ahora mismo usáramos Wireshark para capturar el tráfico, ya tendríamos los paquetes que se transfieren ambos. Pero lo que realmente se quiere es poder obtener credenciales cuando la víctima introduzca sus datos de acceso en cuentas como gmail, paypal, twitter, etc... Para ello se utilizará la herramienta sslstrip, sslstrip es una aplicación para sistemas operativos Linux capaz de "descifrar todo el tráfico HTTPS" que viaja a través de la red y sniffar el tráfico que viaja a través de la red en "HTTPS (cifrado)". Para ello lo que hace es hacer creer al usuario que está navegando por una conexión cifrada por https cuando en realidad está navegando en http.

Para ello se usará el siguiente comando:

Python sslstrip.py –w misticSSL –l 10000

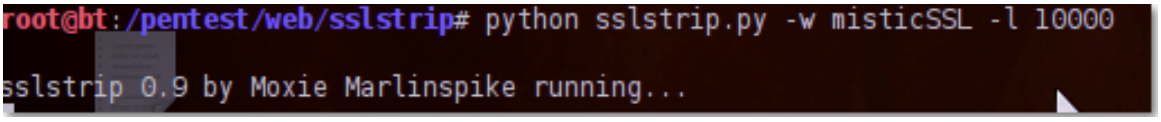


Figura 4.16: SSLstrip

Ahora solo queda esperar a que la víctima se conecte a algún servicio de login y obtener sus credenciales.

Como se puede observar en la siguiente imagen, la victima accede a tuenti sin la protección de SSL e introduce sus credenciales

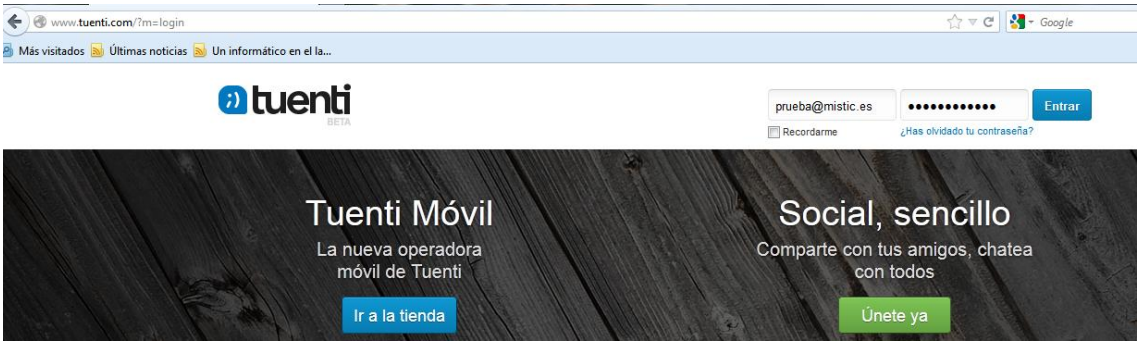


Figura 4.17: Tuenti sin https

Abriendo el fichero guardado en misticSSL encontramos los credenciales introducidos por la víctima. Como se puede observar al introducir el siguiente comando:

Cat misticSSL

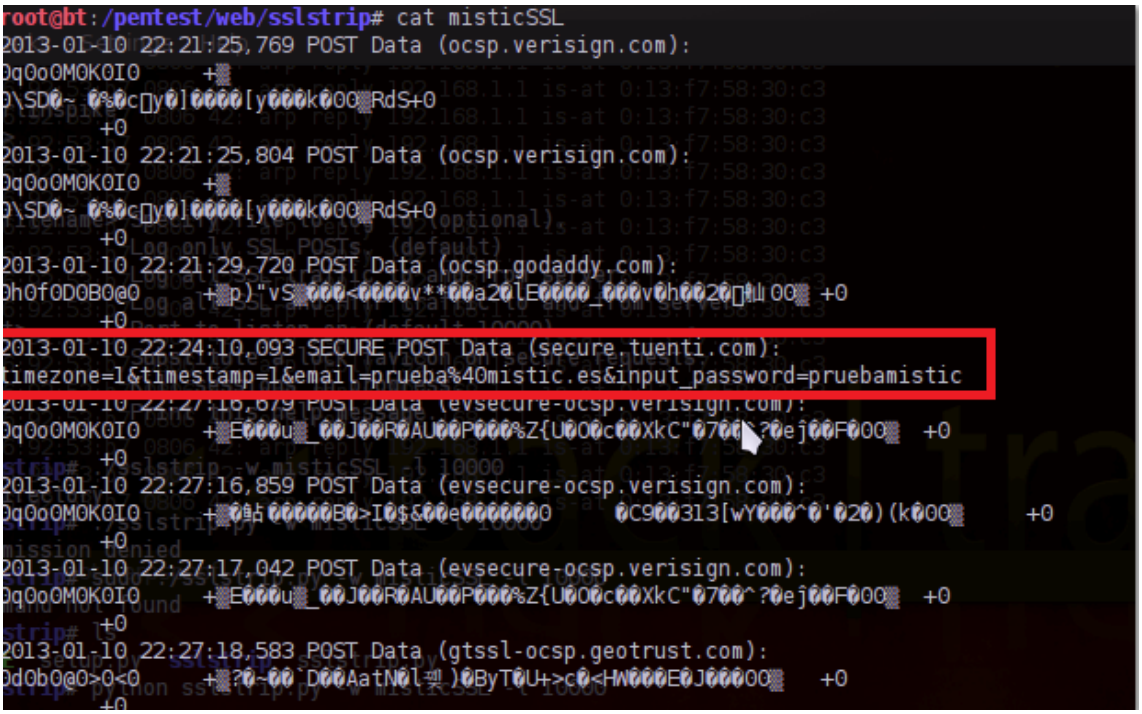


Figura 4.18: Archivo MisticSSL

Donde se puede ver que el mail era prueba@mistic.es y el password: pruebamistic.

5. CONCLUSIÓN

Objetivos Conseguidos:

Los objetivos de este proyecto eran:

Demostrar las vulnerabilidades del protocolo SSL/TLS a partir de un ataque man-in-the-middle con los siguientes subapartados.

1. Estudio del protocolo SSL/TLS

2. Ataques sobre el protocolo SSL/TLS

3. Demostración de un ataque en un entorno de pruebas

Como se ha podido comprobar los protocolos SSL son vulnerables a ataques MITM al acceder a un servidor web por HTTPS. Además se han cumplido todos los objetivos marcados al inicio del presente proyecto.

Planificación Final:

La planificación inicial era la correspondiente a las entregas finales que corresponden a las evaluaciones del TFM

- 5 de Octubre: Entrega Planificación y Objetivos
- 14 de Enero: Entrega de la memoria TFM
- 21-25 de Enero: Exposición del TFM

Por lo que la planificación general y todos los hitos intermedios se han cumplido con éxito tal y como estaban expresado en la planificación inicial.

Resultados

Uno de los principales fallos de seguridad que se han visto en el proyecto es la facilidad para quebrantar la seguridad del protocolo WEP, dicho protocolo ya no es recomendado para su utilización en redes WIFI, ya que su seguridad es completamente nula, y la obtención de la contraseña es fácilmente obtenible. Por ello se recomienda utilizar otro tipo de encriptaciones más seguras como puede ser WPA, WPA2 o WPA-PSK. Además se puede utilizar otro tipo de protecciones como el filtrado por MAC o ocultar el essid para intentar poner más difícil un posible ataque.

La protección de una red wifi es primordial, pero generalmente el sentido común y la prevención suponen un gran aliado a la hora de securizar nuestros sistemas, no obstante, si en la plataforma de pruebas la víctima se hubiera percatado que ya no disponía de una protección segura al mirar la URL, nunca se hubiera obtenido sus credenciales, ya que cabe recordar que lo que hace

SSLStrip no es desencriptar ssl sino establecer una conexión en texto plano en entornos de conexión segura entre el servidor y la victima.

Debido a que se ha demostrado que se pueden realizar ataques MITM sobre el protocolo SSL, se está implementando el protocolo HTTP Strict Transport Security (HSTS). El 19 de Noviembre del 2012 se publicó la especificación que regula el protocolo (RFC 6797)

Dicho protocolo de seguridad plantea una política mediante la cual un servidor web sólo pueda obtener conexiones con los agentes de usuario (IE, Chrome, Firefox) de manera segura sobre el protocolo SSL/TLS, evitando de esta forma que se pudiera realizar el ataque Man-In-The-Middle sobre este protocolo.

6. BIBLIOGRAFIA

[STEPTH] SSL AND TLS ESSENTIALS: SECURING THE WEB – STEPHEN THOMAS

[DAVBRU] ANALYSIS OF THE SSL 3.0 PROTOCOL – DAVID WAGNER, BRUCE SCHNEIER

[JOMAMO] SSL, SECURE SOCKETS LAYER Y OTROS PROTOCOLOS SEGUROS PARA EL COMERCIO ELECTRÓNICO – JOSE MARIA MORALES

[ACATEL] PROTOCOLOS DE SEGURIDAD – ACADEMICA TELMEX
http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/protocolos-de-seguridad

[CRIPCS] SECURE SOCKET LAYER (SSL) – CRIPTONOMICOM, CSIC
http://www.iec.csic.es/criptonomicon/ssl.html

[ARHERO] EL PROTOCOLO SSL – ARIEL HERNÁN ROEL
http://penta2.ufrgs.br/gereseg/unlp/tut1998/ssl.htm

[RAFPAL] PROTOCOLO DE SEGURIDAD EN LA CAPA DE TRANSPORTE – RAFAEL PALACIOS
http://www.iit.upcomillas.es/palacios/seguridad_dr/tema4_ssl.pdf

[DANSEP] SSL (SECURE SOCKET LAYER) Y TLS (TRANSPORT LAYER SECURE) – DANIEL SEPULVEDA
http://www.wikilearning.com/curso_gratis/protocolos_seguros_para_el_web-ssl_secure_socket_layer_y_tls_transport_layer_secure/6091-4

[JMLOF] EL WEB, SERVIDORES Y TECNOLOGIAS ASOCIADAS – JOSE MANUEL LOPEZ FRANCO
http://trevinca.ei.uvigo.es/~txapi/espanol/proyecto/superior/memoria/node136.html

[CARERA] PROTOCOLO TLS (TRANSPORT LAYER SECURITY) – CARLOS ERAZO
http://www.monografias.com/trabajos74/protocolo-tls-transport-layer-security/protocolo-tls-transport-layer-security2.shtml

[IDEAS] PROTOCOLO SSL – INSTITUTO DE APRENDIZAJE SUPERIOR (IDEAS)
http://www.ideaseducativas.org/inet/redes2/redes2rep7protocolossl.pdf

[XAPER] MECANISMOS DE PROTECCIÓN – XAVIER PERRAMON
http://deic.uab.es/material/26118-ssl.pdf

[JALTHS] SSLSTRIP: MANUAL DE SSLSTRIP PARA DESCIFRAR TRAFICO HTTPS – JALTHS, REDESZONE.NET
http://www.redeszone.net/seguridad-informatica/sslstrip-manual-de-sslstrip-para-descifrar-todo-el-trafico-https/

[CHEALO] ATAQUE MAN IN THE MIDDLE CON DHCP ACK INJECTOR – CHEMA ALONSO
http://www.elladodelmal.com/2011/10/ataque-man-in-middle-con-dhcp-ack.html

[INTEC] CATALOGO DE VULNERABILIDADES – INTECO
http://cert.inteco.es/Catalogo_STIC/Catalogo/

[3MFUT] ARP GUARD – PROTECTION FROM ARP SPOOFING ATTACKS – 3MFUTURE
http://www.3mfuture.com/network_security/arp-guard-arp-spoofing.htm

[PASWIN] MAN-IN-THE-MIDDLE ATTACK PROTECTION – PASSWINDOW
http://www.passwindow.com/security.html

[ALTAY] MITM SSL CERTIFICATES FOR SALE TO CORPORATIONS – ALAN TAYLOR
http://www.pgpboard.com/viewtopic.php?f=2&t=620

[ANON] ANALIZANDO SSL: FUNCIONAMIENTO Y EJEMPLOS – ANONIMO
http://lobobinario.blogspot.com.es/2011/02/analizando-ssl-iii-de-iii.html

[MICRO] OVERVIEW OF SSL/TLS ENCRYPTION – MICROSOFT
http://technet.microsoft.com/en-us/library/cc781476(v=ws.10).aspx