Universidad de Granada

Escuela Técnica Superior de Ingeniería Informática y Telecomunicaciones
Grado en Ingeniería Informática

Asignatura: Servidores Web de Altas Prestaciones

Seguridad en servidores y ciber-ataques

Autores: Míriam Mengíbar Rodríguez
Juan Anaya Ortiz

Granada, 17 de mayo de 2017

Índice

1. Introducción

En la actualidad, la seguridad es un aspecto relevante en el ámbito de la informática, ya que cualquier sistema necesita protección para evitar que el contenido del mismo se vea comprometido. El diseño de un sistema requiere la incorporación de elementos que aporten seguridad al sistema, ya que debemos asegurar los recursos del sistema de forma que se utilicen de la manera en la que se planeo y que el acceso y/o modificación de la información solo sea posible a las personas que se encuentren acreditadas.

Si consideramos la seguridad en cualquier máquina sencilla que pueda tener una persona, es cierto que se deben tomar precauciones, ya que, aunque a veces no nos demos cuenta, nuestra máquina posee datos personales (por ejemplo cuentas bancarias, DNI, números de tarjetas de crédito, etc). Pero en el ámbito de los servidores, este tema tendría que ser aún más importante si cabe, ya que los servidores, por lo general, están destinados a manejar grandes cantidades de datos personales y críticos de empresas y de sus clientes. Así pues, una vulneración del servidor haría que dichas empresas perdiesen grandes cantidades de dinero además del valor ético que supone que vulneren la privacidad de muchas personas.

Por ello, surgen empresas destinadas a mantener y proteger la seguridad del resto de empresas. Pero ¿qué pasa con los usuarios estándares que no reciben esta protección? Por todos los motivos anteriores y más que pueden escaparse del ámbito de la asignatura, los informáticos jugamos un papel importante, pero la concienciación de la sociedad, tanto a nivel de usuarios estándares como a nivel de empresas, debe comenzar en una etapa temprana, pues la mayoría de personas no conocen los riesgos que pueden suponer ciertas acciones y de lo que podemos encontrar en Internet.

2. La seguridad [1]

La seguridad informática [9] [10] es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. A decir verdad, no hay ningún algoritmo para conseguir la seguridad completa e impenetrable, así que solo trata de poner barreras que eviten la vulneración del sistema.

La seguridad informática se dedica principalmente a proteger la confidencialidad, la integridad, no repudio, autenticación y la disponibilidad de la información. Estas características podemos describirlas como sigue:

1. **Disponibilidad:** capacidad de un servicio,de unos datos o de un siste-

ma a ser accesible y utilizable por los usuarios o procesos autorizados cuando lo requieran.

2. **Confidencialidad:** cualidad que debe poseer un documento o archivo para que éste solo se entienda de manera comprensible o sea leído por la persona o sistema que esté autorizado.

3. **Integridad:** cualidad que posee un documento o archivo que no ha sido alterado y que además permite comprobar que no se ha producido manipulación alguna en el documento original.

4. **Alta disponibilidad**: son sistemas que están disponibles las 24 horas al día, 7 días a la semana, 365 días al año.

5. **Autenticación:** Es la situación en la cual se puede verificar que un documento ha sido elaborado o pertenece a quien el documento dice. La autenticación de los sistemas informático se realizan habitualmente mediante nombre y contraseña.

6. **No repudio:** El no repudio es un servicio de seguridad estrechamente relacionado con la autenticación y que permite probar la participación de las partes en una comunicación. Puede existir repudio en el origen (el emisor no puede negar el envío porque el destinatario tiene pruebas del mismo el receptor recibe una prueba infalsificable del envío.) o de destino (el receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción.)

3. Amenazas [1]

El sistema puede estar expuesto a una serie de amenazas que hace que perturbe su seguridad. Las podemos clasificar como sigue.

3.1. Humanas

Las personas pueden comprometer la seguridad del sistema por fallos humanos o por acciones intencionadas:

1. **Personal:** las personas pueden comprometer la seguridad del sistema por fallos humanos.

2. **Hackers:** una persona que intenta tener acceso no autorizado a los recursos de la red con intención maliciosa o no.

3. **Crackers:** es un término mas preciso para describir una persona que intenta obtener acceso no autorizado a los recursos de la red con intención maliciosa.

4. **Intrusos remunerados:** se trata de personas con gran experiencia en problemas de seguridad y un amplio conocimiento del sistema que son pagados por una tercera parte para vulnerar el sistema.

3.2. Amenazas lógicas

Son aquellas que son a causa del software que usemos o que usen terceras personas:

1. **Software incorrecto:** A los errores de programación en el software se les llama *Bugs* y a los programas para aprovechar uno de estos fallos se les llama *Exploits*.

2. **Herramientas de seguridad:** Cualquier herramienta de seguridad representa un arma de doble filo de la misma forma que un administrador las utiliza para detectar y solucionar fallos en sus sistemas o la subred completa un intruso las puede utilizar para detectar esos mismos fallos y aprovecharlos para atacar los equipos.

3. **Puertas traseras:** Son parte de código de ciertos programas que permanecen sin hacer ninguna función hasta que son activadas. En ese punto la función que realizan no es la original del programa si no una acción perjudicial.

4. **Canales cubiertos:** Son canales de comunicación que permiten a un proceso trasferir información de forma que viole la política de seguridad del sistema.

5. Virus: Es una secuencia de código que se inserta en un fichero ejecutable de forma que cuando el archivo se ejecuta el virus también lo hace.

6. **Gusanos:** Programa capaz de ejecutarse y propagarse por si mismo a través de redes, en ocasiones portando virus o aprovechando *Bugs* de los sistemas a los que se conecta, para dañarlos. Son difíciles de programar, pero son muy dañinos.

7. **Troyanos:** Su nombre es debido al famoso caballo de Troya, son instrucciones escondidas en un programa de forma que este parezca realizar las tareas que un usuario espera de el pero que realmente ejecuta

funciones ocultas. Este programa no hace nada útil, simplemente se delimitan a reproducirse hasta que el número de copias acaba con los recursos del sistema.

3.3. Amenazas físicas

Estas se refieren a las amenazas que puede sufrir un sistema debido a robos, sabotajes, destrucción de sistemas, suministro eléctrico, condiciones atmosféricas, catástrofes naturales, etc.

4. Honeypots [2] [3]

Para intentar paliar alguna de estas amenazas, muchos expertos en seguridad hacen uso de los *honeypots*. Un *Honeypot* es el software o conjunto de ordenadores cuya intención es atraer a atacantes, simulando ser sistemas vulnerables o débiles a los ataques. Es una herramienta de seguridad informática utilizada para recoger información sobre los atacantes y sus técnicas. Los *Honeypots* pueden distraer a los atacantes de las máquinas más importantes del sistema, y advertir rápidamente al administrador del sistema de un ataque, además de permitir un examen en profundidad del atacante, durante y después del ataque al *Honeypot*.
Actualmente existen multitud de herramientas libres que pueden ayudar en el despliegue de *Honeypots* para este tipo de investigaciones o incluso, teniendo los conocimientos necesarios, desarrollar software personalizado. Algunas de estas herramientas se usan desde consola de comandos, siendo necesario analizar e interpretar los *logs* que generan para saber que está ocurriendo, otras poseen un entorno gráfico desde donde poder visualizar los datos obtenidos.

5. Los ciberataques [4] [5]

A medida que evoluciona el entorno de las amenazas cibernéticas, también debe desarrollarse la protección frente a dichas amenazas. Con la aparición de los ataques dirigidos y las amenazas persistentes avanzadas, queda claro que es necesario utilizar un nuevo enfoque de seguridad cibernética. Las técnicas tradicionales simplemente ya no resultan adecuadas para proteger los datos frente a los ciberataques.
Las amenazas persistentes avanzadas y los ataques dirigidos han demostrado su capacidad para penetrar en las defensas de seguridad estándares y permanecer ocultos durante meses mientras sustraen datos valiosos o llevan a cabo acciones destructivas. Muchas de las empresas en las que confía son

algunos de los objetivos principales: instituciones financieras, organizaciones sanitarias y grandes minoristas, entre otros.
En 2011, PC World detectó un aumento del 81 % en los ataques de hackers avanzados y dirigidos a equipos y, según los resultados de una investigación de Verizon en 2012, se alcanzó la asombrosa cifra de 855 incidentes de seguridad cibernética y 174 millones de registros atacados.

A continuación vamos a describir los ataques más populares:

5.1. DDoS o DoS [6] [7]

El objetivo de un ataque *DDoS (Distributed Denegation of Service)* es inhabilitar un servidor, un servicio o una infraestructura sobrecargando el ancho de banda del servidor o acaparando sus recursos hasta agotarlos. Durante un ataque *DDoS*, se envían multitud de peticiones simultáneamente desde múltiples puntos de la red. La intensidad de este "fuego cruzado"desestabiliza el servicio o, aún peor, lo inhabilita. El *DoS (Denegation of Service)*, es prácticamente lo mismo, solo que el ataque se realiza desde una única máquina, es decir, no es distribuido. Un esquema del ataque lo podemos observar en la figura 1: Existen tres estrategias que pueden inhabilitar un sistema:

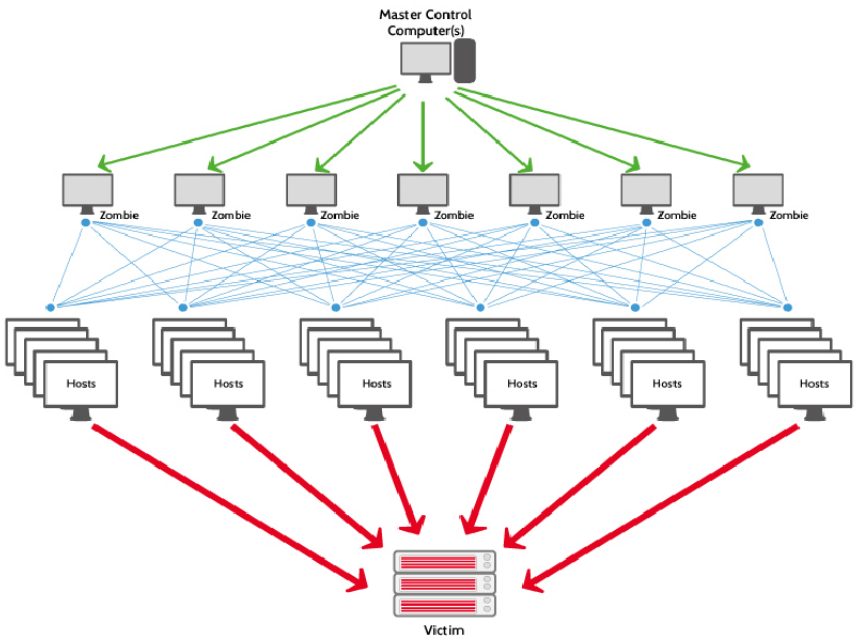

Figura 1: Ataque DDoS.

1. **Ancho de banda:** Ataque que consiste en saturar la capacidad de la red del servidor, haciendo que sea imposible llegar a él.

2. **Recursos:** Ataque que consiste en agotar los recursos del sistema de la máquina, impidiendo que esta pueda responder a las peticiones legítimas.

3. **Explotación de fallos de software:** Categoría de ataque que explota fallos en el software que inhabilitan el equipo o toman su control.

Basándonos en el modelo OSI, podemos encontrar que el ataque DDoS se puede hacer en diferentes capas. Se muestra en la imagen 2: Para detectar

Nombre del ataque	Nivel OSI	Tipo de ataque	Explicación del ataque
ICMP Echo Request Flood	L3	Recursos	También denominado Ping Flood. Envío masivo de paquetes (ping), que implican una respuesta por parte de la víctima (pong) con el mismo contenido que el paquete de origen.
IP Packet Fragment Attack	L3	Recursos	Envío de paquetes IP que remiten voluntariamente a otros paquetes que nunca se envían, saturando así la memoria de la víctima.
SMURF	L3	Ancho de banda	Ataque por saturación ICMP que usurpa la dirección de origen para redirigir las múltiples respuestas hacia la víctima.
IGMP Flood	L3	Recursos	Envío masivo de paquetes IGMP (protocolo de gestión de grupos de internet)
Ping of Death	L3	Explotación	Envío de paquetes ICMP que explotan fallos del sistema operativo
TCP SYN Flood	L4	Recursos	Envío masivo de solicitudes de conexión TCP
TCP Spoofed SYN Flood	L4	Recursos	Envío masivo de solicitudes de conexión TCP usurpando la dirección de origen
TCP SYN ACK Reflection Flood	L4	Ancho de banda	Envío masivo de solicitudes de conexión TCP a un gran número de máquinas, usurpando la dirección de origen por la dirección de la víctima. En ancho de banda de la víctima queda saturada por las respuestas a dichas peticiones.
TCP ACK Flood	L4	Recursos	Envío masivo de acuses de recibo de segmentos TCP
TCP Fragmented Attack	L4	Recursos	Envío de segmentos TCP que remiten voluntariamente a otros que nunca se envían, saturando la memoria de la víctima
UDP Flood	L4	Ancho de banda	Envío masivo de paquetes UDP (sin necesidad de establecer conexión previa)
UDP Fragment Flood	L4	Recursos	Envío de datagramas que remiten voluntariamente a otros datagramas que nunca se envían, saturando así la memoria de la víctima
Distributed DNS Amplification Attack	L7	Ancho de banda	Envío masivo de peticiones DNS usurpando la dirección de origen de la víctima hacia un gran número de servidores DNS legítimos. Como la respuesta tiene un mayor volumen que la pregunta, el ataque se amplifica
DNS Flood	L7	Recursos	Ataque de un servidor DNS mediante el envío masivo de peticiones
HTTP(S) GET/POST Flood	L7	Recursos	Ataque de un servidor web mediante el envío masivo de peticiones
DDoS DNS	L7	Recursos	Ataque de un servidor DNS mediante el envío masivo de peticiones desde un gran número de máquinas controladas por el atacante

Figura 2: Diferentes ataques en modelo OSI.

el ataque, podemos usar estudiar el flujo enviado por los routers. Se analiza ese resumen y se compara con posibles ataques anteriores. Si la comparación es positiva, se activaría el servicio de mitigación en pocos segundos. Esto se

basa en los umbrales de tráfico en "paquetes por segundo"(pps, Kpps, Mpps, Gpps) o "bytes por segundo"(bps, Kbps, Mbps, Gbps), por ejemplo. Esta mitigación puede ser un poco delicada ya que debemos tener cuidado al elegir el umbral. Hay empresas que se dedican a ofrecer este tipo de servicios, como por ejemplo OVH.

Pero si nos tenemos que encargar de la defensa de nuestra propio servidor, podemos seguir los siguientes consejos:

1. Un buen diseño de un servidor debe tener en cuenta que estos ataques se pueden producir, así que lo lógico es tener un servicio que monitorice la actividad del servidor, y cuando detecte una situación no usual, avise al administrador del servidor.

2. También es importante escoger un umbral adecuado para nuestro servidor. Es decir, deberemos estudiar la actividad cotidiana de nuestro servidor para no elegir un valor ni muy bajo (podiendo confundirse con un ataque DDoS cuando solo es una pequeña subida en la cantidad de peticiones) ni un valor muy alto (permitiendo demasiadas peticiones de un posible ataque DDoS que, aunque no llegue a echar abajo nuestro servidor, consuma gran parte de sus recursos.)

3. Además, interesa tener bien configurados todos los servicios de seguridad disponibles, como un cortafuegos, que permita el paso del tráfico únicamente que le interese a nuestro servidor. Lo ideal sería hacer uso de una DMZ (zona desmilitarizada).

4. Finalmente, hay que prestar especial atención a los puertos que tenemos abiertos. Esto es que solo debemos tener los puertos abiertos de los servicios que estemos usando.

En la referencia [11], se puede ver un ataque DDoS en tiempo real, y en la referencia [12] se pueden ver los ataques en vivo de cualquier país del mundo.

5.2. SQL Injection [8]

Consiste en la inserción de código SQL por medio de los datos de entrada desde la parte del cliente hacia la aplicación. Es decir, por medio de la inserción de este código el atacante puede modificar las consultar originales que debe realizar la aplicación y ejecutar otras totalmente distintas con la intención de acceder a la herramienta, obtener información de alguna de las tablas o borrar los datos almacenados, entre otras muchas cosas.

Como consecuencias de estos ataques y dependiendo de los privilegios que tenga el usuario de la base de datos bajo el que se ejecutan las consultas, se podría acceder no sólo a las tablas relacionadas con la aplicación, sino también a otras tablas pertenecientes a otras bases de datos alojadas en ese mismo servidor.

Lo comentado anteriormente es posible gracias a que el uso de ciertos caracteres en los campos de entrada de información por parte del usuario, ya sea mediante el uso de los campos de los formularios que son enviados al servidor mediante POST o bien por medio de los datos enviados mediante GET en las urls de las páginas web, posibilitan coordinar varias consultas SQL o ignorar el resto de la consulta, permitiendo al hacker ejecutar la consulta que elija, de ahí que sea necesario realizar un filtrado de esos datos enviados para evitar problemas.
Podemos ver un esquema del ataque en 3:

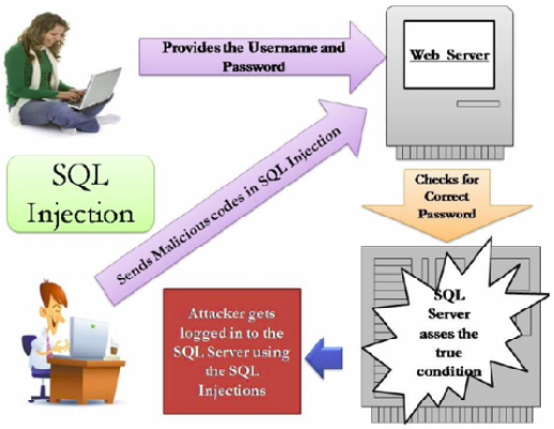

Figura 3: Cómo funciona un ataque SQL Injection.

Podemos seguir ciertos consejos para evitar los ataques o por lo menos minimizar el riesgo de los mismos:

1. Escapar los caracteres especiales utilizados en las consultas SQL: Añadir la barra invertida delante de las cadenas utilizadas en las consultas SQL para evitar que estas corrompan la consulta. Algunos de estos caracteres especiales que es aconsejable escapar son las comillas dobles o las comillas simples.
   En el caso de PHP podemos optar por la función mysql_real_scape_string(), que toma como parámetro una cadena y la modifica evitando todos los caracteres especiales, devolviéndola totalmente segura para ser ejecutada dentro de la instrucción SQL.

2. Delimitar los valores de las consultas: Aunque el valor de la consulta sea un entero, es aconsejable delimitarlo siempre entre comillas simples.

3. Verificar siempre los datos que introduce el usuario: Si en una consulta estamos a la espera de recibir un entero, no confiemos en que sea así, sino que es aconsejable tomar medidas de seguridad y realizar la comprobación de que realmente se trata del tipo de dato que estamos esperando. Para realizar esto, los lenguajes de programación ofrecen funciones que realizan esta acción, como pueden ser ctype_digit() para saber si es un número o ctype_alpha().

4. Asignar mínimos privilegios al usuario que conectará con la base de datos. El usuario que utilicemos para conectarnos a la base de datos desde nuestro código debe tener los privilegios justos para realizar las acciones que necesitemos.

5.3. Ataque MITM (Man In The Middle) [13]

Es un tipo de ataque informático en el que el atacante tiene conexiones independientes con las víctimas y trasmite mensajes entre ellos, haciéndoles creer que están hablando directamente entre sí a través de una conexión privada, cuando en realidad toda la conversación es controlada por el atacante. El atacante debe ser capaz de interceptar todos los mensajes que van entre las dos víctimas e inyectar nuevos, lo cual es sencillo en muchas circunstancias (por ejemplo: un atacante dentro del rango de recepción de un punto de acceso de una red inalámbrica Wi-Fi sin encriptar, puede insertarse cómo un hombre en el medio).

Este ataque puede tener éxito solo cuando el atacante puede hacerse pasar por cada punto final a satisfacción de la otra (esto es un ataque de autenticación mutua). La mayoría de los protocolos criptográficos incluyen alguna forma de autentificación de extremos específicamente para prevenir los ataques MITM. Por ejemplo: SSL autentifica al servidor utilizando una autoridad de certificación de confianza mutua. El esquema de este ataque se puede ver en las figuras 4 5. La posibilidad de este ataque siendo un problema potencial de seguridad muy serio, incluso para muchos cripto-sistemas basados en clave pública.
Existen varios tipos de defensa contra estos ataques MITM, estas defensas emplean técnicas de autenticación basadas en:

1. Infraestructura de claves públicas

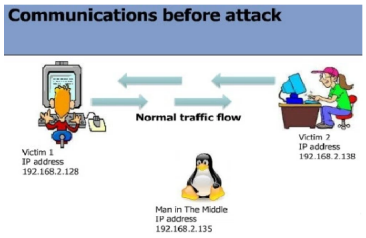

Figura 4: Cómo funciona un ataque MITM (I).

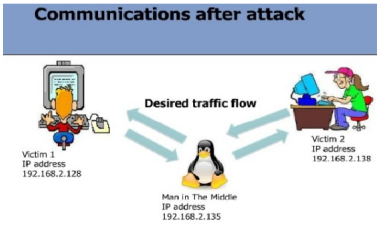

Figura 5: Cómo funciona un ataque MITM (II).

2. Autenticación mutua fuerte.

3. El examen de latencia, como con mucho los cálculos de la función hash criptográfica que conducen a decenas de segundos, si ambas partes toman normalmente 20 segundos y el cálculo de 60 segundos para llegar a cada parte, esto puede indicar a un tercero en la comunicación.

4. Un segundo canal de verificación (seguro): por ejemplo el protocolo HTTPS (SSL).

5. Pads(almohadillas) de una sola vez son inmunes a los ataques MITM, en el supuesto caso de la seguridad y la confianza de la plataforma de una sola vez.

6. Ejemplos prácticos de ataques cibernéticos

Para realizar los ataques, hemos usado el Sistema Operativo Kali Linux, el cual posee una serie de herramientas preinstaladas para la auditoría y seguridad de sistemas informáticos.

6.1. Ejemplo práctico ataque DoS

Para este ataque, hemos usado la herramienta para pentesting Metasploit. Ésta nos permite mandar muchas peticiones SYN (recordemos el triple handshaking de TCP) haciendo que el servidor reserve recursos para todas las peticiones y quedando a la espera porque nunca se enviará el *ACK*. Usamos los comandos que se muestran en la siguiente figura 6: El modulo

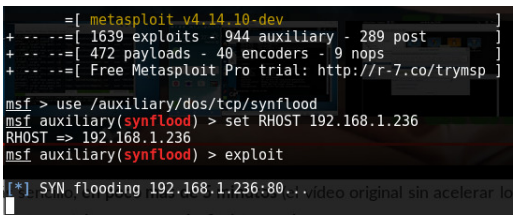

Figura 6: Comandos Metasploit.

auxiliary contiene herramientas externas como pueden ser escáners de vulnerabilidades, sniffers, etc. En este caso usaremos *synflood*. Con *set RHOST 192.168.1.236*, indicamos la ip de nuestra víctima. Por último, "explotamos" el ataque para que se haga efectivo.
El escenario del ataque es el siguiente: Hay un servidor *XAMPP*, en el sistema operativo Windows 10, cuya IP es 192.168.1.236, la cual suponemos en la misma red que nuestra máquina atacante. En la siguiente captura, podemos observar la página de inicio del servidor, pero cuando realizamos el ataque, la aplicación que nos permite configurar el servidor falla, y el servidor cae 7:

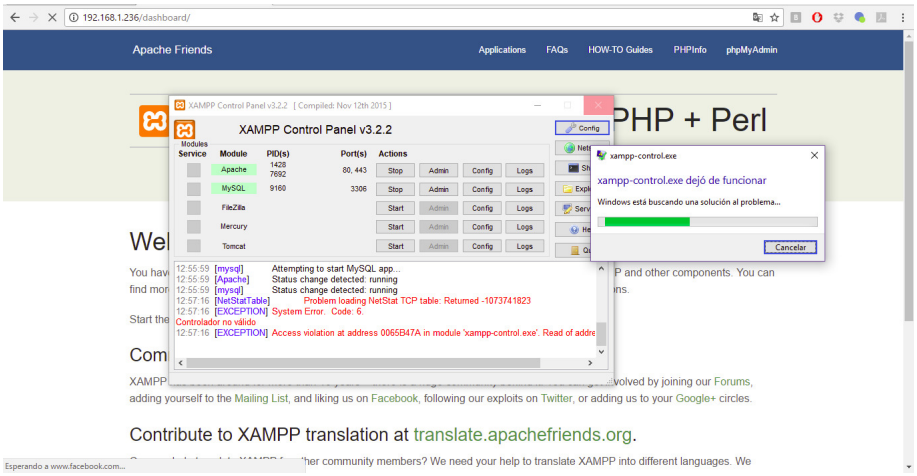

Figura 7: Servidor caído.

6.2. Ejemplo práctico SQL Injection

Para este caso, hemos usado la herramienta *SQLMap*, la cual nos permite descubrir las vulnerabilidades y obtener los datos de las base de datos. Se basa en realizar consultas a las que la base de datos puede ser vulnerable por no haberse realizado una correcta validación de las mismas.

En la referencia [14] podemos consultar una base de datos con los distintos tipos de vulnerabilidades que pueden surgir en las páginas web. En nuestro caso hemos usado 'inurl:item_id='. Si buscamos esto en Google, observamos qué páginas web pueden ser vulnerables. Escogeremos *www.dcmsee.org*. Para comprobar si realmente la página es vulnerable, entramos en el link, y al final de este ponemos una comilla simple. Si es vulnerable, saldrá un error SQL 8:

Figura 8: Error SQL.

Una vez que sabemos que la página es vulnerable, empezaremos nuestro ataque: introducimos el comando *sqlmap -u http://www.dcmsee.org/en/countries/?item_id=5 –dbs*. Este comando, nos devolverá las base de datos disponibles 9.

En nuestro caso, hay dos disponibles: *dmcseems* e *information_schema*. Intuimos que la información de los posibles usuarios estará en la primera de ellas, así que lanzamos el comando *sqlmap -u http://www.dcmsee.org/en/countries/?item_id=5 -D dmcseems –tables*. Así obtenemos las tablas de esta base de datos. Hay una tabla que nos llaman la atención: *site_users*, así que el siguiente paso será ver que hay en ella, para ello introducimos el comando *sqlmap -u http://www.dcmsee.org/en/countries/?item_id=5 -D dmcseems –tables -T site_users –columns* y el resultado es el siguiente 10: De esta tabla, podemos escoger las columnas que queramos, en nuestro caso para ver que el ataque ha sido efectivo cogeremos *password* y *username*. Para ello introducimos el comando *sqlmap -u http://www.dcmsee.org/en/countries/?item_id=5*

Figura 9: Bases de datos disponibles.

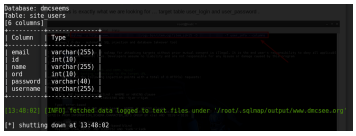

Figura 10: Tabla *site_users*.

-D dmcseems -T site_users -C username –dump para los nombres de usuario y *sqlmap -u http://www.dcmsee.org/en/countries/?item_id=5 -D dmcseems -T site_users -C username –dump* para las contraseñas. Si tenemos suerte, puede que las contraseñas no estén encriptadas. En nuestro caso, estaban encriptadas, pero el propio *SQLMAP* ofrece un algoritmo de fuerza bruta para desencriptar. Así pues, obtenemos el nombre de usuario y contraseña de alguno de ellos y lo probamos 11. Cuando le damos a *Members Section*

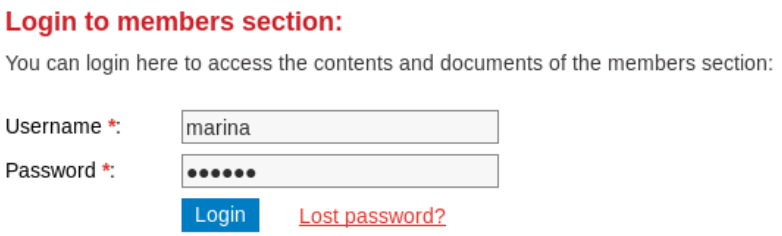

Figura 11: Comprobando contraseña.

(donde salía el menú del login) de nuevo vemos que nos hemos identificado correctamente e incluso podemos modificar la contraseña 12:

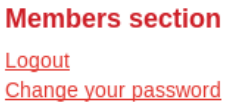

Figura 12: Comprobando contraseña.

6.3. Ejemplo práctico ataque MITM

Este ataque se puede hacer en pocos servidores, pero se pretende mostrar que el uso de las conexiones seguras como por ejemplo a través del protocolo *HTTPS* son imprescindibles para mantener nuestro servidor seguro.

En este caso, usaremos la herramienta *arpspoof* y *Wireshark*. Con la primera de ellas, lo que haremos será un ataque al protocolo ARP, el cual se encarga de traducir la ip de un dispositivo a su *MAC*. Para ello, falsearemos los paquetes IP, redirigiendo los paquetes de la víctima hacia nuestra máquina atacante, y una vez que pasa por nuestras manos, los reencaminaremos al router. También haremos esta operación en sentido inverso, es decir, el router[1] nos mandará los paquetes que se correspondan con la IP de la víctima, y nuestra máquina los reencaminará hacia la víctima. Todo el proceso lo auditaremos con *Wireshark* para obtener los datos sensibles.

Nuestro escenario es el siguiente:

1. IP víctima: 192.168.1.46, MAC víctima: 90:48:9A:3E:64:37
2. IP router: 192.168.1.1, MAC víctima: 54:67:51:59:FE:90

Así pues, el primer paso será falsear los paquetes, modificando las MAC tal y como se ha explicado anteriormente. El proceso se muestra en la siguiente figura 13. Una vez hecho esto, tendremos que esperar a que el usuario ingrese en una página con el protocolo *HTTP*. Nosotros estaremos observando con *Wireshark* la actividad del usuario. Por ejemplo, podemos obtener el usuario y la contraseña de un sitio web 14 15.

[1]Cuando se dice que el dispositivo es un router no tiene por qué ser un router, podría ser cualquier otro dispositivo con el que se comunicase la víctima, pero en este caso, lo que obtendremos será los datos de usuario de una página, así que será el router.

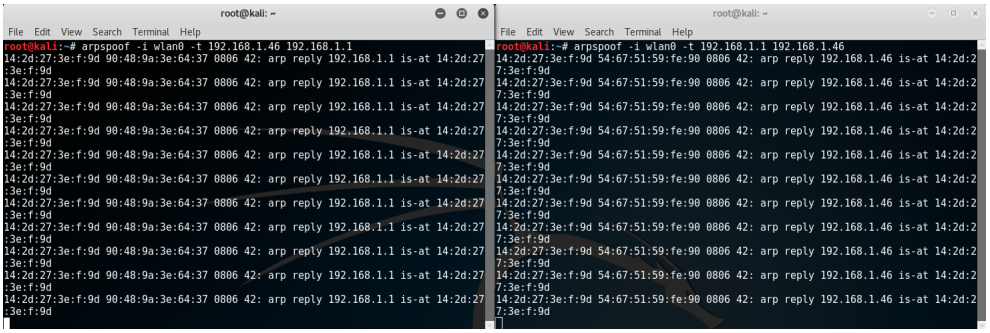

Figura 13: Comando arpspoofing.

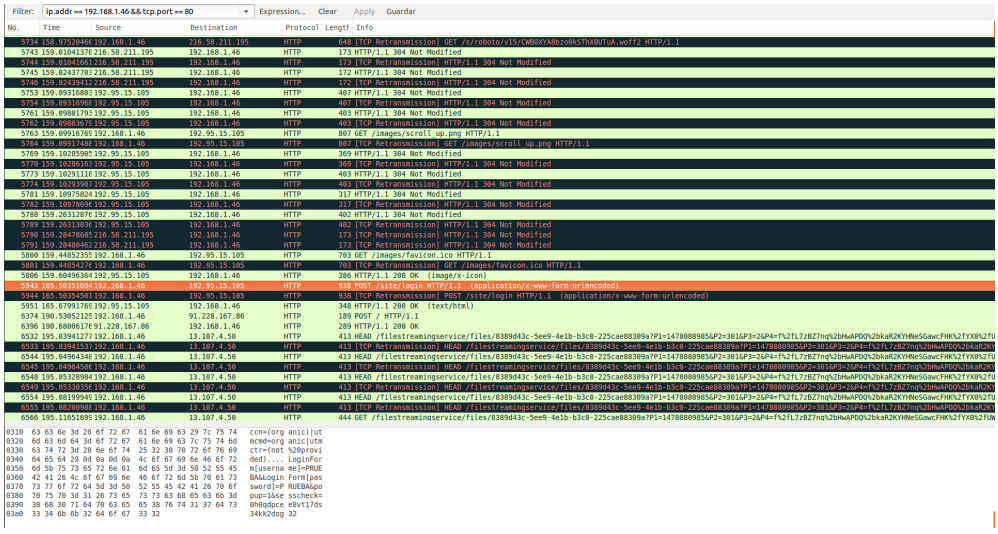

Figura 14: Captura Wireshark (I).

0310 63 63 6e 3d 28 6f 72 67 61 6e 69 63 29 7c 75 74 ccn=(org anic)|ut
0320 6d 63 6d 64 3d 6f 72 67 61 6e 69 63 7c 75 74 6d mcmd=org anic|utm
0330 63 74 72 3d 28 6e 6f 74 25 32 30 70 72 6f 76 69 ctr=(not %20provi
0340 64 65 64 29 0d 0a 0d 0a 4c 6f 67 69 6e 46 6f 72 ded).... LoginFor
0350 6d 5b 75 73 65 72 6e 61 6d 65 5d 3d 50 52 55 45 m[userna me]=PRUE
0360 42 41 26 4c 6f 67 69 6e 46 6f 72 6d 5b 70 61 73 BA&Login Form[pas
0370 73 77 6f 72 64 5d 3d 50 52 55 45 42 41 26 70 6f sword]=P RUEBA&po
0380 70 75 70 3d 31 26 73 65 73 73 63 68 65 63 6b 3d pup=1&se sscheck=
0390 30 68 30 71 64 70 63 65 65 38 76 74 31 37 64 73 0h0qdpce e8vt17ds
03a0 33 34 6b 6b 32 64 6f 67 33 32 34kk2dog 32

Figura 15: Captura Wireshark (II).

Referencias

[1] https://seguridadinformaticasmr.wikispaces.com/TEMA+1-+SEGURIDAD+IFORM%C3%81TICA.

[2] http://conexioninversa.blogspot.com.es/2009/07/redes-trampa-honeypots-de-baja.html.

[3] https://www.certsi.es/blog/honeypots.

[4] http://www.cursodehackers.com/.

[5] http://es.docs.kali.org/.

[6] https://www.ovh.es/anti-ddos/principio-anti-ddos.xml.

[7] https://www.ovh.es/anti-ddos/analisis.xml.

[8] https://pressroom.hostalia.com/white-papers/ataques-inyeccion-sql.

[9] http://definicion.de/seguridad-informatica/.

[10] http://es.ccm.net/contents/622-introduccion-a-la-seguridad-informatica.

[11] http://es.gizmodo.com/asi-se-ve-un-ataque-ddos-en-tiempo-real-482581412.

[12] https://threatmap.checkpoint.com/ThreatPortal/livemap.html.

[13] https://seguridadpcs.wordpress.com/terminologias-2/ataque-man-in-the-middle/.

[14] https://www.exploit-db.com/google-hacking-database/.