# Taxonomía de un ataque


OpenWebinars

# ÍNDICE

- Inyección sql y fuerza bruta.
- XXS, SSI.
- Phishing.
- Reto: Probar los diferentes ataques web en un entorno controlado.

# INYECCIÓN SQL Y FUERZA BRUTA

- Inyección SQL es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.
- El origen de la vulnerabilidad radica en la incorrecta comprobación o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o script que esté embebido dentro de otro.
- Se conoce como Inyección SQL, indistintamente, al tipo de vulnerabilidad, al método de infiltración, al hecho de incrustar un código SQL intruso y a la porción de código incrustado.

- Se dice que existe o se produjo una inyección SQL cuando, de alguna manera, se inserta o "inyecta" código SQL invasor dentro del código SQL programado, a fin de alterar el funcionamiento normal del programa y lograr así que se ejecute la porción de código "invasor" incrustado, en la base de datos.

- Este tipo de intrusión normalmente es de carácter malicioso, dañino o espía, por tanto es un problema de seguridad informática, y debe ser tomado en cuenta por el programador de la aplicación para poder prevenirlo. Un programa elaborado con descuido, displicencia o con ignorancia del problema, podrá resultar ser vulnerable, y la seguridad del sistema (base de datos) podrá quedar eventualmente comprometida.

# EL CÓDIGO SQL ORIGINAL Y VULNERABLE ES:

consulta:="SELECT*FROM usuarios WHERE nombre ="+nombreUsuario+"';"

- Si el operador escribe un nombre, por ejemplo "Pepe", nada anormal sucederá, la aplicación generaría una sentencia SQL similar a la siguiente, que es perfectamente correcta, en donde se seleccionarán todos los registros con el nombre "Pepe" en la base de datos:

SELECT*FROM usuarios; SELECT* FROM datos WHERE nombre LIKE%

Se generaría la siguiente consulta SQL,

```
SELECT A * FROM usuarios WHERE nombre='Alicia';

drop table USUARIOS;

SELECT * FROM datos WHERE nombre LIKE'%';
```

# INYECCIONES SQL BASADAS EN:

- **Boolean-Based Blind:** la inyección SQL ciega basada en expresiones Boleanas (verdadero o falso), nos indica que la URL responde a estas inyecciones y a su vez produce cambiar en el aplicativo web como tal.
- Por ejemplo:

www.sitioweb.com/index.php?id=3 AND 1=1//verdadero

- Esta condición es verdadera, y en el sitio web no se ve reflejado ningún tipo de cambio.

www.sitioweb.com/index.php?id=3 AND 1=0 // falso

- Esta condición es falsa y en el sitio web se ven cambios por ejemplo, desaparece un botón o se cambia de posición, etc.

- **Time-Based Blind:** como ya vimos anteriormente la inyección SQL ciega basada en tiempo hace que la base de datos sea pausada por un intervalo de tiempo.

- Ejemplo:

www.sitioweb.com/index.php?id=3 ' AND SLEEP(10)=' //MYSQL

www.sitioweb.com/index.php?id=3 WAITFOR DELAY'0:0:5' //MSSQL

- **Error-Based:** este tipo de inyecciones (basadas en error), son vulnerabilidades que nos devuelven un error de forma visible dentro del aplicativo web.

www.sitioweb.com/index.php?id=3'

warning mysql*fetch*array() expects parameter 1 to be resource boolean given in

www.sitioweb.com/index.php?id=3'

Server error in '/' Application. Undosed quotation mark before the character string 'attack;'.

- **Union Query-Based:** se basa en añadir una consulta que empiece con UNION ALL SELECT, revelando información sensible sólo si la aplicación web vuelca toda la información devuelta por la BD en la página web atacada.

www.sitioweb.com/index.php?id=3 UNION ALL SELECT 1.2.3.4.5.6.7.8 -

- **Stacked-queries:** funcional sólo en aquellos casos en los que la aplicación web permite la ejecución múltiple de consultas (separadas por ';'), y aprovecha esta funcionalidad para añadir todo tipo de consultas de ataque después de la consulta válida enviada.

www.sitioweb.com/index.php?id=3; DELETE FROM products

**Mensajes de error en relación a Motores de Base de Datos**

# MICROSOFT SQL SERVER

Server Error in '/'Application. Undosed quotation mark before the character string 'attack'.

Exception details: System. DataSqlException: Unclosed quotation mark before the character string 'attack'.

# MYSQL

Warning: mysql*fetch*array(): supplied argument is not a valid MySQL result resource in /var/www/myawesomestore.com/buystuff.php on line 12

Error. You have an error in your SQL syntax: check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 12

# ORACLE

java.sql.SQLException: ORA-00933: SQL command not properly ended at

oracle.jbdc.dbaccess.DBError.throwEqlException(DBError.java:180)at

oracle.jdbcttc7.TTloerprocessError(TTloer.java:208)

Error. SQLExceptionjava.sql.SQLException: ORA-01756:quoted string not properly terminated

# POSTGRESQL

Query failed: ERROR: unterminated quoted string at or near """

# XSS y SSI

- **Cross-site scripting** (XSS) es un tipo de vulnerabilidad informática o agujero de seguridad típico de las aplicaciones Web, que puede permitir a una tercera persona inyectar en páginas web visitadas por el usuario código JavaScript o en otro lenguaje similar (ej:VBScript). Se puede evitar usando medidas como CSP Política del mismo origen.

- Es posible encontrar una vulnerabilidad de Cross-Site Scripting en aplicaciones que tengan entre sus funciones presentar la información en un navegador web u otro contenedor de páginas web. Sin embargo, no se limita a sitios web disponibles en Internet, ya que puede haber aplicaciones locales vulnerables a XXS, o incluso el navegador en si.

- XXS es un vector de ataque que puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, subyugando la integridad del sistema. Las vulnerabilidades XXS han existido desde los primeros días de la Web.

- Esta situación es habitualmente causada al no validar correctamente los datos de entrada que son usados en cierta aplicación, o no sanear adecuadamente para su presentación como página web.

Esta vulnerabilidad puede estar presente de las siguientes formas:

- **Directa** (también llamada persistente): este tipo de XXS comúnmente filtrado, y consiste en insertar código HTML peligroso en sitios que lo permitan; incluyendo así etiquetas como *script* o *iframe*.

- **Indirecta** (también llamada reflejada): este tipo de XSS consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una cookie, o cualquier otra cabecera HTTP (en algunos navegadores y aplicaciones web, esto podría extenderse al DOM del navegador).

# XSS INDIRECTO

- Supongamos que un sitio web tiene la siguiente forma:

*http://www.example.com/home.asp?frame=menu.asp*

y que al acceder se creará un documento HTML enlazando con un frame a menu.asp

- En este ejemplo, ¿qué pasaría si se pone como URL del frame yn código javascript?

*javascript:while(1)alert(*Este mensaje saldrá indefinidamente");*

# XSS Directo (persistente):

- Funciona localizando puntos débiles en la programación de los filtros de HTML, si es que existen, para publicar contenido (como blogs, foros, etc.)

- Normalmente el atacante tratará de insertar tags como *iframe*, o *script*, pero en caso de fallar, al atacante puede tratar de poner tags que casi siempre están permitidas y es poco conocida su capacidad de ejecutar código. De esta forma el atacante podría ejecutar código malicioso.

**Ejemplos**: Una posibilidad es usar atributos que permiten ejecutar código.

BR STYLE='behavior: url(http://yoursite/xss.htc);"

DIV STYLE='background-image: url(javascript:alert('XSS))'

IMG SRC=X ONERROR="alert(/XSS/)'

# SERVER SIDE INCLUDES

- **Server Side Includes** (SSI) es un conjunto de directivas que se escriben en las páginas HTML y que se evalúan en el servidor web cuando se solicita la página HTML. SSI permite añadir contenido generado de forma dinámica a las páginas web, sin tener que programar toda la página mediante CGI, ASP, PHP o alguna tecnología similar.

- El SSI no se encuentra estandarizado por ningún organismo, así que casa desarrollador de software de servidores web es libre de incluir e interpretar estas directivas como mejor le parezca. Por tanto, lo más recomendable es consultar la documentación del servidor web para averiguar qué directivas reconoce y con qué sintaxis.

| Directiva | Parámetros | Descripción | Ejemplo |
|---|---|---|---|
| include | file, direct o virtual | Esta es probablemente la directiva más empleada, ya que permite incluir en un documento el contenido de otro documento. El parámetro file o virtual indica el archivo (HTML page, text file, script, etc.) que se desea incluir. El parámetro file indica que la ruta del archivo a incluir es relativa a la ruta del documento actual; el parámetro virtual indica que la ruta del archivo a incluir es relativa a la raíz de la ruta del documento actual. | `<!--#include virtual="header.html" -->` o `<!--#include file="footer.html" -->` |
| exec | cgi o cmd | Esta directiva ejecuta un programa, script o comando del sistema operativo. | `<!--#exec cgi="/cgi-bin/foo.cgi" -->` o `<!--#exec cmd="ls -l" -->` |
| echo | var | Esta directiva muestra el contenido de la variable de entorno especificada, como por ejemplo HTTP_USER_AGENT, LAST_MODIFIED y HTTP_ACCEPT. | `<!--#echo var="REMOTE_ADDR" -->` |
| config | timefmt, sizefmt o errmsg | Esta directiva configura el formato de visualización de las fechas, de las horas, del tamaño de los ficheros y de los mensajes de error (devueltos cuando una directiva SSI falla). | `<!--#config timefmt="%y %m %d" -->` o `<!--#config sizefmt="bytes" -->` o `<!--#config errmsg="<b>El comando SSI ha fallado</b>" -->` |
| flastmod | file o virtual | Esta directiva muestra la fecha cuando el documento especificado fue modificado por última vez. | `<!--#flastmod virtual="index.html" -->` |
| fsize | file o virtual | Esta directiva muestra el tamaño del documento. | `<!--#fsize file="script.pl" -->` |
| printenv | | Esta directiva muestra una lista de todas las variables de entorno con sus respectivos valores. | `<!--#printenv -->` |

# PHISHING

- **Phishing**, conocido como suplantación de identidad, es un término informático que denomina un modelo de abuso informático y que se comete mediante el uso de un tipo de ingeniería social.

- Caracterizado por intentar adquirir información confidencial de forma fraudulenta (como puede ser una contraseña, información detallada sobre tarjetas de crédito u otra información bancaria)

- El cibercriminal, conocido como phisher, se hace pasar por una persona o empresa de confianza en una aparente comunicación oficial electrónica, por lo común un correo electrónico, o algún sistema de mensajería instantánea o incluso utilizando también llamadas telefónicas.

- Dado el creciente número de denuncias de incidentes relacionados con el phishing o pharming, se requieren métodos adicionales de protección. Se han realizado intentos con leyes que castigan la práctica y campañas para prevenir a los usuarios con la aplicación de medidas técnicas a los programas. Se considera phishing también, la lectura por parte de terceras personas, de las letras y números que se marcan en el teclado de un ordenador o computadora.

- La mayoría de los métodos de phishing utilizan la manipulación en el diseño del correo electrónico para lograr que un enlace parezca una ruta legítima de la organización por la cual se hace pasar el impostor.

- URLs manipuladas, o el uso de subdominios, son trucos comúnmente usados por phishers; por ejemplo en esta URL: *http://www.nombresuplantado/ejemplo, en la cual el texto mostrado en pantalla no corresponde con la dirección real a la cual conduce.

- Por ejemplo, el enlace:

http://www.google.com@members.xxx.com

(y al intentar entrar con el nombre de usuario de www.google.com, si no existe tal usuario, la página abrirá normalmente).

- Éste método ha sido erradicado desde entonces en los navegadores de Mozilla e Internet Explorer.
- Otros intentos de phishing utilizan comandos en JavaScript para alterar la barra de direcciones.
- Esto se hace poniendo una imagen de la URL de la entidad legítima sobre la barra de direcciones, o cerrando la barra de direcciones original y abriendo una nueva que contiene la URL ilegítima.

# FASES

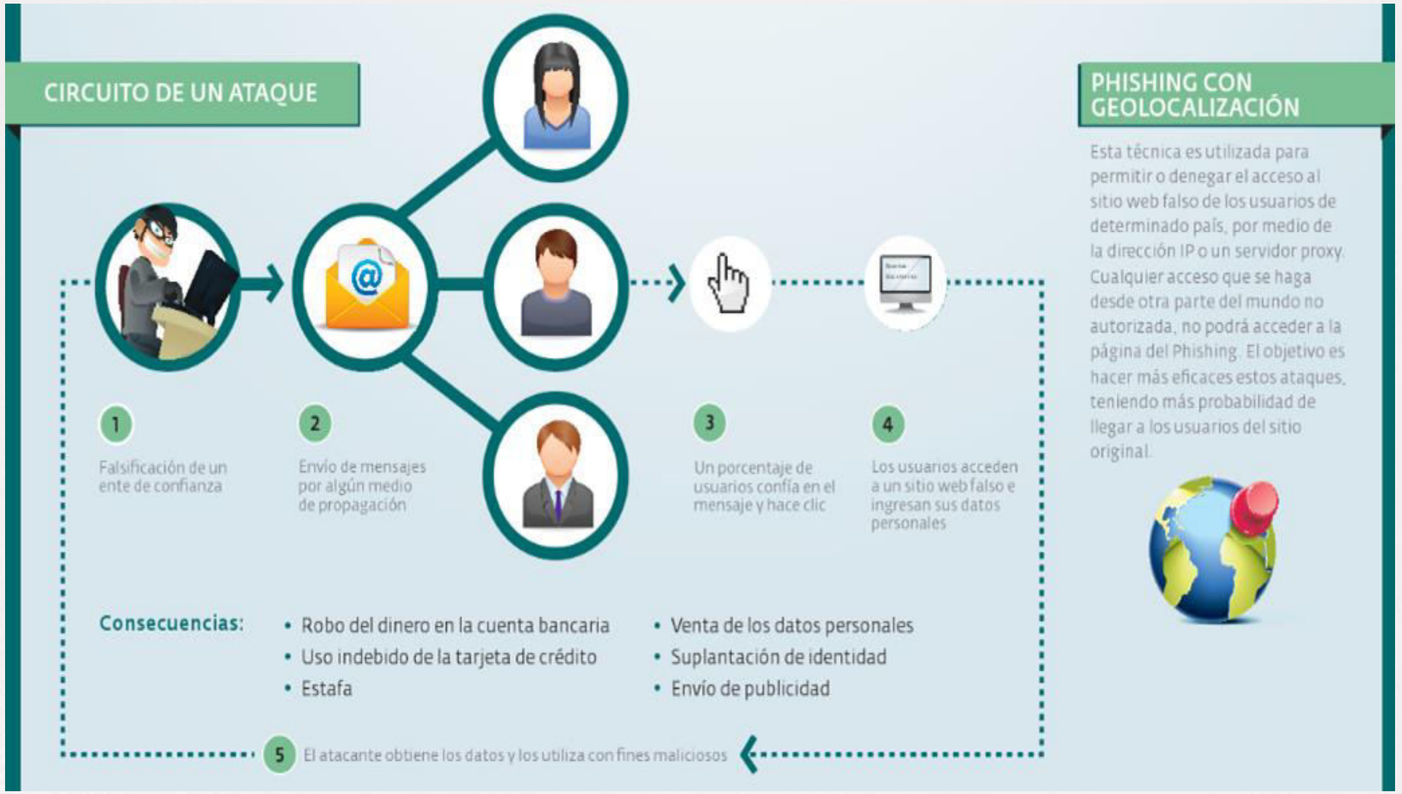
- En la primera fase, la red de estafadores se nutre de usuarios de chat, foros o correos electrónicos, a través de mensajes de ofertas de empleo con una gran rentabilidad o disposición de dinero (**hoax o scam**).

- En el caso de que caigan en la trampa, los presuntos intermediarios de la estafa, deben rellenar determinados campos, tales como: datos personales y cuenta bancaria.

- Se comete el *phishing*, ya sea el envío global de millones de correos electrónicos bajo la apariencia de entidades bancarias, solicitando las claves de la cuenta bancaria (phishing) o con ataques específicos.

- El tercer paso consiste en que los estafadores comienzan a retirar sumas importantes de dinero, las cuales son transmitidas a las cuentas de los intermediarios (muleros).

- Los intermediarios realizan el traspaso a las cuentas de los estafadores, llevándose éstas las cantidades de dinero y aquellos -los intermediarios-el porcentaje de la comisión.



# RECONOCER UN ATAQUE DE PHISHING

- Distinguir un mensaje de phishing de otro legítimo puede no resultar fácil para un usuario que haya recibido un correo de tales características, especialmente cuando es efectivamente cliente de la entidad financiera de la que supuestamente proviene del mensaje.

- El campo de: el mensaje muestra una dirección de la compañía en cuestión. No obstante, es sencillo para el estafador modificar la dirección de origen que se muestra en cualquier cliente de correo.

- El mensaje de correo electrónico presenta logotipos o imágenes que han sido recogidos del sitio web real al que el mensaje fraudulento hace referencia.

- El enlace que se muestra parece apuntar al sitio web original de la compañía, pero en realidad lleva a una página web fraudulenta, en la que se solicitarán datos de usuarios, contraseñas, etc.

- Normalmente estos mensajes de correo electrónico presentan errores gramaticales o palabras cambiadas, que no son usuales en las comunicaciones de la entidad por la que se están intentando hacer pasar.

# CONSEJOS PARA PROTEGERSE DEL PHISHING

- La regla de oro, nunca le entregue sus datos por correo electrónico. Las empresas y bancos jamás le solicitarán sus datos financieros o de sus tarjetas de crédito por correo.
- Si duda de la veracidad del correo electrónico, jamás haga clic en un link incluido en el mismo.
- Si aún desea ingresar, no haga clic en el enlace. Escriba la dirección en la barra de su navegador.
- Si duda de su veracidad, llame o concurra a su banco y verifique los hechos.

- Si recibe un email de este tipo de phishing, ignórelo y jamás lo responda.
- Compruebe que la página web en la que ha entrado es una dirección segura, ha de empezar con: *https://*, y un pequeño candado cerrado debe aparecer en la barra de estado de nuestro navegador.
- Cerciórese de siempre escribir correctamente la dirección del sitio web que desea visitar ya que existen cientos de intentos de engaños de las páginas más populares con solo una o dos letras de diferencia.
- Si sospecha que fue víctima del phishing, cambie inmediatamente todas sus contraseñas y póngase en contacto con la empresa o entidad financiera para informarles.

# RETO: Taxonomía de un ataque

- En el presente reto se propone experimentar y probar los diferentes ataques explicados en el módulo, sobre un entorno controlado.
- Seguiremos los siguientes pasos:
  - Instalación del entorno web vulnerable DVWA.
  - Pruebas de SQL injection.
  - Pruebas de XSS reflejado.