citmatel
Tecnologías de la Información y Servicios Telemáticos

Lic. Yaremis Miranda Soria

UEB. Operaciones de Internet

SQL Injection: una técnica para atacar a sitios web

Introducción

- En los últimos dos años los ataques contra las aplicaciones web han requerido de mayor atención por parte de los profesionales de la seguridad.

- No importa cuan fuerte sea la seguridad de un host, si los desarrolladores de aplicaciones web no siguen prácticas seguras de código, pues los hackers se introducirán en el sistema a través del puerto 80.

- Las dos principales técnicas de ataques más usadas son SQL Injection y Cross Site Scripting. Hablaremos de la primera que ya atacó a un sitio web cubano.

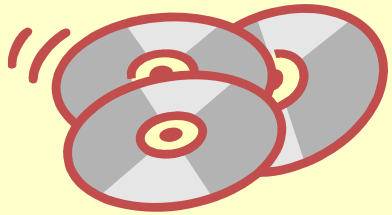
Qué es SQL Injection?

- Es un conocido problema de seguridad que afecta a las aplicaciones web dinámicas que usan bases de datos.
- Una inyección SQL sucede cuando se inserta un código SQL "invasor" dentro de otro código SQL para alterar su funcionamiento normal, y hacer que se ejecute maliciosamente el código "invasor" en la base de datos.
- Mediante la explotación de este fallo es posible tomar control sobre una base de datos e incluso sobre el sistema operativo.

Principales aplicaciones afectadas

- Bases de Datos

  - SQL Server
  - Access
  - MySQL
  - Oracle



- Scripts

  - ASP
  - PHP

SQL Injection en ASP

Es común en una aplicación ASP usar lo que el usuario pone en un formulario para montar una consulta SQL, contra una base de datos. Una típica página de validación de usuario y contraseña sería:

```
<%
  usuario=request.form("usuario")
  pass=request.form("pass")

  sql="SELECT * FROM   usuarios  WHERE
  user='  " & usuario & " ' and  pw=' " & pass & " ' "
    .... y abrir un recordset sobre ese SQL  para comprobar si existe el usuario tecleado
%>
```

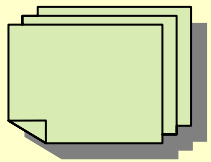
El problema es que pasamos al motor de SQL la cadena que tecleó el usuario directamente.

usuario: " a' or true "
pass : " a' or true "

La consulta resultado sería:

```
SELECT * FROM usuarios WHERE user='a' or true and pw='a' or true
```

Y esto ya cambia el sentido de la consulta original, pues con esta simple entrada tecleada se recupera la tabla **usuarios** completa.



Cómo lo resolvemos en este ejemplo?

Evitando que al formar la cadena SQL esta tenga un sentido inesperado, pues hay que procesar los parámetros que el usuario introduce antes de insertarlo en la consulta.

```
<%usuario=request.form("usuario")
pass=request.form("pass")
usuario=limpia(usuario)
pass=limpia(pass)
sql="select * from usuarios where user=' " & usuario & " ' and pw=' " & pass & " ' “
....
funcion limpia(t)
dim tt
tt=t
tt=replace(tt," ' ","")
limpia=tt
end funcion
%>
```

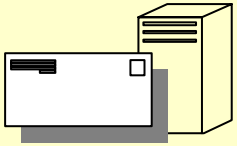
Otra modalidad en ASP con SQL Server

Imaginemos un sitio http://www.con-problemas.com

http://www.con-problemas.com/descripcion.asp?ID=999

El código interno de la página ASP será similar a este:

SELECT * FROM Libros WHERE Codigo = ID

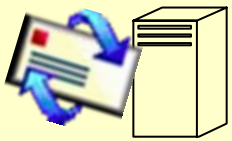


SELECT * FROM Libros WHERE Codigo = 999

La base de datos devolverá todos los campos del registro de la tabla libro que tiene el código 999.

Si modificamos la URL del navegador de esta manera:

http://www.conproblemas.com/descripcion.asp?ID=999;DELETE * FROM Libros



SELECT * FROM Libros WHERE Codigo = 999;DELETE * FROM Libros

Sql Injection en PHP con mySQL

Considere la siguiente consulta:

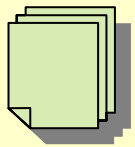
```
$result=mysql_query( ' SELECT * FROM users WHERE username=" ' . $_GET['username'] . ' " ' );
```

Selecciona todas las filas de la tabla users donde el username es igual a lo que se entró en el query string. Las comillas en $_GET['username'] no están escapadas.

$_GET['username'] = " OR 1 OR username = "

SELECT * FROM users WHERE username = "" OR 1 OR username = ""

Esto finalmente selecciona todas las filas de la tabla **users**



';DELETE * FROM forum WHERE title !='

SELECT * FROM users WHERE username = '';DELETE * FROM forum WHERE title != ''

Solución para este ejemplo

Se validan los datos antes de procesarlo por reconocimiento de patrones

```
if  ( preg_match("/^\w{8,20}$/", $_GET['username'], $matches) )
    $result = mysql_query("SELECT * FROM users WHERE username=$matches[0]");
 else // no se hace la consulta a la base de datos.
    echo "username inválido";
```

Otra forma es usar la función que PHP provee para tratar las cadenas

mysql_real_escape_string(string unescaped_string[, resource link_identifier])

que escapa todos los caracteres potencialmente peligrosos en la cadena.

```
$result=mysql_query( ' SELECT * FROM users
WHERE username=" '.mysql_real_escape_string($_GET['username']).' " ' );
```

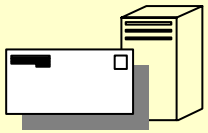
';DELETE * FROM forum WHERE title !='

SELECT * FROM users WHERE
username = ' **\';DELETE * FROM forum WHERE title != \'** '

Formas de detectar Inyección SQL

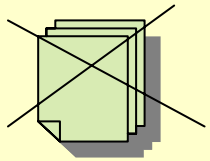
Para saber si una página es vulnerable a la inyección de código

http://www.conproblemas.com/descripcion.asp?ID='999



SELECT * FROM Libros WHERE Codigo = ' 999

Esto devolverá error y el hacker entrará en acción

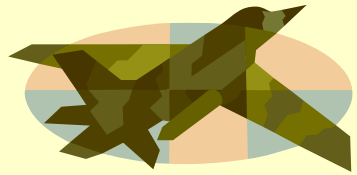


Otra manera, es usar expresiones regulares para detectar meta caracteres SQL, por ejemplo, la siguiente:

/ (\%27) | (\') | \-\-) | (\%23) | (#) / ix

Expresiones como estas se adicionan a reglas del conocido IDS Snort y permite alertarnos de los intentos de ataques a través de esta técnica.

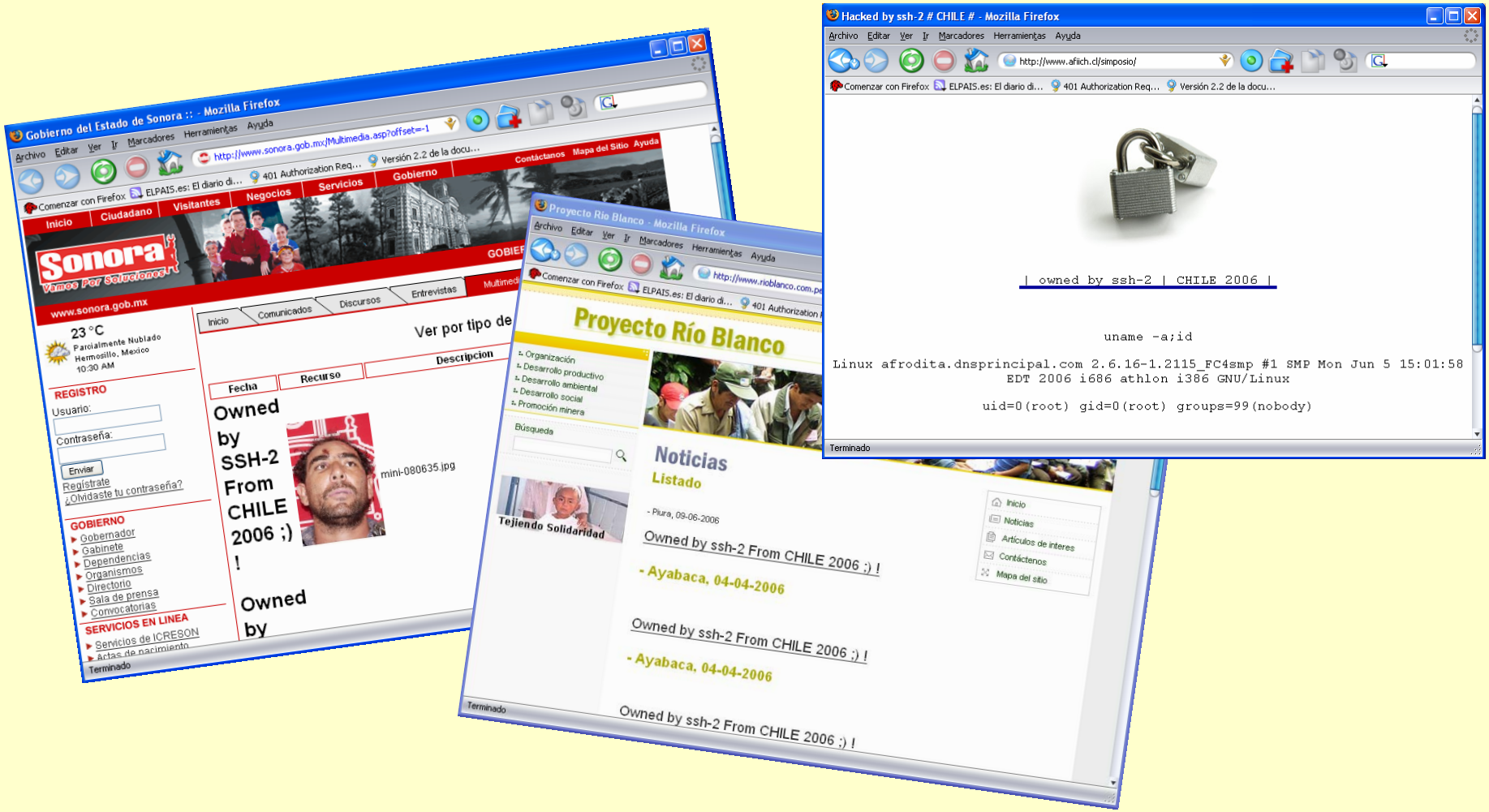
Peligros de SQL Injection



Si una aplicación admite SQL Injection

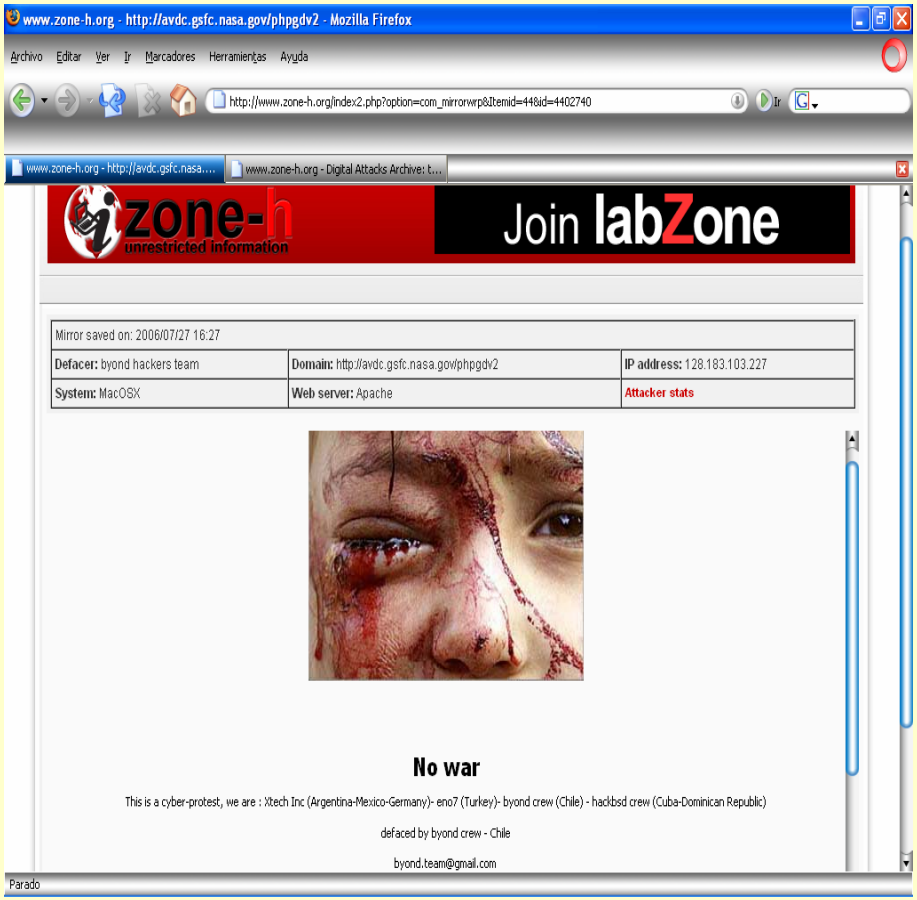
- Se tiene control sobre la base de datos y es posible realizar consultas, modificaciones, eliminaciones o inserciones de datos.
- Por ejemplo, se podría modificar el precio de los artículos, el texto de las noticias que aparecen en la web, acceder a números de tarjetas de crédito, etc.
- Además, el que la aplicación o web admita SQL Injection, en la mayoría de los países suele ser un incumplimiento grave de las leyes de protección de datos.

Imágenes de este tipo de ataque



SSH2 - Byond Hackers Crews - Chile

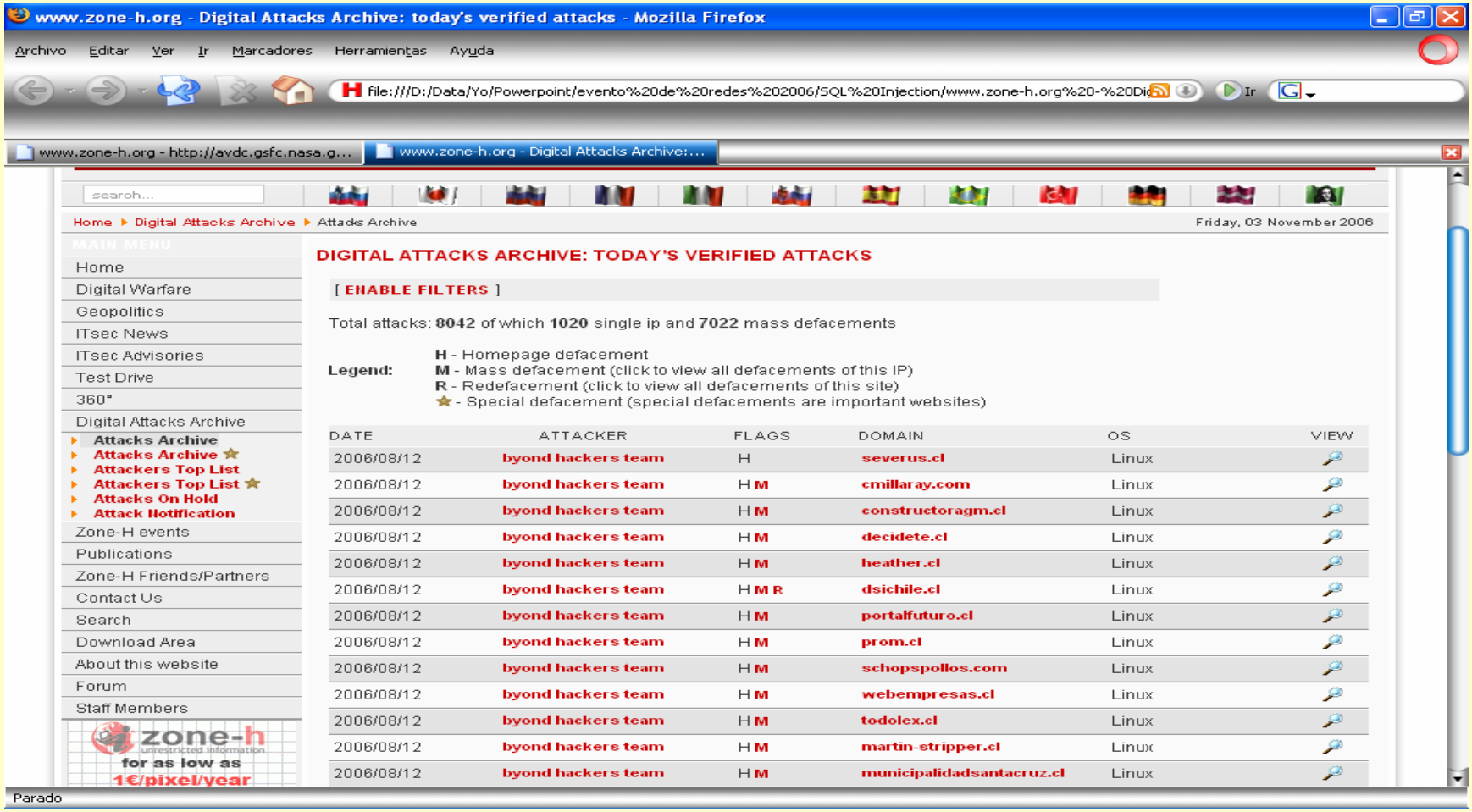
Es un grupo de hackers chilenos, que han atacado a numerosos sitios web.



Obtuvieron los nombres de usuario y claves de dos sitios web de la NASA usando Inyección SQL en las páginas web que permitía la administración, y les permitió reemplazar la portada por esa.

Esto fue parte de un ataque masivo al sitio de la Universidad de Berkeley, Microsoft y sitios del Gobierno de EE.UU. como protesta en rechazo de la guerra de Israel contra Líbano, donde ha muerto mucha gente inocente.

Estadísticas de ataques de ese grupo



Conclusiones

- La solución al SQL Injection consiste en una fuerte verificación de todos los parámetros recogidos por formulario (o querystring) que vayan a formar parte de una consulta SQL.

- Hay que mantener actualizada la base de datos y limitar el acceso sólo a las partes necesarias.

- Es preciso estar al día en materia de seguridad informática para controlar las nuevas técnicas de SQL Injection que surgen continuamente.

PREGUNTAS