Universidad Nacional Autónoma de México

Facultad de Ciencias

Criptografía y Seguridad

Proyecto 03: Reporte

Sebastián Alamina Ramírez - 318685496

Carlos Alberto Desiderio Castillo - 312183839

Camila Alexandra Cruz Miranda - 316084707

Trabajo presentado como parte del curso de **Criptografía y Seguridad**, impartido por el profesor **Manuel Díaz Díaz** durante el semestre 2023-1 en la Facultad de Ciencias, UNAM.

Fecha de entrega: **Jueves 8 de Diciembre del 2022**.

Especificación del reporte: En clase vimos las técnicas de **XSS** (Cross Site Scripting) y **SQLinjection**. Utilizando el código visto en clase para **XSS** o la herramienta **sqlmap**, elaborar un breve reporte escrito con capturas de pantalla referente a estas herramientas.

Nota: Basta con documentar una técnica de las mencionadas, ustedes deciden cual eligen.

1. Para **SQLinjection**, utilizar la herramienta **sqlmap**. Con base en la documentación de **sqlmap** y el análisis de los comandos vistos en clase, del sitio **http://testphp.vulnweb.com/**:

   (a) Utilizar el parámetro **cat** en la url para detectar fallas y poder usar **sqlmap**.
   Usamos la linea **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1** , para detectar vulnerabilidades en el enlace con el parametro cat.

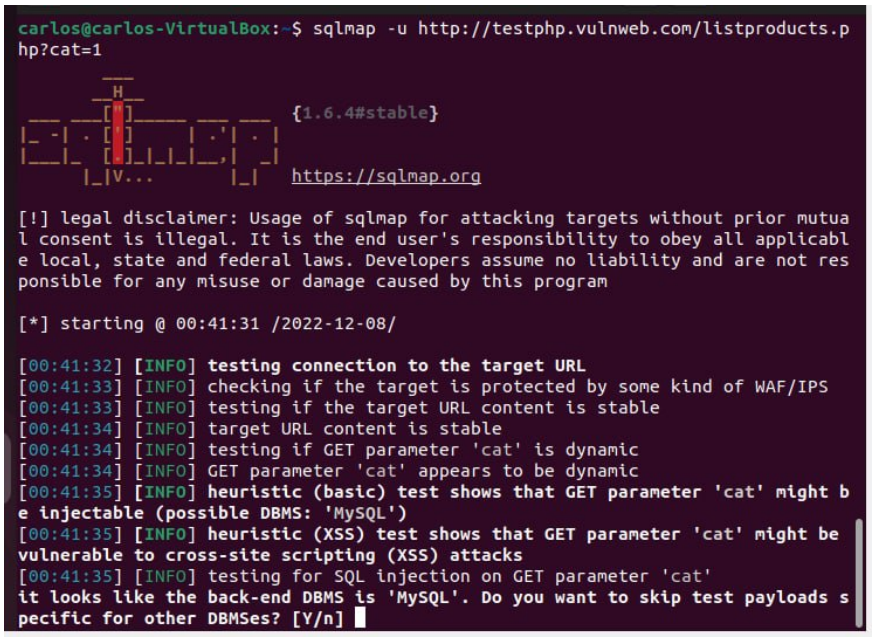

Figure 1: Comando `sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1`

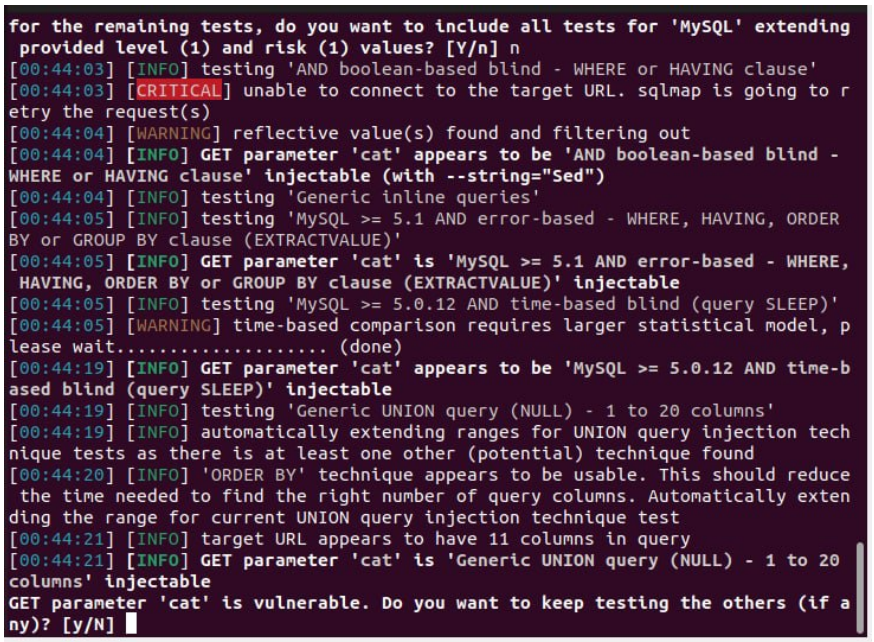

Figure 2: `Se muestra que el parámetro cat es vulnerable.`

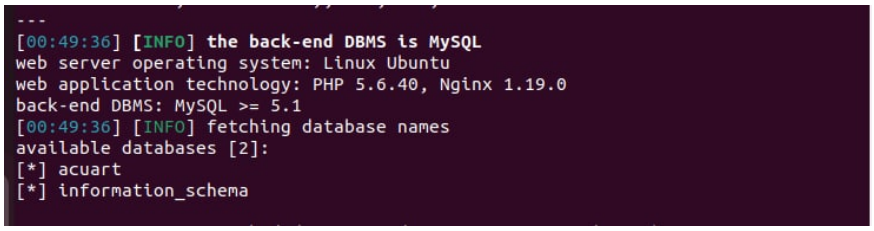

Figure 3: `Información sobre las tecnologías usadas en el desarrollo de la pagina web`

Encontramos información sobre el sistema operativo del servidor, el sistema manejador de base de datos, además de que se usó PHP y Nginx para el desarrollo de la aplicación web, con la respectiva versión de cada tecnología usada. Por último las bases de datos disponibles con las que cuenta el sistema.

(b) Obtener tablas de la base de datos: **information_schema**.
Usamos el mismo comando agregando los parametros **-D information_schema** –**tables**, para acceder a la base de datos **information_schema** y obtener las tablas de dicha base de datos.
Y como resultado obtenemos que la base de datos tiene 79 tablas en total.

Figure 4: `sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D information_schema --tables`

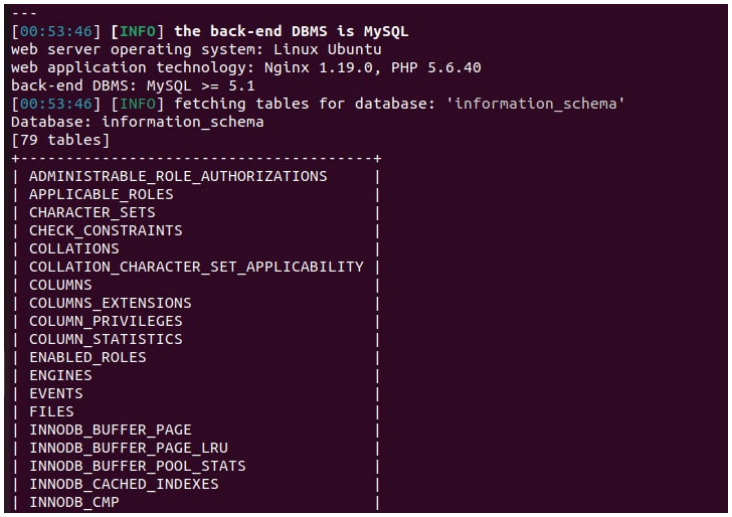

Figure 5: Tablas de `information_schema`

Figure 6: Tablas de `information_schema`

(c) Obtener el nombre de las columnas de la tabla: **KEYWORDS**.
Ahora, del comando antes usado, quitamos el parámetro –**tables**, y agregamos los parámetros: **-T KEYWORDS** –**columns**, para acceder a la tabla **KEYWORDS** y obtener las columnas de dicha tabla, en donde encontramos que solo tiene las columnas **RESERVED** y **WORD**, con sus respectivos tipos de datos en cada columna.
Entonces el comando usado es: **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D information_schema -T KEYWORDS** –**columns**.

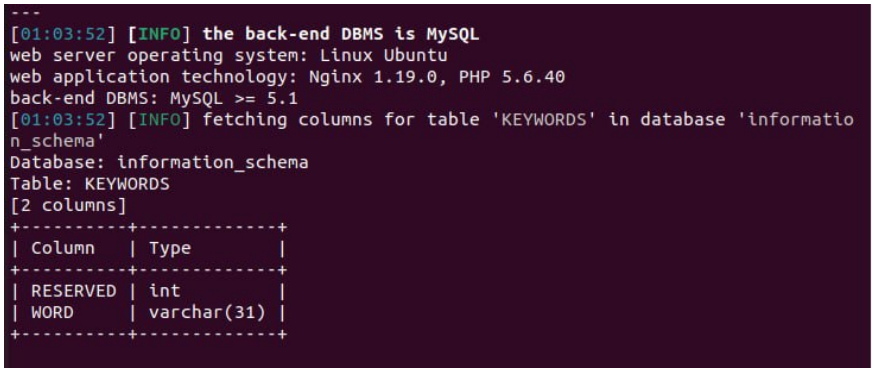

Figure 7: Información de la tabla `KEYWORDS`

(d) Obtener los datos de las columnas: **RESERVED** y **WORD**.
Primero se obtienen los datos en la columna **RESERVED**, quitando del comando anterior el parámetro –**columns** y agregando los parámetros **-C RESERVED** –**dump**, para acceder a la columna **RESERVED** y obtener los datos almacenados en dicha columna. El comando usado fue:
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D information_schema -T KEYWORDS -C RESERVED –**dump**.

Figure 8: Datos de la columna `RESERVED`

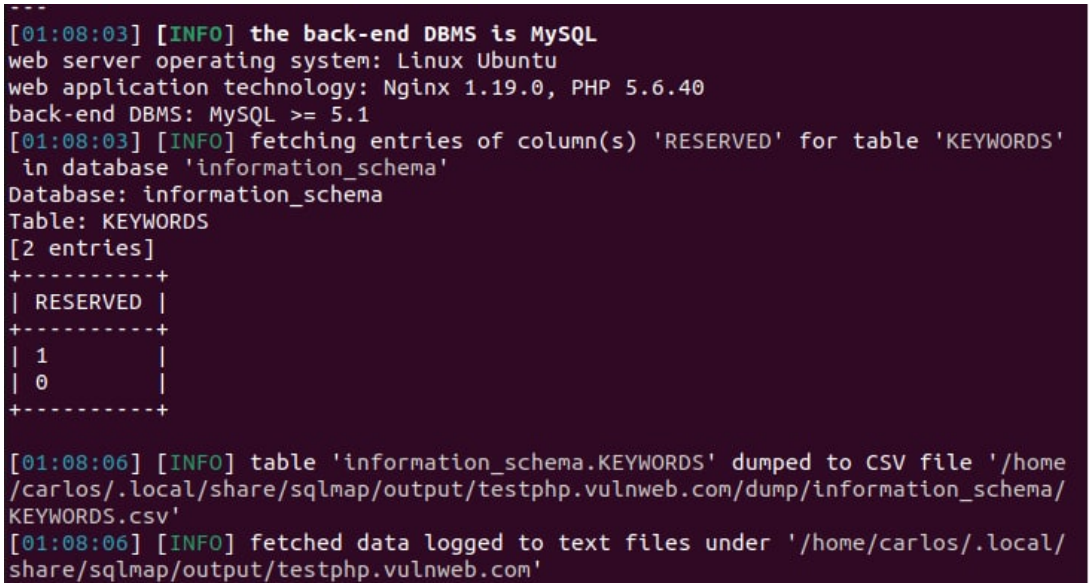

Figure 9: Datos de la columna `RESERVED`

Por último, de igual manera para obtener los valores de los datos almacenados en la columna **WORD**, remplazamos el valor del parámetro **-C RESERVED** por **-C WORD**, y así obtener sus valores. El comando usado fue:
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D information_schema -T KEYWORDS -C WORD –**dump**.

Figure 10: Datos de la columna `WORD`

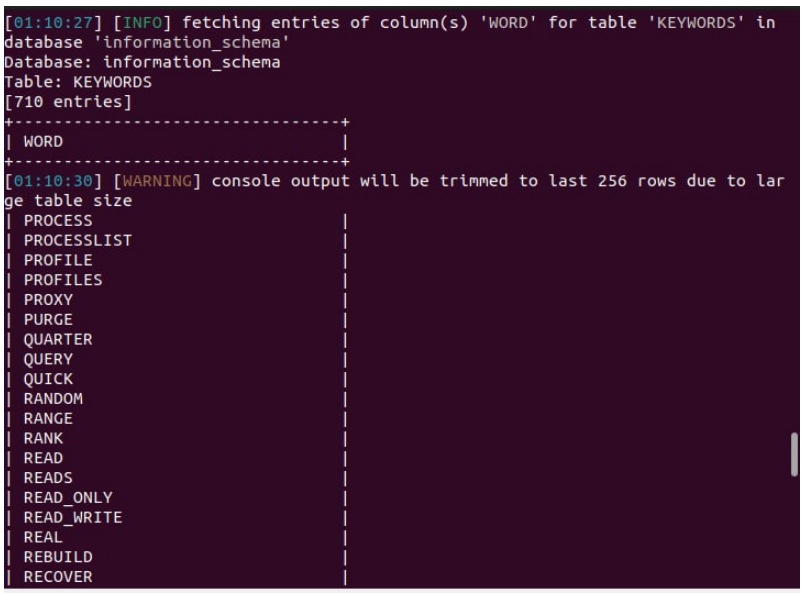

Figure 11: Datos de la columna `WORD`

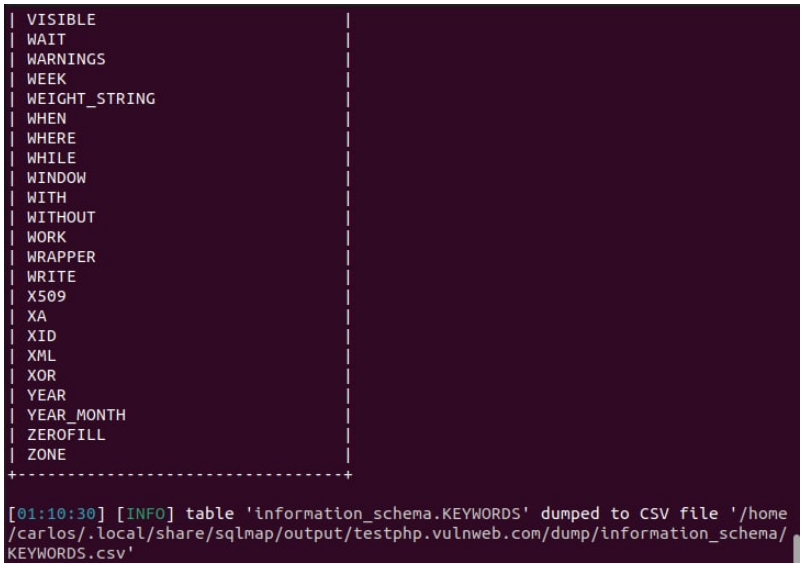

Figure 12: Datos de la columna `WORD`