МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Базы данных»
Тема: Тестирование БД на безопасность

Студент гр. 1303	____________________	Бутыло Е.А.
Преподаватель	____________________	Заславский М.М.

Санкт-Петербург
2023

Цель работы.

Сделать простой web-сервер и проверить реализованное API на безопасность.

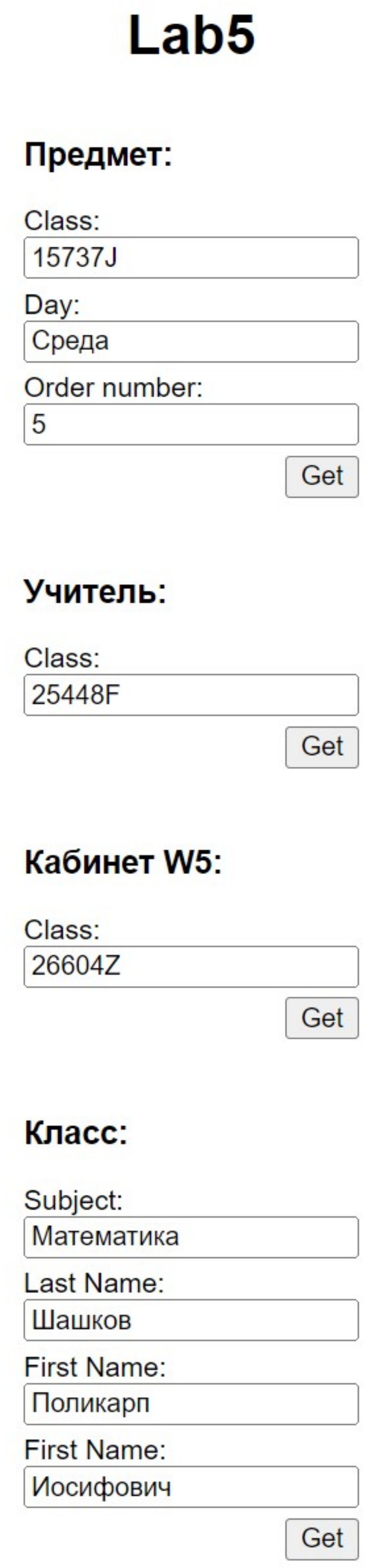
Текст задания

<u>Вариант 3</u>

1. Сделать простой web-сервер для выполнения запросов из ЛР3, например с (express.js). Не обязательно делать авторизацию и т.п., хватит одного эндпоинта на каждый запрос, с параметрами запроса как query parameters.
2. Намеренно сделайте несколько (2-3) запроса, подверженных SQL-инъекциям
3. Проверьте Ваше API с помощью sqlmap (или чего-то аналогичного), передав эндпоинты в качестве целей атаки. Посмотрите, какие уязвимости он нашёл (и не нашёл), опишите пути к исправлению.

Выполнение работы

Для выполнения работы были использованы модели из лабораторной работы №3. Реализовано API для доступа к БД.



Lab5

Предмет:

Class: 15737J

Day: Среда

Order number: 5

Get

Учитель:

Class: 25448F

Get

Кабинет W5:

Class: 26604Z

Get

Класс:

Subject: Математика

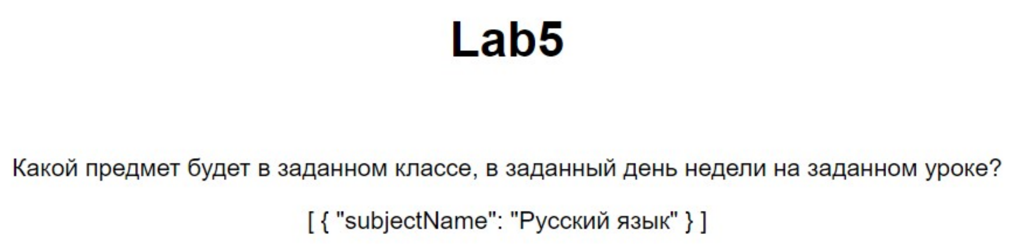
Last Name: Шашков

First Name: Поликарп

First Name: Иосифович

Get

Рисунок 1. – Реализованный web-сервер для выполнения запросов.

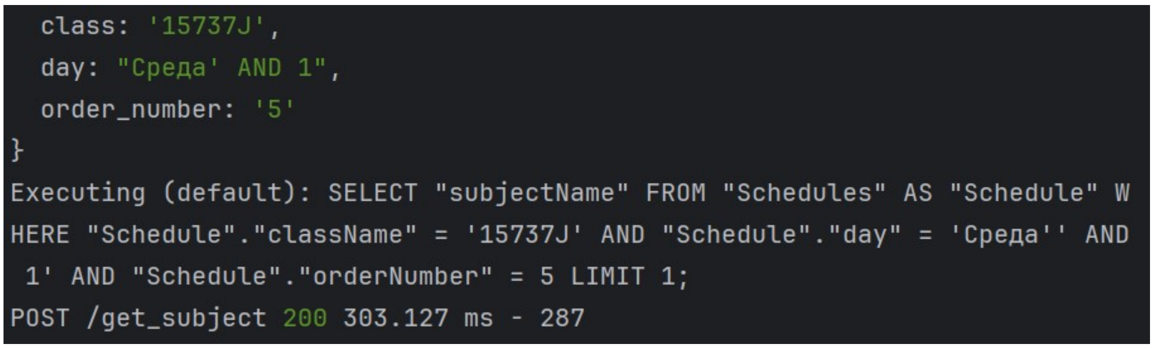

Lab5

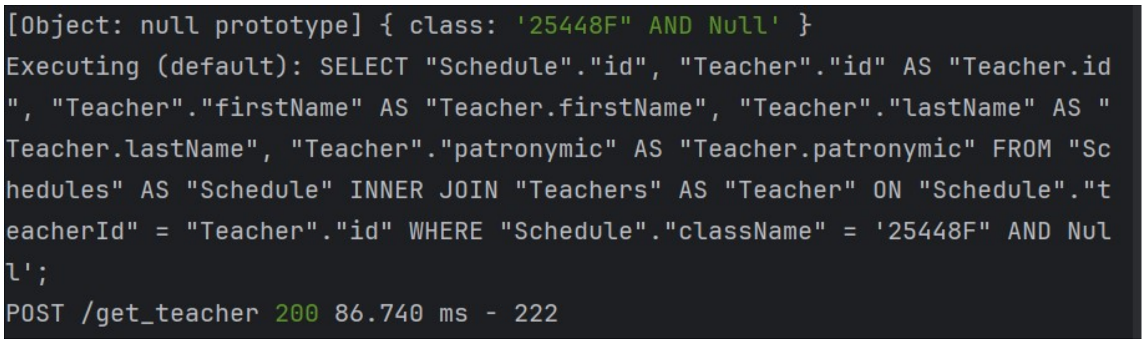
Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

[{ "subjectName": "Русский язык" }]

Рисунок 2. – Пример ответа web-сервера.

Намеренно сделано несколько запросов, подверженных SQL-инъекциям:

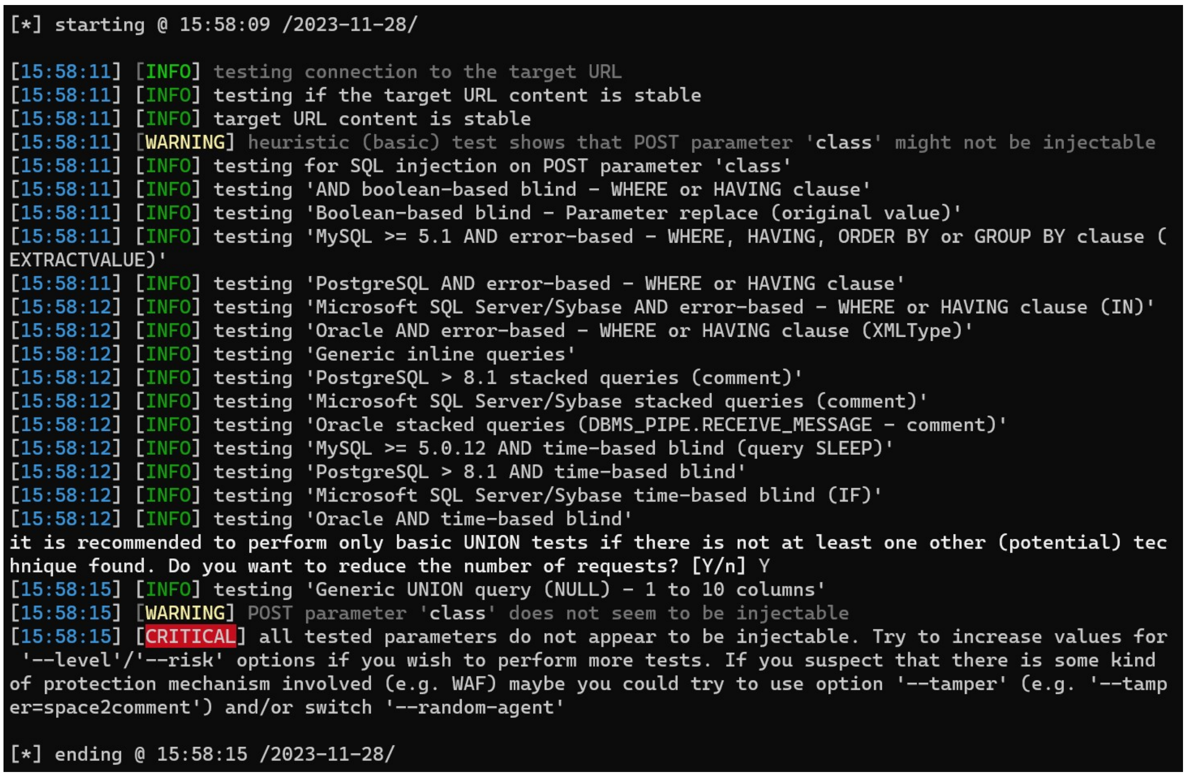

```
  class: '15737J',
  day: "Среда' AND 1",
  order_number: '5'
}
Executing (default): SELECT "subjectName" FROM "Schedules" AS "Schedule" W
HERE "Schedule"."className" = '15737J' AND "Schedule"."day" = 'Среда'' AND
 1' AND "Schedule"."orderNumber" = 5 LIMIT 1;
POST /get_subject 200 303.127 ms - 287
```


```
[Object: null prototype] { class: '25448F" AND Null' }
Executing (default): SELECT "Schedule"."id", "Teacher"."id" AS "Teacher.id
", "Teacher"."firstName" AS "Teacher.firstName", "Teacher"."lastName" AS "
Teacher.lastName", "Teacher"."patronymic" AS "Teacher.patronymic" FROM "Sc
hedules" AS "Schedule" INNER JOIN "Teachers" AS "Teacher" ON "Schedule"."t
eacherId" = "Teacher"."id" WHERE "Schedule"."className" = '25448F" AND Nul
l';
POST /get_teacher 200 86.740 ms - 222
```

Как видим никаких особых результатов запросы не принесли.

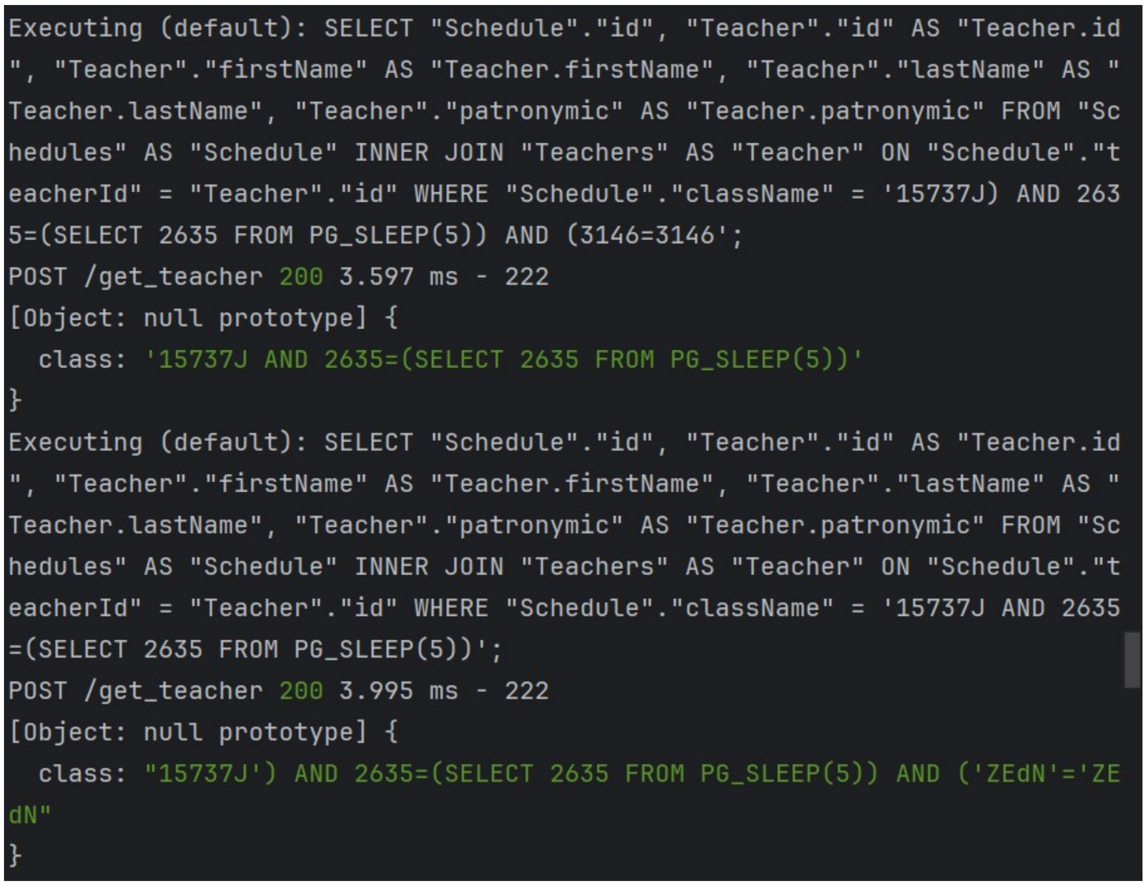
Проверим API с помощью sqlmap:

1. `python .\sqlmap.py --method POST -u "http://localhost:3000/get_teacher" --data "class=15737J" -p "class"`



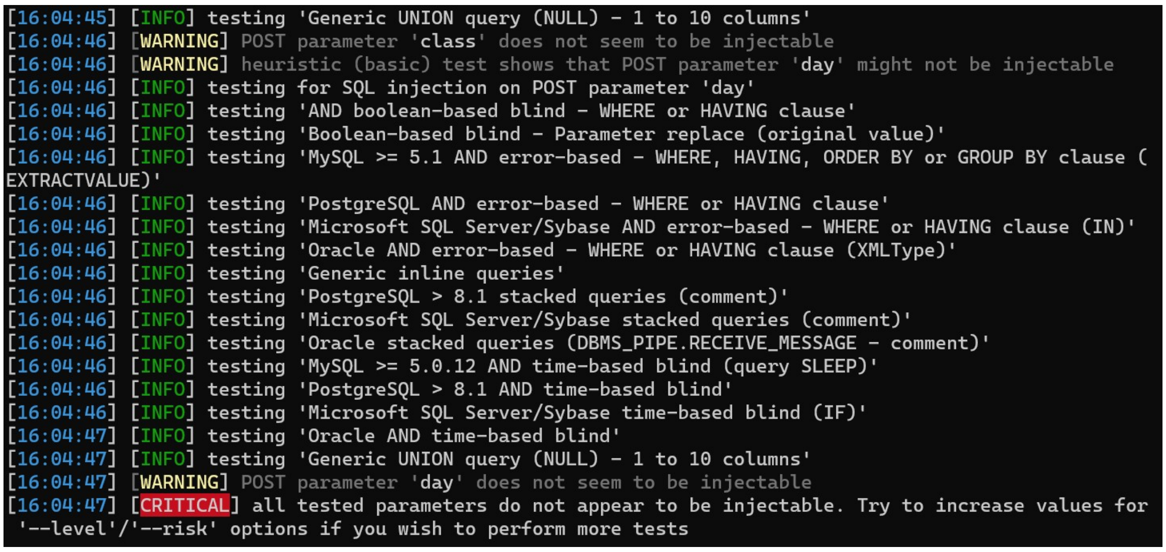
```
[*] starting @ 15:58:09 /2023-11-28/

[15:58:11] [INFO] testing connection to the target URL
[15:58:11] [INFO] testing if the target URL content is stable
[15:58:11] [INFO] target URL content is stable
[15:58:11] [WARNING] heuristic (basic) test shows that POST parameter 'class' might not be injectable
[15:58:11] [INFO] testing for SQL injection on POST parameter 'class'
[15:58:11] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:58:11] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[15:58:11] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (
EXTRACTVALUE)'
[15:58:11] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[15:58:12] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[15:58:12] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[15:58:12] [INFO] testing 'Generic inline queries'
[15:58:12] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:58:12] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:58:12] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:58:12] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[15:58:12] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[15:58:12] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[15:58:12] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) tec
hnique found. Do you want to reduce the number of requests? [Y/n] Y
[15:58:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:58:15] [WARNING] POST parameter 'class' does not seem to be injectable
[15:58:15] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for
 '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind
of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamp
er=space2comment') and/or switch '--random-agent'

[*] ending @ 15:58:15 /2023-11-28/
```

Поля не подвержены инъекции. Далее пример запросов, для проверки угрозы инъекции:



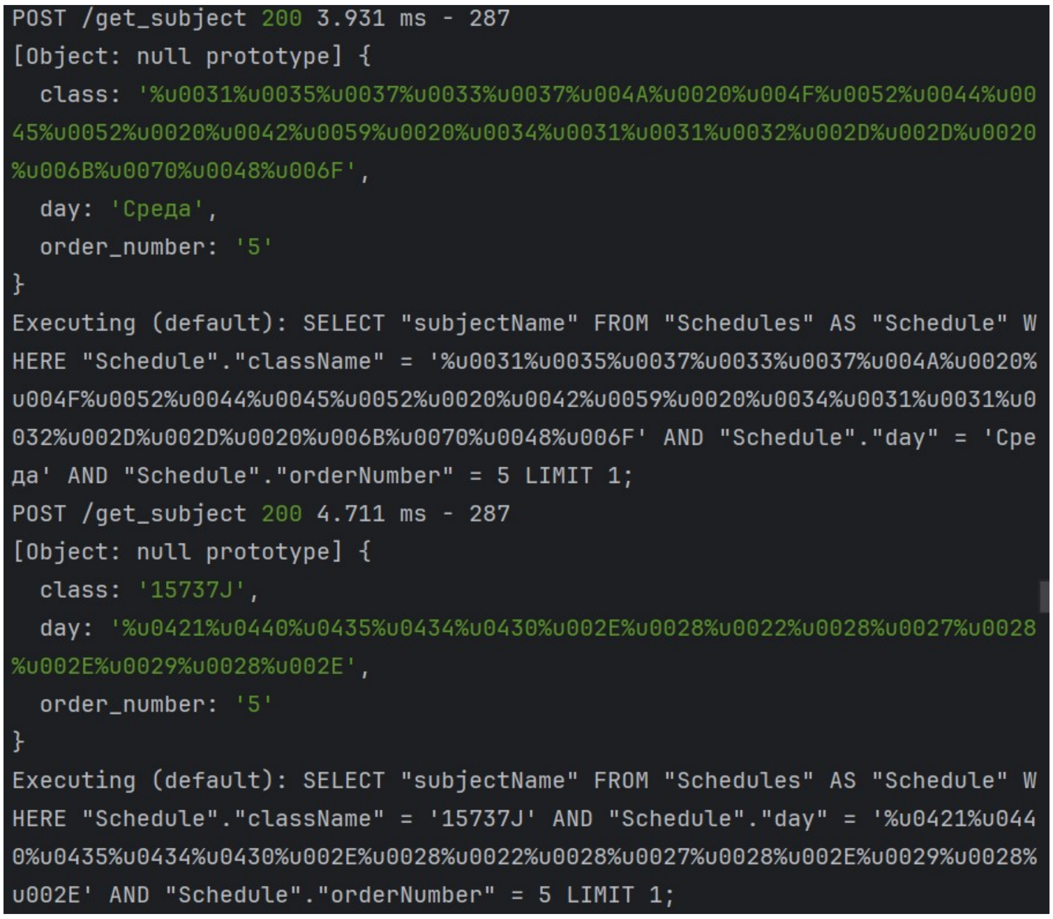
```
Executing (default): SELECT "Schedule"."id", "Teacher"."id" AS "Teacher.id
", "Teacher"."firstName" AS "Teacher.firstName", "Teacher"."lastName" AS "
Teacher.lastName", "Teacher"."patronymic" AS "Teacher.patronymic" FROM "Sc
hedules" AS "Schedule" INNER JOIN "Teachers" AS "Teacher" ON "Schedule"."t
eacherId" = "Teacher"."id" WHERE "Schedule"."className" = '15737J) AND 263
5=(SELECT 2635 FROM PG_SLEEP(5)) AND (3146=3146';
POST /get_teacher 200 3.597 ms - 222
[Object: null prototype] {
  class: '15737J AND 2635=(SELECT 2635 FROM PG_SLEEP(5))'
}
Executing (default): SELECT "Schedule"."id", "Teacher"."id" AS "Teacher.id
", "Teacher"."firstName" AS "Teacher.firstName", "Teacher"."lastName" AS "
Teacher.lastName", "Teacher"."patronymic" AS "Teacher.patronymic" FROM "Sc
hedules" AS "Schedule" INNER JOIN "Teachers" AS "Teacher" ON "Schedule"."t
eacherId" = "Teacher"."id" WHERE "Schedule"."className" = '15737J AND 2635
=(SELECT 2635 FROM PG_SLEEP(5))';
POST /get_teacher 200 3.995 ms - 222
[Object: null prototype] {
  class: "15737J') AND 2635=(SELECT 2635 FROM PG_SLEEP(5)) AND ('ZEdN'='ZE
dN"
}
```

2. python .\sqlmap.py --method POST -u "http://localhost:3000/get_subject" --data "class=15737J&day=Среда&order_number=5" -p "class,day" --tamper=charunicodeencode



```
[16:04:45] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[16:04:46] [WARNING] POST parameter 'class' does not seem to be injectable
[16:04:46] [WARNING] heuristic (basic) test shows that POST parameter 'day' might not be injectable
[16:04:46] [INFO] testing for SQL injection on POST parameter 'day'
[16:04:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:04:46] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[16:04:46] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (
EXTRACTVALUE)'
[16:04:46] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[16:04:46] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[16:04:46] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[16:04:46] [INFO] testing 'Generic inline queries'
[16:04:46] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[16:04:46] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[16:04:46] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[16:04:46] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[16:04:46] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[16:04:46] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[16:04:47] [INFO] testing 'Oracle AND time-based blind'
[16:04:47] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[16:04:47] [WARNING] POST parameter 'day' does not seem to be injectable
[16:04:47] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for
 '--level'/'--risk' options if you wish to perform more tests
```

Поля не подвержены инъекции. Далее пример запросов, для проверки угрозы инъекции:



```
POST /get_subject 200 3.931 ms - 287
[Object: null prototype] {
  class: '%u0031%u0035%u0037%u0033%u0037%u004A%u0020%u004F%u0052%u0044%u00
45%u0052%u0020%u0042%u0059%u0020%u0034%u0031%u0031%u0032%u002D%u002D%u0020
%u006B%u0070%u0048%u006F',
  day: 'Среда',
  order_number: '5'
}
Executing (default): SELECT "subjectName" FROM "Schedules" AS "Schedule" W
HERE "Schedule"."className" = '%u0031%u0035%u0037%u0033%u0037%u004A%u0020%
u004F%u0052%u0044%u0045%u0052%u0020%u0042%u0059%u0020%u0034%u0031%u0031%u0
032%u002D%u002D%u0020%u006B%u0070%u0048%u006F' AND "Schedule"."day" = 'Сре
да' AND "Schedule"."orderNumber" = 5 LIMIT 1;
POST /get_subject 200 4.711 ms - 287
[Object: null prototype] {
  class: '15737J',
  day: '%u0421%u0440%u0435%u0434%u0430%u002E%u0028%u0022%u0028%u0027%u0028
%u002E%u0029%u0028%u002E',
  order_number: '5'
}
Executing (default): SELECT "subjectName" FROM "Schedules" AS "Schedule" W
HERE "Schedule"."className" = '15737J' AND "Schedule"."day" = '%u0421%u044
0%u0435%u0434%u0430%u002E%u0028%u0022%u0028%u0027%u0028%u002E%u0029%u0028%
u002E' AND "Schedule"."orderNumber" = 5 LIMIT 1;
```

Выводы.

Протестировав доступ к БД с помощью ORM Sequelize на безопасность, получили, что система успешно справляется с угрозами SQL-инъекции.

ПРИЛОЖЕНИЕ А

Pull request: https://github.com/moevm/sql-2023-1303/pull/61