CSCB845 Компютърна сигурност

Кует Хъу Нгуен - F89497

Проект:
SQL Injection атаката на Java

Здравейте, Аз съм Кует Нгуен, факутет номер F89497. Аз съм студент по Информатика. В този проект аз ще направя проект за „*SQL Injection атаката на Java*“

Какво представлява SQL Injection атаката?

SQL Injector е техника за инжектиране на код който се използва за атака. Тя позволява да вземе информацията на една база данни. Първите признаци за уязвимост се появяват когато потребителят който използва **SQL Injection** въведе знаци които не могат да се филтрират правилно.

Demo за атаката

Проектът е направен от Spring Boot и Bootstrap 5

Demo за атаката

Имам бази данни с потребителски имена и пароли

id	acc_number	balance	branch_id	customer_id	password	username
1	1	1000.00	1	1	123456	user1
2	2	2000.00	2	2	123456	user2
3	3	1000.00	3	3	123456	user3
4	4	1000.00	4	4	123456	user4

Demo за атаката

Влизам в системата с user1 и парола = "123456"

Demo за атаката

Влизам в системата с грешно потребителско име или грешна парола

Demo за атаката

Влизам в системата с SQL Injection атаката.

Username = “sql-injection”

Password=“anypass ' or '1'='1”

Как да се предпазиме от такъв вид атаки?

unsafeFindAccountsByUsernameAndPassword

```java
public List<AccountDTO> unsafeFindAccountsByUsernameAndPassword(String username, String password) {

    String sql = "select "
            + "username,password,customer_id,acc_number,branch_id,balance from Accounts where username = '"
            + username + "' and password =  '"
            + password + "'";

    try (Connection c = dataSource.getConnection();
            ResultSet rs = c.createStatement()
                    .executeQuery(sql)) {
        List<AccountDTO> accounts = new ArrayList<>();
        while (rs.next()) {
            AccountDTO acc = AccountDTO.builder()
                    .customerId(rs.getString(columnLabel:"customer_id"))
                    .branchId(rs.getString(columnLabel:"branch_id"))
                    .accNumber(rs.getString(columnLabel:"acc_number"))
                    .balance(rs.getBigDecimal(columnLabel:"balance"))
                    .username(rs.getString(columnLabel:"username"))
                    .password(rs.getString(columnLabel:"password"))
                    .build();

            accounts.add(acc);
        }

        return accounts;
    } catch (SQLException ex) {
        throw new RuntimeException(ex);
    }
}
```

safeFindAccountsByUsernameAndPassword

```java
public List<AccountDTO> safeFindAccountsByUsernameAndPassword(String username, String password) {

    String sql = "select customer_id, branch_id,acc_number,balance from Accounts where username = ? and password = ?";

    try (Connection c = dataSource.getConnection();
            PreparedStatement p = c.prepareStatement(sql)) {
        p.setString(parameterIndex:1, username);
        p.setString(parameterIndex:2, password);
        ResultSet rs = p.executeQuery();
        List<AccountDTO> accounts = new ArrayList<>();
        while (rs.next()) {
            AccountDTO acc = AccountDTO.builder()
                    .customerId(rs.getString(columnLabel:"customer_id"))
                    .branchId(rs.getString(columnLabel:"branch_id"))
                    .accNumber(rs.getString(columnLabel:"acc_number"))
                    .balance(rs.getBigDecimal(columnLabel:"balance"))
                    .username(rs.getString(columnLabel:"username"))
                    .password(rs.getString(columnLabel:"password"))
                    .build();

            accounts.add(acc);
        }

        return accounts;
    } catch (SQLException ex) {
        throw new RuntimeException(ex);
    }
}
}
```

Благодаря за вниманието !