Izvestaj o SonarQube analizi projekta

- Pronadjeno je ukupno 50 *Security Hotspots* (potencijalnih) slabosti
- Napomena: deo liste pod nazivom *Linija slabosti* se odnosi na mesto gde je SonarQuebe javio gresku, ne na pravo mesto!

Sadrzaj

Lista slabosti

CSRF slabosti

- Pronadjena je jedna CSRF slabost

CSRF - Slabost 1

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/config/SecurityConfig.java

Linija/Linije slabosti: 27

Deo koda:

```
@Override

protected void configure(HttpSecurity http) throws Exception {

    http

        .csrf().disable()

        .authorizeRequests()

        .antMatchers("/login").permitAll()

        .antMatchers("/**").authenticated()

        .and()

        .formLogin()
```

Ishod: True Positive

Pojasnjenje: SonarQube je detektovao ispravno slabost (`csrf.disable()`), ali, ukoliko smo mi *custom* implementirali CSRF zastitu, ovo moze da se zanemari

SQLi

- Pronadjeno ukupno potencijalnih 24 slabosti

SQLi - Slabost 1

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 87

Deo koda:

```
public Object getRestaurant(String id) {

String query = "SELECT r.id, r.name, r.address, rt.name  FROM restaurant AS r JOIN restaurant_type AS rt ON r.typeId = rt.id WHERE r.id=" + id;

try (Connection connection = dataSource.getConnection();

        Statement statement = connection.createStatement();

        ResultSet rs = statement.executeQuery(query)) {

    if (rs.next()) {

        return createRestaurant(rs);

    }
```

Ishod: True Positive

Pojasnjenje: U ovoj situaciji je jasno da treba iskoristiti ***PreparedStatement*** umesto jednostavnog konkateniranja stringa. Napadac lako moze da izvrsi SQLi napad.

SQLi - Slabost 2

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 104

Deo koda:

```
public void deleteRestaurant(int id) {

    String query = "DELETE FROM restaurant WHERE id=" + id;

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {
        e.printStackTrace();

    }

}
```

Ishod: False Positive

Pojasnjenje: Ako pogledamo argument metode *deleteReastaurant* mozemo da vidimo da je id tipa int. Znajuci to, jasno je da ne moze da dodje do injekcije nezeljenog stringa te je ovaj kod u redu.

SQLi - Slabost 3

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 115

Deo koda:

```
public void updateRestaurant(RestaurantUpdate restaurantUpdate) {

    String query = "UPDATE restaurant SET name = '" + restaurantUpdate.getName() + "', address='" + restaurantUpdate.getAddress() + "', typeId =" + restaurantUpdate.getRestaurantType() + " WHERE id =" + restaurantUpdate.getId();

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}
```

Ishod: True Positive

Pojasnjenje: Konkateniramo stringove, neophodno je izmeniti kod tako da koristi ***PreparedStatement***

SQLi - Slabost 4

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 126

Deo koda:

```
public Customer getCustomer(String id) {
    String sqlQuery = "SELECT id, username, password FROM users WHERE id=" + id;
    try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement();
            ResultSet rs = statement.executeQuery(sqlQuery)) {
            if (rs.next()) {
                return createCustomerWithPassword(rs);
            }
```

Ishod: True Positive

Pojasnjenje: Konkateniramo stringove, neophodno je izmeniti kod tako da koristi ***PreparedStatement***

SQLi - Slabost 5

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 151

Deo koda:

```
public void deleteCustomer(String id) {
    String query = "DELETE FROM users WHERE id=" + id;
    try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement()
    ) {
        statement.executeUpdate(query);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Ishod: True Positive

Pojasnjenje: Konkateniramo stringove, neophodno je izmeniti kod tako da koristi ***PreparedStatement***

SQLi - Slabost 6

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 162

Deo koda:

```
public void updateCustomer(CustomerUpdate customerUpdate) {

    String query = "UPDATE users SET username = '" +
customerUpdate.getUsername() + "', password='" + customerUpdate.getPassword() +
"' WHERE id =" + customerUpdate.getId();

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}
```

Ishod: True Positive

Pojasnjenje: Konkateniramo stringove, neophodno je izmeniti kod tako da koristi ***PreparedStatement***

SQLi - Slabost 7

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 172

Deo koda:

```
public List<Address> getAddresses(String id) {

    String sqlQuery = "SELECT id, name FROM address WHERE userId=" + id;

    List<Address> addresses = new ArrayList<Address>();

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(sqlQuery)) {
```

```
        while (rs.next()) {
            addresses.add(createAddress(rs));
        }
```

Ishod: True positive

Pojasnjenje: Konkateniramo stringove, neophodno je izmeniti kod tako da koristi ***PreparedStatement***

SQLi - Slabost 8

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 196

Deo koda:

```
public void deleteCustomerAddress(int id) {
    String query = "DELETE FROM address WHERE id=" + id;
    try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement()
    ) {
        statement.executeUpdate(query);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Ishod: False Positive

Pojasnjenje: Vrsi se konkateniranje String i int, u ovom slucaju ne moze da dodje do SQLi jer int ne moze da sadrzi nizove karaktera koji bi bili maliciozne prirode

SQLi - Slabost 9

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 207

Deo koda:

```
public void updateCustomerAddress(Address address) {
    String query = "UPDATE address SET name = '" + address.getName() + "' WHERE id =" + address.getId();

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}
```

Ishod: True Positive

Pojasnjenje: Konkateniraju se stringovi, konkretno je address.GetName() string dok address.GetId moze da bude i int. Neophodno je koriscenje ***PreparedStatement***

SQLi - Slabost 10

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 218

Deo koda:

```
public void putCustomerAddress(NewAddress newAddress) {

        String query = "INSERT INTO address (name, userId) VALUES ('"+newAddress.getName()+"' , "+newAddress.getUserId()+")";

        try (Connection connection = dataSource.getConnection();

             Statement statement = connection.createStatement()

        ) {

            statement.executeUpdate(query);

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}
```

Ishod: True Positive

Pojasnjenje: Konkateniranje stringova, neophodno koriscenje ***PreparedStatement***

SQLi - Slabost 11

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/DeliveryRepository.java

Linija/Linije slabosti: 56

Deo koda:

```
public ViewableDelivery getDelivery(String id) {

    String sqlQuery = "SELECT d.id, d.isDone, d.date, d.comment, u.username,
r.name, rt.name, a.name FROM delivery AS d JOIN users AS u ON d.userId = u.id
JOIN restaurant as r ON d.restaurantId = r.id JOIN address AS a ON d.addressId =
a.id JOIN restaurant_type AS rt ON r.typeId= rt.id WHERE d.id = " + id;

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(sqlQuery)) {

        if (rs.next()) {

            return createDelivery(rs);

        }
```

Ishod: True Positive

Pojasnjenje: Konkateniranje stringova, neophodno koriscenje ***PreparedStatement***

SQLi - Slabost 12

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/DeliveryRepository.java

Linija/Linije slabosti: 74

Deo koda:

```
List<DeliveryDetail> details = new ArrayList<>();

String sqlQuery = "SELECT di.id, di.amount, f.name, f.price FROM delivery_item
AS di JOIN food AS f ON di.foodId = f.id WHERE deliveryId = " + id;

try (Connection connection = dataSource.getConnection();

        Statement statement = connection.createStatement();

        ResultSet rs = statement.executeQuery(sqlQuery)) {

    while (rs.next()) {

        details.add(createDetail(rs));
    }
```

Ishod: True Positive

Pojasnjenje: Konkateniranje stringova, neophodno koriscenje ***PreparedStatement***

SQLi - Slabost 13

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/DeliveryRepository.java

Linija/Linije slabosti: 117

Deo koda:

```
                + "OR UPPER(r.name) LIKE UPPER('%" + searchQuery + "%')"
                + "OR UPPER(rt.name) LIKE UPPER('%" + searchQuery + "%')"
                + "OR UPPER(a.name) LIKE UPPER('%" + searchQuery + "%')";
try (Connection connection = dataSource.getConnection();
        Statement statement = connection.createStatement();
        ResultSet rs = statement.executeQuery(sqlQuery)) {
    while (rs.next()) {
        cars.add(createDelivery(rs));
    }
}
return cars;
```

Ishod: True Positive

Pojasnjenje: Radi se konkateniranje stringova (searchQuery). Treba ***PreparedStatement***

SQLi - Slabost 14

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/HashedUserRepository.java

Linija/Linije slabosti: 27

Deo koda:

```
public HashedUser findUser(String username) {
    String sqlQuery = "select passwordHash, salt, totpKey from hashedUsers where username = '" + username + "'";
    try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(sqlQuery)) {

        if (rs.next()) {

            String passwordHash = rs.getString(1);

            String salt = rs.getString(2);

            String totpKey = rs.getString(3);

            return new HashedUser(username, passwordHash, salt, totpKey);
```

Ishod: True Positive

Pojasnjenje: Radi se konkateniranje stringova, treba koristiti ***PreparedStatement***

SQLi - Slabost 15

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 30

Deo koda:

```
public List<Food> getMenu(int id) {

    List<Food> menu = new ArrayList<>();

    String sqlQuery = "SELECT id, name FROM food WHERE restaurantId=" + id;

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(sqlQuery)) {

        while (rs.next()) {

            menu.add(createFood(rs));

        }

    } catch (SQLException e) {
```

Ishod: False Positive

Pojasnjenje: Vrsi se konkateniranje String i int, int nema mogucnost da prenosi nizove karaktera sa malicioznim kodom

SQLi - Slabost 16

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 56

Deo koda:

```
        "values (FALSE, " + userId + ", " + newOrder.getRestaurantId() + ", " +
newOrder.getAddress() + "," +

        "'" + date.getYear() + "-" + date.getMonthValue() + "-" +
date.getDayOfMonth() + "', '" + newOrder.getComment() + "')";

try {

    Connection connection = dataSource.getConnection();

    Statement statement = connection.createStatement();

    statement.executeUpdate(sqlQuery);

    sqlQuery = "SELECT MAX(id) FROM delivery";

    ResultSet rs = statement.executeQuery(sqlQuery);

    if (rs.next()) {
```

Ishod: True Positive

Pojasnjenje: Konkateniramo stringove (getComment i getAddress, ako pretpostavimo da su ostali int), treba koristiti ***PreparedStatement***

SQLi - Slabost 17

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 59

Deo koda:

```
Connection connection = dataSource.getConnection();

Statement statement = connection.createStatement();

statement.executeUpdate(sqlQuery);

sqlQuery = "SELECT MAX(id) FROM delivery";

ResultSet rs = statement.executeQuery(sqlQuery);

if (rs.next()) {

    int deliveryId = rs.getInt(1);

    sqlQuery = "INSERT INTO delivery_item (amount, foodId, deliveryId)" +
```

Ishod: False Positive

Pojasnjenje: Ovde se radi obican SQL upit bez parametara

SQLi - Slabost 18

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 76

Deo koda:

```
            }

            deliveryItem += "(" + item.getAmount() + ", " + item.getFoodId() + ", " + deliveryId + ")";

            sqlQuery += deliveryItem;

        }

        System.out.println(sqlQuery);

        statement.executeUpdate(sqlQuery);

    }

} catch (SQLException e) {

    e.printStackTrace();

}
```

Ishod: False Positive

Pojasnjenje: Radi se konkatenacija sa stringovima. FoodItem se sastoji od intova, deliveryId je iz baze selektovan i tipa je int.

SQLi - Slabost 19

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 91

Deo koda:

```
public Object getAddresses(int userId) {

    List<Address> addresses = new ArrayList<>();

    String sqlQuery = "SELECT id, name FROM address WHERE userId=" + userId;

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();
            ResultSet rs = statement.executeQuery(sqlQuery)) {

        while (rs.next()) {

            addresses.add(createAddress(rs));

        }

    } catch (SQLException e) {
```

Ishod: False Positive

Pojasnjenje: Konkatenacija String + int, sve je u redu

SQLi - Slabost 20

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/PermissionRepository.java

Linija/Linije slabosti: 32

Deo koda:

```
public List<Permission> findByRoleId(int roleId) {

    List<Permission> permissions = new ArrayList<>();

    String query = "SELECT id, name FROM permissions WHERE id IN (SELECT permissionId FROM role_to_permissions WHERE roleId=" + roleId + ")";

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(query)) {

        while (rs.next()) {

            int id = rs.getInt(1);

            String name = rs.getString(2);

            permissions.add(new Permission(id, name));

        }
```

Ishod: False positive

Pojasnjenje: Sve je u redu, konkatenacija String + int

SQLi - Slabost 21

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/RoleRepository.java

Linija/Linije slabosti: 32

Deo koda:

```
public List<Role> findByUserId(int userId) {

    List<Role> roles = new ArrayList<>();

    String query = "SELECT id, name FROM roles WHERE id IN (SELECT roleId FROM
user_to_roles WHERE userId=" + userId + ")";

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(query)) {

        while (rs.next()) {

            int id = rs.getInt(1);

            String name = rs.getString(2);

            roles.add(new Role(id, name));

        }
```

Ishod: False Positive

Pojasnjenje: Konkatenacija String + Int

SQLi - Slabost 22

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/UserRepository.java

Linija/Linije slabosti: 29

Deo koda:

```
public User findUser(String username) {

    String query = "SELECT id, username, password FROM users WHERE username='" +
username + "'";

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(query)) {

        if (rs.next()) {

            int id = rs.getInt(1);

            String username1 = rs.getString(2);

            String password = rs.getString(3);

            return new User(id, username1, password);
```

Ishod: True positive

Pojasnjenje: Konkatenacija stringova, neophodan ***PrepareStatement***

SQLi - Slabost 23

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/UserRepository.java

Linija/Linije slabosti: 46

Deo koda:

```
public boolean validCredentials(String username, String password) {

    String query = "SELECT username FROM users WHERE username='" + username + "' AND password='" + password + "'";

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(query)) {

        return rs.next();

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return false;
```

Ishod: True Positive

Pojasnjenje: Konkatenacija String, neophodno koriscenje ***PrepareStatement***

SQLi - Slabost 24

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/UserRepository.java

Linija/Linije slabosti: 59

Deo koda:

```
public void delete(int userId) {

    String query = "DELETE FROM users WHERE id = " + userId;

    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

    ) {

        statement.executeUpdate(query);
    } catch (SQLException e) {

        e.printStackTrace();

    }

}

}
```

Ishod: False Positive

Pojasnjenje: Sve je u redu, radi se konkatenacija String + int

Insecure Configuration

- Pronadjeno ukupno potencijalnih 25 slabosti

Insecure Configuration - Slabost 1

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 47

Deo koda:

```
        while (rs.next()) {

            customers.add(createCustomer(rs));

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return customers;

}

private com.zuehlke.securesoftwaredevelopment.domain.Customer
createCustomer(ResultSet rs) throws SQLException {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 2

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 68

Deo koda:

```
        while (rs.next()) {

            restaurants.add(createRestaurant(rs));

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return restaurants;

}

private Restaurant createRestaurant(ResultSet rs) throws SQLException {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 3

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 94

Deo koda:

```
            if (rs.next()) {

                return createRestaurant(rs);

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

        return null;

    }

    public void deleteRestaurant(int id) {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 4

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 106

Deo koda:

```
    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}

public void updateRestaurant(RestaurantUpdate restaurantUpdate) {

    String query = "UPDATE restaurant SET name = '" + restaurantUpdate.getName() + "', address='" + restaurantUpdate.getAddress() + "', typeId =" + restaurantUpdate.getRestaurantType() + " WHERE id =" + restaurantUpdate.getId();
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 5

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 117

Deo koda:

```
    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);
    } catch (SQLException e) {

        e.printStackTrace();

    }

}

public Customer getCustomer(String id) {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 6

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 133

Deo koda:

```
        if (rs.next()) {

            return createCustomerWithPassword(rs);

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return null;

}

private Customer createCustomerWithPassword(ResultSet rs) throws SQLException {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 7

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 153

Deo koda:

```
    try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}

public void updateCustomer(CustomerUpdate customerUpdate) {

    String query = "UPDATE users SET username = '" + customerUpdate.getUsername() + "', password='" + customerUpdate.getPassword() + "' WHERE id =" + customerUpdate.getId();
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 8

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 164

Deo koda:

```
    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}

public List<Address> getAddresses(String id) {

    String sqlQuery = "SELECT id, name FROM address WHERE userId=" + id;
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 9

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 180

Deo koda:

```
        while (rs.next()) {
            addresses.add(createAddress(rs));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return addresses;
}
private Address createAddress(ResultSet rs) throws SQLException {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 10

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 198

Deo koda:

```
    try (Connection connection = dataSource.getConnection();
            Statement statement = connection.createStatement()
    ) {
        statement.executeUpdate(query);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```
public void updateCustomerAddress(Address address) {

    String query = "UPDATE address SET name = '" + address.getName() + "' WHERE
id =" + address.getId();

```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 11

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 209

Deo koda:

```
    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement()

    ) {

        statement.executeUpdate(query);

    } catch (SQLException e) {

        e.printStackTrace();

    }

}

public void putCustomerAddress(NewAddress newAddress) {

    String query = "INSERT INTO address (name, userId) VALUES
('"+newAddress.getName()+"' , "+newAddress.getUserId()+")";

```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 12

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/CustomerRepository.java

Linija/Linije slabosti: 220

Deo koda:

```
        try (Connection connection = dataSource.getConnection();

             Statement statement = connection.createStatement()
        ) {
            statement.executeUpdate(query);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 13

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/DeliveryRepository.java

Linija/Linije slabosti: 33

Deo koda:

```
        while (rs.next()) {
            deliveries.add(createDelivery(rs));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return deliveries;
}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 14

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/DeliveryRepository.java

Linija/Linije slabosti: 63

Deo koda:

```
        if (rs.next()) {
            return createDelivery(rs);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}
public List<DeliveryDetail> getDeliveryDetails(String id) {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 15

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/DeliveryRepository.java

Linija/Linije slabosti: 81

Deo koda:

```
        while (rs.next()) {
            details.add(createDetail(rs));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return details;
}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 16

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/HashedUserRepository.java

Linija/Linije slabosti: 35

Deo koda:

```
            String salt = rs.getString(2);

            String totpKey = rs.getString(3);

            return new HashedUser(username, passwordHash, salt, totpKey);

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return null;

}

public void saveTotpKey(String username, String totpKey) {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 17

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/HashedUserRepository.java

Linija/Linije slabosti: 49

Deo koda:

```
            statement.setString(1, totpKey);

            statement.setString(2, username);

            statement.executeUpdate();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 18

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 36

Deo koda:

```
        while (rs.next()) {

            menu.add(createFood(rs));

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return menu;

}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 19

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 80

Deo koda:

```
            System.out.println(sqlQuery);

            statement.executeUpdate(sqlQuery);

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 20

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/OrderRepository.java

Linija/Linije slabosti: 97

Deo koda:

```
        while (rs.next()) {

            addresses.add(createAddress(rs));

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return addresses;

}

private Address createAddress(ResultSet rs) throws SQLException {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 21

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/PermissionRepository.java

Linija/Linije slabosti: 39

Deo koda:

```
                int id = rs.getInt(1);

                String name = rs.getString(2);

                permissions.add(new Permission(id, name));

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

        return permissions;

    }
```

```
}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 22

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/RoleRepository.java

Linija/Linije slabosti: 39

Deo koda:

```
                int id = rs.getInt(1);

                String name = rs.getString(2);

                roles.add(new Role(id, name));

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

        return roles;

    }

}
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 23

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/UserRepository.java

Linija/Linije slabosti: 37

Deo koda:

```
            String username1 = rs.getString(2);

            String password = rs.getString(3);

            return new User(id, username1, password);

        }

    } catch (SQLException e) {
        e.printStackTrace();

    }

    return null;

}

public boolean validCredentials(String username, String password) {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 24

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/UserRepository.java

Linija/Linije slabosti: 49

Deo koda:

```
    try (Connection connection = dataSource.getConnection();

            Statement statement = connection.createStatement();

            ResultSet rs = statement.executeQuery(query)) {

        return rs.next();

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return false;

}

public void delete(int userId) {
```

Ishod:

Pojasnjenje:

Insecure Configuration - Slabost 25

Fajl: src/main/java/com/zuehlke/securesoftwaredevelopment/repository/UserRepository.java

Linija/Linije slabosti: 61

Deo koda:

```
        try (Connection connection = dataSource.getConnection();
             Statement statement = connection.createStatement();

        ) {

            statement.executeUpdate(query);

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}
```

Ishod:

Pojasnjenje: