VILNIUS TECH
Faculty of Fundamental Sciences

Secure Programming Practical Work #2

Dr. Dainius Čeponis

2024

dainius.ceponis@vilniustech.lt

VILNIUS TECH
Faculty of Fundamental Sciences

Blind SQL injection

Purpose of the task:

1. Learn to find **vulnerable code** chunks
2. Learn to exploit **blind SQL injection** in order to understand the risks
3. Learn to **fix SQL injection** vulnerabilities

VILNIUS TECH
Faculty of Fundamental Sciences

Task sequence

Given an app with REST **services running on docker server image**:

- GET/users
- POST/users
- POST/login

Show the **proof of vulnerability by logging in as administrator**. App must response "**Logged in OK**" in API message.

*/*Consider you do not know salt value.*/*

VILNIUS TECH
Faculty of Fundamental Sciences

Tasks and points

- Exploiting
  - Identify vulnerable service *(1 point)*
  - Create new user *(1 point)*
  - Write your own application/script that is capable to extract your new user hash value ***(3 points)***
  - Replace administrator password hash with yours and try to login (could be done with single “command” by using UPDATE) ***(3 points)***
- Try automated tools like "sqlmap" or any other *(1 point)*
- Try find and edit app server code in order to fix SQL injection vulnerability *(1 point)*

test1:a63c638ff562b88a2538f627d356706d1cdfe365 Total: *(10 points)*

The report

- The report (with source code) must be uploaded to the corresponding TurnitIn section in the course Moodle.

VILNIUS TECH
Faculty of Fundamental Sciences

How to launch an application

- Install **docker-compose** (might be already in the system)
- Install **Curl, Postman** or **HTTPie**
- Install **SQLMap** or any other tool
- Download task folder from the Moodle
- Launch application using docker composer:

```
> cd Task2
> docker-compose up
```

VILNIUS TECH
Faculty of Fundamental Sciences

How to fix vulnerability

- In order to fix server source code, docker cache clean must be done (in other case old server code could be used from cache):

```
docker system prune -a
```

VILNIUS TECH
Faculty of Fundamental Sciences

Sample GET request for checking if user exists

- *curl "http://localhost:8080/users?username=admin"*

Sample POST request to the application for login:

- **Linux**: `curl --header "Content-Type: application/json" --data '{"userName":"admin","password":"guessme"}' http://localhost:8080/login`
- **Windows**: `curl --header "Content-Type: application/json" --data {\"userName\":\"admin\",\"password\":\"guessme\"} http://localhost:8080/login`

VILNIUS TECH
Faculty of Fundamental Sciences

Sample POST request to the application for new user

- **Linux:** *curl --header "Content-Type: application/json" --data '{"userName":"alex1","userFName":"Alexander","userLName":"Bob","password":"guessme"}' http://localhost:8080/users*
- **Windows:** *curl --header "Content-Type: application/json" --data {\"userName\":\"alex1\",\"userFName\":\"Alexander\",\"userLName\":\"Bob\",\"password\":\"guessme\"} http://localhost:8080/users*