Міністерство освіти і науки України
Національний технічний університет України
„КПІ імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЗВІТ
з лабораторної работи №1

з курсу «Безпека програмного забезпечення»
на тему “Огляд основних методів авторизації”

Перевірив:
Іваніщев Б. В.
Виконала:
Рекечинська Любов Русланівна
ІП-04

Київ 2023

Завдання: Викачати репозиторій з лекціями
https://github.com/Kreolwolf1/auth_examples
Запустити кожен з 3 аплікейшенів та зробити скріншити запитів до серверу.

Виконання завдання

Скопіюємо репозиторій цією командою:
`git clone https://github.com/Kreolwolf1/auth_examples`

Перед початком роботи, слід оглянути файли та перевірити, які залежності може містити кожен з них.
В ході огляду файлів було виявлено, що необхідно задовольнити залежність від цих пакетів NPM:

- `express`
- `uuid`
- `cookie-parser`
- `on-finished`
- `body-parser`

Задовольнити ці залежності можна у такий спосіб:

1. Запустити на рівні папки auth_examples команду `npm init` для ініціалізації файлу package.json. Він міститиме інформацію про встановлені залежності.
2. Встановити залежності за допомогою команди `npm i --save uuid express cookie-parser on-finished body-parser`.

Після цього вже можна запускати приклади авторизації.

Почнемо із методу **Basic Auth**:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples · (main±)
❯ cd basic_auth/

~/Documents/bubochka/volokyta-security/lab1/auth_examples/basic_auth · (main±)
❯ node index.js
Example app listening on port 3000
```

Запустимо ще один термінал для взаємодії з сервером.

Для того, щоб можна було авторизуватись, потрібно вказати заголовок (header) запиту “Authorization: Basic [user:password]”.

Зазвичай пара user:password закодовується за допомогою base64 для збереження цілісності інформації, оскільки цей формат використовує найбільш розповсюджені 64 символи для кодування інформації.

В коді вказані логін та пароль для авторизації:

```
if (login === 'DateArt' && password === '2408') {
    req.login = login;
    return next();
}
```

Закодуємо цю інформацію та зробимо запит за допомогою утиліти curl:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/basic_auth · (main±)
❯ printf DateArt:2408 | base64
RGF0ZUFydDoyNDA4

~/Documents/bubochka/volokyta-security/lab1/auth_examples/basic_auth · (main±)
❯ curl -X GET localhost:3000 -H 'Authorization: Basic RGF0ZUFydDoyNDA4'
Hello DateArt⏎
```

Token Auth

Цей формат використовує токен, створений після логування в систему. Цей метод є кращим за Basic Auth тим, що логін та пароль достатньо використати лише раз, щоб мати доступ до системи протягом сесії.

Логіни та паролі на цей раз дещо інші:

```
const users = [
    {
        login: 'Login',
        password: 'Password',
        username: 'Username',
    },
    {
        login: 'Login1',
        password: 'Password1',
        username: 'Username1',
    }
]
```

Запустимо сервер:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples · (main±)
❯ cd token_auth/

~/Documents/bubochka/volokyta-security/lab1/auth_examples/token_auth · (main±)
❯ node index.js
Example app listening on port 3000
```

У коді виявлено точку входу /api/login, яка приймає на вхід запит POST з параметрами login та password.

Зробимо запит POST з параметрами, переданими у тілі запиту формату JSON та перевіримо результат:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/token_auth · (main±)
❯ curl -X POST localhost:3000/api/login -H "Content-Type: application/json"
  -d '{ "login": "Login", "password": "Password" }'
{"token":"e29fce1d-a28c-460b-8cb1-b92397a7cd49"}⏎
```

Як наслідок, ми отримали токен. Використаємо його при створенні запиту до головної сторінки із заголовком “Authorization: [token]”:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/token_auth · (main±)
❯ curl -X GET localhost:3000/ -H 'Authorization: e29fce1d-a28c-460b-8cb1-b92397a
7cd49'
{"username":"Username","logout":"http://localhost:3000/logout"}⏎
```

В результаті запиту ми отримали JSON з назвою користувача Username та посиланням для виходу із сесії. Спробуємо вийти із сесії та побачимо, що буде:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/token_auth · (main±)
❯ curl -X GET localhost:3000/logout -H 'Authorization: e29fce1d-a28c-460b-8cb1-b
92397a7cd49'
Found. Redirecting to /⏎
```

Forms Auth

Цей спосіб авторизації відрізняється від двох попередніх тим, що результат авторизації записується в cookies, таким чином, немає потреби вказувати вручну в заголовках запиту спосіб авторизації.

За замовчуванням, запити curl не зберігають та не обробляють cookies.
Однак, ця функція підтримується, у доволі неординарний спосіб.
Параметр -c записує у файл cookies, отримані як результат HTTP запиту.
Параметр -b дістає з існуючого файлу cookies та додає їх у тіло запиту.

У скрипті для Forms Auth використовуються ті ж логіни та паролі, що і для Token Auth:

Зробимо запит з параметром -c:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/forms_auth · (main±)
❯ curl -X POST localhost:3000/api/login -H "Content-Type: application/json"
  -d '{ "login": "Login", "password": "Password" }' -c forms_auth_cookies
{"username":"Login"}⏎
```

На цей раз токен не було повернено. Перевіримо, що було записано у файл forms_auth_cookies:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/forms_auth · (main±)
❯ cat forms_auth_cookies
# Netscape HTTP Cookie File
# https://curl.se/docs/http-cookies.html
# This file was generated by libcurl! Edit at your own risk.

#HttpOnly_localhost     FALSE   /api/   FALSE   0       session d1a329e5-330d-4d
ad-8288-db4f0f2b9e9b
```

Як бачимо, натомість ID сесії було записано у cookies.

Використаємо цей файл для запиту до головної сторінки. Для цього після параметру -b треба записати послідовність session_key=session_value. У цьому випадку це session=d1a329e5-330d-4dad-8288-db4f0f2b9e9b

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/forms_auth · (main±)
❯ curl -X GET localhost:3000/ -b "session=d1a329e5-330d-4dad-8288-db4f0f2b9e9b"
{"username":"Username","logout":"http://localhost:3000/logout"}⏎
```

Як і в попередньому прикладі, спробуємо вийти із сесії:

```
~/Documents/bubochka/volokyta-security/lab1/auth_examples/forms_auth · (main±)
❯ curl -X GET localhost:3000/logout -b "session=d1a329e5-330d-4dad-8288-db4f0f2b
9e9b"
Found. Redirecting to /⏎
```

Висновок. Під час виконання лабораторної роботи було досліджено три види авторизації: Basic Auth, Token Auth та Forms Auth, порівняно їх принцип роботи та способи використання. Усі результати виконання команд було надано. Робота була виконана успішно.