jamk.fi

Web application security

Week 07

Timo Lehosvuo, M3426
TTV18S1

Harjoitustyö
Web application security, Heikki Salo, Joni Ahonen
28.3.2021
Tekniikan ala

1. Reading report

Not all authentication methods could be used with CSRF attacks (like non-cookie JWTs), but the two presented in the book can. What are those? (1 point)

- Basic authentication protocol or a cookie

Describe briefly how you can mitigate CSRF attacks

- Create a CSRF token. Sites require a CSRF token when a request that alters data is submitted. Web application creates a CSRF token with two parts, one for the user and the other for the application. When a user makes a request, he would have to submit his part of the token and the application would validate it with its side of the token. Tokens are created in a way that makes them unguessable and only accessible to the user they are assigned.

- The site could check the Origin or Referer header value submitted with HTTP request and ensure it contains the right value.

- Implement a “samesite” cookie attribute that browsers are beginning to support

2. Issue report

2.1 Attacker is able to change a user’s username with CSRF attack

Description: By creating a custom HTML code and sending it to the victim the attacker is able to change the user’s username from another origin

Steps to produce:

- Go to the website: http://htmledit.squarefree.com
- Create the HTML code:

```
<html>
  <body>
    <form action="http://192.168.43.2:3000/profile" method="POST">
      <input type="submit" name="username" value="testi" />
    </form>
    <script>
      // uncomment for automatic sending:
      // document.forms[0].submit();
    </script>
  </body>
</html>
```

testi

- Send it to the victim by clicking the “testi” button
- Observe the change of the username:

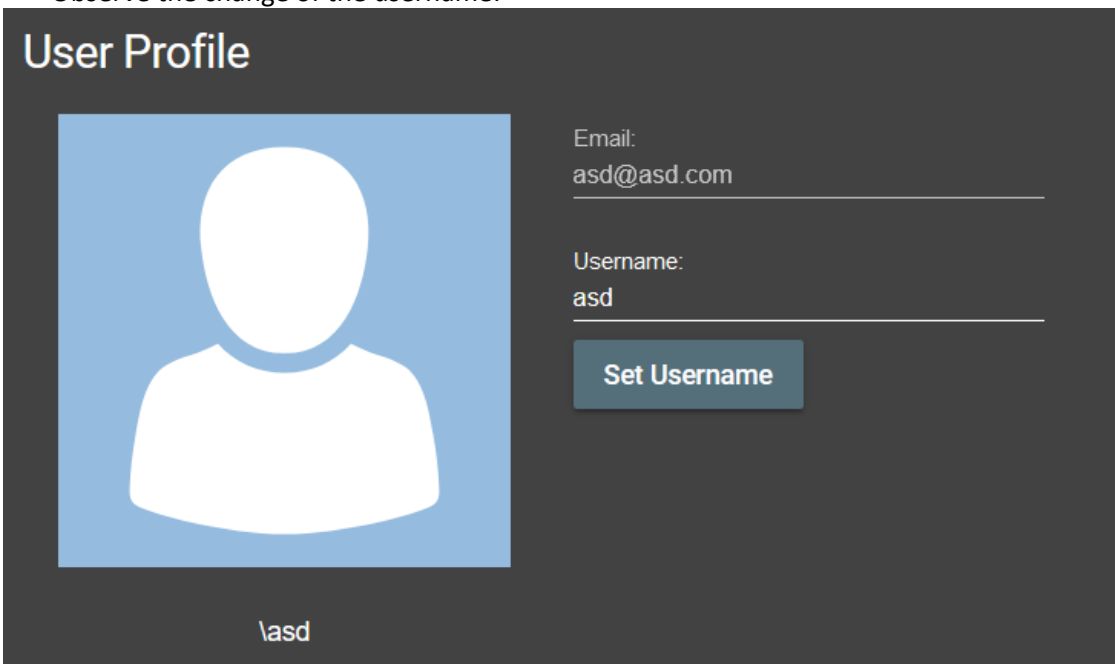


Figure 1: Original username

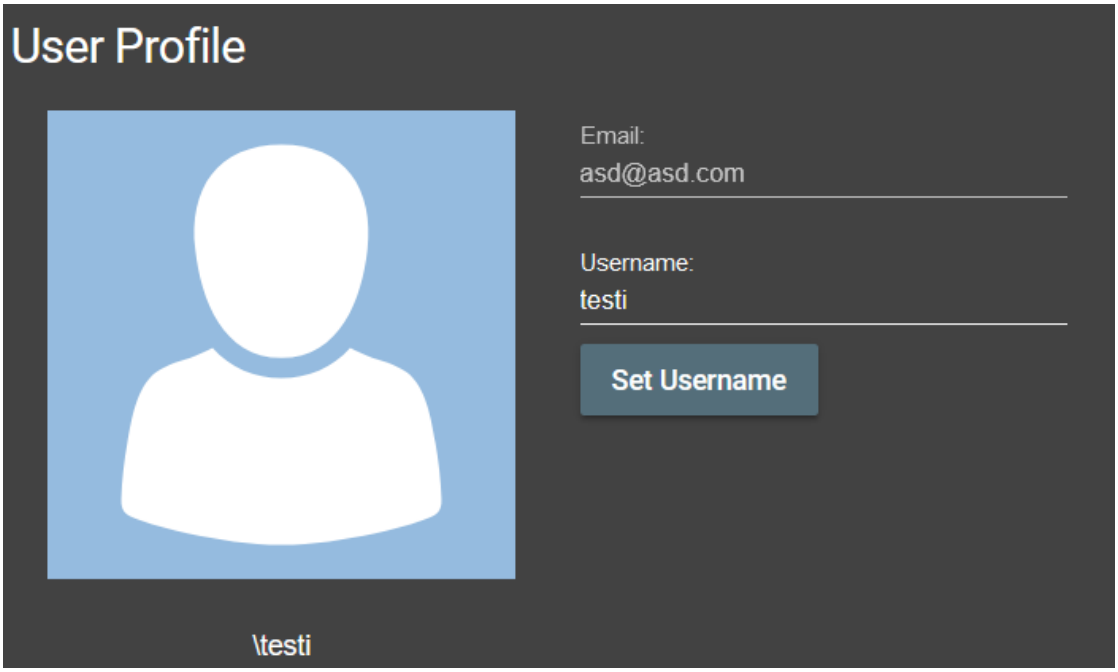
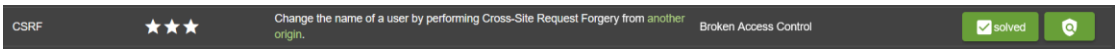


Figure 2: Changed username

Mitigation:

- Use CSRF tokens
- See: https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html

Achievement:



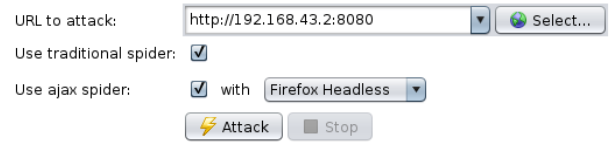
3 Issue report

3.1 Zaproxy scans reveals files that show information about the users

Description: Scanning the server with Zaproxy shows the users under the directory "profiles".

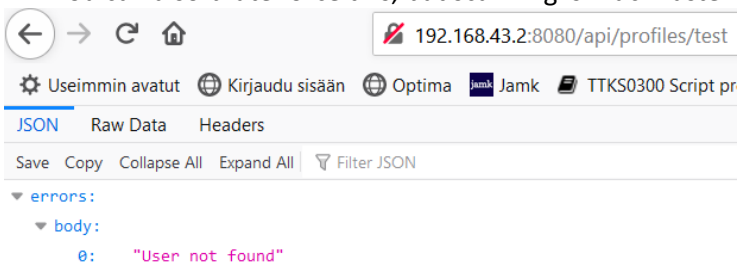
Steps to Produce:

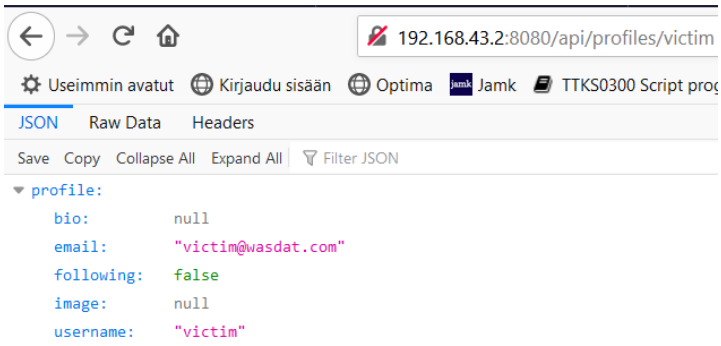
- Start Zaproxy
- Start scanning Wasdat with “ajax spider” on



- Navigate to the “profiles” directory

- You can also brute force this, but scanning is much faster





Mitigation:

- Block access to the API directory and subdirectories for unauthorized users
- See: https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html

Severity:

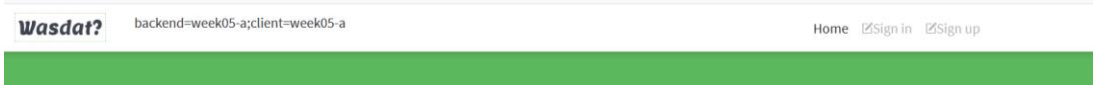
- Low/Medium. Attacker gains access to the users email addresses which they can possibly send phising emails or try to brute force the password.

3.2 Attacker can create articles under another user's name

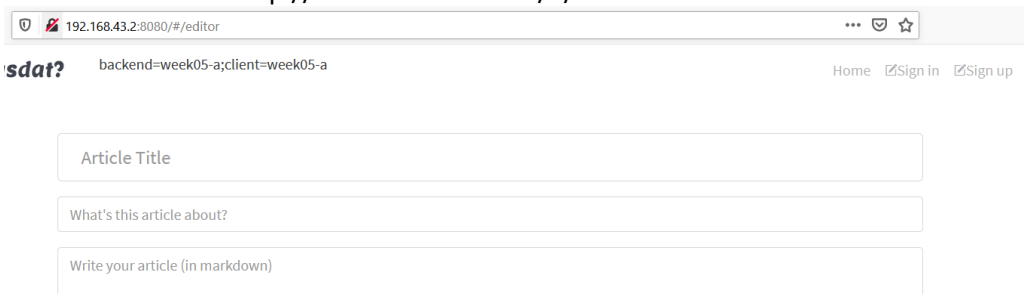
Description: Attacker can create articles under the name of the last user who logout.

Steps to produce:

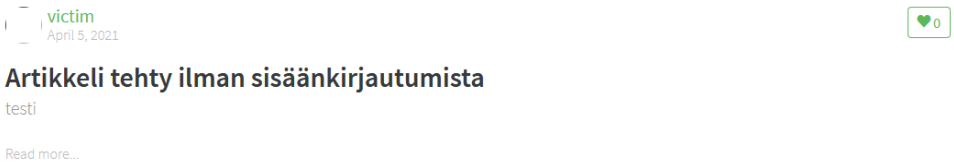
- Go to the "Wasdat" main page, login with any user and then logout



- Go to the URL: http://192.168.43.2:8080/#/editor



- Create a new article



- The article is created under the user "victim" since he was the last to log-out.

Mitigation:

- The JWT token should be invalidated when the user logs out from his account to prevent attacker creating false articles.
- Also access to the editor should be blocked for unauthorized users
- See: https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html

Severity:

- Low/Medium. Vulnerability makes it possible for the attacker to create articles that could be harmful for the reputation of the company or maybe further exploit this vulnerability to gain further access to the server. Articles could also contain links to fake websites that could possibly be used to gain for example user credentials.