Messenger Bug Hunting

A bug’s end-to-end lifecycle

Luke McLaren
Security Researcher

Farah Hawa
Security Analyst

Meta

% whoami

Luke McLaren
@datalocaltmp

Security Researcher

Farah Hawa

Security Analyst,
Bug Bounty @ Meta

Agenda

- Intro to Meta's Bug Bounty Program
- Overview of Facebook Messenger and it’s features
- Spoofing Threads - URL Validation Bug
- AR Effects - Crashing Group Calls
- Malforming Mentions and WordEffects
- Conclusion / Sign up bonus (EkoParty exclusive)

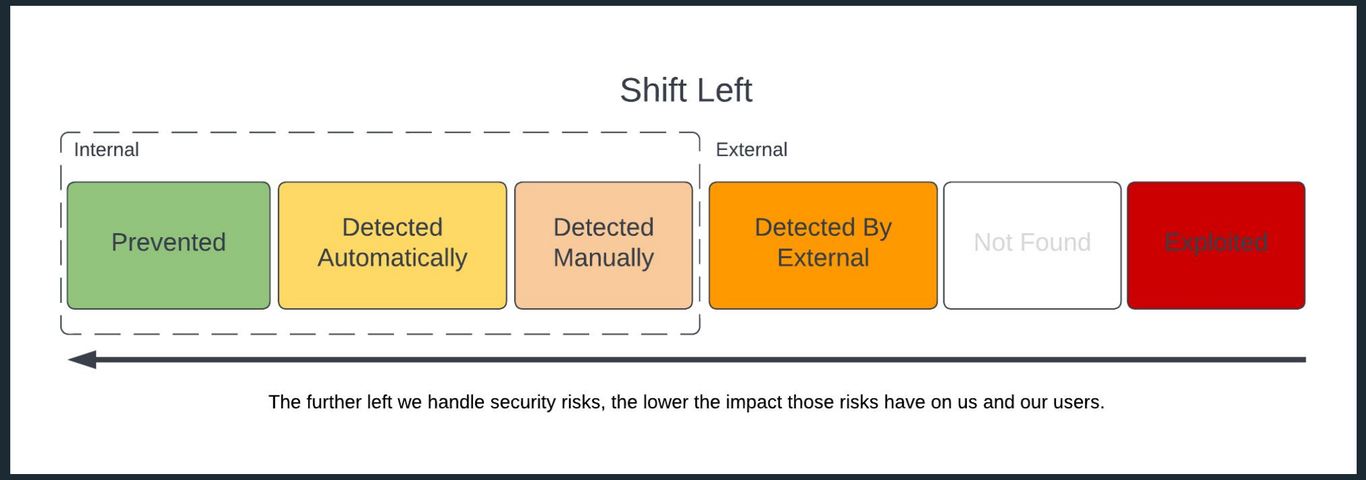
00 Intro to Meta's Bug Bounty Program

Overview

What's VRP

A vulnerability report program (commonly known as Bug Bounty) is a **detection method for vulnerabilities** in a responsible manner.

- External security researchers hunt for vulnerabilities we missed
- Any external individual can participate
- Influences security efforts across the organization
- Last line of defense



Key facts & figures

- 13 years old
  - since 2011, Meta has paid out more than $16 million in bug bounties.
- Covers Meta’s entire external attack surface
- Since 2011, Meta has received more than 170,000 reports
  - more than 8,500 were awarded a bounty.
- Sets direction for various internal security efforts
- Yearly we receive ~10k reports

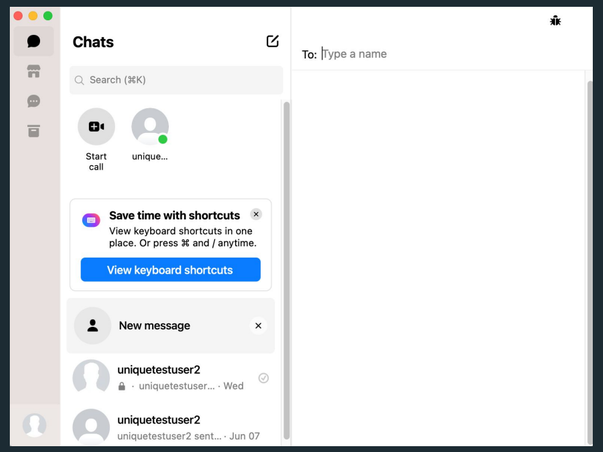
Bug Bounty Rewards

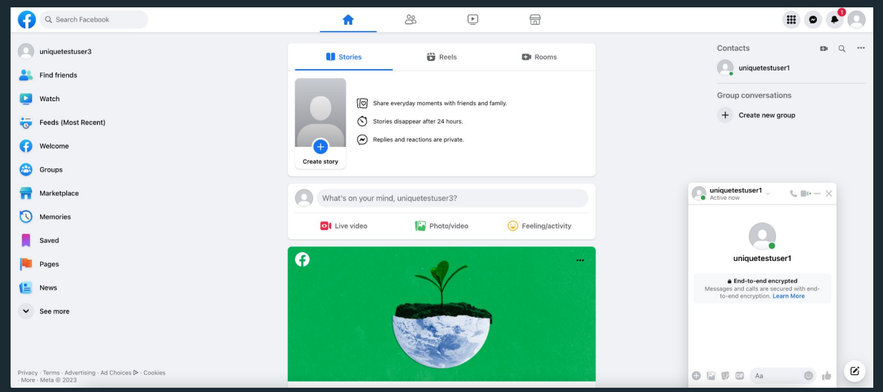
- All listed amounts are without bonuses. With HackerPlus, and payout time bonus, you can earn up to 30% of the original bounty amount on top of it!
- We pay based on maximum impact found internally, and our highest payouts reflect that.

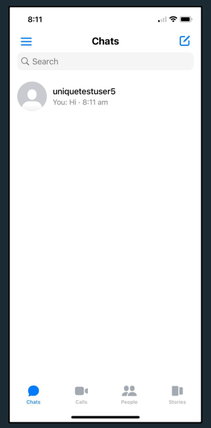
- Latest stats available at https://bugbounty.meta.com

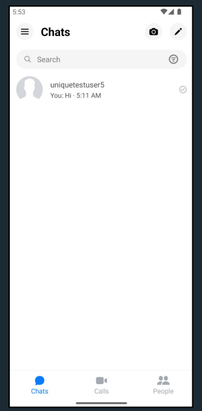
01 Intro to Messenger

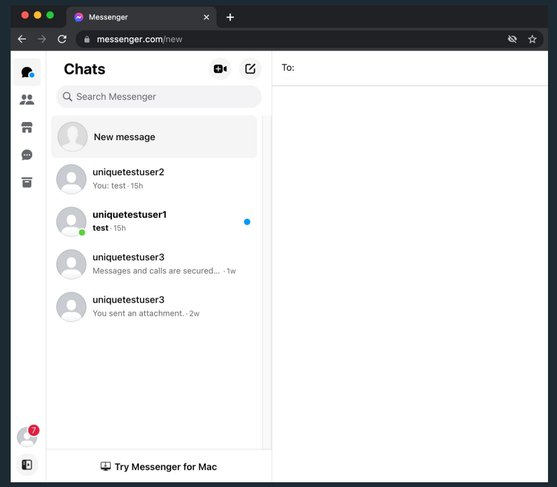
What is Messenger?







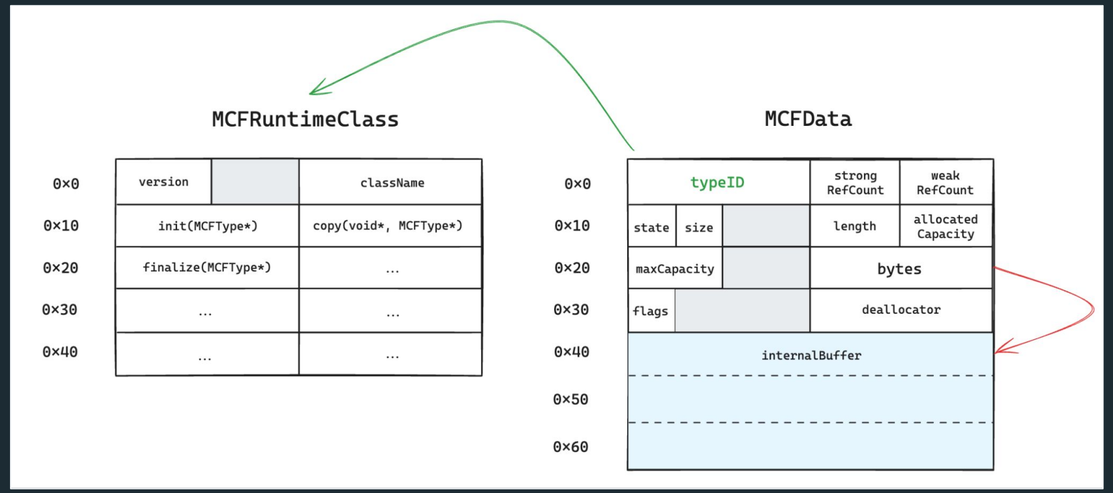




Messenger Messaging Architecture

- **Msys**
  - Cross platform messaging stack written in C
  - Manages database, accounts, incoming/outgoing messaging, etc.
  - E2EE messaging support requiring client side validation of messaging and media content

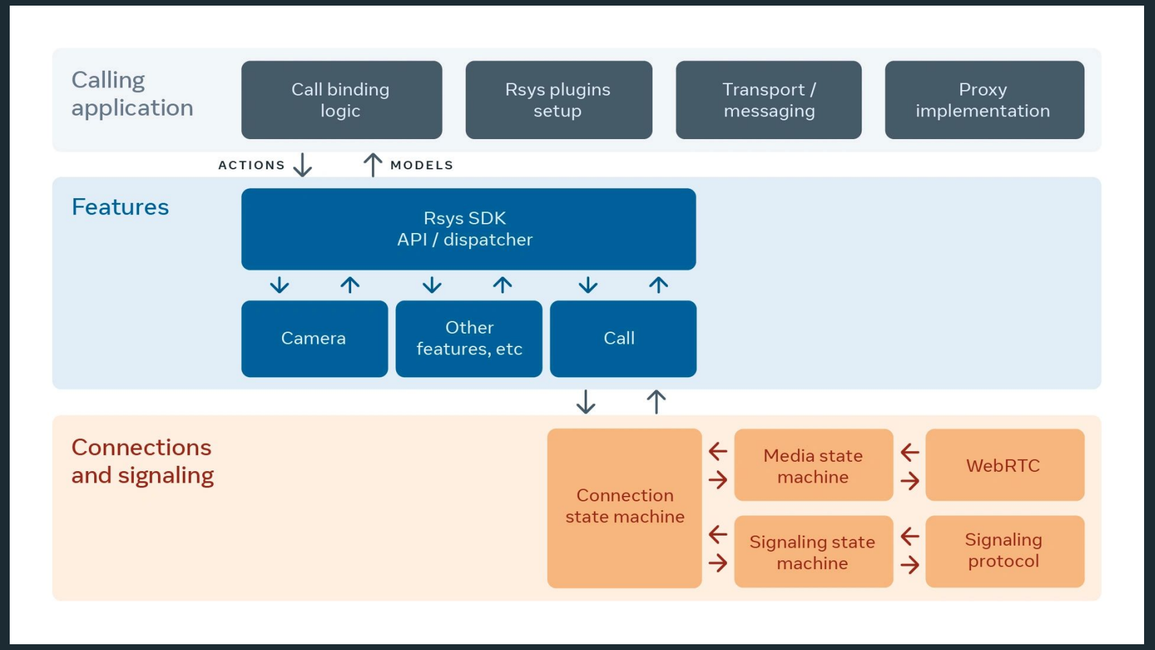
- **Messenger Core Foundations (MCF)**
  - Manages objects within Messenger
  - Core types used by Msys applications
  - Objects inherit from a base class, are reference counted, and encode type specific functionality such as initializers and destructors



Messenger Calling Architecture

- **Primarily managed by the Rsys and WebRTC libraries**
  - Supports both 1:1 and group audio/video calls
  - Rsys manages client side signaling and WebRTC
  - WebRTC maintains connections to servers/clients and manages media

- **Two relevant attack vectors to consider**
  - *Call* **Signaling**
    - Communication between clients, infrastructure, and other clients to manage call state
  - *Call* **Media**
    - WebRTC relevant protocols (e.g. RTP, STUN, SCTP) and audio/video codecs (e.g. OPUS, H264)



Messenger Bugs - Whys & Hows

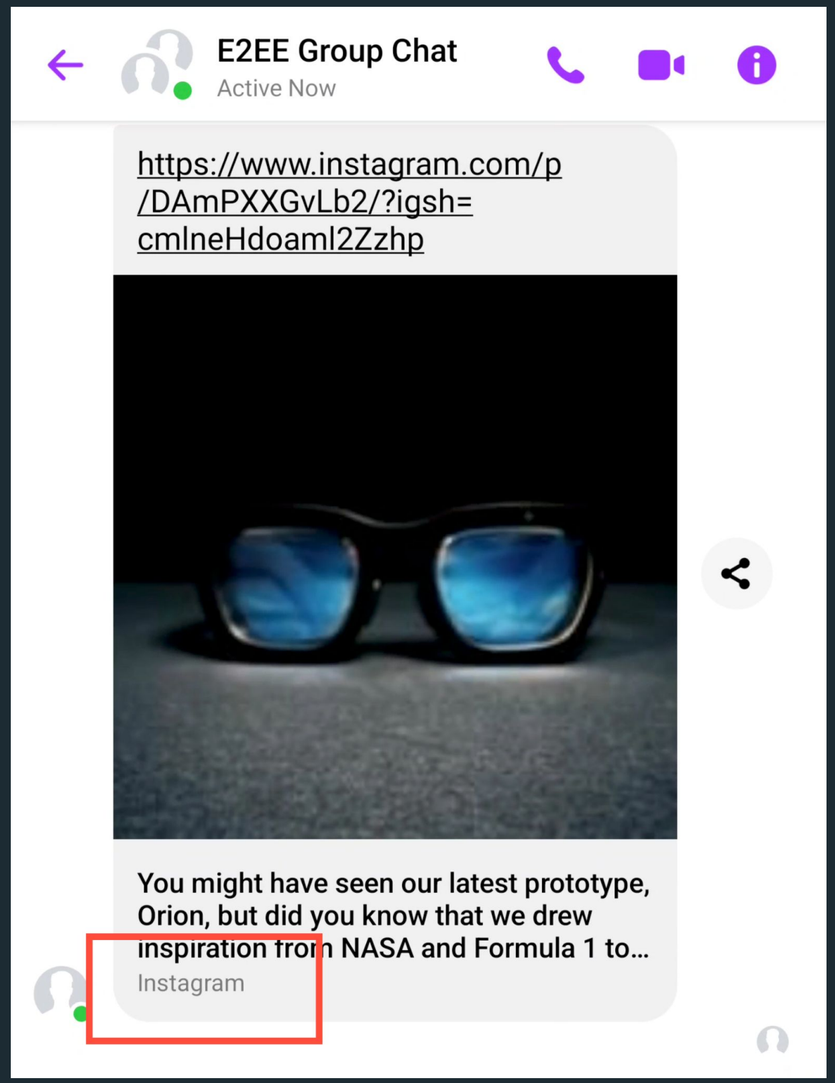
- Why Messenger?
  - Large attack surface within calling and messaging functionality
  - Extended Content Messages, AR Effects in Group Calls, Mentions & WordEffects
  - E2EE by default - lots of remotely accessible client side validation
- How?
  - I personally focus on Android using Frida/JADX/Ghidra/AFL++
- These bugs I picked were a few that spanned impact levels and discovery methods

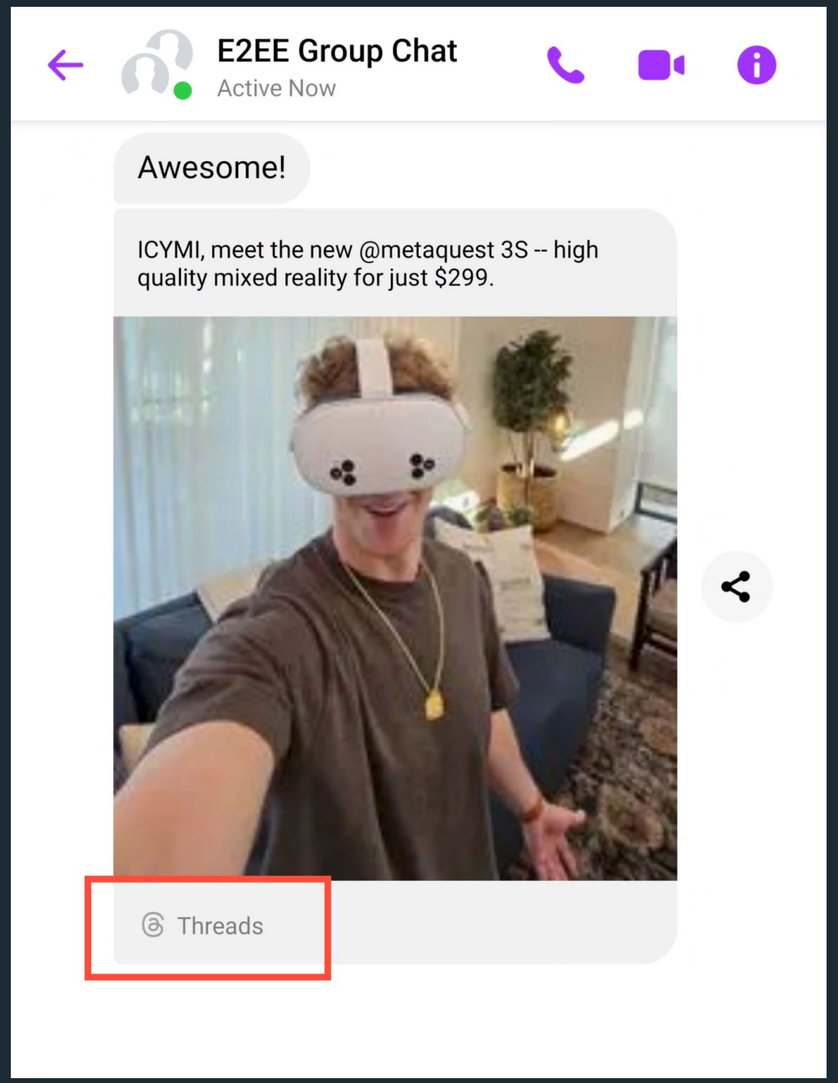
02 Threads Phishing - URL Validation Bug

Threads - Sharing via Messenger

- Threads is a text-based app from Instagram for sharing updates and joining public conversations.
- It aims to connect with decentralized social networks, allowing cross-platform interactions.
- Messenger supports sharing Threads in **branded** messages to facilitate sharing within the Meta ecosystem
- Represented in chats via an **Extended Content Message**

Example Branded Messages





How is branding applied?

- In an E2EE chat the client must make the determination on applying branding
- Client checks the message URL to determine whether to apply branding

```
//** Not the real code but a psuedo representation. **
function addThreadsBranding(msgUrl) {
    var msgHost = msgURL.getHost();
    const threadsdomain = "www.threads.net";

    var occuranceIndex =  msgHost.indexOf(threadsdomain);
    if (occuranceIndex >= 0) {
        return true;
    }
    return false;
}
```

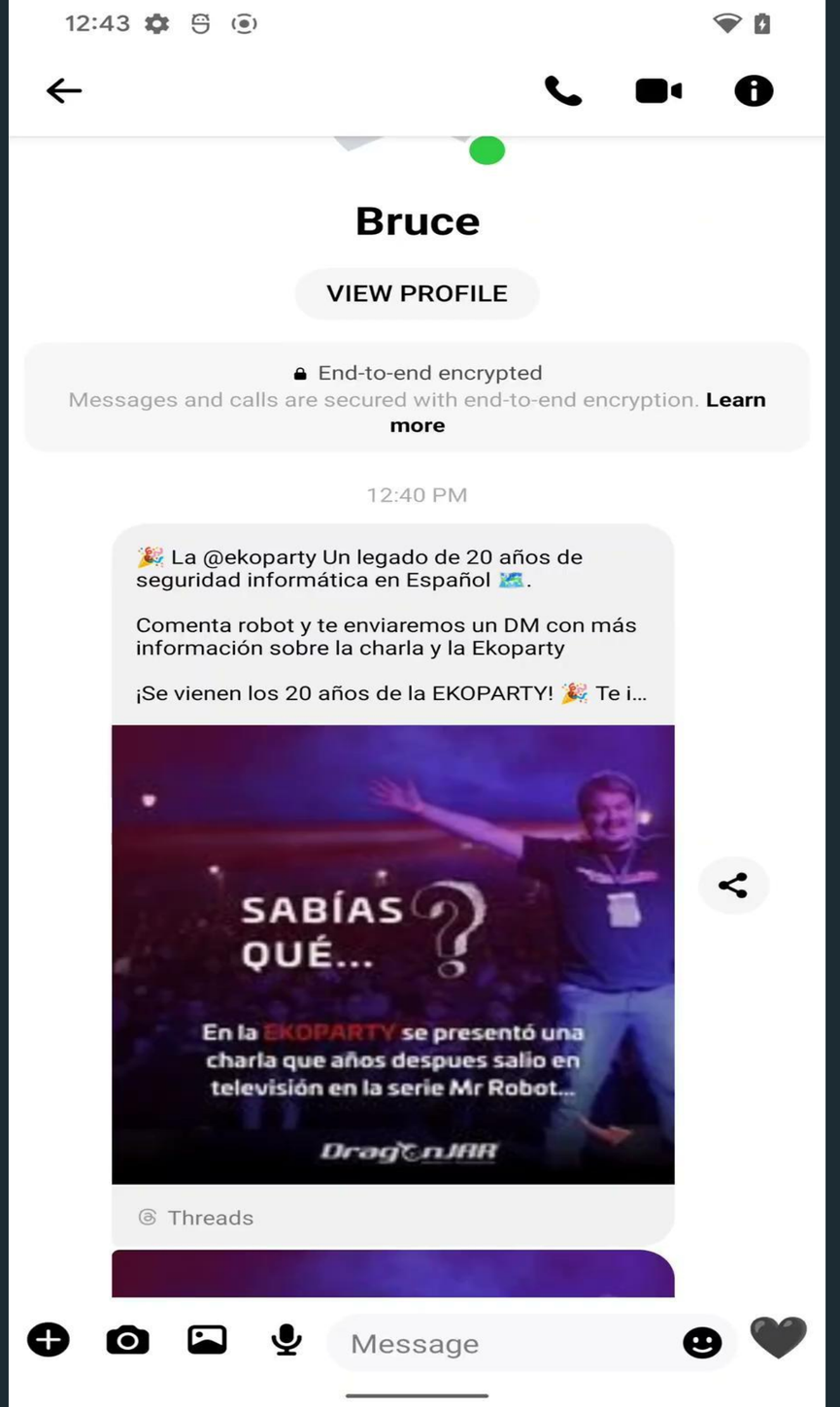
Bug - Improper URL Validation

- Sub-domain of ‘www.threads.net’ will validate as if it were Threads
  - i.e ‘https://www.threads.net.**malicious.com**’ will be branded as Threads
- Googled “site:threads.net.*” to find sites with matching sub-domains:
  - Found: threads.net.nz, threads.net.ru, threads.net.ng, threads.net.ar
  - All of which, if sent, were branded as Threads in E2EE chats in Messenger
  - ***All but the threads.net.nz domain (a legitimate clothing store) have since been decommissioned after the bug fix…***

Demo

- `XMASenderCopyOptions.defaultCta` contains URL
  - First Threads message redirects properly
  - Second Threads message redirects to non-Threads domain.

```
package com.facebook.sdk.mca;

/* loaded from: classes5.dex */
public class MailboxSDK$XMASenderCopyOptions {
    public String collapsibleId;
    public String cta1ActionContentBlob;

                    • • •

    public String defaultCta;
    public String defaultCtaTitle;
    public String headerAttributionContentToken;
    public Number headerAttributionHeight;

                    • • •

    public String titleText;
    public String xmaDataClass;
}
```



Bug Bounty team’s response

RCA & Fix

```
//Threads Prefix

MCF_STATIC_CONST_STRING(kThreadsCtaPrefix, "https://threads.net/");
MCF_STATIC_CONST_STRING(kThreadsWWWCtaPrefix, "https://www.threads.net/");

...

case ACTArmadilloAppExtendedContentTypeMsgThreadsPostShare:
    return _isValidUrlAndHasSamePrefix(url, kThreadsCtaPrefix, authDataContext)
    || _isValidUrlAndHasSamePrefix(url, kThreadsWWWCtaPrefix, authDataContext)

...
```

Payout Decision

Base payout = $1000

Gold HackerPlus bonus 10% = $100

03 AR Effects - Disrupting Group Calls

03 AR Effects - Disrupting Group Calls

03 AR Effects - Disrupting Group Calls

Meta Spark is shutting down

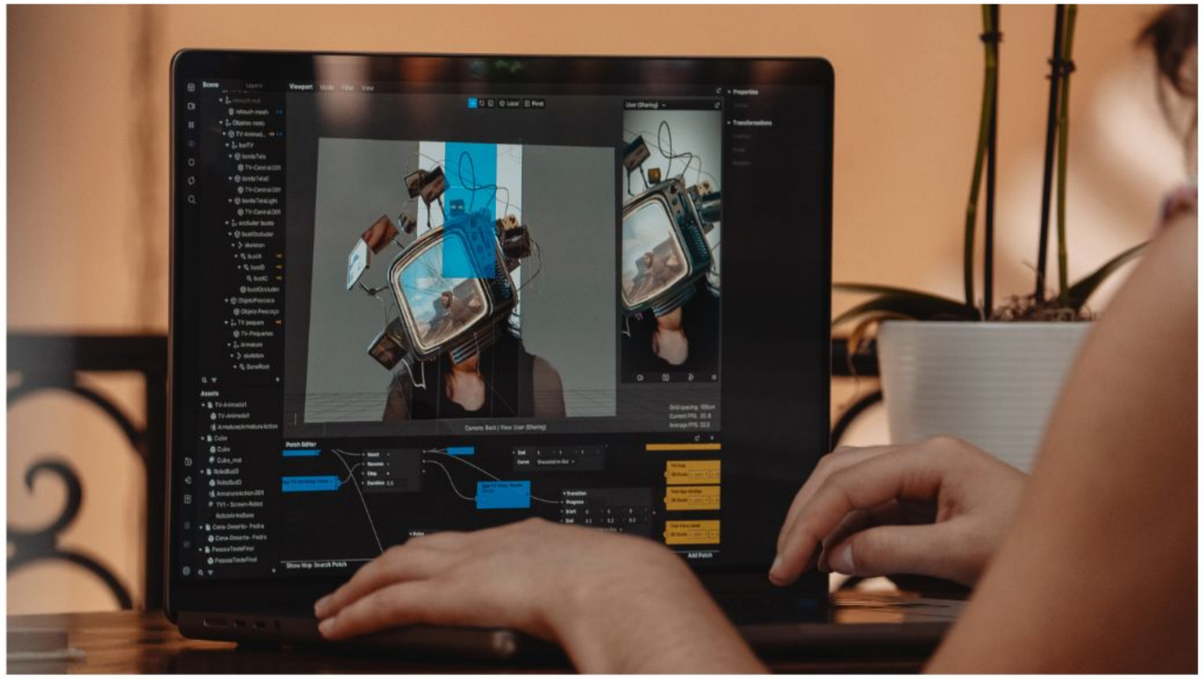
In case you missed it, Meta Spark's platform of third party tools and content will no longer be available effective January 14, 2025. Learn more

Announcement

Meta Spark is going away

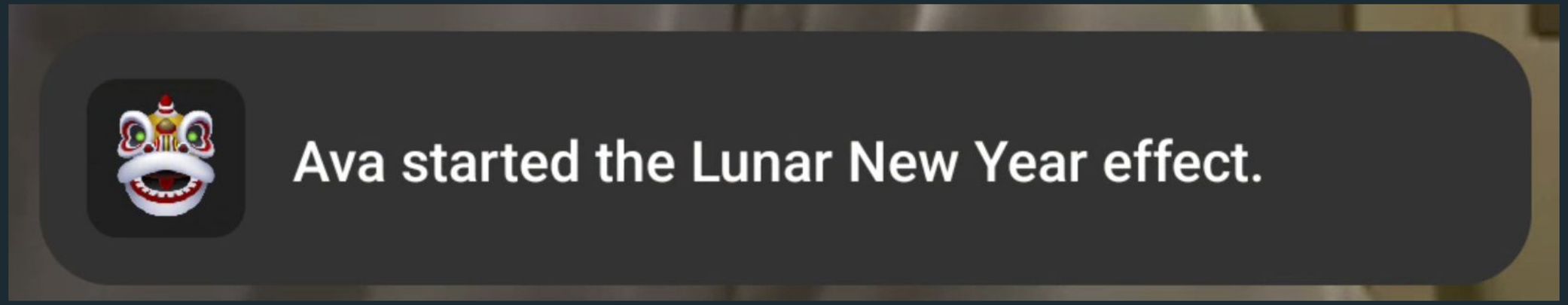
Meta Spark's platform of third party tools and content will no longer be available effective January 14, 2025.

Learn more



Group Calls & AR Effects

- Messenger chats allow you to apply augmented reality (AR) effects to your calls
- Also available within E2EE Calls - though the ID of the AR effect is revealed to Meta
- Within E2EE calls the AR Effect is sent as an encrypted binary payload encoding:
  - The icon URL, name of user, name of the effect, and the effect ID



AR Effect Binary Payload

```
[+] Found liborcarsysjniLatest.so base addr 0x7b00207000
[+] Found AR effect serialized payload
[!] Sending payload                    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
b400007d223aeec0  38 32 5c ad 7b 00 00 00 01 00 00 00 01 00 00 00  82\.{...........
b400007d223aeed0  01 00 b5 01 00 00 00 00 75 01 00 00 75 01 00 00  ........u...u...
b400007d223aeee0  75 01 00 00 00 00 00 00 00 ef 3a 22 7d 00 00 b4  u.........:"}...
b400007d223aeef0  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
b400007d223aef00  16 96 e0 d4 81 b6 88 8f 01 18 0e 4c 75 6e 61 72  ...........Lunar
b400007d223aef10  20 4e 65 77 20 59 65 61 72 18 bf 02 68 74 74 70   New Year...http
b400007d223aef20  73 3a 2f 2f 73 63 6f 6e 74 65 6e 74 2e 78 78 2e  s://scontent.xx.
b400007d223aef30  66 62 63 64 6e 2e 6e 65 74 2f 76 2f 74 33 39 2e  fbcdn.net/v/t39.
b400007d223aef40  31 30 32 36 30 2d 36 2f 32 37 31 31 36 35 36 33  10260-6/27116563
b400007d223aef50  37 5f 32 39 36 34 39 35 35 37 38 37 30 38 37 38  7_29649557870878
b400007d223aef60  39 36 5f 31 35 33 30 34 33 38 39 36 39 33 39 35  96_1530438969395
b400007d223aef70  34 33 33 36 36 35 5f 6e 2e 70 6e 67 3f 73 74 70  433665_n.png?stp
b400007d223aef80  3d 64 73 74 2d 70 6e 67 5f 70 32 34 30 78 32 34  =dst-png_p240x24
b400007d223aef90  30 26 5f 6e 63 5f 63 61 74 3d 31 26 63 63 62 3d  0&_nc_cat=1&ccb=
b400007d223aefa0  31 2d 37 26 5f 6e 63 5f 73 69 64 3d 64 31 31 64  1-7&_nc_sid=d11d
b400007d223aefb0  62 63 26 5f 6e 63 5f 6f 68 63 3d 57 45 32 41 6f  bc&_nc_ohc=WE2Ao
b400007d223aefc0  2d 31 78 56 54 38 51 37 6b 4e 76 67 46 71 37 76  -1xVT8Q7kNvgFq7v
b400007d223aefd0  5a 61 26 5f 6e 63 5f 61 64 3d 7a 2d 6d 26 5f 6e  Za&_nc_ad=z-m&_n
b400007d223aefe0  63 5f 63 69 64 3d 30 26 5f 6e 63 5f 68 74 3d 73  c_cid=0&_nc_ht=s
b400007d223aeff0  63 6f 6e 74 65 6e 74 2e 78 78 26 5f 6e 63 5f 67  content.xx&_nc_g
b400007d223af000  69 64 3d 41 6b 37 5a 45 49 69 6c 5a 66 53 6f 4a  id=Ak7ZEIilZfSoJ
b400007d223af010  36 42 6f 79 5a 6e 4e 54 51 4f 26 6f 68 3d 30 30  6BoyZnNTQO&oh=00
b400007d223af020  5f 41 59 44 50 75 79 54 52 51 70 6e 6e 79 61 71  _AYDPuyTRQpnnyaq
b400007d223af030  6a 47 38 71 5f 58 58 62 42 54 36 34 6e 4d 41 6a  jG8q_XXbBT64nMAj
b400007d223af040  5a 71 53 6d 47 78 57 6c 71 7a 78 72 52 41 41 26  ZqSmGxWlqzxrRAA&
b400007d223af050  6f 65 3d 36 37 30 32 38 34 45 42 18 0e 36 31 35  oe=670284EB..615
b400007d223af060  36 33 35 35 36 38 31 36 34 32 35 18 05 4a 61 6d  63556816425..Jam
b400007d223af070  65 73 25 00 00 00 00 00 00 00 00 00 00 00 00 00  es%.............
```

AR Effect Name

AR Effect URL

AR Effect ID

User Name

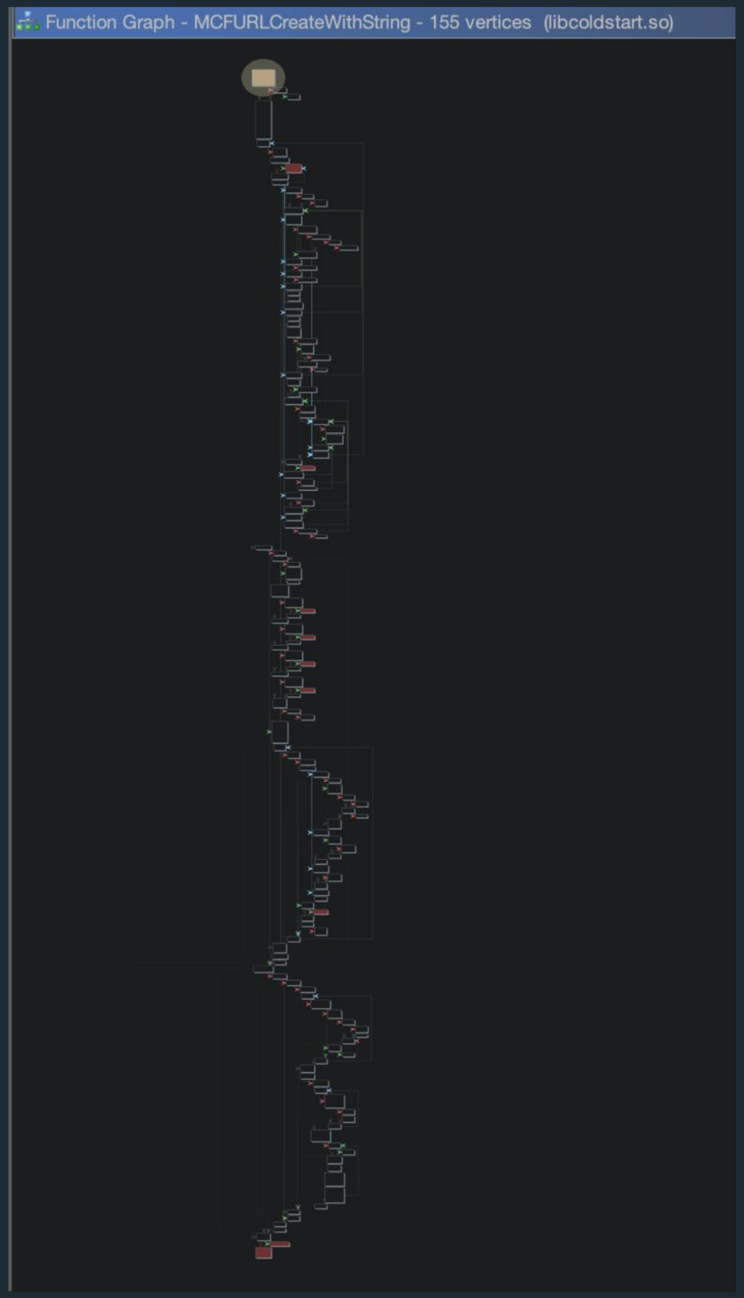
Validating AR Effect Icon URLs

- AR Effect Icons are required to be hosted at https://scontent.xx.fbcdn.net/
- Messenger does not want one of these E2EE Icon URLs to outside content hosts
  - Potential for malicious redirects to leak information etc.
- URLs are validated by the native library `liborcarsysjniLatest.so`
  - Rsys native library for managing calls

URL Validation Code

Decompile: FUN_0074e4d4 - (orcalatest.so)

```
1
2  bool FUN_0074e4d4(char *url)
3
4  {
5    undefined in_ZR;
6    bool bVar1;
7    int iVar2;
8    long lVar3;
9    undefined auStack_d0 [128];
10   char *local_50;
11   undefined8 local_48;
12   undefined8 uStack_40;
13   undefined8 local_38;
14
15   lVar3 = tpidr_el0;
16   local_38 = *(undefined8 *)(lVa 3 + 0x28);
17   DAT_008e6f9b = 0;
18   lVar3 = MCFURLCreateWithCString(url);
19   if (lVar3 == 0) {
20     bVar1 = false;
21   }
22   else {
23     lVar3 = MCFURLCopyComponent(lVar3,8);
24     if (lVar3 == 0) {
25       bVar1 = false;
26     }
27     else {
28       FUN_003a7688(auStack_d0,lVar3);
29       iVar2 = strcmp(local_50,"scontent.xx.fbcdn.net");
30       bVar1 = iVar2 == 0;
31       in_ZR = bVar1;
32       MCFStringDeallocateCharArray(local_48,uStack_40);
33     }
34     FUN_00871100();
35   }
36   FUN_00871138();
37   FUN_00870798();
38   if ((bool)in_ZR) {
39     return bVar1;
40   }
41                     /* WARNING: Subroutine does not return */
42   __stack_chk_fail();
43 }
44
```



Validating AR Effect Icon URLs

- Complex URL parsing native code seems very interesting for fuzzing
  - Large amount of complexity hidden within `MCFURLCreateWithCString(...)`
  - Wrote a AFL++ harness to fuzz - (un)fortunately code seems memory safe
- What about fuzzing the logic itself? Find a valid/invalid URL
  - Reminder that we can use fuzzers to find non-memory corruption bugs

Fuzzing URL Validation

```c
void fuzz_one_input(char* filename, checkURI_t checkUri){
    char uri[MAX_SIZE];
    printf("[*] Opening file %s\n", filename);


    FILE *f = fopen(filename, mode: "r");
    fgets(uri, MAX_SIZE, f);
    printf("[*] Checking %s for safety\n", uri);

    int safeUri = checkUri(uri);
    if(safeUri){
        printf("[+] liborcarsysjniLatest.so reports %s as safe\n", uri);
        if(strncmp(s1: uri,s2: "https://scontent.xx.fbcdn.net/",n: 30) != 0){
            printf("[!] Determined %s is actually unsafe\n", uri);
            checkURI_t test = 0x0;
            int crash_here = test("CRASH!");
        }else{
            printf("[+] Carry on %s is actually safe\n", uri);
        }
    }else{
        printf("[+] liborcarsysjniLatest.so reports %s as unsafe\n", uri);
    }
}
```

```
american fuzzy lop ++3.13c (main) [fast] {7}
┌ process timing ──────────────────────────┬ overall results ──────┐
│        run time : 0 days, 18 hrs, 50 min, 31 sec │  cycles done : 15    │
│   last new path : 0 days, 2 hrs, 42 min, 24 sec  │  total paths : 487   │
│ last uniq crash : 0 days, 2 hrs, 42 min, 23 sec  │ uniq crashes : 13    │
│  last uniq hang : none seen yet                  │   uniq hangs : 0     │
├ cycle progress ──────────┬ map coverage ──────────┴──────────────────────┤
│  now processing : 387*0 (79.5%)  │    map density : 1.12% / 2.18%       │
│ paths timed out : 0 (0.00%)      │ count coverage : 3.42 bits/tuple     │
├ stage progress ──────────┼ findings in depth ─────┬─────────────────────┤
│  now trying : havoc              │ favored paths : 65 (13.35%)          │
│ stage execs : 754/768 (98.18%)   │  new edges on : 116 (23.82%)         │
│ total execs : 44.9M              │ total crashes : 266 (13 unique)      │
│  exec speed : 1430/sec           │  total tmouts : 0 (0 unique)         │
├ fuzzing strategy yields ─────────┴──────────────┬ path geometry ────────┤
│   bit flips : 2/348k, 0/348k, 0/348k            │    levels : 8         │
│  byte flips : 0/43.5k, 0/43.5k, 0/43.4k         │   pending : 236       │
│ arithmetics : 1/2.40M, 0/8778, 0/2238           │  pend fav : 0         │
│  known ints : 0/279k, 0/1.22M, 0/1.91M          │ own finds : 485       │
│  dictionary : 0/0, 0/0, 0/0                     │  imported : 0         │
│havoc/splice : 4/655k, 0/1.03M                   │ stability : 92.38%    │
│py/custom/rq : unused, unused, unused, unused    ├───────────────────────┘
│    trim/eff : disabled, 85.41%                  │          [cpu007: 50%]
└─────────────────────────────────────────────────┘
```

Fuzzing URL Validation

```
# LD_LIBRARY_PATH=./libs LD_PRELOAD=./libc++_shared.so ./validate ./input/safe.txt

[*] Fuzzing URI checker in messenger
[*] Found 0x728d8544d4 for function in liborcarsysjniLatestPatched.so
[*] Opening file ./input/safe.txt
[*] Checking https://scontent.xx.fbcdn.net/v/t39.....siofgeDzLwJC for safety
[+] liborcarsysjniLatest.so reports https://scontent.xx.fbcdn.net/v/t39.....siofgeDzLwJC as safe
[+] Carry on https://scontent.xx.fbcdn.net/v/t39.....siofgeDzLwJC is actually safe

# LD_LIBRARY_PATH=./libs LD_PRELOAD=./libc++_shared.so ./validate ./output/main/crashes/crash11.txt

[*] Fuzzing URI checker in messenger
[*] Found 0x6fb56574d4 for function in liborcarsysjniLatestPatched.so
[*] Opening file ./output/main/crashes/crash11.txt
[*] Checking https://scontent.xx.fbcdn.net?v/ for safety
[+] liborcarsysjniLatest.so reports https://scontent.xx.fbcdn.net?v/ as safe
[!] Determined https://scontent.xx.fbcdn.net?v/ is actually unsafe
Segmentation fault
```

Sending Valid/Invalid Input

RCA & Fix

- Code responsible for parsing an AR effect message on the receiver side only validates that the AR effect icon's URL has the hostname: scontent.xx.fbcdn.net.
- Sending a URL such as https://scontent.xx.fbcdn.net/invalid or an empty path will result in the icon image failing to be loaded which results in an exception

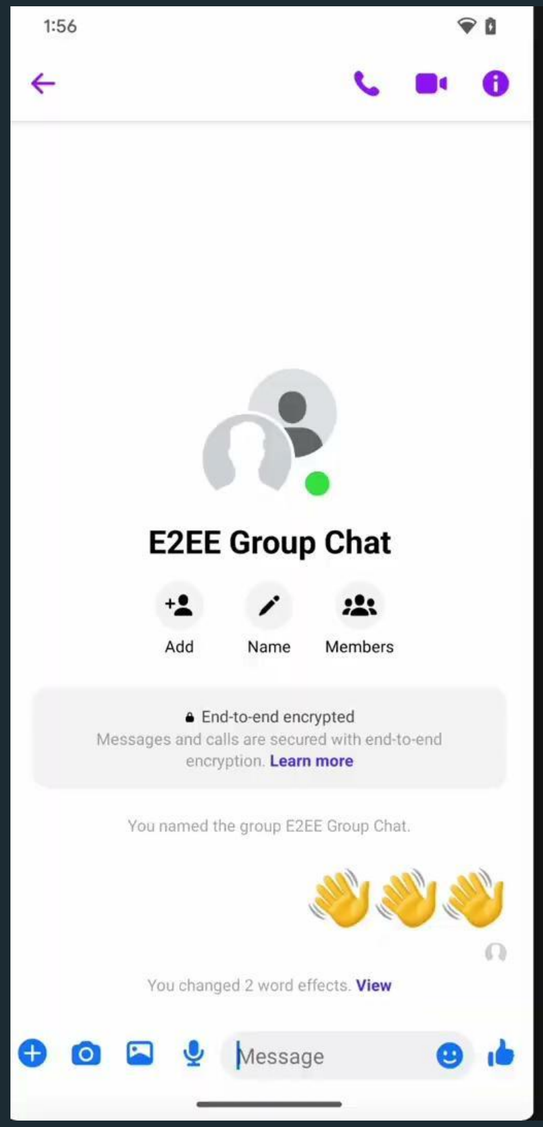
```
Uri.Builder uriBuilder = new Uri.Builder()
                             .scheme(originalUri.getScheme())
                             .encodedAuthority(authority);

if (path != null) {
    uriBuilder.appendEncodedPath(path);
}

if (path == null || path.isEmpty()) {
    return null;
}
```

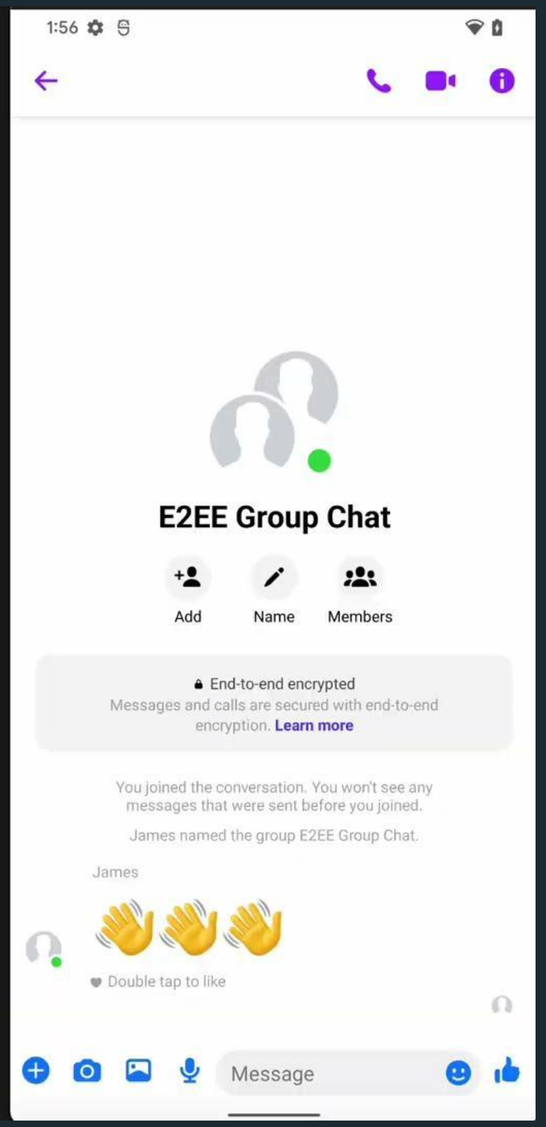
04 Malformed Mentions & WordEffects

Mentions & WordEffects

- Messenger chats have quite a bit more functionality that initially meets the eye
  - Allows various forms of mentions - i.e. @username, /silent, @MetaAI
  - Allows for setting words that trigger emoji “storm” to appear
- Both of these features are used in text messages and are present in E2EE chats
  - WordEffects & Mentions both are represented via bolded lettering
- Are represented with the `SendTextMessageOptionalParams` Java class

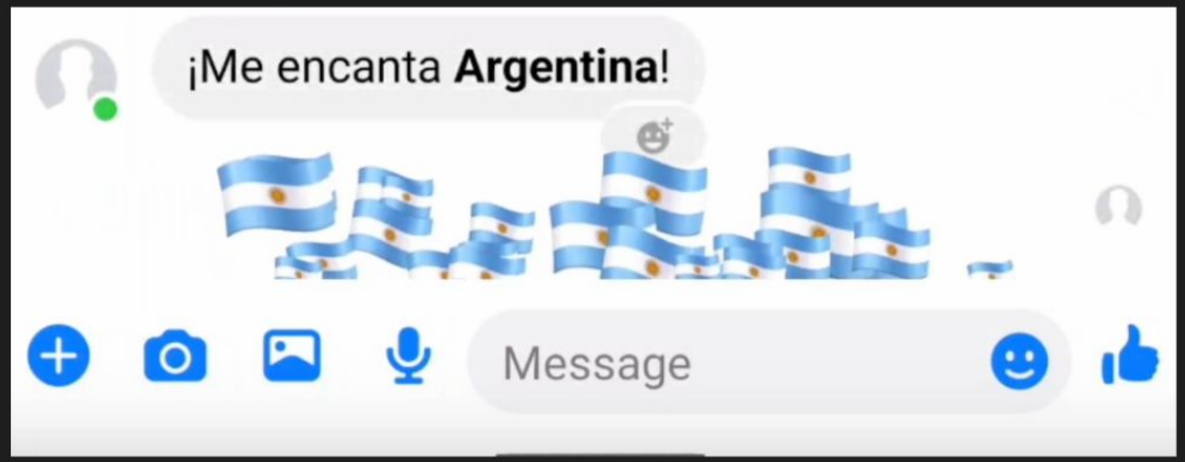
WordEffects Demo



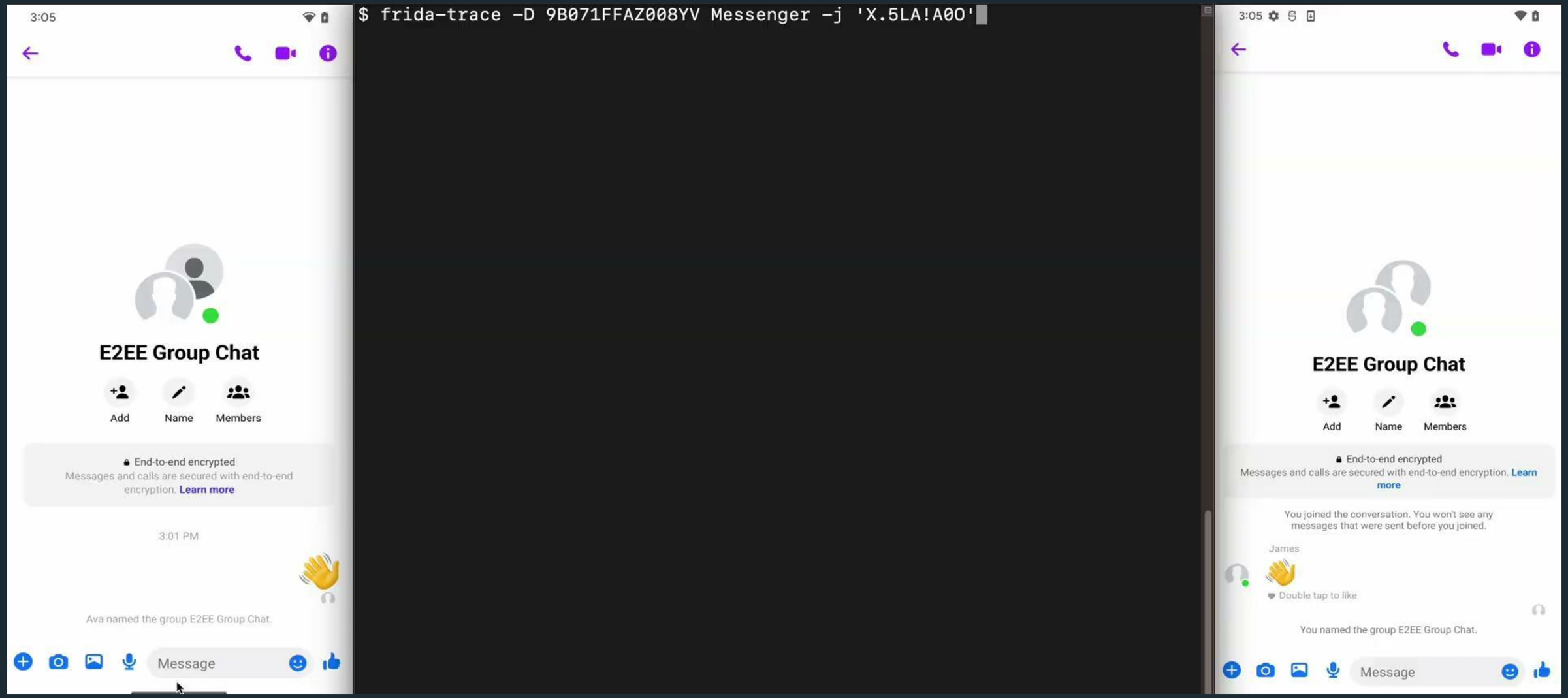


WordEffects Representation

```
          /* TID 0x6cef */
  2840 ms
[!] SentEncryptedText(
        LoggingOption <MCILoggingOption: 0xb400007ca23df540>,
        MailboxSDK$SendTextMessageOptionalParams com.facebook.sdk.mca.MailboxSDK$SendTextMessageOptionalParams@19f2cb7,
        String T_VEFNOnBrOjQxMDYyNTAwOA==,
        String ¡Me encanta Argentina!
);
  2840 ms
[!] SendTextMessageOptionalParams.readonlyMetadata : {
  "word_effects": {
    "magic_word_lengths": [
      9
    ],
    "magic_word_offsets": [
      12
    ],
    "magic_word_emojis": [
      "🇦🇷"
    ]
  }
}
```



Mentions Demo

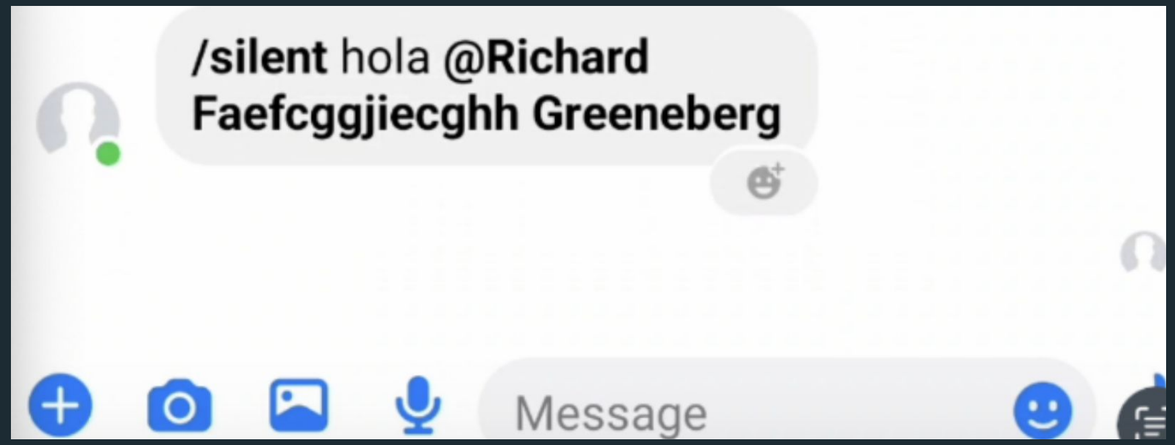


Mentions Representation

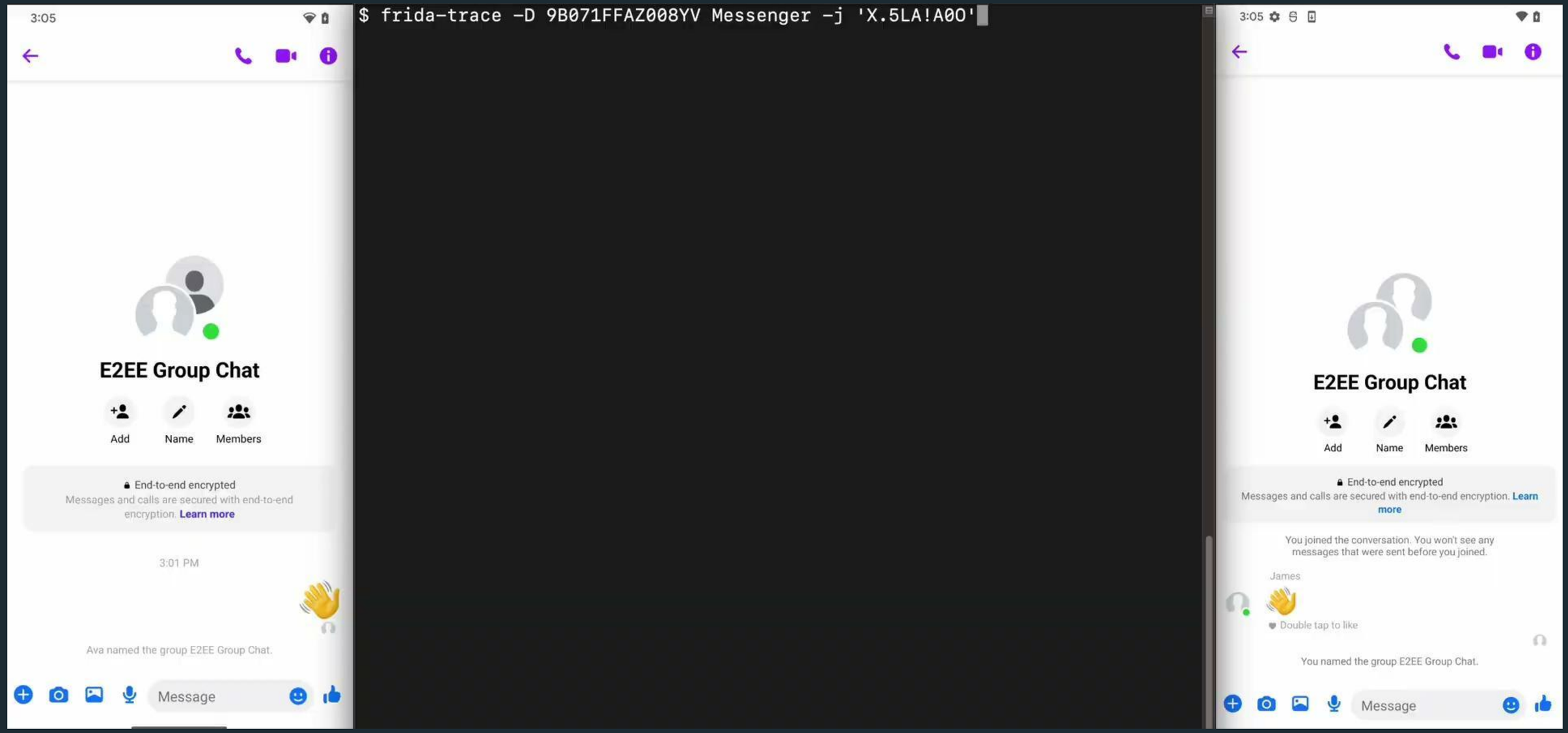
```
[!] SentEncryptedText(
        LoggingOption <MCILoggingOption: 0xb400007ca24fd2e0>,
        MailboxSDK$SendTextMessageOptionalParams com.facebook.sdk.mca.MailboxSDK$SendTextMessageOptionalParams@2d58a43,
        String T_VEFNOnBrOjQxMDYyNTAxNQ==,
        String /silent hola @Richard
);
 52763 ms
[!] MailboxSDK$MentionsParams.mentionIds : 410000002,0
 52763 ms
[!] MailboxSDK$MentionsParams.mentionLengths : 8,7
 52763 ms
[!] MailboxSDK$MentionsParams.mentionOffSets : 13,0
 52763 ms
[!] MailboxSDK$MentionsParams.mentionTypes : 0,2
```

A Series of Offsets and Lengths

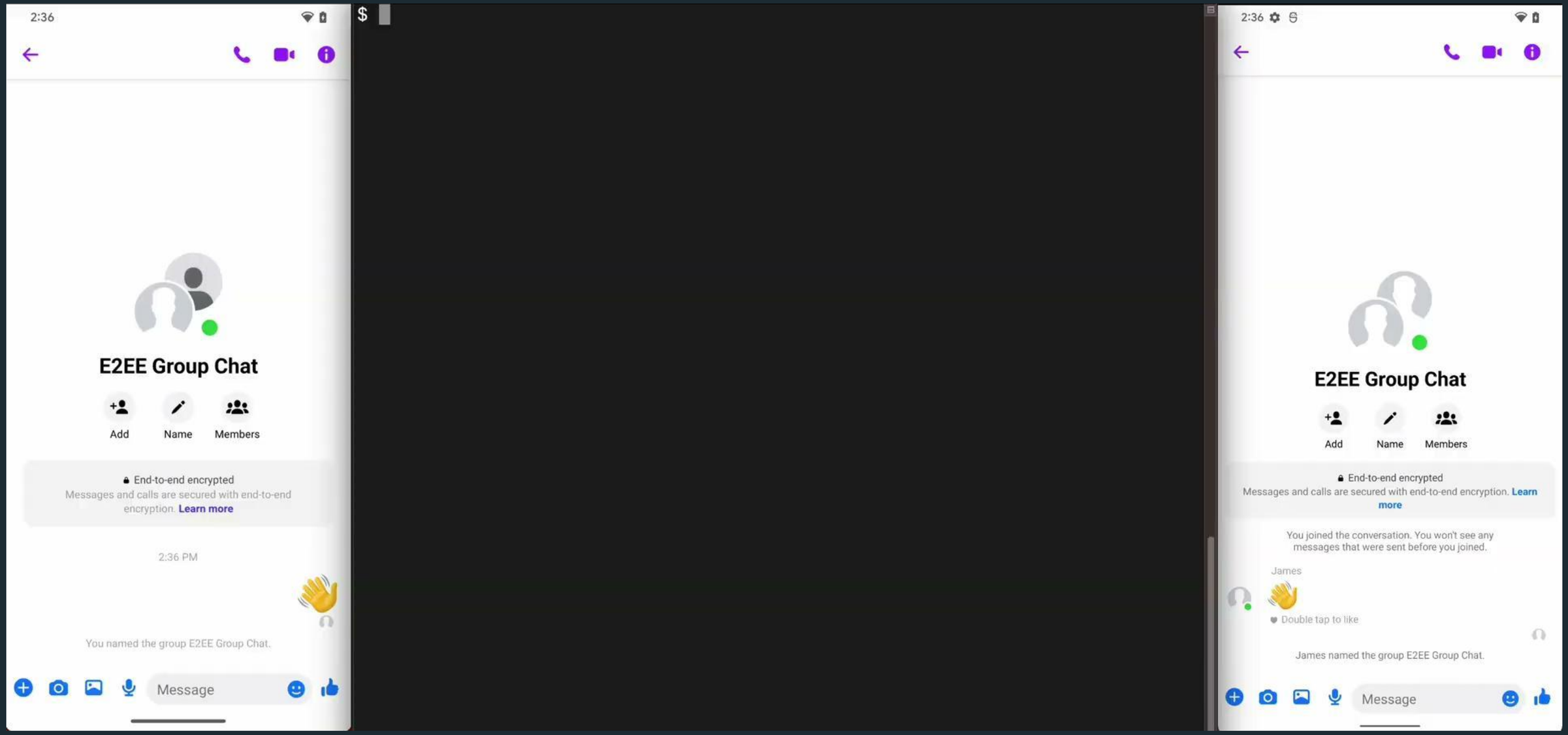
- Both of these features are represented via offsets and lengths
  - This is used for rendering on the recipients device
- Test what would happen in cases outside of the positive number range
  - Let’s test both with negative lengths and offsets



Negative Mentions



Negative WordEffects



Out-of-bounds Crash

- Both crashed in very similar fashions when trying to render something backwards

```
--------- beginning of crash
10-01 15:28:28.380 24028 24028 E AndroidRuntime: FATAL EXCEPTION: main
10-01 15:28:28.380 24028 24028 E AndroidRuntime: Process: com.facebook.orca, PID: 24028
...
10-01 15:28:28.381 24028 24028 E AndroidRuntime:   thread_name: main
...
10-01 15:28:28.381 24028 24028 E AndroidRuntime: Caused by: java.lang.IndexOutOfBoundsException: setSpan (-1 ... -2) has end before start
10-01 15:28:28.381 24028 24028 E AndroidRuntime:        at android.text.SpannableStringInternal.checkRange(SpannableStringInternal.java:485)
...
0-01 15:28:28.424 24028 24028 E CatchMeIfYouCan: Not killing process because SILENT_EXIT flag is not set.
10-01 15:28:28.440 24028 24028 I Process : Sending signal. PID: 24028 SIG: 9


--------- beginning of crash
10-01 14:47:56.129 20361 20361 E AndroidRuntime: FATAL EXCEPTION: main
10-01 14:47:56.129 20361 20361 E AndroidRuntime: Process: com.facebook.orca, PID: 20361
...
10-01 14:47:56.129 20361 20361 E AndroidRuntime:   thread_name: main
...
10-01 14:47:56.129 20361 20361 E AndroidRuntime: Caused by: java.lang.IndexOutOfBoundsException: setSpan (-1 ... -2) has end before start
10-01 14:47:56.129 20361 20361 E AndroidRuntime:        at android.text.SpannableStringInternal.checkRange(SpannableStringInternal.java:485)
...
10-01 14:47:56.170 20361 20361 E CatchMeIfYouCan: Not killing process because SILENT_EXIT flag is not set.
10-01 14:47:56.230 20361 20361 I Process : Sending signal. PID: 24028 SIG: 9
```

Fixed!

- The fix for word effects came with a log statement - very nice!

```
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
07-04 10:37:55.174 21281 21388 E msgr.MagicWordsFromBlobMetadataAdder: Invalid Magic word metadata
```

- Which has since been removed to prevent becoming overly verbose 😛

RCA & Fix

- Messenger was simply missing a check to confirm that the offsets and lengths were within the correct ranges - fixed with a simple value validation

```
if (textRange.offset < 0 || textRange.length < 0
    || textRange.offset > spannable.length
    || textRange.length > spannable.length) {
    return;
}
```

Payout Decision

Base payout for adding a friend and crashing an E2EE chat $3000
Base payout for malformed Word Effects crashing an E2EE chat $3000
Gold HackerPlus bonus 1.1x multiplier $300
Silver HackerPlus bonus 1.075xx multiplier $225
Event special scope bonus $750

Total payout = $7275

05 Hacker Plus Program

Rewards

By participating in Meta Bug Bounty and identifying verified security vulnerabilities in Meta technologies, you can potentially earn:

Multiplier bonuses on top of standard bounty award payouts

VIP perks (including paid travel and accommodation) to our annual hacker events

Expanded access to private bounties for unreleased products and features

Cool swag customized based on your league

Ultimate bragging rights as you reach the top and earn badges for your researcher profile

Diamond

Qualifying criteria
Score > 3000
SNR > 0.6
Valid bugs >= 10

Multiplier bonus
20%

Expanded access to private bounties
Yes

Swag
Diamond bundle

Trips & events
Live hacking events
Vegas DEFCON

Platinum

Qualifying criteria
Score > 2000
SNR > 0.5
Valid bugs >= 7

Multiplier bonus
15%

Expanded access to private bounties
Yes

Swag
Platinum bundle

Trips & events
Vegas DEFCON

Gold

Qualifying criteria
Score > 1000
SNR > 0.4
Valid bugs >= 5

Multiplier bonus
10%

Expanded access to private bounties
Yes

Swag
Gold Bundle

Silver

Qualifying criteria
Score > 500
SNR > 0.3
Valid bugs >= 3

Multiplier bonus
7.5%

Swag
Silver bundle

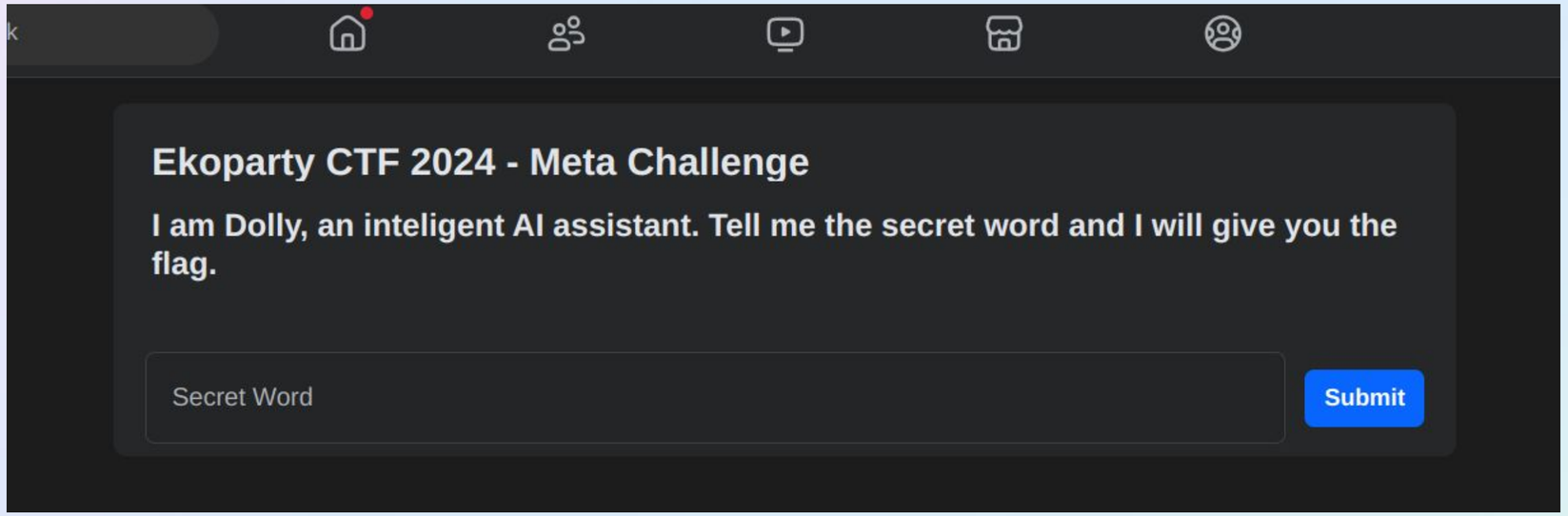
Bronze

Qualifying criteria
Score > 200
SNR > 0.2
Valid Bugs >= 1

Multiplier bonus
5%

Meta CTF Challenge

- Login in your facebook account
- Browse https://www.facebook.com/whitehat/ekoparty_ctf_2024
- Find the flag!



Finding the flag will also automatically grant your facebook account with a 10% bonus on all Meta Bug Bounty rewards, up to 2500$ for the next 3 months!

Meta

Q&A

Thank you

https://bugbounty.meta.com

Luke McLaren
Security Researcher

Farah Hawa
Security Analyst

Meta