Call, Crash, Repeat

WhatsApp Hacking

@datalocaltmp

WhatsApp Overview

Platforms, Cross-Compilation, Calling Architecture, E2EE Messaging

WhatsApp

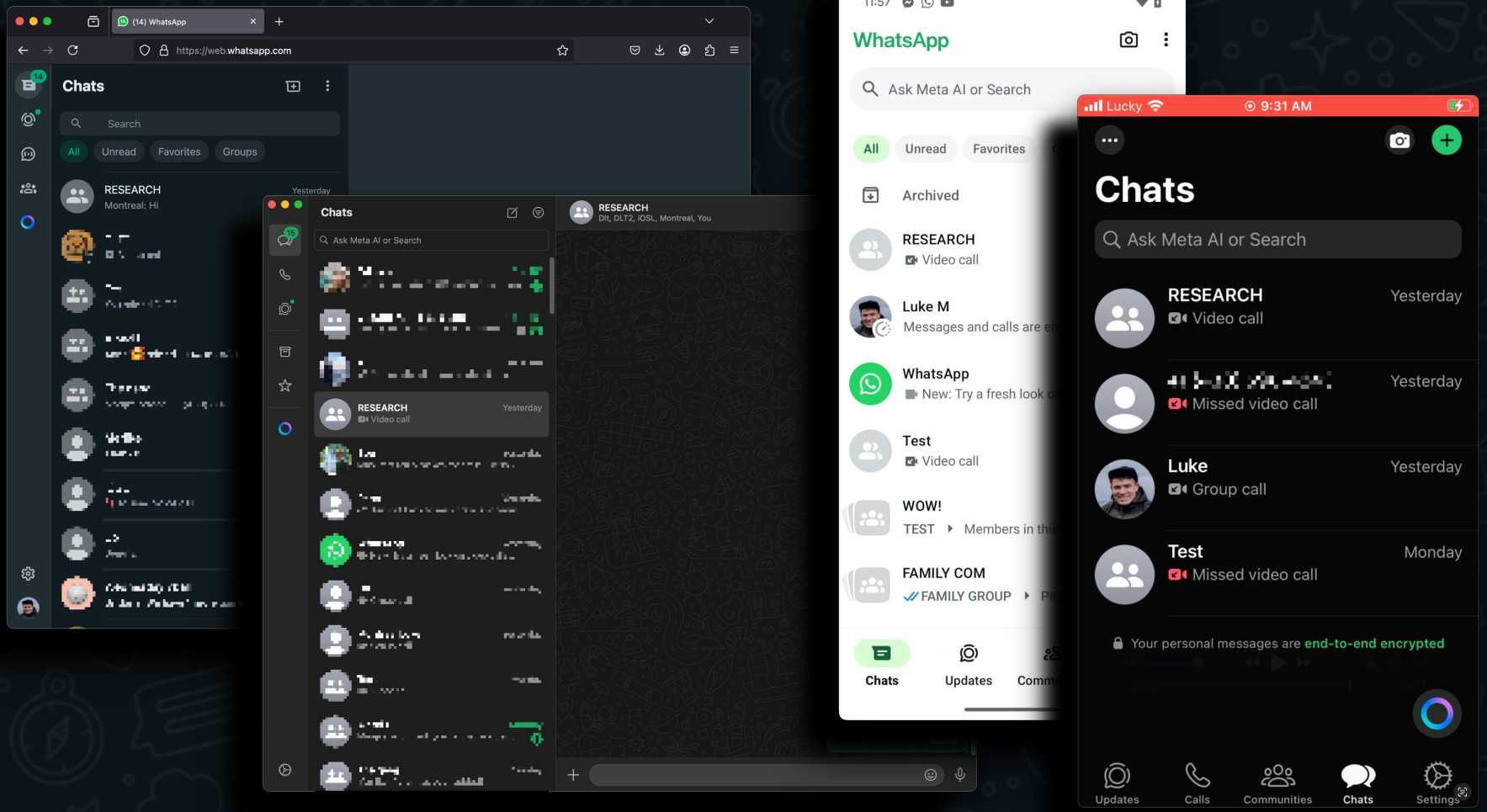
- Massive user base of over 2 billion users across 180+ countries
- Core app for personal, business, and group communications
- Implements end-to-end encryption (E2EE) messaging and calling
  - Not everything is E2EE though - signaling is unencrypted out of necessity
- Extensive features within messaging and calling activities
  - Voice & Video Calls/Event Scheduling/Voice Chats/Screen Sharing/Polls/
  - Voice & Video Notes/File Sharing/Live Location Sharing/Message transcription
  - Video Effects/Communities/etc.
- Attack surface is large and waiting for you!

WhatsApp - Why?

- With a user base of 2 billion - the “Why”s can be different for everyone
- Beaucoup bucks
  - WhatsApp Pwn2Own: 1-click RCE → $200k USD & 0-click RCE → $300k USD
  - Market price: Android/iOS RCE via WhatsApp up to $8m USD
  - Meta vs Market: Submissions need not be stable or last long - all janky bugs accepted
- Protect users from being targeted by 0-days
  - Citizenlab in Toronto captured 0-day and worked with Meta to prevent exploitation
  - Highly recommend Maddie Stone’s presentation “When Exploits Aren’t Binary”
- Get to present at RECON 2025
  - WhatsApp is a hard target and I love sharing research in the space

Platforms

- WhatsApp is available on many platforms
  - Web, MacOS & Windows, iOS & Android
- Web - Acts as a frontend mirror, relies entirely on a paired mobile device
  - Doesn’t support any calling and requires the device to be on
- MacOS & Windows - Companion apps linked to account through mobile app
  - No registration flow available - otherwise equivalent features to mobile apps
- iOS & Android - Standalone apps written in Objective-C/Java/C/C++
  - `libwhatsapp.so` - native library containing the messaging and calling logic
  - `WhatsApp (WhatsApp.ipa)` - native component containing the messaging and calling logic
- `WhatsApp & libwhatsapp.so` represent the some of the most interesting space for RCE research


(14) WhatsApp
https://web.whatsapp.com
Chats
Search
All
Unread
Favorites
Groups
RESEARCH
Montreal: Hi
Yesterday
Chats
Ask Meta AI or Search
RESEARCH
Dit, DLT2, iOSL, Montreal, You
RESEARCH
Video call
Yesterday
Pixel 4
11:57
WhatsApp
Ask Meta AI or Search
All
Unread
Favorites
Archived
RESEARCH
Video call
Luke M
Messages and calls are
WhatsApp
New: Try a fresh look
Test
Video call
WOW!
TEST
Members in this
FAMILY COM
FAMILY GROUP
Chats
Updates
Lucky
9:31 AM
Chats
Ask Meta AI or Search
RESEARCH
Video call
Yesterday
Missed video call
Yesterday
Luke
Group call
Yesterday
Test
Missed video call
Monday
Your personal messages are end-to-end encrypted
Updates
Calls
Communities
Chats
Settings

Cross-Compilation

- C++ codebase is cross-compiled for many different platforms
  - Native vulnerabilities in one platform can appear in others
- Compiled binaries are stripped of symbols
  - However the extensive logging in WhatsApp will often reveal source files and function names
  - Native logs are written to the following files for Android, iOS, MacOS:
    - `/data/data/com.whatsapp/files/Logs/whatsapp.log`
    - `/private/var/mobile/Containers/Shared/AppGroup/<APP UUID>/Logs/`
    - `~/Library/Group/Containers/group.net.whatsapp.WhatsApp.shared/Logs/`
- While binaries are compiled from the same source structures can change between platforms
  - Bug #2 should illustrate changes to structure offsets and how to recognize that in crashes

Android Build

```
if (uVar11 == 0) {
  FUN_0057c860("wa_call_signaling_xml.cc",
               "error filling silence reason, can
}
else {
  pvVar3 = (void *)((ulong)&local_118 | 1);
  if ((local_118 & 1) != 0) {
    pvVar3 = local_108;
  }
  iVar10 = FUN_008f0de4(pvVar3);
  *(int *)((long)param_3 + 0xa45b8) = iVar10;
  if (iVar10 == 8) {
    FUN_008f80ac(&local_138,&local_120,"is_first_
    bVar7 = FUN_008bb148(&local_138,"1");
    *(byte *)((long)param_3 + 0xa45bc) = bVar7 &
    if ((local_138 & 1) != 0) {
      operator.delete(local_128);
    }
  }
  else if (iVar10 == 0) {
    FUN_0057c928("wa_call_signaling_xml.cc",
                 "error filling unknown silence r
  }
}
```

Mac Build

```
if (uVar15 == 0) {
  pcVar10 = "wa_call_signaling_xml.cc";
  FUN_102b4689c("wa_call_signaling_xml.cc",
                "error filling silence reason, cannot
}
else {
  FUN_102e0339c();
  pcVar6 = extraout_x9_01;
  if (cVar2 == cVar1) {
    pcVar6 = extraout_x8_01;
  }
  if (pcVar6 != (char *)0x0) {
    for (lVar14 = 1; lVar14 != 9; lVar14 = lVar14 + 1
      pcVar10 = pcVar6;
      _strcmp(pcVar6,(&PTR_s__103c776b8)[lVar14]);
      if ((int)pcVar10 == 0) {
        iVar3 = (int)lVar14;
        *(int *)(param_3 + 0xa44e8) = iVar3;
        if (iVar3 == 8) {
          FUN_102ded388();
          FUN_102ae5870();
          pcVar10 = (char *)&stack0x00000150;
          FUN_1029bae7c(pcVar10,"1");
          *(char *)(param_3 + 0xa44ec) = (char)pcVar1
          FUN_102df4460();
          goto LAB_102addbe4;
        }
        if (iVar3 == 0) goto LAB_102addbb4;
        goto LAB_102addbe4;
      }
    }
  }
  *(undefined4 *)(param_3 + 0xa44e8) = 0;

  pcVar10 = "wa_call_signaling_xml.cc";
  FUN_102b46920("wa_call_signaling_xml.cc",
                "error filling unknown silence reason
}
```

iOS Build

```
if (uVar14 == 0) {
  pcVar9 = s_wa_call_signaling_xml.cc_1046a5d41;
  FUN_1027e4a34(s_wa_call_signaling_xml.cc_1046a5d41,
                s_error_filling_silence_reason,_ca_1046a777e);
}
else {
  psVar4 = in_stack_00000168;
  if (-1 < (char)in_stack_0000017c._3_1_) {
    psVar4 = (string *)&stack0x00000168;
  }
  if (psVar4 != (string *)0x0) {
    for (lVar13 = 1; lVar13 != 9; lVar13 = lVar13 + 1) {
      iVar3 = _strcmp((char *)psVar4,(&PTR_s__103592878)[lVar13]);
      pcVar9 = (char *)CONCAT44(extraout_var,iVar3);
      if (iVar3 == 0) {
        iVar3 = (int)lVar13;
        *(int *)(param_3 + 0xa45a8) = iVar3;
        if (iVar3 == 8) {
          FUN_102ea4dec();
          FUN_102782c70();
          pcVar9 = (char *)&stack0x00000150;
          FUN_10264f320(pcVar9,s_1_1046eebf7);
          param_3[0xa45ac] = SUB81(pcVar9,0);
          FUN_102ead128();
          goto LAB_10277afd8;
        }
        if (iVar3 == 0) goto LAB_10277afa8;
        goto LAB_10277afd8;
      }
    }
  }
  *(undefined4 *)(param_3 + 0xa45a8) = 0;

  pcVar9 = s_wa_call_signaling_xml.cc_1046a5d41;
  FUN_1027e4ab8(s_wa_call_signaling_xml.cc_1046a5d41,
                s_error_filling_unknown_silence_re_1046a77ad);
}
```

Calling Architecture

- Call Establishment and Maintenance
  - XMPP Signaling (closed-source)
    - Code developed by WhatsApp for WhatsApp (`wa_call_signaling_xml.cc`)
    - Responsible for xmpp signaling to initiate/accept/terminate calls
    - Signals for heartbeats, latency stats, screen sharing, muting, enabling/disabling video
- Network Packet Management
  - PJSIP (semi-open-source)
    - WhatsApp has made heavy closed-source modifications and improvements
    - Manages construction and encryption of the RTP/RTCP media streams
    - CVE-2019-3568 → 1400+ phones exploited
- Voice and Video Data Encoding/Decoding
  - WebRTC (open-source)
    - Responsible for voice and video media data of various encoding types - Opus/MLow/Av1/h264
    - Managing bandwidth usage estimation, echo cancellation, congestion control

XMPP Attributes (Light Foreshadowing…)

- `<Offer>` Silence Attribute
- Prevents Call Notification
- Silence available for:
  - Voice Chat (VC).
  - Scheduled Calls.
  - Group Setting.
  - Etc.

```
if (uVar11 == 0) {
  FUN_0057c860("wa_call_signaling_xml.cc",
               "error filling silence reason, cannot
}
else {
  pvVar3 = (void *)((ulong)&local_118 | 1);
  if ((local_118 & 1) != 0) {
    pvVar3 = local_108;
  }
  iVar10 = FUN_008f0de4(pvVar3);
  *(int *)((long)param_3 + 0xa45b8) = iVar10;
  if (iVar10 == 8) {
    FUN_008f80ac(&local_138,&local_120,"is_first_wave
    bVar7 = FUN_008bb148(&local_138,"1");
    *(byte *)((long)param_3 + 0xa45bc) = bVar7 & 1;
    if ((local_138 & 1) != 0) {
      operator.delete(local_128);
    }
  }
  else if (iVar10 == 0) {
    FUN_0057c928("wa_call_signaling_xml.cc",
                 "error filling unknown silence reas
  }
}
```

```
char * FUN_008f0de4(char *param_1)

{
  int iVar1;

  if (param_1 != (char *)0x0) {
    iVar1 = strcmp(param_1,"scheduled");
    if (iVar1 == 0) {
      param_1 = (char *)0x1;
    }
    else {
      iVar1 = strcmp(param_1,"privacy");
      if (iVar1 == 0) {
        param_1 = (char *)0x2;
      }
      else {
        iVar1 = strcmp(param_1,"lightweight");
        if (iVar1 == 0) {
          param_1 = (char *)0x3;
        }
        else {
          iVar1 = strcmp(param_1,"screensharing");
          if (iVar1 == 0) {
            param_1 = &DAT_00000004;
          }
          else {
            iVar1 = strcmp(param_1,"group_setting");
            if (iVar1 == 0) {
              param_1 = &DAT_00000005;
            }
            else {
              iVar1 = strcmp(param_1,"vc_init");
              if (iVar1 == 0) {
                param_1 = &DAT_00000006;
              }
              else {
                iVar1 = strcmp(param_1,"vc_wave");
                if (iVar1 == 0) {
                  param_1 = &DAT_00000007;
                }
                else {
                  iVar1 = strcmp(param_1,"vc_wave_all");
                  if (iVar1 == 0) {
                    param_1 = (char *)((ulong)(iVar1 == 0) << 3);
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  return param_1;
}
```

PJSIP Presence (Heavy Foreshadowing…)

```
1
2  void FUN_0053fdb8(long *param_1)
3
4  {
5    long lVar1;
6    int *piVar2;
7    int local_3c;
8    int local_2c;
9    timespec local_28;
10   long local_18;
11
12   lVar1 = tpidr_el0;
13   local_18 = *(long *)(lVar1 + 0x28);
14   memset(&local_28,0,0x10);
15   local_3c = clock_gettime(7,&local_28);
16   if (local_3c != 0) {
17     local_3c = clock_gettime(1,&local_28);
18   }
19   if (local_3c == 0) {
20     local_28.tv_nsec = local_28.tv_sec * 1000000000 + local_28.tv_nsec;
21     if (DAT_00e510a0 == 0) {
22       DAT_00e510a0 = local_28.tv_nsec;
23     }
24     *param_1 = local_28.tv_nsec - DAT_00e510a0;
25     local_2c = 0;
26   }
27   else {
28     piVar2 = (int *)__errno();
29     if (*piVar2 == 0) {
30       local_2c = -1;
31     }
32     else {
33       piVar2 = (int *)__errno();
34       if (*piVar2 == 0) {
35         local_2c = 0;
36       }
37       else {
38         piVar2 = (int *)__errno();
39         local_2c = *piVar2 + 120000;
40       }
41     }
42   }
43   lVar1 = tpidr_el0;
44   if (*(long *)(lVar1 + 0x28) != local_18) {
45                     /* WARNING: Subroutine does not return */
46     __stack_chk_fail(local_2c);
47   }
48   return;
49 }
50
```

a85f793 · pjproject / pjlib / src / pj / os_timestamp_posix.c · ↑ Top

Code · Blame · 336 lines (273 loc) · 7.56 KB · Raw

```
227
228  PJ_DEF(pj_status_t) pj_get_timestamp(pj_timestamp *ts)
229  {
230      struct timespec tp;
231      int err = -1;
232
233  #if defined(ANDROID_ALARM_GET_TIME)
234      if (s_alarm_fd == -1) {
235          int fd = open("/dev/alarm", O_RDONLY);
236          if (fd >= 0) {
237              s_alarm_fd = fd;
238              pj_atexit(&close_alarm_fd);
239          }
240      }
241
242      if (s_alarm_fd != -1) {
243          err = ioctl(s_alarm_fd,
244                ANDROID_ALARM_GET_TIME(ANDROID_ALARM_ELAPSED_REALTIME), &tp);
245      }
246  #elif defined(CLOCK_BOOTTIME)
247      err = clock_gettime(CLOCK_BOOTTIME, &tp);
248  #endif
249
250      if (err != 0) {
251          /* Fallback to CLOCK_MONOTONIC if /dev/alarm is not found, or
252           * getting ANDROID_ALARM_ELAPSED_REALTIME fails, or
253           * CLOCK_BOOTTIME fails.
254           */
255          err = clock_gettime(CLOCK_MONOTONIC, &tp);
256      }
257
258      if (err != 0) {
259          return PJ_RETURN_OS_ERROR(pj_get_native_os_error());
260      }
261
262      ts->u64 = tp.tv_sec;
263      ts->u64 *= NSEC_PER_SEC;
264      ts->u64 += tp.tv_nsec;
265
266      return PJ_SUCCESS;
267  }
```

E2EE Messaging

- All messaging on WhatsApp is encrypted end-to-end
- Skip the underlying cryptographic algorithms or key sharing mechanism
  - WhatsApp Encryption Overview Technical Whitepaper
- Messages are serialized to a protobuf and then encrypted
- On Android - Messages can be inspected and modified using Frida
  - `libcrypto.so - EVP_EncryptUpdate`
  - Sharing script that intercepts and hexdumps message contents pre-encryption.

```
int EVP_EncryptUpdate(EVP_CIPHER_CTX *ctx, unsigned char *out, int *outl,
                      const unsigned char *in, int inl)
```

Time for bugs…

Event URL Validation Bug

Intercepting and modifying E2EE messages

WhatsApp
Official WhatsApp account

Sun, Nov 10

WhatsApp

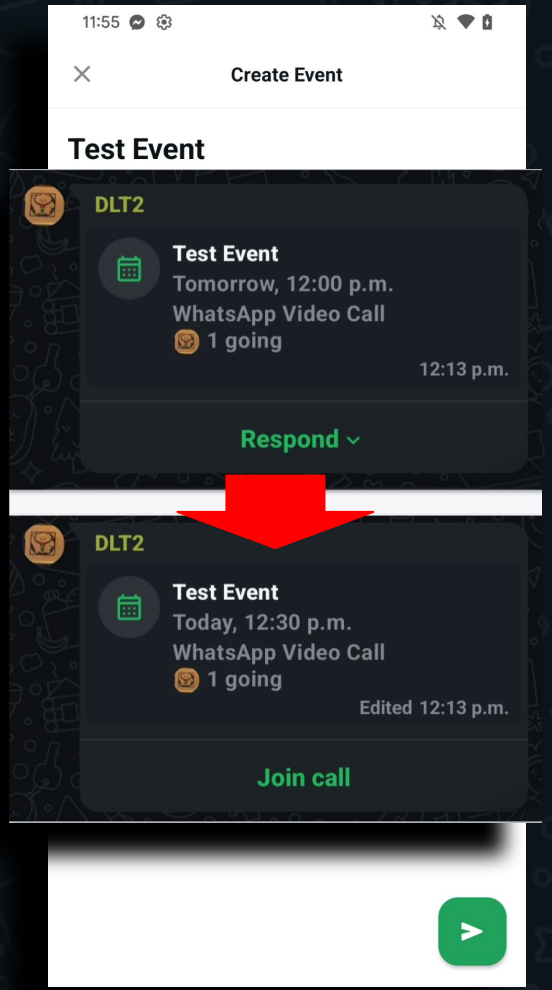
New: Make plans happen with events 🗓️
Now you can send an invitation to the chat instead of going back and forth about the when and where.

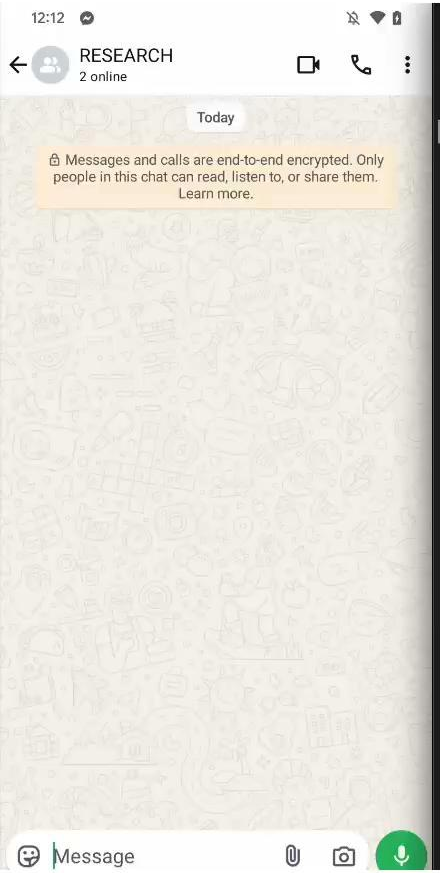
Tap 📎 and select Event to create one. Then the group can RSVP to the invite and get details without ever leaving the chat. 4:11 PM

Learn more

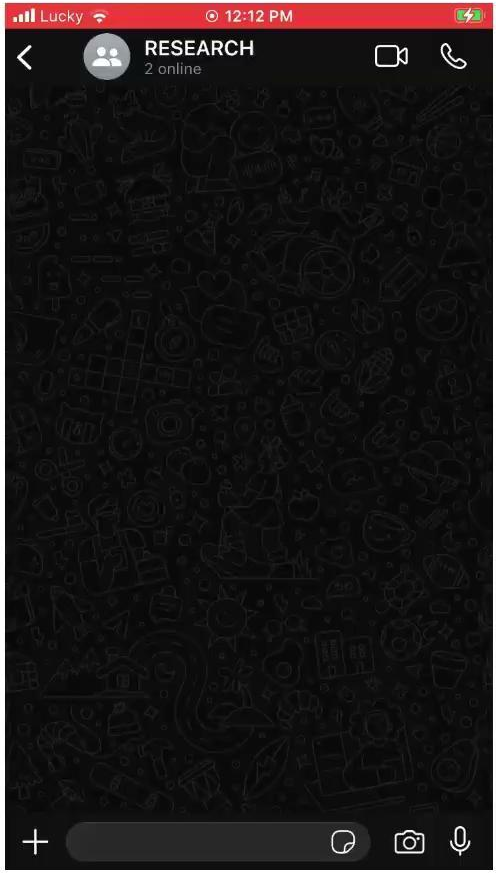
Event Messages

- Events are available in group chats
- Provides ability to set a time and location for event
  - Users can RSVP to the event
- Interestingly - we can associate calls with the event
  - Choice of event call being voice or video
  - About 10 minutes before the event WhatsApp creates the call
  - “Respond” becomes “Join Call”
  - Clicking “Join Call” navigates to a deep link call url
- “Allow Guests” is new
  - Unknown to myself and unresearched





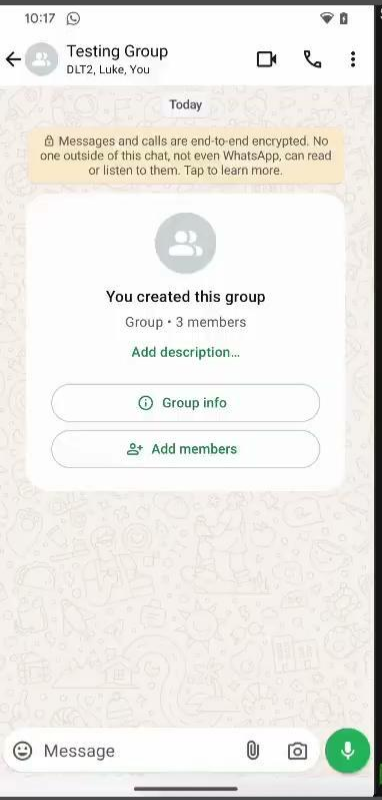

```
(env) dev@devs-Mac-mini res1 % python3 event_poc.py
[*] Starting PoC for event link arbitrary url ...
[!] Found: /system/lib64/libcrypto.so
[!] Found: /apex/com.android.art/lib64/libcrypto.so
[!] Found: /apex/com.android.conscrypt/lib64/libcrypto.so
```

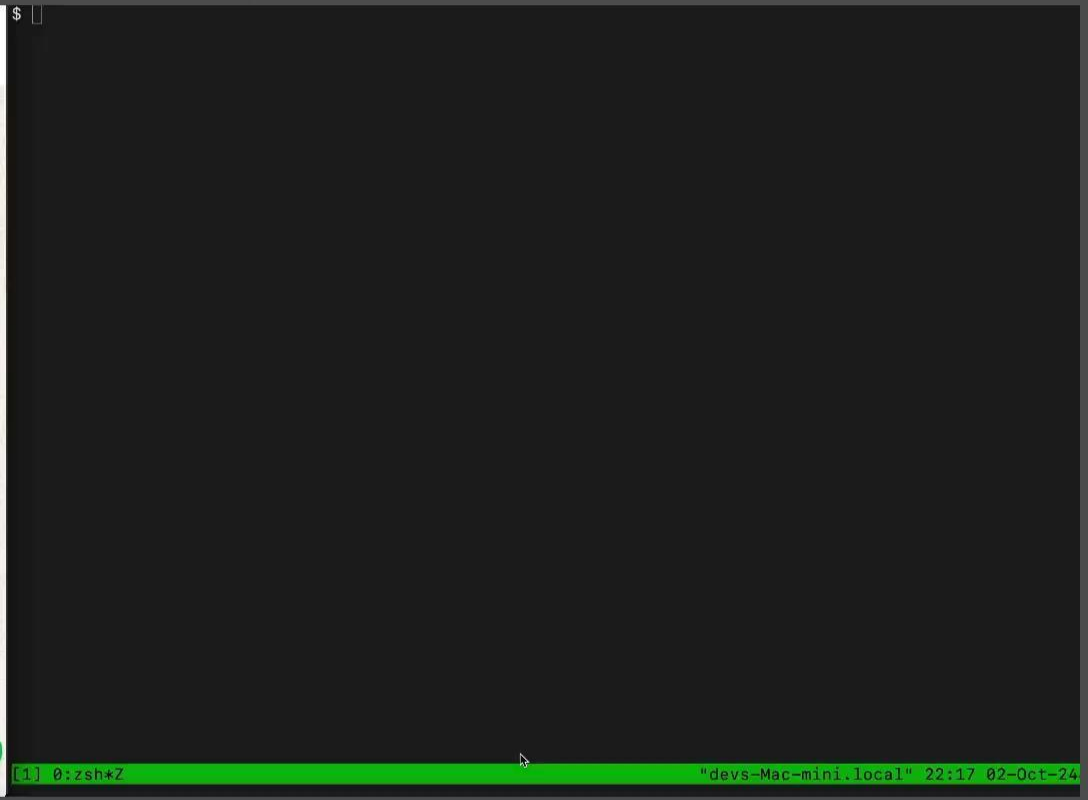


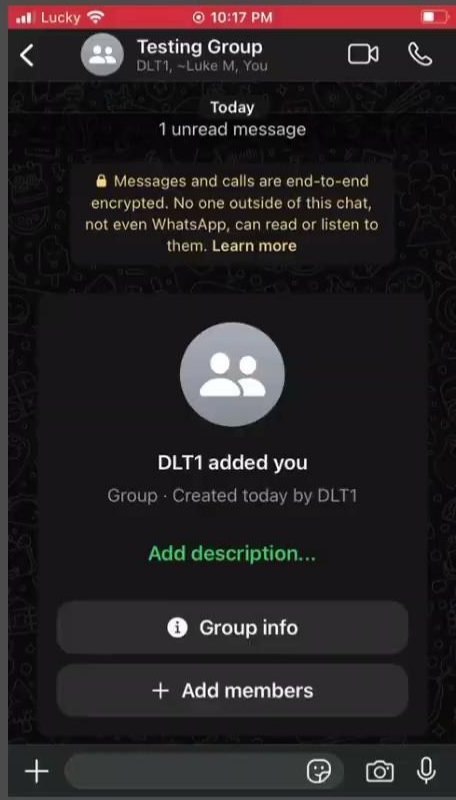
Can you guess the bug?

Android w/ Frida

iOS







Bug & Fix

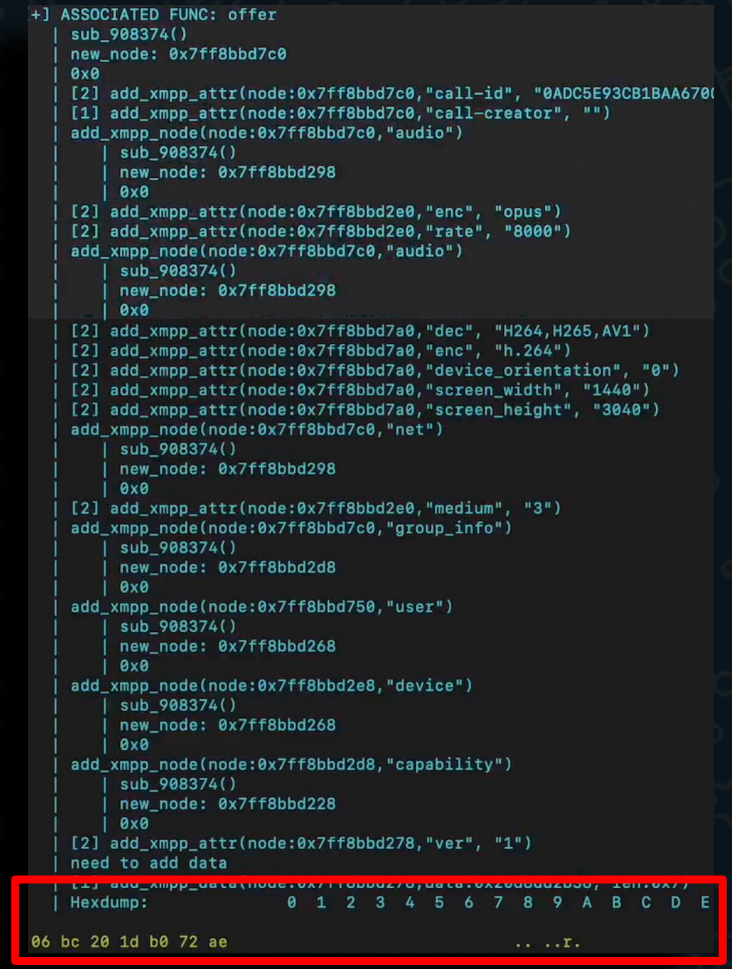
- **Bug**: The iOS client didn’t validate Call URLs before navigating to them
- **Impact:** Vulnerability helps phish end users or trigger deeper exploitation through other deep link urls
- **Fix**: Validation logic implemented to ensure links are either:
  - https://call.whatsapp.com/video/ or https://call.whatsapp.com/voice/

Capabilities Data Bug

Modifying stanza attributes to trigger PJSIP bugs

Starting a WhatsApp Call

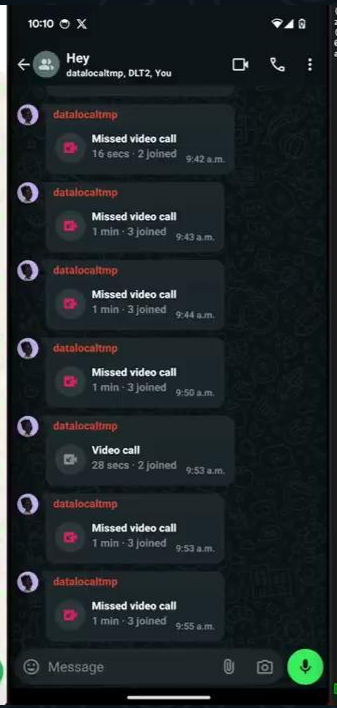
- <offer> stanza initiates a call
  - Contains call & caller metadata
  - Sub stanzas include:
    - <audio> & <video>
      - Encodings and sizes of screens
    - <group_info>
      - Contains a list of <users> stanzas
    - <capability>
      - User capabilities
- <capability> consists of raw data and version
- <capability “data”=”........” “version”=1>

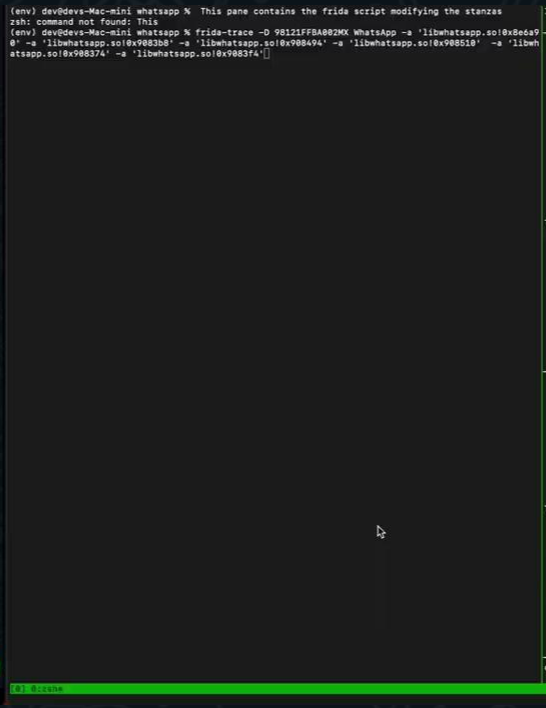


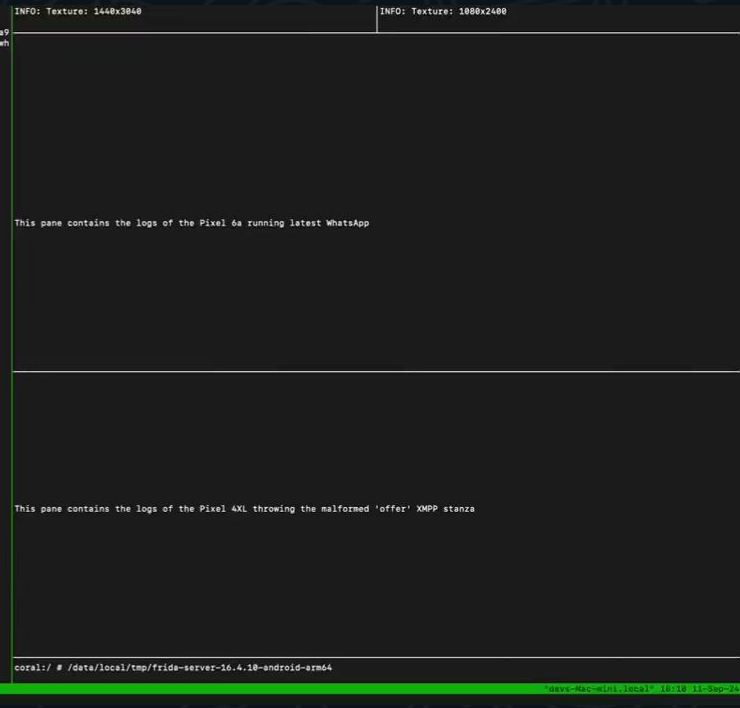
Time to mangle raw data

What happens when we modify capability blobs?


10:10
Hey
DLT, DLT2, You
Video call
1 min · 3 joined
9:37 AM
Video call
11 sec · 2 joined
9:39 AM
Video call
16 sec · 2 joined
9:42 AM
Video call
1 min · 3 joined
9:43 AM
Video call
1 min · 3 joined
9:44 AM
Video call
1 min · 2 joined
9:50 AM
Video call
28 sec · 2 joined
9:53 AM
Video call
1 min · 3 joined
9:53 AM
Video call
1 min · 1 joined
9:55 AM
Message


10:10
Hey
datalocaltmp, DLT2, You
datalocaltmp
Missed video call
16 secs · 2 joined
9:42 a.m.
datalocaltmp
Missed video call
1 min · 3 joined
9:43 a.m.
datalocaltmp
Missed video call
1 min · 3 joined
9:44 a.m.
datalocaltmp
Missed video call
1 min · 3 joined
9:50 a.m.
datalocaltmp
Video call
28 secs · 2 joined
9:53 a.m.
datalocaltmp
Missed video call
1 min · 3 joined
9:53 a.m.
datalocaltmp
Missed video call
1 min · 3 joined
9:55 a.m.
Message


(env) dev@devs-Mac-mini whatsapp % This pane contains the frida script modifying the stanzas
zsh: command not found: This
(env) dev@devs-Mac-mini whatsapp % frida-trace -D 98121FFBA002MX WhatsApp -a 'libwhatsapp.so!0x8e6a90' -a 'libwhatsapp.so!0x9083b8' -a 'libwhatsapp.so!0x908494' -a 'libwhatsapp.so!0x908510' -a 'libwhatsapp.so!0x908374' -a 'libwhatsapp.so!0x9083f4'


INFO: Texture: 1440x3040
INFO: Texture: 1080x2400
This pane contains the logs of the Pixel 6a running latest WhatsApp
This pane contains the logs of the Pixel 4XL throwing the malformed 'offer' XMPP stanza
coral:/ # /data/local/tmp/frida-server-16.4.10-android-arm64

Crash - Android

```
Cmdline: com.whatsapp
pid: 7803, tid: 7884, name: VoIP Signaling  >>> com.whatsapp <<<
uid: 10233
signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x00000000000036b0
    x0  0000000000000000  x1  00000076aca719b8  x2  0000000113979ff  x3  000000000087057c
    x4  00000076aca719c8  x5  0000000000000000  x6  0000000000000000  x7  b4000076d48661b0
    x8  0000001091d90cdf  x9  00000000000036b0  x10 ffffffff991d7991  x11 00000000000082fd
    x12 0000000000000001  x13 0000000000000001  x14 ffffffffc4653600  x15 000000000001103b
    x16 001c2987886b3e02  x17 0000015bdd5950df  x18 000000761f058000  x19 00000000db9c9778
    x20 000000021721a6100  x21 0000000000000008  x22 0000000000000000  x23 000000764915dbf7
    x24 00000076c0e00880  x25 00000076aca725a8  x26 0000000010380011  x27 00000000000000e4
    x28 00000076aca72460  x29 00000076aca719d0
    lr  000000763963fdfc  sp  00000076aca71990  pc  000000763963fed8  pst 0000000080000000

backtrace:
      #00 pc 000000000043fed8  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #01 pc 0000000000762788  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #02 pc 000000000075ec7c  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #03 pc 000000000075eb1c  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #04 pc 000000000091a540  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #05 pc 00000000009196a8  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #06 pc 0000000000917820  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #07 pc 00000000008d91d8  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #08 pc 00000000008d3300  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #09 pc 00000000008d358c  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #10 pc 00000000008e9f44  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #11 pc 00000000008ea2d4  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #12 pc 000000000048a104  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (BuildId: 36ba7fb6c494905f3e6a24173482b2ba1019f801)
      #13 pc 0000000000393a04  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so (Java_com_whatsapp_voipcalling_Voip_nativeHandleIncomingSignalingXmpp+156)
```

Crash - iOS

```
Identifier:          net.whatsapp.WhatsApp
Version:             24.17.78 (635367251)
AppStoreTools:       15F31e
AppVariant:          1:iPhone8,1:15
Code Type:           ARM-64 (Native)
Role:                Foreground
Parent Process:      launchd [1]
Coalition:           net.whatsapp.WhatsApp [410]

Date/Time:           2024-09-11 09:37:59.0628 -0400
Launch Time:         2024-09-11 09:35:36.1556 -0400
OS Version:          iPhone OS 15.5 (19F77)
Release Type:        User
Baseband Version:    9.61.00
Report Version:      104

Exception Type:  EXC_BAD_ACCESS (SIGSEGV)
Exception Subtype: KERN_INVALID_ADDRESS at 0x00000000000036b0
Exception Codes: 0x0000000000000001, 0x00000000000036b0
```

Crash - MacOS

```
Identifier:             net.whatsapp.WhatsApp
Version:                24.18.77 (639364437)
App Item ID:            310633997
App External ID:        868877688
Code Type:              ARM-64 (Native)
Parent Process:         launchd [1]
User ID:                501

Date/Time:              2024-09-11 11:42:48.6829 -0400
OS Version:             macOS 13.6.7 (22G720)
Report Version:         12
Anonymous UUID:         DB56B432-CB06-366F-CC5B-B26FEF39E19E

Sleep/Wake UUID:        9DBF84FE-F0DD-4503-A054-480CA07C36C9

Time Awake Since Boot: 290000 seconds
Time Since Wake:        227888 seconds

System Integrity Protection: disabled

Crashed Thread:         23  net_event_thread

Exception Type:         EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:        KERN_INVALID_ADDRESS at 0x0000000000003728
Exception Codes:        0x0000000000000001, 0x0000000000003728
```

Reverse Engineering

```
backtrace:
      #00 pc 0000000000043fed8  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so
      #01 pc 0000000000082788  /data/data/com.whatsapp/files/decompressed/libs.spo/libwhatsapp.so
      #02 pc 0006
      #03 pc 0006
      #04 pc 0
      #05 pc 0
      #06 pc 0
      #07 pc 0
      #08 pc 0
      #09 pc 0
      #10 pc 0
      #11 pc 0
      #12 pc 0
      #13 pc 0
```

libwhatsapp.so | *libwhatsapp-2.24.17.79.so

```
0053fea8 e8 07 40 f9     ldr        x8,[sp, #local_48]
0053feac 89 48 00 d0     adrp       x9,0xe51000
0053feb0 29 81 02 91     add        x9,x9,#0xa0
0053feb4 28 01 00 f9     str        x8,[x9]=>DAT_00e510a0                    = ??

                     LAB_0053feb8                                    XREF[1]:     0053fea4
0053feb8 88 48 00 d0     adrp       x8,0xe51000
0053febc 08 81 02 91     add        x8,x8,#0xa0
0053fec0 08 01 40 f9     ldr        x8,[x8]=>DAT_00e510a0                    = ??
0053fec4 e9 07 40 f9     ldr        x9,[sp, #local_48]
0053fec8 28 01 08 cb     sub        x8,x9,x8
0053fecc e8 07 00 f9     str        x8,[sp, #local_48]
0053fed0 e8 07 40 f9     ldr        x8,[sp, #local_48]
0053fed4 e9 0f 40 f9     ldr        x9,[sp, #local_18]
0053fed8 28 01 00 f9     str        x8,[x9]
0053fedc bf 43 1e b8     stur       wzr,[x29, #local_2c]

                     LAB_0053fee0                                    XREF[1]:     0053fe64
0053fee0 a0 43 5e b8     ldur       w0,[x29, #local_2c]
0053fee4 48 d0 3b d5     mrs        x8,tpidr_el0
0053fee8 08 15 40 f9     ldr        x8,[x8, #0x28]
0053feec a9 83 5f f8     ldur       x9,[x29, #local_18]
0053fef0 1f 01 09 eb     cmp        x8,x9
0053fef4 81 00 00 54     b.ne       LAB_0053ff04
0053fef8 fd 7b 44 a9     ldp        x29=>local_10,x30,[sp, #0x40]
0053fefc ff 43 01 91     add        sp,sp,#0x50
0053ff00 c0 03 5f d6     ret

                     LAB_0053ff04                                    XREF[1]:     0053fef4
0053ff04 f7 d5 22 94     bl         <EXTERNAL>::__stack_chk_fail                 undef
                         -- Flow Override: CALL_RETURN (CALL_TERMINATOR)

                         *************************************************************
```

```
void FUN_0053fdb8(long *param_1)

{
  long lVar1;
  int *piVar2;
  int local_3c;
  int local_2c;
  timespec local_28;
  long local_18;

  lVar1 = tpidr_el0;
  local_18 = *(long *)(lVar1 + 0x28);
  memset(&local_28,0,0x10);
  local_3c = clock_gettime(7,&local_28);
  if (local_3c != 0) {
    local_3c = clock_gettime(1,&local_28);
  }
  if (local_3c == 0) {
    local_28.tv_nsec = local_28.tv_sec * 1000000000 + local_28.tv_nsec;
    if (DAT_00e510a0 == 0) {
      DAT_00e510a0 = local_28.tv_nsec;
    }
    *param_1 = local_28.tv_nsec - DAT_00e510a0;
    local_2c = 0;
  }
  else {
    piVar2 = (int *)__errno();
    if (*piVar2 == 0) {
      local_2c = -1;
    }
    else {
      piVar2 = (int *)__errno();
      if (*piVar2 == 0) {
        local_2c = 0;
      }
      else {
        piVar2 = (int *)__errno();
        local_2c = *piVar2 + 120000;
      }
    }
  }
  lVar1 = tpidr_el0;
  if (*(long *)(lVar1 + 0x28) != local_18) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail(local_2c);
  }
  return;
}
```

Setup and Payoff

```
1
2  void FUN_0053fdb8(long *param_1)
3
4  {
5    long lVar1;
6    int *piVar2;
7    int local_3c;
8    int local_2c;
9    timespec local_28;
10   long local_18;
11
12   lVar1 = tpidr_el0;
13   local_18 = *(long *)(lVar1 + 0x28);
14   memset(&local_28,0,0x10);
15   local_3c = clock_gettime(7,&local_28);
16   if (local_3c != 0) {
17     local_3c = clock_gettime(1,&local_28);
18   }
19   if (local_3c == 0) {
20     local_28.tv_nsec = local_28.tv_sec * 1000000000 + local_28.tv_nsec;
21     if (DAT_00e510a0 == 0) {
22       DAT_00e510a0 = local_28.tv_nsec;
23     }
24     *param_1 = local_28.tv_nsec - DAT_00e510a0;
25     local_2c = 0;
26   }
27   else {
28     piVar2 = (int *)__errno();
29     if (*piVar2 == 0) {
30       local_2c = -1;
31     }
32     else {
33       piVar2 = (int *)__errno();
34       if (*piVar2 == 0) {
35         local_2c = 0;
36       }
37       else {
38         piVar2 = (int *)__errno();
39         local_2c = *piVar2 + 120000;
40       }
41     }
42   }
43   lVar1 = tpidr_el0;
44   if (*(long *)(lVar1 + 0x28) != local_18) {
45                     /* WARNING: Subroutine does not return */
46     __stack_chk_fail(local_2c);
47   }
48   return;
49 }
50
```

a85f793 · pjproject / pjlib / src / pj / os_timestamp_posix.c · Top

Code · Blame · 336 lines (273 loc) · 7.56 KB · Raw

```
227
228  PJ_DEF(pj_status_t) pj_get_timestamp(pj_timestamp *ts)
229  {
230      struct timespec tp;
231      int err = -1;
232
233  #if defined(ANDROID_ALARM_GET_TIME)
234      if (s_alarm_fd == -1) {
235          int fd = open("/dev/alarm", O_RDONLY);
236          if (fd >= 0) {
237              s_alarm_fd = fd;
238              pj_atexit(&close_alarm_fd);
239          }
240      }
241
242      if (s_alarm_fd != -1) {
243          err = ioctl(s_alarm_fd,
244                ANDROID_ALARM_GET_TIME(ANDROID_ALARM_ELAPSED_REALTIME), &tp);
245      }
246  #elif defined(CLOCK_BOOTTIME)
247      err = clock_gettime(CLOCK_BOOTTIME, &tp);
248  #endif
249
250      if (err != 0) {
251          /* Fallback to CLOCK_MONOTONIC if /dev/alarm is not found, or
252           * getting ANDROID_ALARM_ELAPSED_REALTIME fails, or
253           * CLOCK_BOOTTIME fails.
254           */
255          err = clock_gettime(CLOCK_MONOTONIC, &tp);
256      }
257
258      if (err != 0) {
259          return PJ_RETURN_OS_ERROR(pj_get_native_os_error());
260      }
261
262      ts->u64 = tp.tv_sec;
263      ts->u64 *= NSEC_PER_SEC;
264      ts->u64 += tp.tv_nsec;
265
266      return PJ_SUCCESS;
267  }
```

Bug & Fix

- **Bug**: OOB Write to Unmapped Memory (`str x8, [0x0 + 0x3b60]`)
  - Unfortunately, we cannot control the address - the base pointer is null and the offset is `0x3b60`
  - Hypothesis is that the malformed capabilities data causes the processing of an uninitialized object - getting a timestamp from said null object triggers the bug

Bug & Fix

- **Bug**: OOB Write to Unmapped Memory (`str x8, [0x0 + 0x3b60]`)
  - Unfortunately, we cannot control the address - the base pointer is null and the offset is `0x3b60`
  - Hypothesis is that the malformed capabilities data causes the processing of an uninitialized object - getting a timestamp from said null object triggers the bug

Bug & Fix

- **Bug**: OOB Write to Unmapped Memory (`str x8, [0x0 + 0x3b60]`)
  - Unfortunately, we cannot control the address - the base pointer is null and the offset is `0x3b60`
  - Hypothesis is that the malformed capabilities data causes the processing of an uninitialized object - getting a timestamp from said null object triggers the bug

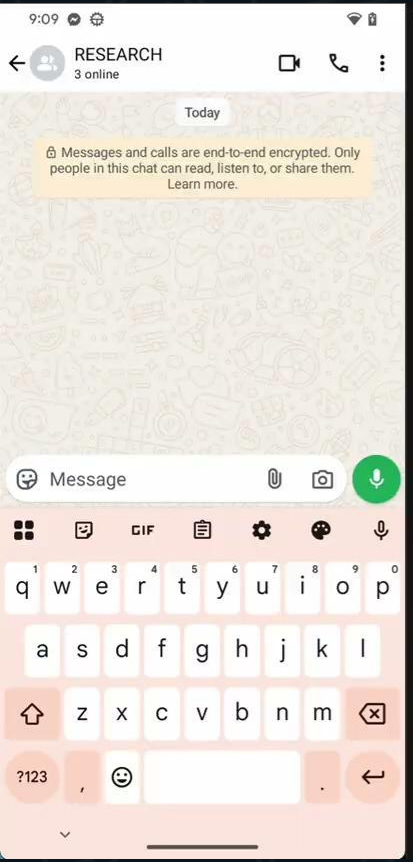
- **Impact:** Crashes all attendees of a call
- **Fix:** Underlying object was being set to null in the case of malformed capabilities data; stop processing object if null

Voice Chat Bug

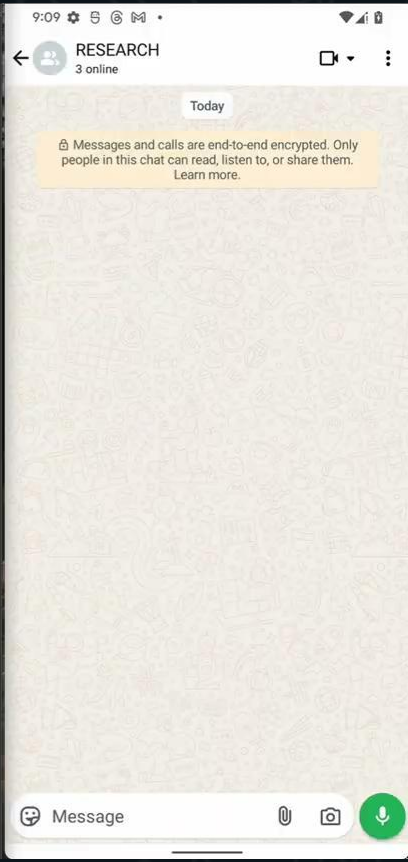
Forcing video streams into voice chats

Voice Chats

- The Voice Chat feature was introduced around November 2023 (v2.23.7.12)
- “Voice chats allow you to instantly talk live with members of a group chat”
  - “Voice chats are only available for groups of 33 to 256 people.”
- Starts quietly without ringing group members - described as similar to discord
  - Simply a push notification rather than a proper ring
- `<offer>` stanza includes the `<silence reason=”vc_init”>` element
  - What if we force `<silence reason=”vc_init”>` into Voice or Video calls?

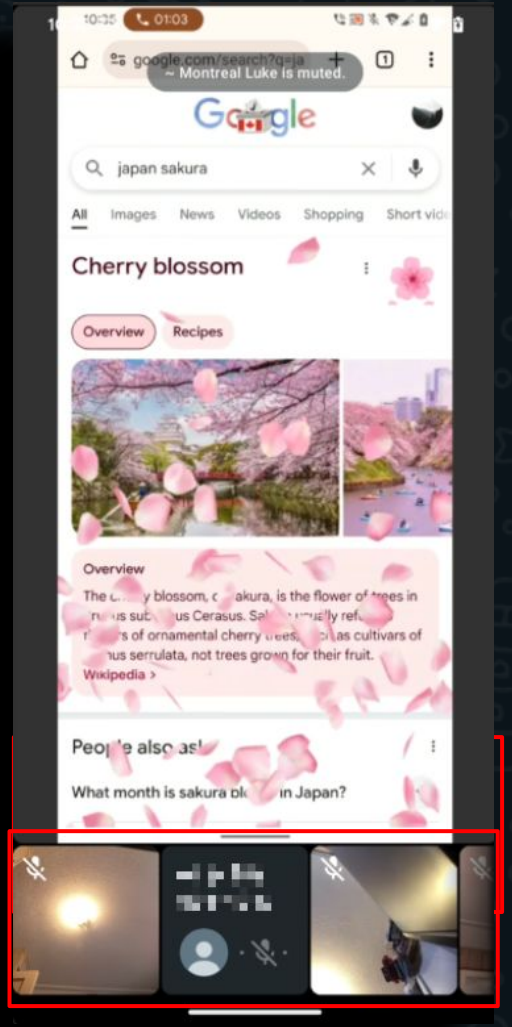



```
(env) dev@Macmini whatsapp %
```

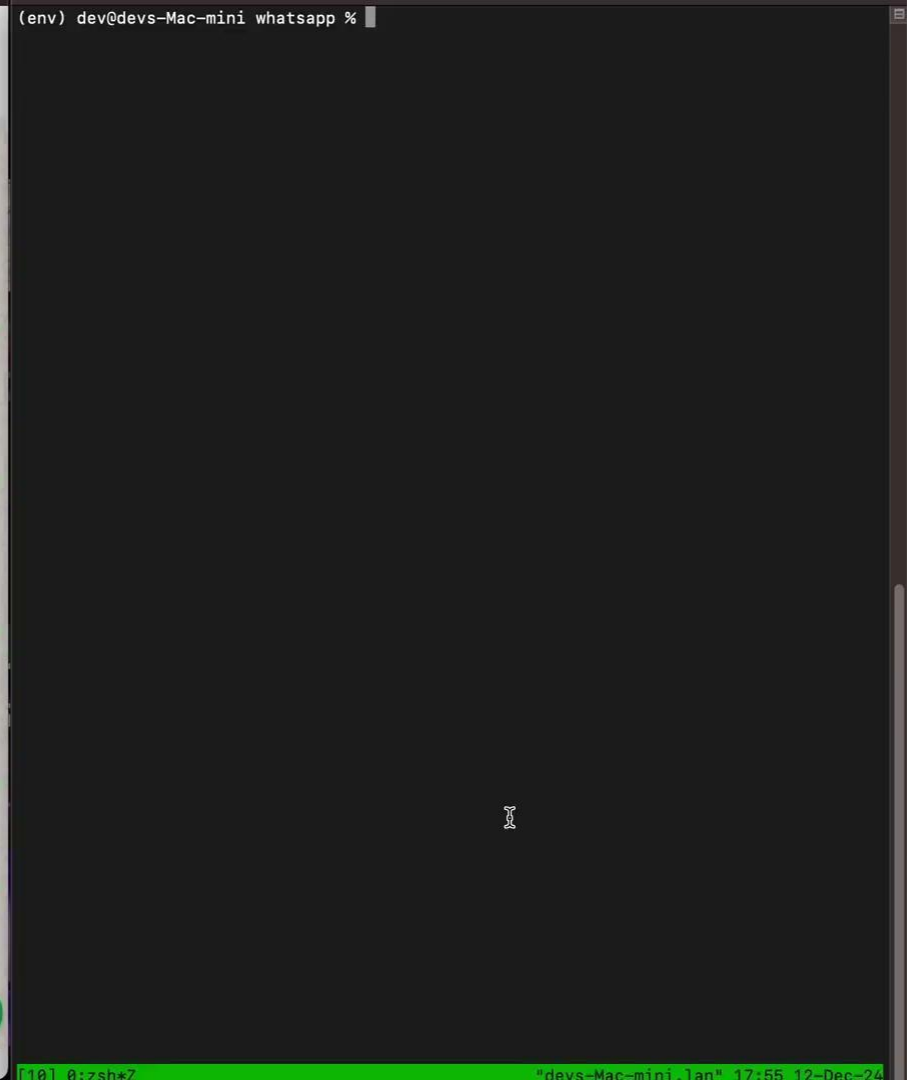


Spoofed Voice Chats

- Including `<silence reason=”vc_init”>` causes:
  - Enables voice chat UI on Android (not iOS)
  - Recipient perceives Voice and Video calls as Voice Chats
  - Video stream appears in strange places
- Video stream is not enabled on recipient device
  - No impact on recipients privacy…
  - But could an attacker turn it on?
- Note: Shares design elements with Screen Sharing…
  - Does screen sharing while in a Voice Chat affect anything?




5:55
Video PoC
Lars, Luke, +1 (438) 596-4243,...
0:01
5:56 PM
Video call
12 min · 4 joined
5:58 PM
Sunday
Video call
10 hr, 53 min · 4 joined
11:07 AM
Monday
Video call
3 hr, 39 min · 4 joined
7:50 AM
Today
Video call
3 min · 4 joined
5:50 PM
Video call
8 sec · You joined
5:54 PM
Message


(env) dev@devs-Mac-mini whatsapp %
[10] 0:zsh*Z
"devs-Mac-mini.lan" 17:55 12-Dec-24


5:55
Thu, Dec 12
-4°C
Messenger
WhatsApp
Play Store
Gmail
Photos
YouTube

Bug & Fix

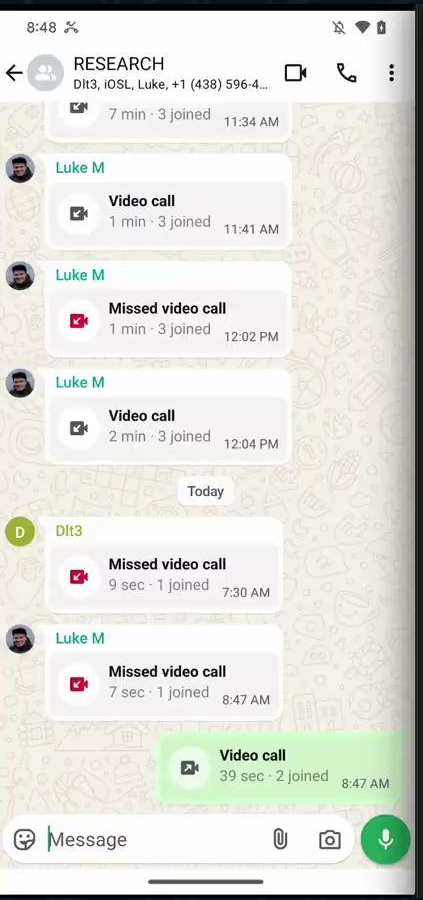
- **Bug**: Under certain conditions it is possible to enable a recipients video stream in Voice Chats without their permission
- Reported December 7th → Patched December 13th
  - Worth noting that with these bugs you have to be ready to answer impact/reliability questions
    - Questions regarding reliability/impact on the 12th; unable to answer if I wait till the 13th
- **Fix**: Video calls will not be routed if they include the “VC_*” silence stanzas
  - Appears that copied design element for screen sharing is also no longer used

Transport Messages

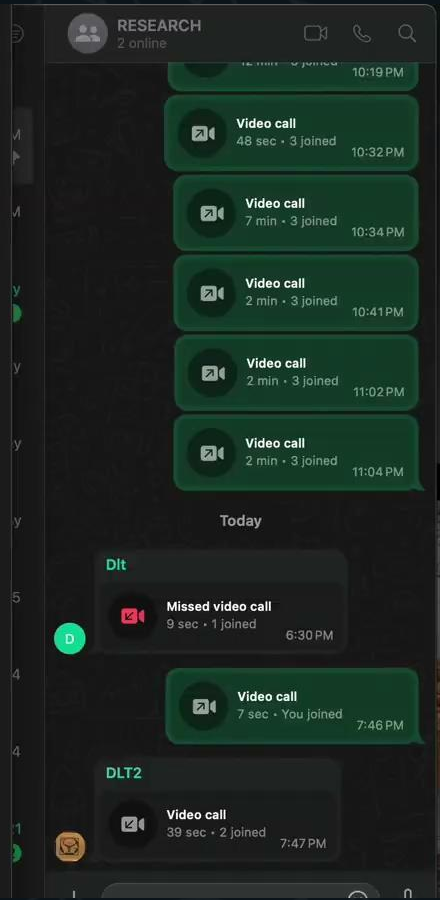
Invalid Transport Stanzas to OOB Read

Transport Messages

- Transport stanzas represent data that is associated with a call connection
- Includes information about relay servers as well as the type of connection
- A transport message can be its own standalone xmpp stanza or a subelement
  - I.e. contained within `<transport>`, `<video>`, or `<relaylantecy>`
- Sent when establishing, reconnecting, or starting video streaming.

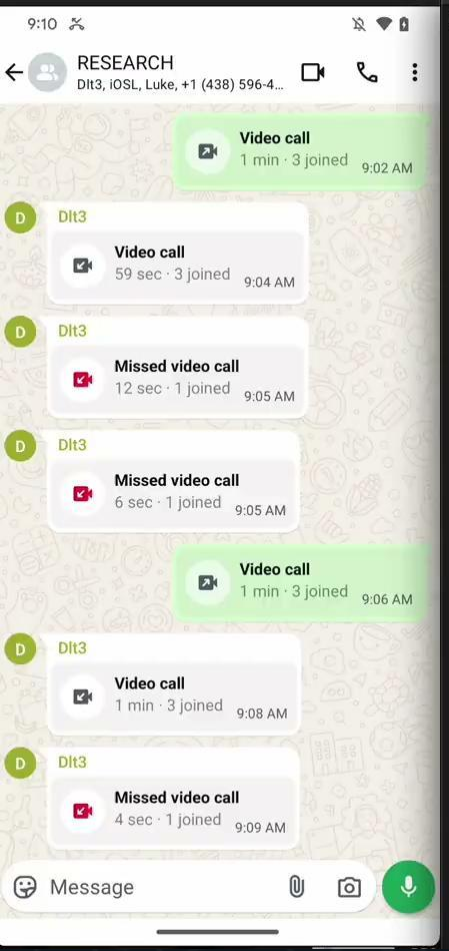



```
(env) dev@balthazar res3 %
```

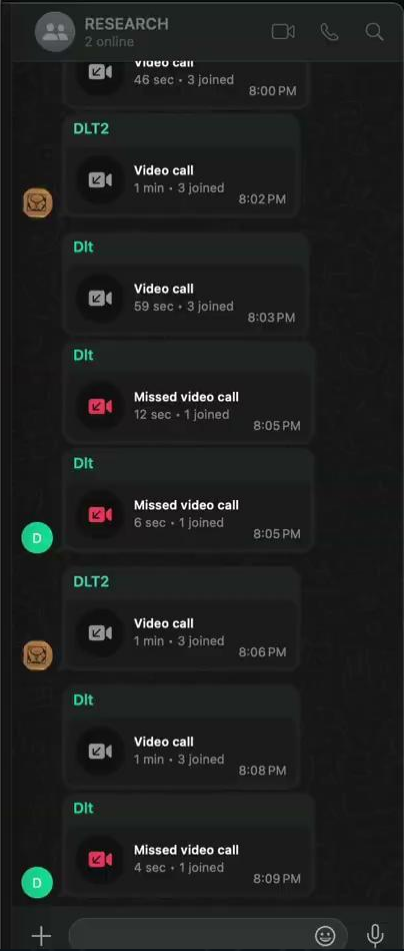


What about other types?

Let's modify and try a few out….




```
(env) dev@balthazar res3 % python3 poc2.py
```



Remote Crashdump

```
Process:               WhatsApp [53578]
Identifier:            net.whatsapp.WhatsApp
Version:               25.9.72 (714854393) (Released: March 31 2025)
Code Type:             ARM-64 (Native)

Crashed Thread:        5  media

Exception Type:        EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:       KERN_INVALID_ADDRESS at 0x000086130c7e1347 -> 0x000006130c7e1347 (possible pointer authentication failure)
Exception Codes:       0x0000000000000001, 0x000086130c7e1347

Thread 5 Crashed:: media
0   WhatsApp                        0x103c04fe8 0x100e00000 + 48254952
1   WhatsApp                        0x1037ba918 0x100e00000 + 43755800
2   WhatsApp                        0x1037ba730 0x100e00000 + 43755312
3   WhatsApp                        0x1037bbdd0 0x100e00000 + 43761104
4   WhatsApp                        0x1037bbd84 0x100e00000 + 43761028
5   WhatsApp                        0x1037bbce0 0x100e00000 + 43760864
6   WhatsApp                        0x1037bad3c 0x100e00000 + 43756860
7   WhatsApp                        0x103a37524 0x100e00000 + 46363940
8   WhatsApp                        0x103a3dbd0 0x100e00000 + 46390224
9   WhatsApp                        0x103a466fc 0x100e00000 + 46425852
10  WhatsApp                        0x103a46664 0x100e00000 + 46425700
11  WhatsApp                        0x103a466a4 0x100e00000 + 46425764
12  WhatsApp                        0x103a423d4 0x100e00000 + 46408660
....
```

Remote Crash - Debugging

```
Target 0: (WhatsApp) stopped.
(lldb) c
Process 53578 resuming
(lldb) c
error: Process is running.  Use 'process interrupt' to pause execution.
Process 53578 stopped
* thread #6, name = 'media', stop reason = EXC_BAD_ACCESS (code=1, address=0x86130c7e1347)
    frame #0: 0x0000000103c04fe8 WhatsApp`_mh_execute_header + 48254952
WhatsApp`_mh_execute_header:
->  0x103c04fe8 <+48254952>: ldrb   w9, [x0, #0x17]
    0x103c04fec <+48254956>: sxtb   w10, w9
    0x103c04ff0 <+48254960>: cmp    w10, #0x0
    0x103c04ff4 <+48254964>: ldp    x11, x10, [x0]
Target 0: (WhatsApp) stopped.
```

Remote Crash - Debugging

```
PC   0000000103C04FE8 | .O...... | => `_mh_execute_header + 0x2E04FE8`
SP   000000016F796970 | piyo.... | => "
X0   000086130C7E1330 | 0.~..... |
X1   000000018F7969A8 | .iyo.... | => 0x6000039BABE0 => "WebRTC-MLowDecoder-lowPassCutoffFreq>
X2   000000000000001B | ........ |
X3   FFFFFFFFFFFFFFF0 | ........ |
X4   0000600003AC43AB | .C...`.. | => "utoffFrequencyHz"
X5   00006000039BABFB | .....`.. | => "utoffFrequencyHz"
X6   000000000000007A | z....... |
X7   0000000000000403 | ........ |
X8   00006000039BABE0 | .....`.. | => "WebRTC-MLowDecoder-lowPassCutoffFrequencyHz"
X9   00006000039BABE0 | .....`.. | => "WebRTC-MLowDecoder-lowPassCutoffFrequencyHz"
X10  000000000000002B | +....... |
X11  0000600003AC4390 | .C...`.. |
X12  00006000036EEEE4 | ..n..`.. | => "@Q"
X13  00000000001FF800 | ........ |
X14  00000000000007FB | ........ |
X15  0000000092E058EB | .X...... |
X16  000000018ECDCD50 | P....... | => `_platform_memcmp + 0x0`
X17  0000000200E99EC0 | ........ | => 0x18ECDF1A0 => `_platform_memmove + 0x0`
X18  0000000000000000 | ........ |
X19  000000000000002B | +....... |
X20  000086130C7E1310 | ..~..... |
X21  000000016F796A50 | Pjyo.... | => 0x6000039BABE0 => "WebRTC-MLowDecoder-lowPassCutoffFreq>
X22  0000000000000008 | ........ |
X23  0000000000003E80 | .>...... |
X24  000060000338FA80 | ..8..`.. |
X25  000000016F796D60 | `myo.... |
X26  00000000000000DD | ........ |
X27  00003AA7BDB00264 | d....:.. |
X28  000000016F796F60 | `oyo.... |
```

Normal Execution - Debugging

```
[disassembly]
WhatsApp`_mh_execute_header:
-> 0x107810fe8 <+48254952>: ldrb   w9, [x0, #0x17]
   0x107810fec <+48254956>: sxtb   w10, w9
   0x107810ff0 <+48254960>: cmp    w10, #0x0
   0x107810ff4 <+48254964>: ldp    x11, x10, [x0]
   0x107810ff8 <+48254968>: ret
   0x107810ffc <+48254972>: mov    x2, x26
   0x107811000 <+48254976>: mov    x3, x25
   0x107811004 <+48254980>: ret
   0x107811008 <+48254984>: str    x27, [sp, #0x58]
   0x10781100c <+48254988>: str    d8, [sp, #0x60]
   0x107811010 <+48254992>: ret
   0x107811014 <+48254996>: ldrsw  x24, [x21]
   0x107811018 <+48255000>: ldr    x0, [x19, x24]
   0x10781101c <+48255004>: ret
   0x107811020 <+48255008>: ldrsw  x23, [x21]
   0x107811024 <+48255012>: ldr    x0, [x19, x23]
   0x107811028 <+48255016>: ret
   0x10781102c <+48255020>: mov    x8, #0x4040000000000000 ; =4629700416936869888
   0x107811030 <+48255024>: fmov   d8, x8
   0x107811034 <+48255028>: ret
   0x107811038 <+48255032>: ldrsw  x8, [x24, #0x8]
   0x10781103c <+48255036>: ldr    x2, [x19, x8]
   0x107811040 <+48255040>: ret
   0x107811044 <+48255044>: ldrsw  x8, [x24, #0x8]
   0x107811048 <+48255048>: ldr    x0, [x20, x8]
   0x10781104c <+48255052>: ret
   0x107811050 <+48255056>: ldr    x0, [x20, #0x7b0]
   0x107811054 <+48255060>: mov    x1, x19
   0x107811058 <+48255064>: ret
   0x10781105c <+48255068>: add    x21, x19, x8
   0x107811060 <+48255072>: ldp    d0, d1, [x21]
   0x107811064 <+48255076>: ret
```

```
[regs:general]
PC  0000000107810FE8 | ........ | => `_mh_execute_header + 0x2E04FE8`
SP  000000016BAFE9A0 | ...k.... | => 0x16BAFEAC0 => 0x108689480 => 0x107583DEC => `_mh_execu>
X0  000000016BAFEA50 | P..k.... | => 0x60000316B150 => "WebRTC-MLowDecoder-lowPassCutoffFreq>
X1  000000016BAFE9D8 | ...k.... | => 0x60000315D560 => "WebRTC-MLowDecoder-lowPassCutoffFreq>
X2  FFFFFFFFFFFFFFFB | ........ |
X3  00000000000000FF | ........ |
X4  000060000315D57B | {....`.. | => "utoffFrequencyHz"
X5  000060000316B16B | k....`.. | => "utoffFrequencyHz"
X6  000000000000007A | z....... |
X7  0000000000000403 | ........ |
X8  000060000315D560 | `....`.. | => "WebRTC-MLowDecoder-lowPassCutoffFrequencyHz"
X9  000060000315D560 | `....`.. | => "WebRTC-MLowDecoder-lowPassCutoffFrequencyHz"
X10 000000000000002B | +....... |
X11 000060000315D560 | `....`.. | => "WebRTC-MLowDecoder-lowPassCutoffFrequencyHz"
X12 0000600003EBDAC4 | .....`.. |
X13 00000000001FF800 | ........ |
X14 00000000000007FB | ........ |
X15 00000000FCC0B108 | ........ |
X16 000000018ECDCD50 | P....... | => `_platform_memcmp + 0x0`
X17 0000000200E99EC0 | ........ | => 0x18ECDF1A0 => `_platform_memmove + 0x0`
X18 0000000000000000 | ........ |
X19 000000000000002B | +....... |
X20 000000016BAFEA50 | P..k.... | => 0x60000316B150 => "WebRTC-MLowDecoder-lowPassCutoffFreq>
X21 0000600001C5BA20 |  ....`.. |
X22 000000016BAFEAC0 | ...k.... | => 0x108689480 => 0x107583DEC => `_mh_execute_header + 0x2>
X23 0000000000003E80 | .>...... |
X24 0000600003BC2840 | @(...`.. |
X25 000000016BAFED60 | `..k.... |
X26 0000000000000410 | ........ |
X27 0000339F8B550264 | d.U..3.. |
X28 000000016BAFEF60 | `..k.... |
X29 n/a
X30 n/a
```

Ghidra Decompilation

Decompilation - V ce

Decompile: mem_copy_config_name - (AARCH64-64-cpu0x0)

```
1
2 void mem_copy_config_name(void *corrupt_ptr)
3
4 {
```

Decompile: FUN_1029ba764 - (AARCH64-64-cpu0x0)

```
1
2 void FUN_1029ba764(long *corrupt_ptr)
3
4 {
5   long *plVar1;
6   undefined8 *unaff_x19;
7   long unaff_x20;
8   long *plVar2;
9   long *plVar3;
10
11  FUN_102de0df8();
12  plVar1 = *(long **)(unaff_x20 + 8);                                    tr);
13  plVar2 = (long *)(unaff_x20 + 8);
14  while (plVar3 = plVar2, plVar1 != (long *)0x0) {
15                    /* Moves x21 into x0 - x0 observed as corrupted at this point. */
16    while (plVar3 = plVar1, mov_x21_x0_crpt_func(), (int)corrupt_ptr == 0) {
17      corrupt_ptr = plVar3 + 4;
18      func_with_corrupt_ptr_(corrupt_ptr);
19      if ((int)corrupt_ptr == 0) goto LAB_1029ba7c4;
20      plVar2 = plVar3 + 1;
21      plVar1 = (long *)*plVar2;
22      if ((long *)*plVar2 == (long *)0x0) goto LAB_1029ba7c4;
23    }                                                                    sVar2);
24    plVar2 = plVar3;
25    plVar1 = (long *)*plVar3;
26  }
27 LAB_1029ba7c4:
28  *unaff_x19 = plVar3;
29  FUN_102dc9978(plVar2);
30  return;
31 }
32
```

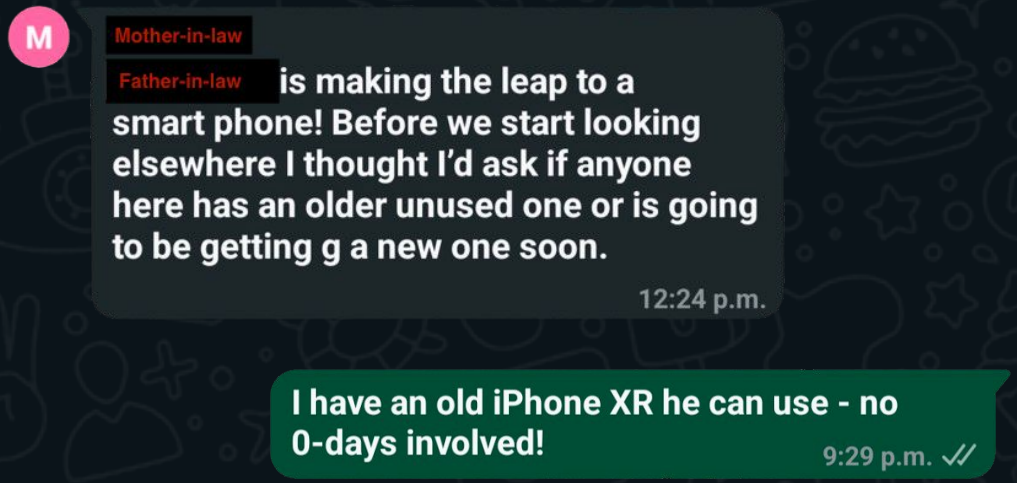
Decompile: func_with_corrupt_ptr__ - (AARCH64-64-cpu0x0)

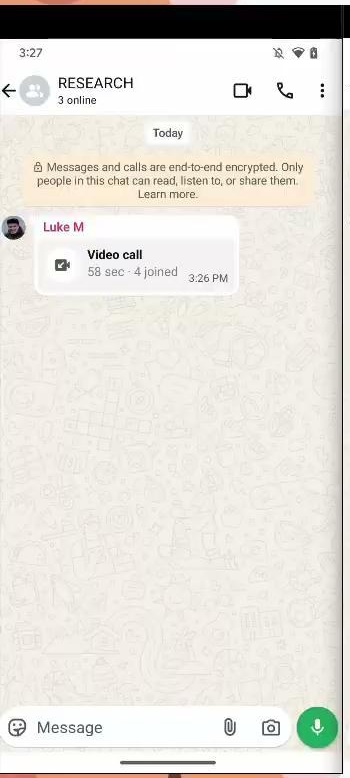
```
1
2 void func_with_corrupt_ptr__(void *corrupt_ptr)
3
4 {
5   undefined8 uStack_28;
6
7   FUN_102de8778(*(undefined8 *)PTR____stack_chk_guard_103a4fb80);
8   FUN_1029ba204();
9   mov_x19_x0();
10  mem_copy_config_name(corrupt_ptr);
11  if (*(long *)PTR____stack_chk_guard_103a4fb80 == uStack_28) {
12    FUN_102dcb120();
13    return;
14  }
15                    /* WARNING: Subroutine does not return */
16  __stack_chk_fail();
17 }
18
```

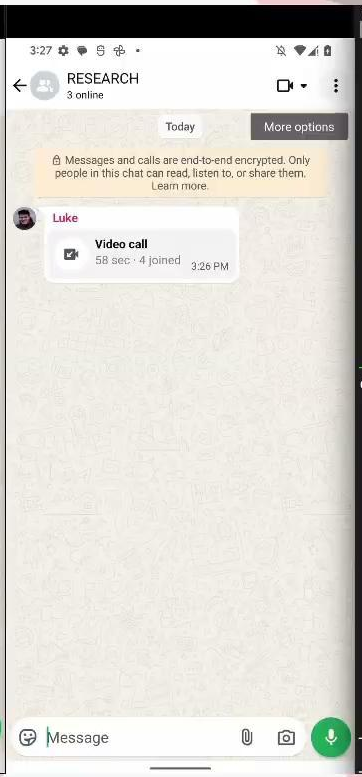
Bug & Fix

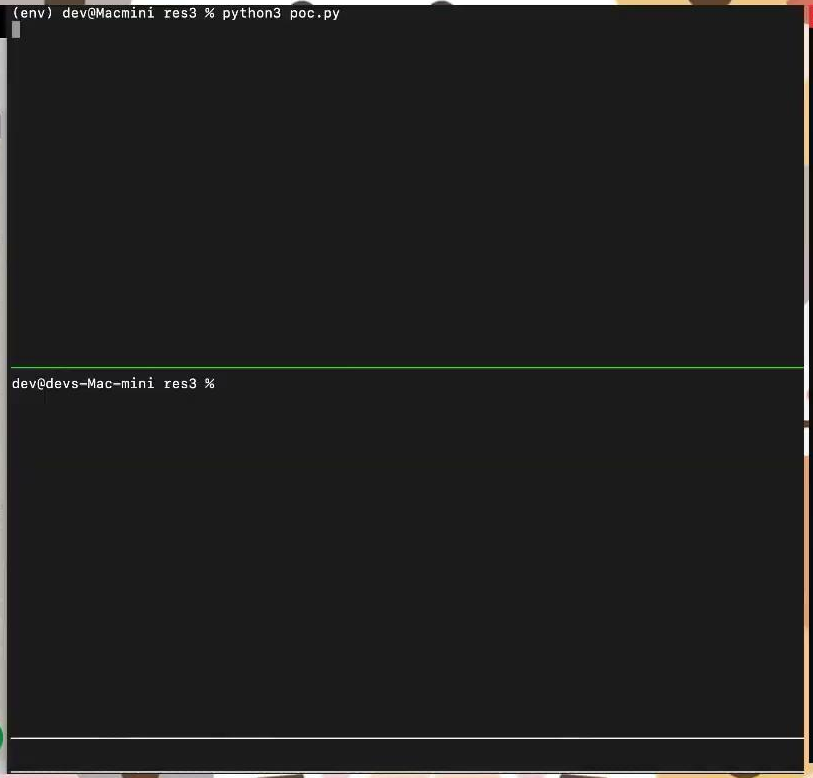
- **Bug:** Hypothesis - the while loop which processes the configuration objects performs a UAF when another thread tears down the call.
- **Impact:** Potential for RCE on MacOS (couldn’t trigger on iOS)
- **Fix:** 🤷‍♂️ This bug was patched through an inadvertent patch in the next WhatsApp release without insights (patched in versions > 2.25.9.72)
- **Open questions:**
  - Is it possible to control the pointer? What patched the vulnerability? Which thread was responsible for freeing the object? Non-determinism caused by A/B bucketing?

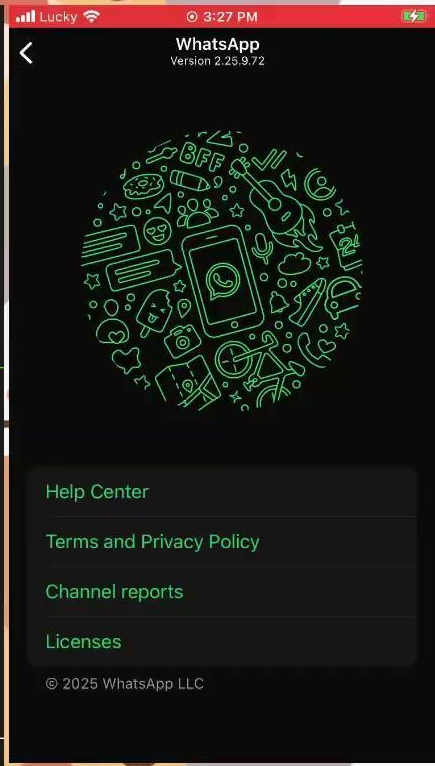
But wait….


M
Mother-in-law
Father-in-law is making the leap to a smart phone! Before we start looking elsewhere I thought I'd ask if anyone here has an older unused one or is going to be getting g a new one soon.
12:24 p.m.
I have an old iPhone XR he can use - no 0-days involved!
9:29 p.m.


3:27
RESEARCH
3 online
Today
Messages and calls are end-to-end encrypted. Only people in this chat can read, listen to, or share them. Learn more.
Luke M
Video call
58 sec · 4 joined
3:26 PM
Message


3:27
RESEARCH
3 online
Today
More options
Messages and calls are end-to-end encrypted. Only people in this chat can read, listen to, or share them. Learn more.
Luke
Video call
58 sec · 4 joined
3:26 PM
Message


(env) dev@Macmini res3 % python3 poc.py
dev@devs-Mac-mini res3 %


Lucky
3:27 PM
WhatsApp
Version 2.25.9.72
Help Center
Terms and Privacy Policy
Channel reports
Licenses
© 2025 WhatsApp LLC

```
Hardware Model:      iPhone8,1
Process:             WhatsApp [41671]
Path:                /private/var/containers/Bundle/Application/A1B108E1-DC7B-43F1-AC6E-9B14C30AA9AF/WhatsApp.app/WhatsApp
Identifier:          net.whatsapp.WhatsApp
Version:             25.9.72 (714854349)
AppStoreTools:       16E137
Code Type:           ARM-64 (Native)
Parent Process:      launchd [1]
Coalition:           net.whatsapp.WhatsApp [397]

OS Version:          iPhone OS 15.5 (19F77)
Release Type:        User

Exception Type:  EXC_BAD_ACCESS (SIGSEGV)
Exception Subtype: KERN_INVALID_ADDRESS at 0x000280da8c400037

Thread 4 name:  media
Thread 4 Crashed:
0   WhatsApp                        0x102cbedb8 0x100670000 + 40168888
1   WhatsApp                        0x102cbed80 0x100670000 + 40168832
2   WhatsApp                        0x102cbeb98 0x100670000 + 40168344
3   WhatsApp                        0x102cc02f0 0x100670000 + 40174320
4   WhatsApp                        0x102cc02a4 0x100670000 + 40174244
5   WhatsApp                        0x102cc0200 0x100670000 + 40174080
6   WhatsApp                        0x102e63ac8 0x100670000 + 41892552
7   WhatsApp                        0x102cbf12c 0x100670000 + 40169772
```

Thank you!

Questions?