Google

Zooming in on Zero-click Exploits

Google

Zooming in on Zero-click Exploits

or

How to Reverse Engineer a Giant Pile of Code

About Me

- Natalie Silvanovich AKA natashenka
- Project Zero member
- Previously did mobile security on Android and BlackBerry

What is Zoom

- Video conferencing meeting solution
- Can join meetings via clients or browser (or phone)
- Clients exist for Linux, Window, iOS, Android and more

Why Zoom?

$200,000 Awarded for Zero-Click Zoom Exploit at Pwn2Own

By Eduard Kovacs on April 08, 2021

Share Tweet Recommend 23 RSS

Two researchers earned $200,000 on the second day of the Pwn2Own 2021 hacking competition for a Zoom exploit allowing remote code execution without user interaction.

The exploit, demonstrated by Daan Keuper and Thijs Alkemade from Computest, involves three vulnerabilities and it works on the latest versions of Windows 10 and Zoom. In the demo at Pwn2Own, the victim saw a meeting invitation from the attacker, but the victim didn't actually have to click anything to trigger the code execution.

Also on the second day of Pwn2Own 2021, Bruno Keith and Niklas Baumstark of Dataflow Security earned $100,000 for an exploit that works both on the Chrome and Microsoft Edge web browsers.

If attempts to hack the Parallels virtualization product failed on the first day, on the second day, Jack Dates from RET2 Systems and Sunjoo Park (aka grigoritchy) earned $40,000

Google

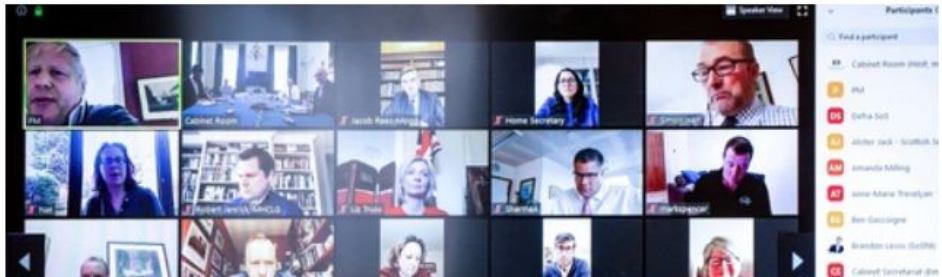
Why Zoom?

Hackers Are Selling a Critical Zoom Zero-Day Exploit for $500,000

People who trade in zero-day exploits say there are two Zoom zero-days, one for Windows and one for MacOS, on the market.

By Lorenzo Franceschi-Bicchierai

April 15, 2020, 6:54am Share Tweet Snap



VICE

Many Trucker Convoy Supporters Worried They're Being Lured Into...

TESS OWEN

Zoom Attack Surface

- Installed Zoom and manipulated UI
- Read website
- Lots of features I never expected
  - Messaging between Zoom Contacts
  - Meetings with many ways to start and join them
  - Webinars
  - On-prem (self-hosted) servers

Zoom Attack Surface

- Looked at past bugs
- Pwn2Own bug (CVE-2021-34407)
  - Memory corruption in crypto key exchange
  - Required target to be a Zoom Contact or join a Meeting
  - Reported by Daan Keuper and Thijs Alkemade

Zoom Attack Surface

- Code Snippet and Giphy Vulnerabilities (CVE-2020-6109/CVE-2020-6110)
  - Directory traversal in special message formats
  - Required target to be a Zoom Contact
  - Reported by Talos
- A few others, mostly directory traversal

Attack Scenarios

- Attacker messages or calls target Zoom Contact
- Attacker convinces target to join their Zoom Meeting, or joins a Zoom Meeting they have information for
- Server?????
- Zoom SDK (analysis only)

Loading Zoom in IDA

- Eek
- Very few symbols or strings
- One big binary in Linux
- Other platforms split out libraries
  - Used this to identify some functions
  - Different compilers have different symbols (ie vtables)
  - Now I know Zoom SDK has more logs

XMPP

- Zoom documents using XMPP for peer-to-peer and peer-to-server communications
- Looked for XMPP libraries (in hopes of hooking)

gloox

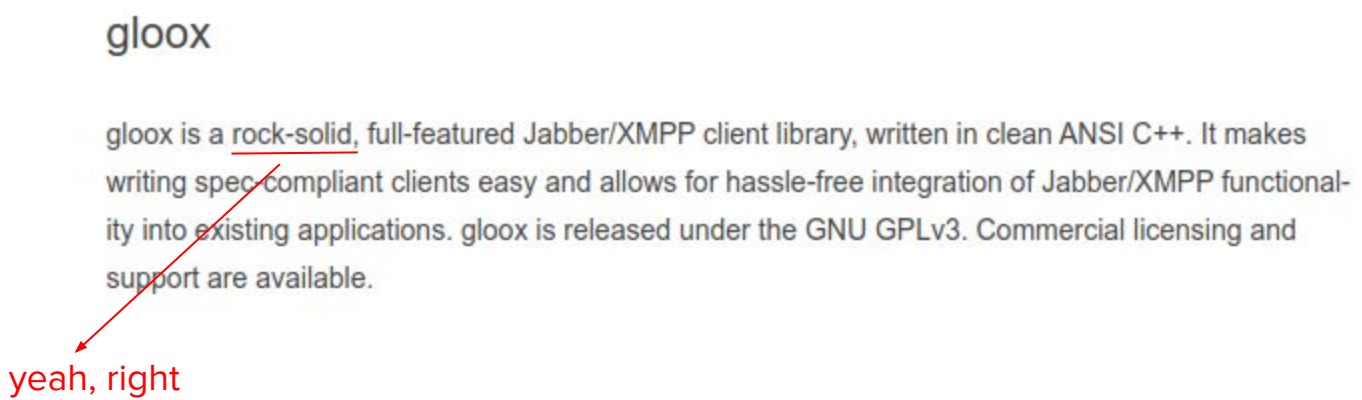
- Third-party XMPP library I had never heard of

gloox

gloox is a rock-solid, full-featured Jabber/XMPP client library, written in clean ANSI C++. It makes writing spec-compliant clients easy and allows for hassle-free integration of Jabber/XMPP functionality into existing applications. gloox is released under the GNU GPLv3. Commercial licensing and support are available.

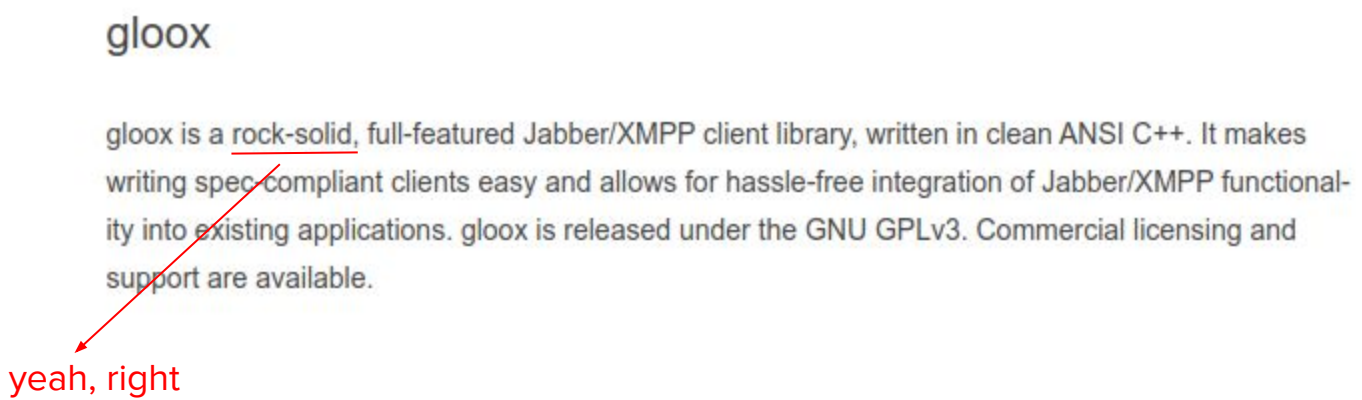
gloox

- Third-party XMPP library I had never heard of



gloox

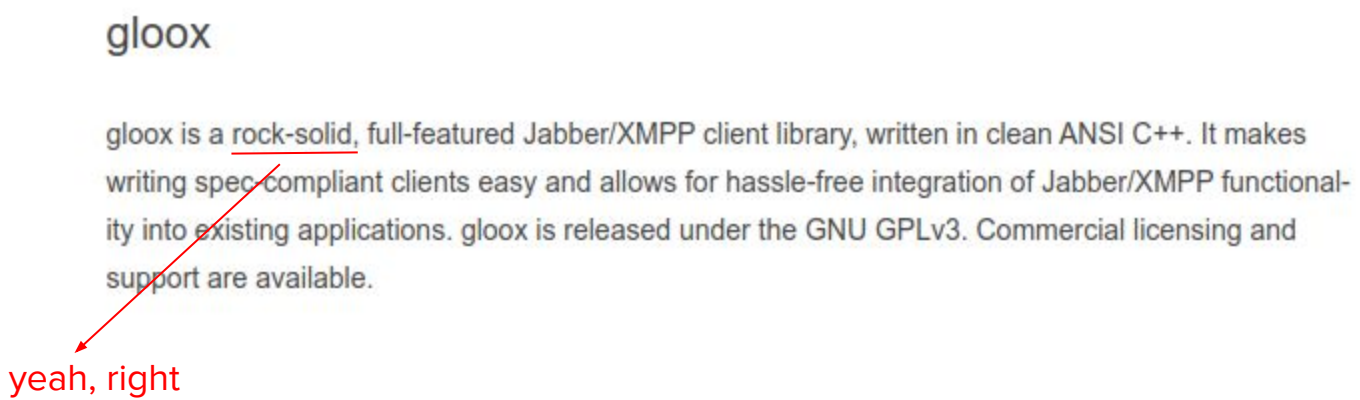
- Third-party XMPP library I had never heard of



- Fuzzed and reviewed code with no results

gloox

- Third-party XMPP library I had never heard of



- Fuzzed and reviewed code with no results
- Wait a sec, what?

Other Messages

- gloox hooking was a pain
- Wireshark
- Located openssl in IDA by looking for specific log entries
  - Multiple copies
  - Really just tried all of them
  - Cipher names are a back-up
- Hooked SSL_write with Frida

```
Interceptor.attach(SSL_write,
    {
        onEnter: function (args){
            //do stuff here
```

Other Messages

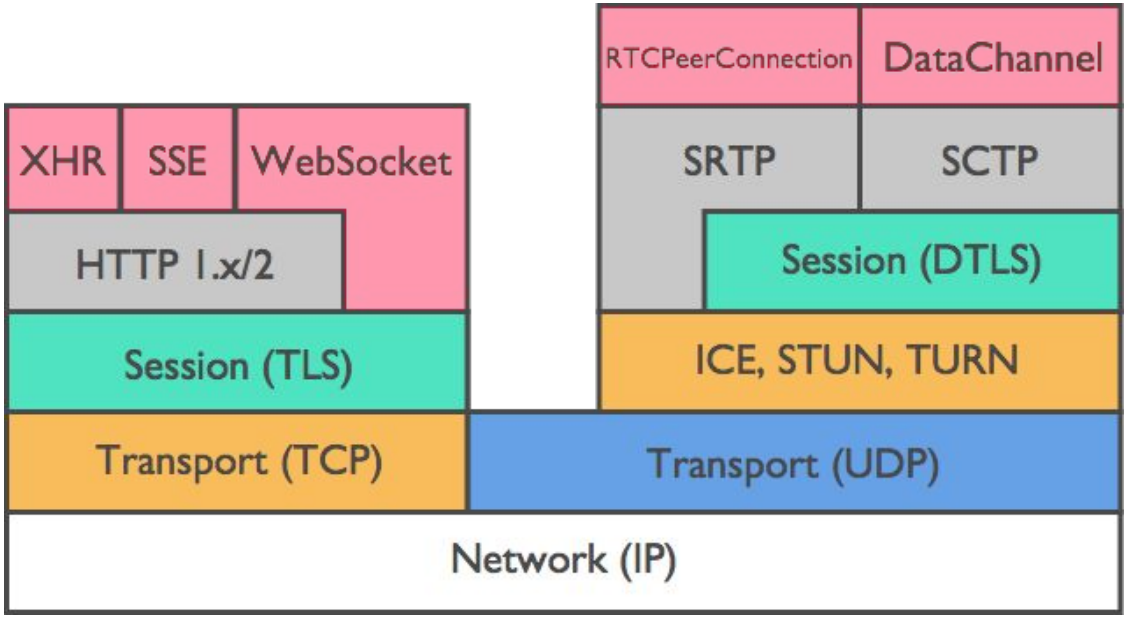
- Output outgoing packets
- Could see XMPP
- Also a lot of other stuff (more on this later)

Symbols!

- Tried to find bugs in XMPP processing but failed
- Located a version of the Zoom SDK with symbols
  - Five years old, but better than nothing
  - No core binary
- Still didn’t find anything
- Pwn2Own bug turned out to be in this area when write-up was released

Realtime Transport Protocol (RTP)

- Leading source of vulnerabilities in video-conferencing software
- Not zero-click, but I was really curious



RTP Fuzzing

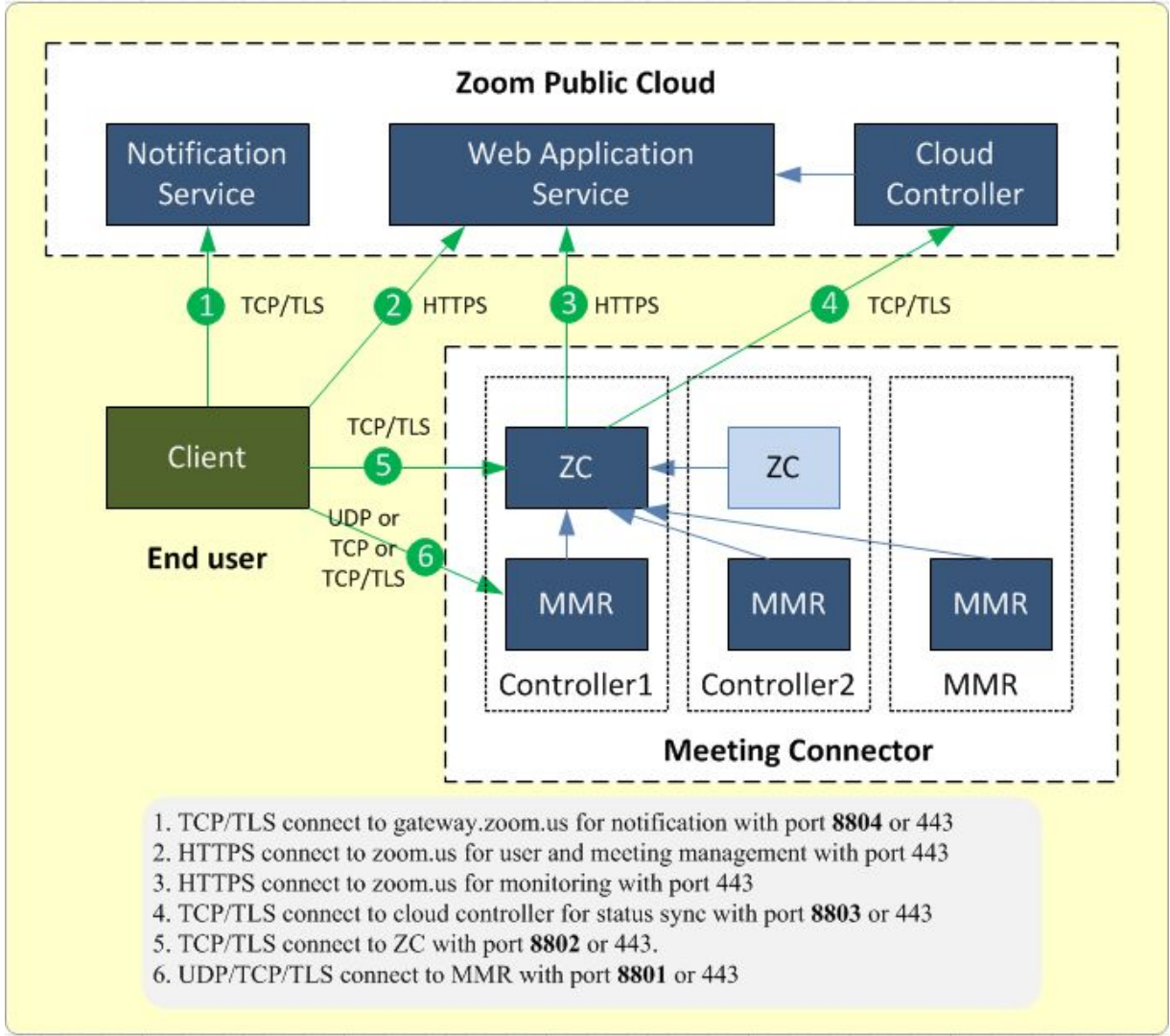
- RTP entry-point was challenging to locate
- Searched all platform binaries for log entries related to RTP features
  - “Extensions” was the winner
- Loaded Zoom as a shared library (on Linux) and called into RTP entry point
- Couldn’t figure out Zealot
  - Unfortunately, functions sprung from ‘context’ object
  - Future work?

RTP Fuzzing

- Used DrSancov for instrumentation
- https://github.com/googleprojectzero/DrSancov
- Found a bug with fuzzing, but couldn’t send RTP from one peer to another

Server Time!

- Set up On-prem Zoom server with help from Zoom security team



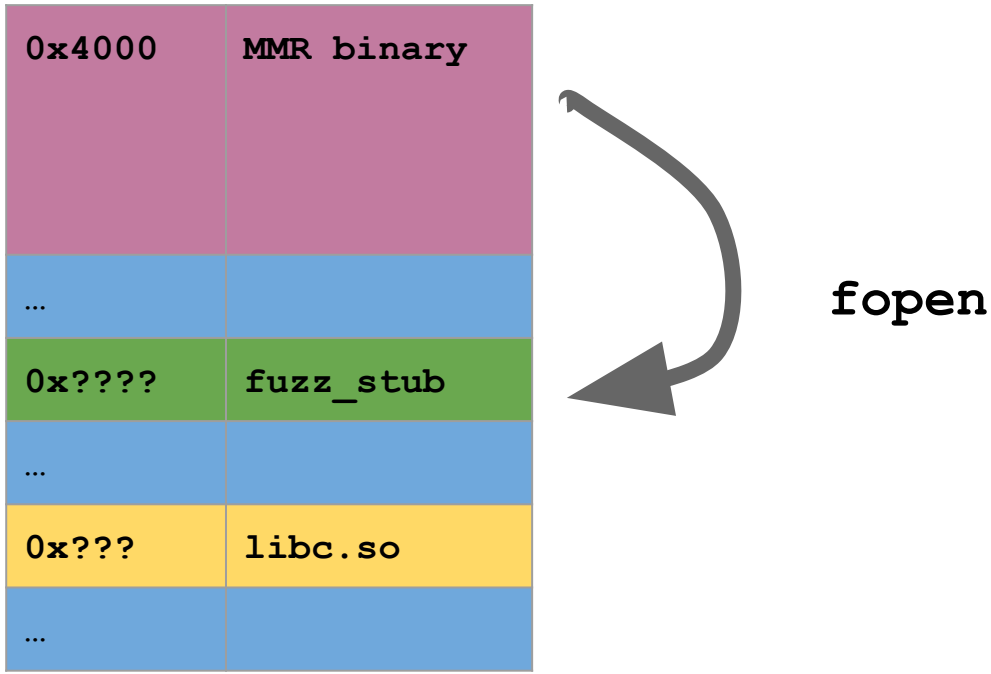
Server Time!

- Multimedia Router (MMR) server transmits audio and video content among peers
- MMR server (unsurprisingly) decodes and re-encodes RTP
- Client RTP bug still didn’t work due to unnoticed checks in the pipeline
  - I was wrong about entry point

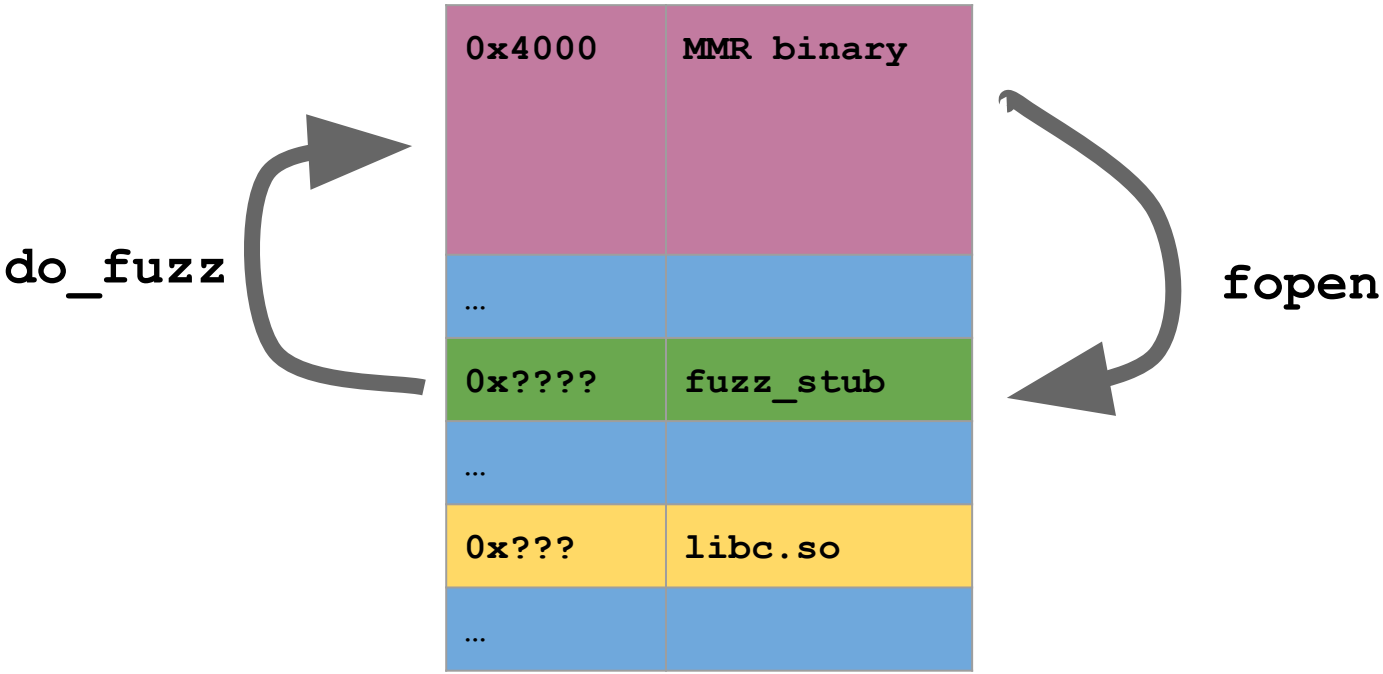
Sever Time!

- Decided to fuzz MMR server RTP processing
- MMR **did not have ASLR**
- Fuzzed server by executing it on the command line and loading fuzz driver stub using LD_PRELOAD and redefining `fopen`

MMR Fuzzing



MMR Fuzzing



Sever Time!

- “This approach has a lot of downsides”
  - Had to do a lot if initialization
- No results

Packet Processing

- Another type of packet found in the SSL_write dump appeared to be processed in libssb_sdk.so in the SDK
- Many classes had **`load_from`** and **`save_to`** defined
- Confirmed that these are part of the remote attack surface with Frida
- Reviewed entire interface

CVE-2021-34423

```
ssb::i_stream_t<ssb::msg_db_t,ssb::bytes_convertor>::operator>>(
        msg_db, &this->str_len, consume_bytes, error_out);
  str_len = this->str_len;
  if ( str_len )
  {
    mem = operator new[](str_len);
    out_len = 0;
    this->str_mem = mem;
    ssb::i_stream_t<ssb::msg_db_t,ssb::bytes_convertor>::
        read_str_with_len(msg_db, mem, &out_len);
```

CVE-2021-34423

```
int __fastcall ssb::i_stream_t<ssb::msg_db_t,ssb::bytes_convertor>::
      read_str_with_len(msg_db_t* msg, signed __int8 *mem,
      unsigned int *len)
{
  if ( !msg->invalid )
  {
      ssb::i_stream_t<ssb::msg_db_t,ssb::bytes_convertor>::operator>>(msg, len, (int)len, 0);
    if ( !msg->invalid )
    {
      if ( *len )
        ssb::i_stream_t<ssb::msg_db_t,ssb::bytes_convertor>::
read(msg, mem, *len, 0);
    }
  }
  return msg;
}
```

CVE-2021-34423

- Tested the bug on the MMR server
- Vulnerable code was behind a flag
- Traced it in IDA and determined it was for Webinar features
- Enabled bug for $81.73/month
- Not willing to try bug on public server

CVE-2021-34424

- The dyna para table is similar to JSON
  - Object with named properties of different types
  - Property values are variants

`ssb::dyna_para_table_t`

CVE-2021-34424

```
struct variant{
    char type;
    short length;
    var_data data;
};
```

```
union var_data{
        char i8;
        char* i8_ptr;
        short i16;
        short* i16_ptr;
        int i32;
        int* i32_ptr;
        long long i64;
        long long i64*;
};
```

CVE-2021-34424

- Identified functions that return value from table
  - Type checks were stringent
- Hooked these functions with Frida to see which values were accessed by the server, and investigated how there were used

CVE-2021-34424

- `user_name` string did not check null terminator
- String is displayed in web client with subsequent memory values appended
- Bug worked on server and client
  - Evidence that CVE-2021-34423 also worked on server

Exploit Attempt

- Tried to exploit server with both bugs
  - CVE-2021-34423 to move IP
    - Not very reliable, as overwritten vtable was not consistent
  - Two missing pieces
    - Controllable buffer I know the address of
    - Heap manipulation to control vtable
  - CVE-2021-34424 for extra pointer leakage

Exploit Attempt

- Got fairly close, but didn’t succeed
- Challenges were:
  - Lack of heap manipulation opportunities
    - Anti-abuse features
  - Threads with separate heap arenas
  - MMR server restart has exponential backoff

Conclusion

- Zoom fixed both reported bugs in November, 2021
- Zoom enabled ASLR in the MMR server
- Zoom has improved auto-updates

Conclusion

- Zoom was a challenging target that took a lot of effort to find viable attack surfaces, mostly due to the sheer quantity of code
  - Would have saved time if I'd focused on entry-points
- Locating symbols, even old ones, made this effort much easier
  - We also found symbols for WhatsApp and Viber
- IDA, Frida and DrSanCov are great tools that made this project easier

Questions

http://googleprojectzero.blogspot.com/

@natashenka
natalie@natashenka.ca