A 3-Years Tale of Hacking a Pwn2Own Target

The Attacks, Vendor Evolution, and Lesson Learned

Orange Tsai

DEVCORE

About This Talk

- Side story while doing Vulnerability Research
  1. Not just about how to reverse, how to exploit or 0day show-off
  2. More focused on thoughts, attempts and self-introspections while researching the target
- Push myself to sort out my workspace...

```
orange@work:~$ ls -hv sonos* | merge -to talk.pptx

sonos-2020:
  a.out        dump_key.py   gggg        parse.py     test.c
  a.tgz        exp.py        gpl/        parse2.py    test.pcap
  crash.py     exp-v2.py     log.txt     run.sh       test.txt
  data         extract/      log2.txt    run2.sh      test.xml
  data2/       ff            note.txt    tcpdump      test2.py
  data3/       file.crt      out/        smb.py       tmp/
  debug.txt    file2.crt     out.bin     s            tmp.txt

sonos-2021:
  dump/        trigger.py    note.txt    x            …

sonos-2022:
  …
```

Orange Tsai

- Specialize in Web and Application Vulnerability Research
  - Principal Security Researcher of DEVCORE
  - Speaker at Numerous Top Hacker Conferences
- Selected Awards and Honors:
  - 2022 - Champion and "Master of Pwn" of Pwn2Own
  - 2021 - Winner of Pwnie Awards "Best Server-Side Bug"
  - 2021 - Champion and "Master of Pwn" of Pwn2Own
  - 2019 - Winner of Pwnie Awards "Best Server-Side Bug"
  - 2018 - 1st place of Top 10 Web Hacking Techniques
  - 2017 - 1st place of Top 10 Web Hacking Techniques

Before the Journey

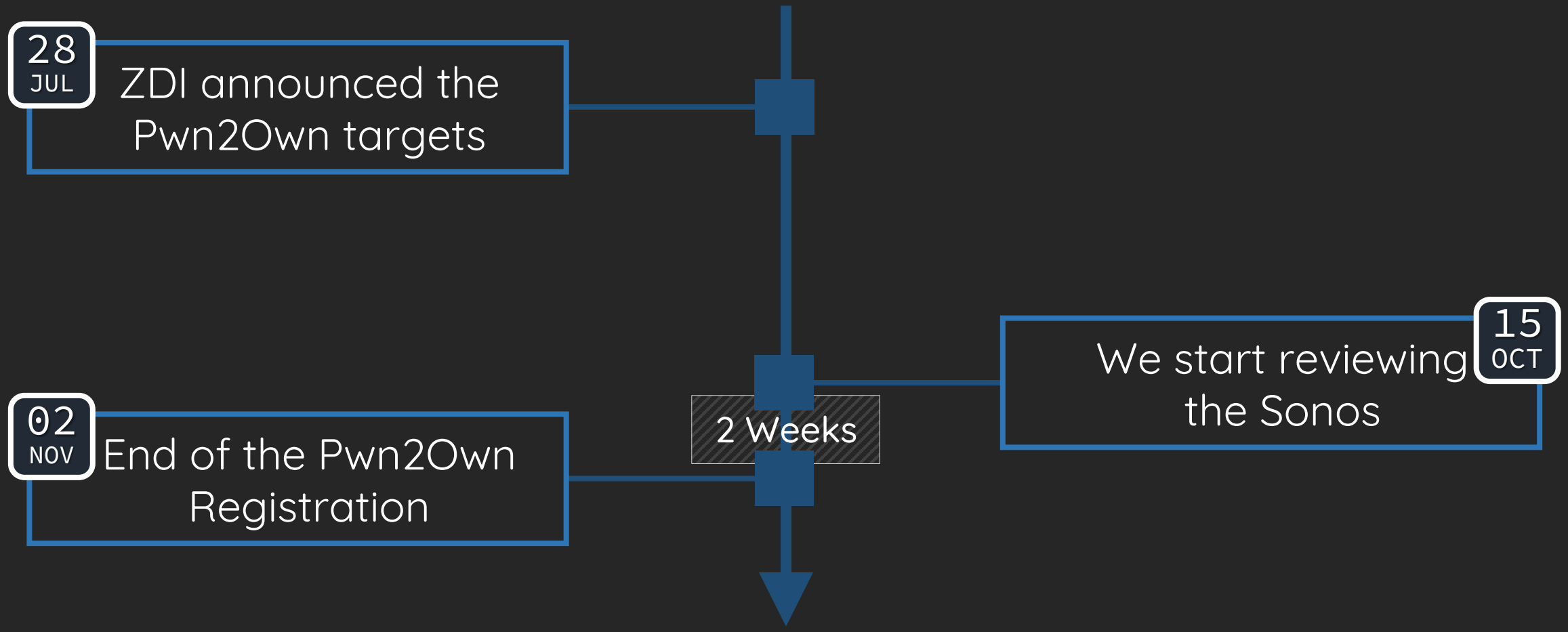
- We are more focusing on the application security. As for low-level views, you can check:
  - **Hardware Attacks:**
    - Dumping the Sonos One Smart Speaker by @_p0ly_
    - Gaining root access on Sonos Play Speakers by @Nicocha30
  - **Trust-Zone Attacks:**
    - Smart Speaker Shenanigans by @bl4sty

Why I am Targeting Sonos?

1. Would like to try something different
2. High rewards and no one has pwned it before

Name	Target	Award	Pwned
Pwn2Own Tokyo 2020	Sonos One Speaker	40,000 USD	0
Pwn2Own Austin 2021	Sonos One Speaker	60,000 USD	2
Pwn2Own Toronto 2022	Sonos One Speaker	60,000 USD	3
Pwn2Own Toronto 2023	Sonos Era 100	60,000 USD	?

Pwn2Own Tokyo 2020



2020 - Our First Year

- We don't have any physical device :(
- Our attempts:
  - ❌ Fuzzing all web inputs
  - ❌ Searching for firmwares
  - ❌ Exploiting the Firmware OTA

Web Interface of Sonos

- Most of the info was collected through fuzzing and the Internet
  - Web pages: mainly for showing status and debugging
    1. /status
    2. /tools.htm
    3. /devmode
    4. /unlock
  - UPnP: Contains hundreds of SOAP actions
    1. AVTransport.play(...)
    2. AlarmClock.CreateAlarm(...)
    3. AVTransport.AddURIToQueue(...)
    4. RenderingControl.SetVolume(...)

192.168.80.239:1400/tools.htm

不安全 | 192.168.80.239:1400/tools.htm

Tools for debugging network issues

Ping

Ping6

Traceroute

Traceroute6

Nslookup

mDNS Announce

Orange
used COMMAND INJECTION!

```
1.2.3.4; sleep 5 | $(sleep 5) & `sleep 5`
```

It's not very
effective...

SoCo: Sonos Controller

```
from soco import SoCo

zone = SoCo('192.168.12.34')
zone.volume += 10
zone.play_uri('http://t.co/music.mp3')

zone.get_current_track_info()['title']
```

SoCo:

```
from so

zone =

zone.vo

zone.pl

zone.ge
```

BeginSoftwareUpdate?

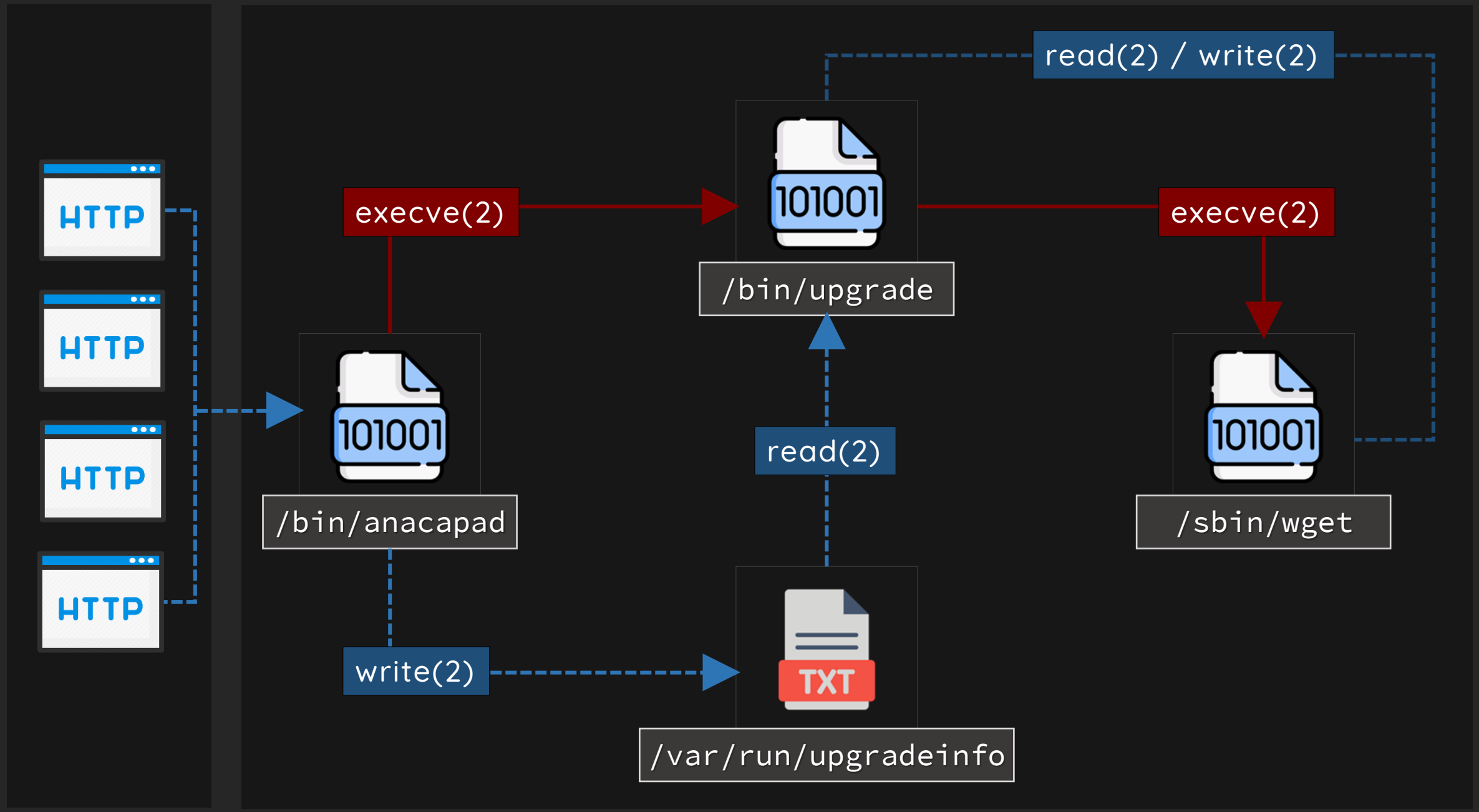
```
from soco import SoCo

zone = SoCo('192.168.12.34')
zone.zoneGroupTopology.BeginSoftwareUpdate((
    ['UpdateURL', 'https://<my-server>/'],
    ['Flags', 1],
    ['ExtraOptions', '']
))
```

```
orange@work:~$ sudo ncat -lp 80

GET /?cmaj=71&cmin=1&cbld=... HTTP/1.1
Host: 10.26.0.34
User-Agent: Wget
Connection: close
```

Collecting Firmwares

- Few firmwares are available on the Internet
  - Newer firmwares are not binwalk-able :(
  - Brute-forcing download URLs (but we failed)
  - The newest and binwalk-able firmware version is 45.1-56150, which is released on 2018-08-15

```
http://update-firmware.sonos.com/firmware/Prod/
57.16-41110-v11.9-wzhipjet-GA-1/^57.16-41110
```


HTTP
HTTP
HTTP
HTTP
execve(2)
101001
/bin/upgrade
read(2) / write(2)
execve(2)
101001
/bin/anacapad
read(2)
101001
/sbin/wget
write(2)
TXT
/var/run/upgradeinfo

Attacking Firmware OTA

- Our attempts:
  - ❌ SSRF!
    - Can't locate a good local service to exploit :(
  - ❌ Wget (bundled in BusyBox) CVEs / Vulnerabilities
  - ❌ Firmware encapsulating attacks
    - Backdooring - Signed with an RSA key stored in the Secure Storage
    - Downgrading- Protected with a SHA-256 Crypt hash
  - ❓ Firmware parser

Attacking Firmware OTA

- Our attempts:
  - ✕ SSRF!

5NeHanrecdehym$x9aL.1kgod2FMyYGKajtuJztE/cy4O2GY64dhTwMTGD

  - ✕ Wget (bundled in BusyBox) CVEs / Vulnerabilities
  - ✕ Firmware encapsulating attacks
    - Backdooring - Signed with an RSA key stored in the Secure Storage
    - Downgrading - Protected with a SHA-256 Crypt hash
  - ? Firmware parser

Exploiting Firmware Parser

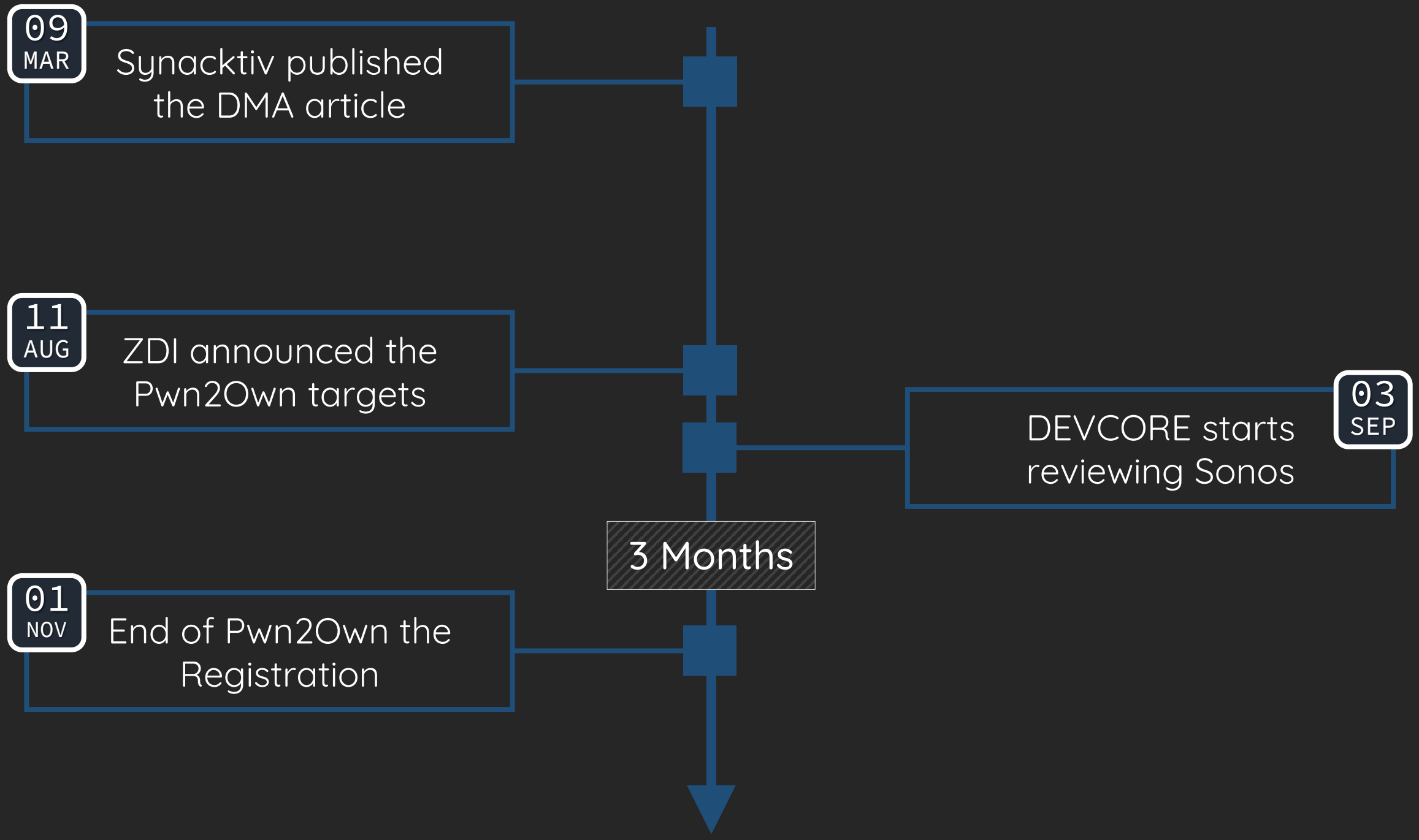
- Just like traditional CTF challenge
  - Standalone binary parsing customized formats...
- Buffer Overflow never die... 😉
  - The exploit works on my local QEMU environment but failed with the latest firmware
  - Adjust the offsets/gadgets blindly until the competition end :(

Summarize Our First Year

1. Got an exploit which works on old firmware (2018 ver.)
2. Self-reflections:
   - Fine, it's fair given the two-weeks time frame
   - My reverse skill is still too slow, especially in C++ :(

Pwn2Own Austin 2021

- Why are you targeting Sonos again?
  - I dislike the feeling of defeat
  - Synacktiv published a detailed article for dumping Sonos memory by DMA attack
    - Can learn a new skill and understand last year's failure. WAKU WAKU!


09 MAR
Synacktiv published the DMA article
11 AUG
ZDI announced the Pwn2Own targets
03 SEP
DEVCORE starts reviewing Sonos
3 Months
01 NOV
End of Pwn2Own the Registration

Dumping the Firmware

- Hardware: Purchased the USB3380 Evaluation Board
  - Mostly sold out, but luckily one of the reseller is based in Taiwan
  - Got within 24 hours
- Software: Perform the DMA Attack by @ufrisk/PCILeech
  - Stuck for an entire week


MY USB3380EVB

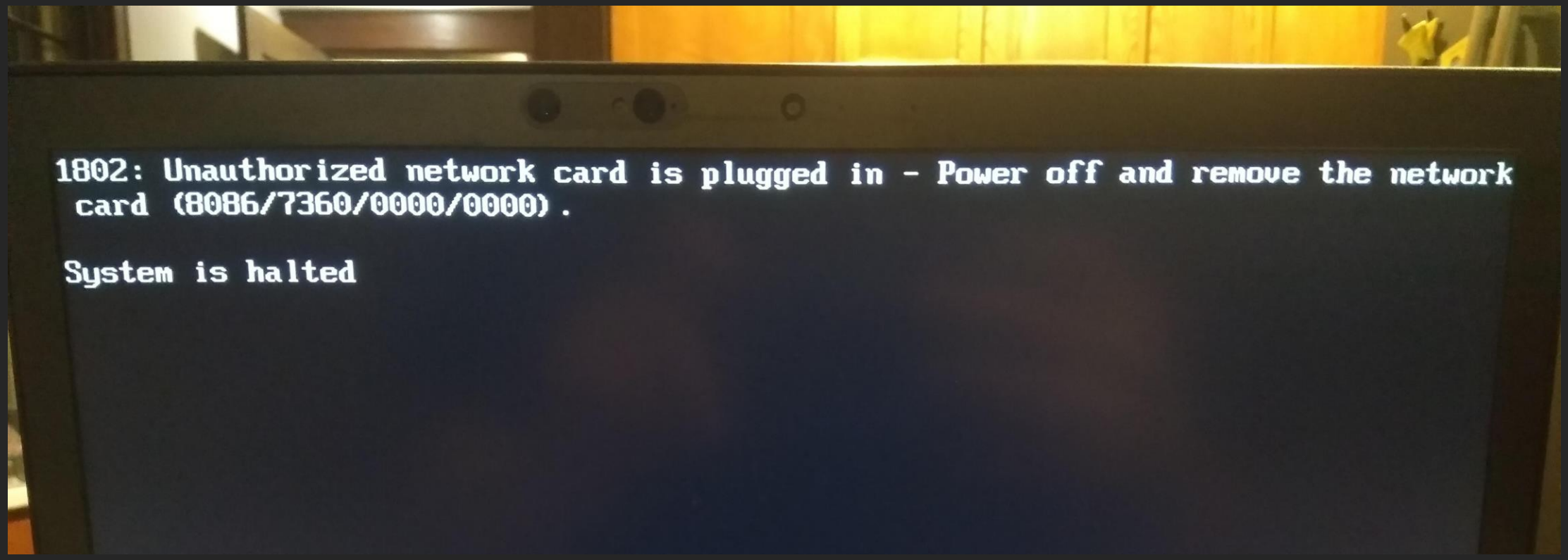
"The USB interface of the USB3380 is however disabled by default and the device would need to be flashed before it's enabled"

Struggling with USB3380EVB

- Flashing USB3380EVB:
  - Hard to find mini PCIe to Micro USB adaptor
  - Personal Computer only has PCIe x1, x4, x8, and x16 slots
  - Modern Laptop only has M.2 slots
- Mini PCIe was only used in laptops during the 2010s
  - Borrowed a Lenovo ThinkPad T430s from my friend

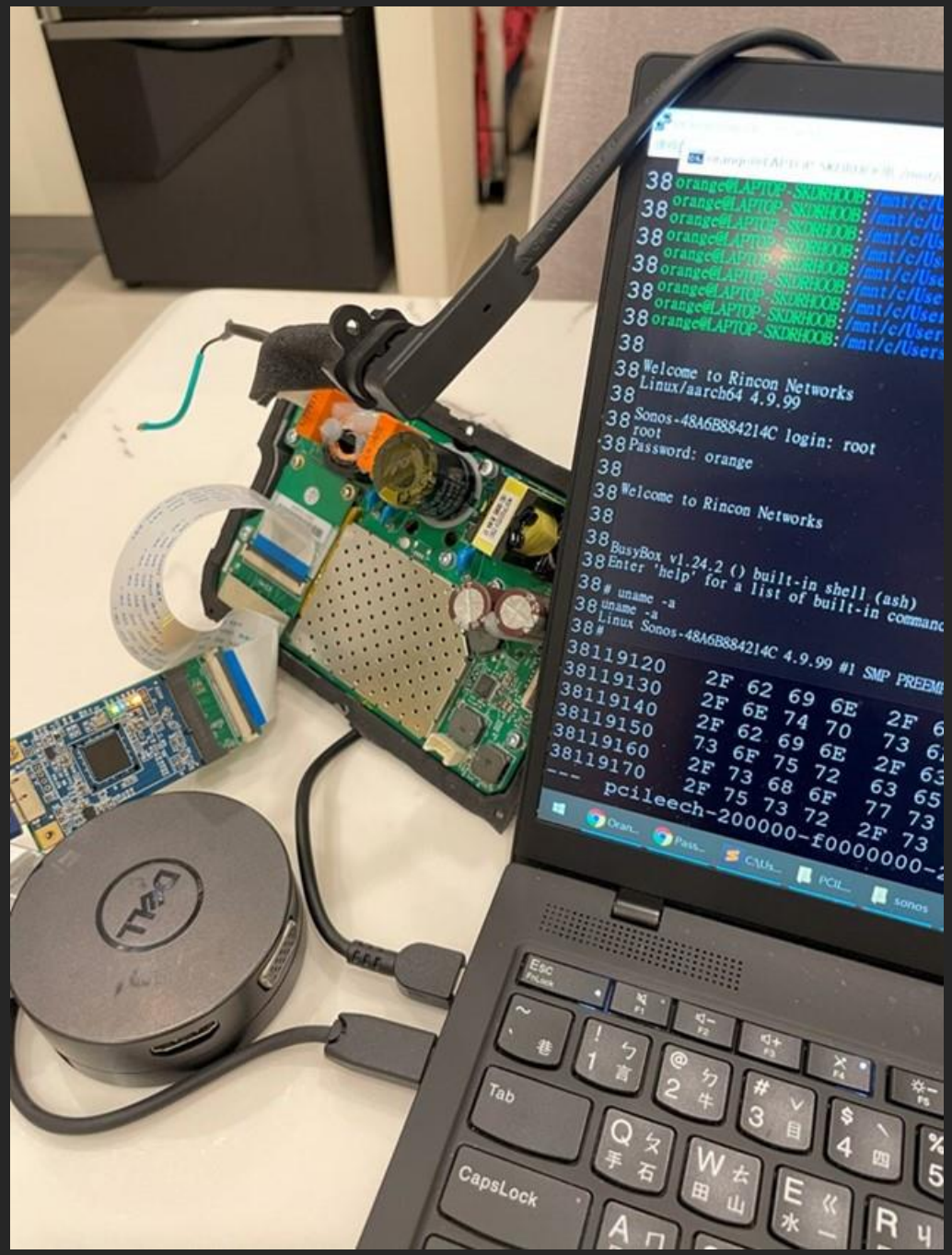
Removing Wireless WIFI Card to Flash EVB

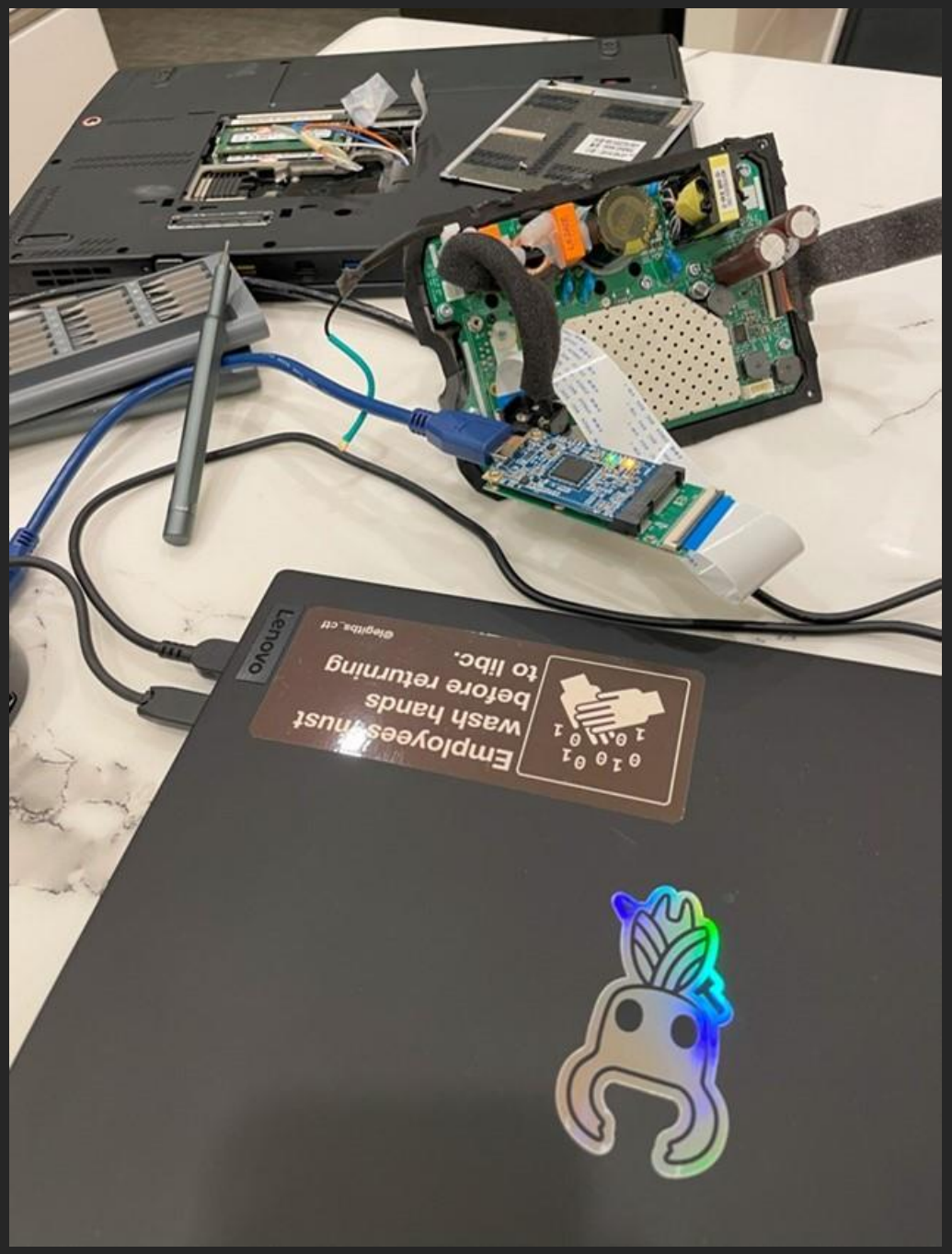
BIOS Mini PCIe Whitelist...😅



Bypass BIOS whitelist check

- Downgrade the BIOS
  - Jailbreak the BIOS
    - Flash the custom BIOS image


orange@LAPTOP-SKDRHOOB:/mnt/c/Users
Welcome to Rincon Networks
Linux/aarch64 4.9.99
Sonos-48A6B884214C login: root
Password: orange
Welcome to Rincon Networks
BusyBox v1.24.2 () built-in shell (ash)
Enter 'help' for a list of built-in commands.
uname -a
Linux Sonos-48A6B884214C 4.9.99 #1 SMP PREEMPT
38119120 2F 62 69 6E 2F 6
38119130 2F 6E 74 70 73 6
38119140 2F 62 69 6E 2F 63
38119150 73 6F 75 72 63 65
38119160 2F 73 68 6F 77 73
38119170 2F 75 73 72 2F 73
--- pcileech-200000-f0000000-
DELL
Tab
CapsLock


Employees must
wash hands
before returning
to libc.
LENOVO

2021 - Our Second Year

- No-Brainer attempts:
  - ❌ Reviewed all web debug routes implementations
  - ❌ Reviewed all system(3) / exec*(3) calls
  - ❌ Reviewed all recv(3) / recvfrom(3) / recvmsg(3) on network services
  - ❌ Reviewed all unsafe string operations
    - Still overlooked an information leak (my fault 🤦‍♀️), which played an important role during the third year's competition

2021 - Our Second Year

- Use the brain to think:
  1. Sonos supports lots of audio formats
  2. Audio parser are all based on open-source projects
     - Fuzzing seems promising, but I prefer discovering bugs with my own eyes
  3. Extracting music metadata (such as song title/album/author) could present a more attractive attack surface 🤔
     - Because all the metadata parsers are handcrafted

2021 - My First Bug

```
size_t read_size = 1;
my_tsclient_read(ctx, &dlen, &read_size, timeval);

dlen = (unsigned __int8) dlen;
if (dlen) {

    my_tsclient_read(ctx, &buffer, &dlen, timeval);
}
```

```
-00000000000000D0 ; D/A/*    : change type (data/ascii/array)
-00000000000000D0 ; N        : rename
-00000000000000D0 ; U        : undefine
-00000000000000D0 ; Use data definition commands to create local variables and function arguments.
-00000000000000D0 ; Two special fields " r" and " s" represent return address and saved registers.
-00000000000000D0 ; Frame size: D0; Saved regs: 0; Purge: 0
-00000000000000D0 ;
-00000000000000D0
-00000000000000D0 buffer          DCQ ?                    ; starting from 0xD0
-00000000000000C8                 DCB ? ; undefined
```

```
-00000000000000D0
-00000000000000D0 buffer
-00000000000000C8
```

```
    my_media_read(ctx, &buffer, &dlen, timeval);
}
```

2021 - My First Bug

```
size_t read_size = 1;
my_tsclient_r                                  e, timeval);

dlen = (unsig
if (dlen) {

    my_tsclient_read(ctx, &buffer, &dlen, timeval);
}
```

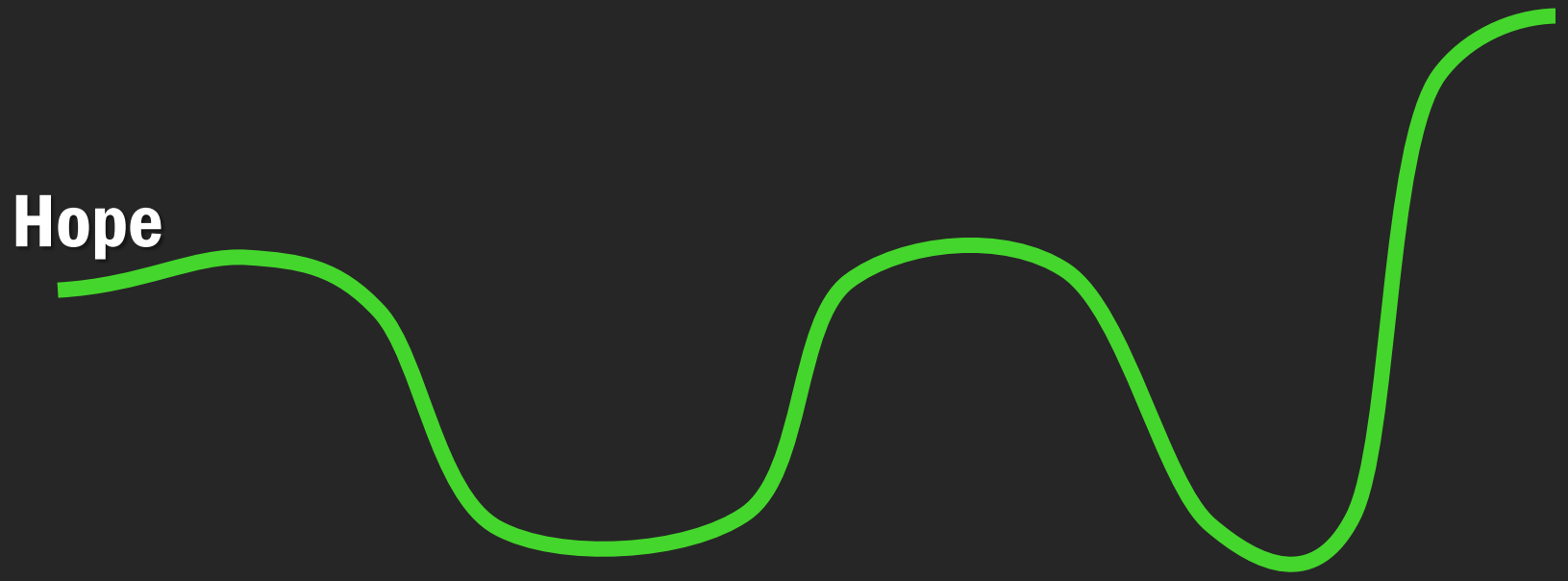
```
MOV     X0, $ctx
SUB     SP, SP, $dlen
MOV     X1, SP
MOV     X2, SP
MOV     X3, $timeval
```

2021 - My First Bug

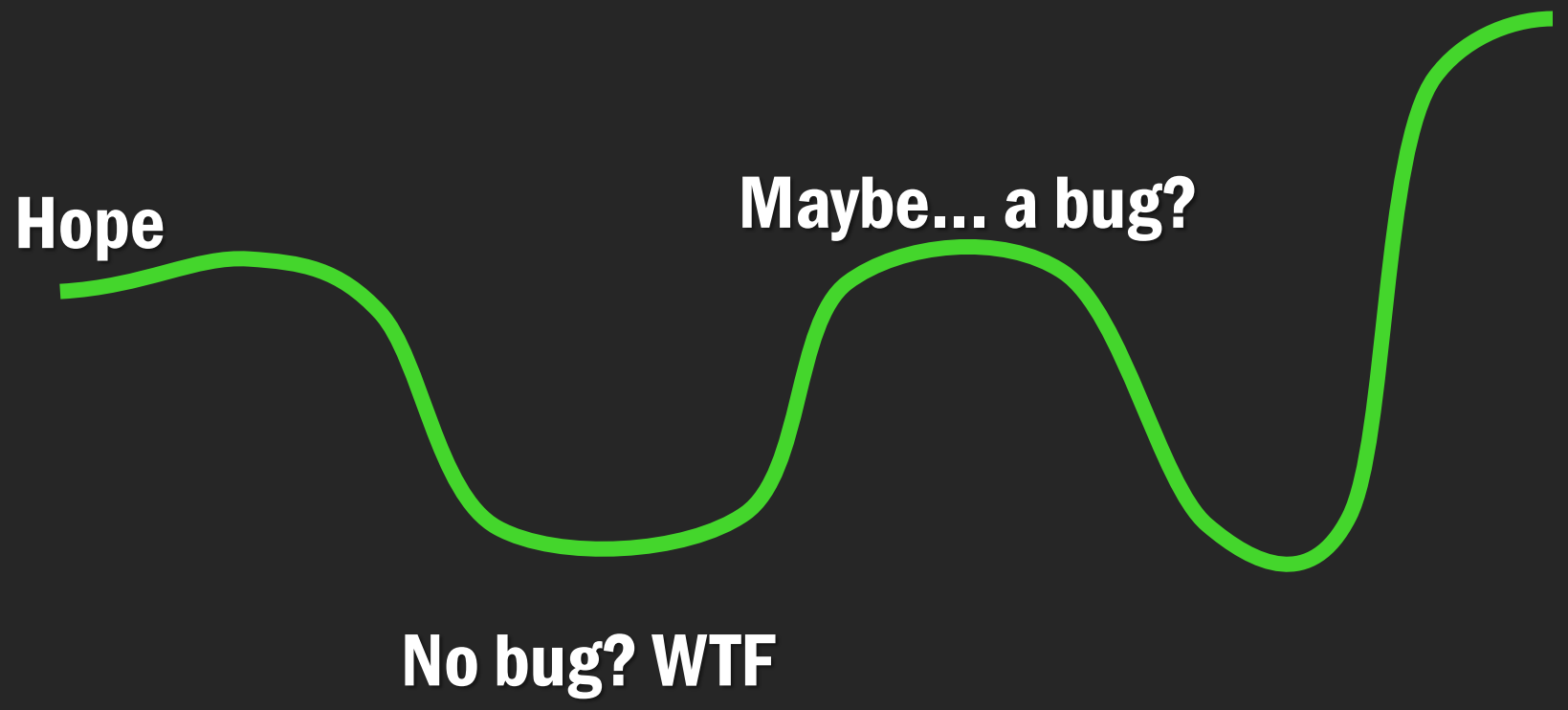
```
size_t read_size = 1;
my_tsclient_read(ctx, &dlen, &read_size, timeval);

dlen = (unsigned __int8) dlen;
if (dlen) {
    char *buffer = alloca(dlen);
    my_tsclient_read(ctx, &buffer, &dlen, timeval);
}
```

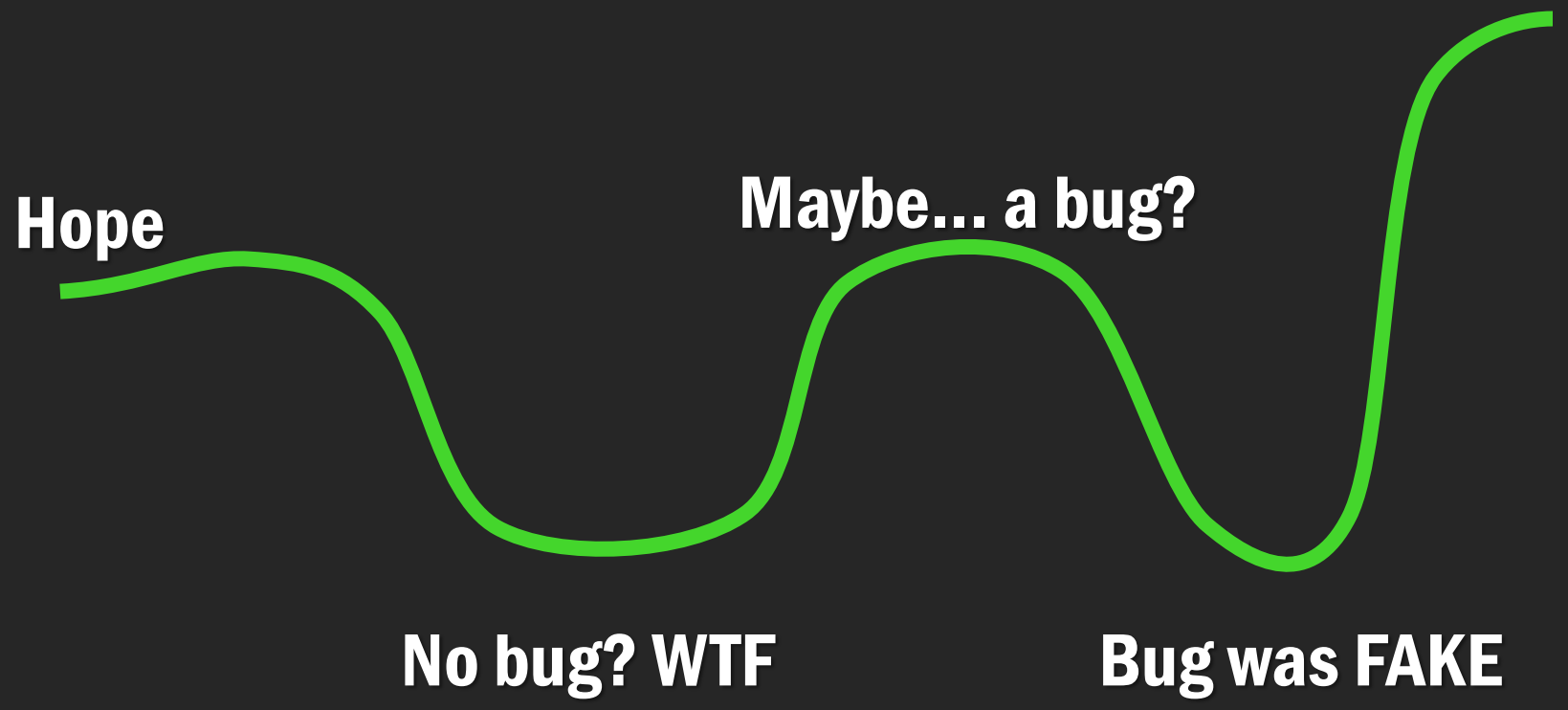
2021 - My First Fake Bug

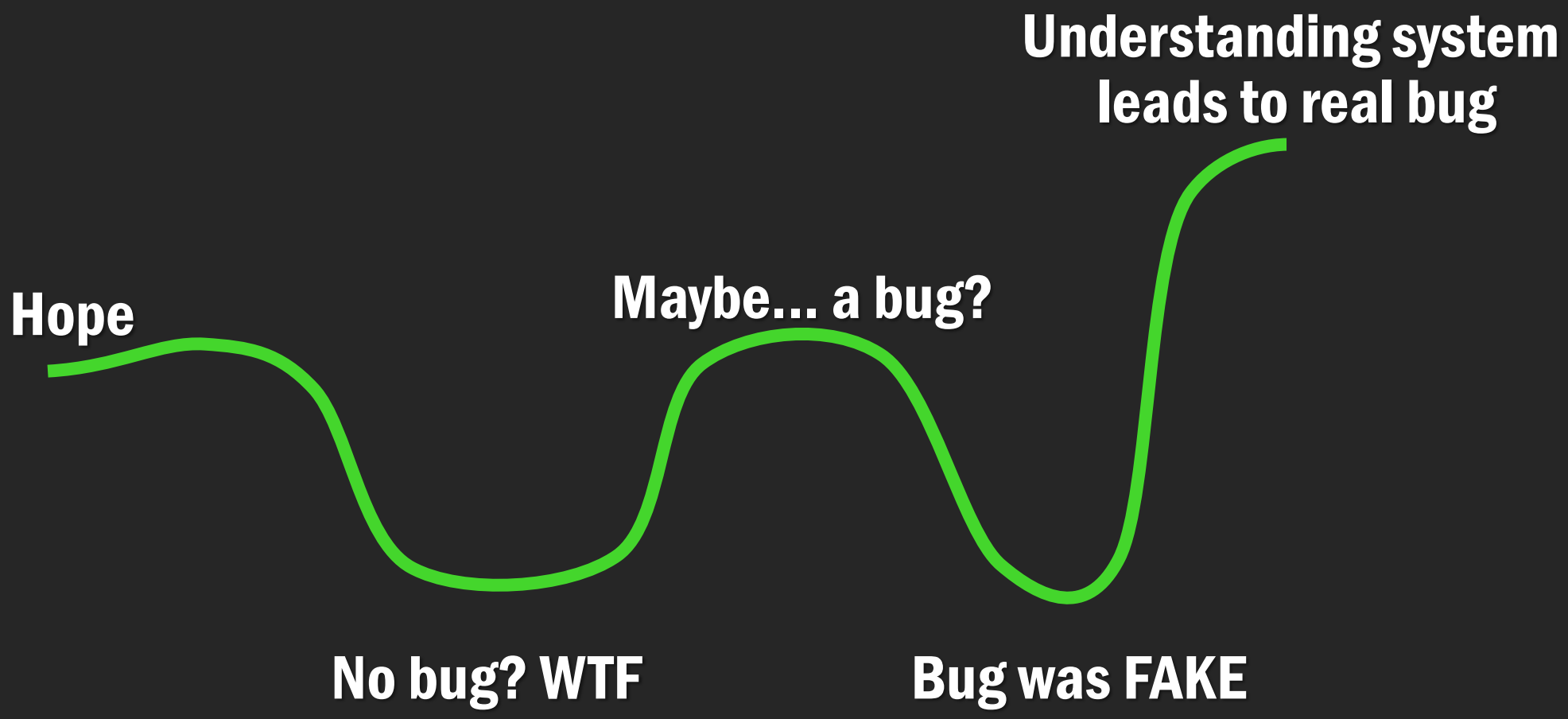
```
size_t read_size = 1;
my_tsclient_read(ctx, &dlen, &read_size, timeval);

dlen = (unsigned __int8) dlen;
if (dlen) {
    void *buffer = alloca(dlen);
    my_tsclient_read(ctx, &buffer, &dlen, timeval);
}
```


Hope


Hope
No bug? WTF


Hope
Maybe... a bug?
No bug? WTF


Hope
Maybe... a bug?
No bug? WTF
Bug was FAKE


Understanding system
leads to real bug
Hope
Maybe... a bug?
No bug? WTF
Bug was FAKE

2021 - My Real Bug

- A bug triggered while parsing a malformed ID3v2 tag
  - An ID3v2 tag consists of multiple frames such as TPE1/COMM

<table>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th></tr>
<tr><td colspan="3">ID3 magic</td><td>version</td><td>revision</td><td>flags</td><td colspan="4">ID3 size</td></tr>
<tr><td colspan="4">frame name</td><td colspan="5">frame size</td><td colspan="2">frame flags</td></tr>
<tr><td colspan="10">frame data</td></tr>
</table>

2021 - My Real Bug

- A bug triggered while parsing a malformed ID3v2 tag
  - An ID3v2 tag consists of multiple frames such as TPE1/COMM
  - Integer Underflow due to the calculation of `**sizeof(string) - 1**`

1	2	3	4	5	6	7	8	9	10
"ID3"			03	00	BF	00 00 1234			
"TPE1"				**00 00 00 00**				03 03	
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA…"									

```
size_t string_len = frame->size - 1;  // underflow

if (obj->unsynchronised_flag) {
    if (string_len > obj->id3_size) goto fail;
    ret = obj->mp3_read(obj, buffer, string_len);
} else {
    while (obj->id3_size && string_len) {
        if (!obj->mp3_read(obj, buffer, 1))
            goto fail;
        *buffer++;
        obj->id3_size--; string_len--;
    }
}
```

WHEN YOUR CAT IS PLAYING OVERFLOW

2021 - My Real Bug

- Replaced the GOT of `strcasecmp` to `system` to execute commands
  - Really Easy!

```
Arch:     aarch64-64-little
RELRO:    Partial RELRO
Stack:    No canary found
NX:       NX enabled
PIE:      No PIE (0x400000)
FORTIFY:  Enabled
```

Summarize Our Second Year

1. A good attack surface achieves twice results with half the effort
2. Self-reflections:
   - Spent too much time blindly trying on the wrong path :(
   - Perhaps I should give fuzzing a try?
   - Always read the manual carefully...😅

Pwn2Own Toronto 2022

- Why are you targeting Sonos again and again?
  - Low cost, high return! This year must be the same 🤑
    - 2020: 2 weeks black-boxing
    - 2021: 1 week for hardware + 2 weeks for reversing/exploiting = 60K USD
    - 2022: (Spoiler) Spend **FULL THREE MONTHS** on this target...

Pwn2Own Toronto 2022

- Before the competition, I am chatting with @FidgetingBits at HITCON CMT 2022:

@FidgetingBits: "Sonos has already enabled all binary protections"

@orange_8361: "??????"

```
Arch:     aarch64-64-little
RELRO:    Full RELRO
Stack:    Canary found
NX:       NX enabled
PIE:      PIE enabled
FORTIFY:  Enabled
```

WTF?

2022 - Our Third Year

- Continue to explore our last year's good attack surface

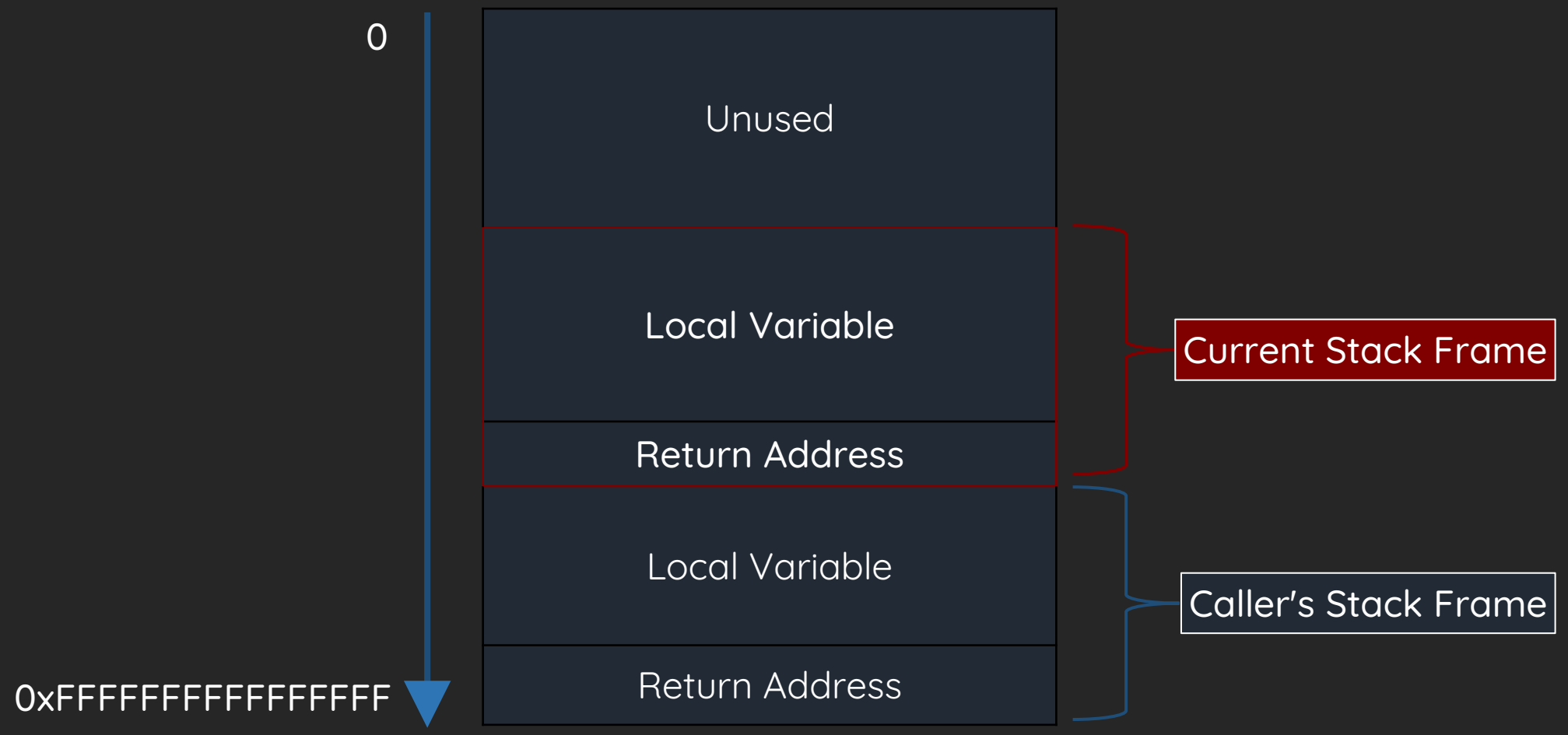
2022 - Our Third Year

- Arbitrary size alloca(3) while parsing MP4 FTYP box

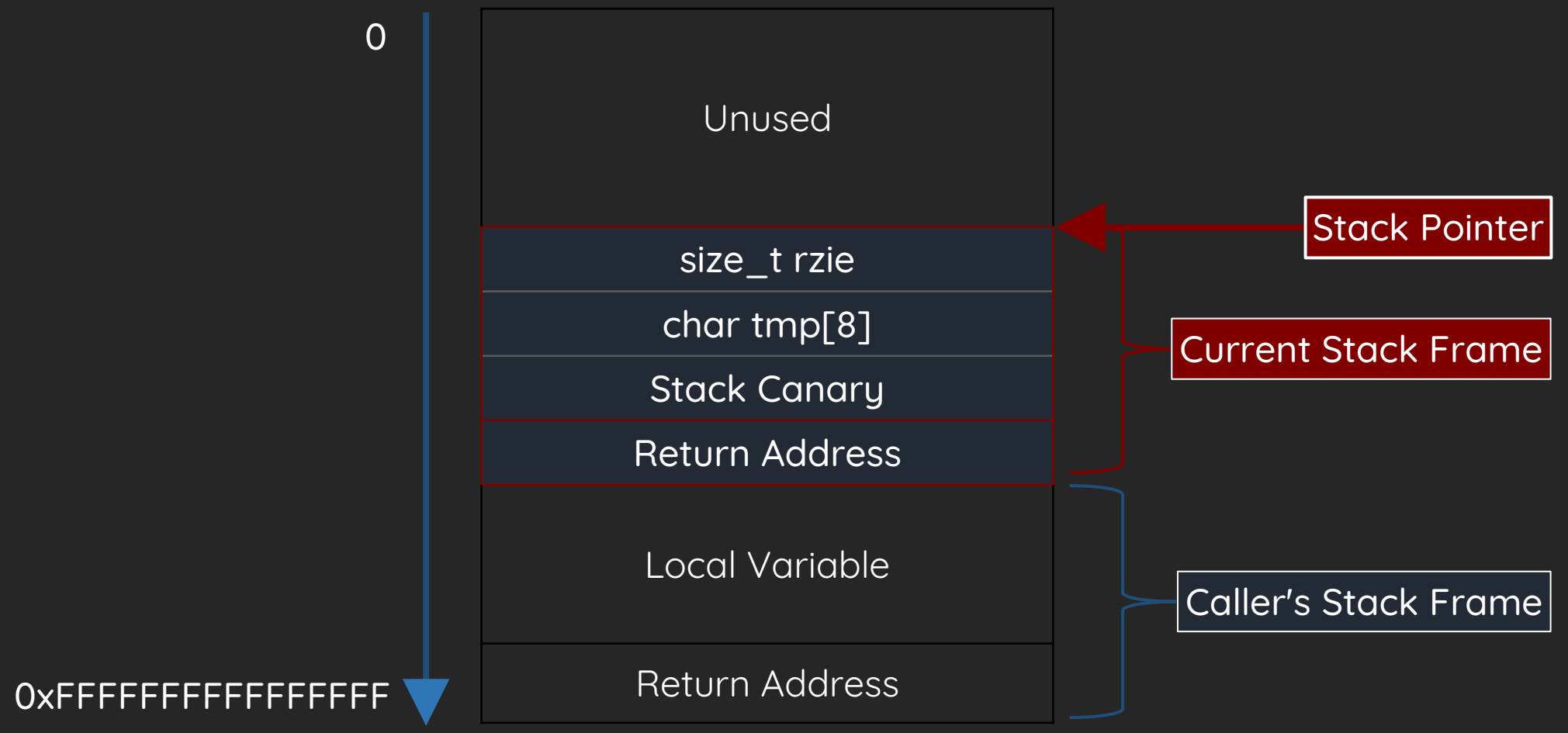
```
┌──────────────────────────┐
│        Box Header        │
│ Size (4) | Type (4)      │  Box Header = 8 Bytes
│ ------------------------ │
│       Box Data (N)       │  Box Data   = N Bytes
└──────────────────────────┘
                 └──────────  Box Size   = 8 + N bytes
```

```c
bool mp4_parse_ftyp(void *ctx, void *s, size_t box_size)
{

  if ( box_size > 7 ) {
    rsize = s->read_mp4(stream, tmp, 8);
    if ( rsize == 8 ) {
      box_size = box_size - 8;
      char *buffer = alloca(box_size);
      rsize = s->read_mp4(stream, buffer, box_size);
    }
  }
}
```

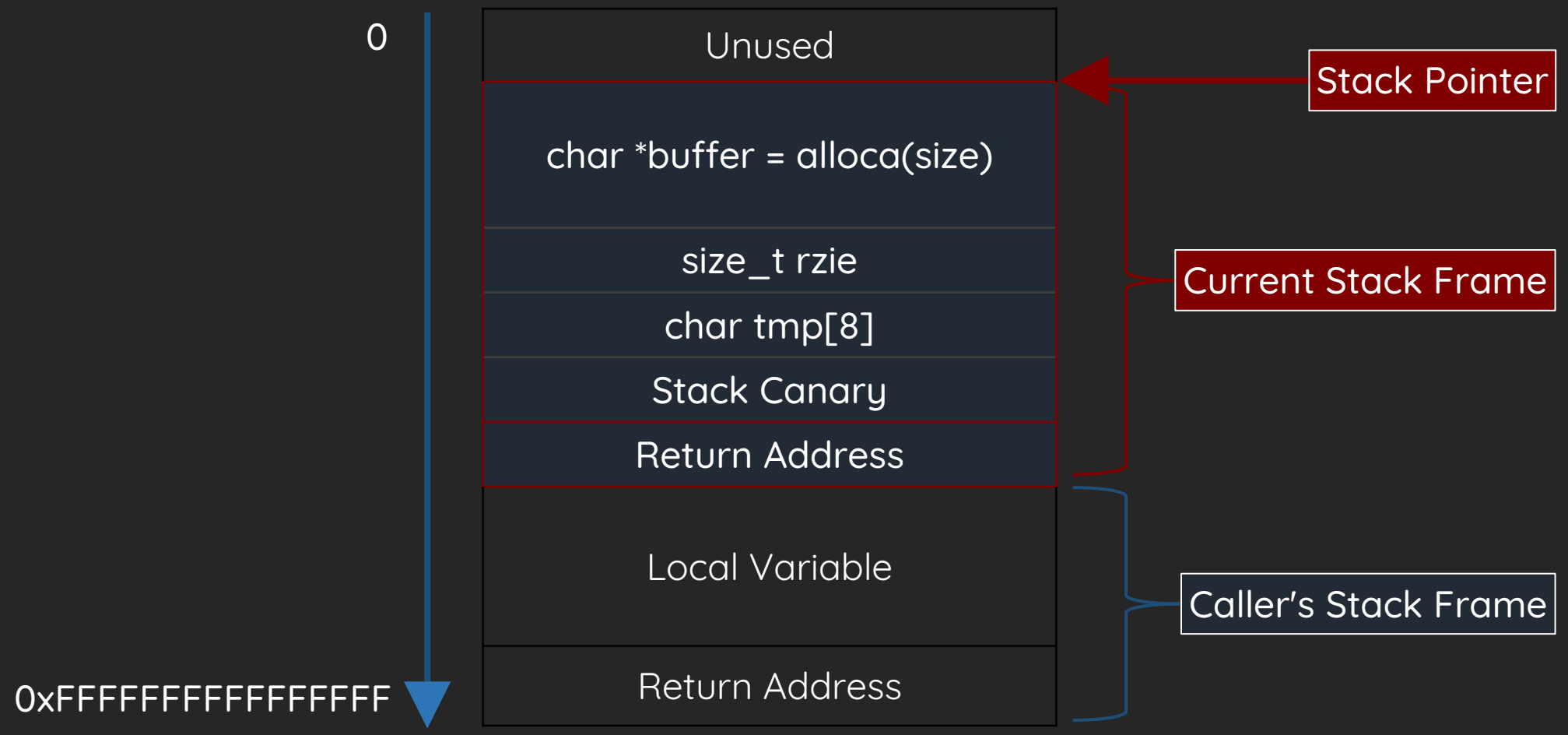
Arbitrary Size alloca(3)?



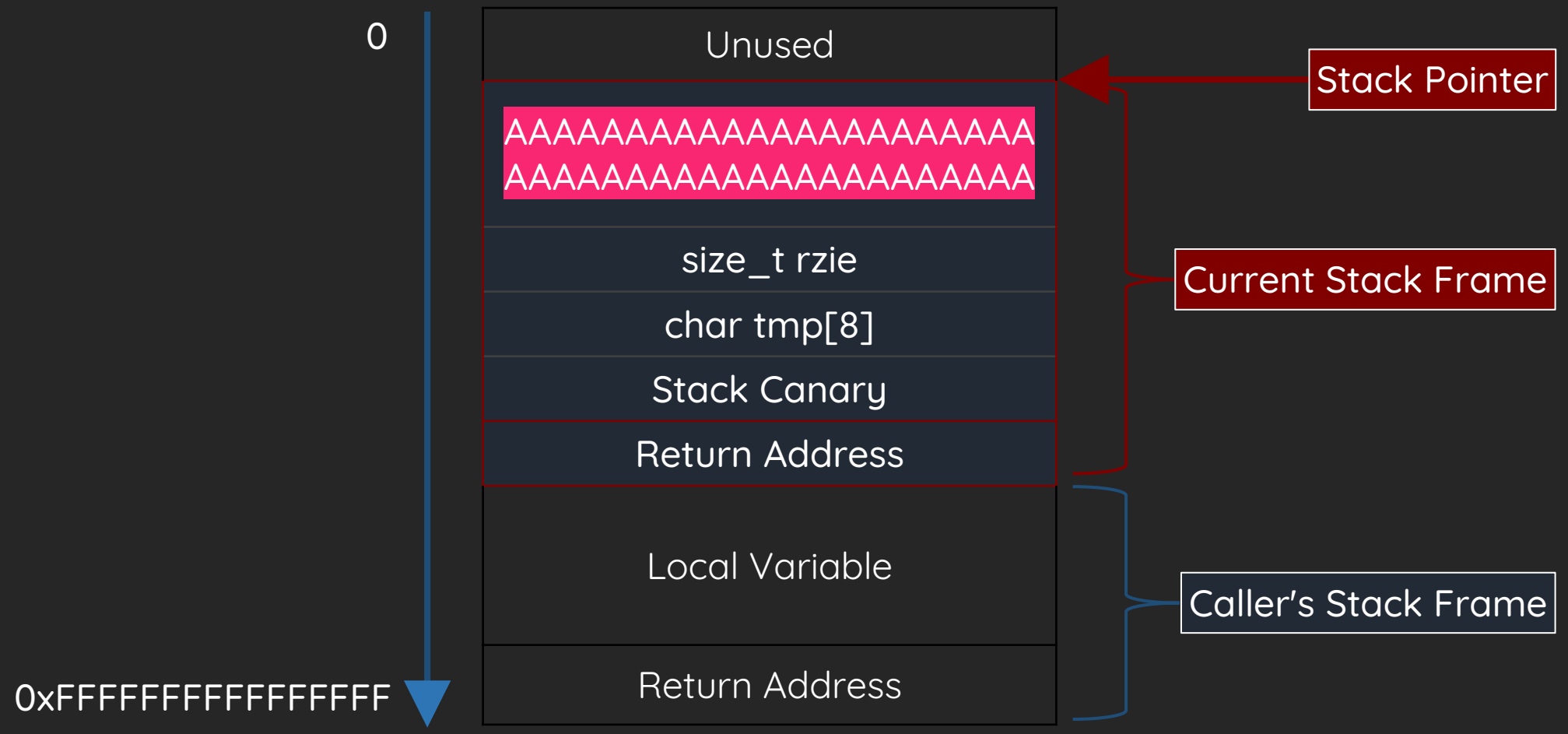
Arbitrary Size alloca(3)?



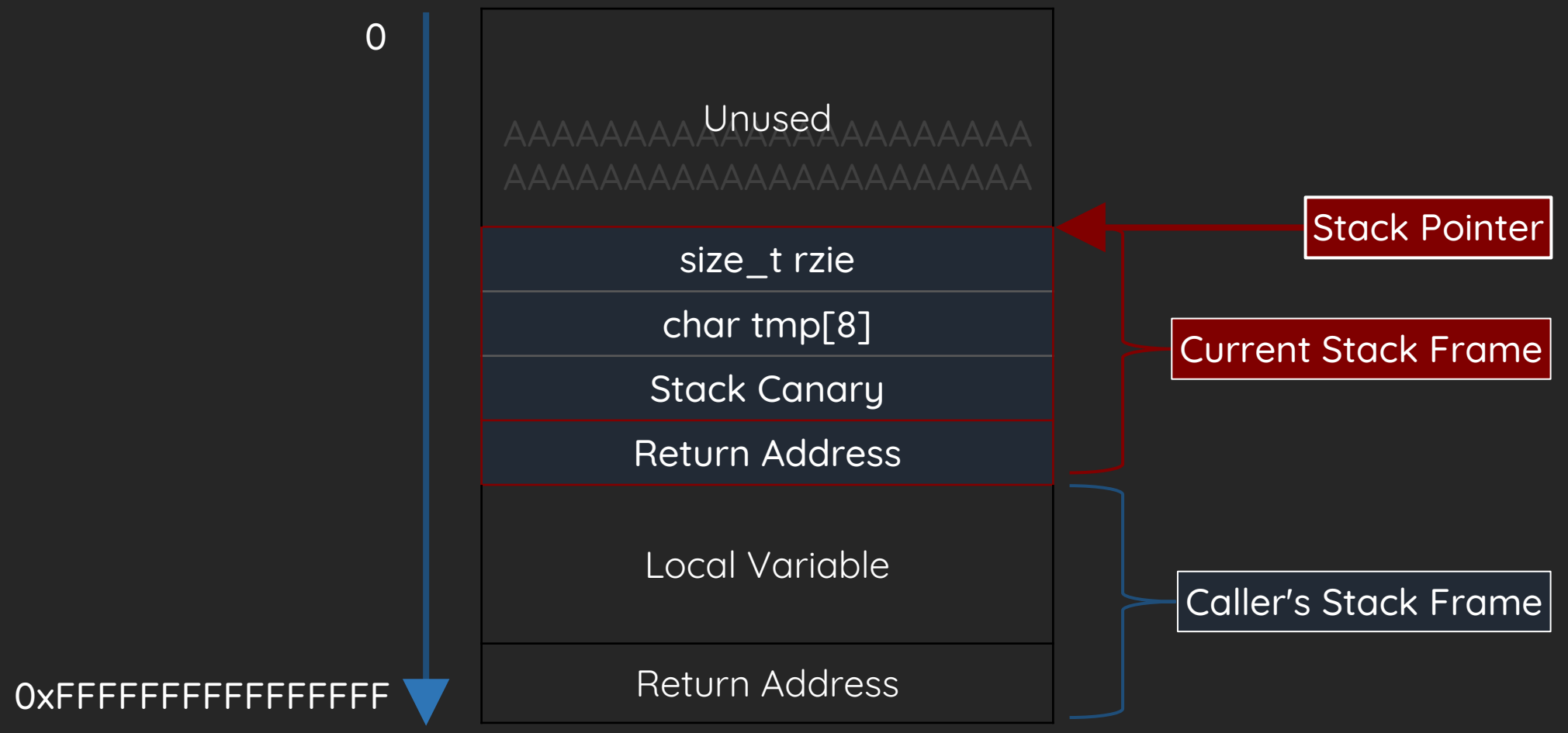
Arbitrary Size alloca(3)?



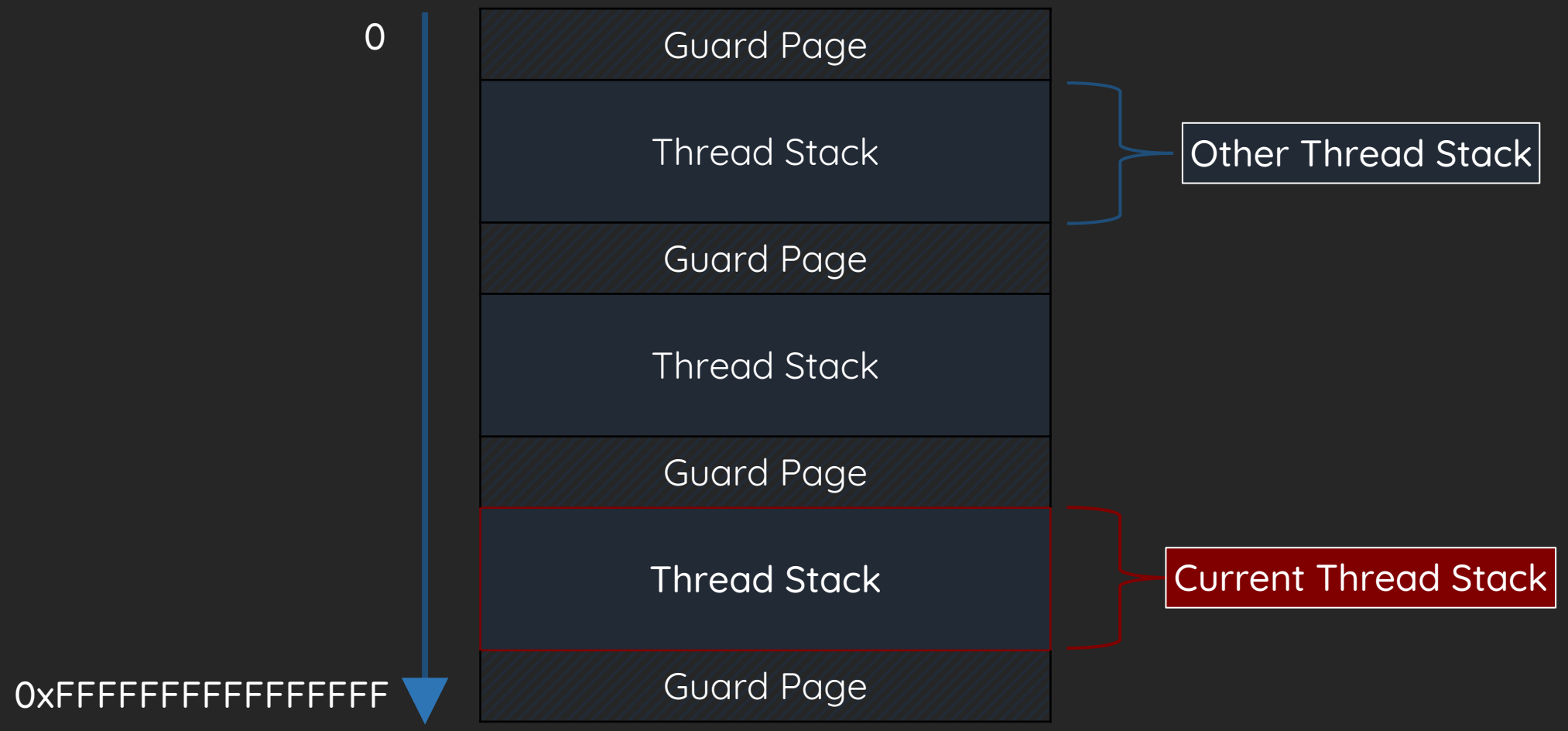
Arbitrary Size alloca(3)?



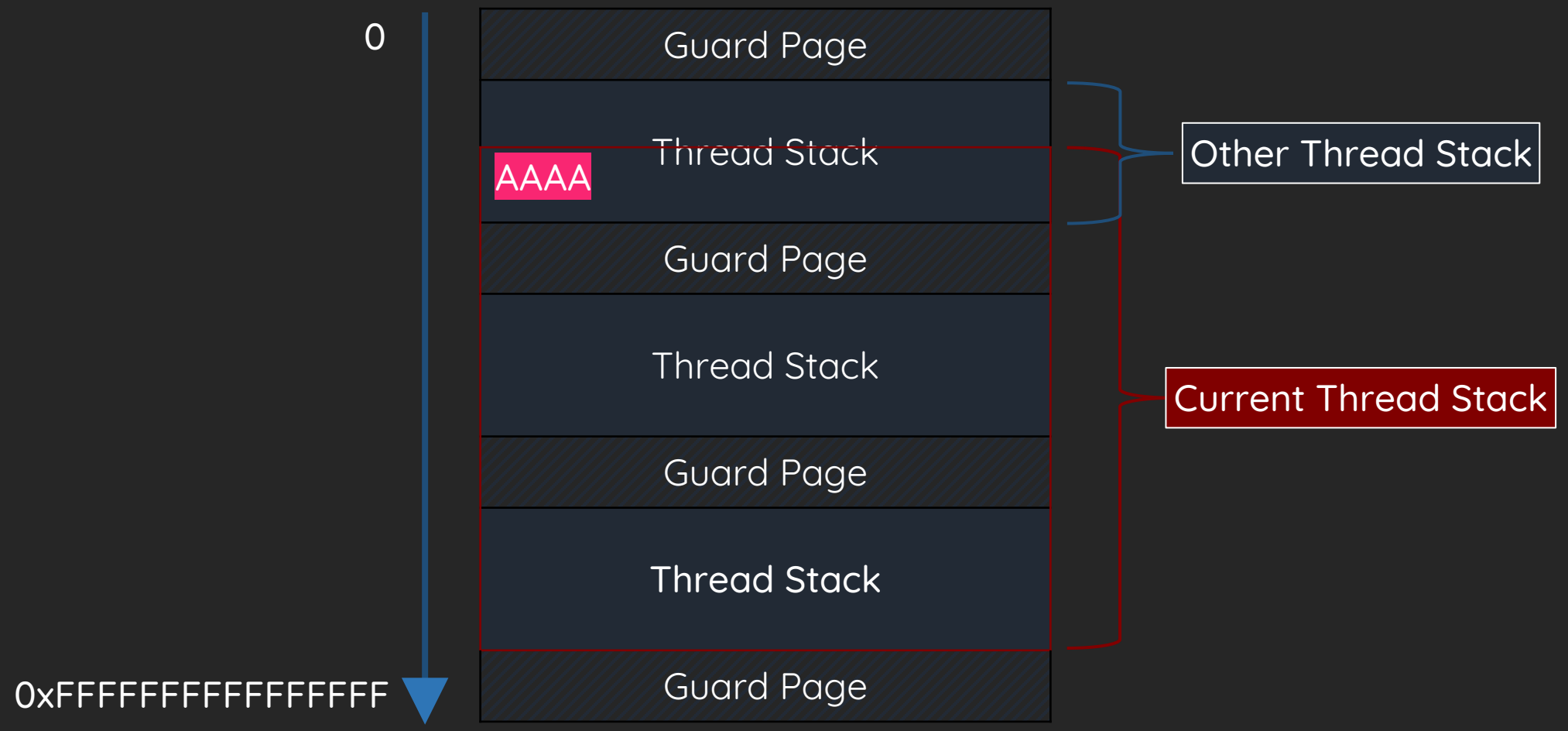
Arbitrary Size alloca(3)?



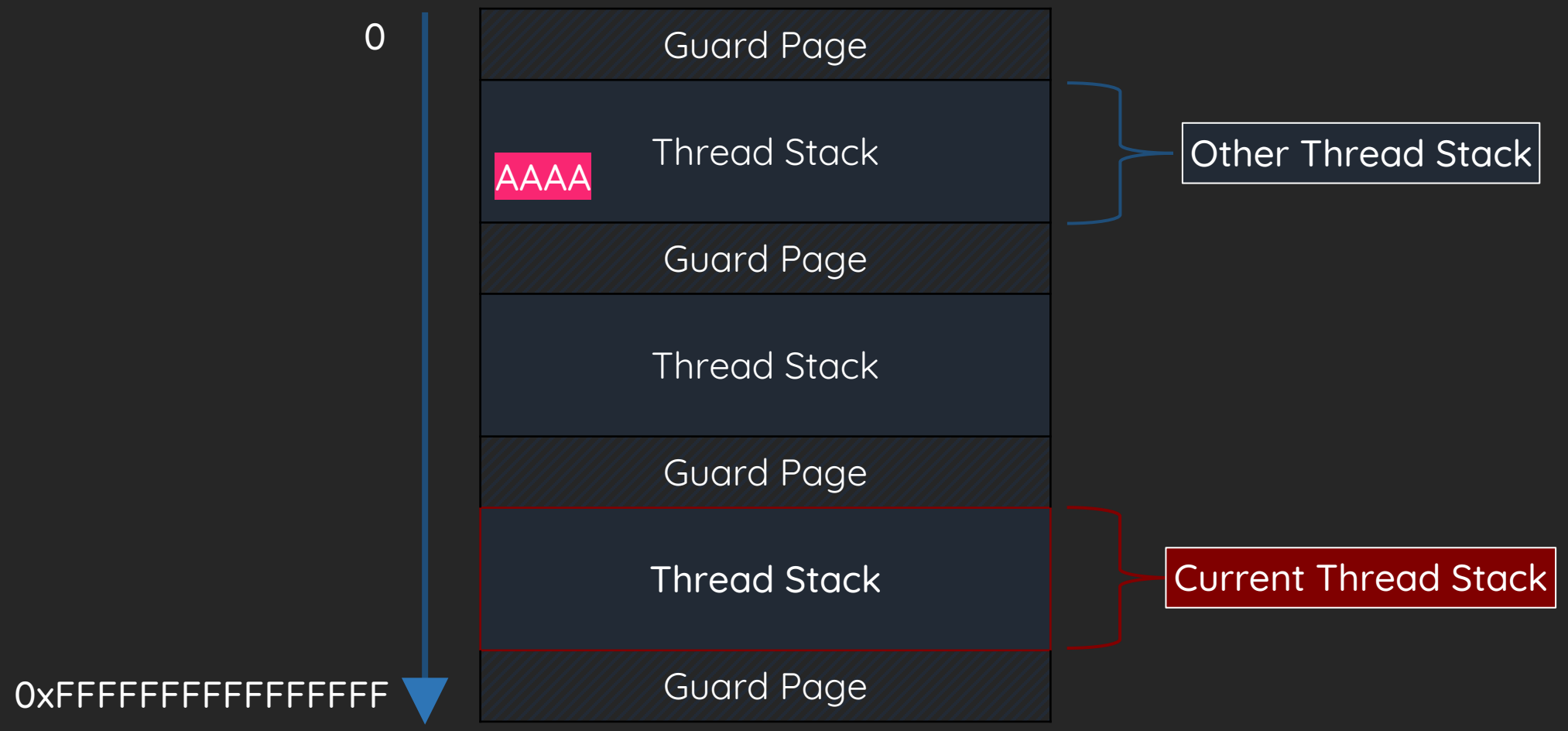
Arbitrary Size alloca(3)?



Arbitrary Size alloca(3)?



Arbitrary Size alloca(3)?



Arbitrary Size alloca(3)?

- Turned this arbitrary size alloca(3) into a Stack Clash bug
  - How to convert this Stack Clash into Read/Write primitives
- Overcame lots of exploitation obstacles:
  - Unstable memory layout
  - The write primitive has side effects...
  - Determining the right moment to overwrite while all other threads continually spin

Arbitr ?

- Turn this ar sh bug
  - How to c nitives
- Lots of que
  - The men
  - The write
  - All other

192.168.80.239:1400/tools.htm

不安全 | 192.168.80.239:1400/tools.htm

Tools for debugging network issues

Ping

Ping6

Traceroute

Traceroute6

Nslookup

mDNS Announce

Exploiting the Stack Clash Stably

- The `/nslookup` is implemented by calling **gethostbyname_r(3)**
  - We run a customized DNS server and delay the response
    - This allows us to control the timing of the write!
    - Leak the stack pointer to bypass PIE & ASLR
    - Write the return address to control the PC

Completed All Tasks on 9/19

SONOS

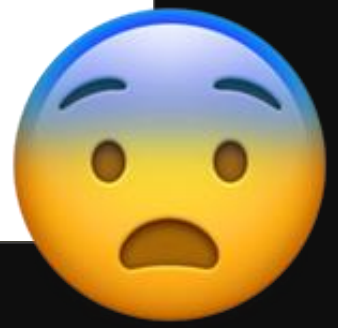
14.16

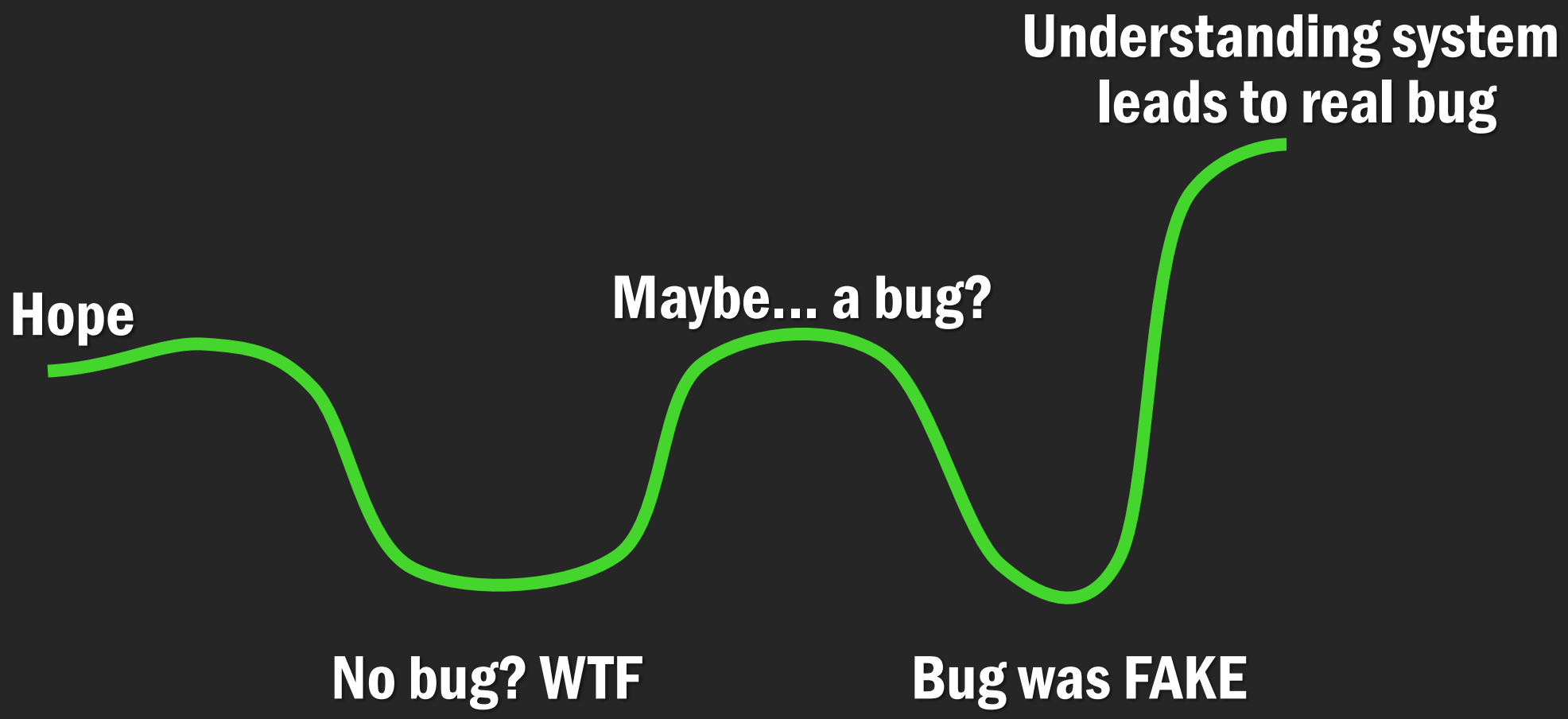
Release date: 9/20/2022

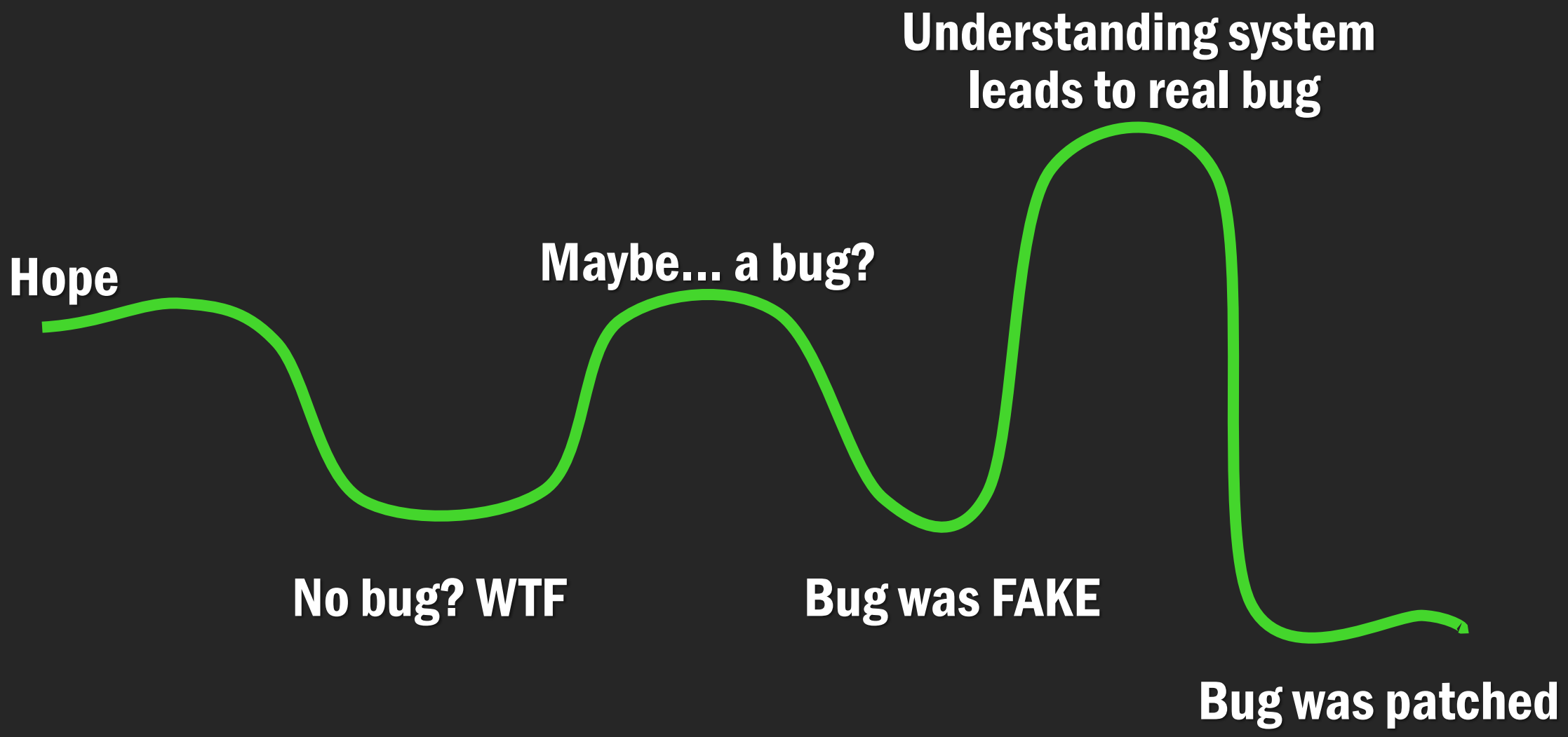
In this update:

- When connected to WiFi, Roam and Roam SL stereo pairs can now play stereo audio from Bluetooth sources. These stereo pairs will no longer separate when switching to Bluetooth mode.

- Bug fixes and performance enhancements.




Understanding system
leads to real bug
Hope
Maybe... a bug?
No bug? WTF
Bug was FAKE


Understanding system
leads to real bug
Hope
Maybe... a bug?
No bug? WTF
Bug was FAKE
Bug was patched

2022 - Our Third Year

- ~~Continue to explore our last year's good attack surface~~
- Have to discover new attack surfaces:
  - The product integration part with Open-Source sounds good!

Insecure Callbacks

- XML Parser - libexpat
  - Assume it's safe because it's used worldwide
  - But is its usage also?

```
void XML_SetElementHandler(
        XML_Parser p,
        XML_StartElementHandler start,  // user callback
        XML_EndElementHandler end       // user callback
);
```

```c
void start(void *userData, char *tag_name, char **attrs) {
    if (!strcmp(tag_name, "block")) {
        userData->block_index += 1;
        if (userData->block_index > 10)
            return;
        else
            userData->blocks[block_index] = (Block *)malloc(0x4070);
    } else if (!strcmp(tag_name, "param")) {
        block = userData->blocks[userData->block_index];
        strlcpy(block->names[userData->param_count], name, name_len);
        strlcpy(block->values[userData->param_count], val, val_len);
    }

}

void end(void *userData, char *tag_name) {
    // handle close tag…
}
```

Insecure Callbacks

```
<root>
    <block>
        <param AAA="BBB">FOO</param>
        <param CCC="DDD">BAR</param>
    </block>

    <block>
        <param EEE="FFF">BAZ</param>
    </block>
</root>
```

```
<root>
    <block>
        <block>
            <block>
                <block>
                    ...
                        <block>
                            <block>
                                <block>
                                    <param AAA="BBB">FOO</param>
                                    <param CCC="DDD">BAR</param>
                                </block>
                            </block>
                        </block>
                    ...
                </block>
            </block>
        </block>
    </block>
</root>
```

2022 - Our Third Year

- Bugs:
  1. ~~Arbitrary size alloca(3) leads to Stack Clash~~ (silent fixed)
     - Transform to Read/Write primitive by delaying the DNS response
  2. Insecure callback leads to OOB-Write
     - Payloads can't consist of any non UTF-8 characters due to the XML spec
     - Have to bypass PIE/ASLR/Stack-Cookie first

```
soapaction = get_header(request, "soapaction");
useragent  = get_header(request, "user-agent");
size = __snprintf_chk(buffer, 4096, 1, 4096,
        "POST %s HTTP/1.1\r\n"
        "CONNECTION: close\r\n"
        "HOST: %s:%d\r\n"
        "USER-AGENT: %s\r\n"
        "CONTENT-LENGTH: %zu\r\n"
        "CONTENT-TYPE: text/xml; charset=\"utf-8\"\r\n"
        "SOAPACTION: %s\r\n"
        "\r\n",
        path,
        host,
        port,
        useragent,
        body_size,
        soapaction);
str.data = buffer;
str.size = size;
send_request(&client, &str, 1, NULL, NULL, …)
```

2022 - Our Third Year

- Bugs:
  1. ~~Arbitrary size alloca(3) leads to Stack Clash~~ (silent fixed)
     - Transform to Read/Write primitive by delaying the DNS response
  2. Insecure callback leads to OOB-Write
     - Payloads can't consist of any non UTF-8 characters due to the XML spec
     - Have to bypass PIE/ASLR/Stack-Cookie first
  3. Unchecked return value of `__snprintf_chk` leads to info leak

Chained All Together Around 10/18

SONOS

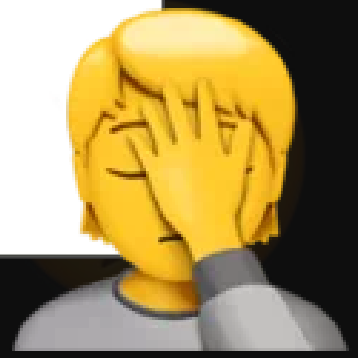
14.18

Release date: 10/18/2022

In this update:

- Bug fixes and performance enhancements, including a fix for an audio quality issue that reduced Sub output for Arc, Beam, or Ray when paired with a Sub or Sub Mini while Trueplay was enabled.

System requirements



2022 - Our Third Year

- Bugs:
  1. ~~Arbitrary size alloca(3) leads to Stack Clash~~ (silent fixed)
     - Transform to Read/Write primitive by delaying the DNS response
  2. ~~Insecure callback leads to OOB-Write~~ (silent fixed)
     - Payloads can't consist of any non UTF-8 characters due to the XML spec
     - Have to bypass PIE/ASLR/Stack-Cookie first
  3. Unchecked return of `__snprintf_chk` leads to info leak

2022 - Our Third Year

- Bugs:
  1. ~~Arbitrary size off~~ (silent fixed)
     ↳ Transform to Re the DNS response

The vulnerable still there, but the entry point moved precisely

3. Unchecked return of `__snprintf_chk` leads to info leak

RCrashdumpUploader?

- We have been aware since 2020
  1. POST to `crash-upload.ws.sonos.com` every 5 minutes
  2. Did the Sonos review the crashdump?
     - We don't know because things went too well last year
     - But that's the most reasonable explanation at that time

2022 - Our Third Year

- Accept it, we still have time (~1.5 months)... (Sigh)

OvEr OnE MoNtH LaTeR...

RUN OUT OF ALL IDEAS...

Unsuccessful Attempts

- Audio Codec:
  - Fuzz OSS libraries
  - Review the integration parts
- SOAP Implementations:
  - Review SOAP Parser
  - Review all SOAP actions
  - Review the deserialization process
  - Review URL clients (HTTP/CIFS/WS/SSDP/RTSP/...)
- Server/Service:
  - Web/WebSocket server
  - SNTP/mDNS/UPnP/... services
  - Communication between services
  - ...


2 WEEKS LEFT

Review Bugs We Have So Far

1. What's the root cause?
2. Why it happened during the development process?
3. How to discover the variant?

2021 - My First Fake Bug

```
size_t read_size = 1;
my_tsclient_read(ctx, &dlen, &read_size, timeval);

dlen = (unsigned __int8) dlen;
if (dlen) {
    char *buffer = alloca(dlen);
    my_tsclient_read(ctx, &buffer, &dlen, timeval);
}
```

After the Patch of October 2022

```
size_t read_size = 1;
my_tsclient_read(ctx, &dlen, &read_size, timeval);

dlen = (unsigned __int8) dlen;
if (dlen) {
    char buffer[184] = {0};
    my_tsclient_read(ctx, &buffer, &dlen, timeval);
}
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA


POWER

Summarize Our Third Year

1. Glad that I didn't give up until the last moment
2. Self-reflections:
   - Coverage should be more comprehensive, such as:
     - Lifetime-style bugs / my overlooked unsafe string operation bug last year
   - Didn't review the `libsmb2` because the name "SMB" scared me
     - All other teams are targeting it
   - Remember to cut off the Internet (set-up a good LAB environment)

Conclusion

1. Persistent is the key
   - Targeting the same targets for 3 years
   - Reversing day and night for 3 months

Conclusion

1. Persistent is the key
   - Targeting the same targets for 3 years
   - Reversing day and night for 3 months
2. On the right path is even more important
   - A good idea or good attack surface

Chose a Good Attack Surface

- Bugs:
  1. Size-checking error in firmware parser leads to Stack Overflow
  2. Integer Underflow in MP3-ID3v2 tag parser leads to Stack Overflow
  3. Arbitrary size `alloca(3)` in MP4-Box parser leads to Stack Clash
  4. Insecure `libexpat` callback leads to OOB-Write
  5. Unchecked return of `__snprintf_chk` leads to info leak
  6. Fixed buffer size in MPEG-TS parser leads to Stack Overflow

Chose a Good Attack Surface

- Bugs:
  1. Size-checking error in firmware parser leads to Stack Overflow
  2. Integer Underflow in MP3-ID3v2 tag parser leads to Stack Overflow
  3. Arbitrary size `alloca(3)` in MP4-Box parser leads to Stack Clash
  4. Insecure `libexpat` callback leads to OOB-Write
  5. Unchecked return of `__snprintf_chk` leads to info leak
  6. Fixed buffer size in MPEG-TS parser leads to Stack Overflow

Persist! Persist! Persist!

Until You found the right path

Celeste - "*Games for Impact*" of The Game Awards 2018

DEVCORE

Thanks!

orange_8361

orange@chroot.org

https://blog.orange.tw