# Threat Research

### New Zero-Day Exploit targeting Internet Explorer Versions 9 through 11 Identified in Targeted Attacks

April 27, 2014 | by Xiaobo Chen, Mike Scott, Dan Caselden

TIREEYE'

FireEye Research Labs identified a new Internet Explorer (IE) zero-day exploit used in targeted attacks. The vulnerability affects IE6 through IE11, but the attack is targeting IE9 through IE11. This zero-day bypasses both ASLR and DEP. Microsoft has assigned CVE-2014-1776 to the vulnerability and released security advisory to

Threat actors are actively using this exploit in an ongoing campaign which we have named "Operation Clandestine Fox." However, for many reasons, we will not provide campaign details. But we believe this is significant zero day as the vulnerable versions represent about a quarter of the total browser market. We commend applying a patch once available

According to NetMarket Share, the market share for the targeted versions of IE in 2013 were:

IF 9 13 9% IF 10 11 04% IF 11 1 3 2 %

Collectively, in 2013, the vulnerable versions of IE accounted for 26.25% of the browser market. The vulnerability, however, does appear in IE6 through IE11 though the exploit targets IE9 and higher

#### The Details

The exploit leverages a previously unknown use-after-free vulnerability, and uses a well-known Flash exploitation que to achieve arbitrary memory access and bypass Windows' ASLR and DEP protections.

#### Exploitation

#### • Preparing the heap

The exploit page loads a Flash SWF file to manipulate the heap layout with the common technique heap feng shui. It allocates Flash vector objects to spray memory and cover address 0x18184000. Next, it allocates a vector object that contains a flash.Media.Sound() object, which it later corrupts to pivot control to its ROP chain.

## · Arbitrary memory access

The SWF file calls back to Javascript in IE to trigger the IE bug and overwrite the length field of a Flash vector object in the heapspray. The SWF file loops through the heapspray to find the corrupted vector object, and uses it to again modify the length of another vector object. This other corrupted vector object is then used for subsequent memory accesses, which it then uses to bypass ASLR and DEP.

#### • Runtime ROP generation

With full memory control, the exploit will search for  $\it ZwProtectVirtual Memory$ , and a stack pivot (opcode 0x94 0xc3) from NTDLL. It also searches for  $\it SetThreadContext$  in kernel32, which is used to clear the debug register  $\it SetThreadContext$  in kernel32. This technique, may be an attempt to bypass protections that use hardware breakpoints, such as EMET's EAF mitigation

With the addresses of the aforementioned APIs and gadget, the SWF file constructs a ROP chain, and prepends it to its RC4 decrypted shellcode. It then replaces the vftable of a sound object with a fake one that points to the newly created ROP payload. When the sound object attempts to call into its vftable, it instead pivots control to the attacker's ROP chair

#### • ROP and Shellcode

The ROP payload basically tries to make memory at 0x18184000 executable, and to return to 0x1818411c to execute the shellcode.

0:008> dds eax

18184100 770b5f58 ntdll!ZwProtectVirtualMemory

18184104 18184110

1818410c 181840e8

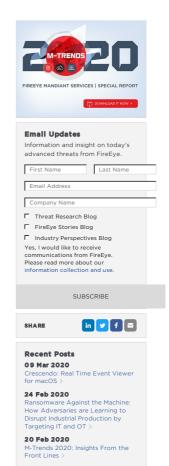
18184110 181840ec

18184114 00000040

18184118 181840e4

Inside the shellcode, it saves the current stack pointer to Ox18181800 to safely return to the caller

mov dword ptr ds:[18181800h],ebp



RSS FEED: STAY CONNECTED

crashes

18184123 b820609f06 mov eax,69F6020h

18184128 90 nop

18184129 90 nop

1818412a c700c0f22169 mov dword ptr [eax],offset Flash32\_11\_7\_700\_261!AdobeCPGetAPI+0x42ac00 (6921f2c0)

18184133 b800401818 mov eax,18184000h

18184138 90 nop

18184139 90 nop

1818413a c700fe030000 mov dword ptr [eax],3FEh ds:0023:18184000=3ffffff0

The shellcode also recovers the ESP register to make sure the stack range is in the current thread stack base/limit

18184140 8be5 mov esp,ebp

18184142 83ec2c sub esp,2Ch

18184145 90 nop

18184146 eb2c jmp 18184174

The shellcode calls SetThreadContext to clear the debug registers. It is possible that this is an attempt to bypass mitigations that use the debug registers.

18184174 57 push edi

18184175 81ece0050000 sub esp,5E0h

1818417b c7042410000100 mov dword ptr [esp],10010h

18184182 8d7c2404 lea edi,[esp+4]

18184186 b9dc050000 mov ecx,5DCh

1818418b 33c0 xor eax,eax

1818418d f3aa rep stos byte ptr es:[edi]

1818418f 54 push esp

18184190 6afe push OFFFFFFEh

18184192 b8b308b476 mov eax,offset kernel32!SetThreadContext (76b408b3)

18184197 ffd0 call eax

The shellcode calls URLDownloadToCacheFileA to download the next stage of the payload, disguised as an .

### Mitigation

Using EMET may break the exploit in your environment and prevent it from successfully controlling your computer. EMET versions 4.1 and 5.0 break (and/or detect) the exploit in our tests.

Enhanced Protected Mode in IE breaks the exploit in our tests. EPM was introduced in IE10.

 $\label{eq:Additionally, the attack will not work without Adobe Flash. Disabling the Flash plugin within IE will prevent the exploit from functioning.$ 

### **Threat Group History**

The APT group responsible for this exploit has been the first group to have access to a select number of browser-based 0-day exploits (e.g. IE, Firefox, and Flash) in the past. They are extremely proficient at lateral movement and are difficult to track, as they typically do not reuse command and control infrastructure. They have a number of backdoors including one known as Pirpi that we previously discussed here. CVE-2010-3962, then a 0-day exploit in Internet Explorer 6, 7, and 8 dropped the Pirpi payload discussed in this previous case.

As this is still an active investigation we are not releasing further indicators about the exploit at this time.

Acknowledgement: We thank Christopher Glyer, Matt Fowler, Josh Homan, Ned Moran, Nart Villeneuve and Yichong Lin for their support, research, and analysis on these findings.

< PREVIOUS POST

NEXT POST >

### News and Events

News and Events
Newsroom
Press Releases
Webinars
Events
Awards and Honors
Email Preferences

### Technical Support

Incident?
Report Security Issue
Contact Support
Customer Portal
Communities
Documentation Portal

FireEye Blogs
Threat Research
FireEye Stories
Industry Perspectives

Threat Map
View the Latest Threats

# Contact Us +1 877-347-3393

### Stay Connected

