APT GROUP UPS TARGETS US GOVERNMENT WITH HACKING TEAM FLASH EXPLOIT

POSTED BY: Bryan Lee and Robert Falcone on July 10, 2015 11:36 AM

FILED IN: Malware, Threat Prevention, Unit 42
TAGGED: ActionScript, Adobe Flash, AutoFocus, Hacking Team

On July 8, 2015, Unit 42 used theAutoFocus Threat Intelligence service to locate and investigate activity consistent with a spear-phishing attack targeting the US Government. The attack exploited an Adobe Flash vulnerability that stems from the zero-day vulnerabilities exposed from this month's Hacking Team data breach.

The spear-phishing attack used a link to a Flash exploit hosted on two subdomains of a legitimate website, perrydale[.]com; rpt.perrydale[.]com and report.perrydale[.]com. Both domains resolve to the same Ukraine-based IP 194.44.130.179.

There are no indications at this time that the actual website has been compromised, rather, this is more likely a case of DNS hijacking. The Flash exploits, specifically located at rpt.perrydale[.]com/en/show.swf and report.perrydale[.]com/ema/show.swf leverage one of the newly disclosed vulnerabilities from the Hacking Team data breach, CVE-2015-5119. Successful exploitation leads to the affected host retrieving a secondary payload, b.gif, also located at the same two subdomains as the Flash exploit.

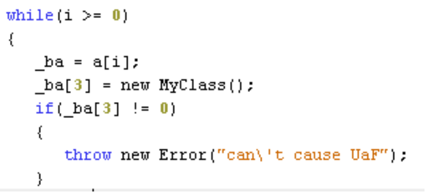
This attack shares similarities with a previous targeted attack, also using a Flash exploit, leveraging what was at the time a zero-day vulnerability in CVE-2015-3113. Analysis of both malicious Flash files indicates both these attacks are attributed to the APT group known as UPS or APT3.

ACTIONSCRIPT

The malicious Flash file named "show.gif" contains ActionScript that attempts to exploit a vulnerability and execute shellcode to ultimately install a payload. Show.swf is composed of the following ActionScript classes:

- MainClass.as
- MyClass.as
- MyClass1.as
- MyClass2.as
- MyUtils.as
- ShellWin32.as

Preliminary analysis of class names revealed overlap with one of the two Flash zero-day exploits disclosed following the Hacking Team breach. When comparing the classes above with those associated with Hacking Team's Flash zero-days, we found that MyClass.as, MyClass1.as, MyClass2.as, MyUtils.as and ShellWin32.as were shared within show.swf and Hacking Team's Flash exploit. In addition, there are several log messages as well as multiple function and variable names that exist in the ActionScript classes in both the UPS and the Hacking Team's Flash files. The most important overlap occurs in the "TryExpl" function within MyClass.as, where the same functions and variables are used to create the use-after-free condition caused during the exploitation of the CVE-2015-5119 vulnerability. Figure 1 shows the code in the "TryExpl" function that causes the use-after-free vulnerability found in both the UPS and the Hacking Team's Flash exploits. Also, the error message "can't cause UaF" is found in both exploits.



```
while(i >= 0)
{
   _ba = a[i];
   _ba[3] = new MyClass();
   if(_ba[3] != 0)
   {
      throw new Error("can\'t cause UaF");
   }
```

Figure 1. ActionScript Causing the Use-After-Free Vulnerability

While analyzing the MainClass portion of show.swf, we also observed shared functions with a previous attack attributed to UPS that was designed to exploit an earlier Flash zero-day, CVE-2015-3113. The CVE-2015-3113 ActionScript is publically available and can be obtained from the following link:

Home
Government
Partners
Unit 42 Threat Intelligence
Technical Documentation
Advanced Endpoint Protection

CATEGORIES & ARCHIVES

Select a Category

Select a Month

MORE →

RECENT POSTS

Cyber Threat Alliance Reflects on the Current State of Cybersecurity at the Churchill Club
posted by Chad Berndtson on October 30, 2015

Channel Scoop – October 30, 2015
posted by Lang Tibbils on October 30, 2015

People of Palo Alto Networks: Get in on the Ground Floor
posted by Justin Hall on October 29, 2015

CryptoWall 3, the Cyber Threat Alliance and the Future of Information Sharing
posted by Rick Howard on October 29, 2015

Understanding and Preventing Point of Sale Attacks
posted by Josh Grunzweig on October 28, 2015

MORE →

https://gist.github.com/mak/bd71962aae98ab0b0441

The shared function names, seen below, include several functions used for data type manipulation, logging, and decrypting the shellcode executed in the event of successful exploitation:

- decode
- hexToIntArray
- logMsg
- func_prepare
- hexToBin

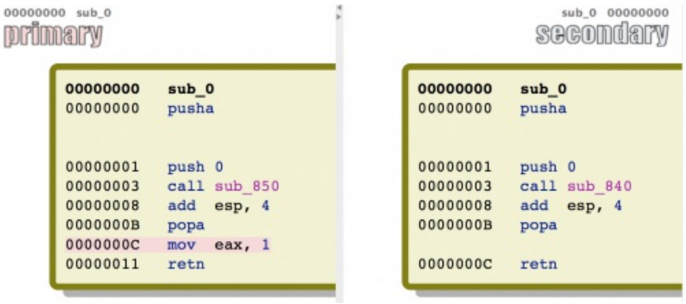
The most obvious overlap between the two ActionScripts involves the shared variable name “m_scKey”, which is a variable that stores the RC4 key that the ActionScript will use to decrypt the shellcode.

SHELLCODE

When the Flash vulnerability is successfully exploited, shellcode executes which then extracts and decrypts a payload embedded in an animated GIF image. During analysis, Unit 42 was unable to obtain the payload; the “b.gif” file received was not weaponized as it does not contain an encrypted payload. There are two likely reasons for this – UPS is known for both only serving malicious payloads within very limited windows of time during an attack, and even then only serving those payloads to victims that fit their desired profile.

The technique of extracting and decrypting a payload from within an animated GIF image was also used by UPS in the attacks exploiting CVE-2015-3113. Using Zynamic’s binDiff tool to compare, we discovered 99% similarity with 99% confidence between the 5119 shellcode and the 3113 shellcode. By manually comparing the code, we confirmed the high similarity and confidence rates as calculated by binDiff.

The technique of locating the payload embedded in the animated GIF is the same within both the 5119 and the 3113 shellcodes. Additionally, both shellcodes use the exact same algorithm and key values to decrypt the payload from ciphertext to cleartext, specifically using an XOR, subtraction and a second XOR instruction using key values 0x12, 0x11 and 0x85, respectively. In fact, we compared the two shellcodes side-by-side and found that there is only one instruction added to the 5119 shellcode as seen highlighted in red in the image below.



```
00000000  sub_0
00000000  pusha

00000001  push 0
00000003  call sub_850
00000008  add  esp, 4
0000000B  popa
0000000C  mov  eax, 1
00000011  retn
```

```
00000000  sub_0
00000000  pusha

00000001  push 0
00000003  call sub_840
00000008  add  esp, 4
0000000B  popa

0000000C  retn
```

CONCLUSION

These attacks highlight how sophisticated APT groups such as UPS can quickly leverage new vulnerabilities in their attacks. A patch is available for this vulnerability, but was only released on the same day of weaponization, which leaves very little time for any organization to patch effectively. Due to the highly targeted nature of this type of attack, traditional detection methods via known IOCs can be challenging. Deployment of automated, behavioral preventative measures such as Palo Alto Networks Traps can significantly reduce organizational risk to these types of attacks.

INDICATORS OF COMPROMISE

SHA256

a2fe113cc13acac2bb79a375f692b8ba5cc2fa880272adc7ab0d01f839e877ff

Domains

rpt.perrydale[.]com

report.perrydale[.]com

IPs

194.44.130.179

URLs

rpt.perrydale[.]com /en/show.swf

report.perrydale[.]com /ema/show.swf

rpt.perrydale[.]com /en/b.gif

report.perrydale[.]com /ema/b,gif

4 PINGBACKS & TRACKBACKS

July 15, 2015 4:40 PM
ste williams – 4 Lasting Impacts Of The Hacking Team Leaks

July 15, 2015 8:43 PM
4 Lasting Impacts Of The Hacking Team Leaks | TechDiem.com

July 21, 2015 6:24 PM
All Recent Unit 42 Threat Intelligence – Right at the Top of Your Inbox | Philip Hung Cao

July 28, 2015 3:53 AM
UPS: Observations on CVE-2015-3113, Prior Zero-Days and the Pirpi Payload | Philip Hung Cao

POST YOUR COMMENT

Name *

Email *

Website

Post Comment