New Flash Player 0-day (CVE-2014-0515) used in watering-hole attacks

Vyacheslav Zakorzhevsky  
Kaspersky Lab Expert  
Posted April 28, 15:25 GMT  
Tags: Adobe Flash, Targeted Attacks, Adobe, Zero-day vulnerabilities

In mid-April we detected two new SWF exploits. After some detailed analysis it was clear they didn't use any of the vulnerabilities that we already knew about. We sent the exploits off to Adobe and a few days later got confirmation that they did indeed use a 0-day vulnerability that was later labeled as CVE-2014-0515. The vulnerability is located in the Pixel Bender component, designed for video and image processing.

We received a sample of the first exploit on April 14, while a sample of the second came on April 16. The first exploit was initially recorded by KSN on April 9, when it was detected by a generic heuristic signature. There were numerous subsequent detections on April 14 and 16. In other words, we succeeded in detecting a previously unknown threat using heuristics.

According to KSN data, these exploits were stored as movie.swf and include.swf at an infected site. The only difference between the two pieces of malware is their shellcodes. It should be noted that the second exploit (include.swf) wasn't detected using the same heuristic signature as the first, because it contained a unique shellcode.

Each exploit comes as an unpacked flash video file. The Action Script code inside was neither obfuscated nor encrypted.

As is usually the case with this kind of exploit, the first stage is a heap spray - preparing the dynamic memory for exploitation of the vulnerability. The exploits are also designed to check the OS version. If Windows 8 is detected, a slightly modified byte-code of the Pixel Bender component is used.

A fragment of the vulnerable Pixel Bender code (the data in the red box is changed according to system version)
Next comes the actual exploitation of the vulnerability, namely modification of one of the indices in the table of methods/virtual functions.

Interestingly, both exploits have two shellcodes. The first is similar in both applications; it is quite short and prepares the memory for the successful functioning of the second shellcode.
A fragment of the first shellcode debugged in WinDBG

Firstly, the current memory is marked as read, write and execute with the API function VirtualProtect, and then additional memory is allocated using VirtualAlloc. The second shellcode is copied to this memory and control is transferred to it. The initialization of API functions and transfer of the control to the second shellcode appear in red boxes in the screenshot above.

The exploits' second shellcodes differ significantly.

The exploit that we detected first has a standard shellcode (movie.swf). It performs a search of system libraries in the memory, and then downloads and runs the payload. Unfortunately, the link turned out to be inactive at the time of our research.
Fragment of the movie.swo exploit's second shellcode responsible for the download and launch of the payload

In the other exploit - include.swo - the second shellcode was unusual. It receives the base DLL address for flash10p.ocx, searching it for specific fragments and interacts with the ciscompeaddin5x0 - Cisco MeetingPlace Express Add-In version 5x0. This add-in is used by web-conference participants to view documents and images from presenter's screen. It should be noted that the exploit will not work if the required versions of Adobe Flash Player ActiveX and Cisco MPE are not present on the system.
It appears that part of the information for the exploit include.swf is passed on from outside. According to KSN data, the referer to include.swf points to another SWF file: stream.swf. At the same time, the referer of the first exploit - movie.swf - points to index.php located in the same folder as the exploit (see below). We couldn't establish the exact payload of the exploit include.swf due to a lack of data relayed from the landing page and/or other exploits.

We are sure that all these tricks were used in order to carry out malicious activity against a very specific group of users without attracting the attention of security solutions. We believe that the Cisco add-in mentioned above may be used to download/implement the payload as well as to spy directly on the infected computer.

Both the exploits detected by us spread from a site located at http://jpic.gov.sy/. The site was launched back in 2011 by the Syrian Ministry of Justice and was designed as an online form for citizens to complain about law and order violations. We believe the attack was designed to target Syrian dissidents complaining about the government.

The site was hacked in September 2013, something the alleged hacker announced on his twitter account.

The link to these exploits is as follows: http://jpic.gov.sy/css/images/_css/*******.. When we entered the site, the installed malware payloads were already missing from the "_css" folder. We presume the criminals created a folder whose name doesn't look out of place on an administration resource, and where they loaded the exploits. The victims were probably redirected to the exploits using a frame or a script located at the site. To date, April 28, the number of detections by our products has exceeded 30. They were detected on the computers of seven unique users, all of them in Syria, which is not surprising considering the nature of the site. Interestingly, all the attacked users entered the website using various versions of Mozilla Firefox.

It's likely that the attack was carefully planned and that professionals of a pretty high caliber were behind it. The use of professionally written 0-day exploits that were used to infect a single resource testifies to this.

Moreover, while the first exploit is pretty standard and can infect practically any unprotected computer, the second exploit (include.swf) only functions properly on computers where Adobe Flash Player 10 ActiveX and Cisco MeetingPlace Express Add-In are installed. The Flash Player Pixel Bender component, which Adobe no longer supports, was used as the attack vector. The authors were counting on the developers not finding a vulnerability in that component and that the exploit would remain active for longer. All this suggests that the attackers were not targeting users en masse.
Registered trademarks and service marks are the property of their respective owners.