Internet Explorer CVE-2019–1367 In the wild Exploitation — prelude

blog.confiant.com/internet-explorer-cve-2019-1367-in-the-wild-exploitation-prelude-ef546f19cd30

taha aka lordx64 July 13, 2020

Photo by on

CVE-2019–1367 background and in-the-wild exploitations

There are some important aspects to know about CVE-2019–1367 before diving into the technical analysis including the intel around it and the series of events following the in-the-wild exploitations and Microsoft patches.

First of all, here is the bug class of this bug based on the Google P0 report:

JScript variable (represented as VAR structure) isn't properly tracked by garbage collector

Is it important to understand the bug class. For example Google P0 will work on what they call a Variant Analysis, to discover additional vulnerabilities from the same bug class. Example CVE-2019–1429 is a result of Variant analysis of CVE-2019–1367.

In fact, the first in the wild exploitation from this bug class was seen in December 2018, exploiting CVE-2018–8653, discovered by Google TAG Team. This bug was then documented by Mcafee here and by Tetrane here.

Google TAG Team discovered CVE-2019–1367 exploited in the wild by a threat actor. No details were given at the time of Microsoft advisory.

But in a recent blog Google TAG Team discussed that North Korea or individuals who worked on North Korea-related issues were the main targets of this in-the-wild exploitations but no more elements were given regarding the threat actors behind these attacks at the time of the reporting.

Who are these threat actors?

Based on OSINT, and the data following this discovery, it seems that there are two main threat actors known to date caught exploiting CVE-2019–1367:

> : A suspected Korean Peninsula APT actor, considered to be a skill-full, active, long-run (+10 years of existence) resourceful (state sponsored?) APT actor.

Magnitude Exploit KIT: An opportunistic Malvertiser, mostly targeting south Korea. Magnitude EK has been there since 2013, and known to drop very known ransomware families including: Locky, Cerber, Magniber, CryptoWall, GranCrab.. Known to rapidly integrate CVE's into their exploitation chains. **Magnitude** has been active for many years, below is an tweet from 2013 showing some of their oldest CVE integrations:

So what's CVE-2019–1367 impact?

CVE-2019–1367 enables Remote Code Execution (RCE) in the context of Internet explorer in all version from 8, 9, 10 and 11 due to a memory corruption in jscript.dll.

This means victims will ultimately be infected with a malware just by browsing to a web page aka 1-click exploit : Victims need to click at least one time into a link to get infected (different from 0-click exploits that requires no user interaction at all).

A scenario of 1-click exploit attack would be of watering hole type of attack, this is usually a technique of nation state APTs.

> Note: most recent iOS exploits exploited in the wild were found in watering hole attacks, targeting certain populations.

We also see 1-click exploit used in malvertising (malicious ads) mostly by cyber crime/FIN actors. In the scenario where 1-click exploits are integrated into a chain of redirects like we see everyday in malicious advertising attack, this 1-click exploits could have a 0-click effect, since no user interaction will be required before getting infected.

Even though 1-click exploits are less valuable than 0-click exploits (and less pricey), they can have a similar devastating effect if integrated at the right place.

Microsoft released a patch and encouraged users to disable jscript.dll (a Legacy dll replaced by jscript9.dll) that can still be called with IE-8 compatibility mode enabled. This is typically enabled via the following tag:

<meta http-equiv="X-UA-Compatible" content="IE=8"></meta>

To give more context regarding CVE-2019–1367 we draw a timeline of events, that we collected (based on OSINT)

Timeline of the events

: from Google TAG Team acknowledged by Microsoft. first time cited as linked to the in the wild exploitation of this bug:

: confirmed that was exploiting this vulnerability:

Based on the above tweet, Google Project Zero maintains a google doc referencing 0-day in-the-wild exploitation where they officially attributed CVE-2019–1367 to **DarkHotel APT.**

: Samples from exploits were uploaded to VirusTotal and flagged as by most of the security vendors:

: Google Project Zero did a Variant Analysis of which resulted in which a variant of the same bug class:

In their presentation they gave indication about the root cause of CVE-2019–1367 bug, which is as following:

> *Function arguments not being tracked by garbage collector during callback, in jscript.dll.*

So now we have a good lead on what type of object and function we need to look for this exploit analysis.

: published details about switching to another 0-day exploit, the attack was called "Double Star" as it involves this time a 0-day in Internet Explorer () and a 0-day in FireFox (CVE-2019–17026). Same day as Microsoft published the . No more additional details were given about the vulnerability.

Based on Google P0 tweet below, it seems that CVE-2020–0674 is patching a misfix of the earlier CVE-2019–1367 suggest that CVE-2020–0674 is a variant from the same bug class:

> : Kang Yang() of Qihoo 360 ATA found that the original proof of concept they sent to Microsoft for was uploaded to VirusTotal. ie, the proof of concept is now available to the security community!

After analysis of this “leak” we confirmed that CVE-2020–0674 is same bug class and is a variant of CVE-2019–1367 :

- : actors caught exploiting the vulnerability in the wild targeting south Korea, based on a report of (a south Korean Security Vendor)
- hile analyzing recent Exploit KIT attacks from different OSINT sources, we stumbled upon a recent attack targeting what seems to be Korean Users. The pcaps of a successful attack were generously provided by !

With **bluegas[.]website** hosting exploit code and **pophot[.]website** hosting ransomware:

After our analysis of the pcaps files, the attack turns out to be exploiting CVE-2019–1367! So we have now a proper source to analyze this exploit.

- : Google TAG Team confirmed in their report that , and are vulnerabilities inside jscript and are abusing the Enumerator object. They also give some indications about threat actor exploiting these vulnerabilities in the wild. Google stated that attack was targeting North Korea or individuals who worked on North Korea-related issues. This is inline with Kaspersky reporting above, and in line with Magnitude Exploit Kit Targeting Korean users.
- : attacks on japan infrastructures involving “Double Star” exploits, attack attributed again to

With **last.tax-lab[.]net** domain hosting exploit code and backdoor

> : Another related scripting engine vulnerability patched by Microsoft, caught exploited in the wild by Google. Microsoft assigned , We do not know if this bug is from the same bug class as and , as no details were provided by Google or by Microsoft:
>
> : from F-secure released a plain CVE-2020–0674 proof of concept exploit on for windows 7 x64 :
>
> : First write up of by F-secure labs, explaining the vulnerability and ways to exploit in x64 systems. This confirms our assumptions of being a variant of (or a missfix depends on which angle we see it).

: from Google TAG Team gave a on 0-day exploitation in the wild and shared and for the first time (AFAIK) important differences between CVE-2019–1367 and CVE-2020–0674 bugs. This confirmed directly some of our assumptions after analyzing the POC of CVE-2020–0674 that was uploaded to VT see (above timeline in 29th February 2020), and clearly helped us differentiating between the two bugs.

Elements we have so far:

The effort spent in the intel was worth it. At this point we are armed with the following:

- We have pcap files from a successful exploitation.
- Proof of concept code of CVE-2020–9674.
- We know the bug class, and the vulnerable code path for both CVE-2020–0674 and CVE-2019–1367.

CVE-2019–1367 analysis

For this analysis, we will start writing a POC for the CVE-2019–1367 vulnerability in part1.

In part2 we will discuss how this vulnerability was exploited by **Magnitude Exploit Kit** and **DarkHotel APT** and shed some lights on the differences.

Finally part3 is the shellcode analysis resulting from a successful exploitation of this bug in the wild.

Below is the table of content broken-down into different parts for readability

Table of Contents:

Tools used of the analysis:

- IDA Pro v7.5
- WinDBG: TTD feature, javascript engine
- radare2 : r2pipe
- Python3
- Microsoft Debug symbols.

Happy reading !