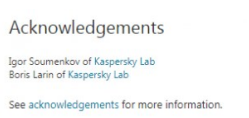
RESEARCH

Zero-day in Windows Kernel Transaction Manager (CVE-2018-8611)

By Boris Larin, Vladislav Stolyarov, Anton Ivanov on December 12, 2018. 8:00 am

Executive summary

In October 2018, our AEP (Automatic Exploit Prevention) systems detected an attempt to exploit a vulnerability in the Microsoft Windows operating system. Further analysis led us to uncover a zero-day vulnerability in ntoskrnl.exe. We reported it to Microsoft on October 29, 2018. The company confirmed the vulnerability and assigned it CVE-2018-8611. Microsoft just released a patch, part of its December update, crediting Kaspersky Lab researchers **Boris Larin** (Oct0xor) and **Igor Soumenkov** (Zigosha) with the discovery.



This is the third consecutive exploited Local Privilege Escalation vulnerability in Windows we discovered this autumn using our technologies. Unlike the previously reported vulnerabilities in win32k.sys (CVE-2018-8589 and CVE-2018-8453), CVE-2018-8611 is an especially dangerous threat – a vulnerability in the Kernel Transaction Manager driver. It can also be used to escape the sandbox in modern web browsers, including Chrome and Edge, since syscall filtering mitigations do not apply to ntoskrnl.exe system calls.

Just like with CVE-2018-8589, we believe this exploit is used by several threat actors including, but possibly not limited to, FruityArmor and SandCat. While FruityArmor is known to have used zero-days before, SandCat is a new APT we discovered only recently. In addition to this zero-day and CHAINSHOT, SandCat also uses the FinFisher / FinSpy framework.

Kaspersky Lab products detected this exploit proactively through the following technologies:

1. Behavioral detection engine and Automatic Exploit Prevention for endpoint products
2. Advanced Sandboxing and Anti Malware engine for Kaspersky Anti Targeted Attack Platform (KATA)

Kaspersky Lab verdicts for the artifacts used in this and related attacks are:

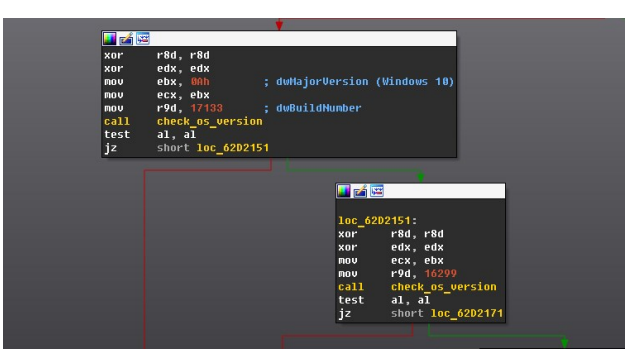
- HEUR:Exploit.Win32.Generic
- HEUR:Trojan.Win32.Generic
- PDM:Exploit.Win32.Generic

Brief details – CVE-2018-8611 vulnerability

CVE-2018-8611 is a race condition that is present in the Kernel Transaction Manager due to improper processing of transacted file operations in kernel mode.

This vulnerability successfully bypasses modern process mitigation policies, such as Win32k System call Filtering that is used, among others, in the Microsoft Edge Sandbox and the Win32k Lockdown Policy employed in the Google Chrome Sandbox. Combined with a compromised renderer process, for example, this vulnerability can lead to a full Remote Command Execution exploit chain in the latest state-of-the-art web-browsers.

We have found multiple builds of exploit for this vulnerability. The latest build includes changes to reflect the latest versions of the Windows OS.

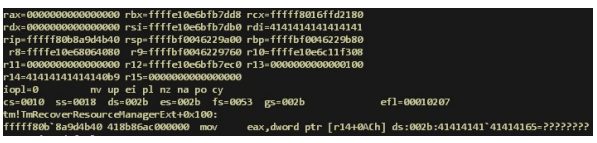


A check for the latest build at the time of discovery: Windows 10 Redstone 4 Build 17133

Similarly to CHAINSHOT, this exploit heavily relies on the use of C++ exception handling mechanisms with custom error codes.

To abuse this vulnerability exploit first creates a named pipe and opens it for read and write. Then it creates a pair of new transaction manager objects, resource manager objects, transaction objects and creates a big number of enlistment objects for what we will call "Transaction #2". Enlistment is a special object that is used for association between a transaction and a resource manager. When the transaction state changes associated resource manager is notified by the KTM. After that it creates one more enlistment object only now it does so for "Transaction #1" and commits all the changes made during this transaction.

After all the initial preparations have been made exploit proceeds to the second part of vulnerability trigger. It creates multiple threads and binds them to a single CPU core. One of created threads calls NtQueryInformationResourceManager in a loop, while second thread tries to execute NtRecoverResourceManager once. But the vulnerability itself is triggered in the third thread. This thread uses a trick of execution NtQueryInformationThread to obtain information on the latest executed syscall for the second thread. Successful execution of NtRecoverResourceManager will mean that race condition has occurred and further execution of WriteFile on previously created named pipe will lead to memory corruption.



Proof of concept: execution of WriteFile with buffer set to 0x41

As always, we provided Microsoft with a proof of concept for this vulnerability, along with source code. And it was later shared through Microsoft Active Protections Program (**MAPP**).

More information about SandCat, FruityArmor and CVE-2018-8611 is available to customers of Kaspersky Intelligence Reports. Contact: intelreports@kaspersky.com

APT MICROSOFT WINDOWS PROOF-OF-CONCEPT TARGETED ATTACKS ZERO-DAY VULNERABILITIES

Share post on:

Related Posts

THERE ARE 3 COMMENTS

Bob
Posted on December 13, 2018. 1:29 am
Care to share your IDA color scheme with the public? It looks really nice.

REPLY

Boris Larin
Posted on December 18, 2018. 6:26 pm
Hi Bob.
Basically its a modified version of https://github.com/eugeii/ida-consonance + my plugin https://github.com/oct0xor/highlight2 to highlight call instructions.

REPLY

Behnam A.Shamshirsaz
Posted on December 13, 2018. 1:40 pm
And what will be happen after crash ? (memory corruption) , is it possible to execute a malware code inside crashed machine or it is just a crash making mechanism ?

REPLY

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Enter your comment here

Name *

Email *

Save my name, email, and website in this browser for the next time I comment.

SUBMIT

IN THE SAME CATEGORY


Assessing the impact of protection from web miners


Agent 1433: remote attack on Microsoft SQL Server


How to steal a million (of your data)


On the IoT road: perks, benefits and security of moving smartly


How we hacked our colleague's smart home