The case of the longjmp from nowhere trying to open a registry key

devblogs.microsoft.com/oldnewthing/20170609-00

June 9, 2017

Raymond Chen

The crash telemetry team brought our attention to a bug a few weeks before the Creators Update was supposed to be released, and based on the high hit count of 3 million crashes in the past 30 days, the bug was marked “Blocking engineering sign-off”.

Life is always exciting when you get a “Blocking engineering sign-off” bug.

Here’s the relevant excerpt from the crashing stack. Let’s see what we can make of it:

```
ntdll!RtlFailFast2
ntdll!RtlGuardCheckLongJumpTarget+0x72f9f
ntdll!RtlGuardRestoreContext+0x360
ntdll!RtlUnwindEx+0x767
0x00007ff7`26040a5a
0x00007ff7`25fcb0c6
0x00007ff7`25fc9bf8
0x00007ff7`25fc9d23
0x00007ff7`25fc9f27
0x00007ff7`25fe1d17
0x00007ff7`25fdfd7f
0x00007ff7`25fca5f1
0x00007ff7`25fca645
0x00007ff7`25fcae25
0x00007ff7`25fca870
0x00007ff7`25fc27be
0x00007ff7`25fc6aaf
0x00007ff7`25fcab3d
0x00007ff7`25fe11ef
0x00007ff7`25fca5f1
0x00007ff7`25fca645
0x00007ff7`25fcae25
0x00007ff7`25fca870
0x00007ff7`25fc27be
0x00007ff7`25fe949c
0x00007ff7`25fe93e6
0x00007ff7`25fe6709
0x00007ff7`25fe6200
ntdll!KiUserApcDispatch+0x2e
ntdll!ZwOpenKeyEx+0x14
KERNELBASE!Wow64pNtOpenKeyInternal+0x16
KERNELBASE!Wow64NtOpenKey+0x7c
KERNELBASE!LocalBaseRegOpenKey+0x1bc
KERNELBASE!RegOpenKeyExInternalW+0x13b
KERNELBASE!RegOpenKeyExW+0x19
windows_storage!SHGetMachineGUID+0x83
...
```

Working upward, the storage system is trying to read the machine GUID, so it's opening the `HKEY_ LOCAL_ MACHINE\ Software\ Microsoft\ Cryptography` registry key. That `RegOpenKeyExW` call eventually reached the kernel at `ZwOpenKeyEx`, and then somehow a user-mode asynchronous procedure call (APC) got dispatched back into the user-mode thread. That user APC executed a lot of code that got injected into the process, not associated with a DLL. It reached a point where it encountered an exception, and the operating system is trying to unwind the exception but something goes wrong with the unwind when it wants to check a long jump target: The jump target is not valid, so the process crashes with a fail-fast exception: Incorrect exception unwind information is not a recoverable error, and it may indicate the presence of malware.

There are multiple levels of mystery here. The first level of mystery is this chunk of code not associated with a DLL. How did it get into our process? This particular process was `RuntimeBroker.exe`, which isn't as promiscuous as `explorer.exe` with respect to shell extensions and other third-party extension points, nor is it a common target for code injection.

Second, why is opening a registry key dispatching user-mode APCs? This is not called out in the documentation, and it is not something expected in general. Dispatching user-model APCs is not something you do just for fun. It creates a situation where code is running inside the context of unrelated code. If a critical section was held at the time `RegOpenKeyExW` was called, that critical sction is still being held when the APC is run, and you are now in danger of creating a deadlock. This is why functions which processes APCs usually make you opt into APCs explicitly: `SleepEx`, `WaitForSingleObjectEx`, and `MsgWaitForMultiple-ObjectsEx` don't process APCs unless you say you want them to, and the non-`Ex` versions never process APCs.

The third mystery is why the injected code is performing a `longjmp`. The exception being propagated is `0x80000026` which is `STATUS_ LONGJUMP`: "A long jump has been executed."

The mystery code is trying to perform a `longjmp`. Thats right, a `longjmp`. Apparently we are still running code written in 1970.

We contacted the registry team for assistance, and they recognized this issue. They suspect that some third-party registry filter driver (perhaps a game's anti-cheat software) is monitoring attempts to access any registry keys under `HKEY_ LOCAL_ MACHINE\ Software\ Microsoft\ Cryptography` and scheduling user-mode APCs as part of its processing. That APC then runs into a situation where it decides to try to `longjmp` out of its normal processing, but the `longjmp` buffer is either corrupted, or the long jump target has not been registered as a control flow guard jump target, so the exception dispatching code says, "No way, I'm not dispatching this exception."

The registry team noted that the vast majority of the crashes are coming from machines running the Anniversary Update, not the Creators Update, so this likely not a case of the Creators Update making a change that exacerbated a pre-existing problem, and the "Blocking engineering sign-off" marking should be removed. Furthermore, even though there were three million hits over the past 30 days, the crashes were not uniformly-distributed. Rather, the issue spiked and then died down within a week. This suggests that the third party recognized the issue and put out their own fix.

Raymond Chen

Follow