OpenSSL* SHA Crash Bug Requires Application Update

ID 673163
Updated 11/1/2019
Version Latest
Public

OpenSSL* 1.0.2 beta (Jun 2014) to OpenSSL 1.0.2k (Jan 2017) contain bugs that either cause a crash or bad SHA (Secure Hash Algorithm) values on processors with the SHA extensions, such as the recently released 10th Generation processor. Both bugs were fixed years ago; however, any application that uses the old version directly, or as one of its dependencies, will fail. Unreal Engine* version 4.13 (Sept 2016) to version 4.21 (Dec 2018) contains the old version of OpenSSL, so any game built using those versions is *possibly* affected.

Short-Term Workaround

This is most useful for testing or identification of the bug. If the ISV (independent software vendor) controls the process launch 100 percent through a script or launcher, it could then be used as a fix.

OpenSSL provides an environment variable control for enabling features, including one that modifies the Intel® CPU identification, OPENSSL_ia32cap:

```
set OPENSSL_ia32cap=:~0x20000000
```

This disables the OpenSSL code check for SHA extensions and runs a different code path that does not contain the crashing bug.

Notes

- Environment variables are per process by default. Some applications or game launchers spawn a new process, so the variable may need to be set globally using setx or the control panel.
- Some applications or game launchers have already-running processes, so a reboot may be needed for the process to see the environment variable.
- In the case of one game, the anti-cheat/DRM code was clearing and/or ignoring environment variables, disabling this workaround.

Fixes

OpenSSL is most commonly statically linked, therefore the application itself needs to be modified by the ISV. The actions depend on if you are using Unreal Engine or not.

Unreal Engine Applications

If the application uses Unreal Engine it may be affected. OpenSSL is used for HTTP transactions and chat via XMPP. We expect multiplayer games to be most likely affected.

Depending on your usage of Unreal Intel recommends you take one of the following steps to resolve the issue:

- Upgrade the title to Unreal Engine 4.22 or later.
- If you are building from source, you can use the patch that was included in 4.22, or copy /Engine/Source/ThirdParty/OpenSSL into your own source. You must have access to the unreal source to view this. If you are building from source, you can also consider the 'Other applications' recommendations below.

Other Applications

If the application uses OpenSSL natively, Intel recommends you take one of the following steps to resolve the issue:

- Upgrade to OpenSSL version 1.1.1. Version 1.1.1 is not a drop-in replacement for version 1.0.2, so may require more extensive development and test but will bring you up to date on all OpenSSL updates.
- Upgrade to the latest version 1.0.2. This should be a drop-in replacement and will also include many other bugs and fixes.
- Upgrade to OpenSSL version 1.0.2L. This is the first version that contains both fixes. While this will fix the issues described here there may be other security issues still remaining.
- Implement these two fixes manually:
  - Fix crash in SHAEXT code on Windows*
  - Work around problem with hex constants in MASM (Microsoft* Macro Assembler). This is the lowest impact fix, but it does not bring OpenSSL up to date – there may be other security issues still remaining.

Notes

Version 1.0.2 will only be supported until December 2019 – choosing options that leave your code on 1.0.2 could leave you with other security issues.

Consider use of OpenSSL unit tests:

- We recommend teams run the OpenSSL unit tests as part of their build process to identify issues like these at build time.
- The source for the unit tests is on GitHub*.
- ISVs need to incorporate these tests into their build process.

The following report may help with identification of this issue: IMPORTANT BUG: Apollo Lake compatablity issues.

If you are an end user who has discovered and confirmed this bug, please contact the application developer directly and refer them to this article. If the developer is out of business or unresponsive, please contact an Intel representative so that we can investigate the issue.