TABLE OF CONTENTS

Third Party Security Advisories

DRAFT FOR REVIEW

07/20/2022 07:44:44

This document will list briefings on each third party security issue found and give a description, a timeline on updating component, an acknowledgment that the solution is included in tagged release.

Acknowledgements

Process

(short form)

1. Security Bugs reported through: National Vulnerability Database
2. The issue is evaluated for EDK2 exposure
3. Determine Timeline for updating to Master
4. Update third party component list fixed tag

Revision History

Revision	Revision History	Date
.001.0	Initial release. Logs 1 - 9	Jul 20, 2022

1. CVE-2021-3449 - OPENSSL

Published: 03/25/2021

Description:

An OpenSSL TLS server may crash if sent a maliciously crafted renegotiation ClientHello message from a client. If a TLSv1.2 renegotiation ClientHello omits the signature_algorithms extension (where it was present in the initial ClientHello), but includes a signature_algorithms_cert extension then a NULL pointer dereference will result, leading to a crash and a denial of service attack. A server is only vulnerable if it has TLSv1.2 and renegotiation enabled (which is the default configuration). OpenSSL TLS clients are not impacted by this issue. All OpenSSL 1.1.1 versions are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1k. OpenSSL 1.0.2 is not impacted by this issue. Fixed in OpenSSL 1.1.1k (Affected 1.1.1-1.1.1j).

Recommendation:

Edk2 TLS supports client mode only. This issue only exists in server mode.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

2. CVE-2021-3450 - OPENSSL

Published: 03/25/2021

Description:

The X509_V_FLAG_X509_STRICT flag enables additional security checks of the certificates present in a certificate chain. It is not set by default. Starting from OpenSSL version 1.1.1h a check to disallow certificates in the chain that have explicitly encoded elliptic curve parameters was added as an additional strict check. An error in the implementation of this check meant that the result of a previous check to confirm that certificates in the chain are valid CA certificates was overwritten. This effectively bypasses the check that non-CA certificates must not be able to issue other certificates. If a "purpose" has been configured then there is a subsequent opportunity for checks that the certificate is a valid CA. All of the named "purpose" values implemented in libcrypto perform this check. Therefore, where a purpose is set the certificate chain will still be rejected even when the strict flag has been used. A purpose is set by default in libssl client and server certificate verification routines, but it can be overridden or removed by an application. In order to be affected, an application must explicitly set the X509_V_FLAG_X509_STRICT verification flag and either not set a purpose for the certificate verification or, in the case of TLS client or server applications, override the default purpose. OpenSSL versions 1.1.1h and newer are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1k. OpenSSL 1.0.2 is not impacted by this issue. Fixed in OpenSSL 1.1.1k (Affected 1.1.1h-1.1.1j).

Recommendation:

X509_V_FLAG_X509_STRICT is never set by edk2.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

3. CVE-2021-3711 - OPENSSL

Published: 08/24/2021

Description:

In order to decrypt SM2 encrypted data an application is expected to call the API function EVP_PKEY_decrypt(). Typically an application will call this function twice. The first time, on entry, the "out" parameter can be NULL and, on exit, the "outlen" parameter is populated with the buffer size required to hold the decrypted plaintext. The application can then allocate a sufficiently sized buffer and call EVP_PKEY_decrypt() again, but this time passing a non-NULL value for the "out" parameter. A bug in the implementation of the SM2 decryption code means that the calculation of the buffer size required to hold the plaintext returned by the first call to EVP_PKEY_decrypt() can be smaller than the actual size required by the second call. This can lead to a buffer overflow when EVP_PKEY_decrypt() is called by the application a second time with a buffer that is too small. A malicious attacker who is able present SM2 content for decryption to an application could cause attacker chosen data to overflow the buffer by up to a maximum of 62 bytes altering the contents of other data held after the buffer, possibly changing application behaviour or causing the application to crash. The location of the buffer is application dependent but is typically heap allocated. Fixed in OpenSSL 1.1.1l (Affected 1.1.1-1.1.1k).

Recommendation:

The SM2 algorithm is not supported by EDK2.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

4. CVE-2021-3712 - OPENSSL

Published: 08/24/2021

Description:

ASN.1 strings are represented internally within OpenSSL as an ASN1_STRING structure which contains a buffer holding the string data and a field holding the buffer length. This contrasts with normal C strings which are repesented as a buffer for the string data which is terminated with a NUL (0) byte. Although not a strict requirement, ASN.1 strings that are parsed using OpenSSL's own "d2i" functions (and other similar parsing functions) as well as any string whose value has been set with the ASN1_STRING_set() function will additionally NUL terminate the byte array in the ASN1_STRING structure. However, it is possible for applications to directly construct valid ASN1_STRING structures which do not NUL terminate the byte array by directly setting the "data" and "length" fields in the ASN1_STRING array. This can also happen by using the ASN1_STRING_set0() function. Numerous OpenSSL functions that print ASN.1 data have been found to assume that the ASN1_STRING byte array will be NUL terminated, even though this is not guaranteed for strings that have been directly constructed. Where an application requests an ASN.1 structure to be printed, and where that ASN.1 structure contains ASN1_STRINGs that have been directly constructed by the application without NUL terminating the "data" field, then a read buffer overrun can occur. The same thing can also occur during name constraints processing of certificates (for example if a certificate has been directly constructed by the application instead of loading it via the OpenSSL parsing functions, and the certificate contains non NUL terminated ASN1_STRING structures). It can also occur in the X509_get1_email(), X509_REQ_get1_email() and X509_get1_ocsp() functions. If a malicious actor can cause an application to directly construct an ASN1_STRING and then process it through one of the affected OpenSSL functions then this issue could be hit. This might result in a crash (causing a Denial of Service attack). It could also result in the disclosure of private memory contents (such as private keys, or sensitive plaintext). Fixed in OpenSSL 1.1.1l (Affected 1.1.1-1.1.1k). Fixed in OpenSSL 1.0.2za (Affected 1.0.2-1.0.2y).

Recommendation:

Analysis has confirmed that the relevant OpenSSL string operations have been used correctly and safely in EDK2.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

5. CVE-2021-4160 - OPENSSL

Published: 01/28/2022

Description:

There is a carry propagation bug in the MIPS32 and MIPS64 squaring procedure. Many EC algorithms are affected, including some of the TLS 1.3 default curves. Impact was not analyzed in detail, because the pre-requisites for attack are considered unlikely and include reusing private keys. Analysis suggests that attacks against RSA and DSA as a result of this defect would be very difficult to perform and are not believed likely. Attacks against DH are considered just feasible (although very difficult) because most of the work necessary to deduce information about a private key may be performed offline. The amount of resources required for such an attack would be significant. However, for an attack on TLS to be meaningful, the server would have to share the DH private key among multiple clients, which is no longer an option since CVE-2016-0701. This issue affects OpenSSL versions 1.0.2, 1.1.1 and 3.0.0. It was addressed in the releases of 1.1.1m and 3.0.1 on the 15th of December 2021. For the 1.0.2 release it is addressed in git commit 6fc1aaaf3 that is available to premium support customers only. It will be made available in 1.0.2zc when it is released. The issue only affects OpenSSL on MIPS platforms. Fixed in OpenSSL 3.0.1 (Affected 3.0.0). Fixed in OpenSSL 1.1.1m (Affected 1.1.1-1.1.1l). Fixed in OpenSSL 1.0.2zc-dev (Affected 1.0.2-1.0.2zb).

Recommendation:

The issue only affects MIPS platforms, and we don’t have MIPS in EDK2.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

6. CVE-2022-0778 - OPENSSL

7. CVE-2022-1292 - OPENSSL

Published: 05/03/2022

Description:

The c_rehash script does not properly sanitise shell metacharacters to prevent command injection. This script is distributed by some operating systems in a manner where it is automatically executed. On such operating systems, an attacker could execute arbitrary commands with the privileges of the script. Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash command line tool. Fixed in OpenSSL 3.0.3 (Affected 3.0.0,3.0.1,3.0.2). Fixed in OpenSSL 1.1.1o (Affected 1.1.1-1.1.1n). Fixed in OpenSSL 1.0.2ze (Affected 1.0.2-1.0.2zd).

Recommendation:

EDK2 never calls the c_rehash.in script in normal build process.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

8. CVE-2022-2068 - OPENSSL

Published: 06/21/2022

Description:

In addition to the c_rehash shell command injection identified in CVE-2022-1292, further circumstances where the c_rehash script does not properly sanitise shell metacharacters to prevent command injection were found by code review. When the CVE-2022-1292 was fixed it was not discovered that there are other places in the script where the file names of certificates being hashed were possibly passed to a command executed through the shell. This script is distributed by some operating systems in a manner where it is automatically executed. On such operating systems, an attacker could execute arbitrary commands with the privileges of the script. Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash command line tool. Fixed in OpenSSL 3.0.4 (Affected 3.0.0,3.0.1,3.0.2,3.0.3). Fixed in OpenSSL 1.1.1p (Affected 1.1.1-1.1.1o). Fixed in OpenSSL 1.0.2zf (Affected 1.0.2-1.0.2ze).

Recommendation:

EDK2 never calls the c_rehash.in script in normal build process.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag

9. CVE-2022-2097 - OPENSSL

Published: 07/05/2022

Description:

AES OCB mode for 32-bit x86 platforms using the AES-NI assembly optimised implementation will not encrypt the entirety of the data under some circumstances. This could reveal sixteen bytes of data that was preexisting in the memory that wasn't written. In the special case of "in place" encryption, sixteen bytes of the plaintext would be revealed. Since OpenSSL does not support OCB based cipher suites for TLS and DTLS, they are both unaffected. Fixed in OpenSSL 3.0.5 (Affected 3.0.0-3.0.4). Fixed in OpenSSL 1.1.1q (Affected 1.1.1-1.1.1p).

Recommendation:

EDK2 does not enable AES OCB mode or use the 32-bit AES-NI assembly optimizations.

Until further notice, the following versions of OpenSSL are appropriate to use within the EDK2 CryptoPkg:

- OpenSSL 1.1.1j, updated in the edk2-stable202105 stable tag
- OpenSSL 1.1.1n, updated in the edk2-stable202205 stable tag