

Multiple OpenSSL vulnerabilities in BVMS

BOSCH-SA-090577-BT

Advisory Information

Advisory ID: BOSCH-SA-090577-BT

CVE Numbers and CVSS v3.1 Scores: This information has been omitted for better readability. Please refer to section "[Vulnerability Details](#)" for the complete CVE list.

Published: 06 Mar 2024

Last Updated: 06 Mar 2024

Summary

BVMS is using a Device Adapter service for communication with Tattile cameras which is also active when no Tattile cameras are added in the BVMS installation. This service uses an OpenSSL library, which has multiple vulnerabilities as published by OpenSSL. When successfully exploited, these vulnerabilities could lead to command injection or denial of service.

Affected Products

Bosch BVMS

Version(s): 10.1 - 12.0.1 (including)

Bosch BVMS Viewer

Version(s): 10.1 - 12.0.1 (including)

Bosch DIVAR IP 7000 R2

Version(s): 10.1 - 12.0.1 (including)

Bosch DIVAR IP all-in-one 5000

Version(s): 10.1 - 12.0.1 (including)

Bosch DIVAR IP all-in-one 7000
Version(s): 10.1 - 12.0.1 (including)
Bosch DIVAR IP all-in-one 7000 R3
Version(s): 10.1 - 12.0.1 (including)
Bosch DIVAR IP all-in-one 4000
Version(s): 10.1 - 12.0.1 (including)
Bosch DIVAR IP all-in-one 6000
Version(s): 10.1 - 12.0.1 (including)

Solution and Mitigations

Software Updates

The recommended approach is to update the software to a fixed version as soon as possible. Please check the Appendix for a list of updated versions for each affected product.

Secure Network Resources

It is advised to protect network access to Tattile devices and to BVMS Central Server. Network administrators should implement the following recommendations in conjunction with laws, regulations, and industry best practices:

Segment and segregate networks

Harden BVMS Central Server by turning off Bosch VMS Device Adapter service if no Tattile cameras are in use:

In the Search bar, search for "Services" and select "Services"

In the "Services" window select the "Bosch VMS Device Adapter", disable and stop the service

Monitor the network and review logs

Validate hardware integrity

Vulnerability Details

CVE-2020-1971

CVE description: The X.509 GeneralName type is a generic type for representing different types of names. One of those name types is known as EDIPartyName. OpenSSL provides a function GENERAL_NAME_cmp which compares different instances of a GENERAL_NAME to see if they are equal or not. This function behaves incorrectly when both GENERAL_NAMES contain an EDIPARTYNAME. A NULL pointer dereference and a crash may occur leading to a possible denial of service attack. OpenSSL itself uses the GENERAL_NAME_cmp function for two purposes: 1) Comparing CRL distribution point names between an available CRL and a CRL distribution point embedded in an X509 certificate 2) When verifying that a timestamp response token signer matches the timestamp authority name (exposed via the API functions TS_RESP_verify_response and TS_RESP_verify_token) If an attacker can control both items being compared then that attacker could trigger a crash. For example if the attacker can trick a client or server into checking a malicious certificate against a malicious CRL then this may occur. Note that some applications automatically download CRLs based on a URL embedded in a certificate. This checking happens prior to the signatures on the certificate and CRL being verified. OpenSSL's s_server, s_client and verify tools have support for the "-crl_download" option which implements automatic CRL downloading and this attack has been demonstrated to work against those tools. Note that an unrelated bug means that affected versions of OpenSSL cannot parse or construct correct encodings of EDIPARTYNAME. However it is possible to construct a malformed EDIPARTYNAME that OpenSSL's parser will accept and hence trigger this attack. All OpenSSL 1.1.1 and 1.0.2 versions are affected by this issue. Other OpenSSL releases are out of support and have not been checked. Fixed in OpenSSL 1.1.1i (Affected 1.1.1-1.1.1h). Fixed in OpenSSL 1.0.2x (Affected 1.0.2-1.0.2w).

Problem Type:

[CWE-476](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 5.9 (Medium)

CVE-2021-23840

CVE description: Calls to EVP_CipherUpdate, EVP_EncryptUpdate and

EVP_DecryptUpdate may overflow the output length argument in some cases where the input length is close to the maximum permissible length for an integer on the platform. In such cases the return value from the function call will be 1 (indicating success), but the output length value will be negative. This could cause applications to behave incorrectly or crash. OpenSSL versions 1.1.1i and below are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1j. OpenSSL versions 1.0.2x and below are affected by this issue. However OpenSSL 1.0.2 is out of support and no longer receiving public updates. Premium support customers of OpenSSL 1.0.2 should upgrade to 1.0.2y. Other users should upgrade to 1.1.1j. Fixed in OpenSSL 1.1.1j (Affected 1.1.1-1.1.1i). Fixed in OpenSSL 1.0.2y (Affected 1.0.2-1.0.2x).

Problem Type:

[CWE-190](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 7.5 (High)

CVE-2021-23841

CVE description: The OpenSSL public API function X509_issuer_and_serial_hash() attempts to create a unique hash value based on the issuer and serial number data contained within an X509 certificate. However it fails to correctly handle any errors that may occur while parsing the issuer field (which might occur if the issuer field is maliciously constructed). This may subsequently result in a NULL pointer deref and a crash leading to a potential denial of service attack. The function X509_issuer_and_serial_hash() is never directly called by OpenSSL itself so applications are only vulnerable if they use this function directly and they use it on certificates that may have been obtained from untrusted sources. OpenSSL versions 1.1.1i and below are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1j. OpenSSL versions 1.0.2x and below are affected by this issue. However OpenSSL 1.0.2 is out of support and no longer receiving public updates. Premium support customers of OpenSSL 1.0.2 should upgrade to 1.0.2y. Other users should upgrade to 1.1.1j. Fixed in OpenSSL 1.1.1j (Affected 1.1.1-1.1.1i). Fixed in OpenSSL 1.0.2y (Affected 1.0.2-1.0.2x).

Problem Type:

[CWE-476](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 5.9 (Medium)

CVE-2021-3449

CVE 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).

Problem Type:

[CWE-476](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 5.9 (Medium)

CVE-2021-3711

CVE 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).

Problem Type:

CWE-120

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H](#)

Base Score: 9.8 (Critical)

CVE-2021-3712

CVE 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 represented 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).

Problem Type:

CWE-125

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:H](#)

Base Score: 7.4 (High)

CVE-2021-4160

CVE 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).

Problem Type:

[Carry-propagating bug](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N](#)

Base Score: 5.9 (Medium)

CVE-2022-0778

CVE description: The BN_mod_sqrt() function, which computes a modular square root, contains a bug that can cause it to loop forever for non-prime moduli. Internally this function is used when parsing certificates that contain elliptic curve public keys in compressed form or explicit elliptic curve parameters with a base point encoded in compressed form. It is possible to trigger the infinite loop by crafting a certificate that has invalid explicit curve parameters. Since certificate parsing happens prior to verification of the certificate signature, any process that parses an externally supplied certificate may thus be subject to a denial of service attack. The infinite loop can also be reached when parsing crafted private keys as they can contain explicit elliptic curve parameters. Thus vulnerable situations include: - TLS clients consuming server certificates - TLS servers consuming client certificates - Hosting providers taking

certificates or private keys from customers - Certificate authorities parsing certification requests from subscribers - Anything else which parses ASN.1 elliptic curve parameters Also any other applications that use the BN_mod_sqrt() where the attacker can control the parameter values are vulnerable to this DoS issue. In the OpenSSL 1.0.2 version the public key is not parsed during initial parsing of the certificate which makes it slightly harder to trigger the infinite loop. However any operation which requires the public key from the certificate will trigger the infinite loop. In particular the attacker can use a self-signed certificate to trigger the loop during verification of the certificate signature. This issue affects OpenSSL versions 1.0.2, 1.1.1 and 3.0. It was addressed in the releases of 1.1.1n and 3.0.2 on the 15th March 2022. Fixed in OpenSSL 3.0.2 (Affected 3.0.0,3.0.1). Fixed in OpenSSL 1.1.1n (Affected 1.1.1-1.1.1m). Fixed in OpenSSL 1.0.2zd (Affected 1.0.2-1.0.2zc).

Problem Type:

[CWE-835](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 7.5 (High)

CVE-2022-1292

CVE 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).

Problem Type:

[CWE-78](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H](#)

Base Score: 9.8 (Critical)

CVE-2022-2068

CVE 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).

Problem Type:

[CWE-78](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H](#)

Base Score: 9.8 (Critical)

CVE-2022-2097

CVE 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).

Problem Type:

[CWE-327](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N](#)

Base Score: 5.3 (Medium)

CVE-2022-4304

CVE description: A timing based side channel exists in the OpenSSL RSA Decryption implementation which could be sufficient to recover a plaintext across a network in a Bleichenbacher style attack. To achieve a successful decryption an attacker would have to be able to send a very large number of trial messages for decryption. The vulnerability affects all RSA padding modes: PKCS#1 v1.5, RSA-OEAP and RSASVE.

For example, in a TLS connection, RSA is commonly used by a client to send an encrypted pre-master secret to the server. An attacker that had observed a genuine connection between a client and a server could use this flaw to send trial messages to the server and record the time taken to process them. After a sufficiently large number of messages the attacker could recover the pre-master secret used for the original connection and thus be able to decrypt the application data sent over that connection.

Problem Type:

[CWE-203](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N](#)

Base Score: 5.9 (Medium)

CVE-2022-4450

CVE description: The function `PEM_read_bio_ex()` reads a PEM file from a BIO and parses and decodes the “name” (e.g. “CERTIFICATE”), any header data and the payload data. If the function succeeds then the “name_out”, “header” and “data” arguments are populated with pointers to buffers containing the relevant decoded data. The caller is responsible for freeing those buffers. It is possible to construct a PEM file that results in 0 bytes of payload data. In this case `PEM_read_bio_ex()` will return a failure code but will populate the header argument with a pointer to a buffer that has already been freed. If the caller also frees this buffer then a double free will occur. This will most likely lead to a crash. This could be exploited by an attacker who has the ability to supply malicious PEM files for parsing to achieve a denial of service attack.

The functions `PEM_read_bio()` and `PEM_read()` are simple wrappers around `PEM_read_bio_ex()` and therefore these functions are also directly affected.

These functions are also called indirectly by a number of other OpenSSL functions including `PEM_X509_INFO_read_bio_ex()` and `SSL_CTX_use_serverinfo_file()` which are also vulnerable. Some OpenSSL internal uses of these functions are not vulnerable because the caller does not free the header argument if `PEM_read_bio_ex()` returns a failure code. These locations include the `PEM_read_bio_TYPE()` functions as well as the decoders introduced in OpenSSL 3.0.

The OpenSSL `asn1parse` command line application is also impacted by this issue.

Problem Type:

CWE-415

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 7.5 (High)

CVE-2023-0215

CVE description: The public API function BIO_new_NDEF is a helper function used for streaming ASN.1 data via a BIO. It is primarily used internally to OpenSSL to support the SMIME, CMS and PKCS7 streaming capabilities, but may also be called directly by end user applications.

The function receives a BIO from the caller, prepends a new BIO_f_asn1 filter BIO onto the front of it to form a BIO chain, and then returns the new head of the BIO chain to the caller. Under certain conditions, for example if a CMS recipient public key is invalid, the new filter BIO is freed and the function returns a NULL result indicating a failure. However, in this case, the BIO chain is not properly cleaned up and the BIO passed by the caller still retains internal pointers to the previously freed filter BIO. If the caller then goes on to call BIO_pop() on the BIO then a use-after-free will occur. This will most likely result in a crash.

This scenario occurs directly in the internal function B64_write_ASN1() which may cause BIO_new_NDEF() to be called and will subsequently call BIO_pop() on the BIO. This internal function is in turn called by the public API functions PEM_write_bio_ASN1_stream, PEM_write_bio_CMS_stream, PEM_write_bio_PKCS7_stream, SMIME_write_ASN1, SMIME_write_CMS and SMIME_write_PKCS7.

Other public API functions that may be impacted by this include i2d_ASN1_bio_stream, BIO_new_CMS, BIO_new_PKCS7, i2d_CMS_bio_stream and i2d_PKCS7_bio_stream.

The OpenSSL cms and smime command line applications are similarly affected.

Problem Type:

CWE-416

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 7.5 (High)

CVE-2023-0286

CVE description: There is a type confusion vulnerability relating to X.400 address processing inside an X.509 GeneralName. X.400 addresses were parsed as an ASN1_STRING but the public structure definition for GENERAL_NAME incorrectly specified the type of the x400Address field as ASN1_TYPE. This field is subsequently interpreted by the OpenSSL function GENERAL_NAME_cmp as an ASN1_TYPE rather than an ASN1_STRING.

When CRL checking is enabled (i.e. the application sets the X509_V_FLAG_CRL_CHECK flag), this vulnerability may allow an attacker to pass arbitrary pointers to a memcmp call, enabling them to read memory contents or enact a denial of service. In most cases, the attack requires the attacker to provide both the certificate chain and CRL, neither of which need to have a valid signature. If the attacker only controls one of these inputs, the other input must already contain an X.400 address as a CRL distribution point, which is uncommon. As such, this vulnerability is most likely to only affect applications which have implemented their own functionality for retrieving CRLs over a network.

Problem Type:

[CWE-843](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:H](#)

Base Score: 7.4 (High)

CVE-2023-0464

CVE description: A security vulnerability has been identified in all supported versions

of OpenSSL related to the verification of X.509 certificate chains that include policy constraints. Attackers may be able to exploit this vulnerability by creating a malicious certificate chain that triggers exponential use of computational resources, leading to a denial-of-service (DoS) attack on affected systems.

Policy processing is disabled by default but can be enabled by passing the -policy' argument to the command line utilities or by calling the X509_VERIFY_PARAM_set1_policies()' function.

Problem Type:

[CWE-295](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

Base Score: 7.5 (High)

CVE-2023-0465

CVE description: Applications that use a non-default option when verifying certificates may be vulnerable to an attack from a malicious CA to circumvent certain checks.

Invalid certificate policies in leaf certificates are silently ignored by OpenSSL and other certificate policy checks are skipped for that certificate. A malicious CA could use this to deliberately assert invalid certificate policies in order to circumvent policy checking on the certificate altogether.

Policy processing is disabled by default but can be enabled by passing the `-policy` argument to the command line utilities or by calling the `X509_VERIFY_PARAM_set1_policies()` function.

Problem Type:

[CWE-295](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N](#)

Base Score: 5.3 (Medium)

CVE-2023-0466

CVE description: The function `X509_VERIFY_PARAM_add0_policy()` is documented to implicitly enable the certificate policy check when doing certificate verification. However the implementation of the function does not enable the check which allows certificates with invalid or incorrect policies to pass the certificate verification.

As suddenly enabling the policy check could break existing deployments it was decided to keep the existing behavior of the `X509_VERIFY_PARAM_add0_policy()` function.

Instead the applications that require OpenSSL to perform certificate policy check need to use `X509_VERIFY_PARAM_set1_policies()` or explicitly enable the policy check by calling `X509_VERIFY_PARAM_set_flags()` with the `X509_V_FLAG_POLICY_CHECK` flag argument.

Certificate policy checks are disabled by default in OpenSSL and are not commonly

used by applications.

Problem Type:

[CWE-295](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N](#)

Base Score: 5.3 (Medium)

CVE-2023-2650

CVE description: Issue summary: Processing some specially crafted ASN.1 object identifiers or data containing them may be very slow.

Impact summary: Applications that use OBJ_obj2txt() directly, or use any of the OpenSSL subsystems OCSP, PKCS7/SMIME, CMS, CMP/CRMF or TS with no message size limit may experience notable to very long delays when processing those messages, which may lead to a Denial of Service.

An OBJECT IDENTIFIER is composed of a series of numbers - sub-identifiers - most of which have no size limit. OBJ_obj2txt() may be used to translate an ASN.1 OBJECT IDENTIFIER given in DER encoding form (using the OpenSSL type ASN1_OBJECT) to its canonical numeric text form, which are the sub-identifiers of the OBJECT IDENTIFIER in decimal form, separated by periods.

When one of the sub-identifiers in the OBJECT IDENTIFIER is very large (these are sizes that are seen as absurdly large, taking up tens or hundreds of KiBs), the translation to a decimal number in text may take a very long time. The time complexity is $O(n^2)$ with 'n' being the size of the sub-identifiers in bytes (*).

With OpenSSL 3.0, support to fetch cryptographic algorithms using names / identifiers in string form was introduced. This includes using OBJECT IDENTIFIERs in canonical numeric text form as identifiers for fetching algorithms.

Such OBJECT IDENTIFIERs may be received through the ASN.1 structure AlgorithmIdentifier, which is commonly used in multiple protocols to specify what cryptographic algorithm should be used to sign or verify, encrypt or decrypt, or digest passed data.

Applications that call OBJ_obj2txt() directly with untrusted data are affected, with any version of OpenSSL. If the use is for the mere purpose of display, the severity is considered low.

In OpenSSL 3.0 and newer, this affects the subsystems OCSP, PKCS7/SMIME, CMS, CMP/CRMF or TS. It also impacts anything that processes X.509 certificates, including simple things like verifying its signature.

The impact on TLS is relatively low, because all versions of OpenSSL have a 100KiB limit on the peer's certificate chain. Additionally, this only impacts clients, or servers that have explicitly enabled client authentication.

In OpenSSL 1.1.1 and 1.0.2, this only affects displaying diverse objects, such as X.509 certificates. This is assumed to not happen in such a way that it would cause a Denial of Service, so these versions are considered not affected by this issue in such a way that it would be cause for concern, and the severity is therefore considered low.

Problem Type:

[CWE-770](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H](#)

Base Score: 6.5 (Medium)

CVE-2023-3817

CVE description: Issue summary: Checking excessively long DH keys or parameters may be very slow.

Impact summary: Applications that use the functions DH_check(), DH_check_ex() or EVP_PKEY_param_check() to check a DH key or DH parameters may experience long delays. Where the key or parameters that are being checked have been obtained from an untrusted source this may lead to a Denial of Service.

The function DH_check() performs various checks on DH parameters. After fixing CVE-2023-3446 it was discovered that a large q parameter value can also trigger an overly long computation during some of these checks. A correct q value, if present, cannot be larger than the modulus p parameter, thus it is unnecessary to perform these checks if q is larger than p.

An application that calls `DH_check()` and supplies a key or parameters obtained from an untrusted source could be vulnerable to a Denial of Service attack.

The function `DH_check()` is itself called by a number of other OpenSSL functions. An application calling any of those other functions may similarly be affected. The other functions affected by this are `DH_check_ex()` and `EVP_PKEY_param_check()`.

Also vulnerable are the OpenSSL `dhparam` and `pkeyparam` command line applications when using the “-check” option.

The OpenSSL SSL/TLS implementation is not affected by this issue.

The OpenSSL 3.0 and 3.1 FIPS providers are not affected by this issue.

Problem Type:

[CWE-834](#)

CVSS Vector String: [CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L](#)

Base Score: 5.3 (Medium)

CVE-2023-4807

CVE description: Issue summary: The POLY1305 MAC (message authentication code) implementation contains a bug that might corrupt the internal state of applications on the Windows 64 platform when running on newer X86_64 processors supporting the AVX512-IFMA instructions.

Impact summary: If in an application that uses the OpenSSL library an attacker can influence whether the POLY1305 MAC algorithm is used, the application state might be corrupted with various application dependent consequences.

The POLY1305 MAC (message authentication code) implementation in OpenSSL does not save the contents of non-volatile XMM registers on Windows 64 platform when calculating the MAC of data larger than 64 bytes. Before returning to the caller all the XMM registers are set to zero rather than restoring their previous content. The vulnerable code is used only on newer x86_64 processors supporting the AVX512-IFMA instructions.

The consequences of this kind of internal application state corruption can be various - from no consequences, if the calling application does not depend on the contents of non-volatile XMM registers at all, to the worst consequences, where the attacker could get complete control of the application process. However given the contents of the registers are just zeroized so the attacker cannot put arbitrary values inside, the most likely consequence, if any, would be an incorrect result of some application dependent calculations or a crash leading to a denial of service.

The POLY1305 MAC algorithm is most frequently used as part of the CHACHA20-POLY1305 AEAD (authenticated encryption with associated data) algorithm. The most common usage of this AEAD cipher is with TLS protocol versions 1.2 and 1.3 and a malicious client can influence whether this AEAD cipher is used by the server. This implies that server applications using OpenSSL can be potentially impacted. However we are currently not aware of any concrete application that would be affected by this issue therefore we consider this a Low severity security issue.

As a workaround the AVX512-IFMA instructions support can be disabled at runtime by setting the environment variable `OPENSSL_ia32cap`:

```
OPENSSL_ia32cap=~0x200000
```

The FIPS provider is not affected by this issue.

Problem Type:

[CWE-374](#)

CVSS Vector String: [CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H](#)

Base Score: 7.8 (High)

Remarks

Security Update Information

With respect to Directive (EU) 2019/770 and Directive (EU) 2019/771 and their national transposition laws, please note:

It is your responsibility to download and/or install any security updates provided by us, for example to maintain product or data security. If you fail to install a security update

provided to you within a reasonable period of time, we will not be liable for any product defect solely due to the absence of such security update.

Alternatively, we are entitled to directly download and/or install security updates regardless of your settings. In these cases, we will provide you with the relevant information, e.g. in this security advisory.

CVSS Scoring

Vulnerability classification has been performed using the [CVSS v3.1 scoring system](#) . The CVSS environmental score is specific to each customer's environment and should be defined by the customer to attain a final scoring.

Additional Resources

[1] BVMS Download Area: <https://downloadstore.boschsecurity.com/index.php?type=BVMS>

[2] BVMS Viewer Download Area: <https://downloadstore.boschsecurity.com/index.php?type=BVMSVWR>

[3] BVMS Appliances (DIVAR IP) Download Area: <https://downloadstore.boschsecurity.com/?type=DIPBVMS>

Please contact the Bosch PSIRT if you have feedback, comments, or additional information about this vulnerability at: psirt@bosch.com .

Revision History

06 Mar 2024: Initial Publication

Appendix

Fixes for the Affected Products

BVMS

Affected versions	Version or patch that fixes the vulnerability
11.1.1	BVMS111165_Patch_VCATattile_431808,419002,399765,38747
12.0.1	BVMS1201375_Patch_Security_419002.zip

[BVMS Download Area](#)

BVMS Viewer

Affected versions	Version or patch that fixes the vulnerability
11.1.1	BVMS111165_VWR_Patch_VCATattile_431808,419002,399765,3
12.0.1	BVMS1201375_VWR_Patch_Security_419002.zip

[BVMS Viewer Download Area](#)

Bosch DIVAR IP all-in-one 7000 R3

Affected BVMS versions	Version or patch that fixes the vulnerability
11.1.1	BVMS_11.1.1_Updates_SystemManager_package_1.3.zip
12.0.1	BVMS_12.0.1_Updates_SystemManager_package_1.0.zip

[BVMS Appliances Download Area](#)

Bosch DIVAR IP 7000 R2

Affected BVMS versions	Version or patch that fixes the vulnerability
11.1.1	BVMS111165_Patch_VCATattile_431808,419002,399765;
12.0.1	BVMS1201375_Patch_Security_419002.zip

[BVMS Appliances Download Area](#)

Bosch DIVAR IP all-in-one 5000

Affected BVMS versions	Version or patch that fixes the vulnerability
11.1.1	BVMS_11.1.1_Updates_SystemManager_package_1.3.zip
12.0.1	BVMS_12.0.1_Updates_SystemManager_package_1.0.zip

[BVMS Appliances Download Area](#)

Bosch DIVAR IP all-in-one 7000

Affected BVMS versions	Version or patch that fixes the vulnerability
11.1.1	BVMS_11.1.1_Updates_SystemManager_package_1.3.zip
12.0.1	BVMS_12.0.1_Updates_SystemManager_package_1.0.zip

[BVMS Appliances Download Area](#)

DIVAR IP all-in-one 4000

Affected BVMS versions	Version or patch that fixes the vulnerability
11.1.1	BVMS_11.1.1_Updates_SystemManager_package_1.3.zip
12.0.1	BVMS_12.0.1_Updates_SystemManager_package_1.0.zip

[BVMS Appliances Download Area](#)

DIVAR IP all-in-one 6000

Affected BVMS versions	Version or patch that fixes the vulnerability
11.1.1	BVMS_11.1.1_Updates_SystemManager_package_1.3.zip
12.0.1	BVMS_12.0.1_Updates_SystemManager_package_1.0.zip

[BVMS Appliances Download Area](#)

Material Lists

BVMS

Family Name	CTN	SAP#	Material description
BVMS	MBV-	F.01U.393.647	License Professional

Family Name	CTN	SAP#	Material description
Professional 12.0.1	BPRO		base
BVMS Plus 12.0.1	MBV- BPLU	F.01U.393.650	License Plus base
BVMS Plus 12.0.1 DIP	MBV- BPLU- DIP	F.01U.374.503	License Plus base for DIVAR IP
BVMS Viewer 12.0.1	MBV- BVWR	F.01U.393.649	License Viewer base
BVMS Lite 12.0.1	MBV- BLIT	F.01U.393.648	License Lite base
BVMS Lite 12.0.1 DIP	MBV- BLIT- DIP	F.01U.358.975	License Lite base for DIVAR IP
BVMS Professional 11.1.1	MBV- BPRO	F.01U.393.647	License Professional base
BVMS Plus 11.1.1	MBV- BPLU	F.01U.393.650	License Plus base
BVMS Plus 11.1.1 DIP	MBV- BPLU- DIP	F.01U.374.503	License Plus base for DIVAR IP

Family Name	CTN	SAP#	Material description
BVMS Viewer 11.1.1	MBV- BVWR	F.01U.393.649	License Viewer base
BVMS Lite 11.1.1	MBV- BLIT	F.01U.393.648	License Lite base
BVMS Lite 11.1.1 DIP	MBV- BLIT- DIP	F.01U.358.975	License Lite base for DIVAR IP

Bosch DIVAR IP 7000 R2

Family Name	CTN	SAP#	Material description
DIVAR IP 7000 R2	DIP-7180-00N	F.01U.314.520	DIVAR IP 7000 2U w/o HDD
DIVAR IP 7000 R2	DIP-7183-4HD	F.01U.314.521	DIVAR IP 7000 2U 4x3TB
DIVAR IP 7000 R2	DIP-7183-8HD	F.01U.314.522	DIVAR IP 7000 2U 8x3TB
DIVAR IP 7000 R2	DIP-7184-4HD	F.01U.314.523	DIVAR IP 7000 2U 4x4TB
DIVAR IP	DIP-7184-8HD	F.01U.314.524	DIVAR IP 7000 2U

Family Name	CTN	SAP#	Material description
7000 R2			8x4TB
DIVAR IP 7000 R2	DIP-71F0-00N	F.01U.314.525	DIVAR IP 7000 3U w/o HDD
DIVAR IP 7000 R2	DIP-71F3-16HD	F.01U.314.526	DIVAR IP 7000 3U 16x3TB
DIVAR IP 7000 R2	DIP-71F4-16HD	F.01U.314.527	DIVAR IP 7000 3U 16x4TB
DIVAR IP 7000 R2	DIP-7186-8HD	F.01U.329.143	DIVAR IP 7000 2U 8x6TB
DIVAR IP 7000 R2	DIP-7188-8HD	F.01U.329.144	DIVAR IP 7000 2U 8x8TB
DIVAR IP 7000 R2	DIP-71F6-16HD	F.01U.329.145	DIVAR IP 7000 3U 16x6TB
DIVAR IP 7000 R2	DIP-71F8-16HD	F.01U.329.146	DIVAR IP 7000 3U 16x8TB
DIVAR IP 7000 R2	DIP-7184-8HD- WAG	F.01U.343.277	DIVAR IP 7000 2U 8x4TB, WAG Kit

Bosch DIVAR IP all-in-one 5000

Family Name	CTN	SAP#	Material description
DIVAR IP all-in-one 5000	DIP-5240IG-00N	F.01U.361.821	Management Appliance w/o HDD
DIVAR IP all-in-one 5000	DIP-5244IG-4HD	F.01U.362.424	Management Appliance 4x4TB
DIVAR IP all-in-one 5000	DIP-5248IG-4HD	F.01U.362.423	Management Appliance 4x8TB
DIVAR IP all-in-one 5000	DIP-524CIG-4HD	F.01U.362.422	Management Appliance 4x12TB
DIVAR IP all-in-one 5000	DIP-5240GP-00N	F.01U.359.551	Management Appliance GPU wo HD
DIVAR IP all-in-one 5000	DIP-5244GP-4HD	F.01U.359.552	Management Appliance GPU 4x4TB
DIVAR IP all-in-one 5000	DIP-5248GP-4HD	F.01U.359.553	Management Appliance GPU 4x8TB
DIVAR IP all-in-one 5000	DIP-524CGP-4HD	F.01U.359.554	Management Appliance GPU 4x12TB

Bosch DIVAR IP all-in-one 7000

Family Name	CTN	SAP#	Material description
DIVAR IP all-in-one 7000	DIP-7280-00N	F.01U.362.591	2U Management Appliance w/o HD
DIVAR IP all-in-one 7000	DIP-7284-8HD	F.01U.362.592	2U Management Appliance 8x4TB
DIVAR IP all-in-one 7000	DIP-7288-8HD	F.01U.362.593	2U Management Appliance 8x8TB
DIVAR IP all-in-one 7000	DIP-728C-8HD	F.01U.362.594	2U Management Appliance 8x12TB
DIVAR IP all-in-one 7000	DIP-72G0-00N	F.01U.362.595	3U Management Appliance wo HDD
DIVAR IP all-in-one 7000	DIP-72G8-16HD	F.01U.362.596	3U Management Appliance 16x8TB
DIVAR IP all-in-one 7000	DIP-72GC-16HD	F.01U.362.597	3U Management Appliance 16x12T

DIVAR IP all-in-one 7000 R3

Family Name	CTN	SAP#	Material description
-------------	-----	------	----------------------

Family Name	CTN	SAP#	Material description
DIVAR IP all-in-one 7000	DIP-7380-00N	F.01U.385.539	Management appliance 2U without HD
DIVAR IP all-in-one 7000	DIP-7384-8HD	F.01U.385.540	Management appliance 2U 8X4TB
DIVAR IP all-in-one 7000	DIP-7388-8HD	F.01U.385.541	Management appliance 2U 8X8 TB
DIVAR IP all-in-one 7000	DIP-738C-8HD	F.01U.385.542	Management appliance 2U 8X12 TB
DIVAR IP all-in-one 7000	DIP-73G0-00N	F.01U.385.543	Management appliance 3U without HD
DIVAR IP all-in-one 7000	DIP-73G8-16HD	F.01U.385.544	Management appliance 3U 16X8TB
DIVAR IP all-in-one 7000	DIP-73GC-16HD	F.01U.385.545	Management appliance 3U 16X12 TB

DIVAR IP all-in-one 4000

Family Name	CTN	SAP#	Material description
DIVAR IP all-in-one 4000	DIP-4420IG-00N	F.01U.404.040	Management appliance w/o HDD
DIVAR IP all-in-one 4000	DIP-4424IG-2HD	F.01U.404.041	Management appliance 2x4TB
DIVAR IP all-in-one 4000	DIP-4428IG-2HD	F.01U.404.042	Management appliance 2x8TB
DIVAR IP all-in-one 4000	DIP-442IIG-2HD	F.01U.404.043	Management appliance 2x18TB

DIVAR IP all-in-one 6000

Family Name	CTN	SAP#	Material description
DIVAR IP all-in-one 6000	DIP-6440IG-00N	F.01U.404.045	Management appliance 1U w/o HDD
DIVAR IP all-in-one 6000	DIP-6444IG-4HD	F.01U.404.046	Management appliance 1U 4x4TB
DIVAR IP all-in-one 6000	DIP-6448IG-4HD	F.01U.404.047	Management appliance 1U 4x8TB
DIVAR IP all-	DIP-644IIG-4HD	F.01U.404.048	Management

Family Name

CTN

SAP#

Material description

in-one 6000

appliance 1U
4x18TB
