MatrixSSL 3.2.2 Open Source Release Notes

PeerSec Networks, Inc. 410 Broadway Ave E. #205 Seattle, WA 98102 **T** 425.646.7850 **F** 206.501.4366

support@peersec.com www.peersec.com

Overview

Thank you for choosing MatrixSSL. The 3.2.2 version is a minor revision to the 3.2.1 release.

Who Is This Document For?

- Software developers that are securing applications with MatrixSSL
- Software developers upgrading to MatrixSSL 3.2.2 from a previous version
- Anyone wanting to learn more about MatrixSSL

PeerSec Networks

MatrixSSL 3.2.2 Release Notes

This section highlights the differences between version 3.2.1 and 3.2.2

Security Improvement

The BEAST Vulnerability Workaround

In Sept. 2011 security researchers demonstrated how a previously known CBC encryption weakness could be used to decrypt HTTP data over SSL. The attack was named BEAST (Browser Exploit Against SSL/TLS).

A new compile-time define `USE_BEAST_WORKAROUND` has been added to *matrixsslConfig.h* to thwart the attack. It is enabled by default.

As with previous SSL vulnerabilities, the attack is generally considered a very low risk for individual browsers as it requires the attacker to have control over the network to become a MIM. They will also have to have knowledge of the first couple blocks of underlying plaintext in order to mount the attack.

A zero length record proceeding a data record has been a known fix to this problem for years and MatrixSSL has always supported the handling of empty records.

This BEAST fix is on the sending side and moves the implementation down to the SSL library level so users do not need to manually send zero length records. This fix uses the same IV obfuscation logic as a zero length record by breaking up each application data record in two. The first being just a single byte of the plaintext message.

This issue only effects TLS 1.0 (and SSL) and only if the cipher suite is using a symmetric CBC block cipher. Enable USE_TLS_1_1 above to completely negate the need for any workaround if TLS 1.1 is also supported by peers.

Feature Additions

PKCS#12 Parsing Added

Support for parsing the most common PKCS#12 formats has been added to this version of MatrixSSL. PKCS#12 is the recommended format for securely storing certificates and their associated private key in the same file. The parsing has been introduced through the new public API `matrixSslLoadPkcs12` and should be used as a replacement for `matrixSslLoadRsaKeys` where appropriate. Please see the full details of this new function in the API documentation.

RC2 Added

PKCS#12 formats often encrypt the public certificate using the legacy RC2 cipher. This should be the only reason to enable this outdated and insecure algorithm.

Public API Change

x509SubjectAltName_t structure renamed to x509GeneralName_t

This structure type has been renamed to reflect the correct generic X.509 type rather than the specific Subject Alternate Name (san) of a certificate. This change should have a very low impact as this structure type was never a direct parameter to any public API. This structure type is currently only used as the san member in the x509v3extensions_t which a user might be examining as part of the custom validation of a certificate in the certificate callback function. See the section The Certificate Validation Callback Function in the API document for full details.

Bug Fixes

Server Clears Session Entry On Failed Handshakes

An issue was discovered in which the server would leave a partial session resumption entry in its table even though the initial handshake with the client failed. If the client, for some reason, choose to use that session ID in subsequent `CLIENT_HELLO` messages, the server was locating the entry and attempting a resumed handshake with an erroneous master secret. Of course, the connection would not succeed since the secret was not correct between the peers but the server should not have been finding the entry to begin with. The table entry is now removed on handshake failures.

AES Cipher Contexts Internally Zeroed

An issue was discovered in which the standalone use of the AES cipher could fail if the initialization function was called with a context structure that contained non-`NULL` data. The key initialization functions now internally `memset` the context structures to `0x0` to prevent this problem.

No Double Frees When Deleting Key Material After Errors

An issue was discovered in which a call to `matrixSslDeleteKeys` after the failure of `matrixSslLoadRsaKeys` would perform a second memory free on data members that had been handled in the error cases of `matrixSslLoadRsaKeys` itself. The error handling internal to `matrixSslLoadRsaKeys` will now NULL any freed memory to prevent this problem.

Minor Tweak

ASN.1 Parser Accepts Indefinite Length Formats

The addition of PKCS#12 uncovered the use of indefinite length encoding. Low level calls to `getAsnLength` will now return `ASN_UNKNOWN_LEN` rather than an error beginning in this release.

Support and Bug Reporting

Contacting Support or Reporting Bugs

Email support@peersec.com