# 510007 - Additional considerations about setting up SSL on Application Server ABAP

Component: BC-SEC-SSL (Basis Components > Security - Read KBA 2985997 for subcomponents > Secure Sockets Layer Protocol), Version: 189, Released On: 31.05.2024

## Symptom

This SAP Note contains additional general information about setting up SSL on Application Server ABAP (AS ABAP).

Sections 1+2: installing & enabling SAPCRYPTOLIB *old* Netweaver Kernels (year 2014 or older) (BC-SEC-SSL)
Section 3: creating ICM SSL Server PSEs for SAP WebAS through transaction STRUST (CA-FLP-ABA, BC-WD-ABA, BC-FES-BUS, BC-ESI-WS-ABA, BC-CST-IC, BC-SEC-SSF)
Section 4+5: creating ICM SSL client PSEs for SAP WebAS through transaction STRUST (BC-MID-ICF, BC-ESI-WS-ABA, OPU-GW-COR, CA-FLP-ABA, BC-CST-IC, BC-SEC-SSF)
Section 6: customizing available TLS cipher suites (BC-SEC-SSL, BC-IAM-SSO-CCL)
Section 7: customizing available TLS protocol versions (how to enable TLSv1.2) (BC-SEC-SSL, BC-IAM-SSO-CCL)
Section 8: Interoperability (issues) with third-party TLS implementations (BC-SEC-SSL, BC-IAM-SSO-CCL)

This Note does not (and can not) substitute documentation on how to configure & enable use of SSL/TLS on Netweaver Kernel components (e.g. icman, sapwebdisp, hostagent, etc.).
This Note also can not substitute application-specific documentation on setting up application-specific "HTTP over TLS" data exchange usage scenarios and all required application-level connection parameters (e.g. trust anchor certificates) for a secure communication to become possible.

## Other Terms

SSL, TLS, Transport Layer Security, HTTPS, encryption, trust manager, STRUST, cipher suites, SAPCRYPTOLIB, SAP WebAS ABAP

## Solution

This SAP Note provides additional considerations and general common guidance on setting up SSL/TLS on Netweaver Application Server ABAP. Documentation for setting up specific application usage scenarios must be provided by each application.

Unless you are using very old Netweaver Kernel software, you should take note of section (2a) about removing old custom profile parameter values, and then continue reading with section (3)

There are two SAPCRYPTOLIB variants (old/new) that can be recognized and distinguished based on their (complete) names. Instances of the "old" library are called "SAPCRYPTOLIB 5.5.5 plXX" (for example, "5.5.5pl38"), whereas the newer variant of the SAPCRYPTOLIB has the name "CommonCryptoLib 8" (shortened to "CCL") and has a version of the form "8.<major>.<minor>" (for example, "8.4.31"). For use in AS ABAP 70x or 71x, CommonCryptoLib 8 requires a downward-compatible 72x Kernel (at least 720 patchno 88). CommonCryptoLib 8 can **not** be used with Netweaver AS ABAP 640.

For SAP NetWeaver 74X, a SAPCRYPTOLIB in the new variant CommonCryptoLib 8 is a regular component of the delivery (kernel CD). As of November 2013, "CommonCryptoLib 8" is also part of new 72x kernel patches (in the download from SAP Service Marketplace, in the packages SAPEXE and dw_utils). This means that manual installation of the library (described in section 1) for Netweaver 74x systems is only necessary to get the most recent library version (newer than what a particular Kernel includes, such as a stack kernel), and Netweaver 74x includes suitable predefined profile parameter values (described in section 2), so custom profile parameters for the library location are no longer necessary (and can be removed after upgrade of AS ABAP to 74x).

1. **Manual Installation SAPCRYPTOLIB (obsolete for Netweaver 72x/74x kernel patches released after November 2013)**

   Install SAPCRYPTOLIB and the corresponding sapgenpse command line tool in the directory $DIR_EXECUTABLE on all application servers. Historic versions of SAPCRYPTO 5.5.5 (prior to pl32) required a separate license ticket file ("ticket"), which had to be installed in the directory $DIR_INSTANCE/sec. CommonCryptoLib 8 never had a ticket file.

   Netweaver Kernel 6.20/6.40 and 70x AS ABAP kernels (disp+work) automatically create a ticket file in $DIR_INSTANCE/sec when the AppServer is started. Beginning with SAPCRYPTOLIB 5.5.5 pl32, the library no longer requires a license ticket file. Anyhow, the SAPCRYPTO 5.5.5 pl32+ download packages still contain a file ticket (because of existing installation instructions that describe the installation of the file ticket). The actual content of the file ticket is irrelevant for SAPCRYPTOLIB 5.5.5pl32 and later libraries.

Independent of SAPCRYPTOLIB, the NetWeaver Administrator in SAP AS Java checks for the existence of a file ticket. If necessary, you must produce a file ticket there with the required contents in $DIR_INSTANCE/sec to satisfy the check function in the NetWeaver Administrator.

Distribution packages of CommonCryptoLib 8 contain no file named ticket. To ensure continuously consistent system behavior, set the environment variable SECUDIR in the environment of the user <sid>adm (or SAPService<SID> or both) in the directory $DIR_INSTANCE/sec on all application servers. If you want to protect the PSEs (key files) with a password, on UNIX systems, you must also set the environment variable USER to the name of the UNIX user under whom the SAP system is running.

2. **a) Netweaver Kernel 74x and newer, plus 72x kernel patches produced after November 2013**

preferably ***remove*** these custom profile parameters from your instance profiles and DEFAULT.PFL, which might still be present from an earlier manual installation of SAPCRYPTOLIB, or from a semi-automatic download & installation of SAPCRYPTOLIB 5.5.5 by 72x SAP installer or SAP upgrade tools:

| | | |
|---|---|---|
| ssf/name | | |
| ssf/ssfapi_lib | | |
| sec/libsapsecu | | |
| ssl/ssl_lib | | |

The predefined / builtin parameter values point to the CommonCryptoLib 8 library in $(DIR_EXECUTABLE), which has been included in all Netweaver Kernel patches since November 2013.

**b) **OBSOLETE** manual First-Time Installation or Configuration of SAPCRYPTOLIB on SAP NetWeaver 6xx, 70X, 71X, 72x Systems with **OLD** Kernels (older than November 2013)**

In SAP Netweaver Release 740+ and in Netweaver 72x Kernel-Patches created in 2014 or later, SAPCRYPTOLIB is automatically installed in $(DIR_EXECUTABLE) along with all other Kernel libraries and executables, so that you should use the following forward-compatible parameter values on all application server platforms with **OLD** kernels (from before November 2013). Each SAP AppServer will substitute the necessary platform-specific parts, like the path seperator character, shared library prefix and shared library filename extension :

| | | | |
|---|---|---|---|
| | ssf/name | = | SAPSECULIB |
| | ssf/ssfapi_lib | = | $(DIR_EXECUTABLE)$(DIR_SEP)$(FT_DLL_PREFIX)sapcrypto$(FT_DLL) |
| | sec/libsapsecu | = | $(DIR_EXECUTABLE)$(DIR_SEP)$(FT_DLL_PREFIX)sapcrypto$(FT_DLL) |
| | ssl/ssl_lib | = | $(DIR_EXECUTABLE)$(DIR_SEP)$(FT_DLL_PREFIX)sapcrypto$(FT_DLL) |

740+ Kernels plus 72x Kernel patches created after Q2/2014 know a predefined variable "$(SAPCRYPTOLIB)", where the platform-specific shared library prefix and filename extension is supplied by the Kernel. You can use tx RZ11, tx RSPFPAR or ABAP report "RSPARAM" to lookup current & predefined profile parameter settings known to AS ABAP, and to check whether it provides a definition for $(SAPCRYPTOLIB). If $(SAPCRYPTOLIB) is predefined in the kernel, then there also will be suitable predefined values for these profile parameters, and you better remove all custom values from your profiles:

| | | | |
|---|---|---|---|
| | ssf/name | = | SAPSECULIB |
| | ssf/ssfapi_lib | = | $(SAPCRYPTOLIB) |
| | sec/libsapsecu | = | $(SAPCRYPTOLIB) |
| | ssl/ssl_lib | = | $(SAPCRYPTOLIB) |

To configure an actual service that uses HTTPS for an incoming connection, use icm service definitions with any of the applicable plugins (such as e.g. the HTTPS plugin). If you do not want to configure a listening port for HTTPS, you can specify PORT=0.

| | | | |
|---|---|---|---|
| | icm/server_port_X | = | PROT=HTTPS, PORT=<TCP port number for HTTPS> |

If you want to suppress/permit/enforce user logon by client certificate in the SSL protocol:

| | | |
|---|---|---|
| icm/HTTPS/verify_client | = | 0 / 1 (default) / 2 |

In SAP Netweaver 6.xx the STRUST user interface does not offer selection of a key size when creating new (SSL) PSEs. As a workaround, you can set the following profile parameter for specifying the keysize that will be used when creating new RSA PSEs (this requires at least an ABAP kernel Release 6.20 or higher, see SAP Note 509495).

| | | |
|---|---|---|
| sec/rsakeylengthdefault | = | 2048 |

3. **Creating ICM SSL Server PSE(s) for Netweaver AS ABAP with ABAP cross-application PSE management tool "Trust Manager" (tx STRUST from BC-SEC-SSF)**

When using SSL/TLS to protect network communication, the server of the communication scenario is typically authenticated by an X.509 certificate, and there is a convention to identify the server by matching the server hostname from the connection parameters (such as the URL) to name attributes in the certificate. This matching is called "server endpoint identification", and was first described in Section 3.1 of rfc2818 "HTTP over TLS" based on the behaviour implemented in common web browsers at the time. Similar checking of server endpoint identification has been adopted by other protocols that use TLS, and has been described in rfc6125.

https://tools.ietf.org/html/rfc2818#section-3.1
https://tools.ietf.org/html/rfc6125#section-6.4

When creating ICM SSL server PSEs, you will have to place the fully qualified DNS hostname(s) ("FQDN") of the servers in the Name part of your server certificate(s), otherwise clients connecting to your server with SSL/TLS will report errors about incorrect/mismatching server certificate(s). Most modern browsers (e.g. Chromium-based browsers Edge and Chrome, and also Firefox) require TLS server certificates to contain the server name in a SubjectAltName attribute. Please refer to SAP Note 2478769 on how to create Certificate Signing Requests (CSRs) with one or multiple SubjectAltName(s) within ABAP transaction STRUST.

You can configure HTTPS access to your SAP WebAS either directly, or through a single entry point that terminates SSL, such as SAP WebDispatcher, or third-party load balancer or reverse proxy. When accessing your SAP WebAS directly with a Web Browser, you will have to create a Certificate Signing Request (CSR) for each of the ICM SSL Server PSEs that you create. You have to submit the CSRs to a Certification Authority of your own choice, and install each certification response from the chosen CA exactly into that SSL Server PSE from which the corresponding CSR was created. A certification response can only be imported into the PSE which contains the private key corresponding to the public key in the CA-signed certificate! The certification response must be a complete certificate chain up to and including a self-signed RootCA certificate, typically a PKCS#7 chain. Should the CA *FAIL* to provide a complete certification response, and instead return only a naked server certificate, then you will have to manually compose a proper certification response by collecting the missing intermediate CA and Root CA certificate(s) and concatenating them for import of the certification response into the PSE (see SAP Note 508307). Running your server with a self-signed certificate will result in a Browser warning about an untrusted server certificate when accessing the server. Clicking through an untrusted certificate warning in your Browser is not secure, and some browsers (such as MSIE) do exhibit occasional visual malfunctions with complex pages after clicking through a server certificate warning, in particular when the server sends request for an SSL client request (the default behaviour of SAP WebAS, see icm/HTTPS/verify_client and VCLIENT option to icm/server_port_XX).

Configuring SSL for use and choosing an appropriate CA / trust model is going to be a trade-off between effort and expense. In principle, you have these three choices:

(1) purchase TLS server certificates from one of the traditional public CAs or one of their resellers, whose RootCA certificates are pre-installed and pre-trusted in most Operating Systems and all Web Browsers.

(2) obtain TLS server certificates from one of the Free Certificate Authorities. Their RootCA certs may not be present on all platforms/browsers, and there may be other challenges, such as short certificate lifetimes (requiring frequent renewals)

(3) run your own organizational CA to sign your own TLS server certificates.

According to the "Baseline Requirement" rules of the CA-Browser Forum (https://cabforum.org) public CAs will issue TLS server certificates only for officially registered DNS domains, but not for server name from private DNS domains. If you are using server hostnames from a private DNS universe, (i.e. underneath a domain name that is not officially registered by your organization), then you are limited to option (3) and have to run your own organizational

CA. In this case, your IT must take care of distributing&installing your private RootCA trust anchor certificate(s) to all relevant TLS clients (desktop and mobile).

To create or maintain ICM SSL server PSE(s), start ABAP transaction STRUST ("Trust Manager"). It is recommended that you select keylength 2048 and algorithm "RSA with SHA-256" when creating new ICM SSL PSEs.

a. Create the default ICM "SSL server Standard" PSE (that will be supplied to / used by all AS ABAP instances that do not have an instance-specific assignment for the ICM SSL server PSE).
When using a wildcard server certificate or a multi-hostname certificate that is valid for the hostnames of all your AppServer Instances, then you do not need any instance-specific ICM "SSL server Standard" PSE(s).

   Choose "Create" in the context menu of the "SSL server" node. ABAP "Trust manager" makes proposals for name components, but it really depends on your chosen Certificate Authority, which name component this wants to see in your Certificate Signing Request. In particular, set the following values ("Name" actually refers to the X.500 attribute "Common Name" [CN]):

| | | |
|---|---|---|
| Name | = | *.<Domain of AS ABAP> |

   Whenever you are going to install instances on multiple different servers, it will be preferable to either use a wildcard / asterisk character (*) for the leftmost DNS label in the system-wide ICM SSL Server PSE, or alternatively to create a multi-domain certificate that enumerates multiple server hostnames, or to use a partial wildcard pattern in the leftmost DNS label. Please refer to SAP note 2478769 for how to create Certificate Signing Requests (CSRs) which are pre-populated with SubjectAltName attributes for your desired server hostnames. Some Certificate Authorities (including Microsoft Certificate Services) desperately require pre-populated SubjectAltName attributes in Certificate Signing Requests. It is OK to use a single hostname in the system-wide ICM "SSL server Standard" PSE for a system with a single instance, or with all instances of your system on a single server.

   **Optionally** create&maintain individual ICM SSL Server PSE assignments for individual instances. While hovering the mouse pointer over the SSL server PSE node name (such as "SSL server Standard") press the right mouse button for the context menu and select menu item "Change" to bring up a configuration window for instance-specific assignments of the selected ICM SSL server PSE in STRUST. The proposed Distinguished Name (DN) contains the following entry:

b.

| | | |
|---|---|---|
| Name | = | <Host name>.<Domain of AS ABAP> |

   Ensure that each instance is assigned the fully qualified host name that is used in the HTTPS protocol. A DN can be assigned to multiple instances at the same time, for example, if a Network Address Translators (NAT) is used - in this case, as CN, the fully qualified host name of the NAT must be specified. All instances with an empty DN receive the default PSE. Two instances can use the same DN, and instances with the same DName assignment will share their instance-specific PSE. Each DN can have a maximum of 253 characters.

c. Create certificate requests for all instance PSEs. Expand the "SSL server" node in the tree control, double-click to load the instance PSE into the relevant node and select the "Generate certificate request" function.

   For the default PSE, you must only create a certificate request if there are instances without their own PSEs (in this case, double-click on node name "SSL server Standard" to load the default PSE into the "SSL server" node). Send the Certificate Signing Requests (CSRs) to a Certification Authority (CA) of your own choice. The certificate response must be either a PKCS#7 package with a complete upwards path, or must be a text file that contains a concatenated list of all of the required certificates in PEM format (i.e. base64-encoded with the header "-----BEGIN CERTIFICATE-----" and the footer "-----END CERTIFICATE-----").

   As of Release 6.20, you can also import the certificate response as an individual PEM certificate if the CA certificate is saved in the database (to search for certificates, select "Import certificate", category = "Server CA"). Using the SAP Trust Center Service ensures that the certificate response has a valid format. Always import the certificate response into the PSE from which the original certificate request was generated (double-click on the corresponding nodes and call the "Import certificate response" function) and save the changes.

d. If you want to allow logon to your server using a TLS client certificate for authentication, you have to import either the issuingCA *or* the RootCA certificate (of each PKI issuing acceptable TLS client certificates) into **one** of the ICM SSL server PSEs. When you save the ICM SSL server PSE in STRUST, the system updates the certificate list of **all** (instance-specific and system-wide) incarnations of that ICM SSL server PSE. The "Certificate List" of ICM SSL Server PSEs should contain only the (Root)CA certificates of those CAs/PKIs that issue (to-be-)acceptable TLS client certificates for authentication by TLS client certificate to your Netweaver system. Additionally, you can add self-signed X.509v1 proto-certificates of direct communication peers, such as a self-signed "Own Certificate" of your ICM "SSL client (Standard)" PSE, or the self-signed "Own Certificate" from the SAPSSLC.pse of a SAP

WebDispatcher, a sapwebdisp that has been installed and configured to operate as reverse proxy in front of your Netweaver system.

4. **Creating the ICM "SSL client (Standard)" PSE / SAPSSLC.pse and the ICM "SSL client (Anonymous)" PSE / SAPSSLA.pse in Netweaver AS ABAP with ABAP cross-application PSE management tool "Trust Manager" (tx STRUST from BC-SEC-SSF)**

   The intended purpose of ICM "SSL client (Standard)" PSE is for communication with other (organization-internal) SAP Systems and other servers of the same SAP system (including itself), using an SSL/TLS client certificate for authentication. Unsolicited SSL/TLS client certificates are prohibited in SSL/TLS. An SSL/TLS client certificate can only be used on a connection when the server explicitly requests it during the SSL/TLS handshake (which for icm is configured through icm/HTTPS/verify_client=1/2 or the service-specific VCLIENT=1/2 setting).
   When creating ICM "SSL client (Standard)" PSE, STRUST proposes the subject name:

| | | |
|---|---|---|
| Name | = | <System SID> SSL client default |

   Should ICM "SSL client (Standard)" (SAPSSLC.pse) not exist, then ICM "SSL server Standard" PSE (SAPSSLS.pse) will instead be used by icman as (historic) fallback. When using ICM "SSL client (Standard)" PSE to communicate with other SAP systems and actually using authentication by TLS client certificate to logon to other systems (sometimes confusingly described as "two-way-TLS" or "mutual TLS"), you might consider installing a CA-signed certificate from your own organizational CA into ICM "SSL client (Standard)" PSE, which may simplify the configuration of the necessary trust relationships in a multi-system landscape. However, when using application-level authentication between systems (such as e.g. user&password, SAML or OAUTH) keeping a long-lived self-signed proto-certificate in ICM "SSL client (Standard)" PSE will avoid the administrative burden to regularly renew a short-lived CA-signed certificate.

   The intended purpose of ICM "SSL client (Anonymous)" PSE is for communication limited to "*unidirectional server-towards-client authentication by certificate*", similar to most web browser usage. When an application requests use of ICM "SSL client (Anonymous)" PSE, *no* TLS client certificate will be sent to any servers. This includes defective/misconfigured servers, that send an incomplete request for a client certificate, and fail to provide a list of acceptable certificate_authorities. Some of those defective servers get highly confused, and erroneously abort the TLS handshake, when receiving an arbitrary self-signed SSL client certificate, or an SSL client certificate from a CA unknown to the server, in response to an incomplete request for an SSL client certificate. (As it turns out, Netweaver AS Java 7.50 is one such defective server, as a result of AS Java Netweaver Administrator creating a malformed SAPSSLS.pse by default. The server-side workaround is to add at least one trusted Certificate Authority to SAPSSLS.pse in AS Java Netweaver Administrator. The client-side workaround is to have applications request use of ICM "SSL client (Anonymous)" PSE) Since the certificate from ICM "SSL client (Anonymous)" PSE will never be sent, the Subject Name in the "Own certificate" of the PSE does not matter (STRUST proposes "CN=anonymous", please *USE* this name), and self-signed is fine.

   When ICM "SSL client PSE (Anonymous)" aka SAPSSLA.pse does not exist, then ICM "SSL client (Standard)" PSE aka SAPSSLC.pse will be used as fallback SSL client PSE (and another fallback to SAPSSLS.pse is used, when ICM "SSL client PSE (Standard)" aka SAPSSLC.pse does not exist either).

   **IMPORTANT:** Different to Web Browsers, **SAP WebAS does *not* come with a huge amount (350+) of pre-installed and pre-trusted (Root)CA certificates of omnipotent public CAs**. Another difference is, that each (client) SSL PSE contains a seperate and independent list of trust anchor certificates for verifying server certificates / server certificate chains. Only the "Certificate List"s of the instance-specific&system-wide incarnations of ICM SSL server PSEs are synchronized by STRUST, so that all Instances of a system will present the same list of acceptable certificate_authorities for TLS client certificates.

   For each communication scenario that uses a particular ICM SSL client PSE, the SSL/TLS certificate (chain) of a server can only be cryptographically verified, when a suitable trust anchor certificate has previously been added to the "Certificate List" of the application-selected ICM SSL client PSE. **Creating&verifying the necessary trust relationship is a required configuration step for every communication scenario that uses SSL/TLS, and each application scenario or web service that publishes/uses a HTTPS URL for access, must describe where to obtain the trust anchor certificate that will be necessary to cryptographically verify the SSL/TLS certificate (chain) of the server**.

   **For short-lived CA-issued TLS server certificates, the appropriate trust anchor is pretty much always the self-signed RootCA certificate of the PKI which issued the CA-signed TLS server certificate. In TLS communication scenarios, that RootCA certificate ought to be sufficient to verify the TLS server certificate, because the TLS specification requires the server to convey the complete certificate chain, and omit at most the self-signed RootCA certificate at the end of the certificate chain. Though, a small number of defective servers fail to send all intermediateCA certificates in violation of the TLS protocol**

**specification, in which case you may consider the workaround of adding that missing intermediateCA certificate as trust anchor to the application-selected ICM SSL client PSE. Please avoid using CA-issued TLS server certificates themselves as trust anchors, because this will result in communication failures and require updating of the trust *whenever* the TLS server certificate is replaced (due to certificate expiration or due to a suspected or real security breach of that server). Use of a CA-signed TLS server certificate also entirely defeats the purpose of using a short-lived CA-signed TLS server certificate. Please do *not* add the entire certificate chain, either. Expiring certificates in "Certificate Lists" of PSEs result in certificate expiration alerts when these certificates approach their expiration date. Excess / unnecessary, short-lived CA-issued TLS server certificates in the Certificate List of an SSL client PSE may therefore result in needless confusion through certificate expiration alerts in tx SM02 system messages shown to every user during system logon, and additionally in daily SM21 syslog alert messages. (See also SAP Note 2890773 on certificate expiration alerts.)**

**For point-to-point TLS-protected communication links and when using self-signed end-entity certificates, there is no alternative to configuring direct trust, however. A common example for point-to-point TLS-protected communication is between sapwebdisp and backend Netweaver system(s).**

If you forgot to configure trust, or when the server certificate was unexpectedly replaced with one from a different issuer, or when a man-in-the-middle attack is performed on the connection, the TLS handshake will (and must) fail with the return code ICM_HTTP_SSL_PEER_CERT_UNTRUSTED / SSSLERR_PEER_CERT_UNTRUSTED. To overcome this handshake failure, you will have to check whether you forgot to configure trust (missing trust anchor), or whether that change in the issuer of the server certificate is appropriate but unexpected (server/service admin forgot to notify about PKI change), in which case you have to update/adjust your application-level trust configuration accordingly. An unexpected change of issuer may indicate a man-in-the-middle attack on the communication, where the attacker is trying to impersonate the real server. In some environments, man-in-the-middle impersonation attacks may be performed by "TLS intercepting proxies" operated by your own IT/Networking department. In such a situation, you will have to talk to your own IT department on guidance how to configure your communication scenario. Should the communication require the use of an explicit SSL client certificate, then this communication scenario might have to be permitted untampered by that TLS intercepting proxy.

Service providers that intend to replace their current SSL/TLS server certificate with one from a different public CA, and even a server certificate issued under a different RootCA certificate of the same public CA, will have to provide ample advance notice to all of the users / consumers of that service, so that the trust configuration of all TLS clients (communication peers) to this service can be updated before the new server certificate is installed.

The return code SSSLERR_SERVER_CERT_MISMATCH for an outgoing TLS-protected communication indicates a fatal rfc2818 "HTTP over TLS" section 3.1 "Server Endpoint Identification" mismatch. It means that the certificate presented by the TLS server is not valid for the target hostname supplied by the calling application, so that the communication had to be safely aborted. Please ensure that your Netweaver Kernel software implements client-side sending of the optional TLS extension SNI (SAP Note 2124480 and SAP Note 2582368 for Netweaver 74x Kernels and SAP Note 2384290 for Netweaver 721 and 722 kernels), and that you *did* enable sending of TLS extension SNI (icm/HTTPS/client_sni_enabled=TRUE, ssl/client_sni_enabled=TRUE and environment variable SAPSSL_CLIENT_SNI_ENABLED=TRUE for legacy saphttp) on Netweaver software Releases *older* than S/4 HANA 1809. Other possible causes for SSSLERR_SERVER_CERT_MISMATCH might be a misconfigured server, i.e a server where the server admin installed a TLS server certificate with incorrect hostnames. Another possibility is an incorrect "Target Host" parameter value, such as use of an unqualified hostname or use of an IPv4 address rather than a fully qualified DNS hostname. Or maybe someone or something is performing a man-in-the-middle attack on your communication, and presenting a fake TLS server certificate with incorrect name attributes in it -- though the last one is much more likely going to fail with SSSLERR_PEER_CERT_UNTRUSTED before any rfc2818 section 3.1 name matching is attempted.

Some communication scenarios may not get as far as the server responding with ServerHello and ServerCertificate TLS handshake messages, but instead, the connection might become unexpectedly closed (SSSLRC_CONN_CLOSED) without a single byte of response received from the server (*SSL_get_state()==0x2120 "TLS read server hello A"*). There exist a few defective TLS server implementations, which choke and drop the network connection entirely without response, if they do not like the TLS protocol options offered in ClientHello. Such server behaviour is in violation of the TLS protocol specification, which requires a server to respond with a fatal TLS alert before closing the network connection. Another notoriously common cause of unexpected network connection closures is a misconfigured and defective network middlebox ("firewall") in the communication path. Defective network middleboxes which erroneously let through the initial 3-way TCP network connection establishment handshake, and which then surreptitiously assassinate the network connection after the client has sent the initial ClientHello handshake message -- making the visible behaviour of a misconfigured and defective network middlebox ("firewall") in the communication path indistinguishable from the behaviour of a defective target TLS server, a server that fails to send a fatal TLS alert before closing the network connection.

A second type of misconfigured and defective network middleboxes ("firewalls") is erroneously letting through the initial 3-way TCP network connection establishment handshake, and then blackholing parts or all of the actual communication, resulting in starvation of the network connection. ICMAN will deliberately abort the network communication at the

application level, if the remote server does not complete the SSL/TLS handshake within the timeout defined by ICM profile parameter "icm/conn_timeout" -- with the dev_icm trace typically erroneously showing the NON-error return code SSSLRC_EWOULDBLOCK instead of explaining ICM's deliberate decision to abort the network connection with an unresponsive server.

A *correct* firewall must block already the 3-way TCP network connection establishment handshake, and must respond with an ICMP destination_unreachable message with an appropriate subtype code between 9 and 13, so that TCP connect() fails upfront, and he complete lack of network connectivity to the desired destination becomes fairly obvious!

5. **Creating Additional ICM SSL Client PSEs (Optional)**

   You can define additional SSL client identities (Environment -> SSL client-> Identities) for individual, application-specific communication scenarios. After creation of new ICM SSL Client identities, new nodes are displayed in Trust manager (STRUST). You can now create the relevant ICM SSL client PSEs files, and when a communication scenario uses authentication by TLS client certificate, get your public key certificates signed by a CA that is accepted by the target server/service for this communication scenario.

   Should authentication by TLS client certificate be desired (or be required by the server), then where&how to obtain a PKI credential with an acceptable TLS client certificate must be specified in the documentation/description of your application scenario *and* also must be documented by the service provider that is operating the target server / providing the target service. In case that the service provider is distributing PKI credentials with an acceptable SSL client certificate as PKCS#12 / PFX file, then you will have to use (1) a procedure similar to what is described in SAP KBA 2148457 in order to convert the PKCS#12 / PFX file into a PSE file, and then (2) use the procedure described in SAP KBA 2148372 to create a custom SSL client Identity in tx STRUST, upload the new PSE file created in step (1) and save it into the new, scenario-specific SSL client Identity.

   You will then have to adjust the configuration of your Netweaver AS ABAP client-side application to request use of your new scenario-specific SSL client PSE instead of "SSL client (Standard)" PSE for communication, and you will have to add the relevant trust anchor certificate for verification of the server certificate (chain) to the "Certificate List" of your new, scenario-specific SSL client PSE -- or else encounter the lack-of-trust error (ICM_SSL_PEER_CERT_UNTRUSTED / SSSLERR_PEER_CERT_UNTRUSTED). Where to obtain a suitable long-lived trust anchor certificate for verification of the TLS server certificate chain of the target service, must also be specified in the documentation/description of your application usage scenario *and* also must be documented by the service provider that is operating the target server / providing the target service.

   Changes made to ICM SSL PSEs in Trust Manager (STRUST), such as importing the certification response of a CA, and changes to the Certificate List of trust anchors, will be updated at runtime and reloaded by icman (with no interruption of service) beginning with Netweaver AS ABAP 710 plus AS ABAP 702. On *OLD* releases of Netweaver 700/701 and 64x, changes to contents of ICM SSL PSEs will take effect in icman only after you manually restart the ICMAN process (transaction SMICM, "Administration -> ICMAN -> Exit Soft -> Global"). Manually restarting the icman process will interrupt&abort active long-running requests, so please do not lightheartedly restart icman on actively used production systems. Changes to SSL PSEs of standalone programs that are not maintained through Trust Manager (STRUST) may also require restart of the affected program (sapwebdisp, msgsrv, sapstartsrv, saphostagent, saphttp, sapkprotp, sldreg). Changing contents of a sapwebdisp's SSL PSEs through sapwebdisp webadmin UI will result in sapwebdisp performing SSL PSE reload, obviating a manual sapwebdisp process restart.

6. **(Optional) Configuration of Available SSL/TLS Cipher Suites -- please use recommended values from beginning of section 7**

   You can configure the available SSL/TLS cipher suites and their order for AS ABAP (icman, sapwebdisp, msg_server, and saphttp), plus **incoming** SSL-protected communication of SAP AS Java 710+ with the following two profile parameters. Outgoing SSL-protected communication from SAP AS Java (all versions), e.g. using Java-based XI/PI/PO communication adapters, are NOT affected by these parameters, because outgoing SSL-protected communication from AS Java uses the native-Java OEM SSL-Implementation "IAIK" included with SAP AS Java rather than SAPCRYPTOLIB (configuring SSL/TLS for AS Java see SAP Note 2284059 and 2708581 instead):

| | | |
|---|---|---|
| Netweaver AS ABAP (all Releases)<br>server-side of Netweaver AS Java (710+) | ssl/ciphersuites | = |
| Netweaver AS ABAP<br>(721/722/740+ Kernels or<br>old 70x/71x/720 Kernels with Kernel patch 1433874) | ssl/client_ciphersuites | = |

   With SAPCRYPTOLIB, the server-side cipher suite preference ordering always takes precedence over the client-proposed list when searching for a common cipher suite with a client.

   **NOTE:** When adding above profile parameters to AppServer profiles with profile maintenance transaction RZ10 on SAP Netweaver 73x or earlier, please ignore RZ10 warning(s) "Unknown Parameter ... a check cannot be performed". The

default cipher suite settings up to SAP Netweaver 73x are built-in defaults rather than predefined profile parameters. Please see section 7 for currently recommended custom settings for the above two profile parameters.

Outgoing SSL connection (SSL client) will all offer the cipher suites configured by (ssl/client_ciphersuites). Netweaver Kernels predating the Kernel patch from SAP Note 1433874 use the "ssl/ciphersuites" setting also for outgoing SSL connections. For backwards compatibility, Kernel patch 1433874 does not have a built-in default setting for "ssl/client_ciphersuites", and will use the "ssl/ciphersuites" setting as fallback unless a custom setting is configured.

Incoming SSL connections (SSL server/services) can optionally be configured to use service-specific cipher suite settings in the SSL configuration part icm/ssl_config_<xx> for an icm server port definition icm/server_port_<xx> via the string parameter CIPHERS:

| | | |
|---|---|---|
| icm/server_port_<xx> | = | ..., SSLCONFIG=ssl_config_<yy> |
| icm/ssl_config_<yy> | = | ..., CIPHERS=... |

The contents of parameter CIPHERS is subject to the same rules and syntax that applies to profile parameter ssl/ciphersuites.

**CAVEAT**: for CommonCryptoLib 8.4.38 and newer, please refer to the output of "sapgenpse tlsinfo 135:PFS:HIGH" (for server) and to the output of "sapgenpse tlsinfo -c 150:PFS:HIGH" (for client) for the list of TLS cipher suites supported by your version of CommonCryptoLib when using the recommended profile parameter values from section 7. For comparison, the output of "sapgenpse tlsinfo HIGH:MEDIUM:+e3DES" (for server) and "sapgenpse tlsinfo -c HIGH:MEDIUM:+e3DES" (for client) shows the TLS cipher suites that are used by Netweaver 70x->753 Kernels in default configuration with your version of CommonCryptoLib.

7. **Recommended Configuration of Available TLS Protocol Versions (required for enabling TLSv1.2)**

Over the course of year 2016, a growing number of TLS servers were reconfigured to abort/reject TLSv1.0 handshakes, or they are requring forward secrecy (PFS) cipher suites for access. **The currently recommended settings for TLSv1.2 interoperability are** (requiring at least CommonCryptoLib version 8.5.4 from Sep-2016):

```
ssl/ciphersuites         = 135:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH

ssl/client_ciphersuites  = 150:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH

icm/HTTPS/client_sni_enabled = TRUE

ssl/client_sni_enabled = TRUE

SETENV_26 = SECUDIR=$(DIR_INSTANCE)$(DIR_SEP)sec
SETENV_27 = SAPSSL_CLIENT_CIPHERSUITES=150:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH
SETENV_28 = SAPSSL_CLIENT_SNI_ENABLED=TRUE
```

The use of the above recommended seven profile paramters with legacy Netweaver 7.0/7.1/7.2/7.3/7.4/7.5 systems **will** resolve vulnerability findings from third-party scan tools about availability RC4 cipher suites (BAR MITZVA / CVE-2013-2566 + CVE-2015-2808) and about availability of 3DES-EDE-CBC cipher suites (SWEET32 / CVE-2016-2183).

**Please add these parameters to your DEFAULT.PFL through tx RZ10, save and activate the profile. Please ignore the tx RZ10/RZ11 warnings in Netweaver 70x/71x/72x/73x/74x about any of these profile parameters being unknown to ABAP tx RZ10 & RZ11. *BEWARE*: parameter value assignments in instance profile take precedence over assignments in DEFAULT.PFL. You will have to remove value assignments for the above parameters from all instance profiles of your system, if you want new value assignments in DEFAULT.PFL to take effect.**

**For a SAP Solution Manager System 7.[012], please use the following value for ssl/client_ciphersuites instead:**

```
ssl/client_ciphersuites = 918:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH
```

to work around a bug in tx SOLMAN_SETUP and bugs in two tx STC01 task lists "SAP_BASIS_CONFIG_OSS_COMM" and "SUPPORT_HUB_CONFIG", which misinterpret the recommended parameter flags value 150 in parameter ssl/client_ciphersuites.

The ciphersuite parameter values recommended above enable TLSv1.2+TLSv1.1+TLSv1.0, support for Perfect Forward

Secrecy (PFS) cipher suites, and blind sending of client certificates for outgoing SSL/TLS-protected communication, and DISable RC4-based TLS cipher suites (which are class MEDIUM). Beginning with CommonCryptoLib 8.5.4 (Sep-2016), the cipher suite 3DES_EDE_CBC was demoted from class HIGH to class MEDIUM, and both 3DES and RC4 cipher suites will be disabled by above recommended parameter values. **Beginning with CommonCryptoLib 8.5.42 (Feb-2022), the cipher suites class MEDIUM is now *empty*, and TLS cipher suites with 3DES and RC4 encryption require an explicit opt-in "e3DES" and "eRC4", see the example parameter values later in this section.**

**The recommended parameter values for ssl/client_ciphersuites and ssl/ciphersuites will provide the best possible features for all existing library versions, including CommonCryptoLib prior to 8.4.31 and predecessor library SAPCRYPTOLIB 5.5.5. It is strongly recommended to use above values, rather than rolling your own custom protocol versions and cipher suites, because you can easily configure real interoperability problems, while gaining ZERO security benefit.**

The recommended parameter values above also enable client-side sending of the optional TLS extension SNI, to further improve interop with Cloud-based servers (Azure, AWS), services hosted a some Content Distribution Networks (CDNs) such as Cloudflare and Akamai, plus Windows 2012R2 and Windows 2016 servers, all of which desperately requiring the presence of the optional TLS extension SNI for access (see SAP Note 2124480 for 74x Kernels and parameter icm/HTTPS/client_sni_enabled, and SAP Note 2384290 for 721/722 Kernels and parameter ssl/client_sni_enabled).

In order to provide parameter values in a fashion that is visible to profile-agnostic Netweaver kernel binaries, such as saphttp, sapkprotp, sldreg, sapcontrol, and a few others, the additional SETENV_XX parameters makes sapstart and sapstartsrv populate the environment with these environment variables before starting instance processes, such as disp+work. Environment variables are inherited by all child processes (see also SAP Note 2384290 issues 2a) and 2b) and SAP Note 2582368)

Please ignore the tx RZ10/RZ11 warnings in Netweaver 70x/71x/72x/73x/74x about any of these profile parameters being unknown to ABAP tx RZ10 & RZ11!!! Parameters *ssl/ciphersuites* and *ssl/client_ciphersuites* are recognized by all 7xx Kernels! Only parameter *ssl/client_sni_enabled* needs a somewhat recent kernel: 721 patchno 920 (Jan 2018), 722 patchno 223 (Jan 2017), see also SAP Note 2384290 -- or SAP Note 2582368 for kernels 745 patchno 623, 749 patchno 415, 753 patchno 110. Support for parameter *icm/HTTPS/client_sni_enabled* is limited to Netweaver 742+ kernels (SAP Note 2124480)

To make the SETENV_XX parameters take effect for AppServer child processes such as "saphttp", a restart of the AppServer instance is necessary.
Where changing the behaviour of icman is sufficient, it is possible to manually restart only the icman process, rather than the entire AppServer. Use tx SMICM, and from Menu "Administration"->"ICM"->"Exit Soft"->"Global"

Manually restarting the icman process may cause currently ongoing communication (such as long running requests) through icman to get terminated/aborted, so please don't restart icman lightheartedly on any actively used productive system. Manually restarting icman will also reset all dynamically changed values -- there is *NO* resync with dynamically changed profile parameter values from disp+work, so please always add profile parameter*icm/HTTPS/client_sni_enabled* to DEFAULT.PFL !!!.

**In case that you want to disable TLSv1.0, or configure your system for TLSv1.2-only, please see the caveats and required software & configuration updates (SAP and third-party) and the recommended profile parameter values for that particular purpose in SAP Note 2384290, more guidance in SAP Note 2384243, and for NW 74x kernels the additional kernel patch from SAP Note 2582368.**

You can see which version of CommonCryptoLib is currently installed and loaded by SapSSL from the SapSSL initialization message in dev_icm (or dev_webdisp) trace file(s), which after system restart should appear near the beginning of the trace file, and which after manual icman process restart, should appear near the end of the dev_icm trace file. For Netweaver 72x Kernels, the values of the ciphersuites profile parameters are reported only with Kernel patch from SAP Note 2384290:

```
] ================================================
] = SSL Initialization    platform tag=(linuxx86_64_gcc43)
] =   (749_STACK patchno 500,Apr 17 2018,mt,ascii-uc, 16/64/64)
] =      resulting Filename = "/usr/sap/C11/DVEBMGS00/exe/libsapcrypto.so"
] =   disabled FIPS 140-2 crypto kernel
] =   found CommonCryptoLib 8.5.20 (Apr  5 2018) [AES-NI,CLMUL,SSE3,SSSE3]
] =   current UserID: "c11adm",  env-var USER="c11adm"
] =   found SECUDIR environment variable
] =   using SECUDIR=/usr/sap/C11/DVEBMGS00/sec
] = [dpf] ssl/client_sni_enabled=TRUE
] =   NOT creating Envvar SAPSSL_CLIENT_SNI_ENABLED=1
] = [dpf] ssl/ciphersuites=135:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH
] =   creating Envvar SAPSSL_CIPHERSUITES=135:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH
```

] = **[dpf] ssl/client_ciphersuites=150:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH**
] = NOT creating Envvar SAPSSL_CLIENT_CIPHERSUITES=150:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH
] = Success SapCryptoLib SSL ready!
] ==========

Some customers newly encountered interoperability problems with the recommended settings, which they did not have with the default settings. This usually is caused by older hardware (we have heard about handheld RFID scanners) or older software (Windows CE, mobile Windows, Windows 2003) that is limited to SSLv3/TLSv1.0 cipher suites from class MEDIUM, and which lacks support for TLSv1.0 AES cipher suites from rfc3268 (June 2002). In face of interoperability problems (SSSLERR_NO_COMMON_CIPHERSUITE), or if you favor full backwards compatibility to Netweaver default settings, you can re-enable the old TLSv1.0 3DES cipher suite with these parameter values:

ssl/ciphersuites = 135:PFS:HIGH:e3DES::EC_X25519:EC_P256:EC_HIGH

ssl/client_ciphersuites = 150:PFS:HIGH:e3DES::EC_X25519:EC_P256:EC_HIGH

For completeness, the parameter values to enable all protocol versions, including the old SSLv3 protocol, and additionally the old RC4 cipher suites, are the following. But you really should not need SSLv3 anymore, and should refrain from enabling SSLv3, and should refrain from enabling RC4 cipher suites lightheartedly. **WARNING:** we have recently seen Microsoft Azure load-balancers failing to recognize TLS extension SNI in ClientHello, when the TLS record that is conveying the initial ClientHello has TLS record layer protocol version 03,00 (SSLv3) -- in spite of TLSv1.2 (rfc5246) explicitly requiring that TLS implementations *MUST* accept (03,XX) as record layer version for the intial TLSv1.2 ClientHello.
**Therefore, please try hard to *AVOID* using the below values (these values should *not* be necessary in year 2020+)!**

ssl/ciphersuites = 199:PFS:HIGH:e3DES:eRC4::EC_X25519:EC_P256:EC_HIGH

ssl/client_ciphersuites = 214:PFS:HIGH:e3DES:eRC4::EC_X25519:EC_P256:EC_HIGH

While it is possible to specify TLS protocol versions TLSv1.1 and TLSv1.2 explicitly, old libraries (CCL before 8.4.31 and SAPCRYPTOLIB 5.5.5) will reject the entire parameter value when they don't implement a protocol version that is explicitly requested, which causes SapSSL to resort to an extremely conservative emergency ciphersuites value.

For CommonCryptoLib version 8.4.31 and newer, the following explicit protocol version enumerations are equivalent to the recommended backwards-compatible values at the beginning of section 7 -- and can be used alternatively:

server-side: (TLSv1.2+TLSv1.1+TLSv1.0+BC) = (512 + 256 + 128 + BC) = 897
(TLSv1.2+TLSv1.1+TLSv1.0+NO_GAP+BEST_AVAILABLE+BC) = (512 + 256 + 128 + 4 + 2 + 1) = 903

client-side: (TLSv1.2+TLSv1.1+TLSv1.0+BLIND_CLIENT_CERT) = (512 + 256 + 128 + 16) = 912
(TLSv1.2+TLSv1.1+TLSV1.0+BLIND_CLIENT_CERT+NO_GAP+BEST_AVAILABLE) = (512 + 256 + 128 + 16 + 4 + 2) = 918

**BEWARE**: When copy&pasting profile parameter names from this SAP Note HTML display into the RZ10 profile editor, you may accidentally copy bogus Unicode Byte-Order-Mark (BOM) characters into the profile, which may cause the parameters to remain invisible to the SAP Netweaver kernel code. This is a shortcoming in the SAP Notes HTML display, and more shortcomings in all involved software components that copy BOMs through into the profile (instead of dropping bogus BOMs).

CommonCryptoLib 8 and the superseded/deprecated predecessor SAPCRYPTOLIB 5.5.5 support the following **default** and (*optional*) SSL & TLS protocol versions:

| | |
|---|---|
| CommonCryptoLib 8 >= 8.4.49 | server: (*SSLv3,*) **TLSv1.0, TLSv1.1, TLSv1.2, "BC"**<br>client: (*SSLv3,*) **TLSv1.0**, (*TLSv1.1, TLSv1.2*) |
| CommonCryptoLib 8 >= 8.4.31 | server: **SSLv3, TLSv1.0, TLSv1.1, TLSv1.2, "BC"**<br>client: **SSLv3, TLSv1.0**, (*TLSv1.1, TLSv1.2*) |
| CommonCryptoLib 8 <= 8.4.30<br>and<br>SAPCRYPTOLIB 5.5.5 >= pl28 | server: **SSLv3, TLSv1.0, "BC"**<br>client: **SSLv3, TLSv1.0** |
| SAPCRYPTOLIB 5.5.5 <= pl26 | server: **SSLv3, "BC"**<br>client: **SSLv3** |

For use in AS ABAP 70x or 71x, CommonCryptoLib 8 requires a downward-compatible 72x Kernel (at least 720 patchno 88). CommonCryptoLib 8 can **not** be used in AS ABAP 640 and neither with Netweaver Kernels 70x and 71x (the latter Kernels have been out of maintenance for quite some time).

Enabling TLSv1.1 or TLSv1.2 for outgoing/client communication (ssl/client_ciphersuites) may result in TLS handshake failures with a decreasing, but still non-marginal number of defective and/or old TLS server implementations, which will otherwise handshake TLSv1.0 just fine. The erratic server behaviour comes in several different flavours, such as choking on the presence of protocol version TLSv1.2 in ClientHello.client_version, choking on the inclusion of TLS extensions in the ClientHello handshake message (elemica) or choking on proposals of TLSv1.2 without TLS extensions (this causes Microsoft Windows 2008R2 and 2012R2 to choke) or the use of SSL/TLS client certificates in TLSv1.2 with (sha256,rsa) signatures (fails with certain SSL Load Balancers). Unfortunately, the IETF TLS working group has not yet standardized a suitable alternative TLS protocol version negotiation scheme, that would allow TLS clients to safely negotiate protocol versions > TLSv1.0 and TLS extensions in a fashion that will not break interoperability with the installed base (i.e. not break interop with TLS version intolerant and/or TLS extension intolerant servers and/or defective TLSv1.2 implementations). Web Browsers invented a heuristics-based, complex and insecure approach ("Downgrade Dance") to this non-marginal TLS interop problem ... and browsers therefore got bitten by POODLE.

The protocol version TLSv1.0, that was added with SAPCRYPTOLIB 5.5.5pl28+ is proposed or used by default if the communication partner also supports it. TLSv1.1 and TLSv1.2 support are enabled for the server-side by default if the library implements them. For the client-side, the default protocol is limited to TLSv1.0 because of a non-marginal number of version-intolerant TLS Servers that will choke and fail the TLS handshake when faced with TLSv1.1 or TLSv1.2 in the TLS ClientHello handshake message.

**NOTE:** In order to ensure interoperability with other communication peers and with itself (client to server), CommonCryptoLib will automatically enable TLSv1.0 and TLSv1.1 whenever TLSv1.1 or TLSv1.2 is requested/enabled through configuration. If you really want to disable TLSv1.0 and/or TLSv1.1, you need at least CommonCryptoLib 8.4.48 and must select "Strict protocol version configuration" (bitflag with value 32) in the protocol version flags. This is discouraged. You may encounter interoperability problems with some commmunication peers that can only be resolved by _not_ using the "strict protocol version configuration".

The protocol option "BC" for "Backwards Compatibility" allows an SSL Version 2.0 CLIENT-HELLO ("SSLv2Hello" in Java) as the first message of an SSLv3 or a TLS handshake. For a description of this backwards compatibility, see the following sections of the relevant standards:

https://tools.ietf.org/html/rfc6176#section-3
https://tools.ietf.org/html/rfc5246#page-89

A possibility for configuring TLS protocol versions was added with the kernel correction described in SAP Note 1433874. A number value (protocol version flags) can be inserted at the beginning of the SSL/TLS cipher suites strings in profile parameters ssl/ciphersuites and ssl/client_ciphersuites. This number has to be computed (added up) from the following (bit-flag) values:

| Value | Description |
|---|---|
| 1 | "BC"- Option (accept SSL Version 2.0 CLIENT-HELLO / SSLv2Hello for TLSv1.x Handshake) |
| 2 | "BEST"- Option (activate highest available TLS protocol version, i.e. TLSv1.2 for CCL 8.4.31+) |
| 4 | "NO_GAP"- Option (no gaps between TLS protocol versions; is forced to date) |
| 16 | Allow blind sending of a client certificate (5.5.5pl36+ and all CCL 8.x.x) |
| 32 | "Strict protocol version configuration" option--do not automatically enable TLSv1.0 (recognized/supported only by CommonCryptoLib (CCL) 8.4.48 or higher) |
| 64 | SSLv3 |
| 128 | TLSv1.0 |
| 256 | TLSv1.1 (only with CommonCryptoLib (CCL) 8.4.31 or higher) |
| 512 | TLSv1.2 (only with CommonCryptoLib (CCL) 8.4.31 or higher) |

With the built-in defaults, the server-side has all SSL&TLS protocol versions plus BC enabled. This corresponds to server-side protocol version flags (128+64+1) = 193 for SAPCRYPTO 5.5.5pl28+ and CommonCryptoLib 8 up to Version 8.4.30, and (512+256+128+64+1) = 961 for CommonCryptoLib 8.4.31+. The built-in defaults for the client-side enables only SSLv3 + TLSv1.0 for SAPCRYPTO 5.5.5pl28+ and CommonCryptoLib 8, corresponding to client-side protocol version flags (128+64) = 192. It is recommended to request TLS protocol version TLSv1.1 and TLSv1.2 with the flags "Best" and "NO_GAP", because only the latter is future-friendly and is fully compatible with older libraries.

Attacks like "BEAST", "CRIME" and "Poodle" do not affect programmatic TLS clients. These attacks require several vulnerabilities to be present in a TLS client, and can only be mounted against feature-bloated clients such as Web browsers. One of the prerequisites is willingness to execute active content provided by the attacker (such as Javascript, Java, Flash, or other powerful active content). These attacks also require that the TLS client gives the attacker free access to a Cross-Site-Request-Forgery (CSRF) facility, a browser misfeature. by which a web browser will happily supply user credentials and insert session cookies into arbitrary GET and POST requests authored by attackers.

8. **Information about Interoperability with other SSL and TLS implementations**

   a. Some Servers (including Servers hosted by Content Distribution Systems such as cloudfront) are being co-hosted with lots of other servers on a single IPv4 address, and are accessible only when Clients include TLS extension server_name_indication (SNI) from rfc6066 in their ClientHello handshake messages. Sending of TLS extensions is unfortunately not backwards compatible with a small, but non-marginal set of old Servers, so TLS extensions are not sent by default. For SAP Netweaver 741+ Kernels, sending of TLS extension SNI can be enabled through profile parameter icm/HTTPS/client_sni_enabled starting with Kernel Patch 2124480. Sending of TLS extension SNI as client can alternatively be enabled in 722 Kernel patchno 223 and higher and 721 Kernel patchno 921 and higher through profile parameter ssl/client_sni_enabled, see SAP Note 2384290.

   b. Microsoft SChannel in Windows 2008R2 and Windows 2012R2 will erroneously choke and abort the TLS handshake when a client offers TLSv1.2 in an extensionless ClientHello handshake message with ClientHello.client_version = TLSv1.2 and the server certificate is signed with sha256WithRsaEncryption. The recommended workaround to this Microsoft SChannel shortcoming is to use CommonCryptoLib 8.4.48+ and enable PFS cipher suites (ssl/client_ciphersuites=150:PFS:HIGH). Lesser desirable workarounds are to limit the protocol version to TLSv1.1 on the Microsoft server side (this server-side limit was the default in original Windows 2008R2) or to limit the protocol version to TLSv1.1 on the SAP client side (ssl/client_ciphersuites=400:PFS:HIGH).

   c. Microsoft Windows 2012 ADFS chokes and aborts on TLS handshakes from TLS client that do not include TLS extension server_name_indication because it fails to define a suitable default server certificate. Possible workarounds to this Windows 2012 shortcoming are manually adding the missing mapping of a default server certificate on the Microsoft Server, or enabling sending of TLS extension SNI in SAP Netweaver 742+ clients as described in SAP Note 2124480, or in Netweaver 722+ clients as described in SAP Note 2384290, or manually fixing the missing backwards compatibility in Windows 2012R2, e.g. https://blogs.technet.microsoft.com/applicationproxyblog/2014/06/19/how-to-support-non-sni-capable-clients-with-web-application-proxy-and-ad-fs-2012-r2/

   d. The protocol option "BC" (Backwards Compatibility) is required for interoperability with JavaSE 6 (still in use in Android 4.4 and J2EE servers based on JavaSE 6), Microsoft Internet Explorer (MSIE) Version 6 on Windows XP and Windows 2003, Firefox up to Version 3.0 and possibly other, mostly older, browsers and SSL clients.

Background: XP and 2003 were delivered with MSIE Version 6, and SSLv2 is activated and TLSv1.0 is deactivated there by default. When you upgrade MSIE to Version 7 or 8, SSLv2is deactivated and TLSv1.0 is activated. However, the basic attributes of the SChannel component of the underlying Windows version are not changed by an MSIE upgrade.

e. The attributes that can be used by Microsoft Internet Explorer for SSL or TLS are determined by the capabilities and attributes of the underlying operating system version, especially the security provider SChannel and the Microsoft CryptoAPI -- regardless of which browser version you have installed.

Some attributes are available on older versions of Microsoft Windows only after a Microsoft HotFix has been installed manually.

Microsoft SChannel support for AES cipher suites:

| | |
|---|---|
| XP 32-bit: | --- |
| 2003, XP 64-bit: | http://support.microsoft.com/kb/948963 |

Microsoft CryptoAPI Support for SHA256-based digital signatures, which includes SSL server certificates:

| | |
|---|---|
| XP 32+64, 2003: | http://support.microsoft.com/kb/968730 |

Microsoft SChannel It supports the use of the AES-based cipher suite (rfc3268) of SAPCRYTPOLIB pl28+ only in combination with protocol version TLSv1.0 and higher since Windows Vista (plus Windows 2003 after the manual installation of Microsoft Hotfix 948963).

Firefox 3+, Google Chrome, and OpenSSL 0.9.8+ support AES-based suites both on Windows XP and in combination with SSLv3

## Attributes

| Key | Value |
|---|---|
| Other Components | Basis Components > Security - Read KBA 2985997 for subcomponents > Secure Store and Forward (BC-SEC-SSF) |
| Other Components | Occasional Platform User > Gateway > Framework (OPU-GW-COR) |
| Other Components | Basis Components > Client/Server Technology > Web Dispatcher (BC-CST-WDP) |
| Other Components | Cross-Application Components > SAP Fiori Launchpad > SAP Fiori Launchpad ABAP Services (CA-FLP-ABA) |
| Other Components | Basis Components > Client/Server Technology > Internet Communication Manager (BC-CST-IC) |
| Other Components | Basis Components > Middleware > Internet Communication Framework (BC-MID-ICF) |
| Other Components | Basis Components > Enterprise Service Infrastructure > Web Service Infrastructure > Web Service and SOAP - ABAP (BC-ESI-WS-ABA) |
| Other Components | Basis Components > Web Dynpro > Web Dynpro ABAP (BC-WD-ABA) |
| Other Components | Basis Components > Frontend Services (SAP Note 1322184) > Netweaver Business Client (BC-FES-BUS) |
| Transaction codes | STRUST |
| Transaction codes | SMICM |

## Software Components

| Software Component | From | To | And subsequent |
|---|---|---|---|
| KRNL32NUC | 6.40 | 6.40EX2 | |
| KRNL32NUC | 7.00 | 7.01 | |
| KRNL32NUC | 7.10 | 7.20 | |
| KRNL32NUC | 7.20EXT | 7.20EXT | |
| KRNL32NUC | 7.21 | 7.21 | |
| KRNL32NUC | 7.21EXT | 7.21EXT | |
| KRNL32UC | 6.40 | 6.40EX2 | |
| KRNL32UC | 7.00 | 7.01 | |
| KRNL32UC | 7.10 | 7.20 | |
| KRNL32UC | 7.20EXT | 7.20EXT | |
| KRNL32UC | 7.21 | 7.21 | |
| KRNL32UC | 7.21EXT | 7.21EXT | |
| KRNL64NUC | 6.40 | 6.40EX2 | |
| KRNL64NUC | 7.00 | 7.01 | |
| KRNL64NUC | 7.10 | 7.20 | |
| KRNL64NUC | 7.20EXT | 7.20EXT | |
| KRNL64NUC | 7.21 | 7.21 | |
| KRNL64NUC | 7.21EXT | 7.21EXT | |
| KRNL64NUC | 7.40 | 7.40 | |
| KRNL64NUC | 7.22 | 7.22 | |
| KRNL64NUC | 7.22EXT | 7.22EXT | |
| KRNL64UC | 6.40 | 6.40EX2 | |
| KRNL64UC | 7.00 | 7.01 | |
| KRNL64UC | 7.10 | 7.20 | |
| KRNL64UC | 7.20EXT | 7.20EXT | |
| KRNL64UC | 8.04 | 8.04 | |
| KRNL64UC | 7.21 | 7.21 | |
| KRNL64UC | 7.21EXT | 7.21EXT | |
| KRNL64UC | 7.40 | 7.40 | |
| KRNL64UC | 7.22 | 7.22 | |
| KRNL64UC | 7.22EXT | 7.22EXT | |
| KERNEL | 6.40 | 6.40 | X |
| KERNEL | 7.00 | 7.01 | |
| KERNEL | 7.10 | 7.11 | |
| KERNEL | 7.20 | 7.22 | X |
| KERNEL | 8.04 | 8.04 | X |
| KERNEL | 7.40 | 7.40 | X |
| KERNEL | 7.70 | 7.70 | X |

## This document refers to

| SAP Note/KBA | Component | Title |
|---|---|---|

| | | |
|---|---|---|
| 2148457 | BC-IAM-SSO-CCL | How to convert the keypair of a PKCS#12 / PFX container into a PSE file |
| 2148372 | BC-SEC-SSF | How to create an own SSL Client PSE Identity |
| 834039 | BC-SEC-SSF | Certificate extension problems, Verisign (Japan) |
| 758667 | BC-OP-AS4 | IBM i: Installing Sapcrypto Library |
| 745103 | | |
| 700659 | SCM-TEC | Security Guide: mySAP Supply Chain Management |
| 698459 | BC-SEC-SSF | Trust manager: New root certificates |
| 662340 | BC-SEC-SSF | SSF Encryption Using the SAPCryptolib |
| 599270 | EP-PCT-SAP | Portal Content performance - composite SAP Note |
| 597959 | EP-PCT-SAP | Portal content performance on EP 5.0 SP 6 - Sammelhinweis |
| 578377 | BC-SEC-SSF | Digital signatures with SAPCRYPTOLIB |
| 517860 | BC-BSP | Logging on to BSP applications |
| 509495 | BC-SEC-SSF | Trust Manager: PSEs carry out generation with key length DSA > 512 bits and RSA > 1024 bits |
| 508307 | BC-SEC-SSF | Trust Manager: Problems importing certificate responses |
| 455033 | BC-SEC-SNC | SAPCRYPTOLIB versions, bugs and fixes |
| 397175 | BC-SEC | SAP Cryptographic software - export control |
| 354819 | BC-SEC-SSF | Collective note SAPSECULIB |
| 2729853 | CA-FLP-ABA | Web Browser warning "Connection is insecure" trying to access Fiori, WebDynpro, BSP, Webgui, SolMan, SUM, S4/HANA or Netweaver WebUI with https:// URL |
| 2708581 | BC-JAS-SEC-CPG | ECC Support for Outbound Connections in SAP NW AS Java |
| 2478769 | BC-SEC-SSF | Obtaining certificates with subject Alternative Name (SAN) within STRUST |
| 2429593 | BC-JVM | SAP JVM 6.1 Patch Collection 94 (build 6.1.099) |
| 2384290 | BC-SEC-SSL-CFG | SapSSL update to facilitate TLSv1.2-only configurations, TLSext SNI for 721+722 clients |
| 2340433 | CA-CL-CL | Error message 26 500 when saving the link classification in a classified document |
| 2323758 | BC-XI-CON-HTP | SSL support extension for TLS in HTTP_AAE adapter |
| 2295870 | BC-XI-CON-RST | TLSv1.2 support in REST adapter |
| 2292139 | BC-XI-CON-AXS | TLSv1.2 support in Axis adapter |
| 2287896 | BC-SRV-COM-FTP | saphttp and SSL - client ciphersuites configuration |
| 2284059 | BC-JAS-SEC-CPG | Update of SSL library within NW Java server |
| 2253695 | BC-IAM-SL | Fixes in CommonCryptoLib 8.4.48 |
| 2181733 | BC-IAM-SL | Fixes and Features in CommonCryptoLib 8.4.38 |
| 2124480 | BC-CST-IC | ICM / Web Dispatcher: TLS Extension Server Name Indication (SNI) as client |
| 2083594 | BC-CST | SAP Kernel Versions and SAP Kernel Patch Levels |
| 2004653 | BC-IAM-SSO-CCL | CommonCryptoLib 8 cryptographic algorithms |
| 1901252 | XX-CSC-PT-EIN | PT: WS - Online communication to AT:Solution Details |
| 1901250 | XX-CSC-PT-EIN | PT: WS - Online communication to AT : Technical Req |
| 1896961 | OPU-GW-COR | HTTP/HTTPS Configuration for SAP NetWeaver Gateway |

| | | |
|---|---|---|
| 1872926 | XX-CSC-PT-EIN | Obsolete Note: PT: WS - Web Service: LC Online communication with Tax Authorities |
| 1848999 | BC-IAM-SSO-CCL | Central Note for CommonCryptoLib 8 (SAPCRYPTOLIB) |
| 1841573 | BC-SEC-SSL | SAPCRYPTOLIB 555pl36: bugfixes, error details, new features |
| 1744209 | BC-CST | SAP Kernel 720, 721 and 722: Versions and Kernel Patch Levels |
| 1688545 | BC-SEC | OAuth 2.0 Server in AS ABAP Troubleshooting |
| 1619442 | BC-SEC-SSL | Error when automatically reloading changed SSL PSEs |
| 1531399 | BC-SEC-SSL | Enabling SSL for Session Protection |
| 1452833 | BC-SEC-SSL | Prerequisites for analyzing support messages on STRUST |
| 1433874 | BC-SEC-SSL | SapSSLReloadCred fix, SSLv3/TLSv1.0 configurability |
| 1408879 | PY-DE-BA | ELENA: Set Up HTTP(S) Connection for Communication Server |
| 1375378 | BC-SEC | Select the right version of an SAP security toolkit |
| 1257108 | BC-SEC-LGN | Collective Note: Analyzing issues with Single Sign On (SSO) |
| 1178155 | BC-SEC-SSL | Replacing PSEs in productive SSL Servers |
| 1175193 | | |

## This document is referenced by

| SAP Note/KBA | Component | Title |
|---|---|---|
| 3209634 | | STRUST Error Message: Certificate response does not match PSE |
| 3306012 | BC-SEC-SSL | ICM Error : SSSLERR_UNSUPP_PROTOCOL_VERSION during TLS version change |
| 3228201 | SV-CLM-INF-CON | SSL handshake failed, SSSLRC_CONN_CLOSED (-10) when registering system to SAP Cloud ALM |
| 2834475 | BC-JAS-SEC-CPG | Does SAP NetWeaver AS Java support TLS 1.3? |
| 2907312 | SV-SMG-LDB | receive method failed with return code SY_SUBRC 1 in job SAP_LMDB_DOWNLOAD_CONTENT |
| 3083680 | CA-ML-IPA-S4 | SEC_E_UNTRUSTED_ROOT error occurred when call API via OS command "curl" |
| 3258627 | CA-FLP-FE-UI | List of documents: Fiori Launchpad + SAML or Web Dispatcher |
| 3256863 | SV-CLM-OP-RUM | SAP Cloud ALM T/S Real User Monitoring System Registration |
| 2513042 | BC-UPG-TLS-TLA | ERR_CONNECTION_REFUSED When accessing SUM GUI |
| 3055796 | BC-SEC-SSL | SSL 3rd-Party Scans & Vulnerabilities |
| 3128830 | BC-MID-ICF | Troubleshooting ICF errors - Guided Answers |
| 3128026 | KM-SEN-CMP | Fiori In-Application Help is missing the question mark ? (Web Assistant question mark missing) or un-editable (unknown) |
| 1792296 | BC-CST-IC | CST – Error: Operation failed (rc=1) Message no. ICM006 |
| 2942034 | BC-CST-IC | "received a fatal TLS certificate unknown alert message from the peer" |
| 3043557 | BC-CST-IC | Unable to activate HTTPS port due to missing or inaccessible PSE file |
| 3019790 | BC-SEC-SSL | Interoperability (issues) with third-party TLS implementations |
| 3019835 | BC-SEC-SSL | Customizing available TLS protocol versions (how to enable TLSv1.2) |
| 3019789 | BC-SEC-SSL | Customizing available TLS cipher suites |
| 3019808 | BC-SEC-SSF | Creating SSL client PSEs for SAP WebAS through transaction STRUST |
| 3019779 | BC-SEC-SSF | Creating SSL Server PSEs for SAP WebAS through transaction STRUST |
| 3019834 | BC-SEC-SSL | Installing & enabling the library (for SAP Netweaver prior to 74x) |
| 3015741 | BC-ESI-WS-ABA-CFG | ESI - "Received a fatal TLS handshake failure alert message from the peer" when connecting to cloud system from on premise through Webservices |
| 3014930 | BC-SEC-SSL | Enabling TLS 1.1 and 1.2 on SAP NetWeaver AS Java for inbound connections |

| | | |
|---|---|---|
| 2712590 | SV-SMG-CM | SAP Business Technology Platform and SAP Solution Manager ChaRM scenario |
| 2962555 | BC-CST-STS | SSSLERR_SSL_CONNECT error when using SAP MMC |
| 2005571 | BC-MID-ICF-LGN | Warnings on the System Logon Page when logging on to an ABAP system via HTTP |
| 2950648 | BC-CST-WDP | "Your connection to this site is not fully secure" message when accessing via SAP Web Dispatcher |
| 2854633 | BC-SEC-WSS | Web service call over HTTP protocol returns 403 Forbidden |
| 2542858 | BC-CST-WDP | "Installation of CA certificate failed" error at the Web Admin page |
| 2839658 | BC-JAS-SEC | SSL Connection to AS Java showing obsolete connection settings |
| 2821444 | BC-CP-CF-ROUTING | TLS/SSL version requirements for SAP BTP and ABAP Platform integration scenarios |
| 2368112 | BC-MID-ICF | Outgoing HTTPS connection does not work in AS ABAP |
| 2801185 | BC-ESI-WS-ABA-CFG | ESI - Issues in WSIL access during Local configuration or Central Configuration. |
| 2482807 | LOD-ANA-LDC | SAP Analytics Cloud Live connections requiring a Secure HTTPS Browser configuration with CORS enabled |
| 2540826 | MOB-UIA-LIB-AUT | Password is sent as plain text when you make a logon on SAP Fiori Launchpad |
| 2734275 | BC-JAS-SEC | Security error appears using Chrome - NET::ERR_CERT_SYMANTEC_LEGACY |
| 2746647 | BC-SEC-SSF | SSSLERR_PEER_CERT_UNTRUSTED after importing the target certificate |
| 2359837 | SV-SMG-INS-CFG-SYP | Troubleshooting for "Support Hub Connectivity" in Solution Manager 7.2 up to SP04 |
| 2170715 | CA-FE-FLP-EU | Configuring Fiori over HTTP or HTTPS |
| 2339387 | BC-SEC-SSF | Warning "There is a problem with this website's security certificate" when accessing AS ABAP via HTTPS URL |
| 2645559 | BC-SRV-KPR | HTTP error 401 Unauthorized in KPRo after activate SNC system-wide |
| 2643182 | BW-BEX-ET-WJR | Result page is not displayed when trying to broadcast a workbook |
| 2626835 | BC-SEC-SSF | Missing SSL PSE entries in STRUST transaction |
| 2618301 | OPU-GW-COR | SAP Gateway returns "Error when processing resource" for OData modifying requests (CUD) |
| 2605515 | BNS-ARI-SE-ERP | ARBERP add-on: SRT: Processing error in Internet Communication Framework: ("SSL handshake with certservice.ariba.com:443 failed") |
| 2389482 | BC-SEC-SSL | Error "SSSLERR_SSL_READ" shows up during outgoing/incoming SSL connection |
| 2265435 | BC-SEC-SSF | Intermittent "Logon not possible (error in license check)" errors during live operation |
| 2177490 | BC | SSSLERR_PEER_CERT_UNTRUSTED in upgrade phase INPUT-OS-USER-PASSWORDS |
| 2160678 | BC-CST-IC | SSO stops working when the ICM trust parameters are configured |
| 2570499 | BC-IAM-SSO-CCL | How to adjust the supported SSL cipher suites in AS ABAP |
| 2553979 | BC-ESI-WS-ABA | ESI - SOAP Web Services ABAP - Guided Answers |
| 2544309 | BC-IAM-SSO-CCL | Overview of Cryptographic Libraries |
| 1692680 | BC-FES-ITS | Security Warning: "The signature could not be read. Please contact your system administrator." occurs while up/downloading via Webgui |
| 2469949 | BC-ESI-WS-ABA-CFG | ESI - "ICF Error when creating HTTP client object by Config for URL" in Web Services ABAP |
| 2203802 | BC-ESI-WS-ABA-RT | ESI - ICM_HTTP_SSL_ERROR when trying to test a Web Service |
| 1936501 | BC-ESI-WS-ABA-CFG | ESI - Transaction SOAMANAGER does not work |
| 2461900 | BC-SEC-SSL | SSSLERR_PEER_CERT_UNTRUSTED error in dev_icm trace |
| 3318423 | BC-SEC-SSF | Is TLS 1.3 Supported by SAP Kernel for Netweaver AS ABAP and S/4HANA? |
| 3311460 | PY-DE-BA | SI: Change to SHI communications server on April 3, 2023 |
| 3291446 | BC-SEC-SSL | Prevent TLS 1.3 Flag from Being Manually Activated in Incompatible Kernels |

| | | |
|---|---|---|
| 3155893 | BC-SEC-SSL | Update for built-in defaults: TLSv1.2, PFS cipher suites and client-side TLS extension SNI in legacy Netweaver Kernels |
| 3133628 | HAN-AS-XS | FAQ: SAP HANA Web Dispatcher |
| 3080055 | PY-CH | ELM 4.0: Update to TLS Version 1.2 or 1.3 until October 2021 is mandatory to keep reporting wages |
| 3066339 | BC-DB-SDB | Configure the cipher suites of the CommonCrypto Library in SAP MaxDB/liveCache |
| 2393060 | BC-INS-FWK | SAPinst Framework 749 Central Note |
| 3038972 | HAN-AS-INA-FL | Aspects concerning https protocol within fileloader scenarios |
| 3024636 | BC-SRV-COM-FTP | FAQ: SSL/TLS-related Connection failures to Archive servers and Content Servers, plus other saphttp connectivity issues |
| 2968560 | KM-SEN-CMP | SAP Enable Now Web Assistant receiving error message 503 Service Unavailable |
| 2957823 | BC-CST-WDP | NET::ERR_SSL_OBSOLETE_VERSION and similar Browser complaints about a server negotiating protocol version TLSv1.0 or TLSv1.1 |
| 2914977 | BNS-CON-SE-S4 | FAQ: Concur Certificates, Authentication, and Connectivity |
| 2923117 | BC-NEO-SEC-CPG | SAP Cloud Platform NEO – TLS 1.2 Migration - How to address problems with old TLS protocol versions in clients of SCP |
| 2846748 | PY-DE-BA | SI: HTTP error code 403 for SHI communications server |
| 2620478 | BC-UPG-OCS | Download Service: Trust anchor certificates required for software & notes downloads |
| 2775416 | BW4-UI | BW/4HANA Cockpit not working in BW/4HANA 2.0 SP0 |
| 2559867 | XX-SER-SAPSMP-SUP | LOP Security Adjustments |
| 2750539 | PY-GB | PY-GB: GB e-Filing Update HMRC Security Changes - MD5, TLS 1.0 and Credential Length |
| 2728600 | LOD-HCI-PI-RT | SSSLERR_ when accessing HCI/(S)CPI/NEO/CF servers under *.hana.ondemand.com or *.cloud.sap |
| 2463712 | SV-SMG-DIA | Diagnostics Agent TLS 1.2 |
| 2710834 | CEC-MKT-CPG-EXE | SSL handshake with 'multichannel-pp.sapmobileservices.com:443' failed: SSSLERR_CONN_CLOSED (-10) |
| 2284059 | BC-JAS-SEC-CPG | Update of SSL library within NW Java server |
| 2688393 | PY-DE-BA | SI: Switching off logs TLS 1.0 and TLS 1.1 as of December 31, 2018 |
| 2637521 | BC-IAM-SSO-CCL | Fixes and Features in CommonCryptoLib 8.5.22 |
| 2582368 | BC-SEC-SSL | SapSSL update for client-side sending of TLS extension SNI by saphttp, sapkprotp, sldreg |
| 2581510 | CA-GTF-CSC-EDO-ES | Integration flow 'eDocument: VAT Register Books for Spain': Frequent Problem Solution |
| 2554853 | BC-UPG-OCS | SAP NetWeaver download service for SAP Notes |
| 2200230 | BC-CST-STS | Problems with use of system PKI |
| 1286897 | FI-GL | Import of Exchange Rates from ECB |
| 2384290 | BC-SEC-SSL-CFG | SapSSL update to facilitate TLSv1.2-only configurations, TLSext SNI for 721+722 clients |
| 2418314 | PY-DE-BA | SI: SSL error when using communication server of pension insurance |
| 2388830 | BC-CST-IC | ICM / Web Dispatcher - HTTP/2: Browser shows error "INADEQUATE_SECURITY" |
| 2384243 | BC-IAM-SSO-CCL | NetWeaver Application Server: How to configure strict TLS 1.2 |
| 2275390 | BC-IAM-SL | Fixes in CommonCryptoLib 8.4.49 |
| 2334940 | PY-DE-BA | B2A: Troubleshooting HTTPS for communication with health insurance funds |
| 2312071 | HAN-DB-SEC | SAP HANA SSL - missing option for SSLv2 client hello |
| 2289979 | PY-DE-BA | SI: Mandatory HTTPS changeover up to 7/1/2016 (SHI communications server) |
| 2289978 | PY-DE-BA | SI: HTTPS and eXTra 1.4 mandatory changeover up to 7/1/2016 (DSRV communications server) |
| 2287896 | BC-SRV-COM-FTP | saphttp and SSL - client ciphersuites configuration |
| 2275993 | HAN-DB-SEC | SAP HANA SSL Problem analysis - SSL Record with a Non-Supported Version Received |
| 2198198 | BC-SEC-SSF | Load of SSF default product SAPSECULIB |

| | | |
|---|---|---|
| 2173384 | HAN-AS-XS | HANA XSEngine: TLS Extension "Server Name Indication" (SNI) as client |
| 2146549 | BC-IAM-SL | Fixes in CommonCryptoLib 8.4.36 |
| 2148808 | BW-WHM-DBA-MD | Starting master data maintenance for InfoObjects using HTTPS |
| 2103200 | BC-FES-JNT | JNet/JGantt with Java 8 - ClassNotFoundException: 'com.sap.tc.webdynpro.ace.AcfApplet' or 'com.sap.jnet.JNetAppletSAPGUI.class' |
| 1848999 | BC-IAM-SSO-CCL | Central Note for CommonCryptoLib 8 (SAPCRYPTOLIB) |
| 2110020 | BC-SEC-SSL | Enabling TLS or disabling SSLv3 protocol versions on SAP WebDispatcher, or SAP WebAS (AS ABAP 6xx, 7xx or AS Java >= 710) |
| 2065806 | BC-IAM-SL | Fixes and Features in CommonCryptoLib 8.4.31 |
| 1529546 | XX-PART-MFS-QUC-ADP | Troubleshooting note for QC Enterprise Integration issues |
| 1920429 | XX-CSC-PT-EIN | PT: WS - process hist. doc., global delivery, BOM |
| 1422864 | XX-PART-CGS | CGsprint 1.x: Installation or upgrade |
| 758667 | BC-OP-AS4 | IBM i: Installing Sapcrypto Library |
| 1901252 | XX-CSC-PT-EIN | PT: WS - Online communication to AT:Solution Details |
| 1636252 | BC-CST | Installing a 7.20 kernel in SAP Web AS 7.00/7.01/7.10/7.11 |
| 1901250 | XX-CSC-PT-EIN | PT: WS - Online communication to AT : Technical Req |
| 455033 | BC-SEC-SNC | SAPCRYPTOLIB versions, bugs and fixes |
| 1896961 | OPU-GW-COR | HTTP/HTTPS Configuration for SAP NetWeaver Gateway |
| 1841573 | BC-SEC-SSL | SAPCRYPTOLIB 555pl36: bugfixes, error details, new features |
| 1688545 | BC-SEC | OAuth 2.0 Server in AS ABAP Troubleshooting |
| 1844549 | XX-PART-CGS | CGsprint 2.x version 200_702: Installation / upgrade (obsolete) |
| 1257108 | BC-SEC-LGN | Collective Note: Analyzing issues with Single Sign On (SSO) |
| 1178155 | BC-SEC-SSL | Replacing PSEs in productive SSL Servers |
| 1433874 | BC-SEC-SSL | SapSSLReloadCred fix, SSLv3/TLSv1.0 configurability |
| 965076 | BC-FES-IGS | Using HTTPS with the IGS |
| 1619442 | BC-SEC-SSL | Error when automatically reloading changed SSL PSEs |
| 1408879 | PY-DE-BA | ELENA: Set Up HTTP(S) Connection for Communication Server |
| 397175 | BC-SEC | SAP Cryptographic software - export control |
| 698459 | BC-SEC-SSF | Trust manager: New root certificates |
| 1452833 | BC-SEC-SSL | Prerequisites for analyzing support messages on STRUST |
| 1375378 | BC-SEC | Select the right version of an SAP security toolkit |
| 354819 | BC-SEC-SSF | Collective note SAPSECULIB |
| 662340 | BC-SEC-SSF | SSF Encryption Using the SAPCryptolib |
| 578377 | BC-SEC-SSF | Digital signatures with SAPCRYPTOLIB |
| 834039 | BC-SEC-SSF | Certificate extension problems, Verisign (Japan) |
| 508307 | BC-SEC-SSF | Trust Manager: Problems importing certificate responses |
| 509495 | BC-SEC-SSF | Trust Manager: PSEs carry out generation with key length DSA > 512 bits and RSA > 1024 bits |
| 700659 | SCM-TEC | Security Guide: mySAP Supply Chain Management |
| 517860 | BC-BSP | Logging on to BSP applications |
| 597959 | EP-PCT-SAP | Portal content performance on EP 5.0 SP 6 - Sammelhinweis |
| 599270 | EP-PCT-SAP | Portal Content performance - composite SAP Note |