

Transport Layer Security (TLS)

General Settings

From this page you can configure security transport options for your system.

TLS, Transport Layer Security, seamlessly ensures transport email encryption for email delivery over SMTP between supporting servers.

The screenshot displays the Libraesva Email Security web interface. At the top, the logo 'LIBRAESVA EMAIL SECURITY' is visible. The navigation bar includes links for Home, System, Reports, Quarantine, Social Graph, and Search. Below this, a secondary bar shows various system components: Appliance, Mail Transport (selected), Content Analysis, Authentication, and High Availability. The main content area is titled 'Transport Layer Security Configuration' and contains a sub-header 'General Settings'. Under 'General Settings', there is a checkbox for 'TLS ACTIVE' which is currently checked, and a 'Disable' button. Below this, the 'TLS Settings' section includes dropdown menus for 'Activity Logging' (set to Normal), 'SMTP Supported Ciphers' (set to High Compatibility), and 'SMTP Server Policy' (set to may). A 'Save' button is located next to the SMTP Server Policy dropdown. Further down, a section titled 'Per-destination SMTP/TLS client policy maps:' contains a table with columns for 'Destination', 'Policy', and 'Attribute'. The table is currently empty, displaying 'No records found'. At the bottom of the table, there are navigation controls including a search icon, first, previous, next, last, and refresh buttons.

Libraesva ESG acts both for receiving and sending emails, so TLS can be configured on both sides: you can define different policies on receive (server) or when sending (client).

All TLS policies can be DISABLED by clicking **Disable** button.

Server Policies

You can choose from one of the followings:

- **none** (No TLS when receiving emails)
- **may** (Try to negotiate a TLS connection)
- **encrypt** (Force a TLS connection when receiving an email and if it fails do not accept incoming mail)

Default policy is may, that means try to negotiate a secure TLS connection but continue receiving the message even if it fails.

×WARNING: According to RFC 2487 encryption MUST NOT be enforced in case of a publicly-referenced SMTP server. You will reject legit email traffic in doing this.

SMTP Supported Ciphers

This setting allows to restrict the available ciphers and protocols.

×WARNING: For a publicly-referenced SMTP server the default is to fallback to plaintext if no shared combination of cipher/protocol can be negotiated with the remote party. This means that by restricting the list of ciphers there are higher chances of falling back to plaintext. The default is “**Medium**” and this is the suggested setting. Libraesva ESG will keep the list of available ciphers updated accordingly to our interpretation of the best balance between cipher strength and their worldwide adoption.

Strict

The “**Strict**” setting is the most restrictive and provides **PCI DSS** compliance.

Supported protocols are **TLSv1.3** and **TLSv1.2**. Diffie-Hellmann parameter size is 2048bits, supports P-256 elliptic curve and provides protection against Poodle, Goldendoodle, Zombie Poodle, Sleeping Poodle, Heartbleed and more.

- **TLS 1.3:**
 - TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519)
 - TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519)
 - TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519)

- TLS_AKE_WITH_AES_128_CCM_SHA256 (ecd_h_x25519)

- **TLS 1.2:**

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048)
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (dh 2048)
- TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048)
- TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048)
- TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 4096)
- TLS_RSA_WITH_AES_256_CCM_8 (rsa 4096)
- TLS_RSA_WITH_AES_256_CCM (rsa 4096)
- TLS_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 4096)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_256_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048)
- TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048)
- TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048)
- TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecd_h_x25519)

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CCM_8 (rsa 4096)
- TLS_RSA_WITH_AES_128_CCM (rsa 4096)
- TLS_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CBC_SHA (rsa 4096)
- **TLS 1.1:** disabled
- **TLS 1.0:** disabled
- **Extra restrictions:**
 - Disable plain SMTP authentication
 - Disable renegotiation

Medium (default)

The “**Medium**” setting is less restrictive than “High” and provides broader compatibility. Supported protocols are **TLSv1.3** and **TLSv1.2**. Diffie-Hellmann parameter size is 2048bits, supports P-256 elliptic curve and provides protection against Poodle, Goldendoodle, Zombie Poodle, Sleeping Poodle, Heartbleed and more.

- **TLS 1.3:**
 - TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519)
 - TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519)
 - TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519)
 - TLS_AKE_WITH_AES_128_CCM_SHA256 (ecdh_x25519)
- **TLS 1.2:**
 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519)
 - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048)
 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519)
 - TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (dh 2048)

- TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048)
- TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048)
- TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (ecdh_x25519)
- TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (ecdh_x25519)
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048)
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 4096)
- TLS_RSA_WITH_AES_256_CCM_8 (rsa 4096)
- TLS_RSA_WITH_AES_256_CCM (rsa 4096)
- TLS_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 4096)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_256_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048)
- TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048)
- TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048)
- TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (dh 2048)

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048)
- TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CCM_8 (rsa 4096)
- TLS_RSA_WITH_AES_128_CCM (rsa 4096)
- TLS_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 4096)
- TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 2048)
- TLS_RSA_WITH_SEED_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 2048)
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 4096)
- **TLS 1.1:** disabled
- **TLS 1.0:** disabled
- **Extra restrictions:**
 - Disable plain SMTP authentication

High Compatibility

The “**High compatibility**” setting is less restrictive than “Medium”, and provides as many ciphers and version as possible, at the expense of some of some security strenght.

Supported protocols are **TLSv1.3**, **TLSv1.2**, **TLSv1.1** and **TLSv1.0**. Diffie-Hellmann parameter size is 2048bits, supports P-256 elliptic curve and provides protection against Poodle, Goldendoodle, Zombie Poodle, Sleeping Poodle, Heartbleed and more

- **TLS 1.3:**

- TLS_AKE_WITH_AES_256_GCM_SHA384 (ecd_h_x25519)
- TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecd_h_x25519)
- TLS_AKE_WITH_AES_128_GCM_SHA256 (ecd_h_x25519)
- TLS_AKE_WITH_AES_128_CCM_SHA256 (ecd_h_x25519)

◦ **TLS 1.2:**

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048)
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (dh 2048)
- TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048)
- TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048)
- TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (ecd_h_x25519)
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecd_h_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048)
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 4096)
- TLS_RSA_WITH_AES_256_CCM_8 (rsa 4096)
- TLS_RSA_WITH_AES_256_CCM (rsa 4096)
- TLS_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 4096)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_256_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecd_h_x25519)

- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048)
- TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048)
- TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048)
- TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (ecdh_x25519)
- TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (dh 2048)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048)
- TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CCM_8 (rsa 4096)
- TLS_RSA_WITH_AES_128_CCM (rsa 4096)
- TLS_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 4096)
- TLS_RSA_WITH_AES_128_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 4096)
- TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 2048)
- TLS_RSA_WITH_SEED_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 2048)
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 4096)

◦ **TLS 1.1:**

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048)
- TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048)

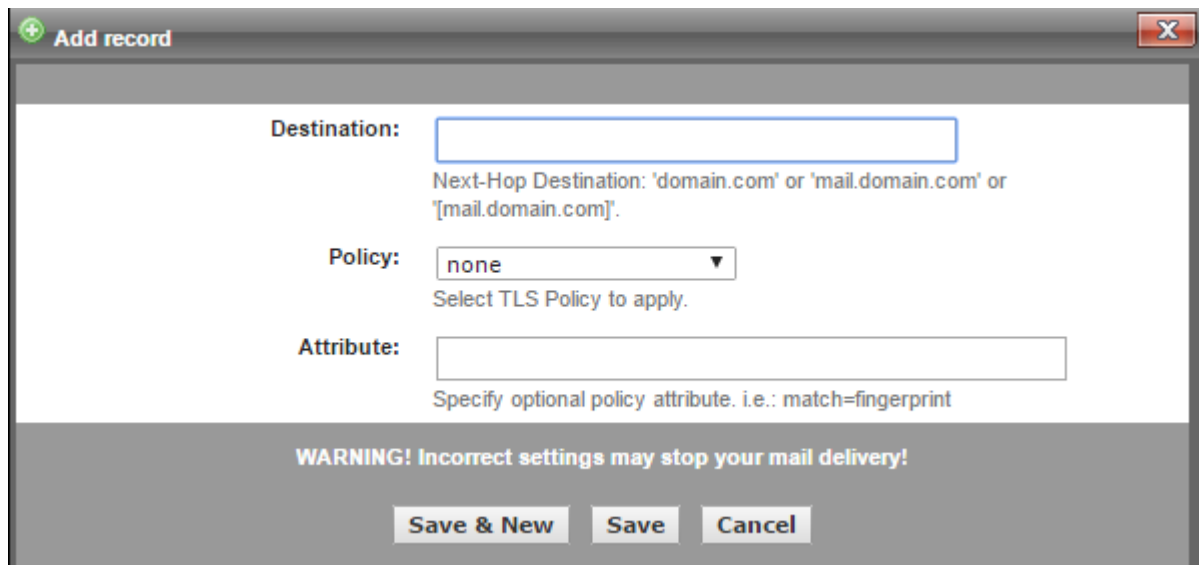
- TLS_RSA_WITH_AES_256_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048)
- TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048)
- TLS_RSA_WITH_AES_128_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 4096)
- TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 2048)
- TLS_RSA_WITH_SEED_CBC_SHA (rsa 4096)
- TLS_RSA_WITH_IDEA_CBC_SHA (rsa 4096)
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (ecdh_x25519)
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 2048)
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 4096)
- **TLS 1.0:**
 - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519)
 - TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048)
 - TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048)
 - TLS_RSA_WITH_AES_256_CBC_SHA (rsa 4096)
 - TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 4096)
 - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519)
 - TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048)
 - TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048)
 - TLS_RSA_WITH_AES_128_CBC_SHA (rsa 4096)
 - TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 4096)
 - TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 2048)
 - TLS_RSA_WITH_SEED_CBC_SHA (rsa 4096)
 - TLS_RSA_WITH_IDEA_CBC_SHA (rsa 4096)
 - TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (ecdh_x25519)
 - TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 2048)

- TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 4096)

Client Policies

You can define per destination TLS Policies.

Press New and fill in the information required:



The screenshot shows a dialog box titled "Add record" with a green plus icon and a red close button. It contains three main fields: "Destination:" with a text input box and a note below it stating "Next-Hop Destination: 'domain.com' or 'mail.domain.com' or '[mail.domain.com]'"; "Policy:" with a dropdown menu currently set to "none" and a note below it stating "Select TLS Policy to apply."; and "Attribute:" with a text input box and a note below it stating "Specify optional policy attribute. i.e.: match=fingerprint". At the bottom, there is a warning message: "WARNING! Incorrect settings may stop your mail delivery!". Below the warning are three buttons: "Save & New", "Save", and "Cancel".

- **Destination:** enter next-hop destination. For example a domain, or a mail server or an IP address. When specifying a mail server or an IP include it in square brackets [1.2.3.4]
- **Policy:** select one of the followings:
 - **None:** no TLS. No additional attributes are supported at this level
 - **May:** opportunistic TLS. The optional “ciphers”, “exclude” and “protocols” attributes are available
 - **Encrypt:** mandatory encryption. Mail is delivered only if the remote SMTP server offers STARTTLS and the TLS handshake succeeds. At this level and higher, the optional “protocols”, “ciphers” and “exclude” attributes are available
 - **Dane:** mandatory encryption. Mail is delivered only if the remote SMTP server offers STARTTLS and the TLS handshake succeeds. At this level and higher, the optional “protocols”, “ciphers” and “exclude” attributes are available
 - **Dane-only:** mandatory DANE TLS. The TLS policy for the destination is obtained via TLSA records in DNSSEC. If no TLSA records are found, or none are usable, no connection is made to the server

- **Fingerprint:** certificate fingerprint verification. At this security level, there are no trusted certificate authorities. The certificate trust chain, expiration date, ... are not checked. Instead, the optional “match” attribute lists the server certificate fingerprints or public key fingerprints. The digest algorithm used to calculate fingerprints is MD5. Multiple fingerprints can be combined with a “|” delimiter in a single match attribute
- **Verify:** mandatory server certificate verification. Mail is delivered only if the TLS handshake succeeds, if the remote SMTP server certificate can be validated (not expired or revoked, and signed by a trusted certificate authority), and if the server certificate name matches the optional “match” attribute
- **Secure:** secure-channel TLS. At this security level, DNS MX lookups, though potentially used to determine the candidate next-hop gateway IP addresses, are not trusted to be secure enough for TLS peer name verification. Instead, the default name verified in the server certificate is obtained directly from the next-hop, or is explicitly specified via the optional match attribute. The match attribute is most useful when multiple domains are supported by common server, the policy entries for additional domains specify matching rules for the primary domain certificate. While transport table overrides routing the secondary domains to the primary next hop also allow secure verification, they risk delivery to the wrong destination when domains change hands or are re-assigned to new gateways. With the “match” attribute approach, routing is not perturbed, and mail is deferred if verification of a new MX host fails

×**NOTE:** The digest algorithm used to calculate and verify certificates fingerprints is MD5

Default Certificate

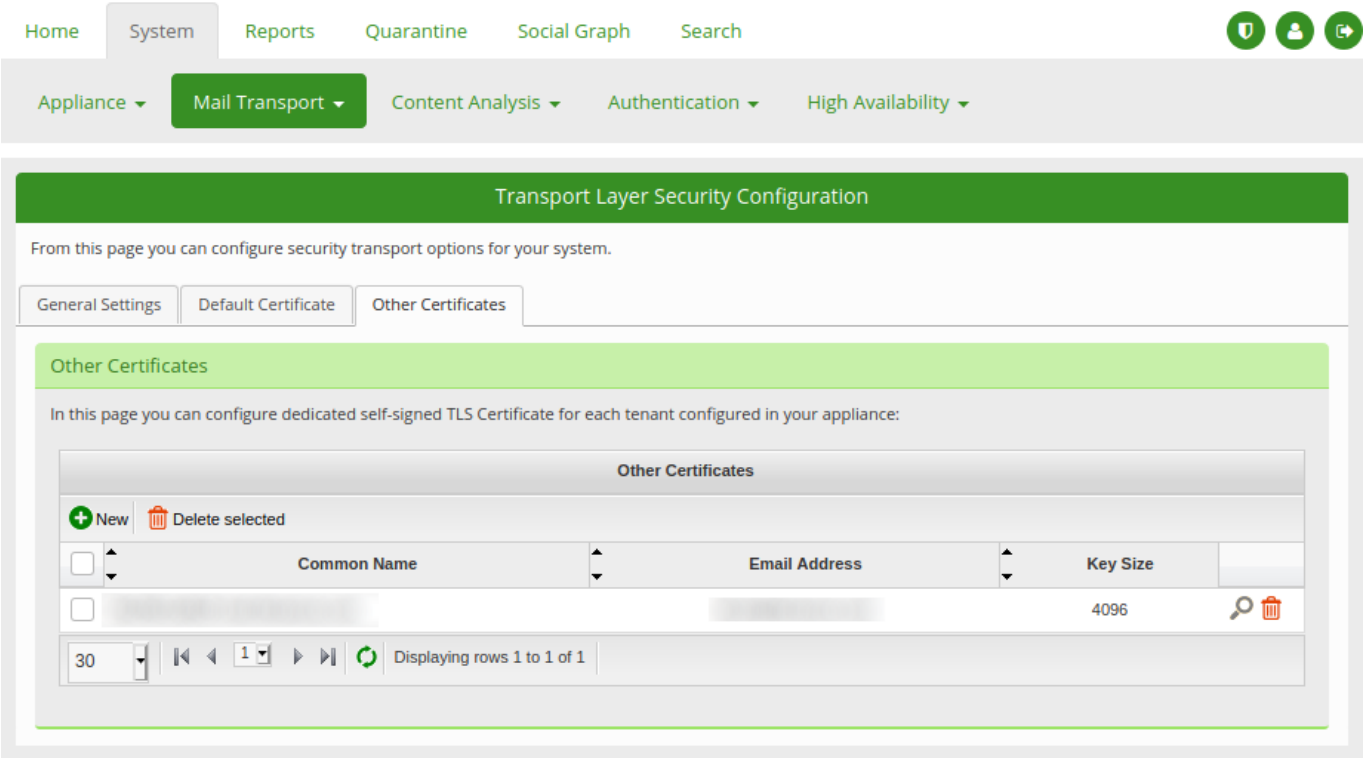
From the Default Certificate Tab you can manage the **TLS certificates** valid for SMTP. You can perform the following actions:

- Read your installed certificate fingerprint
- Use the Let's Encrypt certificate generated for HTTPS
- Generate a new self-signed certificate
- Generate a new certificate request to submit to your CA
- Upload your certificate once emitted starting from your request

×WARNING: We recommend to enable Let's Encrypt TLS certificate in your Libraesva ESG, or to use trusted certificate. Self-signed certificates should be avoided.

Other Certificates

Starting from Libraesva ESG 4.9 you can generate dedicated self-signed certificates. This certificates can be used for relay to internal domain (e.g Microsoft 365 domain) with a dedicated TLS Certificate instead the default one.



To create a new certificate, click the **New** button and modify the pre-filled fields with the desired values

Generate self-signed certificate

New TLS Self-Signed Certificate

Country Code:

IT

Your two-letter IANA country code

Locality Name:

Europe

Your town or city (NO SPACES)

Organization Name:

Libraesva Srl

Your organisation name

Organization Unit:

IT

A dot (period) is usually appropriate here (NO SPACES)

Common Name:

60005c295ce5d-esg-libra-srl

Tenant identifier. Usually replace . (dot) with - (dash) is appropriate. e.g.: yourdomain-tld

Email Address:

admin@libraesva.com

A valid email address that will be shown into the certificate

Key Size:

4096

The size of the encryption key

Generate

Cancel

Country Code	Your two-letter IANA country code
Locality Name	Your town or city (NO SPACES)
Organization Name	Your organisation name
Organization Unit	A dot (period) is usually appropriate here (NO SPACES)
Common Name	Tenant identifier. Usually replace . (dot) with - (dash) is appropriate. e.g.: yourdomain-tld
Email Address	A valid email address that will be shown into the certificate
Key Size	The size of the encryption key

Click **Generate**