Fix Me! Esta página não está totalmente traduzida. Qualquer ajuda para completar a tradução é bem-vinda.
(remover este parágrafo assim que a tradução estiver finalizada)

16. Criptografia

Visão geral

Zabbix supports encrypted communications between Zabbix server, Zabbix proxy, Zabbix agent, zabbix_sender and zabbix_get utilities using Transport Layer Security (TLS) protocol v.1.2. Encryption is supported starting with Zabbix 3.0. Certificate-based and pre-shared key-based encryption is supported.

Encryption is optional and configurable for individual components (e.g. some proxies and agents can be configured to use certificate-based encryption with the server, while others can use pre-shared key-based encryption, and yet others continue with unencrypted communications as before).

Server (proxy) can use different encryption configurations for different hosts.

Zabbix daemon programs use one listening port for encrypted and unencrypted incoming connections. Adding an encryption does not require opening new ports on firewalls.

Compiling Zabbix with encryption support

To support encryption Zabbix must be compiled and linked with one of three crypto libraries:

- mbed TLS (formerly PolarSSL)(version 1.3.9 and later, but mbed TLS 2.x is not currently supported)
- GnuTLS (from version 3.1.18)
- OpenSSL (from version 1.0.1)

The library is selected by specifying an option to “configure” script:

- `--with-mbedtls[=DIR]`
- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]`

For example, to configure the sources for server and agent with OpenSSL you may run something like:

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --
with-net-snmp --with-libcurl --with-libxml2 --with-openssl
```

Different Zabbix components may be compiled with different crypto libraries (e.g. a server with OpenSSL, an agent with GnuTLS).

If you plan to use pre-shared keys (PSK) consider using GnuTLS or mbed TLS libraries in Zabbix components using PSKs. GnuTLS and mbed TLS libraries support PSK ciphersuites with Perfect Forward Secrecy. OpenSSL library (versions 1.0.1, 1.0.2c) does support PSKs but available

ciphersuites do not provide Perfect Forward Secrecy.

Connection encryption management

Connections in Zabbix can use:

- no encryption (default)
- PSK-based encryption
- certificate-based encryption.

There are two important parameters used to specify encryption for connections between Zabbix components:

- TLSConnect
- TLSAccept

TLSConnect specifies what encryption to use for outgoing connections and can take one of 3 values (unencrypted, PSK, certificate). TLSConnect is used in configuration files for Zabbix proxy (in active mode) and Zabbix agentd (for active checks). In Zabbix frontend the TLSConnect equivalent is “Connections to host” field in “Configuration→Hosts→<some host>→Encryption tab” and “Connections to proxy” field in “Administration→Proxies→<some proxy>→Encryption” tab. If the configured encryption type for connection fails, no other encryption types will be tried.

TLSAccept specifies what types of connections are allowed for incoming connections and can take any combination of 3 values (unencrypted, PSK, certificate). TLSAccept is used in configuration files for Zabbix proxy (in passive mode) and Zabbix agentd (for passive checks). In Zabbix frontend the TLSAccept equivalent is “Connections from host” field in “Configuration→Hosts→<some host>→Encryption” tab and “Connections from proxy” field in “Administration→Proxies→<some proxy>→Encryption” tab.

Normally you configure only one type of encryption for incoming encryptions. But you may want to switch encryption type, e.g. from unencrypted to certificate-based with minimum downtime and rollback possibility. To achieve this you can set TLSAccept=unencrypted,cert in agentd configuration file and restart Zabbix agent. Then you can test connection with zabbix_get to the agent using certificate. If it works, you can reconfigure encryption for that agent in Zabbix frontend in “Configuration→Hosts→<some host>→Encryption tab” by setting “Connections to host” to “Certificate”. When server configuration cache gets updated (and proxy configuration is updated if the host is monitoring by proxy) then connections to that agent will be encrypted. If everything works as expected you can set TLSAccept=cert in agent configuration file and restart Zabbix agent. Now the agent will be accepting only encrypted certificate-based connections. Unencrypted and PSK-based connections will be rejected.

In a similar way it works on server and proxy. If in Zabbix frontend in host configuration “Connections from host” is set to “Certificate” then only certificate-based encrypted connections will be accepted from agent (active checks) and zabbix_sender (trapper items).

Most likely you will configure incoming and outgoing connections to use the same encryption type or no encryption at all. But technically it is possible to configure it asymmetrically, e.g. certificate-based encryption for incoming and PSK-based for outgoing connections.

For overview, encryption configuration for each host is displayed in Zabbix frontend

“Configuration→Hosts” on the right side, in column “ENCRYPTION (IN / OUT)”.

Default is unencrypted connections. Encryption must be configured for each host and proxy individually.

Using certificates

Zabbix can use certificates in PEM format, signed by a certificate authority (CA). Certificate verification is done against pre-installed CA certificate. Self-signed certificates are not supported. Optionally a certificate revocation lists (CRL) can be used. Each Zabbix component can have only one certificate configured. Choosing between multiple certificates is not supported.

For more information how to set up and operate internal CA, how to generate certificate requests and sign them, how to revoke certificates you can find numerous how-to's in internet, for example, OpenSSL PKI Tutorial v1.1

Certificate configuration parameters

Parameter	Mandatory	Description
TLSCAFile	*	Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification. In case of certificate chain with several members they must be ordered: lower level CA certificates first followed by certificates of higher level CA(s). Certificates from multiple CA(s) can be included in a single file.
TLSCRLFile		Full pathname of a file containing revoked certificates. If the file defined by TLSCAFile parameter contains several CAs and Zabbix component is compiled with OpenSSL and TLSCRLFile is defined then every CA mentioned in TLSCAFile must have a corresponding CRL (it can be empty CRL) in TLSCRLFile.
TLSCertFile	*	Full pathname of a file containing certificate (certificate chain).
TLSKeyFile	*	Full pathname of a file containing private key. Set access rights to this file - it must be readable only by Zabbix user.
TLSServerCertIssuer		Allowed server certificate issuer.
TLSServerCertSubject		Allowed server certificate subject.

Using pre-shared keys (PSK)

In Zabbix each PSK actually is a pair of:

- non-secret PSK identity string,
- secret string (PSK value).

PSK identity string is a non-empty UTF-8 string. For example, “PSK ID 001 Zabbix agentd”. It is a unique name by which this specific PSK is referred to by Zabbix components. Do not put sensitive information in PSK identity string - it is transmitted unencrypted over network.

PSK value is a hard to guess string of hexadecimal digits, for example, “e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9”.

There are maximum size limits for PSK identity and value in Zabbix, in some cases a crypto library

can have lower limit:

Component	Max PSK identity size	Max PSK value size
Zabbix	128 UTF-8 characters	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)
GnuTLS	128 bytes (may include UTF-8 characters)	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)
mbed TLS (PolarSSL)	128 UTF-8 characters	256-bit (default limit) (32-byte PSK, entered as 64 hexadecimal digits)
OpenSSL	127 bytes (may include UTF-8 characters)	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)

Zabbix frontend allows configuring up to 128-character long PSK identity string and 2048-bit long PSK regardless of crypto libraries used. If some Zabbix components support lower limits it is a user responsibility to configure PSK identity and value with allowed length for these components. Exceeding length limits results in communication failures between Zabbix components.

Before Zabbix server connects to agent using PSK, the server looks up the PSK identity and PSK value configured for that agent in database (actually in configuration cache). Upon receiving a connection the agent uses PSK identity and PSK value from its configuration file. If both parties have the same PSK identity string and PSK value the connection may succeed.

It is a user responsibility to ensure that there are no two PSKs with the same identity string but different values. Failing to do so may lead to unpredictable disruptions of communication between Zabbix components using PSKs with this PSK identity string.

Configuring PSK for server-agent communication (basic example)

On agent host write PSK value into file, for example, `/home/zabbix/zabbix_agentd.psk`. The file must contain PSK in the first text string, for example:

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

Set access rights to PSK file - it must be readable only by Zabbix user.

Edit TLS parameters in agent configuration file `zabbix_agentd.conf`, for example, set:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

The agent will connect to server (active checks) and accept from server and `zabix_get` only connections using PSK. PSK identity will be “PSK 001”.

Restart the agent. Now you can test a connection using `zabbix_get`, for example:

```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --
tls-psk-identity="PSK 001" --tls-psk-file=/home/zabbix/zabbix_agentd.psk
```

(To minimize downtime see how to change connection type in Connection encryption management).

Configure PSK for this host in Zabbix frontend. Go to “Configuration→Hosts”, select the host, go to “Encryption” tab. Set “Connections to host” to PSK. In “Connections from host” mark PSK. Paste into “PSK identity” field “PSK 001” and “1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952” into “PSK” field. Click “Update”.

When configuration cache is synchronized with database the new connections will use PSK. Check server and agent logfiles for error messages.

Configuring PSK for server - active proxy communication (basic example)

On proxy write PSK value into file, for example, `/home/zabbix/zabbix_proxy.psk`. The file must contain PSK in the first text string, for example:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Set access rights to PSK file - it must be readable only by Zabbix user.

Edit TLS parameters in proxy configuration file `zabbix_proxy.conf`, for example, set:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

The proxy will connect to server using PSK. PSK identity will be “PSK 002”.

(To minimize downtime see how to change connection type in Connection encryption management).

Configure PSK for this proxy in Zabbix frontend. Go to “Administration→Proxies”, select the proxy, go to “Encryption” tab. In “Connections from proxy” mark PSK. Paste into “PSK identity” field “PSK 002” and “e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9” into “PSK” field. Click “Update”.

Restart proxy. It will start using PSK-based encrypted connections to server. Check server and proxy logfiles for error messages.

For passive proxy the procedure is very similar. The only difference - set `TLSAccept=psk` in proxy configuration file and set “Connections to proxy” in Zabix frontend to PSK.

Limitations

- Private keys are stored in plain text in files readable by Zabbix components during startup.
- Pre-shared keys are entered in Zabbix frontend and stored in Zabbix database in plain text.
- Encryption does not protect communications:
  - between web server running Zabbix frontend and user web browser,
  - between Zabbix server (proxy) and Zabbix database.

Troubleshooting

`gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.` in TLS client side log.
`gnutls_handshake() failed: -90 The SRP username supplied is illegal.` in TLS server side log.

Possible cause: PSK identity string longer than 128 bytes is passed to GnuTLS.

`ssl_set_psk(): SSL - Bad input parameters to function`

Possible cause: PSK longer than 32 bytes is passed to mbed TLS (PolarSSL)