tsm security

Use the `tsm security` commands to configure [Tableau Server] support for external (gateway) SSL or repository (Postgres) SSL. Repository SSL configuration includes the option to enable SSL over direct connections from Tableau clients---including Tableau Desktop, Tableau Mobile, and web browsers---to the repository.

- tsm security custom-cert
  - tsm security custom-cert add{.MCXref .xref}
  - tsm security custom-cert delete{.MCXref .xref}
  - tsm security custom-cert list{.MCXref .xref}
- tsm security external-ssl
  - tsm security external-ssl disable{.MCXref .xref}
  - tsm security external-ssl enable{.MCXref .xref}
  - tsm security external-ssl list{.MCXref .xref}
- tsm security kms
  - tsm security kms set-mode aws{.MCXref .xref}
  - tsm security kms set-mode local{.MCXref .xref}
  - tsm security kms status{.MCXref .xref}
- tsm security maestro-rserve-ssl
  - tsm security maestro-rserve-ssl disable{.MCXref .xref}
  - tsm security maestro-rserve-ssl enable{.MCXref .xref}
- tsm security maestro-tabpy-ssl
  - tsm security maestro-tabpy-ssl disable{.MCXref .xref}
  - tsm security maestro-tabpy-ssl enable{.MCXref .xref}
- tsm security regenerate-internal-tokens{.MCXref .xref}
- tsm security repository-ssl
  - tsm security repository-ssl disable{.MCXref .xref}
  - tsm security repository-ssl enable{.MCXref .xref}
  - tsm security repository-ssl get-certificate-file{.MCXref .xref}
  - tsm security repository-ssl list{.MCXref .xref}
- As of the 2020.2 release, to configure Rserve and TabPy analytics extensions, use the Tableau Server admin pages. See Configure Connection with Analytics Extensions

Prerequisites

Before you configure SSL, you must acquire certificates, and then copy them to the computer that runs the [Tableau Server] gateway process. Additional preparation is required for enabling direct connections from clients. To learn more, see the following articles:

Configure SSL for External HTTP Traffic to and from Tableau Server

Configure SSL for Internal Postgres Communication

For information about mutual (two-way) SSL, see Configure Mutual SSL Authentication and tsm authentication mutual-ssl commands.

tsm security custom-cert add

Adds a custom CA certificate to [Tableau Server]{.VariablesTabsProductServer}. This certificate is optionally used to establish trust for TLS communication between a SMTP server and Tableau Server.

If a custom certificate already exists, this command will fail. You can remove the existing custom certificate using the `tsm security custom-cert delete` command.

Note: The certificate that you add with this command may be used by other Tableau Server services for TLS connections.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate file in a safe location off of the Tableau Server. The certificate file that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the file is not stored in a recoverable format. See Tableau Server Client File Service.

Synopsis

```
tsm security custom-cert add --cert-file <file.crt> [global options]
```

Options

-c, --cert-file <file.crt>

Required. Specify the name of a certificate file in valid PEM or DER format.

tsm security custom-cert delete

Removes the server's existing custom certificate. Doing this allows you to add a new custom certificate.

Synopsis

```
tsm security custom-cert delete[global options]
```

tsm security custom-cert list

List details of custom certificate.

Synopsis

```
tsm security custom-cert list[global options]
```

tsm security external-ssl disable

Removes the server's existing SSL configuration settings and stops encrypting traffic between external clients and the server.

Synopsis

```
tsm security external-ssl disable [global options]
```

tsm security external-ssl enable

Enable and specify certificate and key files for SSL over external HTTP communication.

Synopsis

```
tsm security external-ssl enable --cert-file <file.crt> --key-file <file.key> [options] [global options]
```

Options

--cert-file <file.crt>

Required. Specify the name of a valid PEM-encoded x509 certificate with the extension .crt.

--key-file <file.key>

Required. Specify a valid RSA or DSA private key file, with the extension .key by convention.

--chain-file <chainfile.crt>

Specify the certificate chain file (.crt)

A certificate chain file is required for Tableau Desktop on the Mac. In some cases, a certificate chain file may be required for Tableau Mobile.

Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate.

All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

--passphrase

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

Note: If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

--protocols <list protocols>

Optional. List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

TLS is an improved version of SSL. Tableau Server uses TLS to authenticate and encrypt connections. Accepted values include protocol versions supported by Apache. To disallow a protocol, prepend the protocol version with a minus (-) character.

Default setting: `"all, -SSLv2, -SSLv3"`

This default explicitly does not allow clients to use SSL v2 or SSL v3 protocols to connect to [Tableau Server]. However, we recommend that you also disallow TLS v1 and TLS v1.1.

Before you deny a specific version of TLS, verify that the browsers from which your users connect to [Tableau Server] support TLS v1.2. You might need to preserve support for TLSv1.1 until browsers are updated.

If you do not need to support TLS v1 or v1.1, use the following command to allow TLS v1.2 (using the value `all`), and explicitly deny SSL v2, SSL v3, TLS v1, and TLS v1.1.

```
tsm security external-ssl enable --cert-file file.crt --key-file file.key --protocols "all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
```

tsm security external-ssl list

Displays a list of settings related to the configuration of gateway external SSL. The list includes the names of the certificate files in use, but not their location.

Synopsis

```
tsm security external-ssl list [global options]
```

tsm security kms set-mode aws

Set the KMS mode to AWS.

You will need the full ARN string from AWS KMS. This string is in the "General configuration" section of the AWS KMS management pages. The ARN is presented in this format: arn:aws:kms:<region>:<account>:key/<CMK_ID>, for example, `arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-110mn1234567`.

For more information, see Key Management System.

Synopsis

```
tsm security kms set-mode aws --key-arn "<arn>" --aws-region "<region>" [global options]
```

Options

--key-arn

Required. The `--key-arn` option takes a direct string copy from the ARN in the "General configuration" section of the AWS KMS management pages.

--aws-region

Required. Specify a region as shown in the Region column in the Amazon API Gateway table[(Link opens in a new window)].

Example

For example, if your AWS KMS instance is running in us-west-2 region, your account number is 867530990073, and your CMK key is 1abc23de-fg45-6hij-7k89-1l0mn1234567, then the command would be as follows:

```
tsm security kms set-mode aws --aws-region "us-west-2" --key-arn "arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-1l0mn1234567"
```

tsm security kms set-mode local

Set the KMS mode to local. Local is the default KMS mode. For more information, see Key Management System.

Synopsis

```
tsm security kms set-mode local [global options]
```

tsm security kms status

View the status of KMS configuration.

The following is returned:

- The ARN (ID) of the customer master key (CMK) .
- The region the CMK is in.
- The ID of the root master key (RMK) in use. The RMK is a key that is encrypted by the CMK. Tableau Server decrypts the CMK by making calls to AWS KMS. The RMK is then used to encrypt/decrypt the master extract key (MEK). The RMK can change, but there will be only one at a time.
- KMS stores a collection of master extract keys (MEKs). Each MEK has:
  - An ID, for example, 8ddd70df-be67-4dbf-9c35-1f0aa2421521
  - Either a "encrypt or decrypt key" or "decrypt-only key" status. If a key is "encrypt or decrypt", Tableau Server will encrypt new data with it. Otherwise, the key will only be used for decryption
  - A creation timestamp, for example, "Created at: 2019-05-29T23:46:54Z."
  - First transition to encrypt and decrypt: a timestamp indicating when the key became an encrypt or decrypt key.
  - Transition to decrypt-only: a timestamp indicating when the key transitioned to decrypt-only.

Synopsis

```
tsm security kms status [global options]
```

tsm security maestro-rserve-ssl disable

Disable the Rserve connection.

For more information, see Use R (Rserve) scripts in your flow.

tsm security maestro-rserve-ssl enable

Configure a connection between an Rserve server and Tableau Server version 2019.3 or later.

For more information, see Use R (Rserve) scripts in your flow.

Synopsis

```
tsm security maestro-rserve-ssl enable --connection-type <maestro-rserve-secure | maestro-rserve> --rserve-host <Rserve IP address or host name> --rserve-port <Rserve port> --rserve-username <Rserve username> --rserve-password <Rserve password> --rserve-connect-timeout-ms <RServe connect timeout>
```

Options

--connection-type

Select `maestro-rserve-secure` to enable a secure connection or `maestro-rserve` to enable an unsecured connection. If you select `maestro-rserve-secure`, specify the certificate file path in the command line.

--rserve-host

Host

--rserve-port

Port

--rserve-username

Username

--rserve-password

Password

--rserve-connect-timeout-ms

The connect timeout in milliseconds. For example `--rserve-connect-timeout-ms 900000`.

tsm security maestro-tabpy-ssl disable

Disable the TabPy connection.

For more information, see Use Python scripts in your flow.

tsm security maestro-tabpy-ssl enable

Configure a connection between a TabPy server and Tableau Server version 2019.3 or later.

For more information, see Use Python scripts in your flow.

Synopsis

```
tsm security maestro-tabpy-ssl enable --connection-type <maestro-tabpy-secure | maestro-tabpy> --tabpy-host <TabPy IP address or host name> --tabpy-port <TabPy port> --tabpy-username <TabPy username> --tabpy-password <TabPy password> --tabpy-connect-timeout-ms <TabPy connect timeout>
```

Options

--connection-type

Select `maestro-tabpy-secure` to enable a secure connection or `maestro-tabpy` to enable an unsecured connection. If you select `maestro-tabpy-secure`, specify the certificate file -cf<certificate file path> in the command line.

--tabpy-host

Host

--tabpy-port

Port

--tabpy-username

Username

--tabpy-password

Password

--tabpy-connect-timeout-ms

The connect timeout in milliseconds. For example `--tabpy-connect-timeout-ms 900000`.

tsm security regenerate-internal-tokens

This command performs the following operations:

1. Stops Tableau Server if it is running.
2. Generates new internal SSL certificates for Postgres repository the search server.
3. Generates new passwords for all of the internally managed passwords.
4. Updates all Postgres repository passwords.
5. Generates a new encryption key for asset key management and encrypts the asset key data with the new key.
6. Generates a new encryption key for configuration secrets (master key) and encrypts the configuration with it.
7. Reconfigures and updates Tableau Server with all of these secrets. In a distributed deployment, this command also distributes the reconfiguration and updates across all nodes in the cluster.
8. Regenerates a new master key, adds it to the master keystore file, and then creates new security tokens for internal use.
9. Starts Tableau Server.

If you plan to add a node to your cluster after you have run this command, then you will need to generate a new node configuration file to update the tokens, keys, and secrets that are generated by this command. See Install and Configure Additional Nodes.

For more information about internal passwords see Manage Server Secrets.

Synopsis

```
tsm security regenerate-internal-tokens [options] [global options]
```

Options

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm security repository-ssl disable

Stop encrypting traffic between the repository and other server components, and stop support for direct connections from Tableau clients.

Synopsis

```
tsm security repository-ssl disable [global-options]
```

tsm security repository-ssl enable

Enables SSL and generates the server's .crt and .key files used for encrypted traffic between the Postgres repository and other server components. Enabling this also gives you the option to enable SSL over direct connections from Tableau clients to the server.

Synopsis

```
tsm security repository-ssl enable [options] [global options]
```

Options

-i, --internal-only

Optional. When set to `--internal-only`, [Tableau Server] uses SSL between the repository and other server components, and it supports but does not require SSL for direct connections through **tableau** or **readonly** users.

If this option is not set, [Tableau Server] requires SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (for connections through the **tableau** or **readonly** users).

When you specify this option, you must also complete the steps described in Configure Postgres SSL to Allow Direct Connections from Clients.

tsm security repository-ssl get-certificate-file

Get the public certificate file used for SSL communication with the Tableau repository. SSL must be enabled for repository communication before you can retrieve a certificate. The certificate file is distributed automatically to internal clients of the repository in the Tableau Server cluster. To enable remote clients to connect over SSL to the repository, you must copy the public certificate file to each client.

Synopsis

```
tsm security repository-ssl get-certificate-file [global-options]
```

Options

-f, --file

Required.

Full path and file name (with .cert extension) where the certificate file should be saved. If a duplicate file exists it will be overwritten.

tsm security repository-ssl list

Returns the existing repository (Postgres) SSL configuration.

Synopsis

```
tsm security repository-ssl list [global-options]
```

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password "my password"
```

-s, --server https://<hostname>:8850

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.