Configure TLS

Configure TLS communication from the command line. Clients and servers communicate using HTTP JSON-RPCs.

EthSigner prerequisites :

- EthSigner's password-protected PKCS #12 keystore.
- File containing the keystore password.

Client TLS connection

Allow clients (for example a dApp, or curl) to send and receive secure HTTP JSON-RPCs.

Client prerequisites :

- The client must be configured for TLS.
- Client's PKCS #12 keystore information.

Create the known clients file

Create a file (in this example,knownClients) that lists one or more clients that are trusted to connect to EthSigner. The file contents use the format where:

-
- is the Common Name used for the client's keystore
-
- is the SHA-256 fingerprint of the client's keystore.

curl_client DF:65:B8:02:08:5E:91:82:0F:91:F5:1C:96:56:92:C4:1A:F6:C6:27:FD:6C:FC:31:F2:BB:90:17:22:59:5B:50 You can useOpenSSL orkeytool to display the fingerprint. For example:

keytool -list -v -keystore -storetype PKCS12 -storepass

Start EthSigner

ethsigner --tls-keystore-file

/Users/me/my_node/keystore.pfx --tls-keystore-password-file

/Users/me/my_node/keystorePassword --tls-known-clients-file

/Users/me/my_node/knownClients --tls-allow-ca-clients The command line:

- Specifies the EthSigner keystore using the--tls-keystore-file
- option.
- Specifies the file that contains the password to decrypt the keystore using the--tls-keystore-password-file
- option.
- Specifies the clients
- that are trusted to connect to EthSigner using the--tls-known-clients-file
- option.
- Allow access to clients with trusted CA certificates using the--tls-allow-ca-clients
- option.

note Use the--tls-allow-any-client option to allow access to any client.

--tls-allow-any-client cannot be used withtls-known-clients-file or--tls-allow-ca-clients .

Server TLS connection

Allow EthSigner to send and receive secure HTTP JSON-RPCs from the server (for example Besu).

Server prerequisites :

- The server must be configured to allow TLS communication
- .
- Server's password-protected PKCS #12 keystore information.

Create the known servers file

Create a file (in this example,knownServers) that lists one or more trusted servers. The file contents use the formatwhere:

-
- is the server hostname
-
- is the port used for communication
-
- is the SHA-256 fingerprint of the server's certificate.

localhost:8590 6C:B2:3E:F9:88:43:5E:62:69:9F:A9:9D:41:14:03:BA:83:24:AC:04:CE:BD:92:49:1B:8D:B2:A4:86:39:4C:BB
127.0.0.1:8590 6C:B2:3E:F9:88:43:5E:62:69:9F:A9:9D:41:14:03:BA:83:24:AC:04:CE:BD:92:49:1B:8D:B2:A4:86:39:4C:BB
note Specify both hostname and IP address in the file if unsure which is used in requests.

Start EthSigner

ethsigner --downstream-http-tls-enabled --downstream-http-tls-keystore-file

/Users/me/my_node/keystore.pfx --downstream-http-tls-keystore-password-file

/Users/me/my_node/keyPassword --downstream-http-tls-known-servers-file

/Users/me/my_node/knownServers The command line:

- Enables TLS using the--downstream-http-tls-enabled
- option.
- Specifies the keystore to present during authentication using the-downstream-http-tls-keystore-file
- option.
- Specifies the file that contains the password to decrypt the keystore using the-downstream-http-tls-keystore-password-file
- option.
- Specifies the servers
- to connect to using the--downstream-http-tls-known-servers-file
- option.

note The--downstream-http-tls-ca-auth-enabled option istrue by default and allows connections to servers with trusted root CAs. Edit this page Last updatedonMar 30, 2023 byEric LinPrevious Using the configuration file Next Use metrics to monitor performance