Encryption of data in transit 9.9

Last Updated: 2024-10-30

Db2® uses the Transport Layer Security (TLS) protocol to securely transmit data between servers and clients. TLS technology uses both asymmetric cryptography (for example, public key encryption) and symmetric cryptography to make this work.

You can use TLS to protect data in transit on all networks that use TCP/IP. In other words, a TLS connection is a secured TCP/IP connection. 9.9

Public key encryption for server authentication

TLS uses public-key algorithms to exchange encryption key information and digital certificate information. Public key encryption is used to ensure that a client can trust the certificate that is used by a server.

Public key cryptography uses two different encryption keys during a TLS session:
– A public key to encrypt data.
– An associated private key to decrypt it.

With public-key cryptography, the public key is not secret, but the messages it encrypts can be decrypted only by using it's associated private key. The private key must be securely stored in a file that is called a keystore.
Public-key algorithms alone do not ensure secure communication, you also need to verify the identity of whoever is communicating with you. To do this authentication, TLS uses digital certificates.

Distribution and use of digital certificates

To facilitate encryption of data in a Db2 environment, the following tasks need to happen for each Db2 server within your organization:

1. A member of your organization uses IBM Global Security Kit (GSKit) to create a public and private key pair.
2. The public key is sent to a certificate authority (CA) where a certificate is created and signed.
3. The server's certificate (which includes the server's public key) is distributed to all of the Db2 clients (and servers) within your organization for storage within their local keystores.

Once the certificates for each server have been distributed within your network, all of the parts needed to make TLS work are in place.

Before data is encrypted for transmission between Db2 nodes in your network, a TLS handshake occurs. This enables a client to check the validity of a server's certificate and, if the certificate is trusted, create a session key by using the server's public key. The session key is used to encrypt data traveling between the client and server for the duration of the connection.

- **Keystores**
  To ensure secure storage of private keys and certificates, you need to use a keystore. You can use the IBM® Global Security Kit (GSKit) to create a PKCS#12 keystore (with the .p12 extension) or a CMS keystore (with the .kdb extension).
- **Digital certificates**
  A digital certificate consists of the public portion of a private/public key pair and metadata values tha identify the holder of the certificate (name, company name, certificate expiry date, etc.). A certificate said to be 'signed' when a CA or individual uses a private key to encrypt a hash of a message.
- **The TLS handshake**
  During a *TLS handshake*, a public-key algorithm is used to securely exchange digital signatures and encryption keys between a client and a server. This identity and key information is used to establish a secure connection for the session between the client and the server. After the secure session is established, data transmission between the client and server is encrypted using a symmetric algorithm, such as AES.
- **Hostname validation for Db2 11.5.6 clients**
  Db2 11.5.6 clients can verify the hostname that appears in a Db2 server's Transport Layer Security (TLS, formerly known as SSL) certificate against the server for which they are configured to connect. Using hostname validation, Db2 clients have an added layer of security when negotiating secure connections to Db2 servers during a TLS handshake.
- **First steps in enabling TLS in Db2 servers and clients**
  Db2 11.5.8 servers and clients support Transport Layer Security (TLS) 1.3. Before enabling TLS 1.3, ensure you are aware of the specific changes that affect certificate use, as well as best practices for supporting older clients.
- **Enabling TLS 1.3 in a Db2 environment where TLS is already in use**
  You can enable TLS 1.3 support in a Db2 environment that already uses TLS.
- **TLS configuration of Db2**
  The Db2 database system supports the use of the Transport Layer Security (TLS) protocol, to enable client to validate the certificate of a Db2 server, and to provide private communication between the client and server by use of encryption.

Parent topic:

→ Data encryption