Contents

SSL commands

The device supports the FIPS mode that complies with NIST FIPS 140-2 requirements. Support for features, commands, and parameters might differ in FIPS mode and non-FIPS mode. For more information about FIPS mode, see *Security Configuration Guide*.

ciphersuite

Use **ciphersuite** to specify the cipher suites supported by an SSL server policy.

Use **undo ciphersuite** to restore the default.

Syntax

In non-FIPS mode:

ciphersuite { **dhe_rsa_aes_128_cbc_sha** | **dhe_rsa_aes_128_cbc_sha256** | **dhe_rsa_aes_256_cbc_sha** | **dhe_rsa_aes_256_cbc_sha256** | **ecdhe_ecdsa_aes_128_cbc_sha256** | **ecdhe_ecdsa_aes_128_gcm_sha256** | **ecdhe_ecdsa_aes_256_cbc_sha384** | **ecdhe_ecdsa_aes_256_gcm_sha384** | **ecdhe_rsa_aes_128_cbc_sha256** | **ecdhe_rsa_aes_128_gcm_sha256** | **ecdhe_rsa_aes_256_cbc_sha384** | **ecdhe_rsa_aes_256_gcm_sha384** | **exp_rsa_des_cbc_sha** | **exp_rsa_rc2_md5** | **exp_rsa_rc4_md5** | **rsa_3des_ede_cbc_sha** | **rsa_aes_128_cbc_sha** | **rsa_aes_128_cbc_sha256** | **rsa_aes_256_cbc_sha** | **rsa_aes_256_cbc_sha256** | **rsa_des_cbc_sha** | **rsa_rc4_128_md5** | **rsa_rc4_128_sha** } *

undo ciphersuite

In FIPS mode:

ciphersuite { **ecdhe_ecdsa_aes_128_cbc_sha256** | **ecdhe_ecdsa_aes_256_cbc_sha384** | **ecdhe_ecdsa_aes_128_gcm_sha256** | **ecdhe_ecdsa_aes_256_gcm_sha384** | **ecdhe_rsa_aes_128_cbc_sha256** | **ecdhe_rsa_aes_128_gcm_sha256** | **ecdhe_rsa_aes_256_cbc_sha384** | **ecdhe_rsa_aes_256_gcm_sha384** | **rsa_aes_128_cbc_sha** | **rsa_aes_128_cbc_sha256** | **rsa_aes_256_cbc_sha** | **rsa_aes_256_cbc_sha256** } *

undo ciphersuite

Default

An SSL server policy supports all cipher suites.

Views

SSL server policy view

Predefined user roles

network-admin

Parameters

dhe_rsa_aes_128_cbc_sha: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA.

dhe_rsa_aes_128_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

dhe_rsa_aes_256_cbc_sha: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA.

dhe_rsa_aes_256_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA256.

ecdhe_ecdsa_aes_128_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

ecdhe_ecdsa_aes_128_gcm_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 128-bit AES_GCM, and MAC algorithm SHA256.

ecdhe_ecdsa_aes_256_cbc_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA384.

ecdhe_ecdsa_aes_256_gcm_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 256-bit AES_GCM, and MAC algorithm SHA384.

ecdhe_rsa_aes_128_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

ecdhe_rsa_aes_128_gcm_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 128-bit AES_GCM, and MAC algorithm SHA256.

ecdhe_rsa_aes_256_cbc_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA384.

ecdhe_rsa_aes_256_gcm_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 256-bit AES_GCM, and MAC algorithm SHA384.

exp_rsa_des_cbc_sha: Specifies the export cipher suite that uses key exchange algorithm RSA, data encryption algorithm DES_CBC, and MAC algorithm SHA.

exp_rsa_rc2_md5: Specifies the export cipher suite that uses key exchange algorithm RSA, data encryption algorithm RC2, and MAC algorithm MD5.

exp_rsa_rc4_md5: Specifies the export cipher suite that uses key exchange algorithm RSA, data encryption algorithm RC4, and MAC algorithm MD5.

rsa_3des_ede_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 3DES_EDE_CBC, and MAC algorithm SHA.

rsa_aes_128_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA.

rsa_aes_128_cbc_ sha256: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

rsa_aes_256_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA.

rsa_aes_256_cbc_ sha256: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA256.

rsa_des_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm DES_CBC, and MAC algorithm SHA.

rsa_rc4_128_md5: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit RC4, and MAC algorithm MD5.

rsa_rc4_128_sha: Specifies key exchange algorithm RSA, data encryption algorithm 128-bit RC4, and MAC algorithm SHA.

Usage guidelines

SSL employs the following algorithms:

- **Data encryption algorithms**—Encrypt data to ensure privacy. Commonly used data encryption algorithms are symmetric key algorithms. When a symmetric key algorithm is used, the SSL server and the SSL client must use the same key.
- **Message Authentication Code (MAC) algorithms**—Calculate the MAC value for data to ensure integrity. Commonly used MAC algorithms include MD5 and SHA. When a MAC algorithm is used, the SSL server and the SSL client must use the same key.
- **Key exchange algorithms**—Implement secure exchange of the keys used by the symmetric key algorithm and the MAC algorithm. Commonly used key exchange algorithms are usually asymmetric key algorithms, such as RSA.

After the SSL server receives a cipher suite from a client, the server matches the received cipher suite against the cipher suits it supports. If a match is found, the cipher suite negotiation succeeds. Otherwise, the negotiation fails.

If you execute this command multiple times, the most recent configuration takes effect.

Examples

Configure SSL server policy **policy1** to support the following cipher suites:

- Key exchange algorithm DHE RSA, data encryption algorithm 128-bit AES, and MAC algorithm SHA.
- Key exchange algorithm RSA, data encryption algorithm 128-bit AES, and MAC algorithm SHA.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] ciphersuite dhe_rsa_aes_128_cbc_sha
rsa_aes_128_cbc_sha
```

Related commands

display ssl server-policy

prefer-cipher

client-verify

Use **client-verify** to enable mandatory or optional SSL client authentication.

Use **undo client-verify** to restore the default.

Syntax

client-verify { **enable** | **optional** }

undo client-verify [**enable**]

Default

SSL client authentication is disabled. The SSL server does not authenticate SSL clients based on digital certificates.

Views

SSL server policy view

Predefined user roles

network-admin

Parameters

enable: Enables mandatory SSL client authentication.

optional: Enables optional SSL client authentication.

Usage guidelines

SSL uses digital certificates to authenticate communicating parties. For more information about digital certificates, see *Security Configuration Guide*.

Mandatory SSL client authentication—The SSL server requires an SSL client to submit its digital certificate for identity authentication. The SSL client can access the SSL server only after it passes identity authentication.

Optional SSL client authentication—The SSL server does not require an SSL client to submit its digital certificate for identity authentication.

- If an SSL client submits its certificate to the SSL server, the server authenticates the client identity. The client must pass authentication to access the server.
- If an SSL client does not submit its certificate to the SSL server, the server does not authenticate the client identity. The client can access the SSL server without authentication.

If SSL client authentication is disabled, the SSL server does not authenticate SSL clients regardless of whether the clients submit digital certificates or not. SSL clients can access the SSL server without authentication.

When authenticating a client by using the digital certificate, the SSL server performs the following operations:

- Verifies the certificate chain presented by the client.
- Checks that the certificates in the certificate chain (except the root CA certificate) are not revoked.

Examples

Enable mandatory SSL client authentication.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] client-verify enable
```

Enable optional SSL client authentication.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] client-verify optional
```

Disable SSL client authentication.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] undo client-verify
```

Related commands

`display ssl server-policy`

display ssl client-policy

Use **`display ssl client-policy`** to display SSL client policy information.

Syntax

`display ssl client-policy` [*`policy-name`*]

Views

Any view

Predefined user roles

network-admin

network-operator

Parameters

policy-name: Specifies an SSL client policy by its name, a case-insensitive string of 1 to 31 characters. If you do not specify a policy name, this command displays information about all SSL client policies.

Examples

Display information about the SSL client policy **policy1**.

```
<Sysname> display ssl client-policy policy1
 SSL client policy: policy1
     SSL version: SSL 3.0
     PKI domain: client-domain
     Preferred ciphersuite:
         RSA_AES_128_CBC_SHA
     Server-verify: enabled
```

Table 1 Command output

Field	Description
Server-verify	Indicates whether the client is enabled to use digital certificates to authenticate servers.

display ssl server-policy

Use **display ssl server-policy** to display SSL server policy information.

Syntax

display ssl server-policy [*policy-name*]

Views

Any view

Predefined user roles

network-admin

network-operator

Parameters

policy-name: Specifies an SSL server policy by its name, a case-insensitive string of 1 to 31 characters. If you do not specify a policy name, this command displays information about all SSL server policies.

Examples

Display information about the SSL server policy **policy1**.

```
<Sysname> display ssl server-policy policy1
 SSL server policy: policy1
     PKI domain: server-domain
     Ciphersuites:
         DHE_RSA_AES_128_CBC_SHA
         RSA_AES_128_CBC_SHA
     Session cache size: 600
     Caching timeout: 3600 seconds
```

```
Client-verify: Enabled
```

Table 2 Command output

Field	Description
Caching timeout	Session cache timeout time in seconds.
Client-verify	SSL client authentication mode, including: • **Disabled**—SSL client authentication is disabled. • **Enabled**—SSL client authentication is mandatory. • **Optional**—SSL client authentication is optional.

pki-domain (SSL client policy view)

Use **pki-domain** to specify a PKI domain for an SSL client policy.

Use **undo pki-domain** to restore the default.

Syntax

pki-domain *domain-name*

undo pki-domain

Default

No PKI domain is specified for an SSL client policy.

Views

SSL client policy view

Predefined user roles

network-admin

Parameters

domain-name: Specifies a PKI domain by its name, a case-insensitive string of 1 to 31 characters.

Usage guidelines

If you specify a PKI domain for an SSL client policy, the SSL client that uses the SSL client policy will obtain its digital certificate through the specified PKI domain.

Examples

Specify PKI domain **client-domain** for SSL client policy **policy1**.

```
<Sysname> system-view
[Sysname] ssl client-policy policy1
[Sysname-ssl-client-policy-policy1] pki-domain client-domain
```

Related commands

display ssl client-policy

pki domain

pki-domain (SSL server policy view)

Use **pki-domain** to specify a PKI domain for an SSL server policy.

Use **undo pki-domain** to restore the default.

Syntax

pki-domain *domain-name*

undo pki-domain

Default

No PKI domain is specified for an SSL server policy.

Views

SSL server policy view

Predefined user roles

network-admin

Parameters

domain-name: Specifies a PKI domain by its name, a case-insensitive string of 1 to 31 characters.

Usage guidelines

If you specify a PKI domain for an SSL server policy, the SSL server that uses the SSL server policy will obtain its digital certificate through the specified PKI domain.

Examples

Specify PKI domain **server-domain** for SSL server policy **policy1**.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] pki-domain server-domain
```

Related commands

display ssl server-policy

pki domain

prefer-cipher

Use **prefer-cipher** to specify a preferred cipher suite for an SSL client policy.

Use **undo prefer-cipher** to restore the default.

Syntax

In non-FIPS mode:

prefer-cipher { **dhe_rsa_aes_128_cbc_sha** | **dhe_rsa_aes_128_cbc_sha256** | **dhe_rsa_aes_256_cbc_sha** | **dhe_rsa_aes_256_cbc_sha256** | **ecdhe_ecdsa_aes_128_cbc_sha256** | **ecdhe_ecdsa_aes_128_gcm_sha256** | **ecdhe_ecdsa_aes_256_cbc_sha384** | **ecdhe_ecdsa_aes_256_gcm_sha384** | **ecdhe_rsa_aes_128_cbc_sha256** | **ecdhe_rsa_aes_128_gcm_sha256** | **ecdhe_rsa_aes_256_cbc_sha384** | **ecdhe_rsa_aes_256_gcm_sha384** | **exp_rsa_des_cbc_sha** | **exp_rsa_rc2_md5** | **exp_rsa_rc4_md5** | **rsa_3des_ede_cbc_sha** | **rsa_aes_128_cbc_sha** | **rsa_aes_128_cbc_sha256** | **rsa_aes_256_cbc_sha** | **rsa_aes_256_cbc_sha256** | **rsa_des_cbc_sha** | **rsa_rc4_128_md5** | **rsa_rc4_128_sha** }

undo prefer-cipher

In FIPS mode:

prefer-cipher { **ecdhe_ecdsa_aes_128_cbc_sha256** | **ecdhe_ecdsa_aes_128_gcm_sha256** | **ecdhe_ecdsa_aes_256_cbc_sha384** | **ecdhe_ecdsa_aes_256_gcm_sha384** | **ecdhe_rsa_aes_128_cbc_sha256** |

ecdhe_rsa_aes_128_gcm_sha256 | **ecdhe_rsa_aes_256_cbc_sha384** | **ecdhe_rsa_aes_256_gcm_sha384** | **rsa_aes_128_cbc_sha** | **rsa_aes_128_cbc_sha256** | **rsa_aes_256_cbc_sha** | **rsa_aes_256_cbc_sha256** }

undo prefer-cipher

Default

In non-FIPS mode:

The preferred cipher suite of an SSL client policy is **rsa_rc4_128_md5**.

In FIPS mode:

The preferred cipher suite of an SSL client policy is **rsa_aes_128_cbc_sha**.

Views

SSL client policy view

Predefined user roles

network-admin

Parameters

dhe_rsa_aes_128_cbc_sha: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA.

dhe_rsa_aes_128_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

dhe_rsa_aes_256_cbc_sha: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA.

dhe_rsa_aes_256_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm DHE RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA256.

ecdhe_ecdsa_aes_128_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

ecdhe_ecdsa_aes_128_gcm_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 128-bit AES_GCM, and MAC algorithm SHA256.

ecdhe_ecdsa_aes_256_cbc_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA384.

ecdhe_ecdsa_aes_256_gcm_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE ECDSA, data encryption algorithm 256-bit AES_GCM, and MAC algorithm SHA384.

ecdhe_rsa_aes_128_cbc_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

ecdhe_rsa_aes_128_gcm_sha256: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 128-bit AES_GCM, and MAC algorithm SHA256.

ecdhe_rsa_aes_256_cbc_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA384.

ecdhe_rsa_aes_256_gcm_sha384: Specifies the cipher suite that uses key exchange algorithm ECDHE RSA, data encryption algorithm 256-bit AES_GCM, and MAC algorithm SHA384.

exp_rsa_des_cbc_sha: Specifies the export cipher suite that uses key exchange algorithm RSA, data encryption algorithm DES_CBC, and MAC algorithm SHA.

exp_rsa_rc2_md5: Specifies the export cipher suite that uses key exchange algorithm RSA, data encryption algorithm RC2, and MAC algorithm MD5.

exp_rsa_rc4_md5: Specifies the export cipher suite that uses key exchange algorithm RSA, data encryption algorithm RC4, and MAC algorithm MD5.

rsa_3des_ede_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 3DES_EDE_CBC, and MAC algorithm SHA.

rsa_aes_128_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA.

rsa_aes_128_cbc_ sha256: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA256.

rsa_aes_256_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA.

rsa_aes_256_cbc_ sha256: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 256-bit AES_CBC, and MAC algorithm SHA256.

rsa_des_cbc_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm DES_CBC, and MAC algorithm SHA.

rsa_rc4_128_md5: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit RC4, and MAC algorithm MD5.

rsa_rc4_128_sha: Specifies the cipher suite that uses key exchange algorithm RSA, data encryption algorithm 128-bit RC4, and MAC algorithm SHA.

Usage guidelines

SSL employs the following algorithms:

- **Data encryption algorithms**—Encrypt data to ensure privacy. Commonly used data encryption algorithms are usually symmetric key algorithms. When using a symmetric key algorithm, the SSL server and the SSL client must use the same key.
- **Message Authentication Code (MAC) algorithms**—Calculate the MAC value for data to ensure integrity. Commonly used MAC algorithms include MD5 and SHA. When using a MAC algorithm, the SSL server and the SSL client must use the same key.
- **Key exchange algorithms**—Implement secure exchange of the keys used by the symmetric key algorithm and MAC algorithm. Commonly used key exchange algorithms are asymmetric key algorithms, such as RSA.

The SSL client sends the preferred cipher suite to the SSL server. The server compares the received cipher suite with the cipher suits it supports. If a match is found, the cipher suite negotiation succeeds. If no match is found, the negotiation fails.

If you execute this command multiple times, the most recent configuration takes effect.

Examples

Configure SSL client policy **policy1** to support the key exchange algorithm RSA, data encryption algorithm 128-bit AES_CBC, and MAC algorithm SHA.

```
<Sysname> system-view
[Sysname] ssl client-policy policy1
[Sysname-ssl-client-policy-policy1] prefer-cipher rsa_aes_128_cbc_sha
```

Related commands

ciphersuite

display ssl client-policy

server-verify enable

Use **server-verify enable** to enable the SSL client to use digital certificates to authenticate the SSL server.

Use **undo server-verify enable** to disable SSL server authentication. The SSL client does not authenticate the SSL server.

Syntax

server-verify enable

undo server-verify enable

Default

The SSL client uses digital certificates to authenticate the SSL server.

Views

SSL client policy view

Predefined user roles

network-admin

Usage guidelines

SSL uses digital certificates to authenticate communicating parties. For more information about digital certificates, see *Security Configuration Guide*.

If you execute the **server-verify enable** command, the SSL server must send its digital certificate to the SSL client for authentication. The client can access the SSL server only after the server passes the authentication.

Examples

Enable the SSL client to use digital certificates to authenticate the SSL server.

```
<Sysname> system-view
[Sysname] ssl client-policy policy1
[Sysname-ssl-client-policy-policy1] server-verify enable
```

Related commands

display ssl client-policy

session

Use **session** to set the maximum number of sessions that the SSL server can cache and the timeout time for cached sessions.

Use **undo session** to restore the default.

Syntax

session { **cachesize** *size* | **timeout** *time* } *

undo session { **cachesize** | **timeout** } *

Default

The SSL server can cache a maximum of 500 sessions, and the timeout time for cached sessions is 3600 seconds.

Views

SSL server policy view

Predefined user roles

network-admin

Parameters

cachesize *size*: Sets the maximum number of cached sessions, in the range of 100 to 20480.

timeout *time*: Sets the session cache timeout in the range of 1 to 4294967295 seconds.

Usage guidelines

The SSL server caches SSL sessions to reuse negotiated session parameters to simplify SSL handshake. Use this command to limit the maximum number and timeout time for cached sessions. When the number of cached sessions reaches the maximum, SSL does not cache new sessions. When the timeout timer for a cached session expires, SSL deletes the session.

Examples

Set the maximum number of cached sessions to 600, and the timeout time for cached sessions to 1800 seconds.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] session cachesize 600 timeout 1800
```

Related commands

display ssl server-policy

ssl client-policy

Use **ssl client-policy** to create an SSL client policy and enter its view, or enter the view of an existing SSL client policy.

Use **undo ssl client-policy** to delete an SSL client policy.

Syntax

ssl client-policy *policy-name*

undo ssl client-policy *policy-name*

Default

No SSL client policies exist.

Views

System view

Predefined user roles

network-admin

Parameters

policy-name: Specifies an SSL client policy by its name, a case-insensitive string of 1 to 31 characters.

Usage guidelines

This command creates an SSL client policy for which you can configure SSL parameters that the client uses to establish a connection to the server. The parameters include a PKI domain and a preferred cipher suite. An SSL client policy takes effect only after it is associated with an application such as DDNS.

Examples

Create an SSL client policy named **policy1** and enter its view.

```
<Sysname> system-view
[Sysname] ssl client-policy policy1
[Sysname-ssl-client-policy-policy1]
```

Related commands

display ssl client-policy

ssl renegotiation disable

Use **ssl renegotiation disable** to disable SSL session renegotiation.

Use **undo ssl renegotiation disable** to restore the default.

Syntax

ssl renegotiation disable

undo ssl renegotiation disable

Default

SSL session renegotiation is enabled.

Views

System view

Predefined user roles

network-admin

Usage guidelines

The SSL session renegotiation feature enables the SSL client and server to reuse a previously negotiated SSL session for an abbreviated handshake.

Disabling session renegotiation causes more computational overhead to the system but it can avoid potential risks. Disable SSL session renegotiation only when explicitly required.

Examples

#Disable SSL session renegotiation.

```
<Sysname> system-view
[Sysname] ssl renegotiation disable
```

ssl server-policy

Use **ssl server-policy** to create an SSL server policy and enter its view, or enter the view of an existing SSL server policy.

Use **undo ssl server-policy** to delete an SSL server policy.

Syntax

ssl server-policy *policy-name*

undo ssl server-policy *policy-name*

Default

No SSL server policies exist.

Views

System view

Predefined user roles

network-admin

Parameters

policy-name: Specifies a name for the SSL server policy, a case-insensitive string of 1 to 31 characters.

Usage guidelines

This command creates an SSL server policy for which you can configure SSL parameters such as a PKI domain and supported cipher suits. An SSL server policy takes effect only after it is associated with an application such as HTTPS.

Examples

Create an SSL server policy named **policy1** and enter its view.

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1]
```

Related commands

display ssl server-policy

ssl version disable

Use **ssl version disable** to disable the SSL server from using specific SSL protocol versions for session negotiation.

Use **undo ssl version disable** restore the default.

Syntax

In non-FIPS mode:

ssl version { **ssl3.0** | **tls1.0** | **tls1.1** } * **disable**

undo ssl version { **ssl3.0** | **tls1.0** | **tls1.1** } * **disable**

In FIPS mode:

ssl version { **tls1.0** | **tls1.1** } * **disable**

undo ssl version { **tls1.0** | **tls1.1** } * **disable**

Default

In non-FIPS mode, the SSL server supports SSL 3.0, TLS 1.0, TLS 1.1, and TLS 1.2.

In FIPS mode, the SSL server supports TLS 1.0, TLS 1.1, and TLS 1.2.

Views

System view

Predefined user roles

network-admin

Parameters

ssl3.0: Specifies SSL 3.0.

tls1.0: Specifies TLS 1.0.

tls1.1: Specifies TLS 1.1.

Usage guidelines

To enhance system security, you can disable the SSL server from using specific SSL protocol versions (SSL 3.0, TLS 1.0, and TLS 1.1) for session negotiation.

Disabling an SSL protocol version does not affect the availability of earlier SSL protocol versions. For example, if you execute the **ssl version tls1.1 disable** command, TLS 1.1 is disabled but TLS 1.0 is still available for the SSL server.

Examples

Disable SSL 3.0 for the SSL server.

```
<Sysname> system-view
[Sysname] ssl version ssl3.0 disable
```

version

Use **version** to specify an SSL protocol version for an SSL client policy.

Use **undo version** to restore the default.

Syntax

In non-FIPS mode:

version { **ssl3.0** | **tls1.0** | **tls1.1** | **tls1.2** }

undo version

In FIPS mode:

version { **tls1.0** | **tls1.1** | **tls1.2** }

undo version

Default

An SSL client policy uses SSL protocol version TLS 1.0.

Views

SSL client policy view

Predefined user roles

network-admin

Parameters

ssl3.0: Specifies SSL 3.0.

tls1.0: Specifies TLS 1.0.

tls1.1: Specifies TLS 1.1.

tls1.2: Specifies TLS 1.2.

Usage guidelines

To ensure security, do not specify SSL 3.0 for an SSL client policy.

If you execute this command multiple times, the most recent configuration takes effect.

Examples

Set the SSL protocol version to TLS 1.0 for SSL client policy **policy1**.

```
<Sysname> system-view
[Sysname] ssl client-policy policy1
[Sysname-ssl-client-policy-policy1] version tls1.0
```

Related commands

display ssl client-policy