目 录

1 SSL ····· 1-1
1.1 SSL配置命令 ····· 1-1
1.1.1 certificate-chain-sending enable ····· 1-1
1.1.2 ciphersuite ····· 1-1
1.1.3 client-verify ····· 1-4
1.1.4 display ssl client-policy ····· 1-5
1.1.5 display ssl server-policy ····· 1-6
1.1.6 pki-domain (SSL client policy view) ····· 1-7
1.1.7 pki-domain (SSL server policy view) ····· 1-8
1.1.8 prefer-cipher ····· 1-9
1.1.9 server-verify enable ····· 1-11
1.1.10 session ····· 1-12
1.1.11 ssl client-policy ····· 1-13
1.1.12 ssl renegotiation disable ····· 1-13
1.1.13 ssl server-policy ····· 1-14
1.1.14 ssl version disable ····· 1-15
1.1.15 version ····· 1-16
1.1.16 version disable ····· 1-16

1 SSL

1.1 SSL配置命令

1.1.1 certificate-chain-sending enable

certificate-chain-sending enable 命令用来配置 SSL 协商时 SSL 服务器端发送完整的证书链。

undo certificate-chain-sending enable 命令用来恢复缺省情况。

【命令】

certificate-chain-sending enable

undo certificate-chain-sending enable

【缺省情况】

SSL 协商时，SSL 服务器端只发送本地证书，不发送证书链。

【视图】

SSL 服务器端策略视图

【缺省用户角色】

network-admin

【使用指导】

仅当 SSL 客户端没有完整的证书链对服务器端的数字证书进行验证时，请通过本命令要求 SSL 服务器端在握手协商时向对端发送完整的证书链，以保证 SSL 会话的正常建立。否则，建议关闭此功能，减轻协商阶段的网络开销。

【举例】

```
<Sysname> system-view
[Sysname] ssl server-policy policy1
[Sysname-ssl-server-policy-policy1] certificate-chain-sending enable
```

1.1.2 ciphersuite

ciphersuite 命令用来配置 SSL 服务器端策略支持的加密套件。

undo ciphersuite 命令用来恢复缺省情况。

【命令】

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

【缺省情况】

SSL 服务器端策略支持所有的加密套件。

【视图】

SSL 服务器端策略视图

【缺省用户角色】

network-admin

【参数】

dhe_rsa_aes_128_cbc_sha：密钥交换算法采用 DHE RSA、数据加密算法采用 128 位的 AES、MAC 算法采用 SHA。

dhe_rsa_aes_128_cbc_sha256：密钥交换算法采用 DHE RSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

dhe_rsa_aes_256_cbc_sha：密钥交换算法采用 DHE RSA、数据加密算法采用 256 位的 AES、MAC 算法采用 SHA。

dhe_rsa_aes_256_cbc_sha256：密钥交换算法采用 DHE RSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA256。

ecdhe_ecdsa_aes_128_cbc_sha256：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

ecdhe_ecdsa_aes_128_gcm_sha256：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 128 位的 AES_GCM、MAC 算法采用 SHA256。

ecdhe_ecdsa_aes_256_cbc_sha384：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA384。

ecdhe_ecdsa_aes_256_gcm_sha384：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 256 位的 AES_GCM、MAC 算法采用 SHA384。

ecdhe_rsa_aes_128_cbc_sha256：密钥交换算法采用 ECDHE RSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

ecdhe_rsa_aes_128_gcm_sha256：密钥交换算法采用 ECDHE RSA、数据加密算法采用 128 位的 AES_GCM、MAC 算法采用 SHA256。

ecdhe_rsa_aes_256_cbc_sha384：密钥交换算法采用 ECDHE RSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA384。

ecdhe_rsa_aes_256_gcm_sha384：密钥交换算法采用 ECDHE RSA、数据加密算法采用 256 位的 AES_GCM、MAC 算法采用 SHA384。

exp_rsa_des_cbc_sha：满足出口限制的算法套件。密钥交换算法采用 RSA、数据加密算法采用 DES_CBC、MAC 算法采用 SHA。

exp_rsa_rc2_md5：满足出口限制的算法套件。密钥交换算法采用 RSA、数据加密算法采用 RC2、MAC 算法采用 MD5。

exp_rsa_rc4_md5：满足出口限制的算法套件。密钥交换算法采用 RSA、数据加密算法采用 RC4、MAC 算法采用 MD5。

rsa_3des_ede_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 3DES_EDE_CBC、MAC 算法采用 SHA。

rsa_aes_128_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 128 位 AES_CBC、MAC 算法采用 SHA。

rsa_aes_128_cbc_sha256：密钥交换算法采用 RSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

rsa_aes_256_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 256 位 AES_CBC、MAC 算法采用 SHA。

rsa_aes_256_cbc_sha256：密钥交换算法采用 RSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA256。

rsa_des_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 DES_CBC、MAC 算法采用 SHA。

rsa_rc4_128_md5：密钥交换算法采用 RSA、数据加密算法采用 128 位的 RC4、MAC 算法采用 MD5。

rsa_rc4_128_sha：密钥交换算法采用 RSA、数据加密算法采用 128 位的 RC4、MAC 算法采用 SHA。

【使用指导】

为了提高安全性，SSL 协议采用了如下算法：

- 数据加密算法：用来对传输的数据进行加密，以保证数据传输的私密性。常用的数据加密算法通常为对称密钥算法，如 DES_CBC、3DES_EDE_CBC、AES_CBC、RC4 等。使用对称密钥算法时，要求 SSL 服务器端和 SSL 客户端具有相同的密钥。
- MAC（Message Authentication Code，消息验证码）算法：用来计算数据的 MAC 值，以防止发送的数据被篡改。常用的 MAC 算法有 MD5、SHA 等。使用 MAC 算法时，要求 SSL 服务器端和 SSL 客户端具有相同的密钥。
- 密钥交换算法：用来实现密钥交换，以保证对称密钥算法、MAC 算法中使用的密钥在 SSL 服务器端和 SSL 客户端之间安全地传递。常用的密钥交换算法通常为非对称密钥算法，如 RSA。

通过本命令可以配置 SSL 服务器端策略支持的各种算法组合。例如，**rsa_des_cbc_sha** 表示 SSL 服务器端策略支持的密钥交换算法为 RSA、数据加密算法为 DES_CBC、MAC 算法为 SHA。

SSL 服务器接收到 SSL 客户端发送的客户端加密套件后，将服务器支持的加密套件与 SSL 客户端支持的加密套件比较。如果 SSL 服务器支持的加密套件中存在 SSL 客户端支持的加密套件，则加密套件协商成功；否则，加密套件协商失败。

多次执行本命令，最后一次执行的命令生效。

【举例】

指定 SSL 服务器端策略支持如下加密套件：

- 密钥交换算法为 DHE RSA、数据加密算法为 128 位的 AES、MAC 算法为 SHA
- 密钥交换算法为 RSA、数据加密算法为 128 位的 AES、MAC 算法为 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
```

【相关命令】

- **display ssl server-policy**
- **prefer-cipher**

1.1.3 client-verify

client-verify 命令用来配置 SSL 服务器端对 SSL 客户端的身份验证方案。

undo client-verify 命令用来恢复缺省情况。

【命令】

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

undo client-verify [**enable**]

【缺省情况】

SSL 服务器端不对 SSL 客户端进行基于数字证书的身份验证。

【视图】

SSL 服务器端策略视图

【缺省用户角色】

network-admin

【参数】

enable：表示 SSL 服务器端要求对 SSL 客户端进行基于数字证书的身份验证。

optional：表示 SSL 服务器端不强制要求对 SSL 客户端进行基于数字证书的身份验证，即身份验证可选。

【使用指导】

SSL 通过数字证书实现对对端的身份进行验证。数字证书的详细介绍，请参见“安全配置指导”中的“PKI”。

设备作为 SSL 服务器端，支持灵活的 SSL 客户端认证方案，具体如下：

- 执行了 **client-verify enable** 命令的情况下，则 SSL 客户端必须将自己的数字证书提供给服务器，以便服务器对客户端进行基于数字证书的身份验证。只有身份验证通过后，SSL 客户端才能访问 SSL 服务器。
- 执行了 **client-verify optional** 命令的情况下，若 SSL 客户端未提供数字证书给服务器，SSL 客户端也能访问 SSL 服务器；若 SSL 客户端提供数字证书给服务器，只有身份验证通过后，SSL 客户端才能访问 SSL 服务器。
- 执行了 **undo client-verify [enable]**命令的情况下，SSL 服务器端不要求 SSL 客户端提供数字证书，也不会对其进行基于数字证书的身份验证，SSL 客户端可以直接访问 SSL 服务器。

SSL 服务器端在基于数字证书对 SSL 客户端进行身份验证时，除了对 SSL 客户端发送的证书链进行验证，还要检查证书链中的除根 CA 证书外的每个证书是否均未被吊销。

【举例】

配置 SSL 服务器端要求对 SSL 客户端进行基于数字证书的身份验证。

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

配置 SSL 服务器端对 SSL 客户端进行基于数字证书的身份验证是可选的。

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

配置 SSL 服务器端不要求对 SSL 客户端进行基于数字证书的身份验证。

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

【相关命令】

- **display ssl server-policy**

1.1.4 display ssl client-policy

display ssl client-policy 命令用来显示 SSL 客户端策略的信息。

【命令】

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

【视图】

任意视图

【缺省用户角色】

network-admin

network-operator

【参数】

policy-name：显示指定的 SSL 客户端策略的信息，为 1～31 个字符的字符串，不区分大小写。如果不指定本参数，则显示所有 SSL 客户端策略的信息。

【举例】

显示名为 policy1 的 SSL 客户端策略的信息。

```
<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
```

表1-1 display ssl client-policy 命令显示信息描述表

字段	描述
SSL client policy	SSL客户端策略名
SSL version	SSL客户端策略使用的SSL协议版本
PKI domain	SSL客户端策略使用的PKI域
Preferred ciphersuite	SSL客户端策略支持的加密套件
Server-verify	SSL客户端策略的服务器端验证模式，取值包括： • disabled：不要求对 SSL 服务器进行基于数字证书的身份验证 • enabled：要求对 SSL 服务器进行基于数字证书的身份验证

1.1.5 display ssl server-policy

display ssl server-policy 命令用来显示 SSL 服务器端策略的信息。

【命令】

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

【视图】

任意视图

【缺省用户角色】

network-admin

network-operator

【参数】

policy-name：显示指定的 SSL 服务器端策略的信息，为 1～31 个字符的字符串，不区分大小写。如果不指定本参数，则显示所有 SSL 服务器端策略的信息。

【举例】

显示名为 policy1 的 SSL 服务器端策略的信息。

```
<Sysname> display ssl server-policy policy1
 SSL server policy: policy1
     Version info:
         SSL3.0: Disabled
         TLS1.0: Enabled
         TLS1.1: Disabled
         TLS1.2: Enabled
     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
```

表1-2 display ssl server-policy 命令显示信息描述表

字段	描述
SSL server policy	SSL服务器端策略名
Version info	SSL服务器端使用的版本： • SSL3.0 • TLS1.0 • TLS1.1 • TLS1.2 SSL服务器端是否允许使用版本： • Enabled：允许使用 • Disabled：不允许使用
PKI domain	SSL服务器端策略使用的PKI域
Ciphersuites	SSL服务器端策略支持的加密套件
Session cache size	SSL服务器端可以缓存的最大会话数目
Caching timeout	SSL服务器端会话缓存超时时间（单位为秒）
Client-verify	SSL服务器端策略的客户端验证模式，取值包括： • Disabled：不要求对客户端进行基于数字证书的身份验证 • Enabled：要求对客户端进行基于数字证书的身份验证 • Optional：SSL 服务器端对 SSL 客户端进行基于数字证书的身份验证是可选的

1.1.6 pki-domain (SSL client policy view)

pki-domain 命令用来配置 SSL 客户端策略所使用的 PKI 域。

undo pki-domain 命令用来恢复缺省情况。

【命令】

pki-domain *domain-name*

undo pki-domain

【缺省情况】

未指定 SSL 客户端策略所使用的 PKI 域。

【视图】

SSL 客户端策略视图

【缺省用户角色】

network-admin

【参数】

domain-name：PKI 域的域名，为 1～31 个字符的字符串，不区分大小写。

【使用指导】

如果通过本命令指定了 SSL 客户端策略使用的 PKI 域，则引用该客户端策略的 SSL 客户端将通过该 PKI 域获取客户端的数字证书。

【举例】

配置 SSL 客户端策略所使用的 PKI 域为 client-domain。

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

【相关命令】

- **display ssl client-policy**
- **pki domain**（安全命令参考/PKI）

1.1.7 pki-domain (SSL server policy view)

pki-domain 命令用来配置 SSL 服务器端策略所使用的 PKI 域。

undo pki-domain 命令用来恢复缺省情况。

【命令】

pki-domain *domain-name*

undo pki-domain

【缺省情况】

未指定 SSL 服务器端策略所使用的 PKI 域。

【视图】

SSL 服务器端策略视图

【缺省用户角色】

network-admin

【参数】

domain-name：PKI 域的域名，为 1～31 个字符的字符串，不区分大小写。

【使用指导】

如果通过本命令指定了 SSL 服务器端策略使用的 PKI 域，则引用该服务器端策略的 SSL 服务器将通过该 PKI 域获取服务器端的数字证书。

【举例】

配置 SSL 服务器端策略所使用的 PKI 域为 server-domain。

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

【相关命令】

- **display ssl server-policy**
- **pki domain**（安全命令参考/PKI）

1.1.8 prefer-cipher

prefer-cipher 命令用来配置 SSL 客户端策略支持的加密套件。

undo prefer-cipher 命令用来恢复缺省情况。

【命令】

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

【缺省情况】

SSL 客户端策略支持的加密套件为 **dhe_rsa_aes_128_cbc_sha**、**dhe_rsa_aes_256_cbc_sha**、**rsa_3des_ede_cbc_sha**、**rsa_aes_128_cbc_sha**、**rsa_aes_256_cbc_sha**。

【视图】

SSL 客户端策略视图

【缺省用户角色】

network-admin

【参数】

dhe_rsa_aes_128_cbc_sha：密钥交换算法采用 DHE RSA、数据加密算法采用 128 位的 AES、MAC 算法采用 SHA。

dhe_rsa_aes_128_cbc_sha256：密钥交换算法采用 DHE RSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

dhe_rsa_aes_256_cbc_sha：密钥交换算法采用 DHE RSA、数据加密算法采用 256 位的 AES、MAC 算法采用 SHA。

dhe_rsa_aes_256_cbc_sha256：密钥交换算法采用 DHE RSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA256。

ecdhe_ecdsa_aes_128_cbc_sha256：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

ecdhe_ecdsa_aes_128_gcm_sha256：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 128 位的 AES_GCM、MAC 算法采用 SHA256。

ecdhe_ecdsa_aes_256_cbc_sha384：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA384。

ecdhe_ecdsa_aes_256_gcm_sha384：密钥交换算法采用 ECDHE ECDSA、数据加密算法采用 256 位的 AES_GCM、MAC 算法采用 SHA384。

ecdhe_rsa_aes_128_cbc_sha256：密钥交换算法采用 ECDHE RSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

ecdhe_rsa_aes_128_gcm_sha256：密钥交换算法采用 ECDHE RSA、数据加密算法采用 128 位的 AES_GCM、MAC 算法采用 SHA256。

ecdhe_rsa_aes_256_cbc_sha384：密钥交换算法采用 ECDHE RSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA384。

ecdhe_rsa_aes_256_gcm_sha384：密钥交换算法采用 ECDHE RSA、数据加密算法采用 256 位的 AES_GCM、MAC 算法采用 SHA384。

exp_rsa_des_cbc_sha：满足出口限制的算法套件。密钥交换算法采用 RSA、数据加密算法采用 DES_CBC、MAC 算法采用 SHA。

exp_rsa_rc2_md5：满足出口限制的算法套件。密钥交换算法采用 RSA、数据加密算法采用 RC2、MAC 算法采用 MD5。

exp_rsa_rc4_md5：满足出口限制的算法套件。密钥交换算法采用 RSA、数据加密算法采用 RC4、MAC 算法采用 MD5。

rsa_3des_ede_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 3DES_EDE_CBC、MAC 算法采用 SHA。

rsa_aes_128_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 128 位 AES_CBC、MAC 算法采用 SHA。

rsa_aes_128_cbc_sha256：密钥交换算法采用 RSA、数据加密算法采用 128 位的 AES_CBC、MAC 算法采用 SHA256。

rsa_aes_256_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 256 位 AES_CBC、MAC 算法采用 SHA。

rsa_aes_256_cbc_sha256：密钥交换算法采用 RSA、数据加密算法采用 256 位的 AES_CBC、MAC 算法采用 SHA256。

rsa_des_cbc_sha：密钥交换算法采用 RSA、数据加密算法采用 DES_CBC、MAC 算法采用 SHA。

rsa_rc4_128_md5：密钥交换算法采用 RSA、数据加密算法采用 128 位的 RC4、MAC 算法采用 MD5。

rsa_rc4_128_sha：密钥交换算法采用 RSA、数据加密算法采用 128 位的 RC4、MAC 算法采用 SHA。

【使用指导】

为了提高安全性，SSL 协议采用了如下算法：

- 数据加密算法：用来对传输的数据进行加密，以保证数据传输的私密性。常用的数据加密算法通常为对称密钥算法。使用对称密钥算法时，要求 SSL 服务器端和 SSL 客户端具有相同的密钥。
- MAC（Message Authentication Code，消息验证码）算法：用来计算数据的 MAC 值，以防止发送的数据被篡改。常用的 MAC 算法有 MD5、SHA 等。使用 MAC 算法时，要求 SSL 服务器端和 SSL 客户端具有相同的密钥。

- 密钥交换算法：用来实现密钥交换，以保证对称密钥算法、MAC 算法中使用的密钥在 SSL 服务器端和 SSL 客户端之间安全地传递。常用的密钥交换算法通常为非对称密钥算法，如 RSA。

通过本命令可以配置 SSL 客户端策略支持的算法组合。例如，**rsa_des_cbc_sha** 表示 SSL 客户端支持的密钥交换算法为 RSA、数据加密算法为 DES_CBC、MAC 算法为 SHA。

SSL 客户端将本端支持的加密套件发送给 SSL 服务器，SSL 服务器将自己支持的加密套件与 SSL 客户端支持的加密套件比较。如果 SSL 服务器支持的加密套件中存在 SSL 客户端支持的加密套件，则加密套件协商成功；否则，加密套件协商失败。

多次执行本命令，最后一次执行的命令生效。

【举例】

配置 SSL 客户端策略支持的加密套件为：密钥交换算法采用 RSA、数据加密算法采用 128 位 AES_CBC、MAC 算法采用 SHA。

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

【相关命令】

- **ciphersuite**
- **display ssl client-policy**

1.1.9 server-verify enable

server-verify enable 命令用来配置对服务器端进行基于数字证书的身份验证。

undo server-verify enable 命令用取消对服务器端进行基于数字证书的身份验证，默认 SSL 服务器身份合法。

【命令】

server-verify enable

undo server-verify enable

【缺省情况】

SSL 客户端需要对 SSL 服务器端进行基于数字证书的身份验证。

【视图】

SSL 客户端策略视图

【缺省用户角色】

network-admin

【使用指导】

SSL 通过数字证书实现对对端的身份进行验证。数字证书的详细介绍，请参见“安全配置指导”中的“PKI”。

如果执行了 **server-verify enable** 命令，则 SSL 服务器端需要将自己的数字证书提供给客户端，以便客户端对服务器端进行基于数字证书的身份验证。只有身份验证通过后，SSL 客户端才会访问该 SSL 服务器。

【举例】

配置 SSL 客户端需要对 SSL 服务器端进行基于数字证书的身份验证。

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

【相关命令】

- **display ssl client-policy**

1.1.10 session

session 命令用来配置 SSL 服务器上缓存的最大会话数目和 SSL 会话缓存的超时时间。

undo session 命令用来恢复缺省情况。

【命令】

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

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

【缺省情况】

SSL 服务器上缓存的最大会话数目为 500 个，SSL 会话缓存的超时时间为 3600 秒。

【视图】

SSL 服务器端策略视图

【缺省用户角色】

network-admin

【参数】

cachesize *size*：指定 SSL 服务器上缓存的最大会话数目。*size* 为缓存的最大会话数目，取值范围为 100～20480。

timeout *time*：指定 SSL 会话缓存的超时时间。*time* 为会话缓存超时时间，取值范围为 1～4294967295，单位为秒。

【使用指导】

通过 SSL 握手协议协商会话参数并建立会话的过程比较复杂。为了简化 SSL 握手过程，SSL 允许重用已经协商出的会话参数建立会话。为此，SSL 服务器上需要保存已有的会话信息。保存的会话信息的数目和保存时间具有一定的限制：

- 如果缓存的会话数目达到最大值，SSL 将拒绝缓存新协商出的会话。
- 会话保存的时间超过设定的时间后，SSL 将删除该会话的信息。

【举例】

配置 SSL 服务器上缓存的最大会话数目为 600 个，SSL 会话缓存超时时间为 1800 秒。

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

【相关命令】

- **display ssl server-policy**

1.1.11 ssl client-policy

ssl client-policy 命令用来创建 SSL 客户端策略，并进入 SSL 客户端策略视图。如果指定的 SSL 客户端策略已经存在，则直接进入 SSL 客户端策略视图。

undo ssl client-policy 命令用来删除指定的 SSL 客户端策略。

【命令】

ssl client-policy *policy-name*

undo ssl client-policy *policy-name*

【缺省情况】

不存在 SSL 客户端策略。

【视图】

系统视图

【缺省用户角色】

network-admin

【参数】

policy-name：SSL 客户端策略名，为 1～31 个字符的字符串，不区分大小写。

【使用指导】

SSL 客户端策略视图下可以配置 SSL 客户端启动时使用的 SSL 参数，如使用的 PKI 域、支持的加密套件等。只有与应用层协议，如 DDNS（Dynamic Domain Name System，动态域名系统），关联后，SSL 客户端策略才能生效。

【举例】

创建 SSL 客户端策略 policy1，并进入 SSL 客户端策略视图。

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

【相关命令】

- **display ssl client-policy**

1.1.12 ssl renegotiation disable

ssl renegotiation disable 命令用来关闭 SSL 重协商。

undo ssl renegotiation disable 命令用来恢复缺省情况。

【命令】

ssl renegotiation disable

undo ssl renegotiation disable

【缺省情况】

允许 SSL 重协商。

【视图】

系统视图

【缺省用户角色】

network-admin

【使用指导】

关闭 SSL 重协商是指，不允许复用已有的 SSL 会话进行 SSL 快速协商，每次 SSL 协商必须进行完整的 SSL 握手过程。关闭 SSL 重协商会导致系统付出更多的计算开销，但可以避免潜在的风险，安全性更高。

通常情况下，不建议关闭 SSL 重协商。本命令仅用于用户明确要求关闭重协商的场景。

【举例】

关闭 SSL 重协商。

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

1.1.13 ssl server-policy

ssl server-policy 命令用来创建 SSL 服务器端策略，并进入 SSL 服务器端策略视图。如果指定的 SSL 服务器端策略已经存在，则直接进入 SSL 服务器端策略视图。

undo ssl server-policy 命令用来删除指定的 SSL 服务器端策略。

【命令】

ssl server-policy *policy-name*

undo ssl server-policy *policy-name*

【缺省情况】

不存在 SSL 服务器端策略。

【视图】

系统视图

【缺省用户角色】

network-admin

【参数】

policy-name：SSL 服务器端策略名，为 1～31 个字符的字符串，不区分大小写。

【使用指导】

SSL 服务器端策略视图下可以配置 SSL 服务器启动时使用的 SSL 参数，如使用的 PKI 域、支持的加密套件等。只有与 HTTPS 等应用关联后，SSL 服务器端策略才能生效。

【举例】

创建 SSL 服务器端策略 policy1，并进入 SSL 服务器端策略视图。

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

【相关命令】

- **display ssl server-policy**

1.1.14 ssl version disable

ssl version disable 命令用来禁止 SSL 服务器使用指定的 SSL 版本进行 SSL 协商。

undo ssl version disable 命令用来恢复缺省情况。

【命令】

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

undo ssl version { **ssl3.0** | **tls1.0** | **tls1.1** | **tls1.2** } * **disable**

【缺省情况】

SSL 服务器允许使用 SSL3.0、TLS1.0、TLS1.1 和 TLS1.2 版本进行协商。

【视图】

系统视图

【缺省用户角色】

network-admin

【参数】

ssl3.0：SSL 协商的版本为 SSL3.0。

tls1.0：SSL 协商的版本为 TLS1.0。

tls1.1：SSL 协商的版本为 TLS1.1。

tls1.2：SSL 协商的版本为 TLS1.2。

【使用指导】

通过本命令禁止 SSL 服务器使用的 SSL 版本时，请至少允许 SSL 服务器使用一个 SSL 版本进行协商。

同时配置本命令和 SSL 服务器端策略视图下的 **version disable** 命令时，系统视图的配置对所有 SSL 服务器端策略都有效，而 SSL 服务器端策略内的配置只对当前 SSL 服务器端策略有效。对于一个 SSL 服务器端策略来说，优先采用该 SSL 服务器端策略内的配置，只有该 SSL 服务器端策略内未进行配置时，才采用全局的配置。

需要注意的是，如果通过本命令关闭了指定版本的 SSL 协商功能，并不会同时关闭比其更低版本的 SSL 协商功能，例如，**ssl version tls1.1 disable** 命令仅表示关闭了 TLS1.1 版本的 SSL 协商功能，不会同时关闭 TLS1.0 版本。

【举例】

关闭 TLS1.0。

```
<Sysname> system-view
[Sysname] ssl version tls1.0 disable
```

【相关命令】

- **version disable**

1.1.15 version

version 命令用来配置 SSL 客户端策略使用的 SSL 协议版本。

undo version 命令恢复缺省情况。

【命令】

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

undo version

【缺省情况】

SSL 客户端策略使用的 SSL 协议版本为 TLS 1.0。

【视图】

SSL 客户端策略视图

【缺省用户角色】

network-admin

【参数】

ssl3.0：SSL 客户端策略使用的 SSL 协议版本为 SSL 3.0。

tls1.0：SSL 客户端策略使用的 SSL 协议版本为 TLS 1.0。

tls1.1：SSL 客户端策略使用的 SSL 协议版本为 TLS1.1。

tls1.2：SSL 客户端策略使用的 SSL 协议版本为 TLS1.2。

【使用指导】

对安全性要求较高的环境下，建议为不要为 SSL 客户端指定 SSL3.0 版本。

多次执行本命令，最后一次执行的命令生效。

【举例】

配置 SSL 客户端策略使用的 SSL 协议版本为 TLS 1.0。

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

【相关命令】

- **display ssl client-policy**

1.1.16 version disable

version disable 命令用来禁止 SSL 服务器使用指定的 SSL 版本进行 SSL 协商。

undo version disable 命令用来恢复缺省情况。

【命令】

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

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

【缺省情况】

SSL 服务器采用的 SSL 协商版本与全局采用的 SSL 协商版本一致。

【视图】

SSL 服务器端策略视图

【缺省用户角色】

network-admin

【参数】

ssl3.0：SSL 协商的版本为 SSL3.0。

tls1.0：SSL 协商的版本为 TLS1.0。

tls1.1：SSL 协商的版本为 TLS1.1。

tls1.2：SSL 协商的版本为 TLS1.2。

【使用指导】

当用户需要灵活控制 SSL 服务器支持的 SSL 版本时，可以在 SSL 服务端策略视图下执行本命令禁止 SSL 服务器使用指定版本进行协商。

通过本命令禁止 SSL 服务器使用的 SSL 版本时，请至少允许 SSL 服务器使用一个 SSL 版本进行协商。

同时配置本命令和系统视图下的 **ssl version disable** 命令时，系统视图的配置对所有 SSL 服务器端策略都有效，而 SSL 服务器端策略内的配置只对当前 SSL 服务器端策略有效。对于一个 SSL 服务器端策略来说，优先采用该 SSL 服务器端策略内的配置，只有该 SSL 服务器端策略内未进行配置时，才采用全局的配置。

如果通过本命令关闭了指定版本的SSL协商功能，并不会同时关闭比其更低版本的SSL协商功能，例如，**version tls1.0 disable** 命令仅表示关闭了 TLS1.0 版本的 SSL 协商功能，不会同时关闭其他 SSL 版本。

【举例】

配置 SSL 服务端策略禁止使用 TLS 1.0。

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

【相关命令】

- **ssl version disable**