1 SSH Commands

Command	Function
clear ssh ip-block	Clear entries about blocked IP addresses and authentication failures.
crypto key generate	Generate the public key of the SSH server.
crypto key zeroize	Delete the public key of the SSH server.
disconnect ssh	Disconnect an established SSH client session.
disconnect ssh-session	Disconnect a suspended SSH client session.
ip scp client source-interface	Configure the source interface of the Secure copy protocol (SCP) client.
ip scp server enable	Enable the SCP server function.
ip scp server topdir	Configure the transmission path for uploading files to or downloading files from the SCP server.
ip ssh access-class	Configure an access control list (ACL) for the SSH server.
ip ssh authentication-retries	Configure the maximum number of user authentication attempts allowed on the SSH server.
ip ssh cipher-mode	Configure the encryption modes supported by the SSH server.
ip ssh compatible-ssh1x enable	Enable the SSHv1 function.
ip ssh hmac-algorithm	Configure the message authentication algorithms supported by the SSH server.
ip ssh ip-block disable	Disable the IP address blocking function of the SSH server.
ip ssh ip-block failed-times	Configure the number of authentication failures for blocking IP addresses and the time period for counting authentication failures on the SSH server.
ip ssh ip-block reactive	Configure the period for awakening blocked IP addresses.
ip ssh key-exchange	Configure the Diffie–Hellman (DH) key exchange algorithms supported by the SSH server.

ip ssh peer	Associate with the public key file and username of a client.
ip ssh port	Configure the listening port of the SSH server.
ip ssh source-interface	Configure the source interface of the SSH client.
ip ssh time-out	Configure the user authentication timeout time on the SSH server.
ip ssh version	Configure the SSH server version.
ipv6 ssh access-class	Configure an IPv6 ACL for the SSH server.
scp	Upload files to or download files from the remote SCP server.
show crypto key mypubkey	Display partial of the public key information of the SSH server.
show ip ssh	Display effective configurations of the SSH server.
show ssh	Display information about established SSH connections.
show ssh ip-block	Display information about blocked IP addresses and authentication failures.
show ssh-sessions	Display information about established SSH client sessions.
ssh	Establish an encrypted session with a remote network device.
ssh-session	Restore an established SSH client session.

1.1 clear ssh ip-block

Function

Run the **clear ssh ip-block** command to clear entries about blocked IP addresses and authentication failures.

Syntax

clear ssh ip-block { **all** | *ipv4-address* | *ipv6-address* }

Parameter Description

all: Clears all entries about blocked IP addresses and authentication failures.

ipv4-address: IPv4 source address based on which entries about blocked IP addresses and authentication failures are cleared.

ipv6-address: IPv6 source address based on which entries about blocked IP addresses and authentication failures are cleared.

Command Modes

Privileged EXEC mode

Default Level

14

Usage Guidelines

The addresses and address description formats to be cleared can be obtained by running the **show ssh ip-block** command.

After entries about blocked IP addresses are cleared, these IP addresses are awakened immediately and can be used by Secure Shell (SSH) clients to log in to the device.

Examples

The following example clears all entries about blocked IP addresses and authentication failures.

```
Hostname> enable
Hostname# clear ssh ip-block all
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.2 crypto key generate

Function

Run the **crypto key generate** command to generate the public key of the SSH server.

No public key is generated on the SSH server by default.

Syntax

crypto key generate { **dsa** | **ecc** | **rsa** }

Parameter Description

dsa: Generates a Digital Signature Algorithm (DSA) key.

ecc: Generates an Elliptic Curves Cryptography (ECC) key.

rsa: Generates a Rivest-Shamir-Adleman (RSA) key.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

When the SSH server service is required, run this command to generate the public key of the SSH server and run the **enable service ssh-server** command to enable the SSH server function. SSHv1 uses an RSA key, and SSHv2 uses an RSA or a DSA key. If an RSA key is generated, both SSHv1 and SSHv2 can use the key. If a DSA key is generated, only SSHv2 can use the key.

An SSH client uses only one of the ECC, DSA, and RSA public key algorithms for authentication in a connection. However, different clients support different public key algorithms. To ensure that clients can successfully log in to the server, you are advised to generate the ECC, DSA, and RSA key pairs on the server.

The minimum modulus length is 512 bits for the RSA host key and 360 bits for the DSA host key. The maximum modulus length of both is 2048 bits. In SSHv2, some clients (for example, the SCP clients) require that the length of the key generated on the server must be greater than or equal to 768 bits. When configuring RSA and DSA host keys, you are advised to set the host key modulus to 768 bits or larger.

The modulus length of an ECC host key can be 256, 384, or 512 bits, and the default modulus length is 512 bits.

You can run the **show crypto key mypubkey** command to check whether the public information about the RSA key exists. If yes, the RSA key has been generated.

Examples

The following example generates the RSA public key of the SSH server.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# crypto key generate rsa
```

Notifications

When the RSA key of the SSH server is generated for the first time, the following notification will be displayed:

```
Hostname(config)#crypto key generate rsa
Choose the size of the rsa key modulus in the range of 512 to 2048
and the size of the dsa key modulus in the range of 360 to 2048 for your
Signature Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
// When the generation of the RSA key is successful, the following information
will be displayed:
% Generating 512 bit RSA1 keys ...[ok]
% Generating 512 bit RSA keys ...[ok]
// When the generation of the RSA key fails, the following information will be
displayed:
% Generating 512 bit RSA1 keys ...[fail]
% Generating 512 bit RSA keys ...[fail]
```

When the RSA key already exists on the server, the following notification will be displayed. If you select key replacement, you need to re-select the key bits, and information will be displayed, indicating whether the generation is successful. Otherwise, the configuration interface is closed.

```
Hostname(config)#crypto key generate rsa
% You already have RSA keys.
% Do you really want to replace them? [yes/no]:yes
Choose the size of the rsa key modulus in the range of 512 to 2048
and the size of the dsa key modulus in the range of 360 to 2048 for your
Signature Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
```

When the entered number of RSA key bits is not within the range of 512 to 2048, the following notification will be displayed:

```
Hostname(config)#crypto key generate rsa
Choose the size of the rsa key modulus in the range of 512 to 2048
and the size of the dsa key modulus in the range of 360 to 2048 for your
Signature Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:360
sshd: rsa key in the range of 512 to 2048
How many bits in the modulus [512]:2590
sshd: bad data bits
How many bits in the modulus [512]:
```

Common Errors

An incorrect command is used to delete a key. The **no crypto key generate** command instead of the **crypto key zeroize** command is used to delete a key.

Platform Description

N/A

Related Commands

- **enable service** (System Configuration/Basic Management)
- crypto key zeroize

1.3 crypto key zeroize

Function

Run the **crypto key zeroize** command to delete the public key of the SSH server.

Syntax

crypto key zeroize { **dsa** | **ecc** | **rsa** }

Parameter Description

dsa: Deletes a DSA key.

ecc: Delete an ECC key.

rsa: Deletes an RSA key.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used to delete the public key of the SSH server. After the public key is deleted, the SSH server state is **DISABLE**. To disable the SSH server, run the **no enable service ssh-server** command.

You can run the **show crypto key mypubkey** command to check whether the public information about the RSA key exists. If no, the RSA key has been deleted.

Examples

The following example deletes the RSA public key of the SSH server.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# crypto key zeroize rsa
```

Notifications

When the RSA key already exists on the server, the following notification will be displayed:

```
Hostname(config)# crypto key zeroize rsa
% Keys to be removed
% All router certs issued using these keys will also be removed.
Do you really want to remove these keys? [yes/no]:yes
```

```
*Jan 16 06:52:57: %P17050-DEBUG: sshd: delete key file /rsa_private.bin
*Jan 16 06:52:57: %P17050-DEBUG: sshd: delete key file /rsa1_private.bin
```

When no RSA key is generated, the following notification will be displayed:

```
Hostname(config)# crypto key zeroize rsa
% The specified RSA keypair does not exist.
```

Common Errors

The **no** or **default** form of this command is used to delete a key.

Platform Description

N/A

Related Commands

- **show crypto key mypubkey**

1.4 disconnect ssh

Function

Run the **disconnect ssh** command to disconnect an established SSH client session.

Syntax

disconnect ssh { **vty** *session-id* | *session-id* }

Parameter Description

vty *session-id*: Specifies the Virtual Teletype (VTY) session ID of an SSH session to be disconnected. The value range is from 0 to 35.

session-id: Session ID of an SSH client session to be disconnected. The value range is from 0 to 35.

Command Modes

Privileged EXEC mode

Default Level

14

Usage Guidelines

Serving as an SSH server, the device may connect to multiple SSH clients. You can run this command to forcibly disconnect a client from the device. The client disconnection methods are as follows:

- Specify an SSH session ID. To display the SSH session ID of a client, run the **show ssh** command.
- Specify a VTY session ID. To display the VTY session ID of a client, run the **show users** command. This command can be used to disconnect SSH connections only.

Examples

The following example disconnects the SSH client session whose session ID is 1.

```
Hostname> enable
Hostname# disconnect ssh 1
```

The following example disconnects the SSH client session whose VTY session ID is 1.

```
Hostname> enable
Hostname# disconnect ssh vty 1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

- **show users** (System Configuration/Line Configuration)

1.5 disconnect ssh-session

Function

Run the **disconnect ssh-session** command to disconnect a suspended SSH client session.

Syntax

disconnect ssh-session *session-id*

Parameter Description

session-id: Session ID of an SSH client session to be disconnected.

Command Modes

Privileged EXEC mode

Default Level

14

Usage Guidelines

After the device connects to the SSH server as an SSH client, you can run the corresponding command to disconnect an SSH session with the specified session ID. To display information about the SSH connections established by the device as an SSH client, run the **show ssh-session** command.

Examples

The following example disconnects the SSH client session whose ID is 1.

```
Hostname> enable
Hostname# disconnect ssh-session 1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

- **show ssh-session**

1.6 ip scp client source-interface

Function

Run the **ip scp client source-interface** command to configure the source interface of the Secure copy protocol (SCP) client.

Run the **no** form of this command to remove this configuration.

No SCP client source interface is configured by default. SSH packets use the packet sending source address queried based on the route as the source address by default.

Syntax

ip scp client source-interface *interface-type interface-number*

no ip scp client source-interface

Parameter Description

interface-type interface-number: Type and number of the SCP client source interface whose IP address is used as the global source address of the SCP client.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used when the device serves as an SCP client. After the source interface is configured, the SCP client uses the IP address on the interface as the global source address during communication. If no source interface is configured, the source address of SCP packets will be obtained by querying the corresponding route based on the destination address. If no source interface or source IP address is independently specified for an SCP connection, the global configuration is used.

Examples

The following example configures Loopback 1 as the source interface of the SCP client and uses the IP address of Loopback 1 as the global source address of the SCP client.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip scp client source-interface loopback 1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.7 ip scp server enable

Function

Run the **ip scp server enable** command to enable the SCP server function.

Run the **no** form of this command to disable this feature.

The SCP server function is disabled by default.

Syntax

ip scp server enable

no ip scp server enable

Parameter Description

N/A

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

After the SCP server function is configured on the device, users can run the **scp** command to upload files to or download files from the device. Data exchanged during the process is encrypted for security. You can run the **show ip ssh** command to check whether the SCP server function is enabled.

Examples

The following example enables the SCP server function.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip scp server enable
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

- **enable service ssh-server** (System Configuration/Basic Management)

1.8 ip scp server topdir

Function

Run the **ip scp server topdir** command to configure the transmission path for uploading files to or downloading files from the SCP server.

Run the **no** form of this command to remove this configuration.

The default transmission path for file upload and download is **flash:/**.

Syntax

ip scp server topdir { **flash:/***path* | **tmp:/***path* | **usb0:/***path* }

no ip scp server topdir

Parameter Description

flash */path*: Selects the file transmission path from the extended flash space.

tmp */path*: Sets the file transmission path to **tmp/vsd/**.

usb0 */path*: Selects the file transmission path from Universal Serial Bus (USB) disk 0. This option is supported only when the device has one USB port with an extended USB flash drive inserted.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used to configure the transmission path for uploading and downloading files.

Examples

The following example sets the transmission path **tmp:/dir** for uploading files to and downloading files from the SCP server.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip scp server topdir tmp:/dir
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.9 ip ssh access-class

Function

Run the **ip ssh access-class** command to configure an access control list (ACL) for the SSH server.

Run the **no** form of this command to remove this configuration.

No ACL is configured on the SSH server by default.

Syntax

ip ssh access-class { *acl-name* | *acl-number* }

no ip ssh access-class

Parameter Description

acl-name: Name of a standard or an extended ACL used for the SSH server. The value is a case-sensitive string of 1 to 99 characters.

acl-number: Number of a standard or an extended ACL used for the SSH server. The value range is from 1 to 199 or from 1300 to 2699.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command can be used to perform ACL filtering for all connections to the SSH server. In line mode, ACL filtering is performed only for specific lines. However, ACL filtering rules of the SSH server are effective to all SSH connections.

Examples

The following example configures ACL testv4 for the SSH server.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh access-class testv4
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.10 ip ssh authentication-retries

Function

Run the **ip ssh authentication-retries** command to configure the maximum number of user authentication attempts allowed on the SSH server.

Run the **no** form of this command to remove this configuration.

The default maximum number of authentication attempts allowed on the SSH server is **3**.

Syntax

ip ssh authentication-retries *retry-times*

no ip ssh authentication-retries

Parameter Description

retry-times: Maximum number of authentication attempts allowed. The value range is from 0 to 5.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used to configure the maximum number of authentication attempts allowed on the SSH server. If authentication still does not succeed when the maximum number of user authentication attempts is reached, user authentication fails.

Examples

The following example sets the maximum number of authentication attempts allowed on the SSH server to **2**.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh authentication-retries 2
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.11 ip ssh cipher-mode

Function

Run the **ip ssh cipher-mode** command to configure the encryption modes supported by the SSH server.

Run the **no** form of this command to remove this configuration.

The encryption modes supported by the SSH server are **ctr** and **gcm** by default.

Syntax

ip ssh cipher-mode { **cbc** | **ctr** | **gcm** | **others** } *

no ip ssh cipher-mode

Parameter Description

cbc: Configures the SSH server to support encryption mode cipher block chaining (CBC). The corresponding encryption algorithms are DES-CBC, 3DES-CBC, AES-128-CBC, AES-192-CBC, AES-256-CBC, and Blowfish-CBC.

ctr: Configures the SSH server to support encryption mode counter (CTR). The corresponding encryption algorithms are AES128-CTR, AES192-CTR, and AES256-CTR.

gcm: Configures the SSH server to support encryption mode Galois/Counter Mode (GCM), with Galois Message Authentication Code (GMAC). The corresponding encryption algorithms are AES128-GCM and AES256-GCM.

others: Configures the SSH server to support encryption mode "others". The corresponding encryption algorithm is RC4.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

SSHv1 supports encryption algorithms DES-CBC, 3DES-CBC, and Blowfish-CBC.

SSHv2 supports encryption algorithms AES128-CTR, AES192-CTR, AES256-CTR, DES-CBC, 3DES-CBC, AES-128-CBC, AES-192-CBC, AES-256-CBC, Blowfish-CBC, AES128-GCM, AES256-GCM, and RC4.

These algorithms can be grouped into four encryption modes: CBC, CTR, GCM (with GMAC), and others.

As the cryptography continuously develops, it is approved that encryption algorithms in the CBC and others modes can be decrypted in a limited period of time. Therefore, organizations or companies that have high security requirements can set the encryption modes supported by the SSH server to CTR and GCM to enhance the security level of the SSH server.

When the CBC, CTR, or GCM mode and algorithms supported in the mode are configured, the configuration result is still the original mode. For example, if **ip ssh cipher cbc 3des-cbc** is configured, the actual configuration result is the CBC mode.

Examples

The following example sets the encryption mode supported by the SSH server to the CTR mode.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh cipher-mode ctr
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.12 ip ssh compatible-ssh1x enable

Function

Run the **ip ssh compatible-ssh1x enable** command to enable the SSHv1 function.

Run the **no** form of this command to disable this feature.

The SSHv1 function is disabled by default.

Syntax

ip ssh compatible-ssh1x enable

no ip ssh compatible-ssh1x enable

Parameter Description

N/A

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

N/A

Examples

The following example enables the SSHv1 function.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh compatible-ssh1x enable
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.13 ip ssh hmac-algorithm

Function

Run the **ip ssh hmac-algorithm** command to configure the message authentication algorithms supported by the SSH server.

Run the **no** form of this command to remove this configuration.

SSHv1 servers do not support any message authentication algorithms, and SSHv2 servers support the MD5, SHA1, SHA1-96, MD5-96, sha2-256, and sha2-512 message authentication algorithms by default.

Syntax

ip ssh hmac-algorithm { **md5** | **md5-96** | **sha1** | **sha1-96** | **sha2-256** | **sha2-512** }

no ip ssh hmac-algorith

Parameter Description

Md5:5: Sets the message authentication algorithm supported by the SSH server to MD5.

md5-96: Sets the message authentication algorithm supported by the SSH server to MD5-96.

sha1: Sets the message authentication algorithm supported by the SSH server to SHA1.

sha1-96: Sets the message authentication algorithm supported by the SSH server to SHA1-96.

sha2-256: Sets the message authentication algorithm supported by the SSH server to SHA2-256.

sha2-512: Sets the message authentication algorithm supported by the SSH server to SHA2-512.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

N/A

Examples

The following example sets the message authentication algorithm supported by the SSH server to SHA1.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh hmac-algorithm sha1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.14 ip ssh ip-block disable

Function

Run the **ip ssh ip-block disable** command to disable the IP address blocking function of the SSH server.

Run the **no** form of this command to enable this feature.

The IP address blocking function of the SSH server is enabled by default.

Syntax

ip ssh ip-block disable

no ip ssh ip-block disable

Parameter Description

N/A

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

When the number of authentication failures for login through SSH reaches the configured limit in an authentication failure count period, the source IP address is blocked. That is, the SSH client that uses this source IP address is not allowed to log in to the device to prevent the device being attacked. The SSH client can log in to the device only after the IP address is awakened.

Examples

The following example disables the IP address blocking function on the SSH server.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh ip-block disable
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.15 ip ssh ip-block failed-times

Function

Run the **ip ssh ip-block failed-times** command to configure the number of authentication failures for blocking IP addresses and the time period for counting authentication failures on the SSH server.

Run the **no** form of this command to remove this configuration.

The allowed maximum number of authentication failures is **6**, and the time period for counting authentication failures is **5** minutes by default.

Syntax

ip ssh ip-block failed-times *failed-times* **period** *period-time*

no ip ssh ip-block failed-times *failed-times* **period** *period-time*

Parameter Description

failed-times: Number of authentication failures for blocking IP addresses. The value range is from 1 to 10.

period-time: Time period for counting authentication failures, in minutes. The value range is from 1 to 120.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

After the IP address blocking function is enabled, if the number of consecutive authentication failures for device login through SSH reaches the configured limit in an authentication failure count period, the source IP address is blocked. If the number of consecutive authentication failures does not reach the configured limit in an authentication failure count period, or one authentication is successful, the authentication failures are cleared.

Examples

The following example sets the number of authentication failures for blocking IP addresses to 3 and the time period for counting authentication failures to 3 minutes on the SSH server.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh ip-block failed-times 3 period 3
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.16 ip ssh ip-block reactive

Function

Run the **ip ssh ip-block reactive** command to configure the period for awakening blocked IP addresses.

Run the **no** form of this command to remove this configuration.

Blocked IP addresses are awakened every **5** minutes by default.

Syntax

ip ssh ip-block reactive *reactive-interval*

no ip ssh ip-block reactive

Parameter Description

reactive-interval: Period for awakening blocked IP addresses. The value range is from 1 to 1000.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

After the time period for awakening the blocked source IP address reaches, the entry with the blocked source IP address is cleared. An SSH client can use this IP address to log in to the device.

Examples

The following example sets the time period for awakening blocked IP addresses on the SSH server to 3 minutes.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh ip-block reactive 3
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.17 ip ssh key-exchange

Function

Run the **ip ssh key-exchange** command to configure the Diffie–Hellman (DH) key exchange algorithms supported by the SSH server.

Run the **no** form of this command to remove this configuration.

Orion_B26Q SSHv2 servers support diffie-hellman-group-exchange-sha1, diffie-hellman-group14-sha1, ecdh_sha2_nistp256, ecdh_sha2_nistp384, and ecdh_sha2_nistp521 for key exchange, and SSHv1 servers support none by default.

Syntax

ip ssh key-exchange { **dh_group_exchange_sha1** | **dh_group14_sha1** | **dh_group1_sha1** | **ecdh_sha2_nistp256** | **ecdh_sha2_nistp384** | **ecdh_sha2_nistp521** }

no ip ssh key-exchange

Parameter Description

dh_group_exchange_sha1: Sets the DH key exchange algorithm to diffie-hellman-group-exchange-sha1. The default key length is **2048** bytes, which is not configurable.

dh_group14_sha1: Sets the DH key exchange algorithm to diffie-hellman-group14-sha1. The key length is 2048 bytes.

dh_group1_sha1: Sets the DH key exchange algorithm to diffie-hellman-group1-sha1. The key length is 1024 bytes.

ecdh_sha2_nistp256: Sets the DH key exchange algorithm to ecdh_sha2_nistp256. The key length is 256 bytes.

ecdh_sha2_nistp384: Sets the DH key exchange algorithm to ecdh_sha2_nistp384. The key length is 384 bytes.

ecdh_sha2_nistp521: Sets the DH key exchange algorithm to ecdh_sha2_nistp521. The key length is 512 bytes.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

The SSHv1 server does not support any DH key exchange algorithm. The SSHv2 server supports the following DH key exchange algorithms: diffie-hellman-group-exchange-sha1, diffie-hellman-group14-sha1, diffie-hellman-group1-sha1, ecdh_sha2_nistp256, ecdh_sha2_nistp384, and ecdh_sha2_nistp521. You can select DH key exchange algorithms supported by the SSH server as required.

Examples

The following example sets the DH key exchange algorithm supported by the SSH server to diffie-hellman-group14-sha1.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh key-exchange dh_group14_sha1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.18 ip ssh peer

Function

Run the **ip ssh peer** command to associate with the public key file and username of a client.

Run the **no** form of this command to remove this configuration.

Syntax

ip ssh peer *username* **public-key** { **dsa** | **ecc** | **rsa** } *filename-path*

no ip ssh peer *username* **public-key** { **rsa** | **dsa** | **ecc** }

Parameter Description

username: Username of a client.

dsa: Sets the public key type to DSA.

ecc: Sets the public key type to ECC.

rsa: Sets the public key type to RSA.

filename-path: Path where the public key file is stored.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

N/A

Examples

The following example associates username **test** with the RSA public key file **flash:rsa.pub**.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh peer test public-key rsa flash:rsa.pub
```

Notifications

When the number of associated public key files exceeds the maximum value 1024, the following notification will be displayed:

```
Hostname(config)# ip ssh peer test public-key rsa flash:rsa.pub
%% Too many public-keys, system support max public key 1024
```

When the name of the associated public key file is not entered, the following notification will be displayed:

```
Hostname(config)#ip ssh peer test public-key rsa flash:
% Invalid file name
```

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.19 ip ssh port

Function

Run the **ip ssh port** command to configure the listening port of the SSH server.

Run the **no** form of this command to remove this configuration.

The default listening port of the SSH server is port **22**.

Syntax

ip ssh port *ssh-monitor-port*

no ip ssh port

Parameter Description

ssh-monitor-port: Listening port of the SSH server. The value range is from 1025 to 65535.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

N/A

Examples

The following example sets the listening port of the SSH server to **10000**.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh port 10000
```

Notifications

When the configured port is the same as the current value, the following notification will be displayed:

```
Hostname(config)# ip ssh port 22
% SSH tcp-port has been 22
```

When a port in the listening state is configured as the listening port of the SSH server, the following notification will be displayed:

```
Hostname(config)# ip ssh port 10000
% SSH open tcp-port(10000) failed, please use another tcp-port,otherwise the
system will use the old tcp-port(22)!
```

When an error occurs after the configured listening port starts listening, the following notification will be displayed:

```
Hostname(config)# ip ssh port 10000
% SSH change to tcp-port(10000) fail!
```

When a listening port is successfully configured, the following notification will be displayed:

```
Hostname(config)# ip ssh port 10000
% SSH change to tcp-port(10000) success!
```

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.20 ip ssh source-interface

Function

Run the **ip ssh source-interface** command to configure the source interface of the SSH client.

Run the **no** form of this command to remove this configuration.

No SSH client source interface is configured by default.

Syntax

ip ssh source-interface *interface-type interface-number*

no ip ssh source-interface

Parameter Description

interface-type interface-number: Type and number of the SSH client source interface whose IP address is used as the global source address of the SSH client.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used to specify the source interface when the device serves as an SSH client, and the IP address of which will be used as the global source address of the SSH client. When the **ssh** command is used to connect to an SSH server, this global configuration will be used if no source interface or source address is specified for this connection. When no SSH client source interface is configured, SSH packets use the packet sending source address queried based on the route as the source address.

Examples

The following example configures Loopback 1 as the source interface of an SSH client.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh source-interface loopback 1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.21 ip ssh time-out

Function

Run the **ip ssh time-out** command to configure the user authentication timeout time on the SSH server.

Run the **no** form of this command to remove this configuration.

The default user authentication timeout time on the SSH server is **120** seconds.

Syntax

ip ssh time-out *timeout-time*

no ip ssh time-out

Parameter Description

timeout-time: User authentication timeout time, in seconds. The value range is from 1 to 120.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used to configure the user authentication timeout time on the SSH server. The SSH server starts the timer after receiving a user connection request. If authentication does not succeed after 120 seconds, authentication fails.

Examples

The following example sets the user authentication timeout time on the SSH server to 100 seconds.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh time-out 100
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.22 ip ssh version

Function

Run the **ip ssh version** command to configure the SSH server version.

Run the **no** form of this command to remove this configuration.

The SSH server is compatible with the SSHv1 and SSHv2 clients by default.

Syntax

ip ssh version *version-type*

no ip ssh version

Parameter Description

version-type: Version of the SSH server. The value is 1 or 2. The value **1** indicates that the SSH server supports only connection requests from SSHv1 clients, and the value **2** indicates that the SSH server supports only connection requests from SSHv2 clients.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command is used to configure the version supported by the SSH server. If the *version-type* parameter is not specified, the SSH server is compatible with SSHv1 and SSHv2 clients. That is, both SSHv1 and SSHv2 clients can connect to the SSH server. If only SSHv1 or SSHv2 is configured, only the SSH client of the configured version can be connected to the SSH server.

Examples

The following example sets the version supported by the SSH server to SSHv2 only.

```
Hostname> enable
Hostname# configure terminal
Hostname(config)# ip ssh version 2
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.23 ipv6 ssh access-class

Function

Run the **ipv6 ssh access-class** command to configure an IPv6 ACL for the SSH server.

Run the **no** form of this command to remove this configuration.

No IPv6 ACL is configured on the SSH server by default.

Syntax

ipv6 ssh access-class *acl-name*

no ipv6 ssh access-class

Parameter Description

acl-name: Name of an IPv6 ACL used for the SSH server.

Command Modes

Global configuration mode

Default Level

14

Usage Guidelines

This command can be used to perform IPv6 ACL filtering for all connections to the SSH server. In line mode, IPv6 ACL filtering is performed only for specific lines. However, IPv6 ACL filtering rules of the SSH server are effective to all SSH connections.

Examples

The following example configures IPv6 ACL **testv6** for the SSH server.

```
Hostname> enable
Hostname# configure terminal
```

```
Hostname(config)# ipv6 ssh access-class testv6
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.24 scp

Function

Run the **scp** command to upload files to or download files from the remote SCP server.

Syntax

scp [**oob**] [**-v** { **1** | **2** } | **-c** { **3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc** | **aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-gcm** | **aes256-gcm** } | **-m** { **hmac-md5-96** | **hmac-md5-128** | **hmac-sha1-96** | **hmac-sha1-160** | **hmac-sha2-256** | **hmac-sha2-512** } | **-p** *port-num*] * *source-file destination-file* [**/source** { **ip** *ipv4-address* | **ipv6** *ipv6-address* | **interface** *interface-type interface-number* }] [**/vrf** *vrf-name*]

Parameter Description

oob: Connects to the remote SCP server through out-of-band communication (over the MGMT port typically). This option is valid only when the device has an MGMT port.

-v { **1 | 2** }: Configures the SSH version. The value **1** indicates SSHv1, and the value **2** indicates SSHv2. If this parameter is not specified, SSHv2 is used.

-c: Configures the data encryption algorithm. During algorithm negotiation, the SSH client sends only the user-specified encryption algorithm to the server. If the server does not support the user-specified encryption algorithm, the server closes the SSH connection. If this parameter is not specified, the SSH client sends all supported algorithms to the server during algorithm negotiation.

-c 3des: Sets the data encryption algorithm to 3DES.

-**c aes128-cbc**: Sets the data encryption algorithm to AES128-CBC (128-bit key).

-**c aes192-cbc**: Sets the data encryption algorithm to AES192-CBC (192-bit key).

-**c aes256-cbc**: Sets the data encryption algorithm to AES256-CBC (256-bit key).

-**c aes128-ctr**: Sets the data encryption algorithm to AES128-CTR (128-bit key).

-**c aes192-ctr**: Sets the data encryption algorithm to AES192-CTR (192-bit key).

-**c aes256-ctr**: Sets the data encryption algorithm to AES256-CTR (256-bit key).

-**c aes128-gcm**: Sets the data encryption algorithm to AES128-GCM (128-bit key).

-**c aes256-gcm**: Sets the data encryption algorithm to AES256-GCM (256-bit key).

-m: Configures the hash-based message authentication code (HMAC) algorithm. During algorithm negotiation, the SCP client sends only the user-specified HMAC algorithm to the server. If the server does not support the user-specified HMAC algorithm, the server closes the SSH connection. If this parameter is not specified, the SSH client sends all supported algorithms to the server during algorithm negotiation. Supported algorithms include hmac-md5-96, hmac-md5-128, hmac-sha1-96, hmac-sha1-160, hmac-sha2-256, and hmac-sha2-512.

-p *port-num*: Configures the destination port in packets sent from the client to the server. The value range is from 0 to 65535. If this parameter is not specified, the destination port is port 22.

source-file destination-file: File copied to the destination path, which can be from the remote server to the device or from the device to the remote server. *source-file* indicates the path where the source file is stored. *destination-file* indicates the path where a file is copied to. Files on the remote server are displayed in *username*@host:/*filename* format, and files on the device are displayed in *path*:/*filename* format. The formats are as follows:

- flash:/filename: Extended flash space.
- usb0:/filename: Extended USB disk 0. This option is supported when the device has one USB port with a USB flash drive inserted.
- tmp:/filename: Temporary tmp/vsd/ directory.

via *mgmt-name*: Specifies the MGMT port used by the SSH server when the oob option is specified.

/source: Specifies the source IP address or interface used by an SCP client.

ip *ipv4-address*: Specifies the source IPv4 address used by an SCP client.

ipv6 *ipv6-address*: Specifies the source IPv6 address used by an SCP client.

interface *interface-type interface-number*: Specifies the source interface used by an SCP client.

/vrf *vrf-name*: Specifies the virtual routing and forwarding (VRF) routing table to be displayed.

Command Modes

Privileged EXEC mode

Default Level

1

Usage Guidelines

When the device serves as an SCP client, the device can run the **scp** command to establish a connection to the SCP server to upload files to and download files from the SCP server.

SSHv1 does not support the HMAC algorithm. If both SSHv1 and the HMAC algorithm are specified, the HMAC configuration will be ignored.

Examples

The following example downloads the **config.text** file from a remote SCP server whose IP address is 192.168.23.122 to the local device using username **admin** and saves the file as **flash:/config.text**.

```
Hostname> enable
Hostname#scp admin@192.168.23.122:/config.text flash:/config.text
```

The following example uploads the **flash:/config.text** file from the local device to the remote SCP server whose IP address is 192.168.23.122 using username **admin** and saves the file as **config.text**.

```
Hostname> enable
Hostname# scp flash:/config.text admin@192.168.23.122:/config.text
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.25 show crypto key mypubkey

Function

Run the **show crypto key mypubkey** command to display partial of the public key information of the SSH server.

Syntax

show crypto key mypubkey { **dsa** | **ecc** | **rsa** }

Parameter Description

dsa: Displays the DSA key information.

ecc: Displays the ECC key information.

rsa: Displays the RSA key information.

Command Modes

All modes except the user EXEC mode

Default Level

14

Usage Guidelines

This command is used to display the public key information of the SSH server, including the key generation time, key name, and partial of key content.

Examples

The following example displays partial RSA key information of the SSH server.

```
Hostname> enable
Hostname# show crypto key mypubkey rsa
% Key pair was generated at: 7:1:25 UTC Jan 16 2013
 Key name: RSA1 private
 Usage: SSH Purpose Key
 Key is not exportable.
 Key Data:
         AAAAAwEA AQAAAEEA 2m6H/J+2 xOMLW5MR 8tOmpW1I XU1QItVN mLdR+G7O Q10kz+4/
         /IgYR0ge 1sZNg32u dFEifZ6D zfLySPqC MTWLfw==
% Key pair was generated at: 7:1:25 UTC Jan 16 2013
 Key name: RSA private
 Usage: SSH Purpose Key
 Key is not exportable.
 Key Data:
         AAAAAwEA AQAAAEEA 0E5w2H0k v744uTIR yZBd/7AM 8pLItnW3 XH3LhEEi BbZGZvn3
         LEYYfQ9s pgYL0ZQf S0s/GY0X gJOMsc6z i8OAkQ==
```

Table 1-1Output Fields of the show crypto key mypubkey Command

Field	Description
Key pair was generated at	Key generation time
Key name	Key name
Usage	Key use description
Key Data	Partial of public key content

Notifications

N/A

Platform Description

N/A

Related Commands

N/A

1.26 show ip ssh

Function

Run the **show ip ssh** command to display effective configurations of the SSH server.

Syntax

show ip ssh

Parameter Description

N/A

Command Modes

All modes except the user EXEC mode

Default Level

14

Usage Guidelines

This command is used to display effective configurations of the SSH server, including the version, whether the SSH server function is enabled, port number, encryption mode, message authentication algorithm, authentication timeout time, and maximum number of authentication attempts allowed.

If an SSH version is configured but the corresponding server key is not generated, a message indicating that the SSH version is unavailable will be displayed.

Examples

The following example displays effective configurations of the SSH server when the SSH server and SCP server functions are disabled.

```
Hostname> enable
Hostname# show ip ssh
SSH Disable - version 1.99
please generate rsa and dsa key to enable SSH
SSH Port:                22
SSH Cipher Mode:         cbc,ctr,others
SSH HMAC Algorithm:      md5-96,md5,sha1-96,sha1
Authentication timeout: 120 secs
Authentication retries: 3
SSH SCP Server:          disabled
SSH dh-exchange min-len: 2048
SSH ip-block:            disabled
```

The following example displays effective configurations of the SSH server when the SSH server and SCP server functions are enabled.

```
Hostname> enable
Hostname# show ip ssh
SSH Enable - version 1.99
SSH Port:                22
SSH Cipher Mode:         cbc,ctr,others
SSH HMAC Algorithm:      md5-96,md5,sha1-96,sha1
Authentication timeout: 120 secs
Authentication retries: 3
SSH SCP Server:          enabled
SSH dh-exchange min-len: 2048
SSH ip-block:            enabled
```

Table 1-1Output Fields of the show ip ssh Command

Field	Description
SSH Enable/Disable	Whether the SSH server function is enabled
version 1|2	SSH version supported by the SSH server

Field	Description
please generate rsa and dsa key to enable SSH	Whether the RSA/DSA public key is generated to enable the SSH server function
SSH Port	Listening port of the SSH server
SSH Cipher Mode	Encryption mode of the SSH server
SSH HMAC Algorithm	Message authentication algorithm of the SSH server
Authentication timeout	User authentication timeout time
Authentication retries	Maximum number of authentication attempts allowed
SSH SCP Server enabled/disabled	Whether the SSH SCP server function is enabled
SSH dh-exchange min-len	Minimum key length negotiated by the key exchange algorithm of the SSH server
SSH ip-block enabled/disabled	Whether the IP address blocking function is enabled on the SSH server

Notifications

N/A

Platform Description

N/A

Related Commands

N/A

1.27 show ssh

Function

Run the **show ssh** command to display information about established SSH connections.

Syntax

show ssh

Parameter Description

N/A

Command Modes

All modes except the user EXEC mode

Default Level

14

Usage Guidelines

This command is used to display information about established SSH connections, including the VTY number occupied by a connection, SSH version, encryption algorithm, message authentication algorithm, connection status, and username.

Examples

The following example displays information about established SSH connections.

```
Hostname> enable
Hostname# show ssh
Connection Version Encryption       Hmac         Compress   State
Username
         0     1.5 blowfish                      zlib       Session started test
         1     2.0 aes256-cbc       hmac-sha1    zlib       Session started test
```

Table 1-1Output Fields of the show ssh Command

Field	Description
Connection	VTY number occupied by a connection
Version	SSH version supported by an SSH client
Encryption	Encryption algorithm
Hmac	Message authentication algorithm
Compress	Compression algorithm
State	Connection status
Username	Username

Notifications

N/A

Platform Description

N/A

Related Commands

N/A

1.28 show ssh ip-block

Function

Run the **show ssh ip-block** command to display information about blocked IP addresses and authentication failures.

Syntax

show ssh ip-block { **all** | **list** }

Parameter Description

all: Displays information about all blocked IP addresses and authentication failures.

list: Displays information about blocked IP addresses.

Command Modes

All modes except the user EXEC mode

Default Level

14

Usage Guidelines

This command is used to display information about all blocked IP addresses and authentication failures. Blocked IP address information includes the source IPv4 or IPv6 addresses and remaining time for awakening blocked IP addresses. Authentication failure information includes the source IPv4 or IPv6 addresses, status, and number of authentication failures.

Examples

The following example displays information about all blocked IP addresses and authentication failures.

```
Hostname> enable
Hostname# show ssh ip-block all
-------------------------
IP Address                                     State       Auth-fail Count
-------------------------
172.30.31.16                                   AUTH FAILED     3
172.30.31.17                                   BLOCKED         6
-------------------------
```

Table 1-1Output Fields of the show ssh ip-block all Command

Field	Description
IP Address	Source IPv4 or IPv6 address
State	Status ● **AUTH FAILED**: Authentication fails but the blocking conditions are not met. ● **BLOCKED**: Blocking conditions are met.
Auth-fail Count	Number of authentication failures

The following example displays information about blocked IP addresses.

```
Hostname> enable
Hostname# show ssh ip-block list
------------------------
```

```
IP Address                                      UnBlock Interval (Seconds)
------------------------
172.30.31.17                                    296
------------------------
```

Table 1-2Output Fields of the show ssh ip-block list Command

Field	Description
IP Address	Source IPv4 or IPv6 address
UnBlock Interval (Seconds)	Remaining time for awakening blocked IP addresses, in seconds

Notifications

N/A

Platform Description

N/A

Related Commands

N/A

1.29 show ssh-sessions

Function

Run the **show ssh-sessions** command to display information about established SSH client sessions.

Syntax

show ssh-sessions

Parameter Description

N/A

Command Modes

All modes except the user EXEC mode

Default Level

14

Usage Guidelines

This command is used to display information about established SSH client sessions, including the VTY number occupied by a connection, SSH version, and server address.

Examples

The following example displays information about established SSH client sessions.

```
Hostname> enable
Hostname# show ssh-sessions
Connect No.  SSH Version Server Address
-----  -----  ----
0            2.0          192.168.23.122
1            1.5          192.168.23.122
```

Table 1-1Output Fields of the show ssh-sessions Command

Field	Description
Connect No.	Client No.
SSH Version	SSH version of a session
Server Address	IP address of the remote SSH server

Notifications

N/A

Platform Description

N/A

Related Commands

N/A

1.30 ssh

Function

Run the **ssh** command to establish an encrypted session with a remote network device.

Syntax

ssh [**oob**] [**-v** { **1** | **2** } | **-c** { **3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc** | **aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-gcm** | **aes256-gcm** } | **-l** *username* | **-m** { **hmac-md5-96** | **hmac-md5-128** | **hmac-sha1-96** | **hmac-sha1-160** | **hmac-sha2-256** | **hmac-sha2-512** } | **-p** *port-num*] * { *ip-address* | *hostname* } [**via** *mgmt-name*] [**/source** { **ip** *ipv4-address* | **ipv6** *ipv6-address* | **interface** *interface-type interface-number* }] [**/vrf** *vrf-name*]

Parameter Description

oob: Connects to the remote SSH server through out-of-band communication (over the MGMT port typically). This option is valid only when the device has an MGMT port.

-v { **1 | 2** }: Configures the SSH version. The value **1** indicates SSHv1, and the value **2** indicates SSHv2. If this parameter is not specified, SSHv2 is used.

-c: Configures the data encryption algorithm. During algorithm negotiation, the SSH client only sends the user-specified encryption algorithm to the server. If the server does not support the user-specified encryption algorithm, the server closes the SSH connection. If this parameter is not specified, the SSH client sends all supported algorithms to the server during algorithm negotiation.

-**c 3des**: Sets the data encryption algorithm to 3DES.

-**c aes128-cbc**: Sets the data encryption algorithm to AES128-CBC (128-bit key).

-**c aes192-cbc**: Sets the data encryption algorithm to AES192-CBC (192-bit key).

-**c aes256-cbc**: Sets the data encryption algorithm to AES256-CBC (256-bit key).

-**c aes128-ctr**: Sets the data encryption algorithm to AES128-CTR (128-bit key).

-**c aes192-ctr**: Sets the data encryption algorithm to AES192-CTR (192-bit key).

-**c aes256-ctr**: Sets the data encryption algorithm to AES256-CTR (256-bit key).

-**c aes128-gcm**: Sets the data encryption algorithm to AES128-GCM (128-bit key).

-**c aes256-gcm**: Sets the data encryption algorithm to AES256-GCM (256-bit key)

-l *username*: Specifies the username used for login.

-m: Configures the HMAC algorithm. During algorithm negotiation, the SCP client sends only the user-specified HMAC algorithm to the server. If the server does not support the user-specified HMAC algorithm, the server closes the SSH connection. If this parameter is not specified, the SSH client sends all supported algorithms to the server during algorithm negotiation. Supported algorithms include hmac-md5-96, hmac-md5-128, hmac-sha1-96, hmac-sha1-160, hmac-sha2-256, and hmac-sha2-512.

-p *port-num*: Configures the destination port in packets sent from the client to the server. The value range is from 0 to 65535. If this parameter is not specified, the destination port is port 22.

ip-address: IPv4/IPv6 address of the remote server.

hostname: IPv4/IPv6 host name of the remote server.

via *mgmt-name*: Specifies the MGMT port used by the SSH server when the oob option is configured.

/source: Specifies the source IP address or interface used by an SCP client.

ip *ipv4-address*: Specifies the source IPv4 address used by an SCP client.

ipv6 *ipv6-address*: Specifies the source IPv6 address used by an SCP client.

interface *interface-type interface-number*: Specifies the source interface used by an SCP client.

/vrf *vrf-name*: Specifies the virtual routing and forwarding (VRF) routing table to be displayed.

Command Modes

Privileged EXEC mode

Default Level

14

Usage Guidelines

When the device serves as an SSH client, the device can run the **ssh** command to establish a connection to the SSH server. This connection is similar to but more secure than a Telnet connection due to the authentication and encrypted transmission features. Different versions support different parameters. During parameter configuration, note the following:

- SSHv1 supports only the DES (56-bit key) and 3DES (168-bit key) encryption algorithms.

- SSHv2 supports the following Advanced Encryption Standards (AES): AES128-CBC, AES192-CBC, AES256-CBC, AES128-CTR, AES192-CTR, AES256-CTR, AES128-GCM, and AES256-GCM.
- SSHv1 does not support the HMAC algorithm.
- If you specify an unmatched encryption or authentication algorithm when selecting an SSH version, the unmatched algorithm will be ignored when a connection is established.

Examples

The following example uses username **admin** to log in to a device that provides the SSH server service and whose IP address is 192.168.23.122 through SSH.

```
Hostname> enable
Hostname# ssh -l admin 192.168.23.122
```

The following example uses username **admin** to log in to a device that provides the SSH server service and whose IP address is 192.168.23.122 through SSHv2. The encryption algorithm is set to **aes128-cbc**, and the authentication algorithm is set to **hmac-md5-128**.

```
Hostname> enable
Hostname# ssh -v 2 -c aes128-cbc  -m hmac-md5-128 -l admin 192.168.23.122
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A

1.31 ssh-session

Function

Run the **ssh-session** command to restore an established SSH client session.

Syntax

ssh-session *session-id*

Parameter Description

session-id: ID of an established SSH client session.

Command Modes

Privileged EXEC mode

Default Level

14

Usage Guidelines

After the device establishes an SSH session with the SSH server as a client, you can press Ctrl+Shift+6+X to exit the session temporarily. After exiting the session, you can run the corresponding command to restore the session.

Examples

The following example restores the SSH client session whose session ID is 1.

```
Hostname> enable
Hostname# ssh-session 1
```

Notifications

N/A

Common Errors

N/A

Platform Description

N/A

Related Commands

N/A