Configuring SSH Services and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) services and Telnet on Cisco MDS devices.

This chapter includes the following sections:

Information About SSH Services

Secure Shell (SSH) is a protocol that provides a secure, remote connection to the Cisco NX-OS CLI. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. You can use SSH keys for the following SSH options:

- SSH2 using RSA
- SSH2 using DSA

Starting from Cisco MDS NX-OS Release 8.2(1), SHA2 fingerprint hashing is supported on all Cisco MDS devices by default.

A secure SSH connection, with a RSA key is available as default on all Cisco MDS 9000 Series Switches. If you require a secure SSH connection with a DSA key, you need to disable the default SSH connection, generate a dsa key, and then enable the SSH connection (see the Generating the SSH Server Key Pair , on page 4 section).

Use the **ssh key** command to generate a server key.

Caution If you are logging in to a switch through SSH and you have issued the **aaa authentication login default none** command, you must enter one or more key strokes to log in. If you press the **Enter** key without entering at least one keystroke, your log in will be rejected.

For more information about configuring SSH services, see Configuring SSH Services and Telnet, on page 1

SSH Server

You can use the SSH server to enable an SSH client to make a secure, encrypted connection to a Cisco MDS device. SSH uses strong encryption for authentication. The SSH server in the Cisco MDS NX-OS software can interoperate with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, LDAP, and the use of locally stored usernames and passwords.

SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco MDS device to make a secure, encrypted connection to another Cisco MDS device or to any other device that runs the SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco MDS device. You can use SSH server keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algrorithm (DSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts two types of key-pairs for use by SSH version 2:

- The **dsa** option generates the DSA key-pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA key-pair for the SSH version 2 protocol.

By default, the Cisco NX-OS software generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)
- Public Key Certificate in Privacy-Enhanced Mail (PEM)

Caution If you delete all of the SSH keys, you cannot start the SSH services.

SSH Authentication Using Digital Certificates

SSH authentication on the Cisco MDS 9000 Family switches provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that vouches for the origin and integrity of a message. It contains encryption keys for secured communications and is "signed" by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through query or notification. Verification of certificates is successful if the certificates are from any of the trusted CAs.

You can configure your switch for either SSH authentication using an X.509 certificate or SSH authentication using a Public Key Certificate, but not both. If either of them is configured and the authentication fails, you will be prompted for a password.

Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

The Telnet server is disabled by default on the Cisco NX-OS device.

Configuring SSH

This section describes how to configure SSH.

Configuring SSH Name

To configure the name of a primary SSH connection for a user, follow these steps:

Before you begin

Enable feature SSH.

Procedure

	Command or Action	Purpose
Step 1	switch#**ssh name** *ssh-nameuser-nameip-address* **Example:** `switch# ssh name myhost user 192.168.1.1`	Configures a SSH name for a primary SSH connection.
Step 2	switch#**no ssh name** **Example:** `switch# no ssh name myhost user 192.168.1.1`	(Optional) Deletes the name for the SSH connection.

	Command or Action	Purpose
Step 3	switch#**show ssh names** **Example:** `switch# show ssh names`	(Optional) Displays the names of the SSH connections.

Configuring SSH Connect

To configure SSH connection for a user, follow these steps:

Before you begin

- Enable feature SSH.
- Configure SSH name. For information on configuring SSH name, refer to Configuring SSH Name, on page 3.

Procedure

	Command or Action	Purpose
Step 1	switch#**ssh connectdummy** **Example:** `switch# ssh connect myhost`	Configures a SSH connection for a SSH name.
Step 2	switch#**no ssh connect** **Example:** `switch# no ssh connect myhost`	(Optional) Deletes the SSH connection.
Step 3	switch#**show ssh names** **Example:** `switch# show ssh names`	(Optional) Displays the names of the SSH connections.

Generating the SSH Server Key Pair

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits. Ensure that you have an SSH server key pair with the appropriate version before enabling the SSH service. Generate the SSH server key pair according to the SSH client version used. The number of bits specified for each key pair ranges from 768 to 2048.

Starting from Cisco MDS NX-OS Release 8.2(1), the minimum RSA key size in FIPS mode should be 2048 bits.

For information about RSA key-pair maximums and defaults, see the Table 1 Maximum Limits for CA and Digital Certificate and Table 2 Default CA and Digital Certificate Parameters

The SSH service accepts two types of key pairs for use by SSH version 2.

- The **dsa** option generates the DSA key pair for the SSH version 2 protocol.

- The **rsa** option generates the RSA keypair for the SSH version 2 protocol.

Caution If you delete all of the SSH keys, you cannot start a new SSH session.

To generate the SSH server key pair, follow these steps:

Procedure

Step 1 switch# **configure terminal**

Enters configuration mode.

Step 2 switch(config)# **ssh key dsa 1024**

Example:

```
generating dsa key.....
generated dsa key
```

Generates the DSA server key pair.

Step 3 switch(config)# **ssh key rsa 1024**

Example:

```
generating rsa key.....
generated rsa key
```

Generates the RSA server key pair.

Step 4 switch(config)# **no ssh key rsa 1024**

Example:

```
cleared RSA keys
```

Clears the RSA server key pair configuration.

Specifying the SSH Key

You can specify an SSH key to log in using the SSH client without being prompted for a password. You can specify the SSH key in three different formats:

- Open SSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Specifying the SSH Key in OpenSSH

To specify or delete the SSH key in OpenSSH format for a specified user, follow these steps:

Procedure

Step 1 switch# **configure terminal**

Enters configuration mode.

Step 2 switch(config)# **username admin sshkey ssh-rsa [illegible]**

Specifies the SSH key for the user account (admin).

Step 3 switch(config)# **no username admin sshkey ssh-rsa [illegible]**

(Optional) Deletes the SSH key for the user account (admin).

Specifying the SSH Key in IETF SECSH

To specify or delete the SSH key in IETF SECSH format for a specified user, follow these steps:

Procedure

Step 1 switch# **copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub**

Downloads the file containing the SSH key in IETF SECSH format.

Step 2 switch# **configure terminal**

Enters configuration mode.

Step 3 switch(config)# **username admin sshkey file bootflash:secsh_file.pub**

Specifies the SSH key for the user account (admin).

Step 4 switch(config)# **no username admin sshkey file bootflash:secsh_file.pub**

(Optional) Deletes the SSH key for the user account (admin).

Specifying the SSH Key in Public Key Certificate in PEM

To specify or delete the SSH key in PEM-formatted Public Key Certificate form for a specified user, follow these steps:

Procedure

Step 1 switch# **copy tftp://10.10.1.1/cert.pem bootflash:cert.pem**

Downloads the file containing the SSH key in PEM-formatted Public Key Certificate form.

Step 2 switch# **configure terminal**

switch(config)#

Enters configuration mode.

Step 3 switch(config)# **username admin sshkey file bootflash:cert.pem**

Specifies the SSH key for the user account (usam).

Step 4 switch(config)# **no username admin sshkey file bootflash:cert.pem**

(Optional) Deletes the SSH key for the user account (usam).

Configuring a Login Grace Time for SSH Connections

You can configure the login grace time for SSH connections from remote devices to your Cisco MDS devices. This configures the grace time for clients to authenticate themselves. If the time to login to the SSH session exceeds the specified grace time, the session disconnects and you will have to login again.

Note Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature ssh**
	Example:
	`switch# feature ssh`
	`switch(config)#`
		Enables SSH.

Step 3	**ssh login-gracetime** *number* **Example:** `switch(config)# ssh login-gracetime 120`	Configures the login grace time in seconds for SSH connections from remote devices to your Cisco MDS device. Specify the time allowed for successful authentication to the SSH server before SSH disconnects the session. The default login grace time is 120 seconds. The range is from 10 to 600. **Note** The **no** form of this command removes the configured login grace time and resets it to the default value of 120 seconds.

	Command or Action	Purpose
Step 4	(Optional) **exit** **Example:** `switch(config)# exit`	Exits global configuration mode.
Step 5	(Optional) **show running-config security** **Example:** `switch(config)# show running-config security`	Displays the configured SSH login grace time.
Step 6	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	Displays the configured or default SSH login grace time.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Overwriting a Generated Key Pair

If the SSH key pair option is already generated for the required version, you can force the switch to overwrite the previously generated key pair.

To overwrite the previously generated key pair, follow these steps:

Procedure

Step 1 switch# **configure terminal**

Enters configuration mode.

Step 2 switch(config)# **ssh key dsa force**

Example:

```
switch(config)# ssh key dsa 512 force
deleting old dsa key.....
generating dsa key.....
generated dsa key
```

Tries to set the server key pair. If a required server key pair is already configured, use the **force** option to overwrite that server key pair. Deletes the old DSA key and sets the server key pair using the new bit specification.

Configuring the Maximum Number of SSH Login Attempts

You can configure maximum number of SSH login attempts. If the user exceeds the maximum number of permitted attempts, the session disconnects.

Note The total number of login attempts includes attempts through public-key authentication, certificate-based authentication, and password-based authentication. If public-key authentication is enabled, it takes priority. If only certificate-based and password-based authentication are enabled, certificate-based authentication takes priority. If you exceed the configured number of login attempts through all of these methods, a message appears indicating that too many authentication failures have occurred.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**ssh login-attempts** *number* **Example:** `switch(config)# ssh login-attempts 5`	Configures the maximum number of times that a user can attempt to log into an SSH session. The default maximum number of login attempts is 3. The range is from 1 to 10. **Note** The **no** form of this command removes the previous login attempts value and sets the maximum number of login attempts to the default value of 3. We recommend that you configure the SSH login attempts value to more than 1.
Step 3	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	Displays the configured maximum number of SSH login attempts.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SSH Cipher Mode

Cisco MDS 9000 switches support strong algorithms by default. You can set the cipher mode for configuring SSH.

To enable weak cipher mode, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters the global configuration mode.

Step 2	**ssh cipher-mode weak**
	Example:
	`switch(config)# ssh cipher-mode weak`
	`switch(config)#`
		Enable weak ciphers.

Customizing SSH Cryptographic Algorithms

Cisco MDS 9000 switches support strong algorithms by default. You can choose to remain with the default mode that enables only strong algorithms as defined by Cisco PSB or allow all supported algorithms. Note that these algorithms are applicable to the incoming server connections. You can also configure support for SSH key exchange algorithms, message authentication codes (MACs), key types, and ciphers.

Note Customizing SSH cryptographic algorithms are supported with x86-based MDS 9000 series switches only. However, this feature is not supported with MDS 9250i, MDS 9148S, and MDS 9396S switches.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters the global configuration mode.

Step 2	**ssh kexalgos** [**all** | WORD]
	Example:
	`switch(config)# ssh kexalgos all`
	Example:
	`switch(config)# ssh kexalgos ecdh-sha2-nistp384`
	`switch(config)# no ssh kexalgos ecdh-sha2-nistp384`
		Use the **all** keyword to enable all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys.
		Supported KexAlgorithmns are:
		• curve25519-sha256
		• diffie-hellman-group-exchange-sha256
		• diffie-hellman-group1-sha1
		• diffie-hellman-group14-sha1

	Command or Action	Purpose
		• diffie-hellman-group1-sha1 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 To enable or disable particular algorithm use the **show ssh kexalgos** command to find the keyword or algorithm name.
Step 3	**ssh macs** [**all** | WORD] **Example:** `switch(config)# ssh macs all`	Enables all supported MACs which are the message authentication codes used to detect traffic modification. Supported MACs are: • hmac-sha1 • hmac-sha2-256 • hmac-sha2-512
Step 4	**ssh ciphers** [**all** | WORD] **Example:** `switch(config)# ssh ciphers all`	Use the **all** keyword to enable all supported ciphers to encrypt the connection. Supported ciphers are: • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com To enable only the **aes256-gcm** cipher, use the **aes256-gcm** keyword. **Note** Ensure that **ssh cipher-mode weak** is disabled before enabling **aes256-gcm**.
Step 5	**ssh keytypes** [**all** | WORD] **Example:** `switch(config)# ssh keytypes all`	Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client.

	Command or Action	Purpose
		Supported key types are: • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521 • ssh-dss • ssh-rsa • rsa-sha2-256

Clearing SSH Hosts

The **clear ssh hosts** command clears the existing list of trusted SSH hosts and reallows you to use SCP/SFTP along with the **copy** command for particular hosts.

When you use SCP/SFTP along with the **copy** command, a list of trusted SSH hosts are built and stored within the switch (see the following example).

Using SCP/SFTP to Copy Files

```
switch# copy scp://abcd@10.10.1.1/users/abcd/abc

bootflash:abc The authenticity of host '10.10.1.1 (10.10.1.1)'
can't be established.
RSA1 key fingerprint is 01:29:62:16:33:ff:f7:dc:cc:af:aa:20:f8:20:a2:db.
Are you sure you want to continue connecting (yes/no)? yes
Added the host to the list of known hosts
(/var/home/admin/.ssh/known_hosts). [SSH key information about the host is
stored on the switch]
abcd@10.10.1.1's password:
switch#
```

Using SCP/SFTP to Copy Files—Error Caused by SSH Key Change

If a host's SSH key changes before you use SCP/SFTP along with the **copy** command, you will receive an error (see the following example).

```
switch# copy scp://apn@10.10.1.1/isan-104

bootflash:isan-ram-1.0.4
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA1 host key has just been changed.
The fingerprint for the RSA1 key sent by the remote host is
36:96:ca:d7:29:99:79:74:aa:4d:97:49:81:fb:23:2f.
Please contact your system administrator.
Add correct host key in /mnt/pss/.ssh/known_hosts to get rid of this
message.
Offending key in /mnt/pss/.ssh/known_hosts:2
```

```
RSA1 host key for 10.10.1.1 has changed and you have requested strict
checking.
```

Enabling SSH or Telnet Service

By default, the SSH service is enabled with an RSA key.

To enable or disable the SSH or Telnet service, follow these steps:

Procedure

Step 1 switch# **configure terminal**

Enters configuration mode.

Step 2 switch(config)# **feature ssh**

Enables the use of the SSH service.

Step 3 switch(config)# **no feature ssh**

(Optional) Disables (default) the use of the SSH service.

Step 4 switch(config)# **feature telnet**

Enables the use of the Telnet service.

Step 5 switch(config)# **no feature telnet**

(Optional) Disables (default) the use of the Telnet service.

Displaying SSH Protocol Status

Displays SSH Protocol Status

Use the **show ssh server** command to display the status of the SSH protocol (enabled or disabled) and the versions that are enabled for that switch (see the following example).

```
switch# show ssh server

ssh is enabled
version 1 enabled
version 2 enabled
```

Displays Server Key-Pair Details

Use the **show ssh key** command to display the server key-pair details for the specified key or for all keys, (see the follwoing example).

Note From Cisco MDS NX-OS Release 8.2(1), the fingerprint value displayed in the output of the show ssh key [rsa | dsa] command will be in SHA-2 value, as SHA-2 value is considered to be secure

```
switch# show ssh key

rsa Keys generated:Thu Feb 16 14:12:21 2017

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDQ7si46R6sYsWNBRFV+v662vbY6wmr9QMBU4N+BK8F
Iez+7U+2VRdyz1Mykbb1HF/2zth3ZWuTkrTX+8cMnVdcwlfrvWY3g7CLmq5Wkxkq5PiSHsG9pnKM0ubw
Unqc4HYrjEiwJKAR2OBAylfH1ajf7wYGQbOiTQMeMyo2nQK8yQ==

bitcount:1024
fingerprint:
SHA256:D4F+Tl7R3fVunGz9A4GKGLWMQ0r4YRbzf5GfNwy1neg
**************************************
dsa Keys generated:Tue Feb 28 07:47:04 2017

ssh-dss AAAAB3NzaC1kc3MAAACBAJan5V/6YiKQZG2SCChmn9Mu5EbUQoTuCDyTCIYM35ofzh+dEALU
11XZrkGl7V2Hfbgp57dcTya1gjeNOzwU32oOvbA8osJ3BWpIePkZv+/t0feOz4LUhBz85ccmQeLJQ86R
UeJ6pAFsq+yk4XB/l5qMv9SN/QY0/95gCIDt8Uq7AAAAFQDZUMiLvTZwIwajLdu8OtLfB1vmuwAAAIAE
7rIwgUlrDTqmzvRdrmayYM2cGfwL4x+8gGpGe2kZoedFzv4vmmW2npD0E8qTWs4nD0k7cioTjdgLXQoZ
yaQIpIEtd+qS8NHuCrtRguVuDDCEOMTlhwNwL0iCHm08YgJIR3ho+V/nm5ko4kp7jA5eOh/9P/Rr4hCO
aZBNxPcSewAAAIBhcNhaVDYvEri7JCH8DbiZr30z2P3PpIQ8YWpHcOE7CBXkp++HjMFUKd9HJlIwd4bA
81tTkTfSxkPBc9ocHOv1vusVufj423HFjcBIODixY76gJzqlt3aNs54MDfiYxyJLh6yp6LZffDn4t2HF
x7tZSb4UJQKHdNR05d63Pybdbg==

bitcount:1024
fingerprint:
SHA256:kbHB73ZEhZaqJp/J68f1nfN9pJaQUkdHt0iKJc0c+Ao
```

Note If you are logging in to a switch through SSH and you have issued the **aaa authentication login default none CLI** command, you must enter one or more key strokes to log in. If you press the **Enter** key without entering at least one keystroke, your log in will be rejected.

Passwordless File copy and SSH

Secure Shell (SSH) public key authentication can be used to achieve password free logins. SCP and SFTP uses SSH in the background and hence these copy protocols can be used for a password free copy with public key authentication. The NX-OS version only supports the SCP and STFP client functionality.

You can create an RSA/DSA identity which can be used for authentication with ssh. The identity will consist of two parts: public and private keys. The public and the private keys are generated by the switch or can be generated externally and imported to the switch. For import purposes, the keys should be in OPENSSH format.

To use the key on a host machine hosting an SSH server, you must transfer the public key file to the machine and add the contents of it to the file 'authorized_keys' in your ssh directory (e.g. $HOME/.ssh) on the server. For import and export of private keys, the key will be protected by encryption. You will be asked to enter a Passphrase for the same. If you enter a passphrase, the private key is protected by encryption. If you leave the password field blank, the key will not be encrypted.

If you need to copy the keys to another switch, you will have to export the keys out of the switch to a host machine and then import the same to other switches from that machine.

- The key files are persistent across reload.

To import and export the key pair, the following CLIs are provided. The CLI command to generate the ssh user key pairs on the switch is defined as follows:

Procedure

Step 1 switch# **configure terminal**

Enters configuration mode.

Step 2 switch(config)# **username admin keypair generate rsa**

Example:

```
generating rsa key(1024 bits).....
generated rsa key
```

Generates public and private RSA keys for the account (admin). It then stores the key files in the home directory of the specified user. Use the force option to overwrite that server keypair.

Note This example is for RSA keys. Replace rsa with dsa for DSA keys.

Step 3 switch(config)# **no username admin keypair generate rsa**

(Optional) Deletes the public and private RSA keys for the account (admin).

Step 4 switch# **show username admin keypair**

Example:

```
**************************************
rsa Keys generated: Thu Jul 9 11:10:29 2009
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD
0P8boZElTfJFx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvq
srU9TBypYDPQkR/+Y6cKubyFWVxSBG/NHztQc3+QC1zdkIxGNJ
bEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbqS33GZsCAX6v0=
bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************
could not retrieve dsa key information
**************************************
```

Shows the public key for the account (admin).

Step 5 switch(config)# **username admin keypair export bootflash:key_rsa rsa**

Example:

```
Enter Passphrase:
switch(config)# dir
 951 Jul 09 11:13:59 2009 key_rsa
 221 Jul 09 11:14:00 2009 key_rsa.pub
```

Exports the keypair from the user’s (admin’s) home directory to the bootflash memory.

The key pair (both public and private keys) will be exported to the specified location. The user will be prompted to enter a Passphrase which will encrypt the private key. The private key will be exported as the file name specified in the uri and the public key will be exported with the same file name followed by a “.pub” extension.

The user can now copy this key pair to any switch, and also copy the public file to the home directory of the SCP server.

Step 6 switch(config)# **username admin keypair import bootflash:key_rsa rsa**

Example:

```
Enter Passphrase:
switch(config)# show username admin keypair
**************************************
rsa Keys generated: Thu Jul 9 11:10:29 2009
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD
0P8boZElTfJFx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvq
srU9TBypYDPQkR/+Y6cKubyFWVxSBG/NHztQc3+QC1zdkIxGNJ
bEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbqS33GZsCAX6v0=
bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************
could not retrieve dsa key information
**************************************
```

Imports the keypair to the home directory of the switch.

The uri given here must be the uri of the private key and the public should be present on the same location with extension “.pub”. The user will be prompted for the passphrase, and the same passphrase must be entered as was used to encrypt the key.

Once the private keys are copied to the switches which need to do passwordless copy to a server, and that server has the public key copied to its authorized_keys file in home directory, the user will be able to do passwordless file copy and ssh to the server from the switches.

Note To copy the public key to the authorized_keys file on the server, user can also copy the key from the show command mentioned above.

Step 7 server# **cat key_rsa.pub** >> **$HOME/.ssh/ authorized_keys**

Appends the public key stored in key_rsa.pub to the authorized_keys file on the SCP server. The passwordless ssh/scp is then enabled from the switch to this server using the standard ssh and scp commands.

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 1: Default SSH Parameters

Parameters	Default
SSH server	Enabled

Parameters	Default
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Maximum number of SSH login attempts	3
SCP server	Disabled
SFTP server	Disabled