[root@localhost .ssh]# ssh root@192.168.100.101

The authenticity of host '192.168.100.101 (192.168.100.101)' can't be established.

ECDSA key fingerprint is 42:c5:02:f7:ae:bb:85:8b:a9:c9:85:be:62:39:9b:0d.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.100.101' (ECDSA) to the list of known hosts.

root@192.168.100.101's password:

The authenticity of host '192.168.100.101 (192.168.100.101)' can't be established.

ECDSA key fingerprint is 42:c5:02:f7:ae:bb:85:8b:a9:c9:85:be:62:39:9b:0d.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.100.101' (ECDSA) to the list of known hosts.

Above red text is host authentication.

After confirming host authentication, you still have to input password.

But after confirming host authentication, on client, public host key of server is added to ~/.ssh/known\_hosts

Exit

Go to server and remove private key ‘ssh\_host\_ecdsa\_key’

On client, try to connect to server ssh root@192.168.100.101 => can’t connect

## Defining Server Host Key

The key pair used for server authentication is defined on the server in the sshd2\_config file with the following parameters:

|  |
| --- |
| HostkeyFile hostkey  PublicHostKeyFile hostkey.pub |

During the setup process, one RSA key pair (with the file names hostkey and hostkey.pub) is generated and stored in the /etc/ssh2/ directory. By default this key pair is used for server authentication. Make sure that only the user running sshd2 has access to the private key.

In SSH Tectia Server for IBM z/OS, each server daemon can have only one host key pair. This is different from SSH Tectia Server on other platforms.

By default, the server uses a public key with the filename of the private key plus the extension .pub. The PublicHostKeyFile keyword has to be defined only if the public-key file is stored with a different filename.

### Generating the Server Host Key Pair

The host public-key pair (1536-bit RSA) is generated during the setup of SSH Tectia Server (Section [Running the Setup Script](https://support.ssh.com/manuals/server-zos-admin/55/Running_the_Setup_Script.html#SERVER-RUN-SETUP)). You only need to regenerate it if you want to change your host key pair.

SSH Tectia Server for IBM z/OS includes a program that generates a key pair, ssh-keygen2, which is located in /usr/lpp/ssh2/bin.

Generate the key pair for the server in such a way that the private key has no passphrase (option -P). The server will then start up without any operator interaction to enter a passphrase. Protect the key with file system access rules. The private key (/etc/ssh2/hostkey) must be accessible only by the SSHD2 user.

To (re)generate the host key, perform the following tasks:

1. Use su to switch to a UID 0 user (if you are not already logged in as one).
2. Run ssh-keygen2 to generate the host key, for example:
3. > /usr/lpp/ssh2/bin/ssh-keygen2 -t rsa -P /etc/ssh2/hostkey

This will generate a 2048-bit RSA key pair without a passphrase and store it under /etc/ssh2. For more information on the key generation options, see the ssh-keygen2 man page (Appendix [ssh-keygen2](https://support.ssh.com/manuals/server-zos-admin/55/ssh-keygen2.html#MAN-SSH-KEYGEN2)).

1. Restart the server as instructed in Section [Restarting sshd2](https://support.ssh.com/manuals/server-zos-admin/55/Restarting_and_Stopping_sshd2.html#RESTART-SSHD2).

### Using an OpenSSH Server Host Key

SSH Tectia Server for IBM z/OS can use a key created with OpenSSH as the server host key. The key must be configured with the HostKeyFile option in sshd2\_config or have the default file names, hostkey and hostkey.pub.

Both RSA and DSA keys with key lengths from 512 (OpenSSH requires at least 768 for DSA keys) to 4096 bits or more are supported.

### Notifying the Users of the Host Key Change

Administrators that have other users connecting to their server should notify the users of the host key change. If you do not, the users will receive a warning the next time they connect because the host key the users have saved on their disk for your server does not match the host key now being actually provided by your server. The users may not know how to respond to this error.

You can run the following to display a fingerprint of your new public host key which you can provide to your users via some unalterable method (for example, by a digitally signed e-mail or by displaying the fingerprint on secured bulletin board):

> /usr/lpp/ssh2/bin/ssh-keygen2 -F hostkey.pub

When the users connect and receive the error message about the host key having changed, they can compare the fingerprint of the new key with the fingerprint you have provided in your e-mail, and ensure that they are connecting to the correct sshd2 daemon. Inform your users to notify you if the fingerprints do not match, or if they receive a message about a host key change and do not receive a corresponding message from you notifying them of the change.

This procedure can help ensure that you do not become a victim of a man-in-the-middle attack, as your users will notify you if the host key fingerprints do not match. You will also be aware if the users encounter host key change messages when you have not regenerated your host key pair.

If you want to avoid the risk associated with the first connection, you can do one of the following:

* As an administrator of both the client and server machines, you can copy the server public key in advance to the /etc/ssh2/hostkeys directory on the client computer as key\_22\_<hostname>.pub (where <hostname> is the hostname the client uses when it connects to the server).

In this case, manual fingerprint check is not needed, and you can also set the StrictHostKeyChecking keyword in the ssh2\_config file on the client to yes. After this, ssh2 will refuse to connect if the server's public key is not in the /etc/ssh2/hostkeys directory.

* The server administrator can also send the public host key to the users via an unalterable method. The users can save the key in their $HOME/.ssh2/hostkeys directory as key\_22\_<hostname>.pub. If all remote host keys are received in this manner, the StrictHostKeyChecking option can be enabled on the client.