

OpenSSH Server

1. [Introduction](#)
2. [Installation](#)
3. [Configuration](#)
4. [SSH Keys](#)
5. [References](#)

Introduction

This section of the Ubuntu Server Guide introduces a powerful collection of tools for the remote control of, and transfer of data between, networked computers called *OpenSSH*. You will also learn about some of the configuration settings possible with the OpenSSH server application and how to change them on your Ubuntu system.

OpenSSH is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling, or transferring files between, computers. Traditional tools used to accomplish these functions, such as *telnet* or *rcp*, are insecure and transmit the user's password in cleartext when used. OpenSSH provides a server daemon and client tools to facilitate secure, encrypted remote control and file transfer operations, effectively replacing the legacy tools.

The OpenSSH server component, *sshd*, listens continuously for client connections from any of the client tools. When a connection request occurs, *sshd* sets up the correct connection depending on the type of client tool connecting. For example, if the remote computer is connecting with the *ssh* client application, the OpenSSH server sets up a remote control session after authentication. If a remote user connects to an OpenSSH server with *scp*, the OpenSSH server daemon initiates a secure copy of files between the server and client after authentication. OpenSSH can use many authentication methods, including plain password, public key, and *Kerberos* tickets.

Installation

Installation of the OpenSSH client and server applications is simple. To install the OpenSSH client applications on your Ubuntu system, use this command at a terminal prompt:

```
sudo apt-get install openssh-client
```

To install the OpenSSH server application, and related support files, use this command at a terminal prompt:

```
sudo apt-get install openssh-server
```

The *openssh-server* package can also be selected to install during the Server Edition installation process.

Configuration

You may configure the default behavior of the OpenSSH server application, *sshd*, by editing the file `/etc/ssh/sshd_config`. For information about the configuration directives used in this file, you may view the appropriate manual page with the following command, issued at a terminal prompt:

```
man sshd_config
```

There are many directives in the *sshd* configuration file controlling such things as communication settings, and authentication modes. The following are examples of configuration directives that can be changed by editing the `/etc/ssh/sshd_config` file.

Prior to editing the configuration file, you should make a copy of the original file and protect it from writing so you will have the original settings as a reference and to reuse as necessary.

Copy the `/etc/ssh/sshd_config` file and protect it from writing with the following commands, issued at a terminal prompt:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
sudo chmod a-w /etc/ssh/sshd_config.original
```

The following are examples of configuration directives you may change:

1. To set your OpenSSH to listen on TCP port 2222 instead of the default TCP port 22, change the Port directive as such:

```
Port 2222
```

2. To have `sshd` allow public key-based login credentials, simply add or modify the line:

```
PubkeyAuthentication yes
```

If the line is already present, then ensure it is not commented out.

3. To make your OpenSSH server display the contents of the `/etc/issue.net` file as a pre-login banner, simply add or modify the line:

```
Banner /etc/issue.net
```

In the `/etc/ssh/sshd_config` file.

After making changes to the `/etc/ssh/sshd_config` file, save the file, and restart the `sshd` server application to effect the changes using the following command at a terminal prompt:

```
sudo /etc/init.d/ssh restart
```

Many other configuration directives for `sshd` are available to change the server application's behavior to fit your needs. Be advised, however, if your only method of access to a server is `ssh`, and you make a mistake in configuring `sshd` via the `/etc/ssh/sshd_config` file, you may find you are locked out of the server upon restarting it. Additionally, if an incorrect configuration directive is supplied, the `sshd` server may refuse to start, so be extra careful when editing this file on a remote server.

SSH Keys

SSH *keys* allow authentication between two hosts without the need of a password. SSH key authentication uses two keys, a *private* key and a *public* key.

To generate the keys, from a terminal prompt enter:

```
ssh-keygen -t dsa
```

This will generate the keys using the *Digital Signature Algorithm (DSA)* method. During the process you will be prompted for a password. Simply hit *Enter* when prompted to create the key.

By default the *public* key is saved in the file `~/.ssh/id_dsa.pub`, while `~/.ssh/id_dsa` is the *private* key. Now copy the `id_dsa.pub` file to the remote host and append it to `~/.ssh/authorized_keys` by entering:

```
ssh-copy-id username@remotehost
```

Finally, double check the permissions on the `authorized_keys` file, only the authenticated user should have read and write permissions. If the permissions are not correct change them by:

```
chmod 600 .ssh/authorized_keys
```

You should now be able to SSH to the host without being prompted for a password.

References

1. [Ubuntu Wiki SSH](#) page.
2. [OpenSSH Website](#)
3. [Advanced OpenSSH Wiki Page](#)