Most Important ssdh Configuration Options

SSH, or Secure SHell, allows the user of one computer on the network to connect to and use the shell of another over a secure connection. It consists of two basic components, the SSH client used to connect to a remote server, and the SSH server daemon (sshd) running on the server to accept SSH connections from elsewhere.

Configuration for the sshd server is found in the /etc/ssh/sshd_config file. The client configuration is in /etc/ssh/ssh_config.

Here are some of the most important configuration options for an SSH server:

**Port**

The default port for SSH is 22, which is typically fine, but it could be changed to some other available port if you want to throw an extra obstacle to would be unauthorized attempts to connect.

**PermitRootLogin**

This option can be set to either yes or no. If it is set to yes then it will allow using SSH to log in directly as root by running something like ssh root@host.net from the client computer. It may be a good idea to set this to “no” in order to close even the remote possibility of someone cracking through the root password and wreaking havoc. Just a decent precaution.

**AllowUsers**

With this option you can set to allow only some users on the system to connect via SSH. For multiple users separate them by
spaces. For example:

```
AllowUsers james kevin
```

That will allow only james and kevin users to connect.

**LoginGraceTime**

This is the amount of time SSH will wait on the user to authenticate before cutting the connection. By default it is set to 120, or 2 minutes, but it can be reduced if you want to diminish chances of someone successfully attempting a brute force attack.

**PasswordAuthentication**

Set to yes by default this enable password authentication, which definitely should be enabled unless you have public key authentication enabled, because otherwise basically anyone could connect.

**PubkeyAuthentication**

An alternative or an addition to PasswordAuthentication setting this to yes could significantly increase security. For it to work you also need an option that specifies where the authorized keys are:

```
AuthorizedKeysFile ~/.ssh/authorized_keys
```

**TCPKeepAlive**

Set to yes by default this option checks the status of your connection by sending keepalive messages to the client. If there are network interruptions it will then close the connection rather than continue to use up resources.

**See Also**

- [SSH_port](#)
- ssh command
- Port Forwarding