

Top 20 OpenSSH Server Best Security Practices - good article

Article Number: 399 | Rating: Unrated | Last Updated: Mon, Oct 1, 2018 11:30 PM

last updated May 16, 2018 in Categories [CentOS](#), [Debian Linux](#), [fedora linux](#), [FreeBSD](#), [Gentoo Linux](#), [Howto](#), [Linux](#), [Networking](#), [package management](#), [RedHat/Fedora Linux](#), [Security](#), [Suse Linux](#), [Sys admin](#), [Ubuntu Linux](#), [UNIX](#)



OpenSSH is the implementation of the SSH protocol. OpenSSH is recommended for remote login, making backups, remote file transfer via scp or sftp, and much more. SSH is perfect to keep confidentiality and integrity for data exchanged between two networks and systems. However, the main advantage is server authentication, through the use of public key cryptography. From time to time there are [articles](#) about OpenSSH zero day exploit. This [page](#) shows how to secure your OpenSSH server running on a Linux or Unix-like system to improve sshd security.

-
- TCP port – 22
 - OpenSSH server config file – sshd_config (located in /etc/ssh/)
-

OpenSSH server supports various authentication. It is recommended that you use public key based authentication. First, create the key pair using following ssh-keygen command on your local desktop/laptop:

DSA and RSA 1024 bit or lower ssh keys are considered weak. Avoid them. RSA keys are chosen over ECDSA keys when backward compatibility is a concern with ssh clients. All ssh keys are either ED25519 or RSA. Do not use any other type.

```
$ ssh-keygen -t key_type -b bits -C "comment"
```

```
$ ssh-keygen -t ed25519 -C "Login to production cluster at xyz corp"
```

```
$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa_aws_$(date +%Y-%m-%d) -C "AWS key for abc corp clients"
```

Next, install the public key using ssh-copy-id command:

```
$ ssh-copy-id -i /path/to/public-key-file user@host
```

```
$ ssh-copy-id user@remote-server-ip-or-dns-name
```

```
$ ssh-copy-id vivek@rhel7-aws-server
```

When prompted supply user password. Verify that ssh key based login working for you:

```
$ ssh vivek@rhel7-aws-server
```

```
Terminal
File Edit View Terminal Tabs Help
Terminal
cowpoke.conf
cracklib
cron.d
cron.daily
cron.hourly
cron.monthly
crontab
cron.weekly
crypttab
cups
cvs-cron.conf
cvs-pserver.conf
darkstat
dbus-1
dconf.conf
dconf-custom.conf
debconf.conf
debian_version
default
defoma
deluser.conf
vivek@vivek-desktop:/etc$
gnome-system-tools
gnome-vfs-2.0
gnome-vfs-mime-magic
google
gre.d
groff
group
group-
grub.d
gshadow
gshadow-
gssapi_mech.conf
gtk
gtk-2.0
hal
hddtemp.db
hdparm.conf
hesiod.conf
highlight
host.conf
hostname
ld.so.preload
lftp.conf
libao.conf
libgda
libgda-3.0
libpaper.d
lighttpd
locale.alias
localtime
logcheck
login.defs
logrotate.conf
logrotate.d
lsb-base
lsb-base-logging.sh
lsb-release
ltrace.conf
lvm
lynx.cfg
lynx.lss
magic
```

For more info on ssh public key auth see:

- [keychain: Set Up Secure Passwordless SSH Access For Backup Scripts](#)
- [sshpass: Login To SSH Server / Provide SSH Password Using A Shell Script](#)
- [How To Setup SSH Keys on a Linux / Unix System](#)
- [How to upload ssh public key to as authorized_key using Ansible DevOPS tool](#)

Before we disable root user login, make sure regular user can log in as root. For example, allow vivek user to login as root using the sudo command.

Allow members of group sudo to execute any command. [Add user vivek to sudo group:](#)

```
$ sudo adduser vivek sudo
```

Verify group membership with [id command](#)

```
$ id vivek
```

Allows people in group wheel to run all commands on a CentOS/RHEL and Fedora Linux server. Use the usermod command to add the user named vivek to the wheel group:

```
$ sudo usermod -aG wheel vivek
```

```
$ id vivek
```

Test it and make sure user vivek can log in as root or run the command as root:

```
$ sudo -i
```

```
$ sudo /etc/init.d/sshd status
```

```
$ sudo systemctl status httpd
```

Once confirmed disable root login by adding the following line to sshd_config:

```
PermitRootLogin no
```

```
ChallengeResponseAuthentication no
```

```
PasswordAuthentication no
```

```
UsePAM no
```

See “[How to disable ssh password login on Linux to increase security](#)” for more info.

All password-based logins must be disabled. Only public key based logins are allowed.

Add the following in your sshd_config file:

```
AuthenticationMethods publickey
```

```
PubkeyAuthentication yes
```

Older version of SSHD on CentOS 6.x/RHEL 6.x user should use the following setting:

```
PubkeyAuthentication yes
```

By default, all systems user can login via SSH using their password or public key.

Sometimes you create UNIX / Linux user account for FTP or email purpose. However, those users can log in to the system using ssh. They will have full access to system tools including compilers and scripting languages such as Perl, Python which can open network ports and do many other fancy things. Only allow root, vivek and jerry user to use the

system via SSH, add the following to sshd_config:

```
AllowUsers vivek jerry
```

Alternatively, you can allow all users to login via SSH but deny only a few users, with the following line in sshd_config:

```
DenyUsers root saroj anjali foo
```

You can also [configure Linux PAM](#) allows or deny login via the sshd server. You can allow [list of group name](#) to access or deny access to the ssh.

You need to explicitly disallow remote login from accounts with empty passwords, update sshd_config with the following line:

```
PermitEmptyPasswords no
```

It cannot be stressed enough how important it is to use strong user passwords and passphrase for your keys. Brute force attack works because user goes to dictionary based passwords. You can force users to avoid [passwords against a dictionary](#) attack and use [john the ripper tool](#) to find out existing weak passwords. Here is a sample random password generator (put in your ~/.bashrc):

Posted - Mon, Oct 1, 2018 11:30 PM. This article has been viewed 12011 times.

Online URL: <http://kb.ictbanking.net/article.php?id=399>