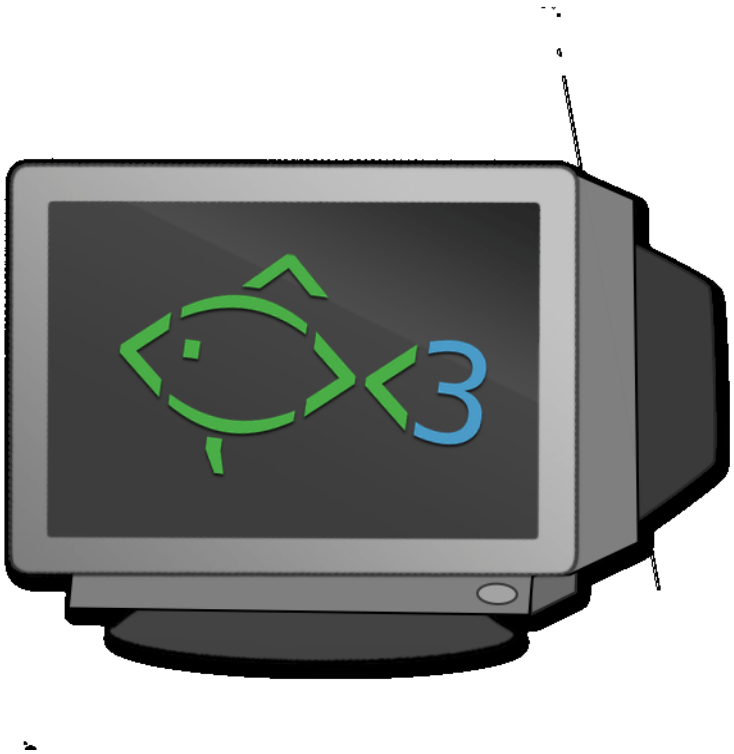
Update reverse sshd config with cronjob to revert if sshd reload issues

Author : admin



Say you're doing ssh hardening modifying /etc/ssh/sshd_config for better system security or just changing options in sshd due to some requirements. But you follow the wrong guide and you placed some ssh variable which is working normally on newer SSH versions ssh OpenSSH_8.0p1 / or 7 but the options are applied on older SSH server and due to that restarting sshd via /etc/init.d/... or systemctl restart sshd cuts your access to remote server located in a DC and not attached to Admin LAN port, and does not have a working ILO or IDRAC configured and you have to wait for a couple of hours for some Support to go to the server Room / Rack / line location to have access to a Linux physical tty console and fix it by reverting the last changes you made to sshd and restarting.

Thus logical question comes what can you do to assure yourself you would not cut your network access to remote machine after modifying OpenSSHD and normal SSHD restart?

There is an old trick, I'm using for years now but perhaps if you're just starting with Linux as a novice system administrator or a server support guy you would not know it, it is as simple as setting a cron job for some minutes to periodically overwrite the sshd configuration with a copy of the old working version

of sshd before modification.

Here is this nice nify trick which saved me headache of call on technical support line to ValueWeb when I was administering some old Linux servers back in the 2000s

```
root@server:~# crontab -u root -e
```

```
# create /etc/ssh/sshd_config backup file
cp -rpf /etc/ssh/sshd_config /etc/ssh/sshd_config_$(date +%d-%m-%y)
# add to cronjob to execute every 15 minutes and ovewrite sshd with the working version just in case
*/15 * * * * /bin/cp -rpf /etc/ssh/sshd_config_$(date +%d-%m-%y) /etc/ssh/sshd_config && /bin/systemctl restart sshd
# restart sshd
cp -rpf /etc/ssh/sshd_config_$(date +%d-%m-%y) /etc/ssh/sshd_config && /bin/systemctl restart sshd
```

Copy paste above cron definitions and leave them on for some time. Do the /etc/ssh/sshd_config modifications and once you're done restart sshd by lets say

```
root@server:~#  killall -HUP sshd
```

If the ssh connectivity continues to work edit the cron job again and delete all lines and save again.
If you're not feeling confortable with vim as a text editor (in case you're a complete newbie and you don't know) how to get out of vim. Before doing all little steps you can do on the shell with **export EDITOR=nano** or **export EDITOR=mcedit** cmds,this will change the default text editor on the shell.

Hope this helps someone... Enjoy :)