# Table of Contents

# OpenSSH SuSE 10

We have a little base of old SuSEs 10, one problem is the public services that are too old, in our case we received news that openssh and openssl are too old.

I need to compile a newer openssh and openssl version, because we can't pay extended support from SuSE.

Another issue is maintain SSH alive to avoid remote access problmes. We made a little trick to get working.

## Compiling OpenSSL

You need to have installed compiling tools, get SuSE documentation for get the trick, we will assume that you already installed.

You need OpenSSL 1.0, because 1.1 need a newer version of Perl. We choose the newer that OpenSSL offer in the page, that is actually maintained.

```
$ wget https://www.openssl.org/source/openssl-1.0.2k.tar.gz
$ gunzip openssl-1.0.2k.tar.gz && tar xvf openssl-1.0.2k.tar
$ cd openssl-1.0.2k
$ ./config
$ make
$ sudo make install
```

Maybe you need to remove openssl-devel package to avoid the use of old library headers.

## Compiling OpenSSH

You need download the Portable version of openssh, the latest version works great at the moment of write this page.

```
$ wget
http://mirrors.evowise.com/pub/OpenBSD/OpenSSH/portable/openssh-7.4p1.tar.gz
$ gunzip openssh-7.4p1.tar.gz && tar xvf openssh-7.4p1.tar
$ cd openssh-7.4p1
$ ./configure
$ make
$ sudo make install
```

If configure detects an old library remove the openssl-devel package or use ./configure —with-ssl-dir=/usr/local/ssl

## Change the daemon

Create a /etc/init.d/opensshd file with this content:

```
#! /bin/sh
# Copyright (c) 1995-2000 SuSE GmbH Nuernberg, Germany.
#
# Author: Jiri Smid <feedback@suse.de>
# Modified: Esteban Monge <estebanmonge@riseup.net>
# /etc/init.d/opensshd
#
#
### BEGIN INIT INFO
# Provides: opensshd
# Required-Start: $network $remote_fs
# Required-Stop: $network $remote_fs
# Default-Start: 3 5
# Default-Stop: 0 1 2 6
# Description: Start the sshd daemon
### END INIT INFO

SSHD_BIN=/usr/local/sbin/sshd
test -x $SSHD_BIN || exit 5

SSHD_SYSCONFIG=/etc/sysconfig/ssh
test -r $SSHD_SYSCONFIG || exit 6
. $SSHD_SYSCONFIG

SSHD_PIDFILE=/var/run/opensshd.init.pid

. /etc/rc.status

# Shell functions sourced from /etc/rc.status:
#      rc_check         check and set local and overall rc status
#      rc_status        check and set local and overall rc status
#      rc_status -v     ditto but be verbose in local rc status
#      rc_status -v -r  ditto and clear the local rc status
#      rc_failed        set local and overall rc status to failed
#      rc_reset         clear local rc status (overall remains)
#      rc_exit          exit appropriate to overall rc status

# First reset status of this service
rc_reset

case "$1" in
    start)
        if ! grep -q '^[[:space:]]*HostKey[[:space:]]'
/usr/local/etc/sshd_config; then
                if ! test -f /etc/ssh/ssh_host_key ; then
                    echo Generating /etc/ssh/ssh_host_key.
                    ssh-keygen -t rsa1 -b 1024 -f /etc/ssh/ssh_host_key -N
''
                fi
```

```
                if ! test -f /etc/ssh/ssh_host_dsa_key ; then
                    echo Generating /etc/ssh/ssh_host_dsa_key.
                    ssh-keygen -t dsa -b 1024 -f /etc/ssh/ssh_host_dsa_key -
N ''
                fi
                if ! test -f /etc/ssh/ssh_host_rsa_key ; then
                    echo Generating /etc/ssh/ssh_host_rsa_key.
                    ssh-keygen -t rsa -b 1024 -f /etc/ssh/ssh_host_rsa_key -
N ''
                fi
        fi
        echo -n "Starting SSH daemon"
        ## Start daemon with startproc(8). If this fails
        ## the echo return value is set appropriate.

        startproc -f -p $SSHD_PIDFILE $SSHD_BIN $SSHD_OPTS -o
"PidFile=$SSHD_PIDFILE"

        # Remember status and be verbose
        rc_status -v
        ;;
    stop)
        echo -n "Shutting down SSH daemon"
        ## Stop daemon with killproc(8) and if this fails
        ## set echo the echo return value.

        killproc -p $SSHD_PIDFILE -TERM $SSHD_BIN

        # Remember status and be verbose
        rc_status -v
        ;;
    try-restart)
        ## Stop the service and if this succeeds (i.e. the
        ## service was running before), start it again.
        $0 status >/dev/null &&  $0 restart

        # Remember status and be quiet
        rc_status
        ;;
    restart)
        ## Stop the service and regardless of whether it was
        ## running or not, start it again.
        $0 stop
        $0 start

        # Remember status and be quiet
        rc_status
        ;;
    force-reload|reload)
        ## Signal the daemon to reload its config. Most daemons
        ## do this on signal 1 (SIGHUP).
```

```
        echo -n "Reload service sshd"

        killproc -p $SSHD_PIDFILE -HUP $SSHD_BIN

        rc_status -v

        ;;
    status)
        echo -n "Checking for service sshd "
        ## Check status with checkproc(8), if process is running
        ## checkproc will return with exit status 0.

        # Status has a slightly different for the status command:
        # 0 - service running
        # 1 - service dead, but /var/run/  pid  file exists
        # 2 - service dead, but /var/lock/ lock file exists
        # 3 - service not running

        checkproc -p $SSHD_PIDFILE $SSHD_BIN

        rc_status -v
        ;;
    probe)
        ## Optional: Probe for the necessity of a reload,
        ## give out the argument which is required for a reload.

        test /usr/local/etc/sshd_config -nt $SSHD_PIDFILE && echo reload
        ;;
    *)
        echo "Usage: $0 {start|stop|status|try-restart|restart|force-
reload|reload|probe}"
        exit 1
        ;;
esac
rc_exit
```

Edit /usr/local/etc/sshd_config and change the port, if you have the firewall up you will need open the port:

```
#Port 22
```

To

```
Port 10001
```

Start the new ssh daemon:

```
service opensshd start
chkconfig opensshd on
```

Logout from all SSH sessions and enter with the new ssh daemon:

```
ssh -p 10001 username@ipofserver
```

Stop old ssh daemon:

```
service sshd stop
chkconfig sshd off
```

Edit /usr/local/etc/sshd_config and revert the change:

```
Port 10001
```

To:

```
#Port 22
```

Finally restart again the service:

```
service opensshd restart
```

Now you can enter to the server in the normal way, maybe the ssh keys must be regenerated.

#### Special note for s390x

The real challenge was that our SuSEs are zLinux or s390x architecture or zEC12 server from IBM, I received this message:

```
configure: error: *** Can't find recent OpenSSL libcrypto (see config.log
for details) ***
```

When try to compile openssh, you can make the fix with:

```
$ ./configure --build=s390x
```

## References

- https://lists.mindrot.org/pipermail/openssh-bugs/2008-April/006660.html