Table of Contents

OpenSSH RHEL 5

We have a little base of old RHEL 5, 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 RHEL is EOL.

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

Other reported issues is trying to enter from RHEL 5 to servers with weak ciphers disabled. This is caused because the openssl is too old to offer secure ciphers.

Compiling OpenSSL

You need to have installed compiling tools:

```
yum install gcc zlib-devel pam-devel
```

You will need OpenSSL 1.0, because 1.1 need a higher version of Perl 5.10, RHEL 5 comes with Perl 5.8. We choose the newer version that OpenSSL offer in the page. Remember this is your last chance to maintain alive your server... please consider upgrade your server to a newer version of RHEL.

```
$ wget https://www.openssl.org/source/openssl-1.0.2u.tar.gz
$ gunzip openssl-1.0.2u.tar.gz && tar xvf openssl-1.0.2u.tar
$ cd openssl-1.0.2u
$ CFLAGS=-fPIC ./config shared
$ 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-8.9p1.tar.gz
$ gunzip openssh-8.9p1.tar.gz && tar xvf openssh-8.9p1.tar
$ cd openssh-8.9p1
$ ./configure --with-ssl-dir=/usr/local/ssl  --
includedir=/usr/local/ssl/include/ --with-pam
$ make
$ sudo make install
```

Change the daemon

I not tested this part... I take this script from my old guide for the same with Suse 10 . Maybe you will need some tweaks.

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

```
#!/bin/bash
#
# Init file for OpenSSH server daemon
# Modified by Esteban Monge estebanmonge@riseup.net
# chkconfig: 2345 55 25
# description: OpenSSH server daemon
#
# processname: sshd
# config: /usr/local/etc/ssh_host_key
# config: /usr/local/etc/ssh_host_key.pub
# config: /usr/local/etc/ssh_random_seed
# config: /usr/local/etc/sshd_config
# pidfile: /var/run/sshd.pid

# source function library
. /etc/rc.d/init.d/functions

# pull in sysconfig settings
[ -f /etc/sysconfig/sshd ] && . /etc/sysconfig/sshd

RETVAL=0
prog="sshd"

# Some functions to make the below more readable
KEYGEN=/usr/local/bin/ssh-keygen
SSHD=/usr/local/sbin/sshd
RSA1_KEY=/usr/local/etc/ssh_host_key
RSA_KEY=/usr/local/etc/ssh_host_rsa_key
DSA_KEY=/usr/local/etc/ssh_host_dsa_key
PID_FILE=/var/run/sshd.pid

runlevel=$(set -- $(runlevel); eval "echo \$$#" )

do_rsa1_keygen() {
    if [ ! -s $RSA1_KEY ]; then
        echo -n $"Generating SSH1 RSA host key: "
        rm -f $RSA1_KEY
        if $KEYGEN -q -t rsa1 -f $RSA1_KEY -C '' -N '' >&/dev/null; then
            chmod 600 $RSA1_KEY
            chmod 644 $RSA1_KEY.pub
            if [ -x /sbin/restorecon ]; then
                /sbin/restorecon $RSA1_KEY.pub
            fi
```

```
            success $"RSA1 key generation"
            echo
        else
            failure $"RSA1 key generation"
            echo
            exit 1
        fi
    fi
}

do_rsa_keygen() {
    if [ ! -s $RSA_KEY ]; then
        echo -n $"Generating SSH2 RSA host key: "
        rm -f $RSA_KEY
        if $KEYGEN -q -t rsa -f $RSA_KEY -C '' -N '' >&/dev/null; then
            chmod 600 $RSA_KEY
            chmod 644 $RSA_KEY.pub
            if [ -x /sbin/restorecon ]; then
                /sbin/restorecon $RSA_KEY.pub
            fi
            success $"RSA key generation"
            echo
        else
            failure $"RSA key generation"
            echo
            exit 1
        fi
    fi
}

do_dsa_keygen() {
    if [ ! -s $DSA_KEY ]; then
        echo -n $"Generating SSH2 DSA host key: "
        rm -f $DSA_KEY
        if $KEYGEN -q -t dsa -f $DSA_KEY -C '' -N '' >&/dev/null; then
            chmod 600 $DSA_KEY
            chmod 644 $DSA_KEY.pub
            if [ -x /sbin/restorecon ]; then
                /sbin/restorecon $DSA_KEY.pub
            fi
            success $"DSA key generation"
            echo
        else
            failure $"DSA key generation"
            echo
            exit 1
        fi
    fi
}

do_restart_sanity_check()
```

```
{
    $SSHD -t
    RETVAL=$?
    if [ ! "$RETVAL" = 0 ]; then
        failure $"Configuration file or keys are invalid"
        echo
    fi
}

start()
{
    # Create keys if necessary
    do_rsa1_keygen
    do_rsa_keygen
    do_dsa_keygen
    cp -af /etc/localtime /var/empty/sshd/etc

    echo -n $"Starting $prog: "
    $SSHD $OPTIONS && success || failure
    RETVAL=$?
    [ "$RETVAL" = 0 ] && touch /var/lock/subsys/sshd
    echo
}

stop()
{
    echo -n $"Stopping $prog: "
    if [ -n "`pidfileofproc $SSHD`" ] ; then
        killproc $SSHD
    else
        failure $"Stopping $prog"
    fi
    RETVAL=$?
    # if we are in halt or reboot runlevel kill all running sessions
    # so the TCP connections are closed cleanly
    if [ "x$runlevel" = x0 -o "x$runlevel" = x6 ] ; then
        killall $prog 2>/dev/null
    fi
    [ "$RETVAL" = 0 ] && rm -f /var/lock/subsys/sshd
    echo
}

reload()
{
    echo -n $"Reloading $prog: "
    if [ -n "`pidfileofproc $SSHD`" ] ; then
        killproc $SSHD -HUP
    else
        failure $"Reloading $prog"
    fi
    RETVAL=$?
```

```
    echo
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    reload)
        reload
        ;;
    condrestart)
        if [ -f /var/lock/subsys/sshd ] ; then
            do_restart_sanity_check
            if [ "$RETVAL" = 0 ] ; then
                stop
                # avoid race
                sleep 3
                start
            fi
        fi
        ;;
    status)
        status -p $PID_FILE openssh-daemon
        RETVAL=$?
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|reload|condrestart|status}"
        RETVAL=1
esac
exit $RETVAL
```

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.

References

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