RSSH as RPM

There is a nice tool **rssh** (http://www.pizzashack.org/rssh/index.shtml). It can be used as a shell for users, and those users may copy files via `scp`, `sftp`, `rsync`, `cvs` and/or `rdist`, but does not allow to

log in. Who is allowed to to what is configured in `/etc/rssh.conf`. Thats a fine thing! 🙂

The software was last updated and is perfect now, or so the author states. However, the installation routines are not.

The first steps are simple:

- on http://www.pizzashack.org/rssh/download.shtml, follow the link to download the .src.rpm and save the file
- as root you install the saved file (`rpm -ivh /patch/to/rssh-2.3.2-1.src.rpm`). After doing so, you have the files `/usr/src/packages/SPECS/rssh.spec` and `/usr/src/packages/SOURCES/rssh-2.3.2.tar.gz` (at least on an openSUSE installation; on other distros, `/usr/src/packages` may be called `/usr/src/redhat` or so...).

Flaw 1: sftp-server

The first weak point is the search for the `sftp-server` binary. Because it is searched in `/usr/lib` only, it is not found on 64 bit system that use `/usr/lib64`. I asked myself why `configure` does not search where this binary is configured, and noticed that my first idea worked for `root` only. So I extended the first version and then got the following patch:

```
--- ./configure 2008-11-23 17:17:59.000000000 +0100
+++ ../rssh-orig/configure      2006-01-07 03:24:58.000000000 +0100
@@ -4984,10 +4984,6 @@
 fi
 scp_path=$ac_cv_path_scp_path

-if test -z "$scp_path"; then
-   scp_path=`which scp`
-fi
-
 if test -n "$scp_path"; then
   echo "$as_me:$LINENO: result: $scp_path" >&5
 echo "${ECHO_T}$scp_path" >&6
@@ -5032,10 +5028,6 @@
   *)
   as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
as_dummy="/usr/libexec:/usr/libexec/openssh:/usr/local/libexec/openssh:/usr/
lib/openssh:/usr/lib:/usr/local/libexec:/usr/lib/ssh"
-as_arch_helper=`uname -m`
-if test "a$as_arch_helper" = "ax86_64"; then
-   as_dummy="$as_dummy:/usr/lib64/openssh:/usr/lib64:/usr/lib64/ssh"
-fi
```

```
  for as_dir in $as_dummy
  do
    IFS=$as_save_IFS
@@ -5054,12 +5046,6 @@
  fi
  sftp_path=$ac_cv_path_sftp_path

-if test -z "$sftp_path"; then
-    if test -r /etc/ssh/sshd_config; then
-        sftp_path=`grep sftp-server /etc/ssh/sshd_config | awk '{print
$3}'`
-    fi
-fi
-
 if test -n "$sftp_path"; then
    echo "$as_me:$LINENO: result: $sftp_path" >&5
 echo "${ECHO_T}$sftp_path" >&6
```

This includes finding `scp` – the openSUSE Build System told me it could not find the `scp` binary, so I added three lines for that...

This code may be saved as `configure.patch`. You have to unpack the source archives now (for example: `cd /usr/src/packages/SOURCES && tar -xzf rssh-2.3.2.tar.gz && cd rssh-2.3.2`). This will bring you in the directory of the extracted sources.

You may want to enter `cp configure configure.orig` to save the original configure script. Afterwards, you can apply the patch with `patch -p0 < /path/to/configure.patch` – this

should do the trick. Of course you can add those code lines after line 5048 manually, too 🙂 .

After applying the patch, you have to rebuild the source RPM: `cd .. && mv rssh-2.3.2.tar.gz rssh-2.3.2.tar.gz.orig && tar -cvzf rssh-2.3.2.tar.gz rssh-2.3.2` will create a safety copy of the sources and pack the modified sources in a .tar.gz file to be used for `rpmbuild`.

Flaw 2: rpmbuild and rssh_chroot_helper

So far, so good. But. When you try `cd /usr/src/packages && rpmbuild -ba SPECS/rssh.spec`, it still won't run through because of a `chmod` that tries to modify the already (but not yet) installed `rssh_chroot_helper` instead of the one that was just compiled.

As a workaround, sourceforge bug #1384981 tells to modify `Makefile.am`. Since that does not help, I applied this manually to `Makefile.in` (line 731). You can do the same with line 19 of

`Makefile.am`, it won't hurt, the line has the same content 🤫

So we have to re-proceed the source code unpacking an repacking from the first flaw: `cd /usr/src/packages/SOURCES && tar -xzf rssh-2.3.2.tar.gz && cd rssh-2.3.2`, use an editor for Makefile.in and look at line 731, it reads `chmod u+s $(libexecdir)/rssh_chroot_helper` and you change it to `chmod u+s $(DESTDIR)$(libexecdir)/rssh_chroot_helper`. Save the file now, quit the editor and

repackage the sources (`cd .. && rm -f rssh-2.3.2.tar.gz && tar -cvzf rssh-2.3.2.tar.gz rssh-2.3.2`).

Summary

After going through all this, we can summarize the proceedings:

- on http://www.pizzashack.org/rssh/download.shtml, follow the link to download the .src.rpm and save the file
- as root you install the saved file (`rpm -ivh /patch/to/rssh-2.3.2-1.src.rpm`). After doing so, you have the files `/usr/src/packages/SPECS/rssh.spec` and `/usr/src/packages/SOURCES/rssh-2.3.2.tar.gz`
- unpack the source archive and change to the extraction directory (`cd /usr/src/packages/SOURCES && tar -xzf rssh-2.3.2.tar.gz && cd rssh-2.3.2`)
- add the 14 lines in the `configure` script as described above
- modify line 731 in `Makefile.in` (and line 19 of `Makefile.am`) as described above
- save the original sources and create a new sources archive (`cd .. && mv rssh-2.3.2.tar.gz rssh-2.3.2.tar.gz.orig && tar -cvzf rssh-2.3.2.tar.gz rssh-2.3.2`)
- maybe you want to leave a footprint in the `spec` file by changing `Release: 1` to `Release: 1m` (m for modified) in `SPECS/rssh.spec`, but of course this is not necessary 🤭
- now start building the RPM: `rpmbuild -ba SPECS/rssh.spec`

When everything was successful, you can install the freshly created rpm with `rpm -ivh RPMS/$(uname -m)/rssh*.rpm` now 🙂

From:
http://wernerflamme.name/ - **Werners Wiki**

Permanent link:
http://wernerflamme.name/doku.php?id=comp:en:rsshrpm

Last update: **2008-12-02 06:09**