RPM in Informatics (abridged)

by Alastair Scobie <ajs@dcs.ed.ac.uk>

Division of Informatics
University of Edinburgh

1 RPMs in Informatics

This abridged tech-note describes the use of the RPM (RedHat Package Manager) in the Division of Informatics [1]; in particular, how a user submits an RPM package for installation and restrictions we place on such user-submitted RPM packages.

The full version of this tech-note includes a short tutorial and some hints and tips on building RPM packages.

1.1 How RPMs are distributed in Informatics

The RPMs that make up the Division's Linux service live, currently, in `/usr/local/linux/rpms`. Every night, each machine runs an application called `updaterpms` which installs, upgrades or deletes RPM packages according to a configuration file (living in `/usr/local/linux/rpmcfg`). `updaterpms` is also called when a machine boots (laptops excepted).

The files in `/usr/local/linux` are themselves replicated copies of files in the master directory `/export/local/linux`; they are served by several file servers to the Linux machines.

The source SRPMs live in `/useful/srpms`.

1.2 Restrictions in managed environment

An RPM can consist of files destined to be installed anywhere in a machine's *file space*. This has to be possible because RPMs are used to install all the permanent files in a system (even including the kernel).

This obviously is a possible security risk; a rogue RPM could install a replacement hacked version of a popular binary in a directory which is earlier in the shell path (`$PATH`) than the official version. Eg. `ls` is officially in `/usr/bin`, but `$PATH` is typically `/usr/etc:/usr/bin:/bin:/sbin` so an RPM could put a hacked version in `/usr/etc`. Not all potential hacks are as simple as this; a rogue RPM could consist of perfectly innocent files, but modify a configuration file (eg `/etc/passwd`) by editing the file from a pre-install or post-install script.

We don't check that all the RedHat RPMs are well behaved. Instead, we check that their RPMs have valid PGP signatures proving that the RPMs really were generated by RedHat; we trust RedHat not to damage their reputation by shipping misbehaving RPMs. RPMs from other sources (including from within the Division) are checked by system staff prior to installation. User-submitted RPMs are not easy to implement with our current technology, but will be allowed in the new Division system.

Of course, we trust divisional staff (and postgraduates) not to create malignant code, but it is very easy to innocently make simple mistakes that have serious repercussions.

A significant problem with the file-system based package distribution used on our Solaris systems is that applications tended to rot and support levels became very unclear over time. One was never very clear without some investigation, which files (particularly in `/usr/local/{share,lib}` were used by the current version of a package.

Linux, where packages are contained in RPMs, has allowed us to adopt a policy of rebuilding RPMs from their SRPMs at every operating system upgrade; usually annually. This allows us to review whether an application is still required, ensure that it still works and because a log is kept of who has submitted each RPM, ensure that each RPM has a valid "owner". To make management of packages easier in the long term, SRPMS must be submitted along with the resultant RPM.

Eventually, when we have transitioned from machine trust to user trust for services (eg network file-systems) users will be able to manage their own machine to a greater extent and will be able to install files anywhere they like on their own machines.

Once we have a good PGP infrastructure in place, only PGP signed RPMs will be accepted for installation. This involves very little extra work for the RPM builder (one extra flag to be specified) but gives extra confidence that an RPM was submitted by the person who claimed to have submitted it.

[1] currently just at KB site

1.3 Submitting an RPM

An RPM which is intended for submission for installation must meet the following conditions :-

- there should be no `setuid` or `setgid` files
- there should be no pre or post-install scripts
- the associated SRPM should also be submitted (unless it is not available).

Upgrades to existing packages (as shipped by RedHat or by systems staff) will not be accepted for distribution to machines other than those allocated to the submitting user.

Mail the location of the RPM and SRPM files and which machines you wish the RPM to be installed on to `support@dcs.ed.ac.uk`. If your RPM meets all the above conditions it should normally be shipped the next day, if your request is made by lunchtime. If it does not meet all the conditions, more work will be required to validate it and it may take several days to ship.

It is realized that this is not ideal, and user-submitted RPMs will be a feature of the new Divisional system.

Systems staff use the `rpmcheck` command to validate submitted RPMs. Users can use this program themselves to discover whether their RPM is likely to meet the above conditions.