AndroidSystemKeyMigration

Background information

The releases are currently signed by the individual developers with their personal gpg keys. During the installation procedure, the people installing the images are very strongly advised to check that kind of signatures. This makes sure that the images that are being installed were really made by the developers that signed them, and that they weren't modified since them. This takes care of the security while installing Replicant releases.

When installing Android applications, there is also a similar system in place, where people or organizations building applications sign their applications. When upgrading an application to a newer version, the signature is checked, and if it matches, the new application version can replace the old version and access the data of the previous application version.

Because of that, when building a Replicant release, we have to generate keys to sign the applications that we build and bundle in the Replicant images. This includes applications like the SMS application, the dialer, the launcher/desktop, etc.

What issue are we trying to solve here?

The releases of Replicant 6.0 0001, 0002 and 0003 were all signed by identical keys that were generated by Wolfgang Wiedmeyer.

We don't have access to these keys, so we needed to generate new keys during the build of the images.

However if the keys are not migrated somehow, after the installation, the first boot will never complete, and the second boot will also end up with the launcher/desktop crashing all the time, which not only blocks the usage of the device but also makes it hard to properly shut it down.

How we solved it

We wrote a tool to create a migration script that can be generated from various data (certificates from previous releases, a running image, etc) so that users running custom builds could also migrate back and forth between different images.

We tried various ways to run that script automatically during the first boot, to make it more easy for less technical users to do the key migration, but doing that in a robust way started to be complex as we either had to make the script more complex (and less robust) or lower the security of Android to enable the startup script to delete itself.

So instead we ended up making an extra -transition release to fix that:

- Users would install the transition release first, which would run the key migration during the first boot but also all subsequent boots.
- Once this is done, users would install the regular release to make sure that the key migration script stops running at each boot.

Doing the later is really necessary as during prolonged use and testing, we found that running the migration script at every boot was unsafe: if the boot is interrupted during the migration, the file that has has the information about the keys (/data/system/applications.xml) can be corrupted. That leads to non booting devices (users would need to wipe their data to fix that issue).

How to install new Releases

If you are using replicant-6.0-0003, and want to test the Replicant 6.0 0004 release, the easiest way to do it would be to first install a replicant-6.0-0004-rc5-transition image and then a replicant-6.0-0004-rc5 image.

If you are using any Replicant 6 0004 Release candidate (RC) images you also need to do the same thing as all the images before the replicant-6.0-0004-rc5 will either not do any miration or migrate the keys on every boot.

Details on the key sets

For Replicant 6, we have now 3 key sets that have been used for official releases (including RC images):

- The first key set has been generated by Wolfgang Wiedmeyer and it was used to sign the Replicant 6.0 0001, 0002, and 0003 releases.
- The second key set has been generated by Denis 'GNUtoo' Carikli and has been used to sign the Replicant 6.0 0004 RC1

image
- The third key set has also been generated by Denis 'GNUtoo' Carikli and has been used to sign the Replicant 6.0 0004 images released after Replicant 6.0 0004 RC1