CIP Core regular meeting

Date: May 25th, 2021 (30min~1h)

Time:

- timezones
- Tokyo (Japan) 17:30
- Taipei (Taiwan) 16:30
- Bangalore (India - Karnataka) 14:00
- Frankfurt (Germany - Hesse) 10:30
- London (United Kingdom - England) 08:30

Zoom

Dial-in numbers

Meeting ID: 917 9128 4612

Passcode: 248841

Past meetings

Rules

- http://www.linuxfoundation.org/antitrust-policy
- Please mark with (PRIVATE) those parts that should not appear in the public version of these minutes.

Roll Call

Participants (italic means did not attend)

- Daniel Sangorrin [TOSHIBA]
- Kazuhiro Hayashi [TOSHIBA]
- Dinesh Kumar [TOSHIBA]
- Venkata [TOSHIBA]
- Shivanand Kunijadar[TOSHIBA]
- Masato Minda [Plat'Home]
- *Chris Paterson [Renesas]*
- *Hung Tran [Renesas]*
- *Minh Tran [Renesas]*
- *Nhat Thieu [Renesas]*

- *Kento Yoshida [Renesas]*
- Kazuhiro Fujita [Renesas]
- Christian Storm [Siemens]
- Hiraku Toyooka [Cybertrust]
- *Sam Wilson [Codethink]*
- Jan Kiszka [Siemens]
- *Jonathan Sambrook [Codethink]*
- *Masashi Kudo [Cybertrust]*
- *SZ Lin [Moxa]*

Discussion

Action items updates (2021/05/11~05/25)

- AI(Daniel): finish cip-core-sec (80%)
  - update the ISAR gitlab-ci integration branch
  - New commit by Iwamatsu-san that adds apt update
    - https://gitlab.com/cip-project/cip-core/isar-cip-core/-/commit/bc28d055ffb190401610a87f831aaa416288bafe
  - New commit by Sarath P T Sarath_PT@mentor.com
    - https://gitlab.com/cip-playground/cip-core-sec/-/merge_requests/4
- AI(Daniel): ask Helmut for an estimation > choose > ask for budget
  - sent an e-mail to him and he kindly answered (see below)
- No activity
  - AI(Daniel): send BBB config for 5.10 and a v2 patch for BBB that includes 4.4
    - https://lore.kernel.org/cip-dev/f8f989ef-7ee2-42ee-a922-f53c6d2b07fe@siemens.com/T/#m4b75aee175603bd1f06a8eec2b47ac08bda85c5a
  - AI(Daniel): prepare Debian repository for CIP Core (70%)

Collaboration with Helmut Grohne

- ISAR: chroot-less and qemu-less installation (using DPKG_ROOT)
  - No need of qemu for cross installation nor root permissions
  - Goal
    - improve the speed of package cross-install, running the scripts, etc
    - improve safety on common build infrastructure by not using root permissions
    - focus on the CIP core packages for now

There are 3 distinct aspects:

a) Support native chrootless installation of the relevant packages (same architecture)

b) Support foreign chrootless installation of the relevant packages (i.e. no need for qemu)
c) Integrate this support for this into ISAR

- this would probably need to be done by us
- AI: let him know the list of CIP Core packages that need to be installed chrootless

Currently known issues

1. glibc problem with qemu-less cross-install
    a. [Note] For merely going chrootless, we likely need to pass -r to ldconfig.
    b. Problem: glibc's postinst runs ldconfig, however ldconfig on cross chroots
    c. Solution: add cross-ldconfig to glibc
        i. For sibling archs (amd64 <--> i386) it is supported
    d. Alternatives
        i. Skip running ldconfig and run it on firstboot ← not elegant
        ii. Emulate ldconfig using qemu ← we want to avoid this
2. bash needs significant work
    a. Problem: bash uses a C program as preinstall (no sh) ← bad for qemu-less
    b. Solution: get rid of that program
        i. We need to clean up a lot of legacy code and test a lot!!
3. base-files has a patch #824594 and likely more work
4. coreutils needs a patch
5. debconf has a patch #983425 ← probably hidden issues
6. debhelper needs a patch
7. debianutils needs work
8. shadow needs work

Approximate estimation (lower bound, there may be hidden ones):

- 1 man*month per issue (lower bound) → 8 man*month (experienced developer)

- Essential package removal
    - Remove perl and bash (and some glibc package) from the essential list
        - Start with bash (easier)
    - Goal
        - avoid GPLv3 packages (bash)
            - because secure boot might require locking them with a key
            - Jan: Collabora/Bosch's internal/public distro (Apertis) has no GPLv3. We should synchronize on this and see if they want to collaborate on this effort.
        - less CIP Core packages to maintain long-term
            - Make sure that the essential packages don't grow over time

Caution: bash is not the only GPLv3 package in essential.

- bsdutils
- coreutils
- findutils

- gcc
- gmp
- gnutls
- gnupg
- grep
- gzip
- libcap-ng
- libgpg-error
- libxcrypt
- sed
- util-linux

Removing BASH (easier than removing Perl) from essential

- remove the legacy about the preinstall C program
    - work shared with the qemu-less cross installation (1 month but shared)
- modify some bash scripts in essential to work with sh/dash (1 week)
    - /usr/bin/tzselect
    - /usr/bin/ldd
    - /usr/share/bug/apt/script

For packages outside essential, add dependencies on bash (no convert to sh). Possible methods to identify packages dependent on bash

- Search the contents of binary packages for “bash” etc (1 month)
    - trim false positives and fill a lot of bug reports
- Running autopkgtests in a modified environment that lacks bash (1 month)
    - Every failure needs to be inspected for its cause.
- Beyond runtime dependencies, Build-Depends are more relevant to this case Therefore, an archive rebuild will also be needed. (1 week and 10000 CPU hours)

Removing Perl (more difficult)

- A lot of packages already depend on perl. If any dependency depends on perl for an internal reason, we'll likely fail to identify a missing direct dependency. This seems acceptable to me. It requires a little preprocessing to ignore all packages whose installation set already forces the presence of perl. For the remainder, the work is similar to that of bash with two notable differences:
    - The work of removing perl use from other essential packages will be far bigger. It certainly won't be less than a month. debconf is written in perl.
    - Since debhelper depends on perl, perl is practically build-essential. The best route likely seems to add a dependency on perl-base to build-essential. Doing so makes the archive rebuild completely unnecessary. We can skip this step.
- After filing all the patches and waiting like three months for maintainers to apply them, we'll likely have to NMU the remaining packages. That usually incurs another week.

AI: prioritize the tasks in cip-members mailing list (low effort and high impact first), look for the approval from TSC (sell it as a collaboration with Debian, and as something that adds value to CIP members). Probably not before October.

CIP Core Testing

Latest information is updated for autopkgtest investigation for openssl, audit, fail2ban, bro package

- https://gitlab.com/cip-project/cip-core/isar-cip-core/-/issues/6
- AI: implement autopkgtests for fail2ban and bro
    - Tried adding autopkgtest for bro and running unit tests using autopkgtest, but there are several errors, as **buster version of bro package is not well maintained** so doing this in next version of bro will be explored
        - Daniel: in 3 months Bullseye will be release so then we could use that and forget about the buster version
        - Jan: unless there is some restriction for the certification (not technical) move forward
        - Daniel: we should also check if we can give feedback to unstable
    - Adding autopkgtest for fail2ban is in progress currently

IEC-62443-4-1 requirements

No update on this AI this week.

Development Environment Security SM-7

- This requirement asks to have a process in place to secure the development environment. This includes protecting various documents, code, binaries, updates etc.
    - To meet this requirement we need to ensure proper permissions for all CIP repositories and only identified people have merge privileges
    - How gitlab is protected as all our artifacts are kept in gitlab
    - What happens when a maintainer having merge privilege leaves CIP do we revoke his permissions?
    - AI(Dinesh): document with policies for managing membership and permissions in Gitlab and AWS

https://gitlab.com/cip-project/cip-documents/-/blob/master/security/development_environment_security.md

Control of Private keys SM-7

- This requirement expects CIP to document about various private keys used during development or code signing etc. How these keys are protected.
  Even if they are for reference or POC purposes.
- Currently used private keys are ?

1. Code Signing Keys
    a. Are we using it? If not can we do this to meet file integrity requirement?
    b. This applies to all CIP repo
2. Image Signing Keys
    a. Not used currently?
3. Secure boot signing keys
    a. It's generated but not stored anywhere
4. swupdate signing key
    a. It's generated but not stored anywhere
5. Other keys (TBD ??)
    a. Tokens for sending artifacts to AWS and jobs to LAVA

AI(Dinesh): create a document to explain how CIP is going to manage the keys

https://gitlab.com/cip-project/cip-documents/-/blob/master/security/private_key_management.md

Define human user accessible interfaces in CIP

- CR1.1 requirement, all human users on all interfaces capable of human user access should be authenticated on all interfaces such as HTTP/HTTPS, FTP/SFTP, SSH etc

- What all interfaces should we document? ssh, http/https?

Reproducible builds

Previous information

Daniel got in contact with Chris Lamb.

- We are still interested in reproducible builds
- We will work on making ISAR-CIP-CORE images reproducible
    - security benefits
    - software updates deltas minimization
        - both block-based and package-based updates
        - there was an experiment with RPM, that used deltas
- Collaboration possibilities
    - They will review our results
    - Write an article on their newsletter
    - Shared presentation in a conference about how reproducible images reduce the size of software updates deltas
    - Document on their site about "know-how" making a build tool reproducible
    - Chris can put us in contact with Roland Clobus
        - he was working on making the Debian Live images reproducible
        - https://lists.reproducible-builds.org/pipermail/rb-general/2020-September/002044.html
    - Chris mentioned that Tails also has final image reproducibility

        - https://tails.boum.org/contribute/build/reproducible/
    - Collaborate on the CI (Holger is the expert)
        - https://reproducible-builds.org/citests/
- We should use the appropriate channels for support
    - rb-general mailing-list
    - https://reproducible-builds.org/contribute/

Next action items:

- Confirm that ISAR-CIP-CORE images are not reproducible in the first place
- Try to figure out easy reasons and fix them
- For the complex ones, consider contacting with Roland or ask members to contribute

Updates to Isar-cip-core

- kas/opt: Restructure ebg-swu.yml and qemu-swupdate.yml
    - Quirin Gylstorff <quirin.gylstorff@siemens.com>
    - https://gitlab.com/cip-project/cip-core/isar-cip-core/-/commit/3d3ca60179e7628671b41b4bc93d2bb0f1e19a01
- Query on secure boot
    - How secure boot was confirmed when unified signed image booted successfully?
        - https://gitlab.com/cip-project/cip-core/isar-cip-core/-/issues/9
    - Shall we enhance README.secureboot to include the details of secureboot keys(PK,KEK,db,dbx) and how they are maintained in CIP currently?
        - AI(Dinesh): please send a change request/issue

Updates to Deby

None

CIP Core lifecycle

Approved and moved to https://gitlab.com/cip-project/cip-lifecycle

Software Updates WG

Next steps

- Move the repository to cip-core/swupdate-handler-roundrobin
    - https://gitlab.com/cip-playground/swupdate-handler-roundrobin
- Get it working on isar-cip-core amd64 QEMU
    - Estimated date: by the next meeting

- Get it working on ARM (u-boot)
- Peer-review on the Lua code and the usage (documentation bugs)
    - This can be done already

Q&A or comments

- None

Items that need approval by TSC voting members

None

Future topics

- SDK images
- Next tiny profile
- Reproducible builds