Qubes build system

requirements and considerations

Maciej Pijanowski

3MDEB

Build system security requirements

Main requirements from the point of QubesOS:

- verify digital signature of ALL downloaded components to make sure they come from trusted source
- sandboxing - malicious components (if any) cannot affect the dom0, Xen or the developer's machine

qubes-builder

- Fedora host required
- keychain of trusted GPG keys for fetched components
- each fetched component's signature is verified with appropriate GPG key
- several patches for fetched code verification (e.g. Xen)

OpenEmbedded (Yocto)

- Yocto uses OpenEmbedded as the core of the build system
- Yocto / OE is targeted at the embedded systems
- What embedded system means nowadays?
- Security is not on the first place. No security policy enforcement (it is left to the end user)

OpenEmbedded (Yocto)

- main use-case is to create one single image to flash on target
- can be used to create package feed (rpm, deb, ipk available)
- Flexible framework - different projects pick up OE as a base build system (OpenBMC, isar, OpenXT)
- Source integrity checks via `git` and `sha256` sums

OpenXT

- 64-bit Debian host required (build scripts install stuff from `apt`) https://openxt.atlassian.net/wiki/spaces/OD/pages/10747922/How+to+build+OpenX
- LXC Containers used during build process
- at a first glance, no additional source verification is implemented
- rather standard OE mechanism are used, i.e. `sha256sum` for integrity check
- most of the custom components (`recipes-openxt`) do not specify particular commit: `SRCREV = "${AUTOREV}"`

Summary

- OE is not designed with the goals as the qubes-builder have
- However, OE may provide some common ground as a base
- Thanks to the flexibility, it might be possible to implement additional security checks we need
- OE community might gladly welcome optional patches which improve build security