#

CIP Development Environment Security

## Table of contents


1. Objective
2. Assumptions
3. Scope
4. Security Requirement
5. CIP Development Environment
   - 5.1 Protection During Development
   - 5.2 Protection During Production
   - 5.3 Protection During Delivery
6. Policy for CIP repository maintainer privilege
7. Current CIP repositories and maintainers
8. Cloud Account Policy
9. Reproducible CIP builds for integrity verification


---

### Revision History

| Revision No | Date | Change description | Author | Reviewed by |
|---|---|---|---|---|
| 001 | 2021-06-14 | Draft development environment security | Dinesh Kumar | To be reviewed by CIP Security WG members |
| 002 | 2021-06-19 | Updated based on Daniel-san's comment, email dated 15/06/2021 | Dinesh Kumar | To be reviewed by CIP Security WG members |
| 003 | 2021-08-30 | Updated based on Yasin's comment in gitlab | Dinesh Kumar | Yasin |
| 004 | 2022-04-14 | Reviewed and updated all gitlab owners and maintainers | Dinesh Kumar | TBR |
| 005 | 2022-06-20 | Updated owners table | Dinesh Kumar | TBR |

---

---

## 1. Objective

The primary objective of this document is to document current development environment security, development flow and how security is maintained.

Moreover, subsequent revisions of this document may consider additional details of existing development or changes and enhancement to improve development environment security.

## 2. Assumptions

| Assumption | Impact |
|---|---|
| The development environment of upstream developers(Debian and Mainline kernel) is protected | Any security loopholes embedded in upstream project will impact CIP directly |

## 3. Scope

Scope of this document is to consider current development model for CIP. Current CIP development model follows open source development method where everyone is allowed to contribute and only few members have privilege to merge the changes in CIP repositories.

## 4. Security Requirement

IEC-62443-4-1 has following development environment security requirements

- CIP shall define process which has procedural as well as technical control for protecting a product during development, production and delivery. This includes following
    1. Update patches
    2. Design
    3. Implementation
    4. Testing
    5. Releases
- Having this process in place means CIP provides ways to protect integrity of following
    1. Code
    2. Documents (User manuals, Design, )
    3. Configuration setting
    4. Private keys
    5. Authenticators (password, access control list, code signing certificates)

## 5. CIP Development Environment

### 5.1 Protection During Development

As CIP re-uses open source components hence it is assumed upstream components are protected by respective component owner. However, artifacts such as various documents are kept in gitlab and they are protected by gitlab authentication mechanism. Similarly meta-data of recipes(.bb, .bbclass files etc) is protected by gitlab authentication mechanism.

During development any CIP developers can send merge requests with their changes. Here one thing to note is CIP developers can send merge requests, whereas CIP contributors can only send patches to the mailing list. All changes/patches are reviewed by CIP peer developers and feedback is provided. Once all the review comments are fixed, CIP maintainer of the respective repository merges the changes.

### 5.2 Protection During Production

### 5.3 Protection During Delivery

This requirement should be met by CIP users.(**TO_BE_MET_BY_CIP_USERS**)

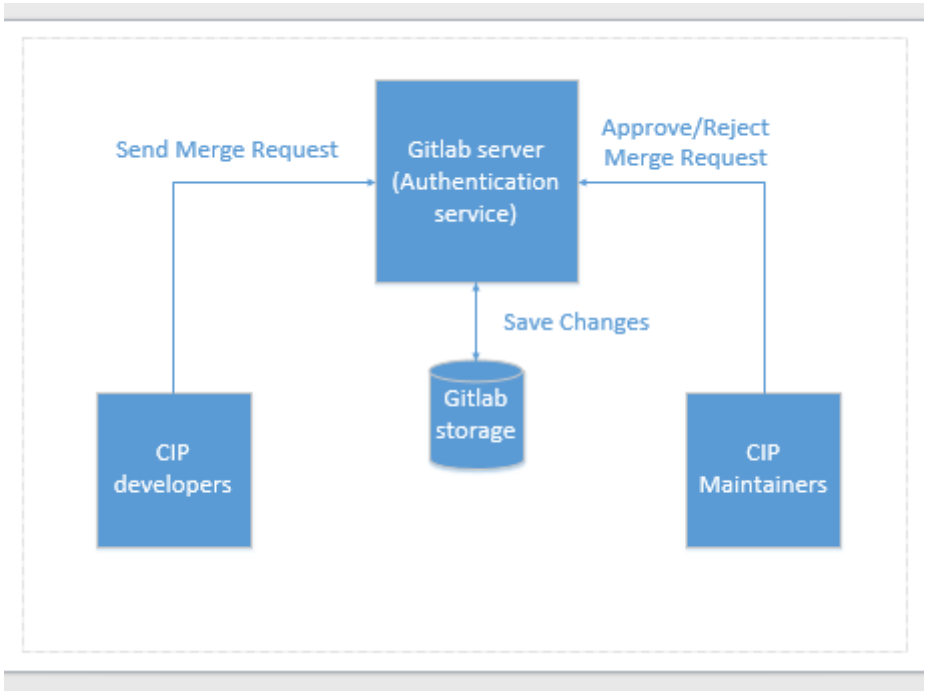


Figure 1: CIP development flow

Currently there is no production by CIP. When CIP adds production of images there will be information added here. Currently CIP does only deliver source code via Gitlab. Therefore, CIP relies on Gitlabs security

## 6. Policy for CIP repository maintainer privilege

CIP has following policy for reviewing maintainers privilege to control CIP repositories

- If a CIP maintainer leaves CIP development and has not contributed in 6 months, his merge privilege is revoked in all CIP repositories and downgraded to developer privilege.
- All repositories are reviewed annually. If a maintainer has not contributed in 6 months, his maintainer privileges are revoked.
- Any CIP member can be given maintainer rights for any repositories after TSC members approval.

CIP has following policy for reviewing maintainers privilege to control CIP AWS accounts

- If a CIP contributer on AWS leaves CIP development and has not contributed in 6 months, his AWS account access is revoked.
- All AWS accounts are reviewed annually. If a contributer has not contributed in 6 months, his access is revoked if he left his role.
- Any CIP member can be given AWS access rights after TSC members approval.
- No shared machine accounts are allowed to have remote login access enabled. (Feedback needed)

## 7. Current CIP repositories and maintainers

| CIP gitlab group | Repo and it's URL | Owners | Maintainers | Last review date |
|---|---|---|---|---|
| cip-project | https://gitlab.com/cip-project | Yoshitake Kobayashi, Chris Paterson, Takehisa Katayama, Hidehiro Kawai, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder, Kento Yoshida, Jan Kiszka | None | 20 Jun 2022 |
| cip-security | iec_62443-4-x https://gitlab.com/cip-project/cip-security/iec_62443-4-x | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Yasin Demirci, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi | Stefan shroeder | 14 Apr 2022 |
| cip-sw-update | cip-sw-updates-demo https://gitlab.com/cip-project/cip-sw-updates/cip-sw-updates-demoswupdate-handler-roundrobinhttps://gitlab.com/cip-project/cip-sw-updates/swupdate-handler-roundrobincip-sw-updates-taskshttps://gitlab.com/cip-project/cip-sw-updates/cip-sw-updates-tasks | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi, Akihiro Suzuki | swupdate-handler-roundrobin maintainer Christian Storm | 14 Apr 2022 |
| cip-core | isa-cip-core https://gitlab.com/cip-project/cip-core/isar-cip-core deby https://gitlab.com/cip-project/cip-core/deby cip-pkglist https://gitlab.com/cip-project/cip-core/cip-pkglist | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi, Daniel Sangorrin | Alice Ferrazzi | 14 Apr 2022 |

| CIP gitlab group | Repo and it's URL | Owners | Maintainers | Last review date |
|---|---|---|---|---|
| cip-kernel | cip-kernel-sec https://gitlab.com/cip-project/cip-kernel/cip-kernel-seclinux-cip https://gitlab.com/cip-project/cip-kernel/linux-cip cip-kernel-config https://gitlab.com/cip-project/cip-kernel/cip-kernel-config lts-commit-list https://gitlab.com/cip-project/cip-kernel/lts-commit-listcip-kernel-testshttps://gitlab.com/cip-project/cip-kernel/cip-kernel-testsclassify-failed-patcheshttps://gitlab.com/cip-project/cip-kernel/classify-failed-patches | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi | cip-kernel-sec: Nobohiro Iwamatsu, Pavel Machek, Masami Ishikawalinux-cip, cip-kernel-config, classify-failed-patches: Nobohiro Iwamatsu, Pavel Macheklts-commit-list: Nobohiro Iwamatsu, Pavel Machek, Ulrich Hechtcip-kernel-tests: Robert Marshal, Nobohiro Iwamatsu, Pavel Machek | 14 Apr 2022 |
| cip-testing | cip-testing https://gitlab.com/cip-project/cip-testing | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi | Alice Ferrazzi | 14 Apr 2022 |
| cip-lifecycle | cip-lifecycle https://gitlab.com/cip-project/cip-lifecycle | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi | None | 14 Apr 2022 |
| cip-documents | cip-documentshttps://gitlab.com/cip-project/cip-documents | Hidehiro Kawai,Kento Yoshida, Laurence Urhegyi, Kazuhiro Hayashi, Samuel holder,Jan Kiszka, Chris Paterson,Takehisa Katayama, Yoshitake Kobayashi | Dinesh Kumar | 14 Apr 2022 |

## 8. Cloud Account Policy

CIP primarly uses AWS accounts for keeping CIP kernel images and other artifacts. Following is the detail of AWS account owners

- TBD

## 9. Reproducible CIP builds for integrity verification

In future CIP plans to make CIP builds and image creation process reproducible. This will ensure integrity of all CIP meta-data for image creation as well as process integrity remains intact.

This section will be updated once CIP achieves reproducible builds.