#

CIP Secure Development Process

Table of contents

23. SVV-2 Threat Mitigation testing
24. SVV-3 Vulnerability testing
25. SVV-4 Penetration testing
26. SVV-5 Independence of testers
27. DM-1 to DM-5 Receiving notifications of security issues
28. DM-6 Periodic review of security defect management practice
29. SUM-1 Security Update Qualification
30. SUM-2, SUM-3 Security update documentation
31. SUM-4 Security update delivery
32. SUM-5 Timely delivery of security patches
33. SG-1, SG-2 Product defense in depth
34. SG-3 Security Hardening guidelines
35. SG-4 Security disposable guidelines
36. SG-5 Secure operation guidelines
37. SG-6 Account management guidelines
38. SG-7 Documentation Review
39. Acronyms
40. References
41. Further Pending Items

Revision History

Revision No	Date	Change description	Author	Reviewed by
001	2021-03-14	Draft secure development process	Dinesh Kumar	TBR
002	2021-08-12	Added about CVE tracking process in CIP kernel and CIP Core	Dinesh Kumar	TBR
003	2021-09-03	Added reference for File Integrity document	Dinesh Kumar	TBR
004	2022-08-01	Updated SM-11 with additional information	Dinesh Kumar	TBR

---

1. Overview

This document is based on IEC-62443-4-1 (Edition 1.0 2018-01) secure development process requirements.The Objective is to adhere IEC-62443-4-1 secure development process requirements in CIP development as much as possible.

Adherence to these secure development practices will give an edge to CIP over other distributions at the same time it will reduce IEC-62443-4-x development effort for CIP member companies for making products based on CIP.

1.1 Acronyms

Acronyms	Details
CIP	Civil Infrastructure Platform
CB	Certification Body
SWG	Security Workgroup

2. [SM-1] Secure Development Process

The development process details of CIP are provided in this document.

3. [SM-2] Identification of Responsibilities

CIP has defined roles and responsibilities for the members who are responsible for CIP development. This RACI(Responsible, Accountable, Consulted and informed) is reviewed and updated yearly once or whenever there is change in responsibilities

Detailed RACI MATRIX is available at CIP RACI matrix page.

4. [SM-3] CIP Software version

TODO: Define CIP Core and CIP Kernel versions

5 [SM-4] CIP Developer Security Expertise

TODO: Provide link for security expertise document

6 [SM-5] Process Scoping

Following three documents can be used to list met and unmet requirements by CIP.

1. exida gap assessment report
2. Secure development process document which is current document

3. Application and hardware guidelines document which describes about IEC-62443-4-2

7. [SM-6] File Integrity

Following document explains about CIP File Integrity and how user can verify integrity of CIP deliverables.

About CIP File Integrity

8. [SM-7] Development Environment Security

8.1 Development Security

Development environment security is achieved by providing restricted privileges to developers as well as all developers use certificate based authentication used by gitlab.

8.2 Production Time Security

This requirement is not applicable to CIP.

8.3 Delivery Time Security

TODO: need to be discussed and documented.

9. [SM-8] Private Key Protection

CIP maintains document which explains about CIP Private key management. Please refer following document for more details.

CIP Private Key Management

10. [SM-9] Security Risk for new or externally provided components

According to exida based on gap assessment results, since all components in CIP are developed externally, hence doing risk assessment for all components is impossible. CIP team to decide which components may pose risk to CIP platform. If a component is suspected to have known vulnerabilities then as risk assessment following steps would be taken by CIP members.

- Evaluate the open security issues/CVEs
- Categorize open CVEs from CIP perspective as low/medium/high
- If open CVEs fall in medium and high categories, do a threat modeling of the component and
  decide the mitigation
- In addition CIP users should be notified for the vulnerable component via email notification

11. [SM-10] Custom Developed Components from third party

This requirement is not applicable to CIP. This is applicable to end products.

12. [SM-11] Security Issues Assessment

CIP completely relies on Debian upstream and mainline linux kernel for security issue fixes and all issues tracking. CIP follows upstream first policy and all security issues fixes are first submitted to upstream. Even CIP member companies are advised to directly report issues and submit fixes to upstream projects.

However, CIP uses open source vulnerability scanner and open source databases for security issues and identifying and sharing open CVE details with CIP users.

CIP users should check the CVE list and decide which CVEs may impact product security.

In order to meet this requirement, CIP users are advised to take following actions.

- Regularly review CVE list shared in CIP-DEV ML
- If any CVE is critical for the product, do a risk assessment or wait for the fix to be available
- Overall, ensuring no critical security issues which may compromise product security leak to end users

CIP uses open source vulnerability scanner and open source data bases for security issues. Refer CIP CVE handling

13. [SM-12] Documented Checklist Review

CIP will need a documented checklist for tracking security practices defined by IEC62443-4-1 along with a documented process for ensuring the checklist is reviewed and updated for each release

TODO: This needs to be further discussed within CIP members, how to address this requirement.

14. [SM-13] Define Review frequency

All development process artifacts should be reviewed once in a year. The review should cover following items and review comments and observations should be documented based on IEC-62443-4-1 requirements of review evidence. 1. Issues in current development process. 2. Any critical issues reported by CIP members 3. Actions to be taken to improve on points #1 and #2

15. [SR-1, SR-3, SR-4] Product Security Context

CIP generic security context has been defined in Security Requirement document. The Security Context would be revised as and when new deployment scenarios and requirements are found.

CIP Security Requirements

16. [SR-2] Threat Model

CIP generic Threat Model has been created which defines the condition of Threat Model Review and update frequency.

Threat Model document is available at CIP Threat Model document

17. [SR-5] Security Requirements Review and Approval

Security requirements should be reviewed and approved when created. All review comments should be documented. During review following members should be invited.

- Architects/developers (those who will implement the requirements)
- Testers (those who will validate that the requirements have been met)
- Customer advocate (such as sales, marketing, product management or customer support) and Security Adviser

18. [SD-1] Secure Design Principles

1. The design shows how the system's devices and subsystems are connected, and how external actors are connected to the system.
2. The design shows all protocols used by all external actors to communicate with the system.
3. Trust boundaries are documented.
4. The design document should be updated whenever the design changes

The above mentioned details are documented here.

19. [SD-2] Defense in depth design

This requirement is not applicable to CIP. This should be met by end product owners. Defense in depth design should be created by end product owners as it depends upon end products design and what kind of security layers would be part of the defense layers.

20. [SD-3, SD-4] Security design review

- Create evidence for security design reviews
- Issues identified during security design reviews are tracked using gitlab
- Create Traceability matrix for security requirements to security design
- Create Traceability matrix from threat mitigation to security design
- Create Security guidelines for user

- Include security design best practices used in debian as well as review Design best practices being developed by OpenSSF if suitable include in CIP

Details regarding security design review and best practices in CIP are documented here based on above checklist.

21. [SI-1, SI-2] Security implementation review

- Document Debian secure coding guidelines
- Perform static code analysis or re-us from upstream for critical packages
- Code reviews should document following information
    1. Name of the person who performed the code review,
    2. The date of the code review,
    3. The results of the code review
    4. The name of the person responsible for fixing problems identified in the code review
    5. Date or indication that all problems were fixed.

22. [SVV-1] Security requirement testing

CIP Security requirements testing should be done to cover IEC-62443-4-1 testing requirements applicable to CIP, following are the IEC-62443-4-1 requirements

- Functional testing of security requirements
- Performance and scalability testing and Boundary/edge condition, stress and
  malformed or unexpected input tests not specifically targeted at security.
- General security capabilities (features);
- API (application programming interface);
- Permission delegation;
- Anti-tampering and integrity functionality;
- Signed image verification; and
- Secure storage of secrets.

23. [SVV-2] Threat Mitigation testing

Create a Traceability matrix to show each threat has mitigation and test to verify the mitigation

24. [SVV-3] Vulnerability testing

Vulnerability scanner and Pen testing Tool should be used and following testing should be done.

- Network storm testing should be done to simulate DOS attacks
- Software composition analysis must be done against the binaries
- Attack surface analysis
- Known vulnerability scanning
- Dynamic Runtime Resource Analysis Testing

- Fuzz testing on all protocols sent externally should be included as part of this testing

25. [SVV-4] Penetration testing

Penetration testing should be conducted by using some suitable tool.

26. [SVV-5] Independence of testers

Document testers involved in CIP testing and ensure testers are different than developers.

27. [DM-1 to DM-5] Receiving notifications of security issues

Security issues are tracked using CVE scanner tools for both CIP Kernel and CIP Core.

CIP CVE scanner runs periodically to fetch fixes for CVEs and apply in CIP repos.

Further details can be found about CIP Core CVE scanner at CIP Core CVE scanner

CIP Kernel CVE Scanner

CIP Kernel CVE checking is done weekly and reports are published in cip-dev mailing list.) Currently there is no specific policy to stop releasing because of missing patches as long as stable kernels are released

Release policy is reported at every E-TSC. The latest one is as follows CIP Kernel CVE fixes release policy

Further details of CIP Kernel CVE scanner can be found at CIP Kernel CVE scanner

Notification of CVE fixes are sent by email to CIP users.

CIP does not maintain it's own bug tracking system. Refer this document to see the upstream methods to handle the CVE cycle.

28. [DM-6] Periodic review of security defect management practice

Review current defect management practices and processes once in a year and make required changes as needed.

29. [SUM-1] Security Update Qualification

CIP can not produce evidence for details of testing each patch or security issues. The verification information is not kept at any central location, it's scattered at multiple locations such as KernelCI, LAVA as well as mailing list.

CIP kernelCI reports are available at Kernel CI page

CIP has LAVA automated tests which are executed when some code changes are merged. At the moment there is no tests for confirming side effects.

CIP users should meet this requirement based on the product requirement and frequency of updates needed etc.

30. [SUM-2, SUM-3] Security update documentation

CIP will continue to use mailing as main channel for sharing security issues information with all users.

31. [SUM-4] Security update delivery

CIP releases patches, CIP kernel and CIP Core meta-data which are signed by CIP developers.

32. [SUM-5] Timely delivery of security patches

TBD: Document the frequency of CIP releases.

33. [SG-1, SG-2] Product defense in depth

This requirement is not applicable to CIP. Defense in depth should be done by end products owner.

34. [SG-3] Security Hardening guidelines

CIP has defined security hardening guidelines for following. * Default security policies to meet IEC-62443-4-1 security requirements * Compilation flags * Other configs

Those details can be found in this CIP security hardening document.

35. [SG-4] Security disposable guidelines

This requirement is not applicable to CIP. Disposable guidelines are applicable to end products.

36. [SG-5] Secure operation guidelines

Following operation guidelines should be documented.

- Security Updates
- Monitoring system logs
- IPS and IDS monitoring logs

37. [SG-6] Account management guidelines

CIP users can create CIP security image which has security packages to meet IEC-62443-4-2 security requirements, follow the instructions to create CIP security image, choose security extensions option in the kas menu.

CIP security image has only one default user account which is root account with the password root. For additional user accounts, follow standard linux user management steps.

38. [SG-7] Documentation Review

All CIP development documents are maintained in gitlab. Reviewers can give comments using any of the following methods.

- Create gitlab issues with comments
- Send review comments in email
- Send MR with the changes

Only few CIP members have rights to merge review comments changes in the documents.

39. Acronyms

Acronym	Description
SM	Security Management
SR	Specification of security requirement
SD	Secure by design
SI	Secure implementation
SVV	Security verification and validation
DM	Defect Management
SUM	Security update management
SG	Security guidelines
CIP	Civil Infrastructure Platform
IPS	Intrusion detection system
IDS	Intrusion detection system

40. References

Item	Reference
IEC-62443-4-1	https://webstore.iec.ch/preview/info_iec62443-4-1%7Bed1.0%7Den.pdf

Further pending items

- Create AIs in gitlab for TODO items
- All members need to review and share comments from evidence perspective
- Fix formatting issues
- Add Back to Top button