Introduction

- Secure development is one of the important security requirement in IEC62443 standards and as part of this process implementing and reviewing secure coding standards in the code is a primary goal.
- Secure coding standards help to protect the software from introduction of security vulnerabilities that leads to potential exploits and attacks.
- This document explain how CIP Project and its upstream projects are following security coding guidelines.

CIP Project coding standards

- CIP Project contains two main components CIP Linux and CIP Core, both components follows upstream first policy, that means the development happens in upstream project and CIP uses the upstream directly to create CIP artifacts.
- As CIP Project uses upstream projects directly, so CIP developers follows the Upstream project coding standards while fixing or developing any feature in upstream projects.
- Below are the CIP Project repositories
  - CIP Core
    - Cip-ISAR: https://gitlab.com/cip-project/cip-core/isar-cip-core
    - Deby: https://gitlab.com/cip-project/cip-core/deby
  - CIP Linux: https://gitlab.com/cip-project/cip-kernel/linux-cip

CIP Upstream projects coding standards

- Below are the some of the upstream projects that are used in CIP and their coding standards as defined in their upstream.
  1. CIP Linux: https://www.kernel.org/doc/html/v4.10/process/coding-style.html
  2. Openssh: https://man.openbsd.org/style
  3. Openssl: https://www.openssl.org/policies/codingstyle.txt
- Many upstream projects doesn't define or not published in public the coding standards, such projects may be internally can be validated or review the coding standards using some tools.

Tools to assist security code review

Below are the tools can be used to validate the coding standards or used to review code * Flawfinder * RATS * pscan