ISO 21434 Fact Sheet

Road Vehicles - Cybersecurity Engineering

What's New in ISO 21434?

ISO 21434 prioritizes security throughout the entire lifecycle of a vehicle. This means that car manufacturers and OEMs will need to display due diligence regarding software security. To reach ISO 21434 compliance, many automotive software firms will have to adapt their security measures. Besides specified testing practices, ISO 21434 also provides guidelines for processes, communication and risk management.

Executive Summary of ISO 21434

ISO/SAE 21434 contains over 85 pages of regulations for secure software development. The document can be divided into 3 sections:

General Requirements and Recommendations

This section contains objectives and descriptions, how cybersecurity should be implemented within the automotive industry. It covers governance issues, cybersecurity culture, and organizational aspects of cybersecurity. It also defines requirements for risk assessments, vulnerability analysis, and threat monitoring, with a focus on:

- Cybersecurity Management
- Project Dependencies
- Continuous Cybersecurity Activities
- Risk Assessment Methods

ISO 21434 Fact Sheet

Road Vehicles - Cybersecurity Engineering

Regulations for Software Development

This section contains requirements and recommendations, regarding key metrics, architectural design, and requests for secure software development, along the entire software development process.

- Concept Phase
- Product Development
- Cybersecurity Validation
- Production
- Operating and Maintenance
- Decommissioning
- Distributed Cybersecurity Activities

Guidelines and Templates

The ISO 21434 contains an extended appendix, with numerous examples and best practices for cybersecurity culture and security governance, but also further information and helpful guidelines and templates, for example for risk determination or impact ratings.

You can find an in-depth summary of ISO 21434 here.

Best Practice:
How CARIAD Prepared for ISO 21434

CARIAD, Volkswagen’s new software house provides reliable software and digital best practice for the entire Volkswagen group. Here is how they improved Volkswagen’s software security, in compliance with ISO 21434.

6 TIPS THAT WILL HELP YOU COMPLY WITH ISO 21434

1 Security Culture – Make Security a Priority

The first step towards secure software is cultural acceptance. For a healthy security culture, everyone involved in software development should be educated and made aware of basic security protocols. Similar as in DevSecOps, this means that security controls should be employed directly within the various delivery teams, throughout all stages of the software development lifecycle.

2 Choose the Right Language

To meet the requirements of ISO 21434 security aspects should be kept in mind when choosing a programming language. A secure language is characterized by

- unambiguous syntax
- two semantic definitions
- secure design and coding techniques.

3 Use Language Subsets

A language subset is a carved-out part of a programming language that can help mitigate any activity that poses a security risk. Using a language subset can improve the security of an infrastructure to make it more compliant with ISO 21434 requirements. Since C/C++ is very common in the automotive industry, creating a subset for it is highly recommended. ISO 21434 explicitly recommends compliance with the coding standard MISRA C:2012 revision 1 and 3 the CERT C guidelines for C/C++. Both are centred around language subsets.

6 TIPS THAT WILL HELP YOU COMPLY WITH ISO 21434

4

Enforce Strong Typing

Strong typing means that developers will have to abide by strict typing rules. This can cause exceptions and errors during compiling. Still, it prevents unpredictable results and ensures good code quality. Strong typing is recommended in both the MISRA C:2012 revision 1 and the CERT C guidelines.

5

Create Guidelines for Defensive Implementation

Defensive implementation means preparing a software for unpredictable events. To foresee this, it is crucial to have an understanding of possibly tainted data and the order of evaluation of arithmetic functions. Basically, this means that the code should be easily understandable. To achieve this, you should implement recognized coding guidelines. For C/C++, these guidelines can be found in MISRA C: 2012 revision 1 and CERT C.

6

Implement Fuzzing Into Your CI/CD

Achieving ISO 21434 compliance with external pentests would require a time-consuming increase of pentesting resources and make OEMs and suppliers even more reliant on external testing bodies. ISO 21434 recommends fuzz testing as a method to cope with its increased testing requirements. Instead of outsourcing more pentests, modern fuzz testing can be used to partially automate them. With the addition of automated fuzz testing, dev teams can run continuous security tests. Finding and fixing bugs throughout CI/CD speeds up the development process, as it reduces the need for a large-scale bug-rampdown later on.