GRAMMATECH

Role of SAST and SCA in ISO/SAE 21434 - Road Vehicles Cybersecurity Engineering

As a leading developer of software-assurance and advanced cybersecurity solutions, GrammaTech helps its customers and partners solve the most challenging software issues and safeguards mission-critical software and devices from failure and cyberattack.

GrammaTech solutions include:

CodeSonar: Seamlessly integrates static application security testing (SAST) into DevSecOps processes to analyze source and binary code, address safety and security issues early, improve code quality throughout the software development life cycle and accelerate projects.

CodeSentry: Quickly performs binary software composition analysis on third-party software without access to source code to identify open source components, generate a comprehensive SBOM, detect vulnerabilities and get an overall security and risk score.

As cars become more connected and complex, the amount of software needed is staggering. With 100 million lines of code being standard for current vehicles and up to 300 million for autonomous vehicles. Along with this connectivity and complexity is an ever-increasing cyber-attack surface. Battling this security threat and improving the cybersecurity engineering of automotive software is the goal of ISO/SAE 21434.

This relatively new standard is a descendant of the existing ISO 26262 standard for automotive safety with the intended goal of defining objectives, requirements and guidelines for securing electrical and electronic systems in automobiles.

Rather than evaluating each major clause of ISO/SAE 21434 and analyzing how it applies to tools like SAST and SCA, this post looks at what the standard says about key areas of security and where these tools apply.

Secure Development and Code Testing

ISO/SAE 21434 is clear on the need for a top-down organizational approach to security. From management, corporate culture, project management, supply chain and continuous processes, security needs to be part of each decision-making step. Security is designed in, not added on or “tested in.”

When it comes to specifics about software development practices and the use of security tools, this is spelled out in clause 10 (Product Development) and clause 11 (Cybersecurity Validation.) The objective in product development is to ensure define, design and validate security specifications while also identifying weaknesses and building an audit trail from specification to verification and validation. In cybersecurity validation, the goal is to prove the security claims made for the product and ensure no unreasonable security risks remain.

Integration and Verification

In clause 10.4.2 of ISO/SAE 21434, static analysis is specifically mentioned as a recommended method for verifying security specifications. This also calls for “conformity with the modelling, design and coding guidelines of [RQ-10-05], if applicable.” [RQ-10-05] calls for “for suitable design, modelling or programming languages for cybersecurity that are not addressed by the language itself shall be covered by design, modelling and coding guidelines, or by the development environment.” In other words, use of coding guidelines such as MISRA C or CERT C and other methods of enforcing good programming practices.

GRAMMATECH

How SAST Tools like GrammaTech CodeSonar Help

SAST tools are useful in augmenting existing implementation and testing practices and are meant to provide an additional source of discovery for defects and vulnerabilities. Consider the following strengths of SAST tools which apply for both secure and safety critical development.

- **Enforcing coding standards for safety, security, and style**. Automating code analysis during code development ensures quality in the development stream every day.
- **Reducing manual effort in proving software robustness and behavior**. SAST tools augment software testing by providing more assurance of software quality.
- **Reducing number of defects throughout development**. Code that works the first time is much cheaper to test and integrate than buggy code. Bugs removed from the code before testing (or even source configuration management) reduces costs and risk.
- **Finding serious defects that elude testing.** Despite the testing rigor required for automotive software, SAST tools have found defects that were missed. These are the most worrisome types of defects.
- **Accelerating certification evidence**. Documenting the results of cybersecurity validation is critical to proving compliance to certification standards. SAST tools have rich reporting features to help support certification requirements.

In addition, SAST tools help with vulnerability detection and discovery which is part of the product development in keeping with clause 11, ensuring no unreasonable risk remains in the product. For example, SAST tools provide the following capabilities:

- **Shift left vulnerability detection and prevention**: SAST tools such as GrammaTech CodeSonar are integrated with developer's development environment and project build systems. Early detection of poor security practices and possible vulnerabilities are detected as soon as the code is written. Preventing these kinds of security issues before they enter code repositories or unit testing saves downstream resources.
- **Continuous source code assessment**: SAST is often applied initially to a large codebase as part of its initial integration, however where it really shines is after an initial code quality, safety and security baseline is established. As each new code block is written (file or function), it can be scanned by the SAST tools and developers can deal with the errors and warnings quickly and efficiently before checking code into the build system.
- **Tainted data detection and analysis**: Analysis of the data flows from sources (i.e. interfaces) to sinks (where data gets used in a program) is critical in detecting potential vulnerabilities from tainted data. Any input, whether from a user interface or network connection, if used unchecked, is a potential

security vulnerability. Code injection and data leakage are possible outcomes of these attacks which can have serious consequences.

Software Reuse and the Supply Chain

ISO/SAE 21434 is very clear on the fact that "cybersecurity risk management is applied throughout the supply chain to support cybersecurity engineering." So, security practices and controls need to be applied to suppliers and supplied and reused components, including software. Among the various tiers of suppliers, a cybersecurity interface agreement is used to support "distributed cybersecurity activities" with the intention of having common guidelines and procedures among suppliers and OEMs.

There is also the expectation that suppliers also conform to the standard, "The capability of a candidate supplier to develop and, if applicable, perform post-development activities in accordance with this document shall be evaluated." And "To support a customer's evaluation of supplier capability, a supplier should provide a record of cybersecurity capability." This would include evidence of the supplier's capability concerning cybersecurity such as best practices from development, post-development, governance, quality, and information security. Any transaction in the supply chain is required to conform to ISO/SAE 21434.

Although software bill of materials (SBOM) are not mentioned here, they are a significant component of software transactions amongst suppliers. They play an important part in vulnerability disclosure and ensuring due diligence in the supplier's processes.

How Do SBOMs Benefit the Automotive Software Development?

Aligned with ISO/SAE 21434, adopting software supply chain risk management and using SBOMs to facilitate this goes a long way to improving security posture.

As with physical BOMs which are used to manage the parts supply chain, SBOMs help monitor and manage software components for security vulnerabilities and licensing issues. This also means better supplier decisions based upon actionable information in SBOMs.

Integration of software composition analysis (SCA) in this manner and using SBOMs as a critical development artifact on a regular basis, has many benefits, including:

- **Discover:** Identify open source components in third-party code and COTS/third-party software. Detect known (N-day) and unknown (Zero-day) vulnerabilities in those components.
- **Manage**: Make more intelligent security decisions based on visibility into code/software. Adhere to security, licensing and vendor risk compliance requirements.
- **Remediate**: Protect against cybersecurity threats with actionable vulnerability intelligence. Streamline vulnerability remediation to mitigate software risk.

How Tools like CodeSentry Help

Tools such as GrammaTech CodeSentry can analyze open source, third-party and commercial off the shelf (COTS) software and determine the constituent components even when the only available media is binary

files. In doing do, it generates an SBOM and vulnerability report which determines the risk the SOUP component poses. SBOMs also provide:

- **Identifying and avoiding vulnerabilities** in reused components in your own developed software and software purchased by your organization.
- **Managing software supply chain risk** to remove and reduce the unknown security risk in reused software. SBOMs provide data for business decisions on software purchases and open source reuse.
- **Supply chain qualification** to ensure consistency and accountability from suppliers. Suppliers than meet the SBOM requirements during procurement are given preferential treatment.
- **Improved security and downstream benefits** that come with risk management and mitigation. Avoiding and catching security risks before they become embedded in their product pays huge dividends during development and deployment of your products.
- **Common understanding of software assets** that comes with a standardized SBOM amongst software developers, suppliers and open source projects. SBOMs become a way to communicate software contents and dependencies within and outside an organization.

SBOMs are an important artifact in the software supply chain and will become the common way to assure the provenance of software acquired in automotive software.

Summary

Clearly there is an important role for SBOMs, SCA and SAST tools in the development of safe and secure automobile software. Tools are an important part of security assurance during development and testing of code used in automotive systems. To ensure the security and integrity of the software supply chain, SCA tools play an important part in generating and verifying SBOMs for open source and third-party software.

In addition, SAST tools help software development team follow the guidelines and standards for ensuring software quality, safety and security. Used in conjunction with continuous integration and delivery pipelines, SAST tools automate the detection and prevention of vulnerabilities in some cases before they enter the code repository.

SCA and SAST tools play an increasingly important role in demonstrating due diligence by manufacturers, an important part of conforming to standards likes ISO/IEC 21434.

Contact
Nohau Sweden
Phone Sales: +46 (0) 40 59 22 00
sales@nohau.se

Nohau Denmark
Phone Sales: +45 44 52 16 50
info@nohau.dk

Nohau Finland and Baltic
Phone Sales: +358 40 546 1469
sales@nohau.fi