Dynamic Analysis Report

Revision

Version 2
11/15/21 10:15 AM

SME

Matthew Bourdua

Abstract

This document describes the process to create a dynamic analysis report.

Group / Owner

Development / Software Developer

Motivation

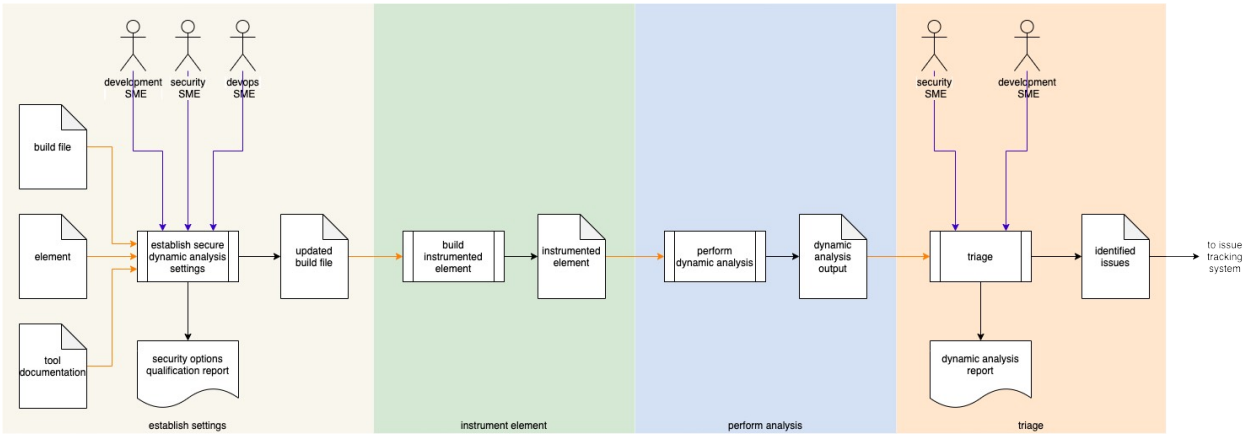
This document is motivated by the need to have runtime-specific, security-related feedback in the development of software for use within safety-critical, cyber-physical systems for certification of compliance to standards such as **ISO 21434** and **ISO 26262**.

License

Overview

Although the quality of security-related feedback from the compiler and static analysis tools has become much better over time, there remains much these tools do not consider. This is where dynamic analysis comes in. Whether code coverage or stack patterns, data gathered during this dynamic analysis is very helpful in identifying security-related issues.

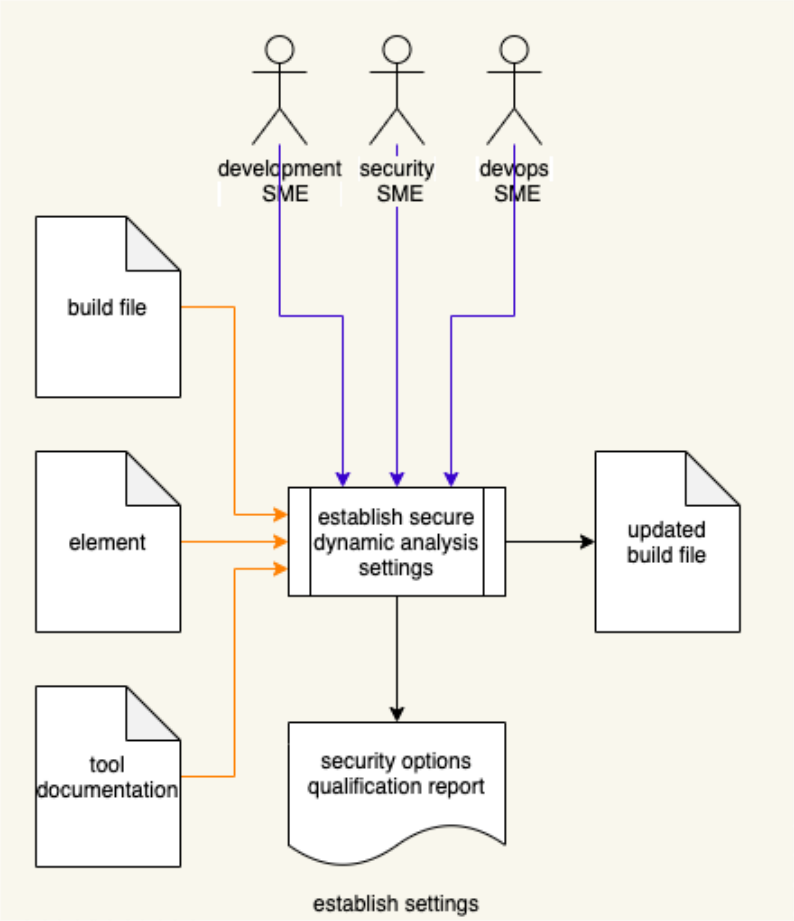
The following diagram illustrates the process to be used:



Process

Establish Settings

Inputs	Build file Element Tool documentation
Outputs	Updated build file Security options qualification report
Participants	Development SME Security SME DevOps SME



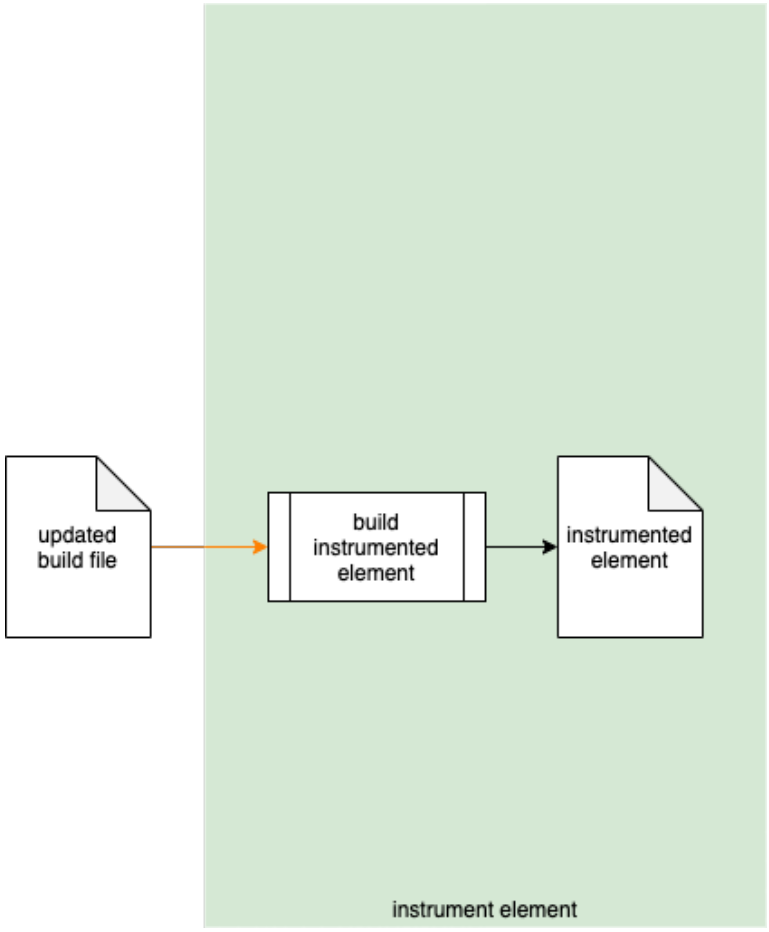
Using the **Element** under consideration, dynamic analysis **Tool Documentation**, and **Build File**; the Development SME, Security SME, and Devops SME will work together following the **Secure Settings Document** [2] process to determine what element aspect needs to be tested and

whether the element will require any kind of instrumentation to enable that testing. An **Updated Build File** will be produced. A **Security Options Qualifications Report** is generated.

Note: The scope of the element may be as small as a single file or as large as the entire project.

Instrument Element

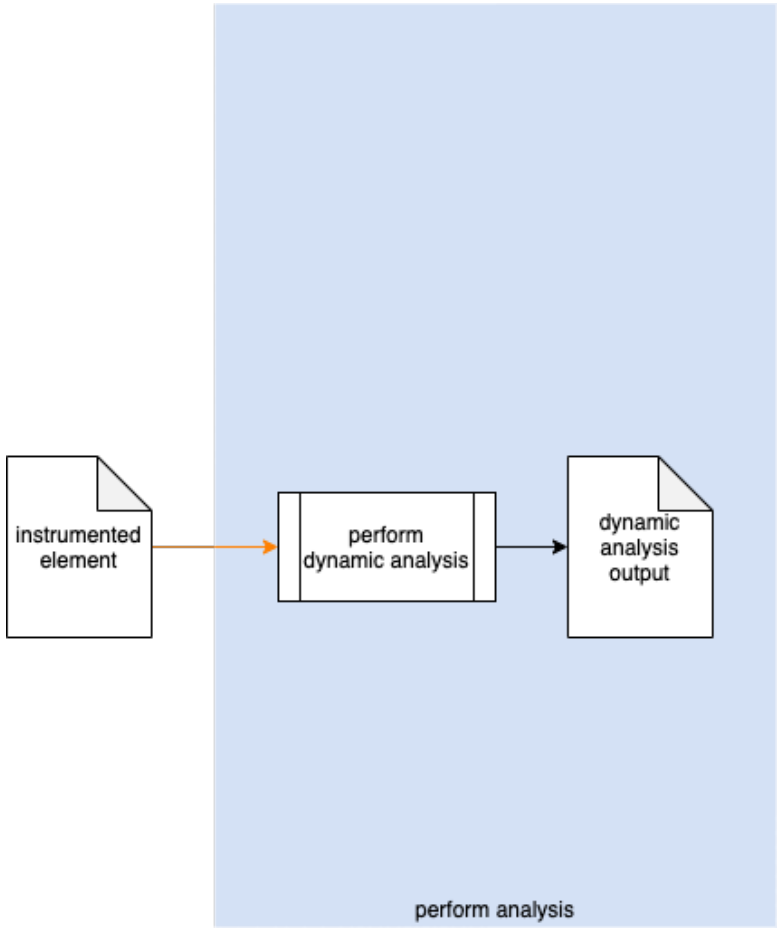
Inputs	Updated build file
Outputs	Instrumented element
Participants	**none**



Using the **Updated Build File,** the build system creates an **Instrumented Element**.

Perform Analysis

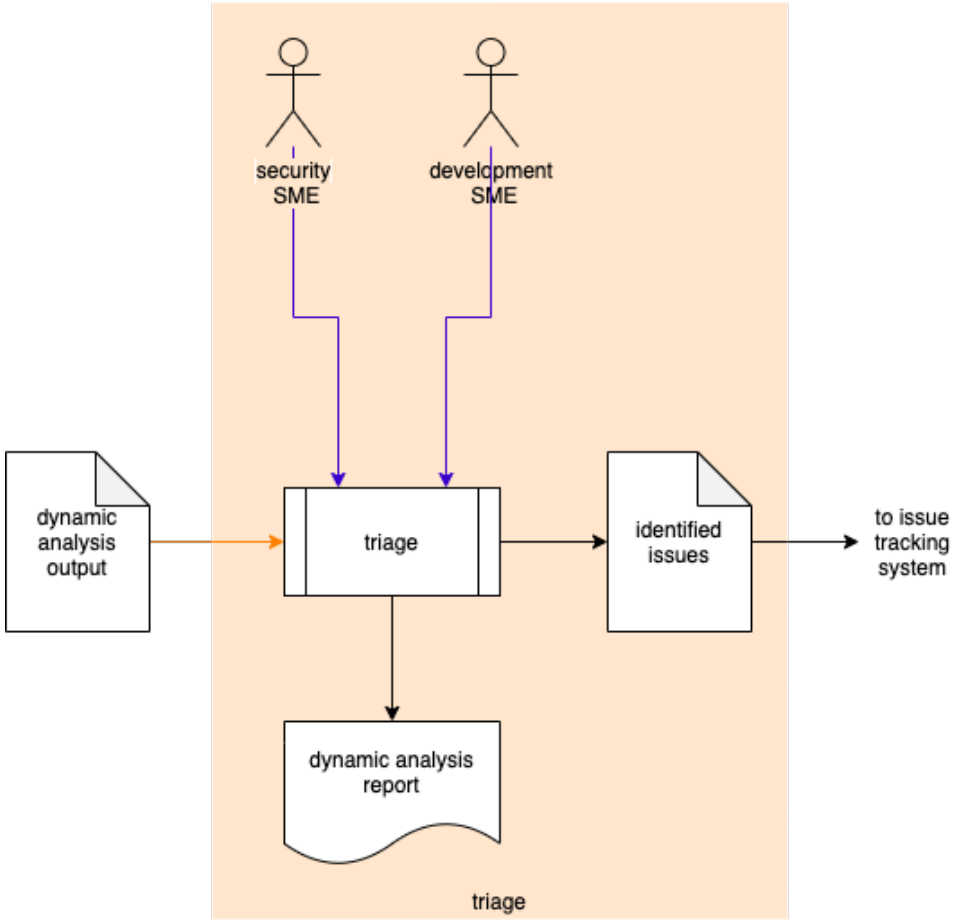
Inputs	Instrumented element
Outputs	Dynamic analysis output
Participants	**none**



The build system invokes the **Instrumented Element**. The generated output is captured into **Dynamic Analysis Output**.

Triage

Inputs	Dynamic analysis output
Outputs	Identified issues Dynamic analysis report
Participants	Security SME Development SME



The Security SME and the Development SME review the **Dynamic Analysis Output** to identify any anomalous behavior needing further investigation. For any such anomalous behavior, an issue will be created in the issue tracking system. A **Dynamic Analysis Report** will be generated.

Identified Issues

The recommended form of the **Identified Issues** artifact is a Static Analysis Results Interchange Format (**SARIF**) encoded JSON. This document assumes SARIF version 2.1.0 [1] or later.

Dynamic Analysis Report

The **Dynamic Analysis Report** is recommended to be produced from the **Identified Issues** artifact and should detail the issues exposed by the dynamic analysis.

The report contains one or more analysis runs. Each run includes:

- Description of the tool used
- Description of the instrumented element
- Results of the analysis

The tool description includes:

- Name
- Version
- URI to tool documentation
- Rules applied
- Reference (optional) to associated taxonomy entry (Common Weakness Enumeration, …)

The tool rules (one or more) convey the classes of analysis performed (software crash, time out, memory limits). Each includes:

- ID (unique)
- Name
- Short description of the rule
- Full description of the rule
- URI to rule documentation

The instrumented element description (one or more) provides information related to the element under consideration. Each includes:

- URI to instrumented element
- URI to repository the element came from
- Element properties

The element properties further describe the element. Each includes:

- Build switches required to perform dynamic analysis on the element (`-g, -O0, -fsanitize=address`, …)
- Environment (ARMv8, QNX10, …)
- Compiler (gcc, clang, …)
- Analysis type (memory pressure, address sanitization, memory sanitization, …)

The analysis results (one or more) describe the issues exposed by the analysis. Each includes:

- Human-readable description
- Location within the element of the issue
- Reference (optional) to the associated taxonomy entry

References

1. **Static Analysis Results Interchange Format (SARIF) Version 2.1.0**
   https://docs.oasis-open.org/sarif/sarif/v2.1.0/os/sarif-v2.1.0-os.pdf
2. **Secure Settings Document** (AVCDL secondary document)
3. **Fuzz Testing Report** (AVCDL secondary document)
4. **Static Analysis Report** (AVCDL secondary document)
5. **Security Options Qualification Report** (AVCDL tertiary document)