Software Deployment Report

Revision

Version 2
1/24/22 1:25 PM

SME

Charles Wilson

Abstract

This document describes the content of the software deployment report.

Group / Owner

DevOps / Information Systems Security Developer

Motivation

This document is motivated by the need to have formal processes in place tracking the software and configuration information deployed onto safety-critical, cyber-physical systems for certification of compliance to standards such as **ISO 21434** and **ISO 26262**.

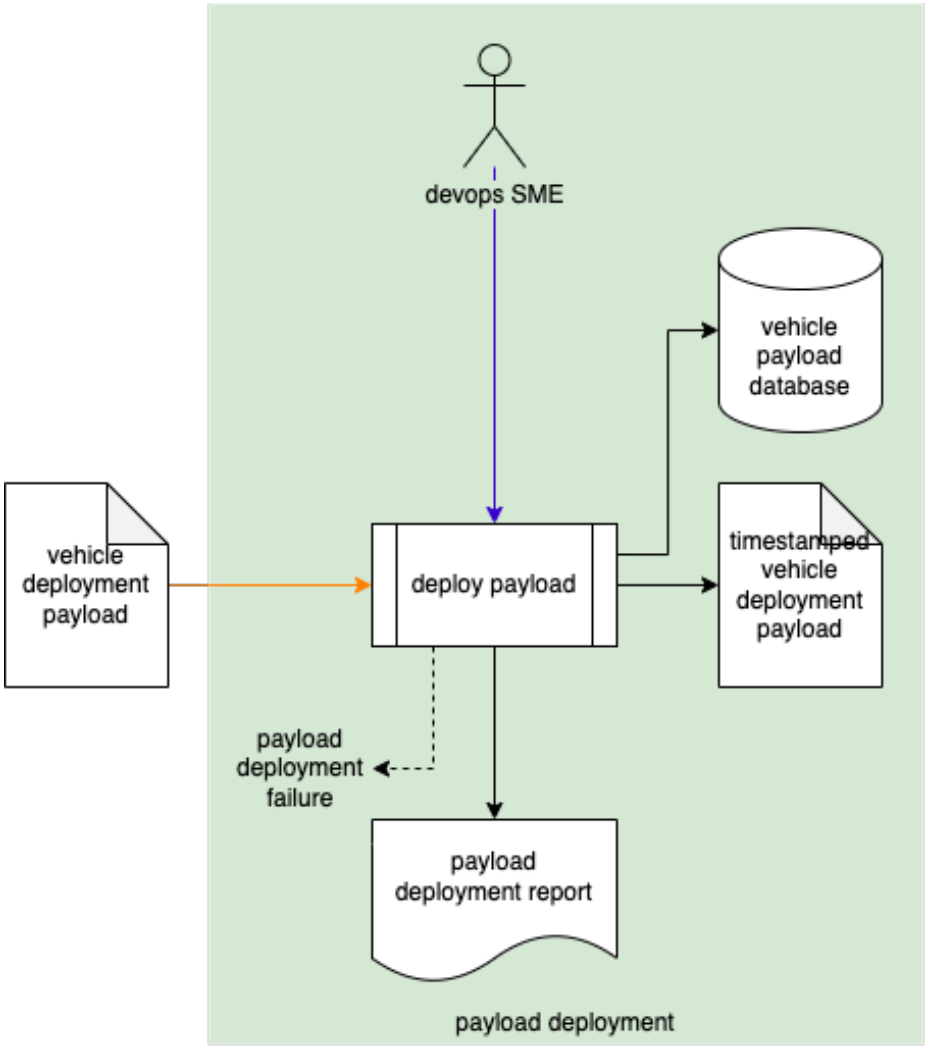
License

Overview

It is critical that all software and configuration information necessary to the operation of a product are archived and that a manifest of these materials is created. This serves to ensure that the product release can be updated should the need arise. It also provides a mechanism to allow for the quick determination of vulnerability for any of the elements used in the product's creation. It can also serve as a baseline for determination of attempted alteration of the system.

Process

The **Software Deployment Report** can be considered the same as or a subset of the **Payload Deployment Report** called out in the **Deployment Plan** [4]. The report is generated during the **Payload Deployment** activity (shown below).



Report

There are three sections contained in the report:

- Summary
- Software manifest information
- Configuration information

Summary

The summary section contains general information including, but not limited to:

- Deployment date
- Deployment package name
- Deployment package verification information (checksum, cryptographic hash, …)
- Deployment target device
- Deployment target device unique identifier

Software Manifest Information

The software manifest is a list of uniquely identified software information showing the general composition of the deployed software.

Note: The preferred executable software information representation form is the Software Identification Tag format (**SWID**) [2,3].

The software manifest is expected to contain a hierarchical representation of the software to be deployed to the system (or element). The scope of the manifest will be determined by whether the deployment is a full or incremental (update) one. The following shows the types of changes that the SWID encodes.

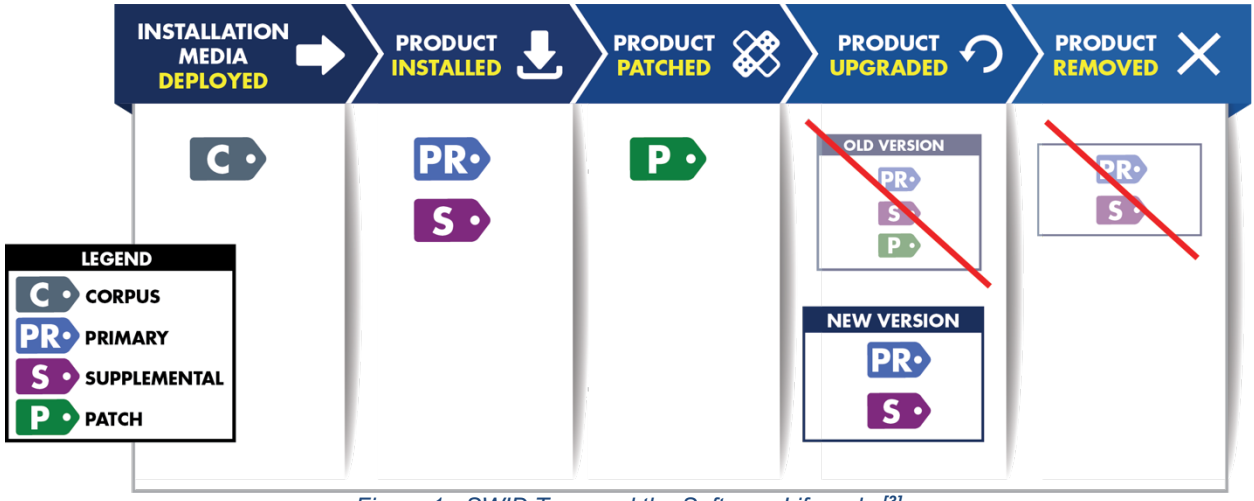


Figure 1 - SWID Tags and the Software Lifecycle [3]

This section of the report is hierarchically structured mirroring the target.

It is not the intent of this document to reiterate the content and capabilities of **SWID**. For each software deployed, the information from the **`<SoftwareIdentity>`** tag should be included.

- Name
- Version
- Tag ID
- Patch event (update, remove, add)

Note: This list represents the minimum information required to uniquely identify the software being deployed. Additional information as may be helpful (previous version, entity, …)

Configuration Information

The configuration information is a collection of the data used to configure the system's software and hardware presented in human-readable form. There are numerous encoding formats for configuration file data. This process does not have a recommendation for a preferred encoding.

Note: Information in this section of the report is not intended to be derived from or conform to the SWID-based manifest.

The configuration information should be presented hierarchically (element, sub-system, setting). Information should include:

- Name
- Value
- Event (update, remove, add)

Note: This list represents the minimum information required. Additional information as may be helpful (previous value, …)

References

1. **Archive Manifest** (AVCDL secondary document)
2. ISO 19770-2:2015 **Information technology - IT asset management - Part 2: Software identification tag**
   https://www.iso.org/standard/65666.html
3. NIST IR 8060 **Guidelines for the Creation of Interoperable Software Identification (SWID) Tags**
   https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8060.pdf
4. **Deployment Plan** (AVCDL secondary document)