Software Bill of Materials Lifecycle

Revision

Version 5
4/19/24 2:33 PM

Author

Charles Wilson

Abstract

This document describes the lifecycle of a software bill of materials (SBOM) within the context of the **AVCDL**.

Audience

The audience of this document are the cybersecurity development lifecycle practice leads who will be guiding **AVCDL** adoption within their organization.

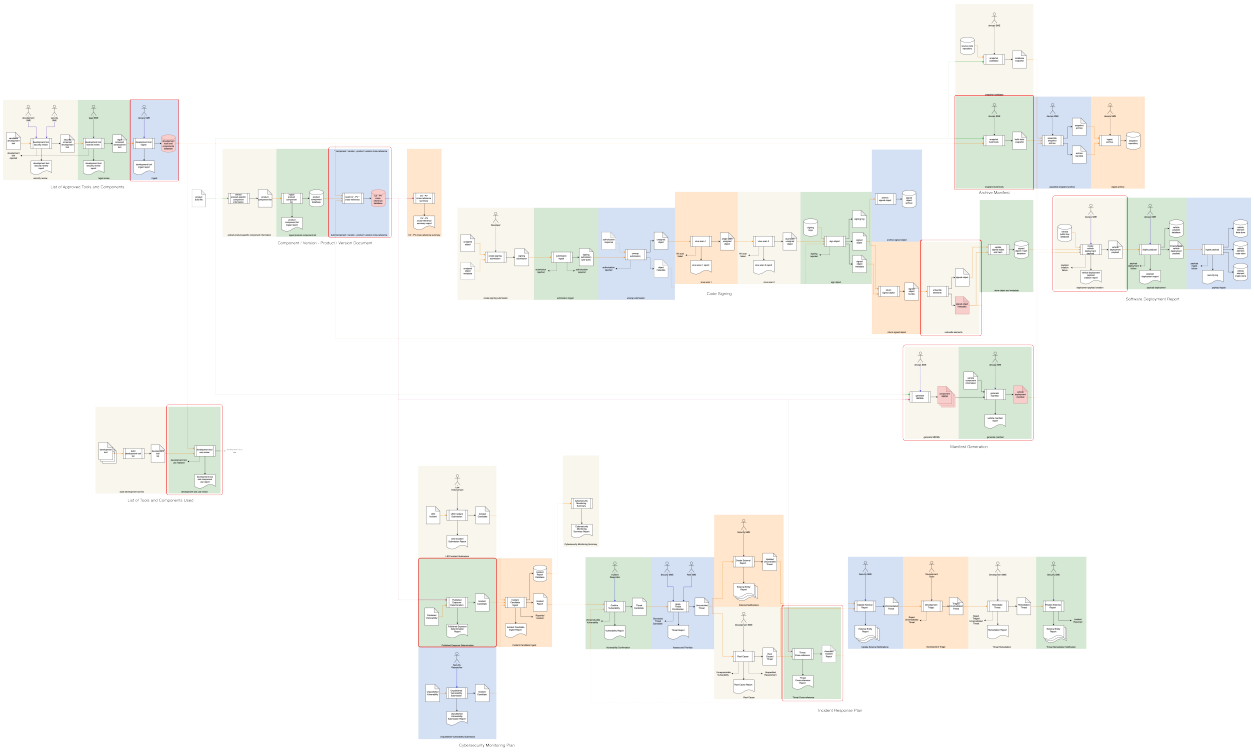
Note: This document is not subject to certification body review.

License

Overview

The SBOM is an artifact created during the development lifecycle that provides information directly supporting product cybersecurity. Although it is alluded to, the **AVCDL** [1] does not require the creation of an SBOM. This document will elaborate on the processes within the **AVCDL** that are involved in the overall SBOM lifecycle.

The following shows processes within the **AVCDL** that interact within the SBOM's lifetime.



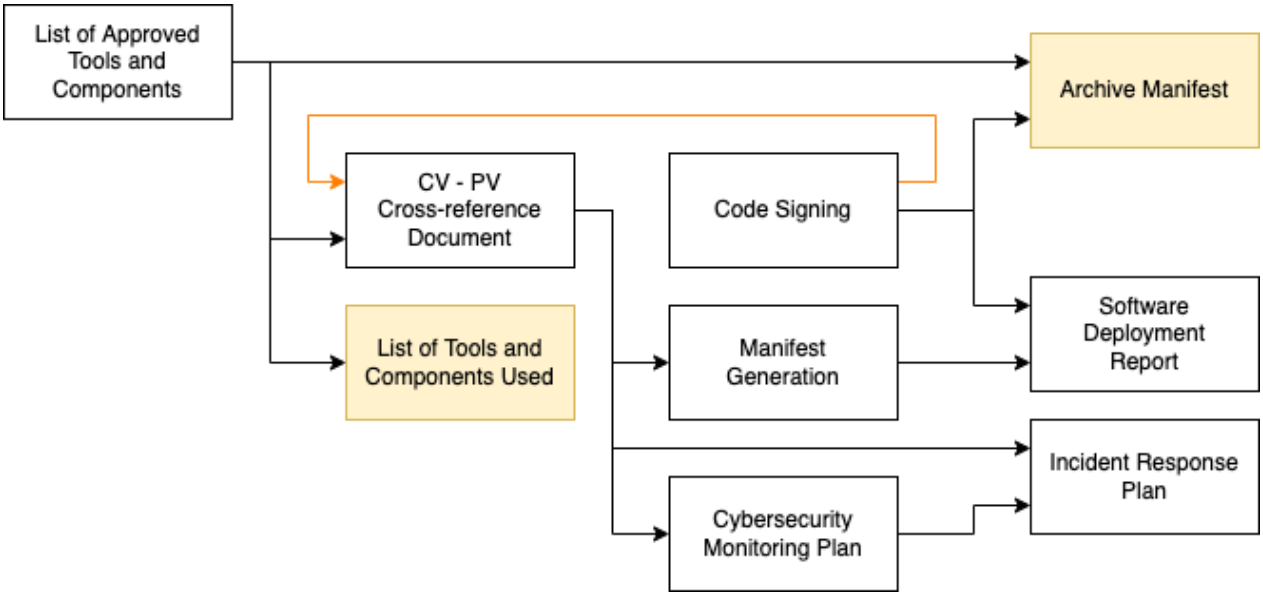
Here the various **AVCDL** processes have be connected into a single workflow diagram. This allows us to appreciate the scope of the SBOM material's lifetime.

Note: Items enclosed in red represent activities directly supporting SBOM-related activities.

Note: It is recommended that if the reader wishes to view or work with this diagram that they refer to the source of this document in the **AVCDL** GitHub repository.

Simplified SBOM Interactions Visualized

You can't really gather much from the above diagram. Let's first look at just the major process interactions.



There are seven processes that leverage information typically embodied in an SBOM. Each of the above blocks represents the **AVCDL** secondary document by the same name. It is important to observe that the flow of SBOM-related information is not entirely in one direction. During the code signing process [11], additional information is fed back into the cross-reference process [4].

Note: Two additional (highlighted) processes (**Archive Manifest**, **List of Tools and Components Used**) have been included as they use the database created by the **List of Approved Tools and Components** process.

SBOM Information Storage

It would be both impractical and error-prone to pass SBOM-related information between processes using an SBOM or similar artifacts. This information needs to be centralized, easily updated, queryable, and available to multiple processes. This implies the use of a database. The creation of an SBOM / manifest artifact is only a small part of the overall component information lifecycle. As can be seen above there are many processes which consume information related to an SBOM.

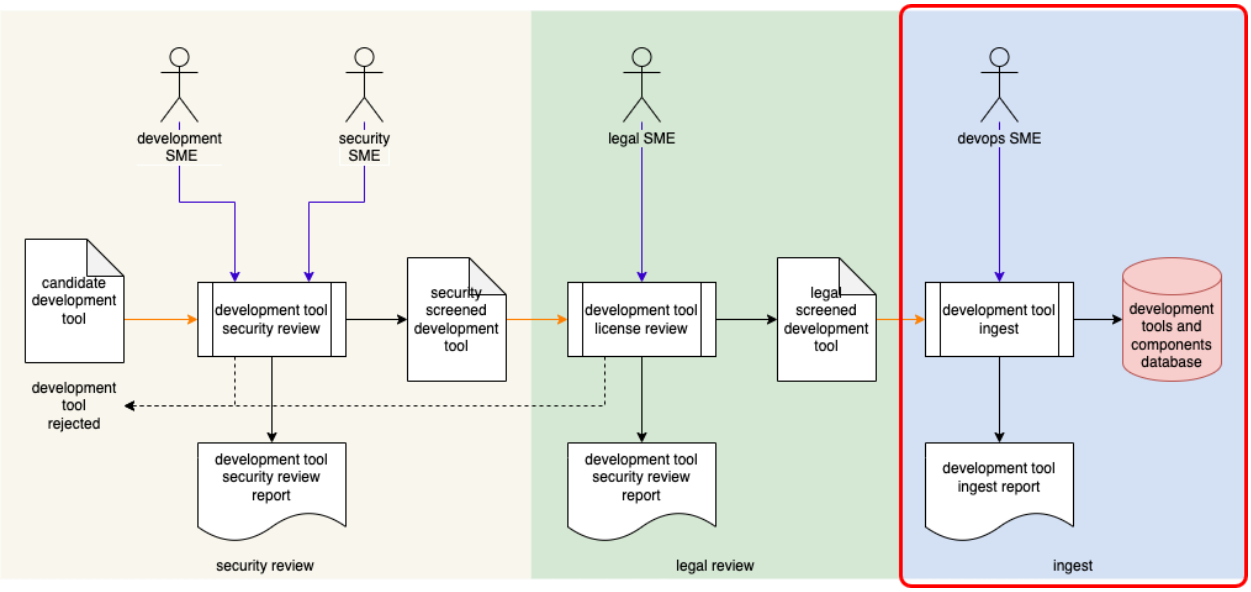
SBOM Interaction by AVCDL Process

In this section, we'll look at the interactions from each of the nine **AVCDL** processes impacting the SBOM lifecycle.

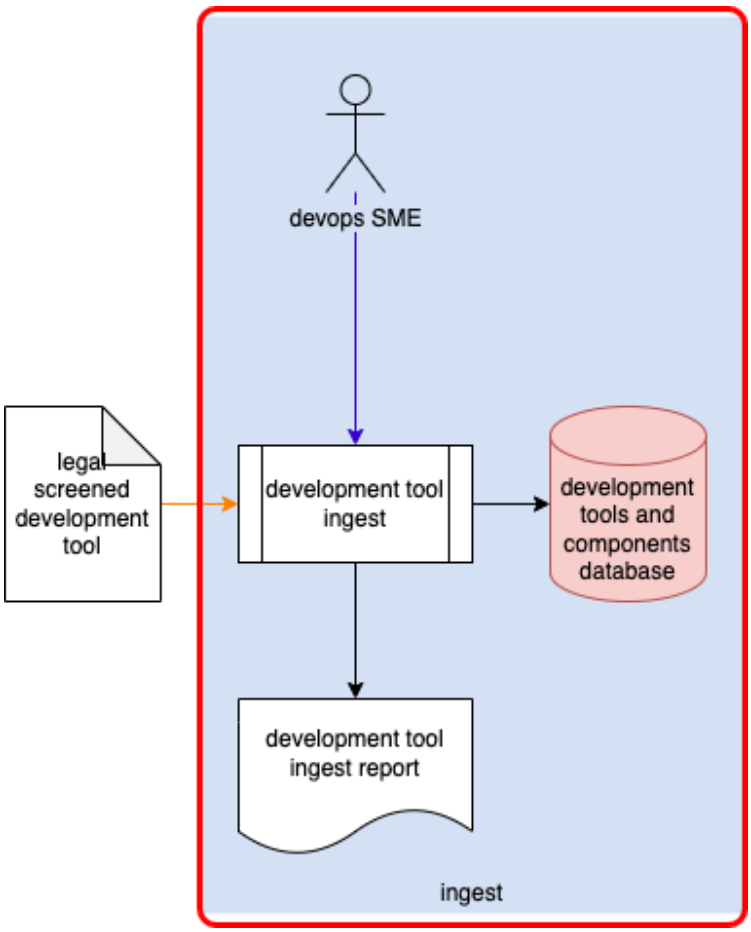
Note: This is not a reiteration of the material within these processes, but rather an elaboration on their SBOM-relevant aspects.

List of Approved Tools and Components [7]

The lifecycle of an SBOM begins with the ingest of information relating to the tools and components (applications, libraries, source code, …) used to build the product.



The activity of interest is the **ingest** step.



There is no specified mechanism by which the information regarding the tools and components is extracted. Common mechanisms include (but not limited to):

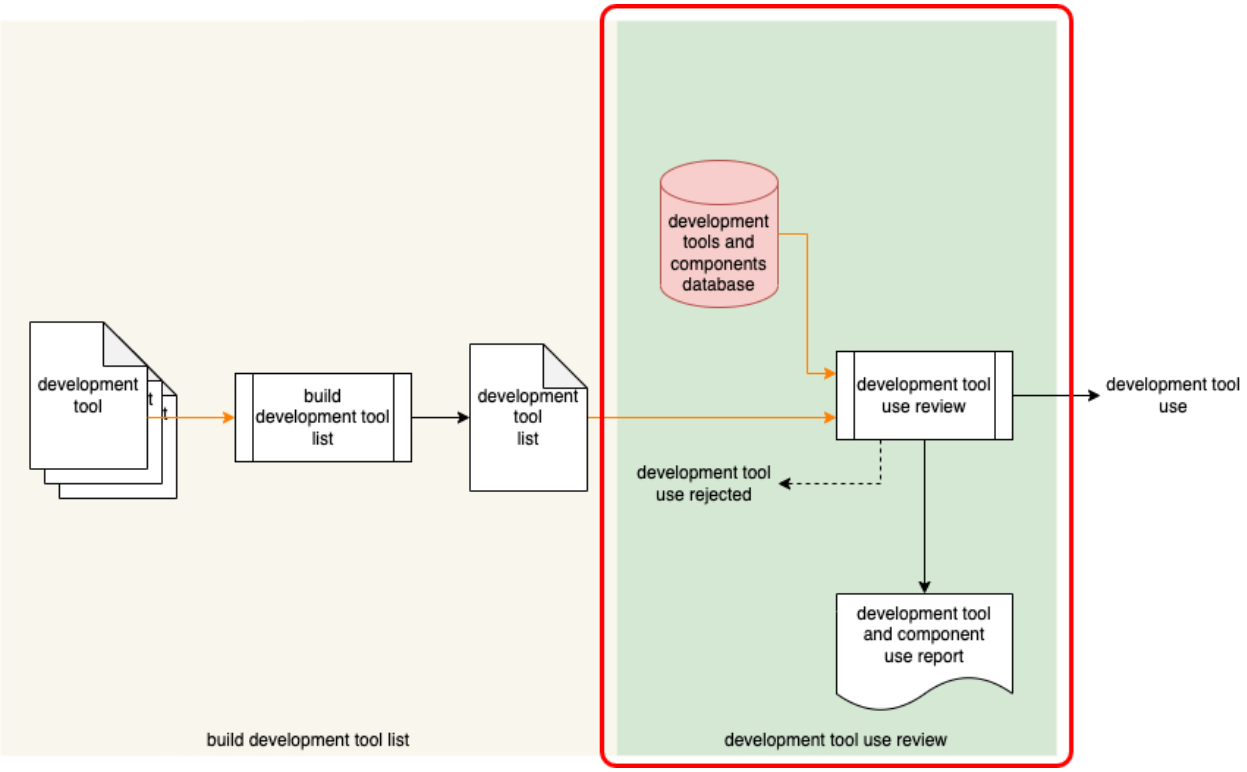
- Import from provided SBOM artifact
- Manual assembly from documentation
- Automatic assembly from forensic tools

This information is ingested into a **development tools and components database** for later use.

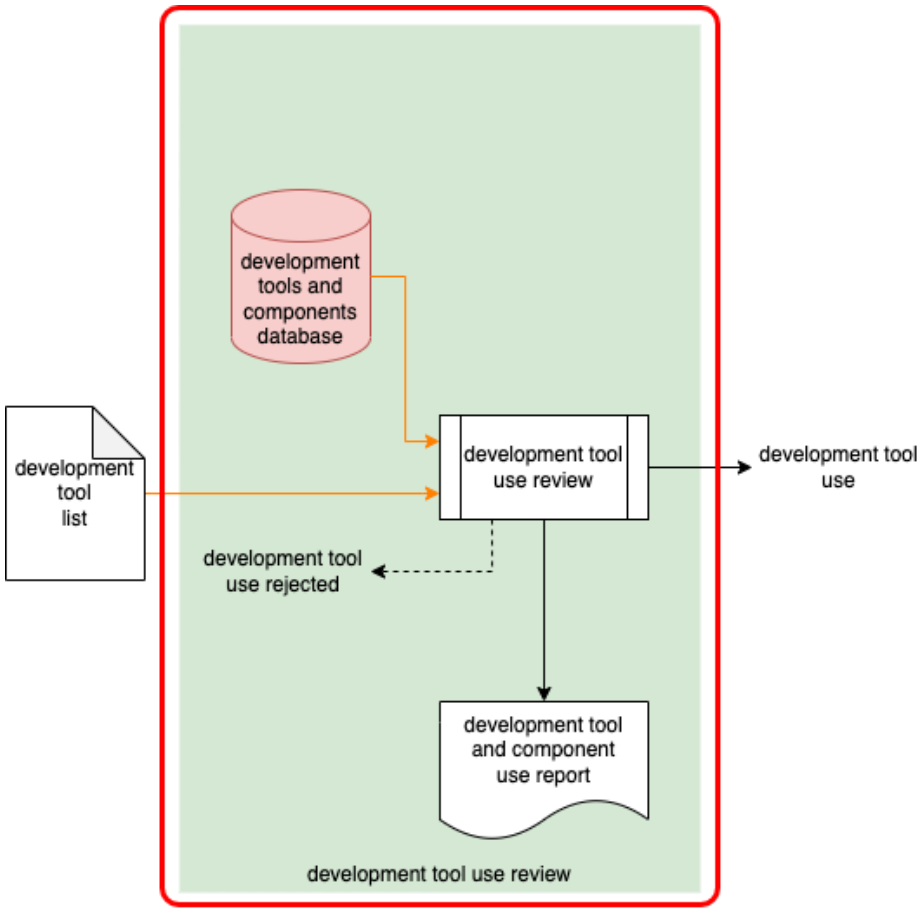
List of Tools and Components Used [8]

Note: This process does not directly impact the SBOM but is of interest in the overall SBOM lifecycle.

The **List of Tools and Components Used** process is responsible for ensuring that the build is only performed when qualified tools and components are being used.



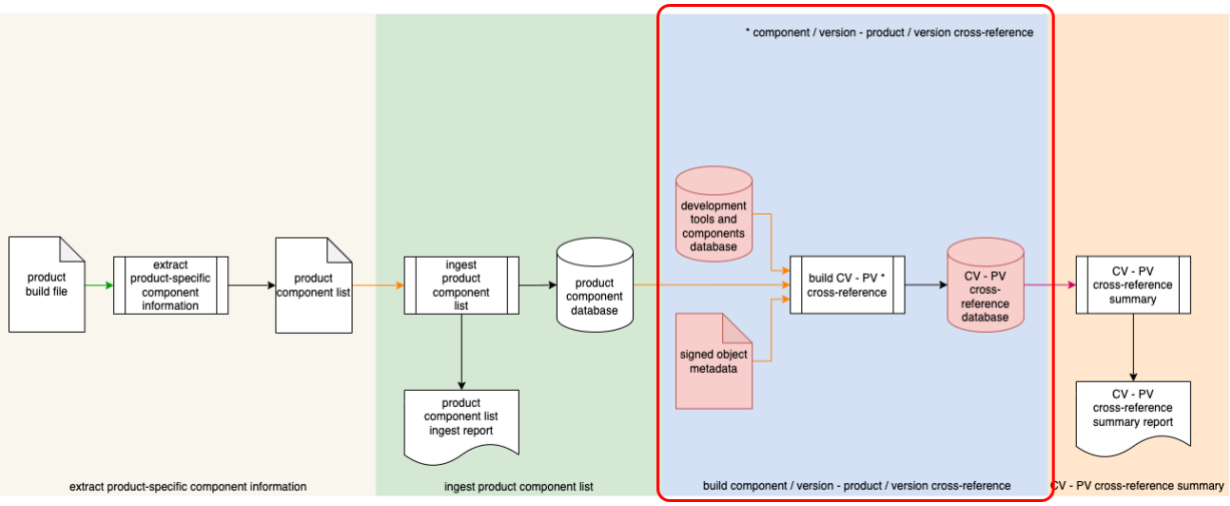
The focus here is **development tool use review** activity.



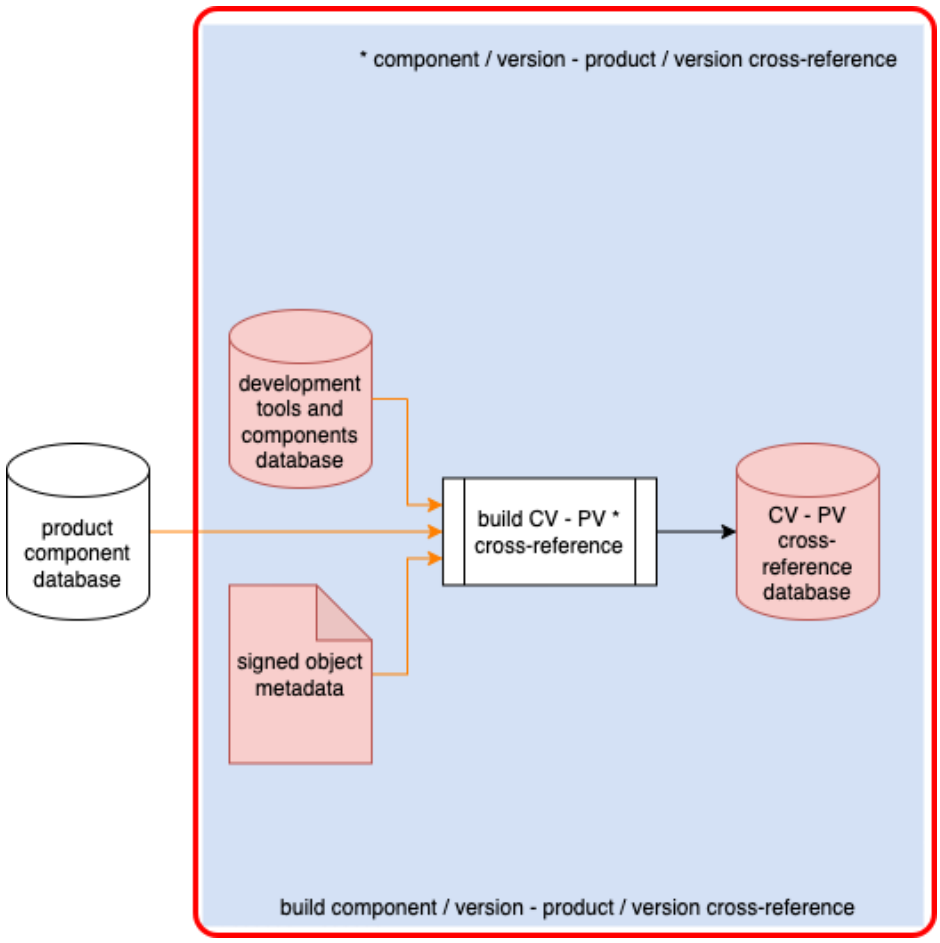
This activity is included because it accesses the **development tools and components database** created by the **List of Approved Tools and Components** process. This process has no direct down-stream impact on the SBOM lifecycle.

Component / Version – Product / Version Cross-reference Document [4]

The **Component / Version – Product Version Cross-reference Document** process is the core of the SBOM lifecycle.



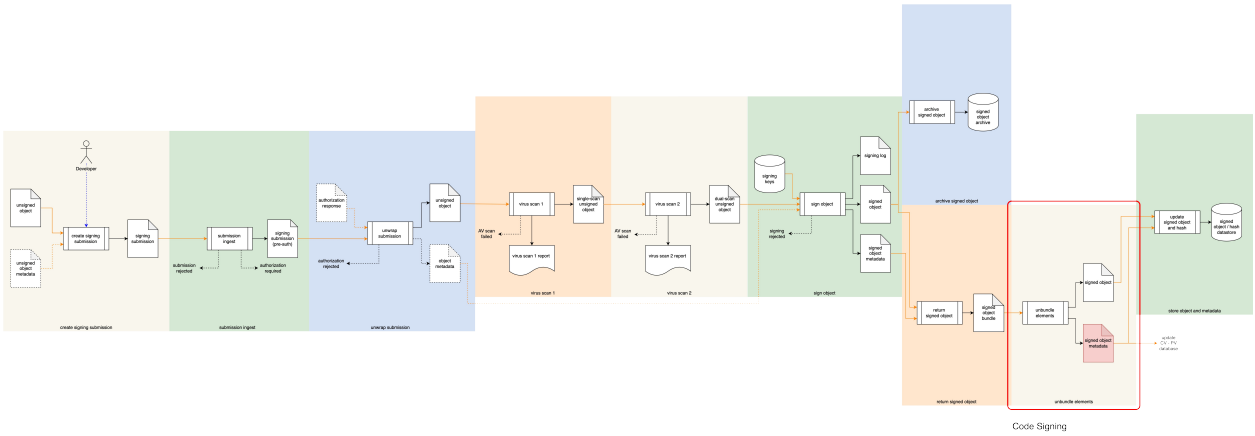
This is the process responsible for the creation and management of all metadata relating to all products and the components they include.



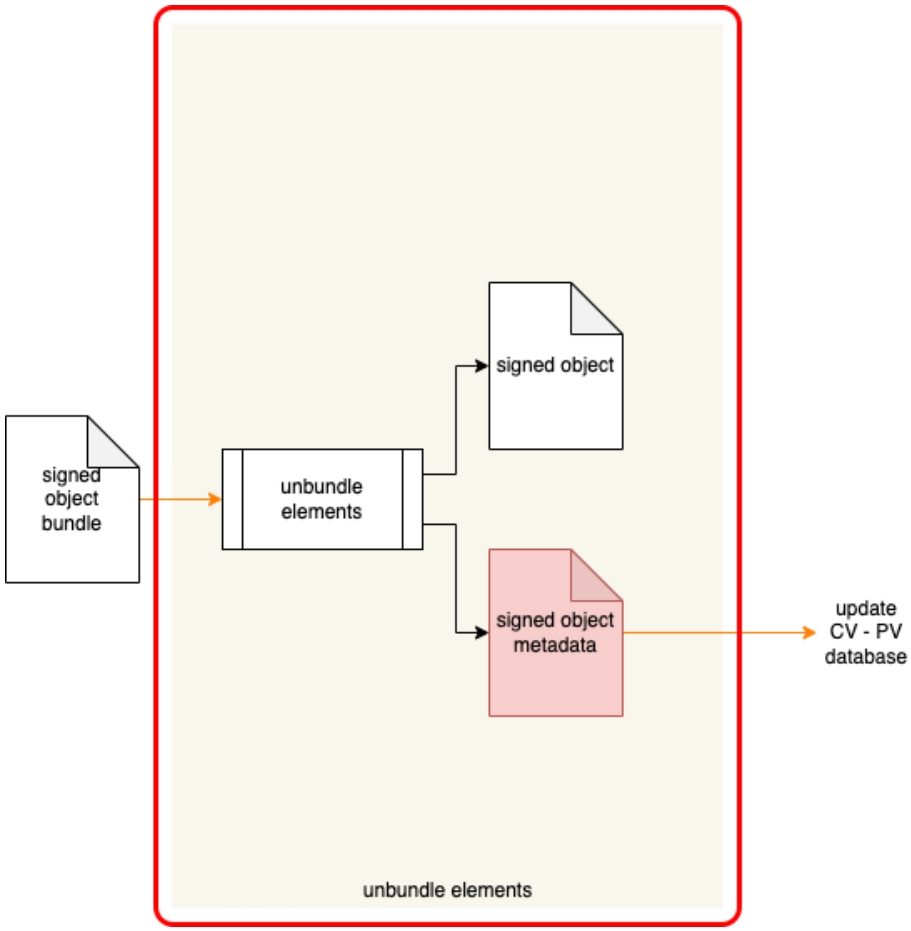
All downstream SBOM lifecycle related processes depend upon the accuracy and completeness of the data stored in the **component / version – product / version cross-reference database**.

Code Signing [11]

The **Code Signing** process is key to providing a means to verify the integrity and creator of an object. Within the context of the SBOM lifecycle, it provides chain of custody support.



The activity of interest is the **unbundle elements** activity near the end of the process.

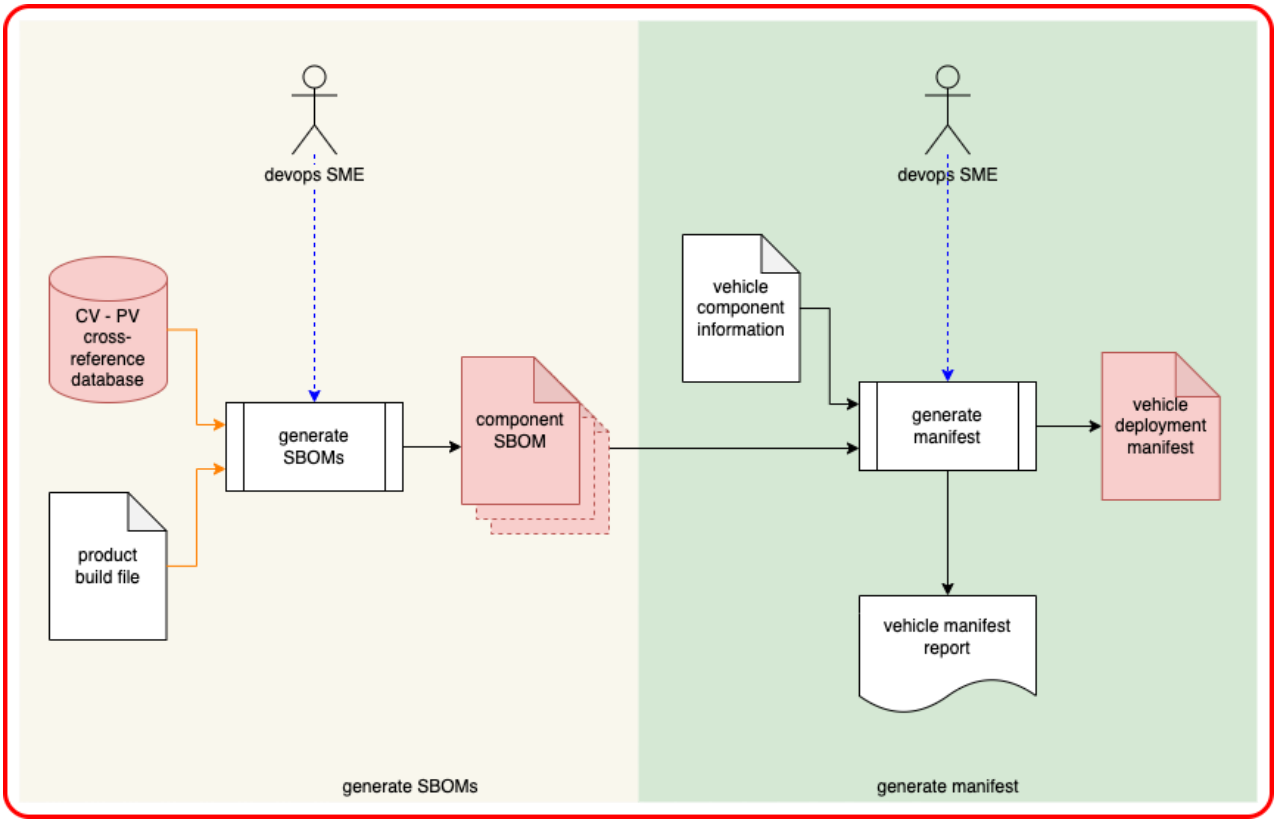


During the **unbundle elements** activity **signed object metadata** is made available. This information is sent to the **update CV – PV data** activity of the **Component / Version – Product Version Cross-reference Document** process. The metadata of interest includes (but not be limited to) HMAC [13] and CMAC [14].

As one could imagine, the information within the **Component / Version – Product Version Cross-reference database** is built up along with the growth of the components being used and products being created.

Manifest Generation [12]

The **Manifest Generation** process is the one people generally focus on in the context of the SBOM.

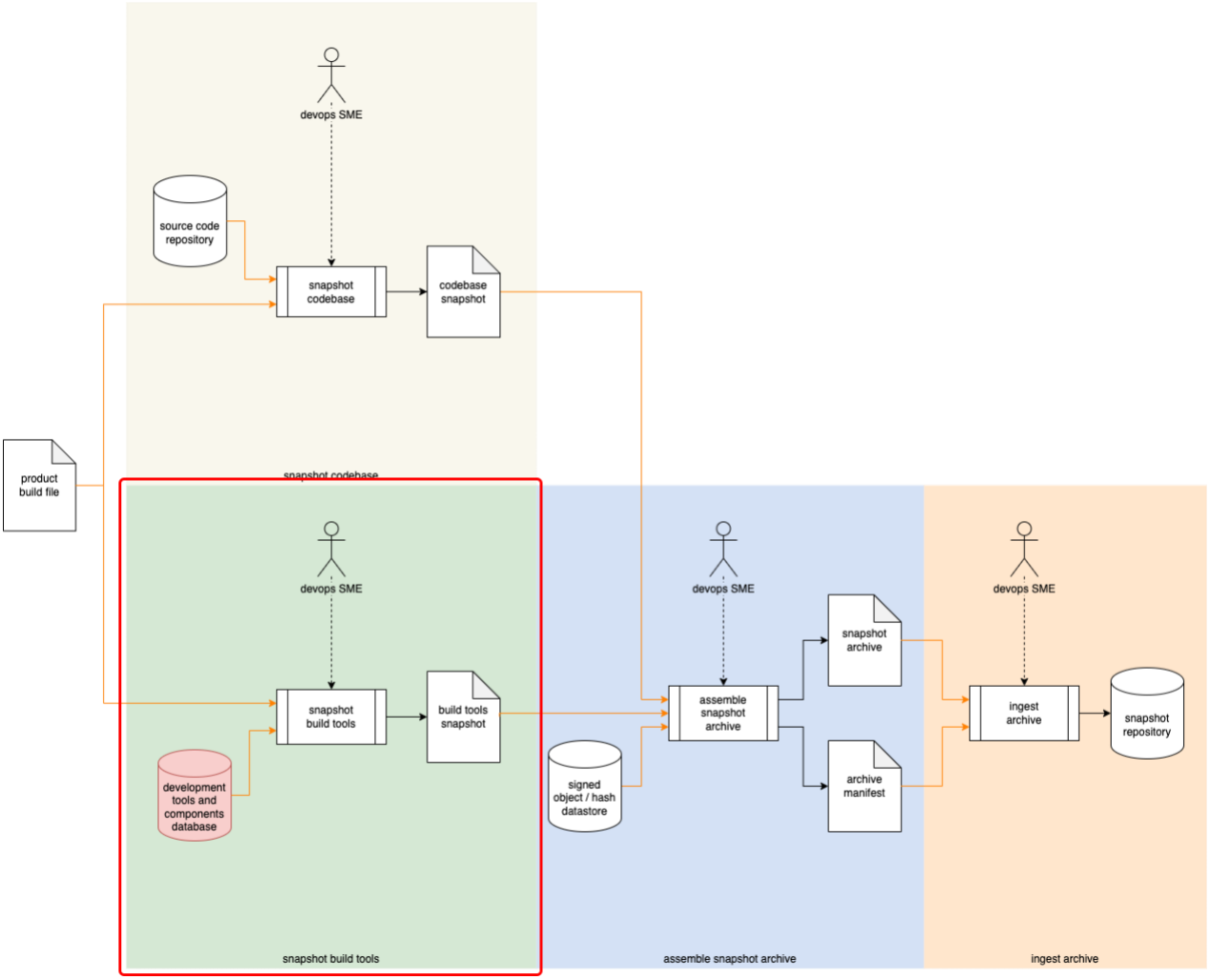


The **component / version – product / version cross-reference database** is used to create a plurality of **component SBOMs**. These are then combined to generate a **vehicle deployment manifest**.

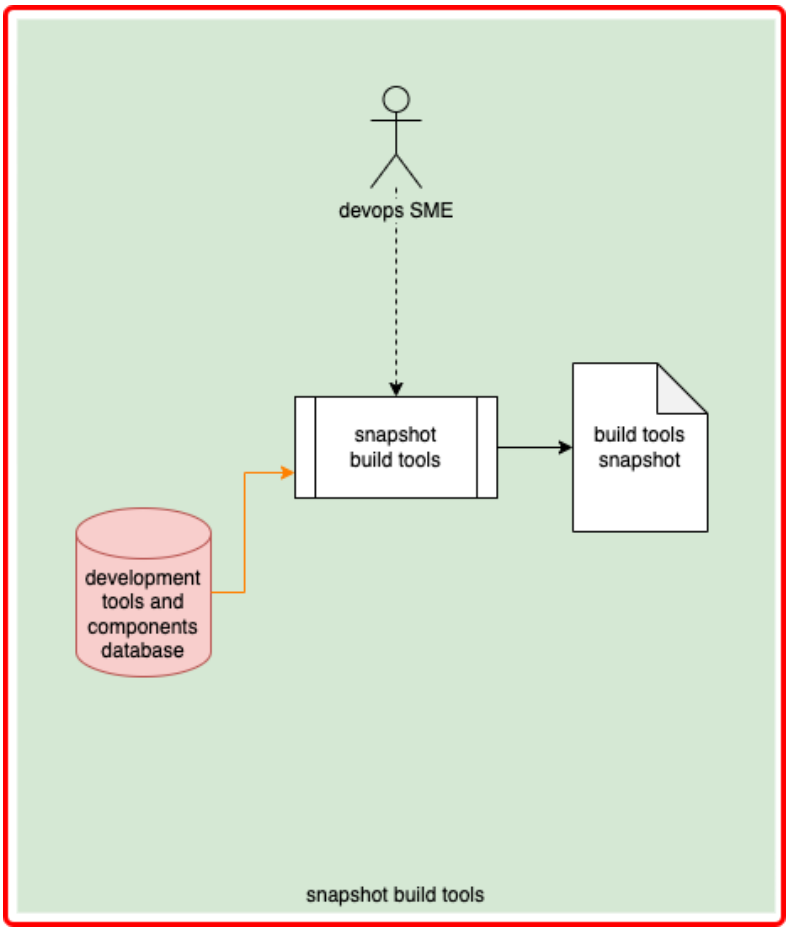
Archive Manifest [2]

Note: This process does not directly impact the SBOM but is of interest in the overall SBOM lifecycle.

The **Archive Manifest** process is responsible for capturing the state of the development build system for auditing, issue mitigation, and disaster recovery purposes.



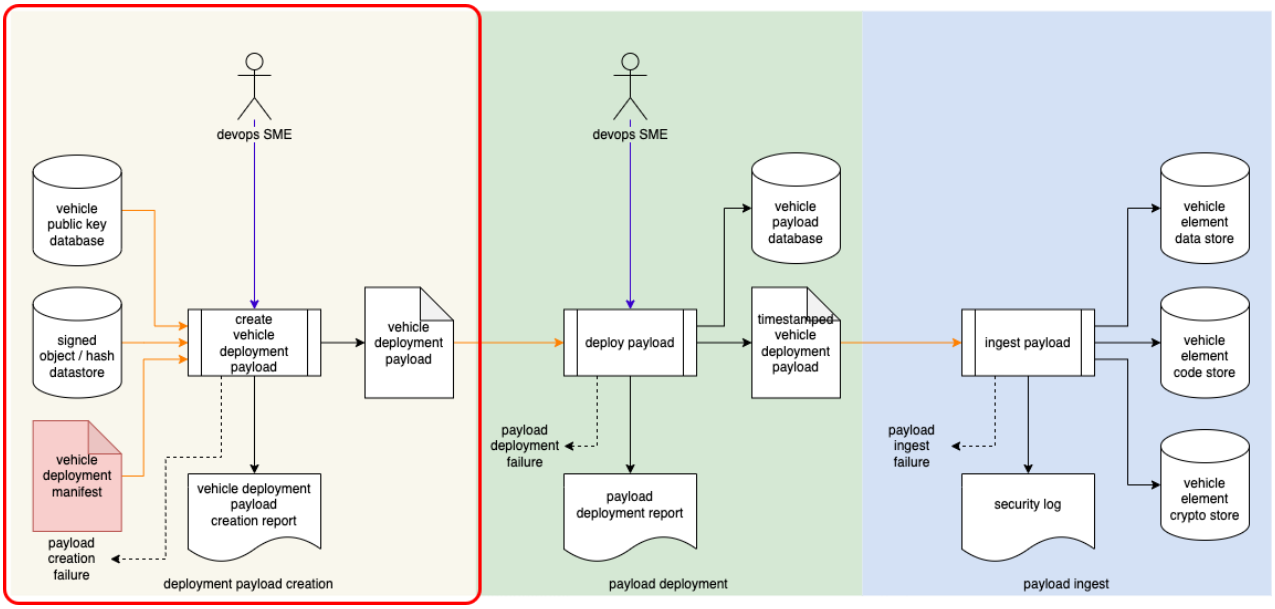
The **snapshot build tools** activity is of interest in this process.



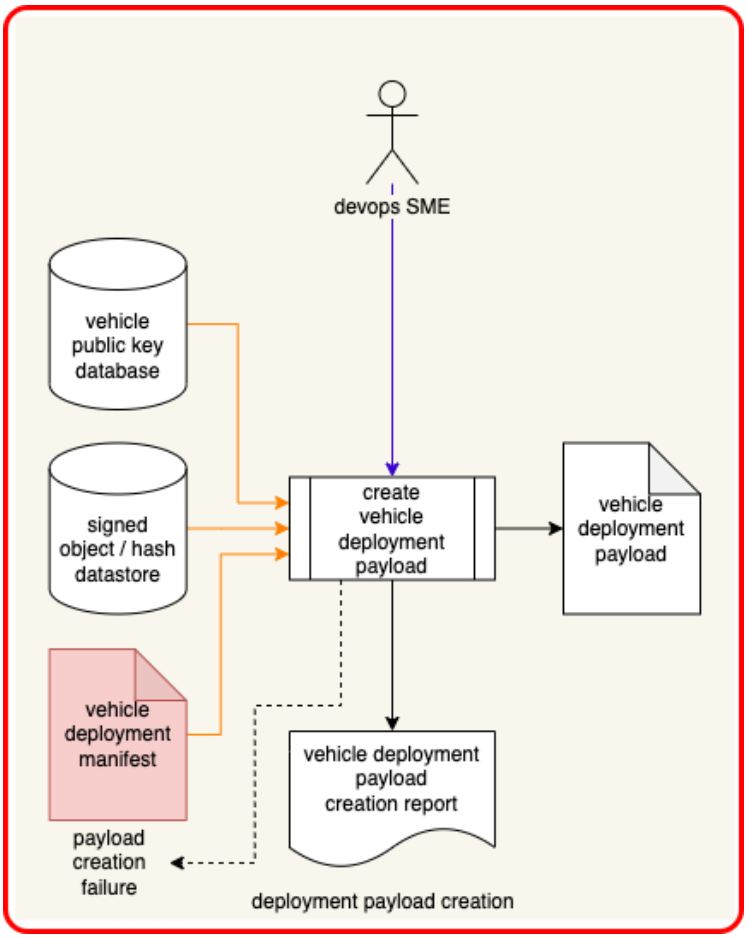
As with the **List of Tools and Components Used** process, the **Archive Manifest** process draws material from the **development tools and components database** created during the **List of Approved Tools and Components** process.

Software Deployment Report [10]

The **Software Deployment Report** process encompasses the deployment of software and configuration information (including the **vehicle deployment manifest**) to the vehicle.



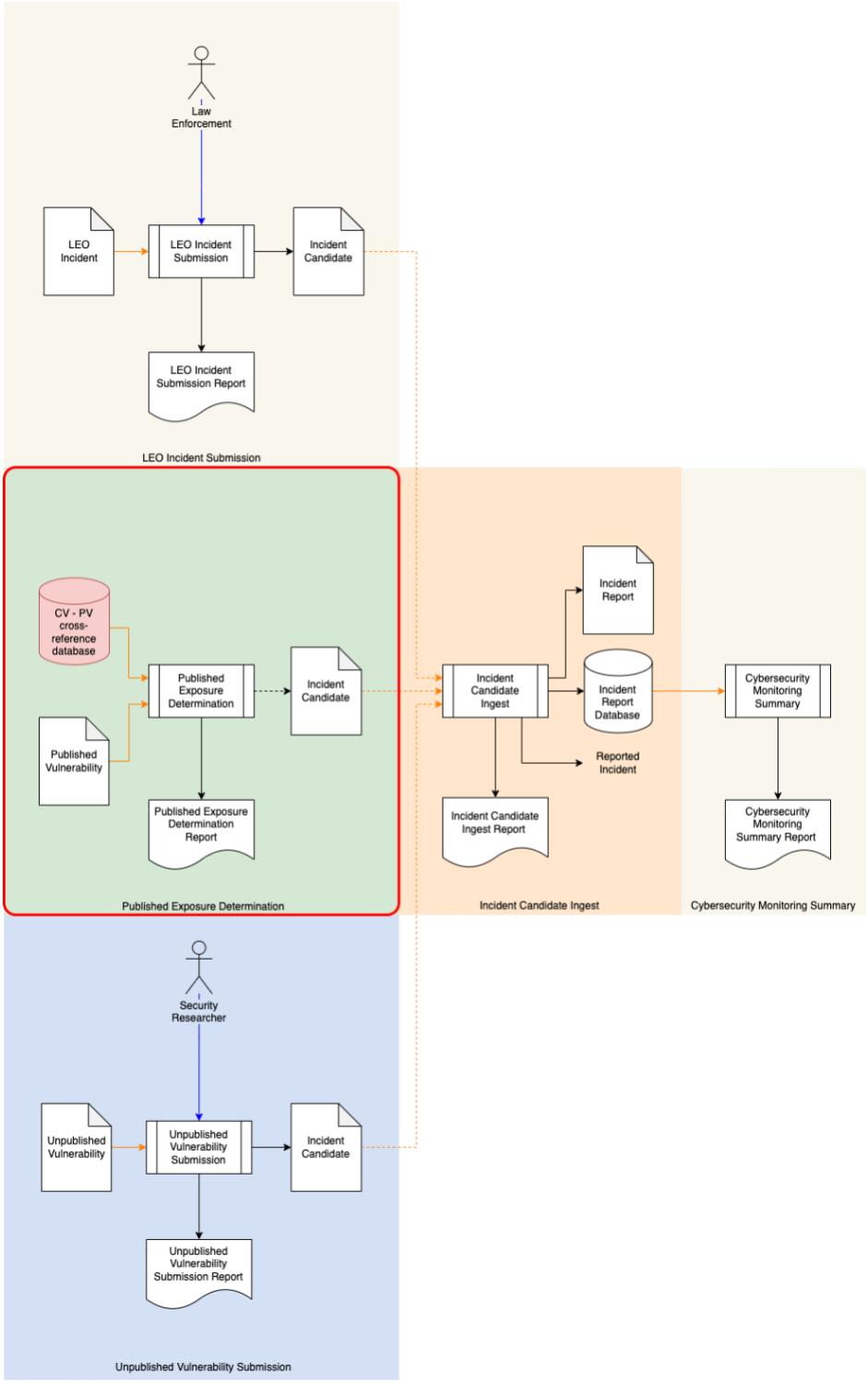
This is the process which hands off the SBOM information to the vehicle. The activity of interest is **deployment payload creation**.



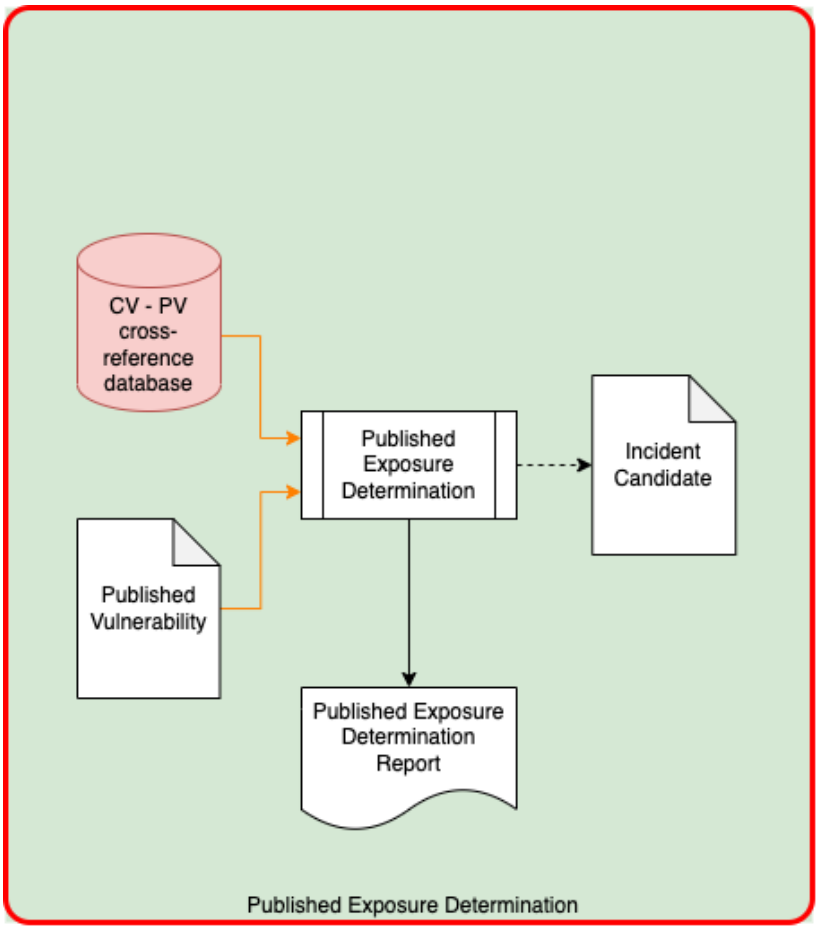
During **deployment payload creation** the **vehicle deployment manifest** (created during the **Manifest Generation** process) is combined with various other elements to produce a **vehicle deployment payload**.

Cybersecurity Monitoring Plan [5]

The **Cybersecurity Monitoring Plan** is the first of the two operational processes. It is used to monitor various sources for possible threats.



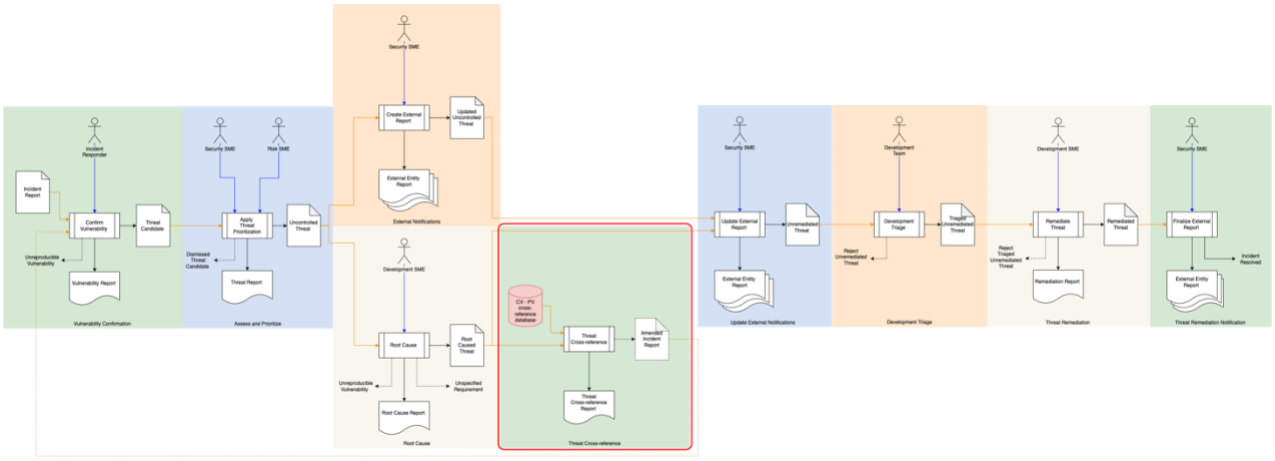
The activity of interest is the **Published Exposure Determination** activity.



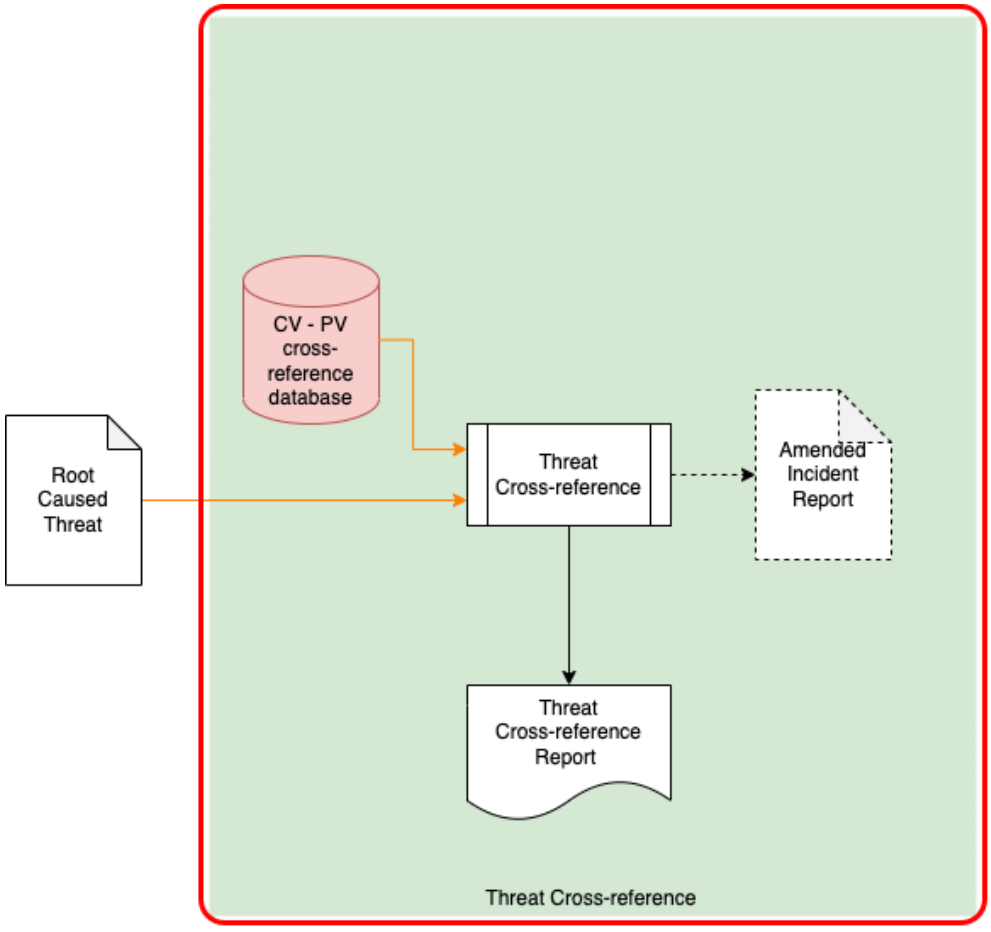
The **Published Exposure Determination** uses the **component / version – product / version cross-reference database** to screen for impactful vulnerabilities.

Incident Response Plan [6]

The **Incident Response Plan** is the second of the two operational processes. It is responsible for handling threats from the field.



The SBOM lifecycle material is used once a threat has been root caused and needs to be cross-referenced.



The **component / version – product / version cross-reference database** is used to determine what the scope of an identified threat is.

References

1. **AVCDL** (AVCDL primary document)
2. **Archive Manifest** (AVCDL secondary document)
3. **Build Process Documentation** (AVCDL secondary document)
4. **Component / Version – Product / Version Cross-reference Document** (AVCDL secondary document)
5. **Cybersecurity Monitoring Plan** (AVCDL secondary document)
6. **Incident Response Plan** (AVCDL secondary document)
7. **List of Approved Tools and Components** (AVCDL secondary document)
8. **List of Tools and Components Used** (AVCDL secondary document)
9. **Release Integrity Plan** (AVCDL secondary document)
10. **Software Deployment Report** (AVCDL secondary document)
11. **Code Signing** (AVCDL elaboration document)
12. **Manifest Generation** (AVCDL elaboration document)
13. **HMAC**
    https://en.wikipedia.org/wiki/HMAC
14. **CMAC**
    https://en.wikipedia.org/wiki/One-key_MAC