AVCDL: The Autonomous Vehicle Cybersecurity Development Lifecycle

Charles Wilson, Principal Engineer, Cybersecurity Development Lifecycle Practice

11/3/20 10:47:00 AM

Category: security-governance

Tags: security, cybersecurity, autonomous vehicles, certification, ISO 21434, ISO 15288, ISO 26262, ISO 12207, AVPDL, AVCDL, NCWF, MSSDL

In my post, **Purpose-driven Security** [8], an approach for the application of security controls was given. In **Aligning the Organization with the AVPDL** [9], the motivation for having an overarching framework where various development lifecycles and coexist was presented.

In **Certifiably Secure: Does It Matter** [10], the case for attaining certification was laid out. In **Traceability: Making the Case for Cybersecurity** [11], I showed why we should attain certification. In **Policy – Process – Procedure: What's in a Name?** [12], the relationships between the major structural components needed to define lifecycle were explored.

In this post all those elements will be brought together in the introduction of a formal autonomous vehicle cybersecurity development lifecycle (**AVCDL**).

Let's Review

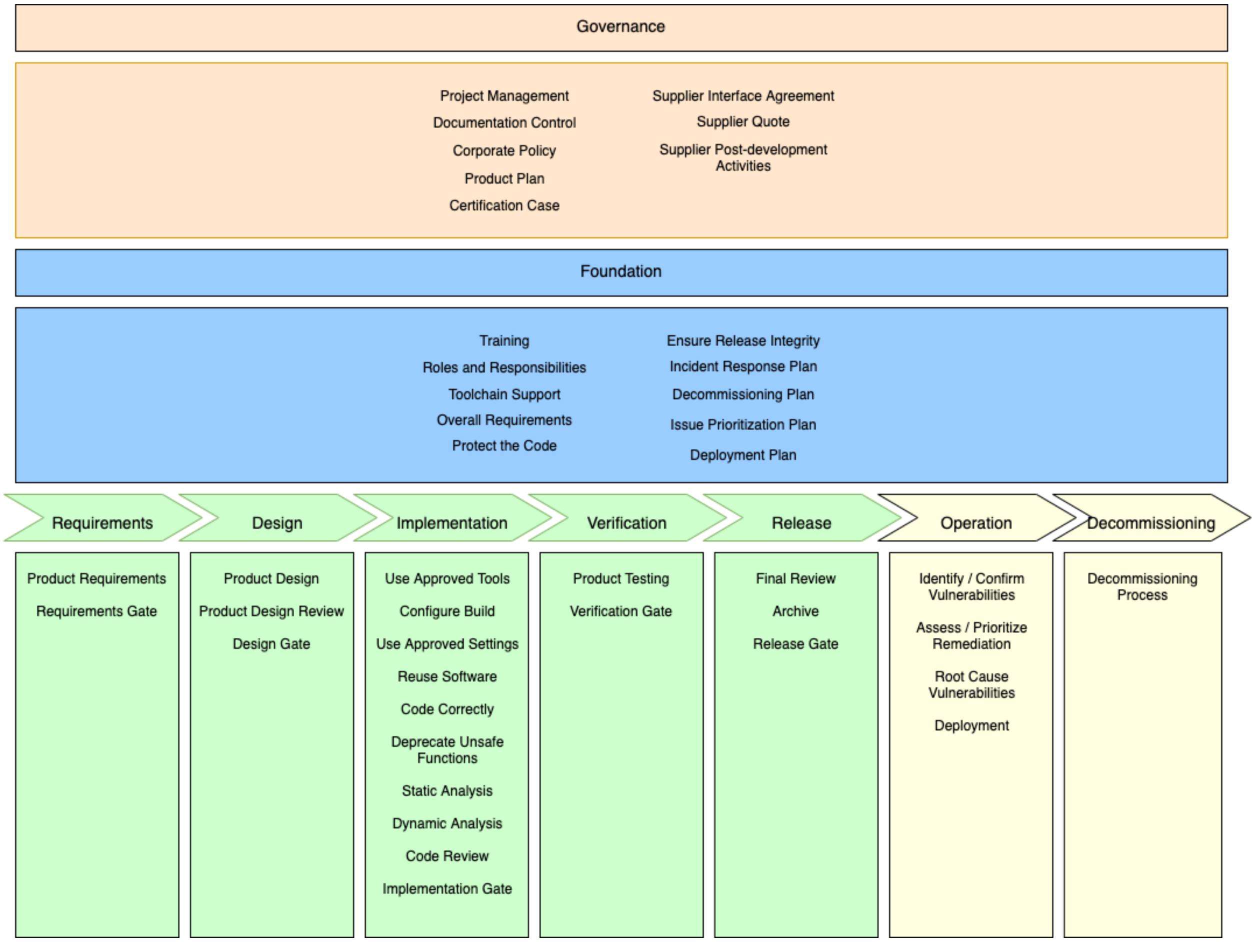
In presenting the **AVPDL** (Autonomous Vehicle Product Development Lifecycle), the four primary autonomous vehicle standards governing the product development lifecycle were introduced. Once again, they are:

Standard	Description
ISO 15288	Systems Development Lifecycle
ISO 12207	Software Development Life Cycle (SDLC)
ISO 26262	Road Vehicles – Functional Safety
ISO 21434	Road Vehicles – Cybersecurity Engineering

With these a loose association matrix was constructed, and an aligning framework created.

AVPDL	15288	12207	26262	21434
organization processes	technical processes	technical processes	management of functional safety	overall cybersecurity management
			supporting processes	project dependent cybersecurity management
foundation phase	N/A	N/A	concept phase	concept phase
requirements phase	requirements definition	requirements definition	safety requirements	cybersecurity requirements
	requirements analysis	system requirements analysis	hazard analysis / risk assessment	cybersecurity assessment
design phase	architectural design	system architectural design	architectural design	cybersecurity design
implementation phase	implementation	implementation	implementation	development
	integration	system integration	integration and verification	integration and verification
verification phase	verification	system qualification testing		
	transition	software installation		
		software acceptance support		
release phase	validation		production	production
operation phase	operation	software operation	operation, service and decommissioning	continuous cybersecurity activities
	maintenance	software maintenance		operation and maintenance
decommissioning phase	disposal	software disposal		decommissioning
supplier processes	agreement processes	agreement processes	supporting processes	distributed cybersecurity activities

Eight phases and two processes were identified and detailed. Then following example lifecycle implementation discussed.



With these as a basis, let’s discuss how they can be applied to product development from the perspective of cybersecurity.

The AVCDL

The **AVCDL** is an **AVPDL**-compatible development lifecycle suitable for use in attaining certification. More specifically, the **AVCDL** in its current form documents how to attain **ISO 21434** [5] certification.

Background Material

The **AVCDL** is based on a combination of industry-proven methodologies and standards bodies' recommendations. Sources include:

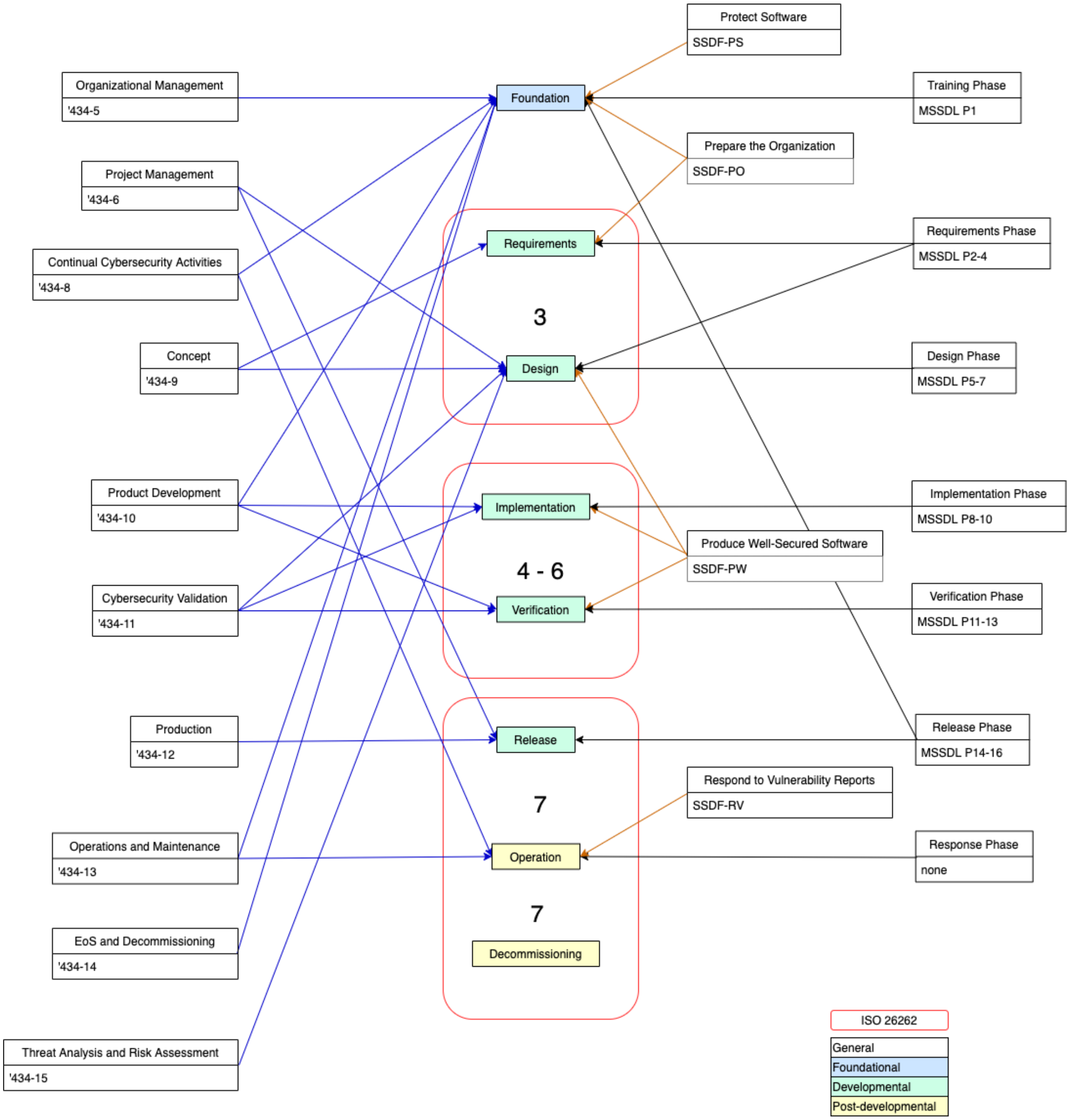
Source	Description
MSSDL	Microsoft Security Development Lifecycle
NIST SSDF	NIST Secure Software Development Framework
ISO 26262	Road Vehicles – Functional Safety
ISO 21434	Road Vehicles – Cybersecurity Engineering

Each of these sources elaborates on a different aspect of product development. The **MSSDL** has been the gold standard for developing secure software for nearly 20 years. In 2019, NIST compiled the **SSDF**, a set of best practices surrounding secure software development.

ISO 26262 is a well-established standard covering vehicle safety. **ISO 21434** is a draft standard addressing vehicle security. I won't get into the specifics of these here as they are covered in greater depth in the **AVCDL** itself.

Contributions Visualized

We can visualize the relationship between the various background sources as follows:



The center column represents the eight phases of the **AVCDL**. Flanking it are **ISO 21434** (left), **SSDF** (right), and **MSSDL** (far right). **ISO 26262** is shown as rounded red rectangles bearing the section numbers from that standard surrounding the **AVCDL** phases (**ISO 26262** section 3 encompasses the **AVCDL** *requirements* and *design* phases). As we can see, the motivating material relationships are not always to the corresponding **AVCDL** phase.

Why Not Just Clone ISO 21434?

The question arises as to why we can't just make a check list from **ISO 21434** and call it a day. Fair question. I get it a lot.

The short answer is that **ISO 21434** isn't a good mapping to the processes typical to a product's development. It has about 40 work products with about 120 requirements split among them. That's a lot to ask.

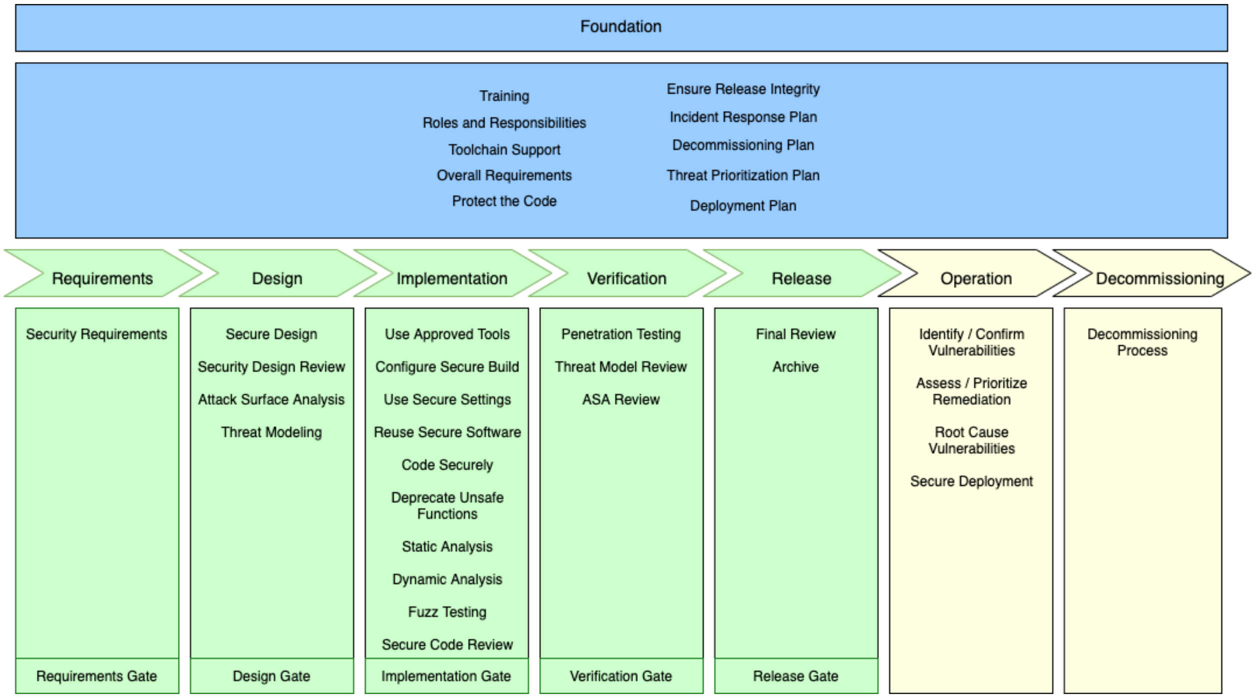
As can be seen from the blue arrows in the above diagram, these don't necessarily apply to the phase in which they are called out. Also, **ISO 21434** doesn't include all the security activities needed in product creation. Finally, **ISO 21434** does not provide phase alignment with other lifecycles (no gates).

Like **ISO 21434**, the **AVCDL** has about 40 phase requirements, but only 46 products. Five of these phase requirements / products are phase gates. As such, these don't really impact the ask made by security to the development team. All of this aligns cleanly to the **AVPDL**.

The implementation of the **AVCDL** puts the task of building the security case into the process itself rather than making it into a giant punch list.

Implementation Framework

Since the **AVCDL** is **AVPDL**-compatible, it shares the same framework with the exception of the governance elements as those are managed at the organizational level. We can visualize the **AVCDL** as follows:



Foundation

Training
Roles and Responsibilities
Toolchain Support
Overall Requirements
Protect the Code

Ensure Release Integrity
Incident Response Plan
Decommissioning Plan
Threat Prioritization Plan
Deployment Plan

Requirements	Design	Implementation	Verification	Release	Operation	Decommissioning
Security Requirements	Secure Design Security Design Review Attack Surface Analysis Threat Modeling	Use Approved Tools Configure Secure Build Use Secure Settings Reuse Secure Software Code Securely Deprecate Unsafe Functions Static Analysis Dynamic Analysis Fuzz Testing Secure Code Review	Penetration Testing Threat Model Review ASA Review	Final Review Archive	Identify / Confirm Vulnerabilities Assess / Prioritize Remediation Root Cause Vulnerabilities Secure Deployment	Decommissioning Process
Requirements Gate	Design Gate	Implementation Gate	Verification Gate	Release Gate		

Comparing the **AVCDL** framework to that of the AVPDL, we can see that most of the phase requirements are specializations. There are also a few additions in the design and verification phases specific to cybersecurity.

What's Inside

The **AVCDL** main document is organized in three sections:

- Introductory material
- Framework (main **AVCDL** material)
- Supplementary material

Introductory Material

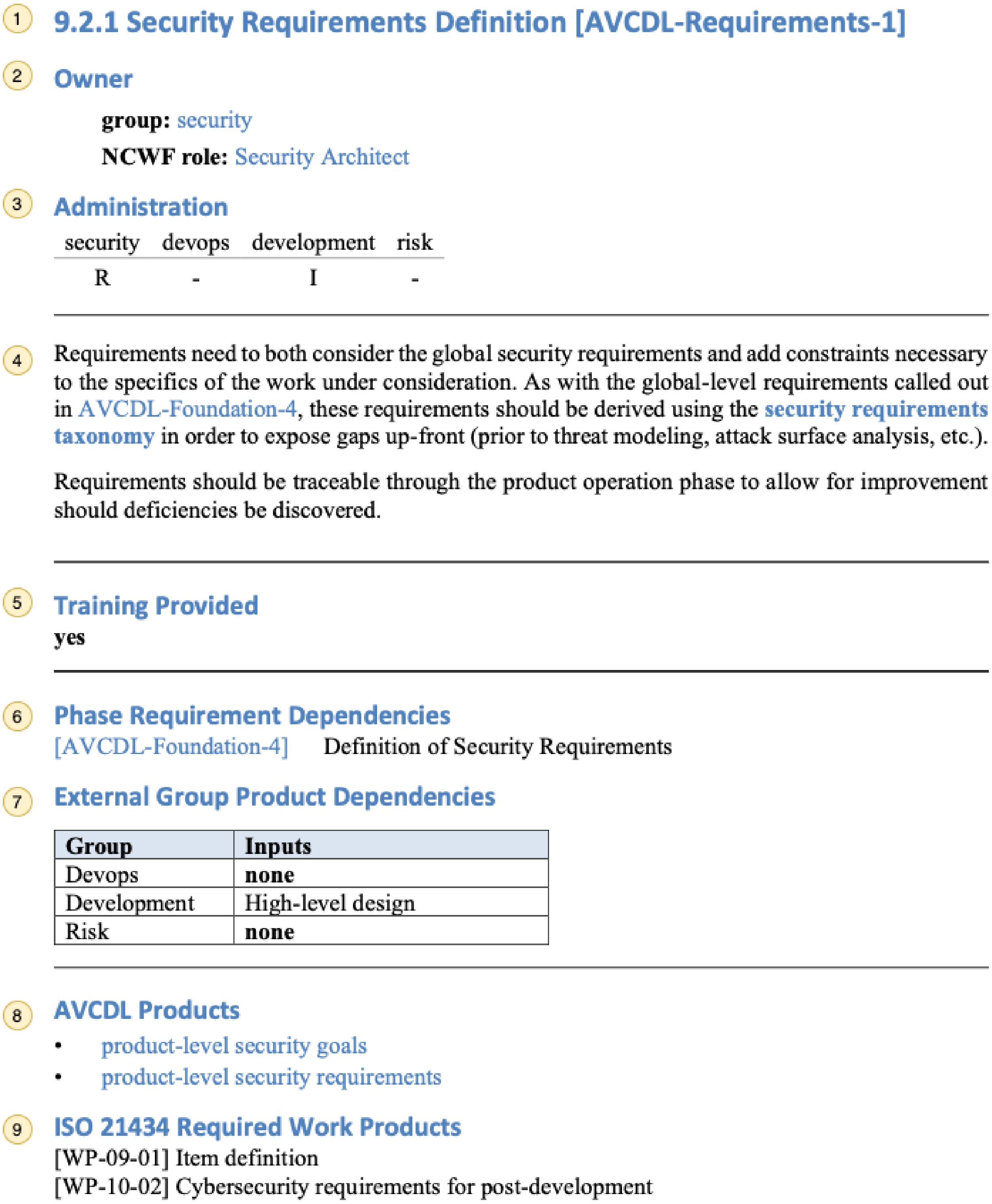
Before diving into the details of the **AVCDL**, a set of introductory information is presented. Included are:

- General overview
- Philosophy
- Background material
- Continuous improvement
- Relationship to certification standards

Framework

The core of the **AVCDL** is presented here. Following a general overview of the framework itself, the phases are examined. Each phase begins with a summary of goals and list of the phase requirements along with the source(s) underpinning them. A roll-up of the **ISO 21434** work products is enumerated.

After the summary, each of the phase requirements is covered in turn. Let’s look at an example.



① 9.2.1 Security Requirements Definition [AVCDL-Requirements-1]

② Owner

group: security

NCWF role: Security Architect

③ Administration

security	devops	development	risk
R	-	I	-

④ Requirements need to both consider the global security requirements and add constraints necessary to the specifics of the work under consideration. As with the global-level requirements called out in AVCDL-Foundation-4, these requirements should be derived using the **security requirements taxonomy** in order to expose gaps up-front (prior to threat modeling, attack surface analysis, etc.).

Requirements should be traceable through the product operation phase to allow for improvement should deficiencies be discovered.

⑤ Training Provided

yes

⑥ Phase Requirement Dependencies

[AVCDL-Foundation-4]	Definition of Security Requirements

⑦ External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	High-level design
Risk	**none**

⑧ AVCDL Products

- product-level security goals
- product-level security requirements

⑨ ISO 21434 Required Work Products

[WP-09-01] Item definition
[WP-10-02] Cybersecurity requirements for post-development

Here's some additional detail regarding each section.

Item	Section	Description
1	Title	The title of the phase requirement and its ID. Each **AVCDL** phase requirement has a unique ID comprised of 'AVDCL,' the phase (here 'requirements') and a sequence number.
2	Owner Group / Role	The group accountable for the activity and the NCWF role. These link to a summary of the particular group's accountable phase requirements and the NIST SP 800-181 workforce job description.
3	RACI	RACI information for the various groups possibly involved in the activity
4	Description	A general description of the activity and its application
5	Training	Whether training is provided for the activity
6	Internal Dependencies	Predecessor AVCDL phase requirements
7	External Dependencies	Non-security group dependent materials
8	Phase Products	Products created as a result of the activity. These are linked to secondary documents providing more specific information into the process needed to create them.
9	21434 Work Products	Specific **ISO 21434** work products / requirements satisfied by the activity.

Supplementary Material

There are two types of supplementary material. The first is background material. Where possible summary information from major sources in provided to allow people to grasp the material without having to slog through the original. These include:

- NCWF (roles needed by **AVCDL**)
- SSDF

Unfortunately, **ISO 21434** [5] and MSSDL [2] must be obtained separately.

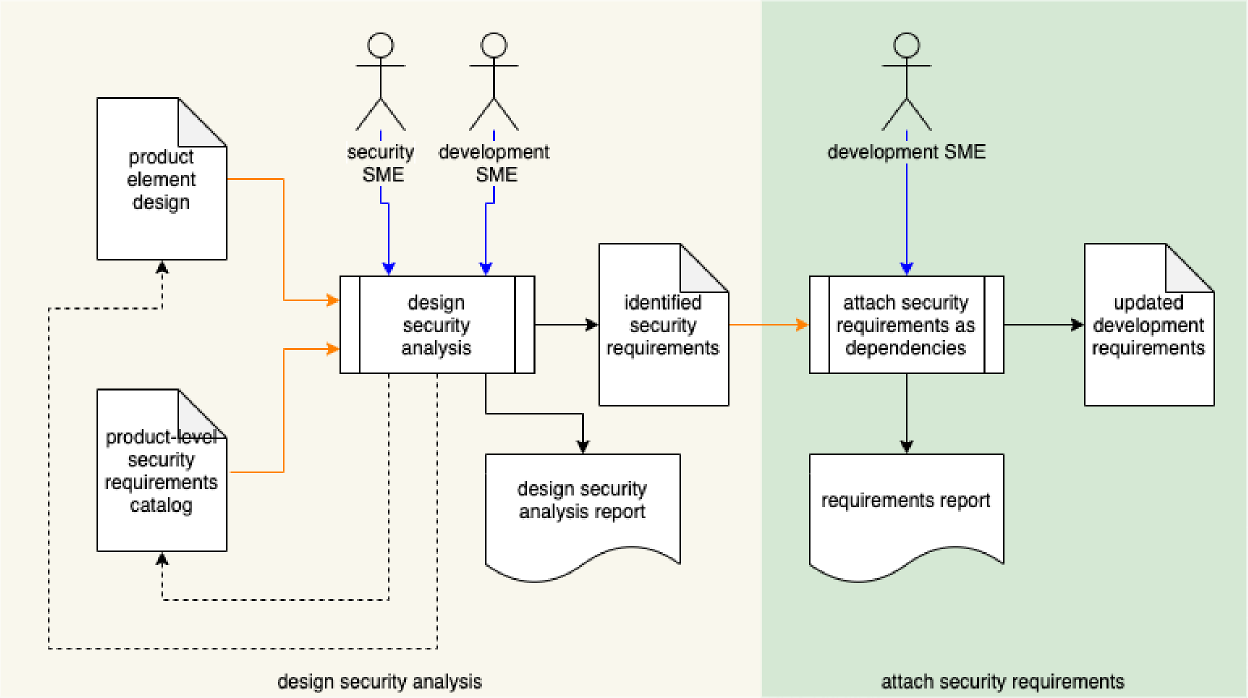
The second type is summary information. Given the sheer number of moving parts, the summaries allow for quick up-take of information. The information summarized includes:

- Requirement role assignments
- Reference documents (standards, secondary documents, working materials [spreadsheets])
- Continuous improvement
- Phase dependencies graph
- Recommended training sequence

Secondary Documents

Several times now I've referenced secondary documents. What are they and why not include them in the **AVCDL** main document?

There is a secondary document for each **AVCDL** phase requirement product. Each document details the process required to create the product. Keep in mind that each is a process and not a procedure document. There may be multiple, distinct implementations of the process. To illustrate this difference, let's look at the workflow for attaching security requirements to a design.


security SME
development SME
development SME
product element design
design security analysis
identified security requirements
attach security requirements as dependencies
updated development requirements
product-level security requirements catalog
design security analysis report
requirements report
design security analysis
attach security requirements

Here we see various inputs, participants, activities, reports, and outputs. All elements are generic. We get the sense of the sequencing.

Specifics not seen include:

- Requirements management system
- Requirements report specification
- Requirements report format

As you might imagine, the more complex the process, the more specifics are at play. The purpose the process documents serve is as a logical verification of methodology that people can agree upon. From this it is anticipated that there exist tertiary documents covering procedures for specific use cases.

Through the use of separation between the process and procedure documents, we are able to share the **AVCDL** widely without requiring that people use specific tools.

There are several reasons as to why the secondary documents are not embedded within the **AVCDL** main document. First, it would be too difficult to maintain. Between the sheer size and the potential for additional internal linking, the document would quickly become a time sink.

Second, updates are more granular and targeted. When updates are necessary and only impact the secondary documentation, people can quickly see if they are impacted.

Third, people don't want to have to sift through multi-hundred-page documents to find things which apply to them. If someone is working on decommissioning protocols for the product, it is very unlikely that they will need information regarding threat modeling. We should be able to give someone exactly what they need to understand the security aspects of the activity they are impacted by.

Learn More

We're releasing the **AVCDL** along with its source material and hope the community will engage with this material by adopting it and contributing improvements. You can find it on GitHub: **https://github.com/nutonomy/AVCDL**.

More to Come

In future posts we'll cover the specifics of the **AVCDL** phase requirement products and how they relate to **ISO 21434**.

References

1. **NIST NICE Workforce Framework for Cybersecurity (NCWF)**
https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center/current-version
2. **Simplified Implementation of the Microsoft SDL**
http://download.microsoft.com/download/F/7/D/F7D6B14F-0149-4FE8-A00F-0B9858404D85/Simplified%20Implementation%20of%20the%20SDL.doc
3. **NIST Secure Software Development Framework (SSDF)**
https://csrc.nist.gov/publications/detail/white-paper/2020/04/23/mitigating-risk-of-software-vulnerabilities-with-ssdf/final
4. **ISO 26262 Road Vehicles – Functional Safety**
https://en.wikipedia.org/wiki/ISO_26262
5. **ISO/SAE DIS 21434 Road Vehicles – Cybersecurity Engineering**
https://www.iso.org/standard/70918.html
6. **ISO/IEC/IEEE 15288 Systems and Software Engineering – System Life Cycle Processes**
https://en.wikipedia.org/wiki/ISO/IEC_15288
7. **ISO/IEC 12207 Systems and Software Engineering – Software Life Cycle Processes**
https://en.wikipedia.org/wiki/ISO/IEC_12207
8. **Purpose-driven Security**
https://github.com/nutonomy/AVCDL/blob/main/background_material/blog%20posts/purpose-driven%20security.pdf
9. **Aligning the Organization with the AVPDL**
https://github.com/nutonomy/AVCDL/blob/main/background_material/blog%20posts/aligning%20the%20organization%20with%20the%20AVPDL.pdf
10. **Certifiably Secure: Does It Matter**
https://github.com/nutonomy/AVCDL/blob/main/background_material/blog%20posts/certifiably%20secure%20-%20does%20it%20matter.pdf
11. **Traceability: Making the Case for Cybersecurity**
https://github.com/nutonomy/AVCDL/blob/main/background_material/blog%20posts/traceability%20-%20making%20the%20case%20for%20certification.pdf
12. **Policy – Process – Procedure: What's in a Name?**
https://github.com/nutonomy/AVCDL/blob/main/background_material/blog%20posts/policy%20-%20process%20-%20procedure%20-%20whats%20in%20a%20name.pdf